

ENERGY-CONSERVING DISCRETIZATION METHODS FOR THE  
INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

*Application to the simulation of wind-turbine wakes*

Benjamin Sandeise



# Energy-conserving discretization methods for the incompressible Navier-Stokes equations

Application to the simulation of wind-turbine wakes

## PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de  
Technische Universiteit Eindhoven, op gezag van de  
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een  
commissie aangewezen door het College voor  
Promoties in het openbaar te verdedigen  
op dinsdag 19 maart 2013 om 16.00 uur

door

Benjamin Sanderse

geboren te 's-Gravenhage

Dit proefschrift is goedgekeurd door de promotor:

prof.dr.ir. B. Koren

A catalogue record is available from the Eindhoven University of Technology library.

ISBN: 978-90-386-3338-1

NUR: 919

The research in this thesis has been supported by the Energy research Centre of the Netherlands (ECN). It was carried out at ECN and Centrum Wiskunde & Informatica (CWI).



Cover design: Joren Vis en kinderen van basisschool de Korenaar te Oostzaan.

© Benjamin Sanderse 2013



## PREFACE

---

This thesis is the result of four years of work at the Energy research Centre of the Netherlands (ECN) in Petten and at the Centrum Wiskunde & Informatica (CWI) in Amsterdam. ECN's practical interest in wind-turbine wake effects in wind farms provided a very fruitful research area for the development of new numerical methods at CWI. Many people have contributed to this thesis and deserve to be acknowledged.

First and foremost I want to thank Barry Koren for offering me this PhD position. His enthusiasm and positive mindset created an excellent working atmosphere at CWI. The freedom he gives to his students is a true privilege.

Secondly, I want to thank ECN, and in particular Peter Eecen and Michiel Houkema, for giving me the opportunity to do my work in Petten, Amsterdam and at NREL in Boulder, Colorado. There must have been moments of doubt when I came up with another Runge-Kutta article instead of a wind-turbine wake simulation.

There are many individuals who helped me during my PhD. Sander van der Pijl was a great help during the start of my work, and taught me a very critical view on CFD methods. Jan Verwer suggested the Gauss methods to me, while standing in the queue of the canteen during lunch break. Willem Hundsdorfer's door was always open for discussions on time integration methods. Piet Hemker was a great help in asking the questions that led to the heart of problems. Jason Frank gave important pointers to literature. Ernst Hairer helped with the analysis with trees. Roel Verstappen and Arthur Veldman hosted me in Groningen to discuss spatial discretization methods and turbulence modeling. Their work was the starting point for my thesis. Matt Churchfield and Pat Moriarty hosted me for two months in Boulder, Colorado. These months were without doubt the most productive of my entire PhD. The trips with Bernard Bulder, the office view and the cycling hours in the Rockies certainly contributed to that. Blair Perot gave many valuable suggestions I had never thought about, and made me realize how much there is still to learn. Kees Oosterlee helped to keep the broader picture in mind.

My office mates, Bram van Es and Willem Haverkort, were good distractions during long office days. Willem, your knowledge of physics and mathematics is truly amazing. It was a pleasure to sit together in the same office for four years. Bram, your Monday morning stories are something to look forward to.

Margreet Nool and Steven van Haren performed most of the parallelization work. Margreet, thanks for all your efforts, and Steven, thank you for your pragmatism.

Many other colleagues were responsible for a nice working environment. At CWI I enjoyed many hours of ping-pong (tournaments), soccer matches, Praethuys drinks and lunch discussions, thanks to Wagner Fortes, Marjon Ruijter, Linda Plantagie, Joost Batenburg, Shashi Jain, Lech Grzelak, Bin Chen, and the PhD students of MAC 1 and MAC 3. The kind help of the CWI staff, especially Nada Mitrovic and

Minnie Middelberg, is very much appreciated.

Having an office next to the coffee machine at ECN was certainly a good distraction, thanks to all the colleagues of the wind department. A special thanks to Hüseyin Özdemir, Arne van Garrel, Marc van Raalte, Henny Bijleveld, Özlem Ceyhan, Francesco Grasso, Jan Willem Wagenaar, Gerard Schepers and Herman Snel.

Als laatste wil ik mijn ouders, mijn broer en Lilian bedanken. Jullie zijn altijd geïnteresseerd in mijn werk en hebben me altijd gesteund.

Benjamin Sanderse  
Amsterdam, January 2013

# CONTENTS

---

1	CFD FOR WIND-TURBINE WAKE AERODYNAMICS	1
1.1	Introduction	1
1.2	Classification of wake models	2
1.3	Wake modeling with CFD	4
1.3.1	The incompressible Navier-Stokes equations	4
1.3.2	RANS	5
1.3.3	LES	7
1.3.4	Numerical issues	10
1.4	Rotor modeling	11
1.4.1	Generalized actuator modeling	11
1.4.2	Direct modeling	16
1.5	Research goal and thesis outline	17
I	SPATIAL DISCRETIZATION	19
2	ENERGY CONSERVATION	21
2.1	Introduction	21
2.2	Continuous energy equation	22
3	DISCRETIZATION ON A STAGGERED GRID	27
3.1	Introduction	27
3.2	Interior discretization	28
3.2.1	General formulation	28
3.2.2	Second order	29
3.2.3	Fourth order	30
3.2.4	Extension to higher order	32
3.3	Numerical experiments	32
3.3.1	Taylor-Green vortex	32
3.3.2	Shear layer roll-up	33
4	BOUNDARY TREATMENT	37
4.1	Conservation properties of the discrete equations	37
4.1.1	Mass	37
4.1.2	Momentum	40
4.1.3	Kinetic energy	47
4.1.4	Extension to higher order	52
4.1.5	Summary	53
4.1.6	Pressure	54
4.2	Order of accuracy analysis	56
4.2.1	Discretization	56
4.2.2	Local truncation error	58
4.2.3	Global truncation error	59
4.3	Numerical experiments	62

4.3.1	1D convection-diffusion equation . . . . .	62
4.3.2	Lid-driven cavity . . . . .	67
4.3.3	Taylor-Green vortex . . . . .	70
4.3.4	Energy conservation in an inviscid cavity . . . . .	70
4.3.5	Turbulent channel flow . . . . .	72
4.4	Conclusions . . . . .	74
II	TEMPORAL DISCRETIZATION . . . . .	75
5	RUNGE-KUTTA METHODS FOR INCOMPRESSIBLE NAVIER-STOKES . . . . .	77
5.1	Introduction . . . . .	77
5.2	Differential-algebraic equations . . . . .	77
5.3	Runge-Kutta methods . . . . .	79
5.3.1	General formulation . . . . .	79
5.3.2	Playing with the pressure . . . . .	80
6	EXPLICIT RUNGE-KUTTA METHODS . . . . .	85
6.1	Introduction . . . . .	85
6.2	Order conditions . . . . .	86
6.2.1	Local and global error . . . . .	86
6.2.2	A short introduction to trees . . . . .	87
6.2.3	Application to the incompressible Navier-Stokes equations . . . . .	89
6.2.4	Half-explicit methods . . . . .	89
6.2.5	Time-dependent operators . . . . .	90
6.2.6	A note on space-time errors . . . . .	91
6.3	The accuracy of the pressure . . . . .	96
6.3.1	Single Butcher tableau (Method 1) . . . . .	96
6.3.2	Reconstruction (Method 2) . . . . .	98
6.3.3	Steady boundary conditions (Method 3) . . . . .	103
6.4	Results . . . . .	103
6.4.1	Taylor-Green vortex . . . . .	104
6.4.2	An actuator disk in an unsteady inflow field . . . . .	106
6.5	Conclusions . . . . .	109
7	IMPLICIT RUNGE-KUTTA METHODS . . . . .	111
7.1	Introduction . . . . .	111
7.2	Formulation for implicit methods . . . . .	113
7.3	Order considerations . . . . .	115
7.4	Energy conservation and time reversibility . . . . .	116
7.4.1	Energy conservation . . . . .	116
7.4.2	Time reversibility . . . . .	118
7.4.3	Other properties . . . . .	120
7.5	New additive Runge-Kutta methods . . . . .	121
7.5.1	Order conditions . . . . .	121
7.5.2	Stability . . . . .	122
7.5.3	Radau IIA/B pair . . . . .	123
7.5.4	Lobatto IIIC/E pair . . . . .	125

7.5.5	DIRK pair . . . . .	125
7.5.6	Summary . . . . .	127
7.6	The accuracy of the pressure . . . . .	127
7.6.1	Radau IIA, Lobatto IIIC . . . . .	129
7.6.2	Gauss . . . . .	129
7.6.3	Lobatto IIIA . . . . .	129
7.6.4	A unified approach . . . . .	130
7.7	Implementation issues . . . . .	132
7.7.1	System of equations . . . . .	132
7.7.2	Splitting . . . . .	133
7.7.3	Linearization . . . . .	134
7.8	Results . . . . .	134
7.8.1	Taylor-Green vortex . . . . .	134
7.8.2	Inviscid shear layer roll-up . . . . .	136
7.8.3	Corner flow . . . . .	142
7.9	Conclusions . . . . .	148
III	ACTUATOR METHODS . . . . .	151
8	IMMERSED INTERFACE METHODS . . . . .	153
8.1	Introduction . . . . .	153
8.1.1	Background . . . . .	153
8.1.2	Approach . . . . .	154
8.2	One-dimensional preliminaries . . . . .	155
8.2.1	Steady . . . . .	155
8.2.2	Unsteady, fixed source . . . . .	158
8.2.3	Unsteady, moving source . . . . .	162
8.2.4	Summary . . . . .	167
8.3	Incompressible Navier-Stokes, stationary forcing terms . . . . .	169
8.3.1	Continuous . . . . .	169
8.3.2	Discrete . . . . .	170
8.3.3	Handling discontinuities . . . . .	171
8.3.4	Extension to fourth order . . . . .	172
8.3.5	Contribution to the energy equation . . . . .	174
8.3.6	Results . . . . .	174
8.4	Incompressible Navier-Stokes, moving forcing terms . . . . .	178
8.4.1	Continuous . . . . .	178
8.4.2	Discrete . . . . .	179
8.4.3	Handling discontinuities . . . . .	180
8.4.4	Results . . . . .	181
8.5	New actuator methods for wind-turbine wake simulations . . . . .	187
8.6	Conclusions . . . . .	189
9	3D FLOW EXAMPLES . . . . .	191
9.1	Parallelization . . . . .	191
9.2	Flow over a wing . . . . .	192

9.3	Flow through an actuator disk . . . . .	193
9.4	Flow in a wind farm . . . . .	194
10	CONCLUSIONS AND RECOMMENDATIONS	197
10.1	Conclusions . . . . .	197
10.1.1	Spatial discretization . . . . .	197
10.1.2	Temporal discretization . . . . .	198
10.1.3	Actuator methods . . . . .	198
10.2	Recommendations . . . . .	199
10.2.1	Spatial discretization . . . . .	199
10.2.2	Temporal discretization . . . . .	200
10.2.3	Actuator methods . . . . .	201
A	DETAILS OF SPATIAL DISCRETIZATION	203
A.1	Boundary stencil for the Poisson equation . . . . .	203
A.2	Boundary contributions to the kinetic energy, fourth order method . . . . .	205
A.2.1	Convective terms . . . . .	205
A.2.2	Diffusive terms . . . . .	209
A.3	Generalized Taylor expansions . . . . .	211
A.3.1	Approximating integrals . . . . .	211
A.3.2	Approximating derivatives . . . . .	212
B	DETAILS OF RUNGE-KUTTA METHODS	213
B.1	W-transformation . . . . .	213
B.2	Energy-conserving (S)DIRK methods . . . . .	214
B.3	Stability of additive Runge-Kutta methods . . . . .	215
B.3.1	Algebraic stability . . . . .	215
B.3.2	Linear stability . . . . .	216
B.4	Partial Runge-Kutta methods . . . . .	217
	Bibliography	221
	Index	237
	Summary	239
	Samenvatting	241
	List of publications	243
	Curriculum Vitae	245

## 1.1 INTRODUCTION

*adapted  
from [154]*

Arguably, there is no better picture to start this thesis than the one shown in figure 1.1. The wakes of the wind turbines in the off-shore wind farm Horns Rev are beautifully visualized due to condensation of water vapor in the wake.



Figure 1.1: Horns Rev wind farm in the North Sea, close to the coast of Denmark.

The grouping of turbines in farms such as Horns Rev introduces two major issues compared to single wind turbines: *reduced power production*, due to wake velocity deficits, and *increased dynamic loads on the blades*, due to increased turbulence levels. Depending on the layout of and wind conditions at a wind farm the power loss of a downstream turbine can easily reach 40% in full-wake conditions, see figure 1.2b. When averaged over different wind directions, losses of approximately 8% are observed for onshore farms, and 12% for offshore farms (see e.g. Barthelmie et al. [12, 13]). Figure 1.2 also reveals that for the full-wake (aligned) condition different simulation models agree rather well with the experimental data, but for non-aligned conditions there is a large discrepancy between experiments and models. It is therefore our intention to develop simulation tools that can better predict wind turbine wakes in farms in order to improve wind turbine designs and optimize wind farm layouts. This is of practical relevance for the Energy research Centre of the Nether-

lands (ECN). Our ‘tentative’ research goal - which will be elaborated in section 1.5 - can be formulated as:

*Accurate and efficient numerical simulation of wind-turbine wakes in wind farms.*

To sharpen this statement (what is ‘accurate’? what is ‘efficient’?) we make an overview of the current approaches to wake modeling, and where they need improvement. This will be done in sections 1.2-1.4. Section 1.5 then summarizes the state-of-the-art and provides the thesis outline.

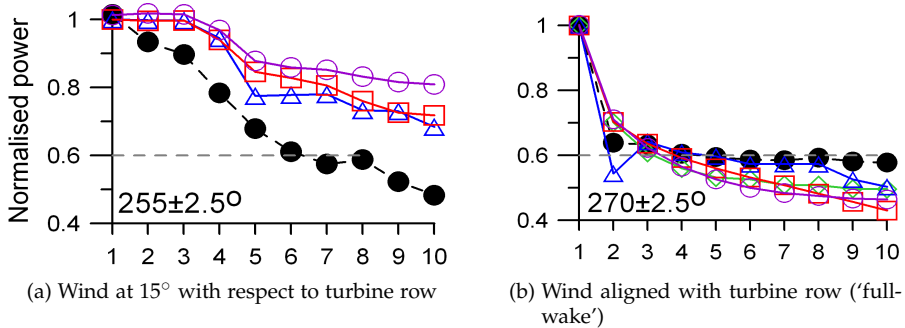


Figure 1.2: Comparison of different models (both engineering and CFD models, open symbols) and experiments (solid dots) in predicting the power output of Horns Rev for two different wind directions. Reproduced from [13].

## 1.2 CLASSIFICATION OF WAKE MODELS

When studying power losses and blade loading, wind-turbine wakes are typically divided into a near and a far wake [199]. The near wake is the region from the turbine to approximately one or two rotor diameters downstream, where the turbine geometry directly affects the flow, leading to the presence of distinct tip vortices. Tip and root vortices lead to sharp gradients in the velocity and peaks in the turbulence intensity. For very high tip-speed ratios the tip vortices form an almost continuous vorticity sheet: a shear layer. The turbine extracts momentum and energy from the flow, causing a pressure jump and consequently an axial pressure gradient, an expansion of the wake and a decrease of the axial velocity. In the far wake the actual rotor shape is only felt indirectly, by means of the reduced axial velocity and increased turbulence intensity. The far wake is the region of interest when studying wind farm aerodynamics. *Turbulence* is the dominating physical process in this part of the wake. Three sources can be identified: atmospheric turbulence (from surface roughness and thermal effects), mechanical turbulence (from the blades and the tower) and wake turbulence (from tip vortex break-down / shear-layer instabilities). Turbulence acts as an efficient mixer, leading to the recovery of the velocity



deficit and a decrease in the overall turbulence intensity. Far downstream the velocity deficit becomes approximately Gaussian, axisymmetric and self-similar. Wake meandering, the large-scale movement of the entire wake, might further reduce the velocity deficit, although it can considerably increase fatigue and extreme loads on a downwind turbine. It is believed to be driven by the large-scale turbulent structures in the atmosphere [49, 89, 90].

The distinction between near and far wake is also apparent when classifying existing numerical models for wind-turbine wake aerodynamics, see table 1.1.

method	blade model	wake model
kinematic	thrust coefficient	self-similar solutions
BEM	actuator disk + blade element	quasi 1D momentum theory
vortex-lattice, -particle	lifting line/surface + blade element	vorticity sheet, particles
panels	surface mesh	vorticity sheet
generalized actuator	actuator disk/line	RANS/LES
direct	volume mesh	RANS/LES

Table 1.1: Classification of models.

The first and simplest approach is an analytical method that exploits the self-similar nature of the far wake to obtain expressions for the velocity deficit and turbulence intensity. The second, Blade Element Momentum (BEM) theory, uses a global momentum balance together with 2D blade elements to calculate aerodynamic blade characteristics. The vortex-lattice and -particle methods assume inviscid, incompressible flow and describe it with vorticity concentrated in sheets or particles. Panel methods similarly describe an inviscid flow field, but the blade geometry is taken into account more accurately and viscous effects can be included with a boundary-layer code; the wake follows as in vortex-wake methods. These four methods have been extensively discussed in previous reviews, such as Vermeer et al. [199], Crespo et al. [39], Snel [170, 171] and Hansen et al. [66]. The last two methods, the generalized actuator disk method and the direct method, are relatively new and are commonly called Computational Fluid Dynamics (CFD) methods. CFD is the most ‘fundamental’ approach, in the sense that the equations of motion (the Navier-Stokes equations) are solved. This approach is therefore expected to give the most accurate results, and it is the approach that we take in this thesis. In the remainder of this chapter we will discuss the state-of-the-art in CFD for wind-turbine wake modeling: section 1.3 discusses the governing equations and turbulence models for the wake and section 1.4 discusses the various ways to model the effect of the rotor on the wake.

### 1.3 WAKE MODELING WITH CFD

#### 1.3.1 *The incompressible Navier-Stokes equations*

It is reasonable to assume that the flow field in wind-turbine wakes is incompressible, since the velocities upstream and downstream of a turbine placed in the atmosphere are typically in the range of 5-25 m/s. Only when calculating the aerodynamics at blade tips compressibility effects may be important. Since in most calculations of wind-turbine wakes the rotor is not modeled directly (this will be discussed in section 1.4), the incompressible Navier-Stokes equations are a suitable model to describe the aerodynamics of wind-turbine wakes:

$$\nabla \cdot \mathbf{u} = 0, \quad (1.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{u}) = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u}, \quad (1.2)$$

supplemented with initial and boundary conditions. The density is assumed to be constant. In the case of a non-neutral atmosphere the Boussinesq approximation is typically employed to account for buoyancy effects, and an extra equation for the temperature has to be solved. The effect of the rotation of the Earth, given by the Coriolis term, is neglected in many wake studies, but can have an effect when computations involve large wind turbines and wind farms (e.g. [132]).

Although this set of equations provides a complete model for the description of turbulent flows, it is not easily solved. The difficulty associated with turbulent flows is the presence of the non-linear convective term, which creates a wide range of time and length scales [40]. For example, in the atmospheric boundary layer the largest turbulent scales are of the order of 1 km, while the smallest scales are of the order of 1 mm [185]. Inside the blade boundary layers the scales are even smaller. The range of scales depends on the Reynolds number (Re), the dimensionless parameter that indicates the ratio of convective forces to viscous forces in the flow. Large values of the Reynolds number, encountered in blade and wake calculations, lead to a large range of scales, making computer simulations extremely expensive. Resolving all scales in the flow, so-called Direct Numerical Simulation (DNS), is therefore not feasible. Turbulence models need to be constructed, modeling the effect of the unresolved small scales based on the behavior of the large scales. However, even with the cost reduction provided by a turbulence model, one cannot resolve both the boundary layers on the turbine blades and the turbulent structures in the wake. This necessitates a simplified representation of the wind turbine in case of wake calculations (and a simplified representation of the wake in case of blade calculations).

A large number of turbulence models have been constructed in the last decennia, see e.g. [210, 143, 58]. This section will discuss the two most important methodologies in turbulence modeling for wind-turbine wakes, namely RANS and LES.

### 1.3.2 RANS

RANS (Reynolds-Averaged Navier-Stokes) methods aim for a statistical description of the flow. Flow quantities such as velocity and pressure are split in an average and a fluctuation, the so-called Reynolds decomposition:

$$\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}) + \mathbf{u}'(\mathbf{x}, t). \quad (1.3)$$

The averaging procedure, ensemble averaging, is such that  $\overline{\bar{\mathbf{u}}(\mathbf{x})} = \bar{\mathbf{u}}(\mathbf{x})$  and  $\overline{\mathbf{u}'(\mathbf{x}, t)} = 0$ . The Reynolds decomposition (1.3) is substituted into the Navier-Stokes equations, which are then averaged, resulting in [40]:

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} \bar{\mathbf{u}}) = -\frac{1}{\rho} \nabla \bar{p} + \nu \nabla^2 \bar{\mathbf{u}} - \nabla \cdot (\overline{\mathbf{u}' \mathbf{u}'}). \quad (1.4)$$

The term  $\overline{\mathbf{u}' \mathbf{u}'}$  is called the Reynolds stress tensor, which appears as a consequence of the non-linearity of the convective term, and represents the averaged momentum transfer due to turbulent fluctuations. The Reynolds stresses can be interpreted as turbulent diffusive forces. In wind-turbine wakes they are much larger than the molecular diffusive forces  $\nu \nabla^2 \bar{\mathbf{u}}$ , except near solid boundaries. In order to close the system of equations, a model is needed to express the Reynolds stresses in terms of mean flow quantities.

A widely adopted approach of modeling the Reynolds stresses exploits the Boussinesq hypothesis [19] (not to be confused with the Boussinesq approximation mentioned earlier). Based on an analogy with laminar flow it states that the Reynolds stress tensor can be related to the mean velocity gradients via a turbulent ‘eddy’ viscosity  $\nu_T$ ,

$$\overline{\mathbf{u}' \mathbf{u}'} = -\nu_T \left( \nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T \right), \quad (1.5)$$

so that the RANS equations (1.4) become:

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + (\bar{\mathbf{u}} \cdot \nabla) \bar{\mathbf{u}} = -\frac{1}{\rho} \nabla \bar{p} + \nabla \cdot \left( (\nu + \nu_T) (\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T) \right). \quad (1.6)$$

This approach of modeling the effect of turbulence as an added viscosity is widely used for turbulent flow simulations. It is very useful as engineering method, because the computational time is only weakly dependent on the Reynolds number. However, the validity of the Boussinesq hypothesis is limited. In contrast to  $\nu$ ,  $\nu_T$  is not a property of the fluid, but rather a property of the type of flow in question. Since eddies are fundamentally different from molecules, there is no sound physical basis for equation (1.5) [210] and DNS calculations have indeed not shown a clear correlation between  $\overline{\mathbf{u}' \mathbf{u}'}$  and  $\nabla \bar{\mathbf{u}}$  [157]. The Boussinesq hypothesis is therefore inadequate in many situations, for example for flows with sudden changes in mean strain rate (e.g. the shear layer of the wake), anisotropic flows (e.g. the atmosphere) and three-dimensional flows [40, 210].

Many different methods have been suggested to calculate  $\nu_T$ , typically called zero-equation (algebraic closure, like mixing length), one-equation and two-equation models (see e.g. Wilcox [210] and references therein). The  $k - \epsilon$  model is an example of a two-equation model often encountered in wind-energy wake applications, the  $k - \omega$  model (with SST limiter) is more convenient near blade surfaces. In the  $k - \epsilon$  model two additional partial differential equations are introduced, one for the turbulent kinetic energy  $k$  and one for the turbulent diffusion  $\epsilon$ . They contain a number of constants that have been determined by applying the model to some very general flow situation (isotropic turbulence decay, flow over a flat plate).

Although the application of the averaging procedure to the Navier-Stokes equations, resulting in the RANS equations, leads to a significant reduction in computational effort, solving equations (1.4) is still a formidable job for wake calculations. One of the reasons for this is the divergence-free constraint in equations (1.1)-(1.2), which requires that the pressure is calculated implicitly via an elliptic equation, the pressure Poisson equation. Two simplifications to this approach, parabolization and linearization, have been suggested and are described in [154]. In this section we continue to describe efforts in literature with the -more complete- elliptical model.

Different researchers found that the standard  $k - \epsilon$  and  $k - \omega$  models result in ‘diffusive’ wakes: the velocity deficits are too small and the turbulence intensity does not show the distinct peaks observed in experiments [139, 26, 27, 47, 133, 136]. The reasons for the failure of these standard turbulence models in wind-turbine wakes were explained by Réthoré [139], and are basically caused by the limited validity of the Boussinesq hypothesis, mentioned before. To improve correspondence with experimental data, several adaptations of the  $k - \epsilon$  model have been suggested. These adaptations all strive to reduce (directly or indirectly) the eddy-viscosity, and hence the diffusion, in the near wake.

Firstly, El Kasmi and Masson [47] added an extra term to the transport equation for the turbulent energy dissipation ( $\epsilon$ ) in a region around the rotor, based on the work of Chen and Kim [36]. This method leads to the introduction of two additional parameters: a model constant and the size of the region where the model is applied. Compared to experimental data, significant improvements over the original  $k - \epsilon$  model were observed. Cabezón et al. [27] and Rados et al. [136] have also applied this approach and found acceptable agreement with other experimental data. Réthoré [139] noticed however that increasing the dissipation proportionally to the production of turbulence contradicts LES results.

A second approach is the realizability model. In this model the eddy viscosity is reduced to enforce that Reynolds stresses respect so-called realizability conditions (see e.g. [158]). According to Réthoré these conditions are *not* satisfied in the near wake by the eddy-viscosity based  $k - \epsilon$  model, due to the large strain rate at the edge of the wake, where the Boussinesq hypothesis is inadequate [139]. Both Réthoré [139] and Cabezón [27] used a realizable model based on the work of Shih [164]. Compared to the standard  $k - \epsilon$  model, better results are obtained, although the prediction of both wake deficit and wake spreading [139] and turbulence intensity [27] remains unsatisfactory.

More adaptations are under investigation. Prospathopoulos et al. [133] adapted the constants of the  $k - \epsilon$  and  $k - \omega$  model for atmospheric flows and investigated the effect of complex terrain. Rados et al. [136] used Freedman's model [54] to change the constants of the  $\epsilon$  equation in case of stable atmospheric conditions. Prospathopoulos et al. [134] obtained good results with Durbin's correction [46], in which the turbulent length scale is bounded. Another possibility comes from an analogy with forest and urban canopy simulations, where obstacles are also modeled as body forces [172, 173]. Réthoré adapted the  $k$  and  $\epsilon$  equations [139] to take into account the extraction of turbulent kinetic energy by the actuator and found significant improvement with respect to the standard  $k - \epsilon$  model.

A fundamentally different approach is the Reynolds stress model (RSM) [91], also called differential second-moment closure model (DSM or SMC), which does not rely directly on the Boussinesq hypothesis. In the RSM all the components of the Reynolds stress tensor are modeled, which makes it suitable for anisotropic flows. However, it leads to six additional PDEs, making the approach expensive. Moreover, these PDEs contain terms which have to be modeled again, and often closure relations resembling the Boussinesq hypothesis are still employed. Lastly, the disappearance of the (stabilizing) eddy-viscosity term can lead to numerical problems. This RSM approach was employed by Cabezón et al. [27] and gave more accurate results than both the El Kasmi-Masson model and the realizability model. However, in predicting the velocity deficit in the near wake the parabolic UPM-PARK code outperformed the elliptic models [26], because in the parabolic code the streamwise diffusive effects are totally neglected. In the far wake the velocity deficit is predicted similarly to the elliptic models, but the turbulence intensity is overestimated and does not show the distinct peaks.

In case of stratification the production of turbulence due to buoyancy has to be taken into account in turbulence closure models - see [154] and references therein.

### 1.3.3 LES

In recent years LES (Large Eddy Simulation) is receiving more attention in the wind-energy wake community, due to its ability to handle unsteady, anisotropic turbulent flows dominated by large-scale structures and turbulent mixing. This is a significant advantage over RANS methods, but the drawback is that the computational requirements for LES are much higher than for RANS. In LES the large eddies of the flow are calculated while the eddies smaller than the grid are modeled with a sub-grid scale model. This is based on the assumption that the smallest eddies in the flow have a more or less universal character that does not depend on the flow geometry. Mathematically this scale separation is carried out by spatially filtering the velocity field, splitting it in a resolved (also called large-scale, simulated, or filtered) velocity

and an unresolved (small-scale) part. In general this filtering operation is defined as a convolution integral:

$$\tilde{\mathbf{u}}(\mathbf{x}, t) = \int \mathbf{u}(\boldsymbol{\zeta}, t) \mathbf{G}(\mathbf{x} - \boldsymbol{\zeta}, \Delta) d\boldsymbol{\zeta}, \quad (1.7)$$

where  $\mathbf{G}(\mathbf{x} - \boldsymbol{\zeta}, \Delta)$  is the convolution kernel, depending on the filter width  $\Delta$ . The sub-grid velocity is then defined as the difference between the flow velocity and the filtered velocity:

$$\mathbf{u}'(\mathbf{x}, t) = \mathbf{u}(\mathbf{x}, t) - \tilde{\mathbf{u}}(\mathbf{x}, t). \quad (1.8)$$

This decomposition resembles that of Reynolds-averaging, but with the difference that in general  $\tilde{\tilde{\mathbf{u}}} \neq \tilde{\mathbf{u}}$  and  $\tilde{\mathbf{u}}' \neq 0$ . Applying the filtering operation to the Navier-Stokes equations (and assuming certain properties of the filter) leads to the following equation:

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} + \nabla \cdot (\tilde{\mathbf{u}} \tilde{\mathbf{u}}) = -\frac{1}{\rho} \nabla \tilde{p} + \nu \nabla^2 \tilde{\mathbf{u}} - \nabla \cdot (\tilde{\mathbf{u}} \tilde{\mathbf{u}} - \tilde{\mathbf{u}} \tilde{\mathbf{u}}). \quad (1.9)$$

As in the RANS equations (1.4) a new term appears, the subgrid-scale (SGS) stresses. These stresses represent the effect of the small (unresolved) scales on the large scales. A widely used model to calculate these stresses is the Smagorinsky model [169], which employs the Boussinesq hypothesis again:

$$\boldsymbol{\tau}_{SGS} = \tilde{\mathbf{u}} \tilde{\mathbf{u}} - \tilde{\mathbf{u}} \tilde{\mathbf{u}} = -\nu_{SGS} \left( \nabla \tilde{\mathbf{u}} + (\nabla \tilde{\mathbf{u}})^T \right). \quad (1.10)$$

Possible ways to calculate the subgrid-scale eddy-viscosity  $\nu_{SGS}$  are to use an analogy of the mixing-length formulation or to use one- or two-equation models involving kinetic energy and turbulent dissipation. Due to use of the Boussinesq hypothesis similar limitations as in RANS are encountered. A large number of other subgrid-scale models have therefore been proposed, for example dynamic models, regularization models and variational multi-scale models (see e.g. [143, 58]).

In contrast to RANS, where the computational cost is only weakly dependent on  $Re$ , the computational cost of LES scales roughly with  $Re^2$ . Near solid boundaries, where boundary layers are present, LES is extremely expensive because it requires refinement in three directions, whereas RANS only requires refinement in the direction normal to the wall. A possibility is to employ a hybrid approach: RANS to resolve the attached boundary layers and LES outside the wall region, so-called Detached Eddy Simulation [179, 180]. Since equations (1.5) and (1.10) both have a similar form, this switch between RANS and LES can be made by changing the eddy viscosity based on a wall-distance function.

As mentioned before, LES has the advantage over most RANS models that it is better able to predict the unsteady, anisotropic turbulent atmosphere. Jimenez et al. [81] used a dynamic sub-grid scale model with the rotor represented by a uniformly loaded actuator disk. A good comparison with experiments was found. The same

model was used in [82] to study the spectral coherence in the wake and in [80] to study the wake deflection due to yaw; in both cases reasonable agreement with experiments and an analytical model was observed.

Troldborg et al. [191, 190, 192] used a vorticity-based mixed-scale sub-grid scale model with the actuator line technique to study the influence of shear and inflow turbulence on wake behavior. The atmospheric boundary layer is imposed with the force field technique of Mikkelsen [112] and atmospheric turbulence is generated with Mann's method. It was found that shear and ground effects cause an asymmetric development of the wake with larger expansions upward and sideward than downward. Inflow turbulence, when compared with laminar inflow, destabilized the vortex system in the wake, resulting in an improved recovery of the velocity deficit, but also an increased turbulence intensity. Furthermore, it was found that the wake is also unstable and becomes turbulent for uniform, laminar inflow, especially at low tip-speed ratios.

In the work of Ivanell [76] the same LES model was used for farm simulations. Twenty non-uniformly loaded actuator disks, in combination with periodic boundary conditions, were used to simulate the 80 turbines that comprise the Horns Rev wind farm (figure 1.3a). The power production of downstream turbines agreed reasonably well with experimental data (figure 1.3b).

Meyers and Meneveau [108] performed LES with a (non-dynamic) Smagorinsky model to study infinite arrays of wind turbines in staggered and non-staggered arrangement, and it was observed that the staggered arrangement had a higher power production. Porté-Agel et al. [132, 211] employ a Lagrangian scale-dependent SGS model, a parameter-free model that performs better in ABL simulations than the traditional Smagorinsky and standard dynamic models [182]. Similar to what Réthoré [139] shows for the  $k - \epsilon$  model, Yu and Porté-Agel [211] show that the Smagorinsky coefficient is not constant in the wake, but increases in the center of the wake and decreases near the ground and in the shear layer. With an actuator-type approximation for the turbine this model provides a very good agreement with atmospheric wind tunnel data [32]. Furthermore, the effect of a wind turbine on a stable atmospheric boundary layer was investigated, demonstrating that the momentum and buoyancy flux at the surface are reduced and as a result possibly influence the local meteorology.

A comparison between RANS (standard  $k - \epsilon$ ) and LES (method of Bechmann [14]) was made by Réthoré [139] in order to explain the aforementioned discrepancies between the  $k - \epsilon$  model and experimental data. The LES results are clearly superior to RANS when comparing both mean velocity profiles and turbulence quantities, but the computational time increases from hours (RANS) to days (LES) for a single wake case. In the study of Stovall et al. [183] the standard  $k - \epsilon$  model and LES with a one-equation SGS model were compared. Again it was observed that the LES results are closer to experimental data in a single wake calculation, and that the wake recovery due to turbulent mixing with the outer flow is much better captured by LES in a multiple wake situation. The difference in computational time was a factor 60, when the same grid for both RANS and LES was used.

However, a well-resolved LES should require a much finer grid than RANS, making it even more expensive.

An estimate of the necessary grid resolution can be obtained based on the Reynolds number. For a sufficiently resolved LES one needs cell sizes far enough in the inertial subrange, i.e., of the order of the Taylor microscale. Since the Taylor microscale scales with  $Re^{-1/2}$ , and the Reynolds number based on the diameter of a modern-size wind-turbine is  $\mathcal{O}(10^8)$ , this leads to the requirement of cell sizes around  $1\text{ cm}^3$ . This is somewhat conservative; for atmospheric flows (based on an integral length scale of  $1\text{ km}$ ) the Taylor microscale was estimated at  $10\text{ cm}$  [212]. Most of the LES computations to date use cell sizes of approximately  $1\text{--}10\text{ m}$ , which might be too coarse to resolve enough scales. Computations require typically  $10^7\text{--}10^8$  grid points and run on supercomputers for several days or weeks, even for single wake situations. With the continuous advance in computer power finer meshes will be possible, but the associated increase in data analysis will leave this approach unattractive for many engineering purposes.

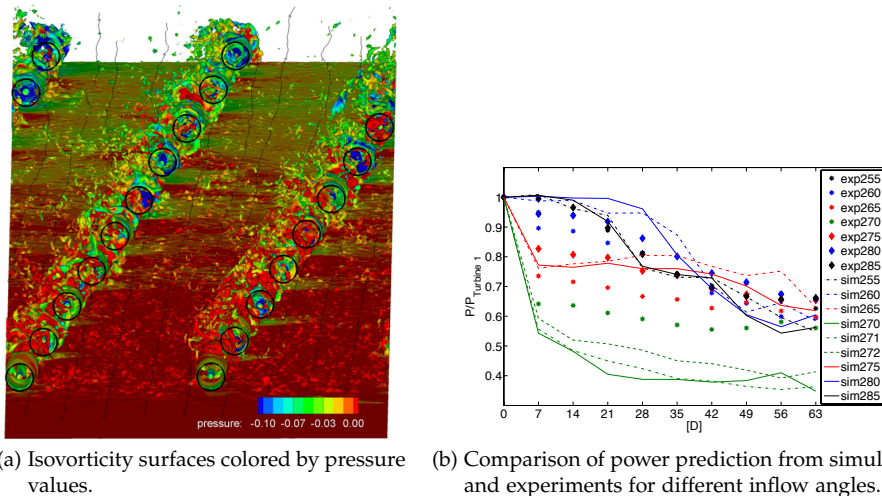


Figure 1.3: LES of the flow through 20 actuator disks, representing the Horns Rev wind farm. Reproduced from [76].

#### 1.3.4 Numerical issues

Most RANS codes use second-order accurate finite volume schemes on structured meshes, with upwind discretization of the convective terms and central discretization of the diffusive terms, leading to stable and robust schemes. Implicit methods are normally used to find a steady-state solution. In LES, the temporal and spatial discretization of the convective terms should be done more carefully. The use of upwind schemes for the spatial discretization can influence the energy cascade from large to small scales due to the introduction of numerical dissipation [103].



Several authors have indeed mentioned premature turbulence decay and related it to possible numerical dissipation, see e.g. [14, 192, 183]. This forms the rationale for the application of central or spectral schemes, see e.g. Bechmann [14] and Meyers et al. [108]. Apart from the spatial discretization the temporal discretization can also introduce numerical dissipation. In [14] this limits the time step, partly removing the advantages of an implicit method. High-order low-dissipative explicit schemes, such as the standard four-stage Runge-Kutta method, used in [28], can then be an attractive alternative.

Lately, high-order (typically fourth order) schemes have also been employed for the spatial discretization [14, 108], reducing interference between the sub-grid scale model and the discretization. A blend with third order upwind schemes is sometimes made to ensure numerical stability [190, 14, 76]. However, it should be realized that the formal order of accuracy of a discretization scheme is only obtained in the limit of sufficiently high spatial resolution, something typically not encountered in LES of wind turbine wakes. Mesh refinement studies are often computationally too expensive, and one has to rely on the energy spectrum to check if the inertial sub-range is captured well. Furthermore, partial cancellation of sub-grid modeling and numerical errors may occur, leading to the counterintuitive conclusion that high-order accurate schemes, improved sub-grid models or finer meshes may lead to worse results [143, 58, 107].

To conclude, when performing LES simulations of wind-turbine wakes, numerical methods are needed that are non-dissipative (in space and time) and stable, even on coarse grids.

## 1.4 ROTOR MODELING

To solve the RANS equations (1.4) or LES equations (1.9) in the near and far wake of a wind turbine, a representation of the blades is necessary. Basically, two approaches exist: the generalized actuator disk approach, in which the blades are represented by a body force (section 1.4.1), or the direct approach, in which the presence of the blades is taken into account by discretizing the actual blades on a computational mesh (section 1.4.2).

### 1.4.1 *Generalized actuator modeling*

The first work on actuator disks was probably by Rankine [138] and extended by R.E. Froude [55]. The concept of a disk that changes the momentum and energy of a fluid is often used to derive the optimum power coefficient of a wind turbine and the corresponding Betz limit. When the actuator concept is used together with solving the Navier-Stokes equations, it is known as the generalized actuator model, and the forcing terms appear as a source term in the momentum equations. These forces are assumed to be known or are calculated based on the local flow field. In this way the computational requirements associated with constructing a body-fitted mesh

and resolving boundary layers on the body can be greatly alleviated. Examples of applications are, besides wind turbines, propellers on aircraft and on ships [205], helicopter rotors, and insect or bird wings [48, 119]. For a general introduction to actuator disk theory see [71].

An actuator acts as a momentum sink which is explicitly added to the momentum equations (1.2):

$$\int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} d\Omega + \int_{\partial\Omega} \mathbf{u} \mathbf{u} \cdot \mathbf{n} dS = - \int_{\partial\Omega} \frac{1}{\rho} p \mathbf{n} dS + \int_{\partial\Omega} \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \cdot \mathbf{n} dS + \int_{A \cap \Omega} \mathbf{f} dA, \quad (1.11)$$

which are written in weak form, because the force leads to a discontinuity in pressure. Apart from this momentum sink, one should also introduce sources of turbulence corresponding to the mechanical turbulence generated by the blades. Currently, three different approaches for prescribing the force term  $\mathbf{f}$  exist: the actuator disk, actuator line and actuator surface models, see figure 1.4.

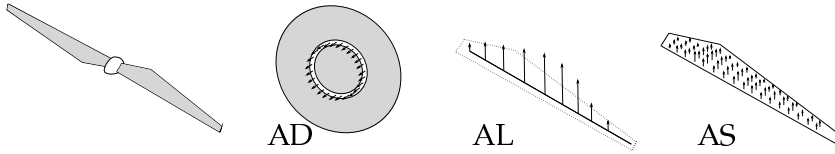


Figure 1.4: Illustration of the actuator disk (AD), line (AL) and surface (AS) concept.

#### 1.4.1.1 Actuator disk

In case of a uniformly loaded actuator disk,  $\mathbf{f}$  acts on the rotor-disk surface  $A$  and is usually expressed in terms of the thrust coefficient  $C_T$  only:

$$\rho \mathbf{f} = \frac{1}{2} \rho V_{\text{ref}}^2 C_T \mathbf{e}_x, \quad (1.12)$$

where the axis of the disk is assumed to be parallel to the  $x$ -axis. The determination of the reference velocity  $V_{\text{ref}}$  in order to calculate  $C_T$  is not obvious. For a turbine facing the undisturbed flow,  $V_{\text{ref}}$  is evidently  $V_{\infty}$ , but for a turbine in the wake of an upstream turbine or in complex terrain this is not the case. Prospathopoulos et al. [134] proposed an iterative procedure to obtain the reference velocity and the thrust coefficient for downstream turbines modeled as actuator disks: starting with a certain  $V_{\text{ref}}$ , one determines the thrust coefficient, from which the axial induction  $a$  follows, and then a new reference velocity based on the local flow field is computed:  $V_{\text{ref}} = V_{\text{local}} / (1 - a)$ . This procedure is repeated until convergence is achieved. Calaf et al. [108, 28] use a similar approach by taking the local velocity and axial induction factor to determine the reference wind speed, but not in an iterative manner. The

local velocity is obtained by disk averaging and time filtering (an LES model is used).

For non-uniformly loaded disks, the force is depending on radial position but constant over an annulus, see figure 1.4. Sectional lift and drag coefficients ( $c_l$  and  $c_d$ ) are then used to find the local forces on the blades, like in BEM:

$$\rho \mathbf{f}_{2D} = \frac{1}{2} \rho V_{\text{rel}}^2 c (c_l \mathbf{e}_L + c_d \mathbf{e}_D), \quad (1.13)$$

where  $\mathbf{e}_L$  and  $\mathbf{e}_D$  are unit vectors in the direction of lift and drag;  $c_l$  and  $c_d$  are functions of the Reynolds number and the angle of attack  $\alpha$ . The force integral in equation (1.11) then becomes a line integral; the disk is recovered as a time average of line forces. The relative velocity  $V_{\text{rel}}$  at the radial positions is found by interpolating the velocity field in the surrounding computational cells. This is different from BEM, where  $V_{\text{rel}}$  is found from an iterative procedure that employs a global momentum balance. Another difference with BEM is the application of a tip-loss correction. The assumption of an infinite number of blades is corrected in BEM by locally changing the induced velocity. In Navier-Stokes computations this is not necessary, because the flow field will notice the presence of the disk so that the induction changes automatically. However, the use of 2D airfoil data still requires a correction to obtain the right flow angle and flow velocity [161]. A related problem is the determination of the local  $\alpha$  to find  $c_l$  and  $c_d$ . Shen et al. [160] developed a technique with which  $\alpha$  can be determined based on information slightly upstream of the rotor.

Rajagopalan et al. [137] were one of the first to use the actuator type approach in a CFD code, for the calculation of vertical axis turbines. Time-averaged forces are prescribed and a finite-difference laminar flow solver is used to solve the steady Navier-Stokes equations. Masson et al. [3, 104] follow the time-averaging approach of Rajagopalan in a control volume finite element (CVFEM) setting. They included a second grid around the turbine in order to evaluate the surface force integral (see equation (1.11)) accurately and linearized the force term in an iterative procedure to a steady-state solution. In [104], the lift and drag coefficients are obtained with a dynamic-stall model. In this work the tower is, like the rotor, also modeled as a porous surface. The forces on this surface are obtained from experimental data on the drag of a cylinder and are enforced by imposing a pressure discontinuity instead of adding the surface force in the equations. The presence of oscillations due to the use of collocated methods is mentioned (i.e. storing pressure and velocity variables at the same location), which is resolved by storing two different pressure values for the points located on the disk surface.

Unsteady computations with the actuator disk approach were made by J.N. Sørensen et al. by using cylindrical coordinates in a rotor-fixed reference frame [174, 175, 177]. In [175] a finite-difference method is employed to solve the unsteady Euler equations in a vorticity-streamfunction formulation (for advantages and disadvantages of such a formulation see [199]). A constant rotor loading is specified, but due to numerical difficulties at the disk edge it is replaced by an elliptic distribution with

equivalent total loading. In [174] viscous terms are included, but only to stabilize the solution. The force on the non-uniformly loaded actuator disk is obtained from tabulated airfoil data, and a non-linear filter is applied to suppress oscillations in the vicinity of the disk. In [177] the vorticity-velocity formulation is used to study different wake states, including windmill brake, turbulent wake, vortex ring and hover state. This formulation has the advantage that there is no pressure-velocity coupling, although a regularization kernel is still necessary to smoothly distribute the loading from the disk to the surrounding mesh points:

$$f_{\text{mesh}} = f_{\text{disk}} * \eta_{\epsilon}. \quad (1.14)$$

The regularization function  $\eta_{\epsilon}$  is for example a Gaussian. The choice for  $\epsilon$ , indicating the amount of smearing, is often a trade-off between stability and accuracy. With this approach the wiggles present in [174] disappeared.

Madsen [101] investigated both uniformly and non-uniformly loaded actuator disks and the effect of turbulent mixing to show the validity of the BEM theory. It was found that BEM, with the application of a tip correction, gives a good correlation with the CFD results. In later work, Madsen et al. [102] proposed a correction to BEM based on the comparison with actuator disk simulations, which showed that BEM overestimates the induction at the inboard part of the rotor and underestimates it at the outboard part.

Apart from the uniform or non-uniform axial loading described above one can also introduce tangential forces on the disk surface to account for rotational effects. Meyers and Meneveau [108] applied this in an LES context and showed that the effect of the tangential forces on the wake and extracted power appears to be negligible in case of moderate power coefficient and high tip-speed ratio. However, Porté-Agel et al. [132, 211] showed that the inclusion of rotation and non-uniform loading leads to significant improvement in the prediction of the mean velocity and turbulence intensity with respect to the uniformly loaded disk. This is especially apparent in the center of the near wake, where the uniformly loaded disk leads to an underestimation of the wake deficit and turbulence intensity. Further downstream the effect of rotation and non-uniform loading disappears.

Another approach to describe an actuator disk is the actuator shape model by Réthoré [139]. The common cells of two different grids, one for the computational domain and one for the actuator (one dimension lower), are determined and based on the intersecting polygons forcing is applied to a cell. A comparison between this actuator model and a full-rotor computation shows that modeling the wake by using forces is a good approximation for the mean flow quantities at distances larger than a rotor diameter from the wind turbine. It is observed that 10 cells per rotor diameter are sufficient; a similar number is typically found in other studies as well. However, the forces fail to represent the mechanical turbulence generated at the blade location. This turbulence can therefore be added at the disk location, independently of the actuator force, but its effect on the far wake in comparison with atmospheric and wake turbulence is small.

As was mentioned, in actuator disk simulations the boundary layers are not explicitly simulated, but their effect is taken into account via the lift and drag coefficient. Correctly simulating the blade Reynolds number is then less important, reducing the required computer resources considerably. It was shown in [177] that the velocity field does not change noticeably when the Reynolds number is larger than 1000 (see also [171, 77, 176]). This corresponds roughly to results obtained by research on the nature of the interface of turbulent wakes and the outer flow, which changes around  $Re \sim 10^4$  [40]. Whale et al. [209] also found that the behavior of the wake is possibly rather insensitive to the blade Reynolds number.

#### 1.4.1.2 Actuator line

As an extension of the non-uniformly loaded actuator disk approach, J.N. Sørensen and Shen [176] introduced the *actuator line* approach, see figure 1.4. The line forces are not averaged over the disk, but depend on time. Whereas in the actuator disk model vorticity is shed into the wake as a continuous sheet, in the actuator line model distinct tip vortices can be calculated. As for the non-uniformly loaded actuator disk, the actuator line method requires knowledge of the lift and drag on the blades. Corrections for Coriolis, centrifugal and tip effects are necessary when 2D airfoil data is used.

In order to transfer the rotating line forces to the stationary mesh a regularization kernel similar to equation (1.14) is used. Mikkelsen investigated the actuator line method in detail [111] and implemented it in *EllipSys3D*, a finite volume code for the solution of the incompressible Navier-Stokes equations in pressure-velocity formulation in general curvilinear coordinates [109, 110, 178]. Howard and Pereira [73] used point forces to represent the blades, but did not take into account the distance between the force location and cell centers, leading to high-frequency noise in the power output. They modeled the tower as a square cylinder and found that it generates a wake that partially destroys the blade tip vortices. They recommend for future work to model the tower by point forces as well.

#### 1.4.1.3 Actuator surface

Shen et al. extended the actuator line method to an *actuator surface* method [162, 163] and applied it to vertical axis wind turbines. Whereas in the actuator line model the blade is represented by a line, in the actuator surface model it is represented by a planar surface, see figure 1.4. This requires more accurate airfoil data; instead of  $c_l$  and  $c_d$ , knowledge of the pressure and skin friction distribution on the airfoil surface is needed:

$$f_{2D}^{AS}(\xi) = f_{2D} F_{\text{dist}}(\xi), \quad (1.15)$$

where  $\xi$  is directed along the chord, and  $F_{\text{dist}}(\xi)$  is determined by fitting empirical functions to chordwise pressure distributions. These are obtained in [163] with *Xfoil*, a highly accurate tool to compute pressure- and skin-friction profiles on airfoils. A comparison with 2D RANS calculations on a body-fitted mesh around an

airfoil shows that the pressure field can be accurately computed approximately one chord away from the airfoil [163]. Furthermore it is shown that the use of a dynamic-stall model in vertical-axis wind turbine calculations significantly improves the agreement with experiments. In [162] the method is applied to 3D turbine calculations and compared with the actuator line technique. Some improvements are seen in the representation of tip vortices and the flow behavior near the airfoil surface. In the actuator surface approach of Dobrev et al. [43] the pressure distribution is represented by a piecewise linear function which is directly calculated from lift and drag coefficients. Like in [162] the flow field is found to be more realistic compared to the actuator line approach, but the method is not yet able to model the wake of an airfoil since the shear forces on the blade are not accounted for.

Sibuet Watters and Masson use a completely different approach with their actuator surface concept [95, 166, 167]. Inviscid aerodynamic theory is used to relate vorticity distributions to both pressure and velocity discontinuities across a porous surface. This bears resemblance with vortex methods and lifting-line theory, which describe an inviscid flow field by concentrated vortex sheets or lines. Given the lift coefficient, the circulation is computed, and subsequently the velocity discontinuity over the surface is obtained by assuming a parabolic or cubic distribution. Viscous drag is not taken into account, but in three dimensions the actuator surface can still extract energy from the flow due to induced drag. In two dimensions, an actuator surface cannot extract energy, and a rotor disk is then modeled by prescribing the vorticity distribution of the slipstream surface, instead of the momentum loss through the disk area. The use of surface forces instead of volume forces was found to be the reason that the solution did not exhibit spurious oscillations.

#### 1.4.2 *Direct modeling*

The complete or direct modeling of the rotor by constructing a body-fitted grid is physically the most sound method to compute the flow around a turbine. Compared to the generalized actuator disk approach, the blade is represented ‘exactly’, instead of a disk/line/surface approximation. However, this approach is computationally very expensive. Firstly, the generation of a high-quality moving mesh is not trivial. Mesh generation is therefore commonly done with so-called ‘overset’ or ‘chimera’ grids: different overlapping grids, often structured, that communicate with each other. An example of such a grid is shown in figure 1.5. Secondly, simulating the boundary layer on the blades, including possible transition, separation and stall, is difficult. Additionally, compressibility effects at blade tips can require the solution of the compressible Navier-Stokes equations, whereas the wake remains essentially incompressible.

Simulations with the direct model are therefore limited to single (near-) wake computations - for examples see [154]. The main contribution to wind-turbine wake studies is to improve the actuator models described in section 1.4.1. Examples are the work of Johansen and N.N. Sørensen [83] and Bechmann and N.N. Sørensen

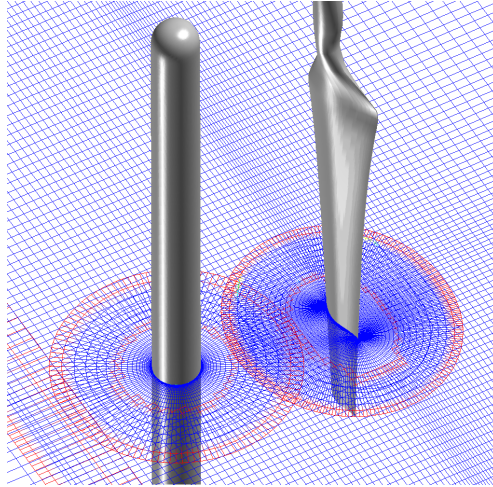


Figure 1.5: Lay-out of overset grids around tower, blade, and far-field. Reproduced from Zahle et al. [214].

[15], who extract 3D airfoil characteristics (i.e. sectional  $c_l$  and  $c_d$  when 3D flow effects are present) from their ‘full’ 3D solution. These characteristics can be used in actuator line modeling. An ongoing issue is the determination of the local angle of attack (see e.g. Hansen et al. [67] and Shen et al. [160]).

## 1.5 RESEARCH GOAL AND THESIS OUTLINE

The previous sections showed that the state-of-the-art in the numerical simulation of wind-turbine wakes is Large Eddy Simulation with the generalized actuator approach. LES methods agree better with experimental data than RANS methods. However, even with the generalized actuator approach, the computational requirements for LES are huge, and most simulations are performed on coarse grids. The main use of LES is not (yet) to design wind farms, but to gain insight in the flow physics and to improve simpler (‘engineering’) methods.

As stated in section 1.1, the goal of this thesis is to develop accurate and efficient numerical methods for the simulation of wind-turbine wakes in wind farms. Sections 1.3 and 1.4 highlighted that we need numerical methods that are

- stable and accurate on coarse meshes and for large time steps,
- stable independent of physical (laminar) or modeled (turbulent) viscosity,
- stable and accurate when body forces and discontinuities are introduced,
- introducing as little artificial diffusion as possible.

Numerical methods that possess such properties are applicable to a much broader field of research than wind-turbine wake aerodynamics alone. In a general context, our research goal can therefore be stated as

*Development of stable, low-dissipative, numerical methods for the simulation of turbulent flows.*

To achieve this goal, we will focus on so-called *energy-conserving* discretization methods. Energy-conserving discretizations have the stability properties mentioned above, and do not introduce artificial diffusion. We have developed a new Navier-Stokes solver that has this property: ECNS - Energy-Conserving Navier-Stokes solver. This solver is particularly suited for wind-turbine wake simulations, but also applicable to other flow problems. The four main ingredients for such a solver are:

1. a spatial discretization,
2. a temporal discretization,
3. an actuator model,
4. a turbulence model.

This thesis discusses the first three items. In part I the spatial discretization of the incompressible Navier-Stokes equations on staggered cartesian grids is explained. Staggering of the variables is a key component in this work, which allows us to obtain an energy-conserving spatial discretization scheme, and a strong pressure-velocity coupling necessary for correctly treating actuator forces. In part II a temporal discretization is proposed which keeps the energy conservation property when marching in time. The resulting (spatial and temporal) discretization is free of numerical viscosity and stable for any mesh and time step. In line with these properties part III proposes a new class of actuator methods that results in a sharper (less diffusive) representation of body forces than conventional actuator methods.



## Part I

### SPATIAL DISCRETIZATION

This part describes a second and fourth order energy-conserving spatial discretization on staggered cartesian grids. The boundary contributions to the discrete energy equation are analyzed, and new boundary conditions for the fourth order scheme are proposed. It is shown that energy conservation and high order of accuracy are conflicting requirements near boundaries.



## ENERGY CONSERVATION

*The key to making progress in the mathematical understanding of the Navier-Stokes equations is the energy equality. [93]*

## 2.1 INTRODUCTION

Parts I and II of this thesis address the spatial and temporal discretization of the incompressible Navier-Stokes equations. In case of inviscid flow with periodic or no-slip boundary conditions these continuous equations possess a number of properties, also called *symmetries* or *invariants*, see e.g. [52]. Such inviscid flows are of interest because many flows of practical importance, like the flow in wind-turbine wakes, are convection-dominated. We focus on one important invariant of inviscid incompressible flows, namely the kinetic energy. Upon discretizing the continuous equations in space and/or time this invariant is often not conserved.

Energy-conserving discretization methods are methods which, like the continuous incompressible Navier-Stokes equations, conserve energy in the absence of boundary conditions and forcing terms and in the limit of vanishing viscosity. The following equation:

$$\frac{dK}{dt} = -\nu \|\nabla \mathbf{u}\|^2, \quad (2.1)$$

is satisfied in a discrete sense: *the total energy of the flow can only decrease due to viscous effects*. This equation can be *derived* from the incompressible Navier-Stokes equations, equations (1.1)-(1.2); this will be done in section 2.2. The energy equation is, unlike for compressible flows, not a separate equation that can be solved, but is a consequence of conservation of mass and momentum.

There are several reasons for a discretization to mimic equation (2.1).

*Firstly*, from a physical point of view, an energy-conserving scheme is free of numerical diffusion. This is important for turbulent flow simulations with DNS or LES, because it prevents numerical diffusion from overwhelming the molecular diffusion (in case of DNS) or the effect of the sub-grid model (in case of LES), so that the energy spectrum is not affected. Energy-conserving discretizations guarantee that all diffusion is modeled (laminar and/or turbulent), and not artificial. This is why energy-conserving schemes are seen as a necessity for DNS and LES by different researchers, see e.g. [103, 126, 114, 121, 64, 203, 202]. Energy-conserving methods necessitate the use of central schemes for the convective terms. Upwind schemes, typically used in RANS simulations of turbulent flows, are robust because they introduce numerical diffusion, but should for this reason not be used in LES or DNS. Even high-order upwind methods can damp turbulence fluctuations and mask the effects of the sub-grid scale models used in LES [84, 143, 189]. Although central

schemes do not have numerical diffusion, they introduce dispersive errors; these were found to be less detrimental than diffusive errors, at least in the simulation of turbulent channel flow [50].

*Secondly*, from a more mathematical point of view, discrete energy conservation provides a non-linear stability bound to the solution (see e.g. [142]). Flow simulations are then stable for any mesh and any time step, so that these parameters can be chosen purely based on accuracy requirements. This is especially important for simulating turbulent flows that involve large time and/or length scales, like weather prediction [4].

*Thirdly*, energy-conserving methods are important when dealing with coarse grids and large time steps. Simulations of turbulence with DNS and LES are computationally very expensive and mesh sizes are kept as large as possible in practice, even under-resolved. On coarse grids it is not obvious whether the order of a method (defined for vanishing mesh sizes and time steps), is still a good measure of accuracy and whether a formally higher order method is preferred over a formally lower order method [51, 58]. Energy-conserving methods are of particular interest then, because they lead to well-posed discrete operators and as a consequence to well-behaved global errors.

## 2.2 CONTINUOUS ENERGY EQUATION

In this section we will derive equation (2.1). In doing so, we derive a number of important properties that will be mimicked in a discrete sense in section 4.1.

For convenience we repeat the incompressible Navier-Stokes equations, written as

$$\nabla \cdot \mathbf{u} = 0, \quad (2.2)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{u}) = -\nabla p + \nu \nabla^2 \mathbf{u}, \quad (2.3)$$

in a domain  $\Omega$  with boundary  $\Gamma = \partial\Omega$ , and supplemented with either no-slip boundary conditions

$$\mathbf{u} = \mathbf{u}_b \quad \text{on } \Gamma \quad (2.4)$$

or periodic boundary conditions, and a divergence-free initial condition

$$\mathbf{u} = \mathbf{u}_0. \quad (2.5)$$

The (constant) density is absorbed in the pressure. See for example Gresho and Sani [59] for well-posedness of these equations. In integral form these equations read:

$$\int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} \, d\Gamma = 0, \quad (2.6)$$

$$\int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \, d\Omega + \int_{\partial\Omega} \mathbf{u} \mathbf{u} \cdot \mathbf{n} \, d\Gamma = - \int_{\partial\Omega} p \mathbf{n} \, d\Gamma + \int_{\partial\Omega} \nu \nabla \mathbf{u} \cdot \mathbf{n} \, d\Gamma, \quad (2.7)$$

expressing conservation of mass and momentum in the domain  $\Omega$ . In case of periodic boundary conditions, all boundary integrals disappear and one obtains the global momentum balance

$$\frac{d}{dt} \int_{\Omega} \mathbf{u} \, d\Omega = 0. \quad (2.8)$$

In case the normal velocity vanishes at the boundary (no penetration,  $\mathbf{u} \cdot \mathbf{n} = 0$ ), only the contribution of the convective terms disappears.

In incompressible flows the (kinetic) energy is a secondary conserved quantity, which follows from conservation of mass and momentum. The energy equation in integral form is derived by taking the inner product of the momentum equation with a function  $v(x, t)$  and integrating it over a domain  $\Omega$ , where  $v$  satisfies the same boundary conditions as  $\mathbf{u}$ . The resulting scalar equation reads:

$$\underbrace{\int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot v \, d\Omega}_I + \underbrace{\int_{\Omega} \nabla \cdot (c \mathbf{u}) \cdot v \, d\Omega}_{II} = - \underbrace{\int_{\Omega} \nabla p \cdot v \, d\Omega}_{III} + \nu \underbrace{\int_{\Omega} \nabla^2 \mathbf{u} \cdot v \, d\Omega}_{IV}. \quad (2.9)$$

Here  $\mathbf{u}(x, t)$  and  $v(x, t)$  are elements of  $L^2(\Omega)$ , the space of square integrable functions on  $\Omega$ , with the following inner product:

$$(\mathbf{u}, v) \equiv \int_{\Omega} \mathbf{u} \cdot v \, d\Omega, \quad (2.10)$$

implying the norm  $\|\mathbf{u}\|^2 = \int_{\Omega} |\mathbf{u}|^2 \, d\Omega$ . It should be stressed here that we have introduced an explicit distinction between the *convecting* quantity  $c$  and the *convected* quantity  $\mathbf{u}$ , see equation (2.9).

The separate terms in equation (2.9) will be rewritten with the divergence theorem to reveal a number of important properties. The second, convective, term is first written by employing the product rule as:

$$\nabla \cdot (c \mathbf{u}) \cdot v = (\mathbf{u} \cdot v) \nabla \cdot c + [(c \cdot \nabla) \mathbf{u}] \cdot v. \quad (2.11)$$

Consequently we have

$$[(c \cdot \nabla) \mathbf{u}] \cdot v = \nabla \cdot ((\mathbf{u} \cdot v) c) - (\mathbf{u} \cdot v) \nabla \cdot c - [(c \cdot \nabla) v] \cdot \mathbf{u}. \quad (2.12)$$

When using  $\nabla \cdot c = 0$ , integrating over the domain, and applying integration by parts (divergence theorem) we obtain

$$II : \quad \int_{\Omega} [(c \cdot \nabla) \mathbf{u}] \cdot v \, d\Omega = \int_{\partial\Omega} [(\mathbf{u} \cdot v) c] \cdot \mathbf{n} \, d\Gamma - \int_{\Omega} [(c \cdot \nabla) v] \cdot \mathbf{u} \, d\Omega. \quad (2.13)$$

In case of periodic or no-penetration boundary conditions the boundary integral on the right-hand side vanishes, and the resulting equation can be written in terms of the inner product (2.10) as

$$((\mathbf{c} \cdot \nabla)\mathbf{u}, \mathbf{v}) = -(\mathbf{u}, (\mathbf{c} \cdot \nabla)\mathbf{v}). \quad (2.14)$$

This relation expresses the skew-symmetry of the convective operator; it holds if  $\nabla \cdot \mathbf{c} = 0$  and boundary conditions are periodic or of no-penetration type. Denoting the convective operator by  $(\mathbf{c} \cdot \nabla)\mathbf{u} = \mathbf{C}(\mathbf{c})\mathbf{u}$  it can alternatively be written as

$$\mathbf{C}(\mathbf{c}) = -\mathbf{C}(\mathbf{c})^*, \quad (2.15)$$

where  $*$  denotes the Hermitian conjugate. Taking  $\mathbf{v} = \mathbf{u}$ , we see that the convective term does not change the total energy of the flow:

$$(\mathbf{C}(\mathbf{c})\mathbf{u}, \mathbf{u}) = 0. \quad (2.16)$$

The third term, which expresses the work done by pressure forces, is written as

$$III : \quad \int_{\Omega} \nabla p \cdot \mathbf{v} \, d\Omega = \int_{\partial\Omega} p \mathbf{v} \cdot \mathbf{n} \, d\Gamma - \int_{\Omega} p \nabla \cdot \mathbf{v} \, d\Omega. \quad (2.17)$$

Again, in case of no penetration, periodic, or zero pressure boundary conditions this relation can be written as

$$(\nabla p, \mathbf{v}) = -(p, \nabla \cdot \mathbf{v}) = 0. \quad (2.18)$$

This equation expresses the compatibility relation between the divergence and gradient operator.

The last term, the viscous dissipation, is rewritten with  $\nabla^2 \mathbf{u} \cdot \mathbf{v} = \nabla \cdot (\mathbf{v} \cdot (\nabla \mathbf{u})^T) - \nabla \mathbf{v} : \nabla \mathbf{u}$ <sup>1</sup>, leading to:

$$IV : \nu \int_{\Omega} \nabla^2 \mathbf{u} \cdot \mathbf{v} \, d\Omega = \nu \int_{\partial\Omega} \mathbf{v} \cdot \frac{\partial \mathbf{u}}{\partial n} \, d\Gamma - \nu \int_{\Omega} \nabla \mathbf{v} : \nabla \mathbf{u} \, d\Omega. \quad (2.19)$$

Ignoring boundary terms and performing integration by parts once again one obtains the symmetry of the diffusive operator  $\mathbf{D}(\mathbf{u}) = \nabla^2 \mathbf{u}$ :

$$\mathbf{D} = \mathbf{D}^*. \quad (2.20)$$

To arrive at the equation for the evolution of the total kinetic energy we assume no-penetration or periodic boundary conditions and take  $\mathbf{c} = \mathbf{u}$ ,  $\mathbf{v} = \mathbf{u}$ . The con-

<sup>1</sup> the Frobenius product : is defined as  $A : B \equiv \sum_i \sum_j A_{ij} B_{ij} = \text{tr}(A^T B)$

vective and pressure terms vanish, so the evolution equation for the kinetic energy  $k = \frac{1}{2}\|\mathbf{u}\|^2$  becomes:

$$\frac{d}{dt} \int_{\Omega} k \, d\Omega = \nu \int_{\partial\Omega} \mathbf{u} \cdot \frac{\partial \mathbf{u}}{\partial n} \, d\Gamma - \nu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{u} \, d\Omega. \quad (2.21)$$

In case of periodic boundary conditions the first term vanishes, so

$$\frac{dK}{dt} = -\nu(\nabla \mathbf{u}, \nabla \mathbf{u}), \quad (2.22)$$

where  $K := \int k \, d\Omega = \frac{1}{2}(\mathbf{u}, \mathbf{u})$  and the term on the right-hand side should be seen as a Frobenius product. Since  $\nabla \mathbf{u} : \nabla \mathbf{u} \geq 0$ , this equation shows that the total energy of the flow decreases:

$$K(t) \leq K(t_0), \quad t \geq t_0. \quad (2.23)$$

This is due to viscous dissipation only; the convective and pressure terms do not change the total energy. If equation (2.22) is satisfied in a discrete sense, then the solution is bounded, independent of grid layout or time step.

A sharper estimate than (2.23) can be obtained by bounding the dissipative terms using the Poincaré equality [168]

$$\|\mathbf{u}\| \leq (\nabla \mathbf{u}, \nabla \mathbf{u}) / \sqrt{C}, \quad (2.24)$$

where  $C > 0$  is the lowest eigenvalue of the diffusive operator. This yields the estimate

$$K(t) \leq K(t_0)e^{-2\nu Ct}. \quad (2.25)$$

When taking body forces,  $\mathbf{f}$ , into account in the momentum equations, an additional term in the kinetic energy equation appears,

$$\frac{dK}{dt} = -\nu(\nabla \mathbf{u}, \nabla \mathbf{u}) + (\mathbf{f}, \mathbf{u}), \quad (2.26)$$

and the estimate (2.25) becomes [168]:

$$K(t) \leq K(t_0)e^{-\nu Ct} + (1 - e^{-\nu Ct}) \frac{\|\mathbf{f}\|^2}{2\nu^2 C^2}. \quad (2.27)$$





DISCRETIZATION ON A STAGGERED GRID

---

*... this is the oldest and most straightforward approach to discretizing the Navier-Stokes equations... on orthogonal grids it remains the method of choice [207]*

## 3.1 INTRODUCTION

The first energy-conserving scheme for the incompressible Navier-Stokes equations is probably Harlow and Welch's staggered grid method [68]. The staggering of the variables leads to a method that conserves mass, momentum, energy and vorticity, and strongly couples pressure and velocity, making it the method of choice for simulating incompressible flows on cartesian grids [207]. Ham et al. [65] extended the method to retain these properties on non-uniform grids. In order to simulate flows around complex geometries, Wesseling et al. [208, 207] extended the method to general structured grids that can be described by a cartesian grid in computational space. On unstructured meshes staggered methods have been investigated by Perot and co-workers [126, 215]. Mahesh et al. [103] also consider energy-conserving methods for unstructured meshes and formulate a second order staggered method for tetrahedral elements, but propose a non-staggered formulation for elements of more general shape. This results in a formulation that is not fully energy-conserving, because the pressure gradient contributes to the kinetic energy.

In fact, the contribution of the pressure gradient to the kinetic energy is intrinsic to non-staggered ('collocated') layouts [116, 50]. Felten and Lund [50] show that this energy error makes staggered schemes superior to collocated schemes in case of inviscid simulations and in case of viscous simulations with relatively coarse meshes. However, operators in a collocated arrangement can be 'shifted' to obtain operators for staggered meshes, as shown by Hicken et al. [69]. This elegant approach can be seen as a generalization of the work of Perot to general meshes [126], including locally refined ones. Numerical experiments show that on anisotropic cartesian grids, where the local truncation error of the gradient is inconsistent (zeroth order), it is still possible to have a first order accurate solution, emphasizing the positive influence of energy-conserving methods on global discretization errors.

High-order (higher than two) energy-conserving methods hold the promise to be more efficient than low-order methods. High-order energy-conserving finite difference methods were addressed by Morinishi et al. [116] and Vasilyev [196], who indicate how to obtain any (even) order of accuracy on uniform grids based on Richardson extrapolation. On non-uniform grids strict conservation and (local) order of accuracy cannot be obtained simultaneously. Verstappen and Veldman [202, 203] employ a finite volume method and construct a fourth order accurate method

on non-uniform grids that retains all properties of the Harlow and Welch scheme. They call their method ‘symmetry-preserving’, because it is based on mimicking symmetry properties of continuous operators (see equations (2.14) and (2.20)) in a discrete sense, instead of on minimizing the local truncation error. Preserving these symmetries leads to energy conservation. Again, it is found that the global order of accuracy cannot be derived from the local truncation error alone; the discretized operator is at least of equal importance, and it is indeed the energy-conserving schemes that lead to a discretized operator that is well-posed, even on coarse meshes. The fourth order scheme appears to be highly efficient: accurate results have been obtained on coarse meshes for turbulent channel flows [203].

Another way to obtain higher order methods is to use compact (implicit) schemes, either in finite difference or finite volume context (see e.g. [87] and [70] for references). Implementation of boundary conditions is easier due to their smaller computational stencil, and furthermore they have better resolution of high wave numbers than explicit schemes. Knikker [87] obtains energy conservation with a fourth order compact finite difference scheme with the nonlinear terms in skew-symmetric form, Hokpunna and Manhart [70] mention that energy conservation for compact finite volume methods is still an open issue.

Lastly, this section would be incomplete without mentioning the recent review article of Perot [128] and the work on ‘mimetic’ methods, like the support-operator method by Shashkov [159], Steinberg, Bochev, Hyman [75], the work on summation-by-parts operators initiated by Kreiss and Scherer [88] and the dual-mesh methods of Nicolaides [122], Subramanian and Perot [186] and Mullen et al. [120].

In this chapter we describe the second order method of Harlow and Welch (HW) [68] and the fourth order method of Verstappen and Veldman (VV) [203]; these methods form the backbone of the ECNS code. We have chosen finite volume methods because of their conservation properties and their ability to handle discontinuities (to be discussed in chapter 8).

## 3.2 INTERIOR DISCRETIZATION

### 3.2.1 General formulation

We spatially discretize equations (2.6)-(2.7) on a staggered cartesian grid. On a staggered grid the pressure and the velocity components are defined at different locations; see figure 3.1a. A staggered grid yields a natural coupling between the divergence and gradient operator, which ensures that the pressure gradient does not contribute to the energy equation, as we will show later. The resulting semi-discrete system of equations reads:

$$Mu(t) = r_1(t), \quad (3.1)$$

$$\Omega \dot{u}(t) = -C(c(t), u(t)) + vDu(t) - Gp(t) + r_2(u(t), t), \quad (3.2)$$

supplemented with suitable initial conditions. We take normal instead of bold font to distinguish between the continuous and discrete quantities.  $c(t)$ ,  $u(t) \in \mathbb{R}^{N_u}$  and  $p(t) \in \mathbb{R}^{N_p}$  are vectors with the unknown *point values* of velocity and pressure in the center of the finite volumes (not volume averages), which depend on time only. In the remainder, their explicit  $t$ -dependence will mostly be left out of the notation.  $M$ ,  $C$ ,  $D$  and  $G$  represent the discrete divergence, convection, diffusion and gradient operators, respectively. Note that  $r_2 \in \mathbb{R}^{N_u}$ ;  $r_1 \in \mathbb{R}^{N_p}$ ;  $M \in \mathbb{R}^{N_u \times N_p}$ ;  $\Omega, C, D \in \mathbb{R}^{N_u \times N_u}$ ,  $G \in \mathbb{R}^{N_p \times N_u}$ .  $r_1(t)$  is a vector with boundary conditions for the continuity equation and  $r_2(u, t)$  is a vector with boundary conditions and forcing terms for the momentum equation. The diagonal matrix  $\Omega = \text{diag}(\Omega^u, \Omega^v)$  contains the finite volume sizes of the  $u$ - and  $v$ -volumes (which are assumed to be independent of  $t$ ). In finite element methods this is the mass matrix.

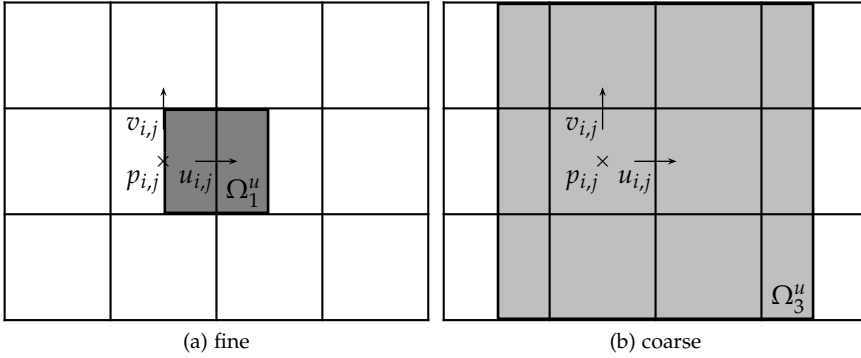


Figure 3.1: Fine and coarse finite volumes centered around  $u_{i,j}$ .

### 3.2.2 Second order

The discretization of HW is constructed on a finite volume  $\Omega_1 = \Delta x \Delta y$  (see figure 3.1a for the  $u$ -component). The discrete operators on this grid are given by  $M = M_1$ ,  $\Omega = \Omega_1$ , etc. The divergence operator is given by

$$(Mu)_{i,j} = (M_1 u)_{i,j} = \bar{u}_{i,j} - \bar{u}_{i-1,j} + \bar{v}_{i,j} - \bar{v}_{i,j-1}, \quad (3.3)$$

where the face-integrated quantities such as  $\bar{u}_{i,j}$  are related to the point values  $u_{i,j}$  via the midpoint rule:

$$\bar{u}_{i,j} = \Delta y u_{i,j}. \quad (3.4)$$

In three dimensions an additional  $\Delta z$  appears in these expressions. The bar  $\bar{(\cdot)}$  indicates integration in  $x$ - or  $y$ -direction; this should be clear from the context. The convective discretization for the  $u$ -component is given by

$$(C^u(\bar{u}, \bar{v})u)_{i,j} = (C_1^u(\bar{u}, \bar{v})u)_{i,j}$$

$$\begin{aligned}
&= \bar{u}_{i+1/2,j} u_{i+1/2,j} - \bar{u}_{i-1/2,j} u_{i-1/2,j} + \\
&\quad \bar{v}_{i+1/2,j} u_{i,j+1/2} - \bar{v}_{i+1/2,j-1} u_{i,j-1/2}.
\end{aligned} \tag{3.5}$$

The convected  $(u, v)$  and convecting  $(\bar{u}, \bar{v})$  velocities are obtained by second order, grid-independent, interpolation of the form

$$\bar{u}_{i+1/2,j} = \frac{1}{2}(\bar{u}_{i,j} + \bar{u}_{i+1,j}). \tag{3.6}$$

The pressure gradient discretization is given by

$$(G^u p)_{i,j} = (G_1^u p)_{i,j} = \bar{p}_{i+1,j} - \bar{p}_{i,j}. \tag{3.7}$$

The diffusive terms are discretized as

$$(D^u u)_{i,j} = (D_1^u u)_{i,j} = \left( \frac{\partial \bar{u}}{\partial x} \right)_{i+1/2,j} - \left( \frac{\partial \bar{u}}{\partial x} \right)_{i-1/2,j} + \left( \frac{\partial \bar{u}}{\partial y} \right)_{i,j+1/2} - \left( \frac{\partial \bar{u}}{\partial y} \right)_{i,j-1/2}, \tag{3.8}$$

where

$$\left( \frac{\partial \bar{u}}{\partial x} \right)_{i+1/2,j} = \frac{\bar{u}_{i+1,j} - \bar{u}_{i,j}}{\Delta x}. \tag{3.9}$$

The diffusive discretization is the product of two first order operators and can be written in short as

$$D_1^u = \Delta_1^u \Lambda_1^u (\Delta_1^u)^T, \tag{3.10}$$

where  $\Delta_1^u$  is a differencing matrix consisting of  $-1, 0$  and  $1$ , and  $\Lambda_1^u$  a diagonal matrix containing metric information  $(\Delta y / \Delta x)$ .

### 3.2.3 Fourth order

The discretization of VV is constructed by combining the discretization on  $\Omega_1$  with the discretization on a coarser finite volume  $\Omega_3$ . The discrete operators on  $\Omega_3$  are indicated by  $M_3, C_3$ , etc. The coarse volumes are three times as large as the fine volumes (in each direction), see figure 3.1b.  $\Omega_1$  and  $\Omega_3$  are combined such that the leading order error term of the local truncation error cancels (Richardson extrapolation), resulting in a fourth order approximation. The fourth order approximation to the  $u$ -component of the divergence equation reads

$$(Mu)_{i,j} = \alpha_1 (M_1 u)_{i,j} - (M_3 u)_{i,j} = \alpha_1 (\bar{u}_{i,j} - \bar{u}_{i-1,j}) - (\bar{\bar{u}}_{i+1,j} - \bar{\bar{u}}_{i-2,j}), \tag{3.11}$$

where the face-integrated quantity  $\bar{\bar{u}}_{i,j}$  is given by

$$\bar{\bar{u}}_{i,j} = \alpha_2 \Delta y u_{i,j}. \tag{3.12}$$

The coefficients  $\alpha_1 = 3^{2+d}$  and  $\alpha_2 = 3^{d-1}$ , with  $d$  the number of dimensions, are such that the leading order error term cancels. The *effective* finite volume size is given by  $\Omega^u = \alpha_1 \Omega_1^u - \Omega_3^u$ . In case of a uniform grid,  $\Omega_3^u = 3^d \Omega_1^u$ , so

$$\Omega^u = 3^d(3^2 - 1)\Omega_1^u. \quad (3.13)$$

Upon division of the semi-discrete equations by this effective finite volume size one obtains the equations in ‘finite-difference form’. For example, the divergence operator becomes

$$\frac{1}{3^d(3^2 - 1)\Delta x \Delta y} (\alpha_1 \Delta y (u_{i,j} - u_{i-1,j}) - \alpha_2 \Delta y (u_{i+1,j} - u_{i-2,j})) = \frac{9}{8} \frac{u_{i,j} - u_{i-1,j}}{\Delta x} - \frac{1}{8} \frac{u_{i+1,j} - u_{i-2,j}}{3\Delta x}. \quad (3.14)$$

The coefficients  $\frac{9}{8}$  and  $-\frac{1}{8}$  are the coefficients that appear in the finite difference method of Morinishi et al. [116] - the methods of Morinishi et al. [116] and VV [203] are equivalent on uniform grids, and the analysis presented in the next sections is therefore equally valid for the finite difference method.

The convective discretization for the  $u$ -component reads

$$C^u(\bar{u}, \bar{v})u = \alpha_1 C_1^u(\bar{u}, \bar{v})u - C_3^u(\bar{u}, \bar{v})u, \quad (3.15)$$

where

$$(C_3^u(\bar{u}, \bar{v})u)_{i,j} = \bar{u}_{i+3/2,j} u_{i+3/2,j} - \bar{u}_{i-3/2,j} u_{i-3/2,j} + \bar{v}_{i+1/2,j+1} u_{i,j+3/2} - \bar{v}_{i+1/2,j-2} u_{i,j-3/2}. \quad (3.16)$$

Like the divergence operator, the convective operator is a combination of two second order operators, with the exception that *all* convecting velocities ( $\bar{u}$ ,  $\bar{v}$ ,  $\bar{\bar{u}}$  and  $\bar{\bar{v}}$ ) are obtained by fourth order, grid-independent interpolation, e.g.:

$$\bar{\bar{u}}_{i+3/2,j} = \frac{1}{2}(\beta_1 \bar{u}_{i,j} + \beta_2 \bar{u}_{i+1,j} + \beta_2 \bar{u}_{i+2,j} + \beta_1 \bar{u}_{i+3,j}), \quad (3.17)$$

with  $\beta_1 = -\frac{1}{8}$  and  $\beta_2 = \frac{9}{8}$ . The convected velocities are obtained as

$$u_{i+3/2,j} = \frac{1}{2}(u_{i,j} + u_{i+3,j}). \quad (3.18)$$

The pressure gradient terms are, like the divergence operator, a linear combination of two second order operators:

$$G^u p = \alpha_1 G_1^u p - G_3^u p, \quad (3.19)$$

with

$$(G_3^u p)_{i,j} = \bar{\bar{p}}_{i+3,j} - \bar{\bar{p}}_{i,j}. \quad (3.20)$$

The diffusive operator is obtained as the product of two fourth order operators, which ensures that the resulting operator is symmetric and positive definite. We focus on the fluxes in  $x$ -direction:

$$(D^u u)_{i,j} = \alpha_1 \left( \left( \frac{\partial \bar{u}}{\partial x} \right)_{i+1/2,j} - \left( \frac{\partial \bar{u}}{\partial x} \right)_{i-1/2,j} \right) - \left( \left( \frac{\partial \bar{u}}{\partial x} \right)_{i+3/2,j} - \left( \frac{\partial \bar{u}}{\partial x} \right)_{i-3/2,j} \right). \quad (3.21)$$

As was the case for the convective operator, the fluxes on both the small and large volume are approximated to fourth order:

$$\left( \frac{\partial \bar{u}}{\partial x} \right)_{i+1/2,j} = \frac{1}{72\Delta x} (\alpha_1 (\bar{u}_{i+1,j} - \bar{u}_{i,j}) - (\bar{\bar{u}}_{i+2,j} - \bar{\bar{u}}_{i-1,j})), \quad (3.22)$$

and similarly for  $\left( \frac{\partial \bar{u}}{\partial x} \right)_{i+3/2,j}$ . In short, the diffusive discretization reads

$$D^u = (\alpha_1 \Delta_1^u - \Delta_3^u)(\alpha_1 \Lambda_1^u - \Lambda_3^u)(\alpha_1 \Delta_1^u - \Delta_3^u)^T. \quad (3.23)$$

### 3.2.4 Extension to higher order

For higher order (e.g. sixth order) the same approach should be followed [116]: an  $n$ th order combination of second order convective discretizations (like (3.15)), where each convective discretization contains an  $n$ th order convecting velocity (like (3.17)).

## 3.3 NUMERICAL EXPERIMENTS

In this section we verify the convergence properties of the second- and fourth-order method by studying the classical Taylor-Green vortex problem and the roll-up of a shear-layer. For the performance on other test cases, such as the flow over a backward-facing step, we refer to [145].

### 3.3.1 Taylor-Green vortex

The Taylor-Green vortex in two dimensions is an exact solution to the Navier-Stokes equations:

$$u(x, y, t) = -\sin(\pi x) \cos(\pi y) e^{-2\pi^2 t / \text{Re}}, \quad (3.24)$$

$$v(x, y, t) = \cos(\pi x) \sin(\pi y) e^{-2\pi^2 t / \text{Re}}, \quad (3.25)$$

$$p(x, y, t) = \frac{1}{4} (\cos(2\pi x) + \cos(2\pi y)) e^{-4\pi^2 t / \text{Re}}. \quad (3.26)$$

The domain on which we define the solution is the square  $[\frac{1}{4}, 2\frac{1}{4}] \times [\frac{1}{4}, 2\frac{1}{4}]$  with periodic boundary conditions. We set  $\text{Re} = 100$  and we integrate from  $t = 0$  to  $t = 1$ . We use an explicit fourth-order, four-stage Runge-Kutta method for this (see chapter 6), which ensures that the temporal errors are negligible compared to the spatial errors. We use different uniform grids ranging from  $5 \times 5$  to  $320 \times 320$  volumes using a fixed time step of  $\Delta t = 10^{-3}$ , and calculate the error in the resulting velocity and pressure fields by comparing with the exact solution at  $t = 1$ .

Figure 3.2 shows that the error in both the velocity and the pressure is second order for the discretization of HW and fourth order for the discretization of VV, in the  $L_\infty$ -norm.

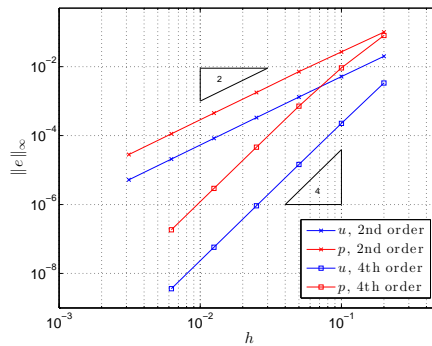


Figure 3.2: Convergence of spatial error for Taylor-Green problem.

### 3.3.2 Shear layer roll-up

As a next test case we consider the roll-up of a shear layer. At the ‘edge’ of a wind-turbine wake such a shear layer is present due to the velocity difference between the flow inside and the flow outside the wake, see figure 3.3. In this section we simulate such a problem in an idealized setting.

Inspired by [21, 87, 70] we take a domain of  $[0, 2\pi] \times [0, 2\pi]$  with periodic boundary conditions. The initial condition is

$$u = \begin{cases} \tanh\left(\frac{y-\pi/2}{\delta}\right), & y \leq \pi, \\ \tanh\left(\frac{3\pi/2-y}{\delta}\right), & y > \pi, \end{cases} \quad v = \varepsilon \sin(x), \quad (3.27)$$

with  $\delta = \pi/15$ ,  $\varepsilon = 0.05$  taken from [87]. The perturbation of the  $v$ -velocity makes sure that the shear layer starts to roll-up, independent of viscosity. An impression of the flow field at  $t = 8$ , for both the second and the fourth order method, is given in figure 3.4. The flow field is slightly under-resolved (the mesh Péclet condition is not satisfied), which manifests itself by noise (‘wiggles’) in the solution. Remarkable



Figure 3.3: Tip vortices originating at blade tips. Reproduced from [2].

is that the fourth order method is *less* susceptible to wiggles than the second order method; this was also noted by Knikker [87].

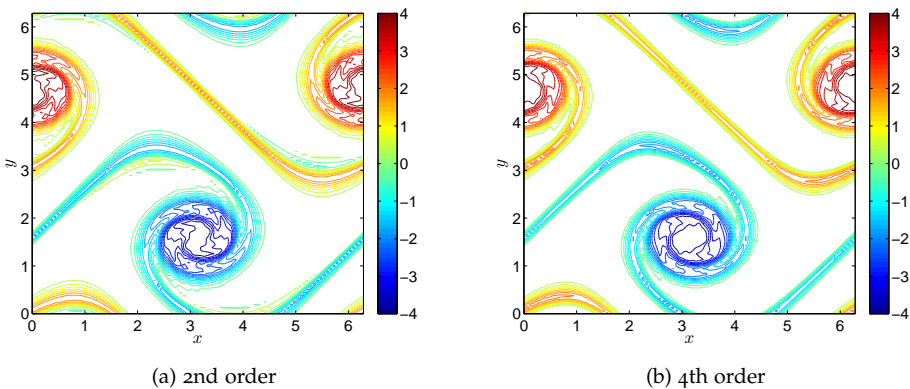
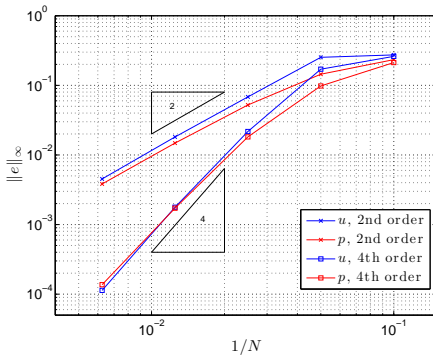


Figure 3.4: Vorticity field at  $t = 8, 100^2$  volumes,  $Re = 1000$ .

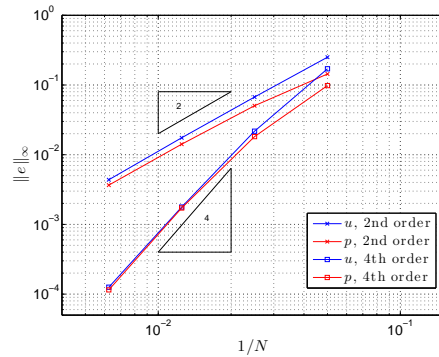
An accuracy study has been performed by computing a reference solution on a  $320 \times 320$  grid (at  $t = 4$  and  $Re = 100$ , to have a sufficiently ‘smooth’ solution) with an explicit Runge-Kutta method and sufficiently small time step ( $\Delta t = 10^{-2}$ ) so that temporal errors are negligible. On a sequence of coarser grids we compute an error by interpolating the reference solution to the coarser grid and subtracting the solution on that coarser grid. The resulting errors in velocity and pressure (in the maximum norm) are shown in figure 3.5a. It is clear that on sufficiently fine meshes both the second and fourth order scheme attain their theoretical convergence rates, for both velocity and pressure. In figure 3.5b we have also shown the results when using implicit Runge-Kutta methods for the temporal integration (to be discussed in chapter 7) - the second order Gauss method (implicit midpoint) with the second order scheme, and the fourth order Gauss method with the fourth order scheme.



The time step is now *much larger*: we use a fixed CFL number of approximately 3. On the coarsest mesh we have  $\Delta t = 1$ , meaning that we only perform 4 time steps. Since the results are indistinguishable compared to figure 3.5a the error is completely dominated by the spatial error, even at this large time step.



(a) Explicit Runge-Kutta 4



(b) Implicit Runge-Kutta (Gauss 2 and Gauss 4)

Figure 3.5: Convergence of spatial error for shear layer problem.



*The fact that momentum and energy conservation are both governed by the same equation makes construction of numerical approximations that conserve both properties difficult. [51]*

*adapted  
from [155]*

In chapter 3 we outlined a second and fourth order (interior) discretization on a staggered grid. In this chapter we will discuss the energy-conserving properties of these methods, and focus on the role of boundary conditions. To obtain a fully energy-conserving method, the discrete boundary contributions to the energy equation should properly mimic the continuous contributions. In literature these boundary contributions are often not thoroughly treated (like in [203, 116, 42]). We propose boundary conditions for the second and fourth order discretizations such that discrete equivalents of the (continuous) equations of section 2.2 are satisfied: conservation of mass (section 4.1.1), conservation of momentum (section 4.1.2), conservation of kinetic energy (section 4.1.3) and the equation for the pressure (section 4.1.6). In section 4.2 we analyse the consequences of the boundary conditions for the local and global error, which we support with numerical experiments in section 4.3.

#### 4.1 CONSERVATION PROPERTIES OF THE DISCRETE EQUATIONS

To mimic the symmetry properties (2.15), (2.18), (2.20) in a discrete setting, and in order to define a discrete kinetic energy, we need a discrete inner product:

$$(u, v) := u^T \Omega v. \quad (4.1)$$

This is the discrete analogue of equation (2.10).

##### 4.1.1 Mass

We will investigate if the second and fourth order discretizations satisfy a discrete equivalent of

$$\int_{\Omega} \nabla \cdot \mathbf{u} \, d\Omega = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} \, d\Gamma = 0. \quad (4.2)$$

4.1.1.1 *Second order*

It suffices to focus on the fluxes of the  $u$ -component (in  $x$ -direction). In matrix notation the discrete divergence operator, equation (3.3), reads

$$M_1^u \bar{u} = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \bar{u}_{b,j} \\ \bar{u}_{1,j} \\ \bar{u}_{2,j} \\ \vdots \end{bmatrix}, \quad (4.3)$$

see figure 4.1b. Discrete global mass conservation is found by summing the contribution of all finite volumes in  $x$ -direction (taking the column sum of (4.3)):

$$1^T M_1^u \bar{u} = -\bar{u}_{b,j} + (\text{right boundary contribution}). \quad (4.4)$$

Here  $1^T$  denotes a row vector with entries one. Equation (4.4) is the discrete equivalent of (4.2). When the midpoint rule (3.4) is employed for  $\bar{u}_{b,j}$ , and  $u_{b,j}$  is not constant along a boundary, an  $\mathcal{O}(h^2)$  error is introduced in this equation, and it does not evaluate to zero (like (4.2)). Instead, one could use the exact integral  $\bar{u}_{b,j}$  in the discretization of the continuity equation, but that is often not available (e.g. a turbulent inflow field). Fortunately, flow problems with an inflow boundary (nonzero  $u_{b,j}$ ) necessarily include an outflow boundary, where the velocity adapts itself such that (4.4) evaluates exactly to 0.

4.1.1.2 *Fourth order*

The divergence operator for the fourth order method is given by equation (3.11). To investigate the boundary contribution we evaluate (3.11) for  $i = 1$ , i.e. the finite volume around  $p_{1,j}$ . Figure 4.1b shows that the coarse volume extends over the physical boundary. We can focus on  $M_3$ . The choice for  $\bar{\bar{u}}_{-1,j}$  should be such that the resulting divergence operator satisfies a summation-by-parts property. By taking the second order extrapolation

$$\bar{\bar{u}}_{-1,j} = 2\bar{\bar{u}}_{b,j} - \bar{\bar{u}}_{1,j}, \quad (4.5)$$

the resulting divergence operator reads

$$M_3^u \bar{\bar{u}} = \begin{bmatrix} -2 & 1 & 1 & & & \\ -1 & & & 1 & & \\ & -1 & & & 1 & \\ & & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \bar{\bar{u}}_{b,j} \\ \bar{\bar{u}}_{1,j} \\ \bar{\bar{u}}_{2,j} \\ \bar{\bar{u}}_{3,j} \\ \vdots \end{bmatrix}, \quad (4.6)$$

and the discrete equivalent of (4.2) is

$$1^T M_3^u \bar{u} = -3\bar{u}_{b,j} + (\text{right boundary contribution}). \quad (4.7)$$

After employing (3.12), a factor 9 results, which is due to coarse grid volumes being  $9\times$  larger than the fine grid volumes (in two dimensions). In other words: each fine grid volume is covered by 9 different coarse volumes. We note that with the boundary condition for the divergence operator suggested in VV [203],

$$\bar{u}_{-1,j} = \bar{u}_{1,j}, \quad (4.8)$$

the summation-by-parts property (4.7) is not satisfied.

In section 4.1.2 the divergence of a coarse volume centered around the ghost point  $p_{0,j}$  will be needed in the discretization of the convective terms (see figure 4.1a):

$$(Mu)_{0,j} = \alpha_1(\bar{u}_{b,j} + \bar{v}_{0,j} - \bar{u}_{-1,j} - \bar{v}_{0,j-1}) - (\bar{u}_{1,j} + \bar{v}_{0,j+1} - \bar{u}_{-2,j} - \bar{v}_{0,j-2}). \quad (4.9)$$

This divergence is not enforced to be zero by the divergence operator. However, by properly choosing the ghost values it can be expressed in terms of  $(Mu)_{1,j}$ , the divergence of the volume centered around  $p_{1,j}$ . Taking, like (4.5),

$$\bar{u}_{-2,j} = 2\bar{u}_{b,j} - \bar{u}_{2,j}, \quad (4.10)$$

$$\bar{u}_{-1,j} = 2\bar{u}_{b,j} - \bar{u}_{1,j}, \quad (4.11)$$

and choosing Neumann-type conditions for  $v$ :

$$\bar{v}_{0,j} = \bar{v}_{1,j}, \quad (4.12)$$

$$\bar{v}_{0,j} = \bar{v}_{1,j}, \quad (4.13)$$

equation (4.9) can be written as

$$\begin{aligned} (Mu)_{0,j} &= \alpha_1(\bar{u}_{b,j} + \bar{v}_{0,j} - \bar{u}_{-1,j} - \bar{v}_{0,j-1}) - (\bar{u}_{1,j} + \bar{v}_{0,j+1} - \bar{u}_{-2,j} - \bar{v}_{0,j-2}) \\ &= \alpha_1(\bar{u}_{1,j} + \bar{v}_{1,j} - \bar{u}_{b,j} - \bar{v}_{1,j-1}) - (\bar{u}_{2,j} + \bar{v}_{1,j+1} + \bar{u}_{1,j} - \bar{v}_{1,j-2} - 2\bar{u}_{b,j}) \\ &= (Mu)_{1,j} = 0. \end{aligned} \quad (4.14)$$

Once the boundary conditions for the divergence operator are chosen, the gradient operator for the pressure is fixed by requiring a discrete equivalent of equation (2.17), as will be discussed in section 4.1.3. The boundary conditions for the pressure should therefore not be set separately from those for the velocity (in contrast to what is done in [116]). The boundary conditions for the divergence operator also influence the boundary treatment of the convective operator. This will also be further detailed in section 4.1.3.

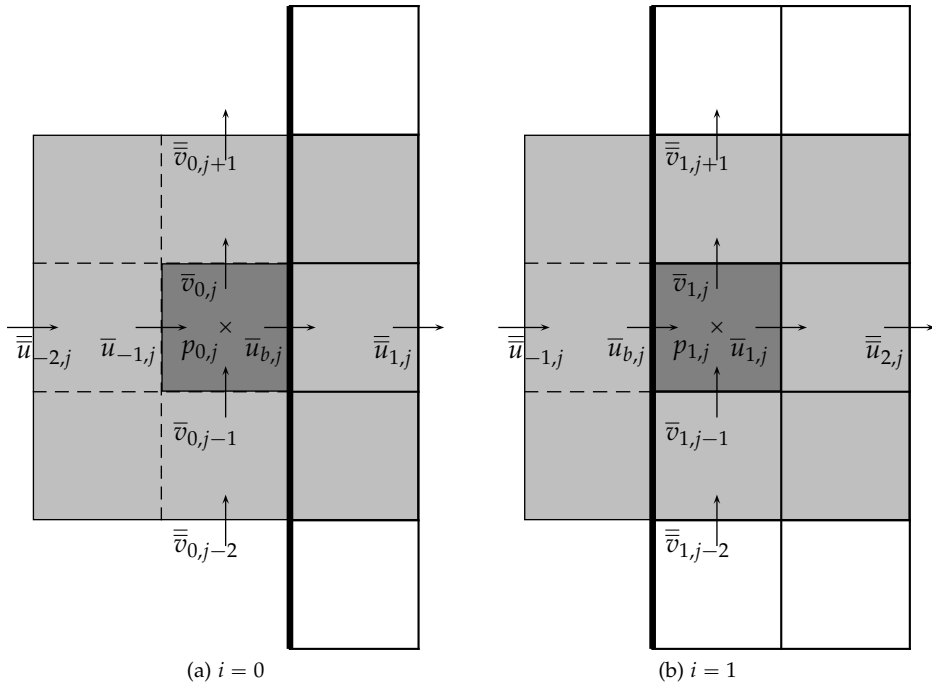


Figure 4.1: Boundary conditions divergence operator.

### 4.1.2 Momentum

In this section we investigate the discrete equivalents of the following identities:

$$\int_{\Omega} \nabla \cdot (\mathbf{u} \mathbf{u}) \, d\Omega = \int_{\partial\Omega} \mathbf{u} \mathbf{u} \cdot \mathbf{n} \, d\Gamma, \tag{4.15}$$

$$\int_{\Omega} \nabla p \, d\Omega = \int_{\partial\Omega} p \mathbf{n} \, d\Gamma, \tag{4.16}$$

and

$$\int_{\Omega} \nabla^2 \mathbf{u} \, d\Omega = \int_{\partial\Omega} \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \, d\Gamma. \tag{4.17}$$

In contrast to the pressure finite volumes, the finite volumes associated with the  $u$ - and  $v$ -velocity components do not cover the entire computational domain (see for example figure 4.2a). In order to mimic (4.15) and (4.16) discretely, we propose a new approach that includes a discretization of the boundary volumes. This allows us to perform a discrete summation over the entire domain. Even though this discretization is not needed in practical computations, it is necessary to study the boundary contributions to the energy equation (section 4.1.3). An alternative approach, which does not require the discretization of the boundary volumes, follows

by writing (4.15) and (4.16) for the domain covered by the interior finite volumes (instead of the complete computational domain), but this approach is less suitable for the fourth order discretization.

#### 4.1.2.1 Second order

We start again with the second order discretization to provide insight into the derivation of the fourth order method. In the interior the discretization conserves momentum since the fluxes of neighboring finite volumes cancel (the telescoping property of the finite volume method); see section 3.2.2. The conserved quantities are

$$1^T \Omega^u u, 1^T \Omega^v v. \quad (4.18)$$

Near a vertical boundary we introduce the discretization of a ‘half’ boundary volume (the hatched area shown in figure 4.2a), so that the entire computational domain is covered. This discretization is taken such that global momentum is conserved:

$$(\Omega_1^u)_{b,j} \frac{du_{b,j}}{dt} + \bar{u}_{1/2,j} u_{1/2,j} - \bar{u}_{b,j} u_{b,j} = -(\bar{p}_{1,j} - \bar{p}_{b,j}) + v \left( \left( \frac{\partial u}{\partial x} \right)_{1/2,j} - \left( \frac{\partial u}{\partial x} \right)_{b,j} \right). \quad (4.19)$$

We have left out the vertical fluxes, because they cancel upon summation in  $y$ -direction. After summation in  $x$ -direction we obtain:

$$1^T C_1^u(\bar{u}, \bar{v}) u = -\bar{u}_{b,j} u_{b,j} + (\text{right boundary contribution}), \quad (4.20)$$

$$1^T G_1^u \bar{p} = -\bar{p}_{b,j} + (\text{right boundary contribution}), \quad (4.21)$$

$$1^T \Delta_1^u \left( \frac{\partial u}{\partial x} \right) = - \left( \frac{\partial u}{\partial x} \right)_{b,j} + (\text{right boundary contribution}). \quad (4.22)$$

These are the discrete counterparts of the  $u$ -component of equations (4.15)-(4.17). Face-integrated quantities such as  $\bar{u}$  and  $\bar{p}$  are approximated by the midpoint method, which is an  $\mathcal{O}(h^2)$  approximation to the exact integral value. The discrete summation-by-parts expressions (4.20) and (4.21) do therefore *not* mimic the continuous relations *exactly*, except if the boundary values are constant or varying linearly along the boundary. One particular and important example is  $u_{b,j} = 0$ , when the discrete convective terms do not contribute to the global momentum balance, and (4.20) mimics (4.15) exactly in a discrete sense.

The finite volumes of the  $v$ -component are aligned with a vertical boundary, so no boundary volume is necessary. The discretization of the first volume reads (leaving out vertical fluxes):

$$(\Omega_1^v)_{1,j} \frac{dv_{1,j}}{dt} + \bar{u}_{1,j+1/2} v_{3/2,j} - \bar{u}_{b,j+1/2} v_{b,j} = v \left( \left( \frac{\partial v}{\partial x} \right)_{3/2,j} - \left( \frac{\partial v}{\partial x} \right)_{b,j} \right). \quad (4.23)$$

The discrete equivalent of the  $v$ -component of (4.15) is

$$1^T C_1^v(\bar{u}, \bar{v})v = -\bar{u}_{b,j+1/2}v_{b,j} + (\text{right boundary contribution}), \quad (4.24)$$

and similarly (4.17) becomes

$$1^T \Delta_1^v \left( \frac{\partial v}{\partial x} \right) = - \left( \frac{\partial v}{\partial x} \right)_{b,j} + (\text{right boundary contribution}). \quad (4.25)$$

The second order discretization of the convective terms does not require the imposition of ghost values. The diffusive gradients  $\frac{\partial u}{\partial x}$  and  $\frac{\partial v}{\partial x}$  are still to be expressed in terms of  $u$  and  $v$ . This will be done in section 4.1.3, in such a way that the diffusive terms are dissipative.

The analysis near a horizontal boundary can be performed in a similar fashion.

#### 4.1.2.2 Fourth order

In the interior, conservation of momentum is obtained as for the second order case, although the cancellation of interior fluxes for the coarse volumes takes place over a distance of  $3\Delta x$ : the flux through the right face of a coarse volume around  $u_{i,j}$  cancels with the flux through the left face of a coarse volume around  $u_{i+3,j}$ . The conserved quantity is  $1^T(\alpha_1\Omega_1^u - \Omega_3^u)u$ . Note that  $1^T\Omega_1^u 1 = \int d\Omega = |\Omega|$ , and that  $1^T\Omega_3^u 1 = 9|\Omega|$ : every small volume is covered by 9 different coarse volumes. The total momentum conserved by the discretization is therefore not the discrete analogue of the single volume integral  $\int_{\Omega} u d\Omega$ , but the analogue of a combination of two instead.

The implied discretization at the boundary is more involved than in the second order case. It is necessary to introduce boundary volumes at  $i = 1$ ,  $i = 2$  and  $i = 3$ , see figures 4.2b-4.2d, which are taken such that the entire domain is covered completely by finite volumes. The boundary volumes are such that on *each* coarse grid discretization (I, II and III) a summation-by-parts identity is satisfied:

$$\begin{aligned} \text{I} : -(\Omega_3^u)_I \frac{du_{b,j}}{dt} - \left( \bar{u}_{b,j}u_{b,j} - \bar{u}_{-1/2,j}u_{-1/2,j} \right) &= -(\bar{p}_{0,j} - \bar{p}_{b,j}) \\ &\quad - v \left( \left( \frac{\partial u}{\partial x} \right)_{b,j} - \left( \frac{\partial u}{\partial x} \right)_{-1/2,j} \right), \end{aligned} \quad (4.26)$$

$$\begin{aligned} \text{II} : (\Omega_3^u)_{II} \frac{du_{b,j}}{dt} + \bar{u}_{1/2,j}u_{1/2,j} - \bar{u}_{b,j}u_{b,j} &= -(\bar{p}_{1,j} - \bar{p}_{b,j}) \\ &\quad + v \left( \left( \frac{\partial u}{\partial x} \right)_{1/2,j} - \left( \frac{\partial u}{\partial x} \right)_{b,j} \right), \end{aligned} \quad (4.27)$$



$$\begin{aligned} \text{III} : (\Omega_3^u)_{III} \frac{du_{b,j}}{dt} + \bar{u}_{3/2,j} u_{3/2,j} - \bar{u}_{b,j} u_{b,j} = & -(\bar{p}_{2,j} - \bar{p}_{b,j}) \\ & + v \left( \left( \frac{\partial \bar{u}}{\partial x} \right)_{3/2,j} - \left( \frac{\partial \bar{u}}{\partial x} \right)_{b,j} \right). \end{aligned} \quad (4.28)$$

Here we have focused again on the horizontal fluxes. The boundary volumes  $(\Omega_3^u)_I$ ,  $(\Omega_3^u)_{II}$  and  $(\Omega_3^u)_{III}$  are illustrated by the hatched areas. The implied discretization of the boundary volume is given by the sum of the three expressions above:

$$\begin{aligned} (\Omega_3^u)_{b,j} \frac{du_{b,j}}{dt} + \bar{u}_{-1/2,j} u_{-1/2,j} + \bar{u}_{1/2,j} u_{1/2,j} + \bar{u}_{3/2,j} u_{3/2,j} - 3\bar{u}_{b,j} u_{b,j} = \\ -(\bar{p}_{0,j} + \bar{p}_{1,j} + \bar{p}_{2,j} - 3\bar{p}_{b,j}) \\ + v \left( \left( \frac{\partial \bar{u}}{\partial x} \right)_{-1/2,j} + \left( \frac{\partial \bar{u}}{\partial x} \right)_{1/2,j} + \left( \frac{\partial \bar{u}}{\partial x} \right)_{3/2,j} - 3 \left( \frac{\partial \bar{u}}{\partial x} \right)_{b,j} \right), \end{aligned} \quad (4.29)$$

where

$$(\Omega_3^u)_{b,j} = -(\Omega_3^u)_I + (\Omega_3^u)_{II} + (\Omega_3^u)_{III} = (\Omega_3^u)_{III}. \quad (4.30)$$

Summing the contribution of all (interior and boundary) fluxes gives

$$1^T C_3^u (\bar{u}, \bar{v}) u = -3\bar{u}_{b,j} u_{b,j} + (\text{right boundary contribution}), \quad (4.31)$$

$$1^T G_3^u p = -3\bar{p}_{b,j} + (\text{right boundary contribution}), \quad (4.32)$$

and

$$1^T \Delta_3^u \left( \frac{\partial \bar{u}}{\partial x} \right) = -3 \left( \frac{\partial \bar{u}}{\partial x} \right)_{b,j} + (\text{right boundary contribution}), \quad (4.33)$$

the discrete counterparts of equations (4.15)-(4.17).

For the  $v$ -component near a vertical boundary a different technique has to be applied. When introducing, like for the  $u$ -component, a ('negative') boundary volume for  $i = 1$  and a ('positive') boundary volume for  $i = 3$ , such that on each coarse grid momentum is conserved, one would obtain a discretization for a boundary volume with zero size, which is not appropriate when studying the energy equation. Instead, we construct the discretization for  $v_{1,j}$  as the sum of the boundary volumes of coarse grid II and III, see figure 4.3:

$$\begin{aligned} (\Omega_3^v)_{1,j} \frac{dv_{1,j}}{dt} + \bar{u}_{1,j+1/2} v_{3/2,j} + \bar{u}_{2,j+1/2} v_{5/2,j} - 2\bar{u}_{b,j+1/2} v_{b,j} = \\ v \left( \left( \frac{\partial \bar{v}}{\partial x} \right)_{5/2,j} + \left( \frac{\partial \bar{v}}{\partial x} \right)_{3/2,j} - 2 \left( \frac{\partial \bar{v}}{\partial x} \right)_{b,j} \right), \end{aligned} \quad (4.34)$$

where  $(\Omega_3^v)_{1,j} = \Omega_{II} + \Omega_{III}$ . In contrast to equation (4.29),  $v_{1,j}$  in equation (4.34) is an unknown in the system of equations and is solved for. Summation in  $x$ -direction gives

$$1^T C_3^v(\bar{u}, \bar{v})v = -3\bar{u}_{b,j+1/2}v_{b,j} + (\text{right boundary contribution}), \quad (4.35)$$

and

$$1^T \Delta_3^v \left( \frac{\partial v}{\partial x} \right) = -3 \left( \frac{\partial v}{\partial x} \right)_{b,j} + (\text{right boundary contribution}), \quad (4.36)$$

which are the discrete equivalents of the  $v$ -component of (4.15) and (4.17).

Finally it may be noted that momentum is conserved independently of the treatment of the ghost values (such as  $\bar{p}_{0,j}$ ,  $u_{-1/2,j}$ , etc.).

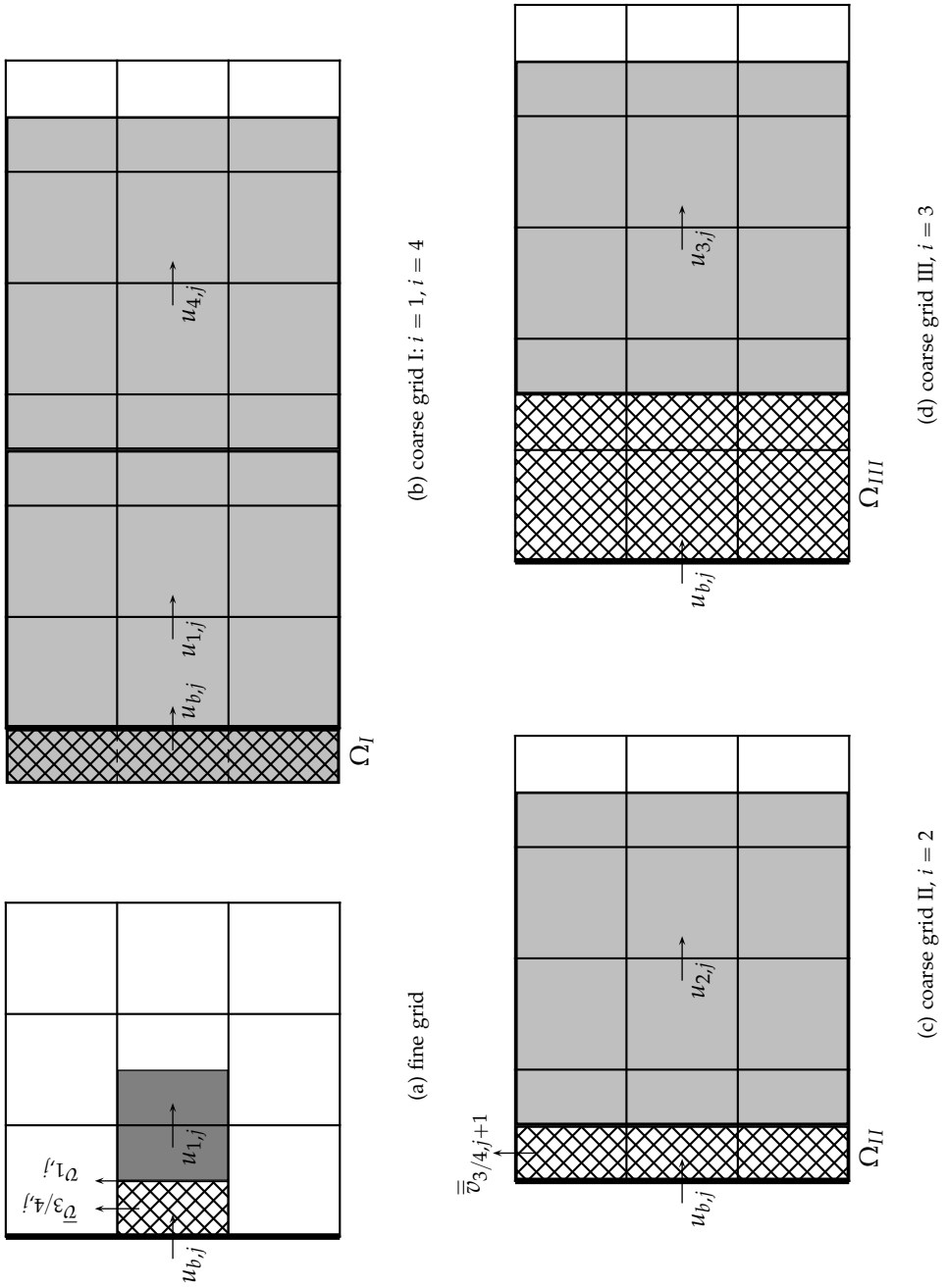


Figure 4.2: Momentum conservation near a boundary,  $u$ -momentum.

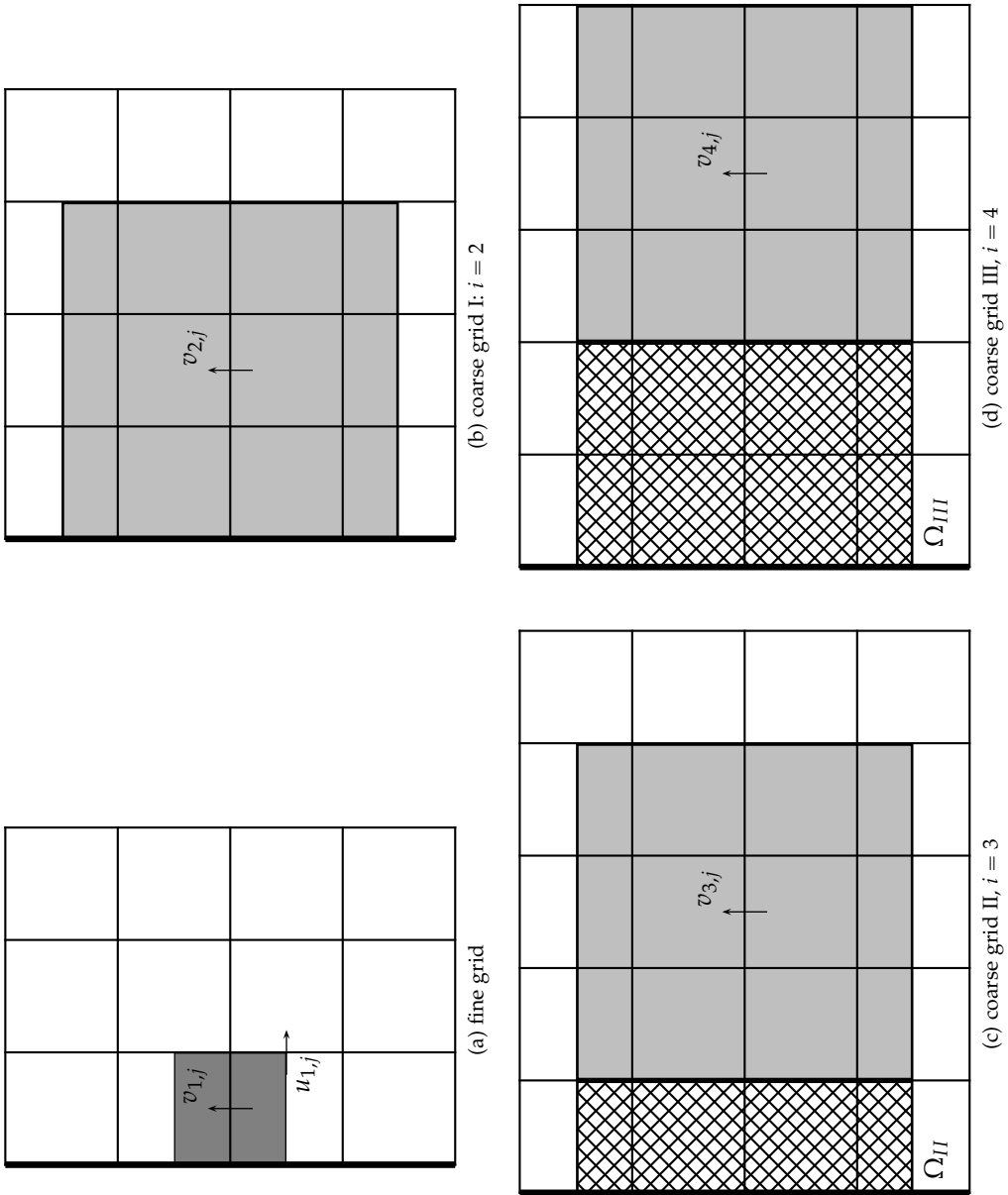


Figure 4-3: Momentum conservation near a boundary;  $v$ -momentum.

### 4.1.3 Kinetic energy

In this section we investigate if equations (2.13), (2.17) and (2.19) (with  $v$  replaced by  $u$ ), i.e.

$$\int_{\Omega} [(\mathbf{u} \cdot \nabla) \mathbf{u}] \cdot \mathbf{u} \, d\Omega = \frac{1}{2} \int_{\partial\Omega} [(\mathbf{u} \cdot \mathbf{u}) \mathbf{u}] \cdot \mathbf{n} \, d\Gamma, \quad (4.37)$$

$$\int_{\Omega} \nabla p \cdot \mathbf{u} \, d\Omega = \int_{\partial\Omega} p \mathbf{u} \cdot \mathbf{n} \, d\Gamma - \int_{\Omega} p \nabla \cdot \mathbf{u} \, d\Omega, \quad (4.38)$$

$$\int_{\Omega} \nabla^2 \mathbf{u} \cdot \mathbf{u} \, d\Omega = \int_{\partial\Omega} \mathbf{u} \cdot \frac{\partial \mathbf{u}}{\partial n} \, d\Gamma - \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{u} \, d\Omega, \quad (4.39)$$

can be satisfied in a discrete sense. The total kinetic energy (which should be conserved by the discretization in case of inviscid flow according to equation (2.22)) is obtained by multiplying the  $u$ -momentum equation with  $u$ , the  $v$ -momentum equation with  $v$ , like in the continuous case:

$$K := \frac{1}{2}(u, \Omega^u u) + \frac{1}{2}(v, \Omega^v v). \quad (4.40)$$

#### 4.1.3.1 Second order

Equation (4.37) indicates that in the interior the convective terms do not contribute to the kinetic energy. This can be proven in a discrete sense by showing that the divergence form (3.5) can be written in skew-symmetric form. Indeed, if  $C(c)$  is skew-symmetric, then  $u^T C(c) u = 0$  for any  $u$ . When constant (grid-independent) weights are used to interpolate  $\bar{u}$  and  $\bar{v}$ , equation (3.5) can be written in the skew-symmetric form

$$\begin{aligned} (C_1^u(\bar{u}, \bar{v})u)_{i,j} = & \bar{u}_{i+1/2,j} \frac{1}{2} u_{i+1,j} - \bar{u}_{i-1/2,j} \frac{1}{2} u_{i-1,j} + \bar{v}_{i+1/2,j} \frac{1}{2} u_{i,j+1} - \bar{v}_{i+1/2,j-1} \frac{1}{2} u_{i,j-1} + \\ & \frac{1}{2} u_{i,j} \left[ \frac{1}{2} (\bar{u}_{i+1,j} - \bar{u}_{i,j} + \bar{v}_{i+1,j} - \bar{v}_{i+1,j-1}) + \frac{1}{2} (\bar{u}_{i,j} - \bar{u}_{i-1,j} + \bar{v}_{i,j} - \bar{v}_{i,j-1}) \right]. \end{aligned} \quad (4.41)$$

The term in square brackets features the sum of the divergence of the pressure volumes to the left and right of  $u_{i,j}$ : the diagonal coefficient of the convective operator is therefore zero, like in the continuous case (equation (2.12)).

Near boundaries the convective discretization should not be skew-symmetric in case energy flows in or out of the domain, corresponding to the boundary integral in equation (2.13). We elaborate the discretization near a boundary, given by (4.19), where now the vertical fluxes are considered too:

$$(C_1^u(\bar{u}, \bar{v})u)_{b,j} = \bar{u}_{1/2,j} u_{1/2,j} - \bar{u}_{b,j} u_{b,j} + \bar{v}_{3/4,j} u_{b,j+1/2} - \bar{v}_{3/4,j-1} u_{b,j-1/2}. \quad (4.42)$$

These terms are used to rewrite the divergence form into skew-symmetric form. In matrix notation  $C_1^u(\bar{u}, \bar{v})$  reads

$$C_1^u(\bar{u}, \bar{v}) = \begin{bmatrix} -\bar{u}_{b,j} + \frac{1}{2}\bar{u}_{1/2,j} + \frac{1}{2}(\bar{v}_{3/4,j} - \bar{v}_{3/4,j-1}) & \frac{1}{2}\bar{u}_{1/2,j} \\ -\frac{1}{2}\bar{u}_{1/2,j} & \begin{array}{ccc} 0 & \frac{1}{2}\bar{u}_{3/2,j} & \\ -\frac{1}{2}\bar{u}_{3/2,j} & 0 & \ddots \\ & \ddots & \end{array} \end{bmatrix}. \quad (4.43)$$

The first row represents the equation for  $u_{b,j}$ , the first column represents the contribution of  $u_{b,j}$ . The lower-right part of the matrix (which is indicated by the line) represents the internal discretization. The matrix  $C_1^u(\bar{u}, \bar{v})$  is skew-symmetric except for its upper left entry, so the discrete version of (4.37) becomes

$$u^T C_1^u(\bar{u}, \bar{v}) u = u_{b,j}^2 \left( -\bar{u}_{b,j} + \frac{1}{2}\bar{u}_{1/2,j} + \frac{1}{2}(\bar{v}_{3/4,j} - \bar{v}_{3/4,j-1}) \right). \quad (4.44)$$

The vertical flux  $\bar{v}_{3/4,j}$  should be computed as  $\bar{v}_{3/4,j} = \frac{1}{2}\bar{v}_{1,j}$  since this leads to

$$\begin{aligned} u^T C_1^u(\bar{u}, \bar{v}) u &= -\frac{1}{2}u_{b,j}^2 \bar{u}_{b,j} + \frac{1}{4}u_{b,j}^2 (\bar{u}_{1,j} - \bar{u}_{b,j} + \bar{v}_{1,j} - \bar{v}_{1,j-1}), \\ &= -\frac{1}{2}u_{b,j}^2 \bar{u}_{b,j}, \end{aligned} \quad (4.45)$$

which is indeed the discrete equivalent of (4.37). Property (4.45) is only exact if  $u_b$  is constant or varying linearly along a boundary; in all other cases it is an  $\mathcal{O}(h^2)$  approximation to equation (4.37).

The convective matrix for the  $v$ -component reads

$$C_1^v(\bar{u}, \bar{v}) v = \begin{bmatrix} \frac{1}{2}\bar{u}_{1,j+1/2} + \frac{1}{2}(\bar{v}_{1,j+1/2} - \bar{v}_{1,j-1/2}) & \frac{1}{2}\bar{u}_{1,j+1/2} \\ -\frac{1}{2}\bar{u}_{1,j+1/2} & 0 \\ & \ddots \\ & & \frac{1}{2}\bar{u}_{2,j+1/2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \end{bmatrix} + \begin{bmatrix} -\bar{u}_{b,j+1/2} v_{b,j} \\ 0 \\ \vdots \end{bmatrix}, \quad (4.46)$$

and the contribution to the energy equation becomes

$$v^T C_1^v(\bar{u}, \bar{v}) v = -\frac{1}{2}v_{1,j} \bar{u}_{b,j+1/2} (2v_{b,j} - v_{1,j}). \quad (4.47)$$

Surprisingly, this term does not evaluate to  $-\frac{1}{2}v_{b,j}^2 \bar{u}_{b,j+1/2}$ , which one might have expected intuitively. However, upon Taylor expanding  $v_{1,j}$  around  $v_{b,j}$ ,

$$v_{1,j} = v_{b,j} + \frac{1}{2}h(v_x)_{b,j} + \mathcal{O}(h^2), \quad (4.48)$$

we see that

$$\frac{1}{2}v_{1,j}\bar{u}_{b,j+1/2}(2v_{b,j} - v_{1,j}) = \frac{1}{2}\bar{u}_{b,j+1/2}v_{b,j}^2 + \mathcal{O}(h^2), \quad (4.49)$$

which is, like (4.45), a second order approximation to the continuous boundary integral. However, in contrast to (4.45), this approximation is *not exact* for constant  $u_b$  and  $v_b$ .

For the pressure gradient terms the discrete second order version of (4.38) reads (here we concentrate again on summing in  $x$ -direction):

$$\begin{aligned} u^T G_1^u \bar{p} + v^T G_1^v \bar{p} &= \bar{u}^T G_1^u p + \bar{v}^T G_1^v p & (4.50) \\ &= \bar{u}_{b,j}(p_{1,j} - p_{b,j}) + \bar{u}_{1,j}(p_{2,j} - p_{1,j}) + \dots \\ &\quad \bar{v}_{1,j}(p_{1,j+1} - p_{1,j}) + \bar{v}_{1,j-1}(p_{1,j} - p_{1,j-1}) + \dots \\ &= -\bar{u}_{b,j}p_{b,j} - p_{1,j} \underbrace{(\bar{u}_{1,j} - \bar{u}_{b,j} + \bar{v}_{1,j} - \bar{v}_{1,j-1})}_{(M_1 u)_{1,j}} - p_{2,j} \underbrace{(\dots)}_{(M_1 u)_{2,j}} + \dots \\ &= -\bar{u}_{b,j}p_{b,j} + (\text{right boundary contribution}). & (4.51) \end{aligned}$$

In equation (4.50) we have switched from  $\bar{p}$  to  $\bar{u}$ : this is valid, even on non-uniform grids, since both face quantities are approximated using (3.4). In matrix notation the above result can be written as

$$u^T G_1 \bar{p} = -\bar{u}_{b,j}p_{b,j} + (\text{right boundary contribution}) - p^T M_1 \bar{u}. \quad (4.52)$$

In case of no-penetration or periodic boundary conditions one obtains the well-known symmetry relation between the divergence and the gradient operator:

$$M_1 = -G_1^T. \quad (4.53)$$

The contribution of the diffusive terms to the energy equation reads

$$\begin{aligned} u^T D_1^u u &= \\ &= u_{b,j} \left( \left( \frac{\partial u}{\partial x} \right)_{1/2,j} - \left( \frac{\partial u}{\partial x} \right)_{b,j} \right) + u_{1,j} \left( \left( \frac{\partial u}{\partial x} \right)_{3/2,j} - \left( \frac{\partial u}{\partial x} \right)_{1/2,j} \right) + \dots & (4.54) \\ &= -u_{b,j} \left( \frac{\partial u}{\partial x} \right)_{b,j} - \left( \frac{\partial u}{\partial x} \right)_{1/2,j} (u_{1,j} - u_{b,j}) - \left( \frac{\partial u}{\partial x} \right)_{3/2,j} (u_{2,j} - u_{1,j}) + \dots \end{aligned}$$

With the following discretization of the first order derivatives,

$$\left(\overline{\frac{\partial u}{\partial x}}\right)_{i+1/2,j} = \Delta y \left(\frac{u_{i+1,j} - u_{i,j}}{\Delta x}\right), \quad (4.55)$$

equation (4.54) can be written as

$$u^T D_1^u u = -u_{b,j} \left(\overline{\frac{\partial u}{\partial x}}\right)_{b,j} - \Delta x \Delta y \left( \left(\frac{u_{1,j} - u_{b,j}}{\Delta x}\right)^2 + \left(\frac{u_{2,j} - u_{1,j}}{\Delta x}\right)^2 + \dots \right), \quad (4.56)$$

which is the discrete equivalent of (4.39). The derivative at the boundary does not have to be specified since it is not needed in practical calculations.

The  $v$ -component can be treated in a similar fashion:

$$\begin{aligned} v^T D_1^v v &= v_{1,j} \left( \left(\overline{\frac{\partial v}{\partial x}}\right)_{3/2,j} - \left(\overline{\frac{\partial v}{\partial x}}\right)_{b,j} \right) + v_{2,j} \left( \left(\overline{\frac{\partial v}{\partial x}}\right)_{5/2,j} - \left(\overline{\frac{\partial v}{\partial x}}\right)_{3/2,j} \right) + \dots \\ &= -v_{1,j} \left(\overline{\frac{\partial v}{\partial x}}\right)_{b,j} - \Delta x \Delta y \left( \left(\frac{v_{2,j} - v_{1,j}}{\Delta x}\right)^2 + \left(\frac{v_{3,j} - v_{2,j}}{\Delta x}\right)^2 + \dots \right). \end{aligned} \quad (4.57)$$

By taking the approximation

$$\left(\overline{\frac{\partial v}{\partial x}}\right)_{b,j} = \Delta y \left(\frac{v_{1,j} - v_{b,j}}{\frac{1}{2}\Delta x}\right), \quad (4.58)$$

equation (4.57) can be written as

$$\begin{aligned} v^T D_1^v v &= -v_{b,j} \Delta y \left(\frac{v_1 - v_b}{\frac{1}{2}\Delta x}\right) \\ &\quad - \Delta x \Delta y \left( \frac{1}{2} \left(\frac{v_{1,j} - v_{b,j}}{\Delta x}\right)^2 + \left(\frac{v_{2,j} - v_{1,j}}{\Delta x}\right)^2 + \left(\frac{v_{3,j} - v_{2,j}}{\Delta x}\right)^2 + \dots \right), \end{aligned} \quad (4.59)$$

which is the discrete counterpart of equation (4.39).

#### 4.1.3.2 Fourth order

Like the second order scheme, the fourth order convective discretization (3.15) can be rewritten in skew-symmetric form. To that end, the diagonal coefficient is expressed in terms of the divergence of four surrounding finite volumes. For details we refer to [203].

Near boundaries this skew-symmetric form is used to derive the contribution to the energy equation. The analysis is detailed in appendix A.2.1. An important point is the choice of the ghost values. Here the ghost values for the  $u$ -component at a vertical boundary are defined by a second-order extrapolation, for example



$u_{-1,j} = 2u_{b,j} - u_{1,j}$ . The ghost values for the  $v$ -component at a vertical boundary are given by Neumann type conditions, see (4.12), (4.13) and (A.26). This leads to

$$u^T C_3^u(\bar{u}, \bar{v})u = -\frac{3}{2}\bar{u}_{b,j}u_{b,j}^2 + \mathcal{O}(h^2), \quad (4.60)$$

$$v^T C_3^v(\bar{u}, \bar{v})v = -\frac{3}{2}\bar{u}_{b,j+1/2}v_{b,j}^2 + \mathcal{O}(h^2). \quad (4.61)$$

These equations reveal that the boundary integrals in (4.37) are not satisfied discretely (not even for constant boundary data) but contain an  $\mathcal{O}(h^2)$  error. However, in the important case of  $\bar{u}_{b,j} = 0$  the discretization mimics the continuous property (no energy contribution by the convective terms) exactly. At a horizontal boundary the same analysis applies.

The contribution of the pressure gradient to the energy equation for the fourth order method is given by

$$\begin{aligned} u^T G_3^u \bar{p} + v^T G_3^v \bar{p} &= \bar{u}_{b,j}(p_{0,j} + p_{1,j} + p_{2,j} - 3p_{b,j}) + \bar{u}_{1,j}(p_{3,j} - p_{0,j}) \\ &\quad + \bar{u}_{2,j}(p_{4,j} - p_{1,j}) + (v\text{-component}) \\ &= -3\bar{u}_{b,j}p_{b,j} + p_{0,j}(\bar{u}_{1,j} - \bar{u}_{b,j}) \\ &\quad + p_{1,j}(\bar{u}_{2,j} - \bar{u}_{b,j} + \bar{v}_{1,j+1} - \bar{v}_{1,j-2}) + \dots \end{aligned} \quad (4.62)$$

$$= -3\bar{u}_{b,j}p_{b,j} + p_{1,j} \underbrace{(\bar{u}_{2,j} + \bar{u}_{1,j} - 2\bar{u}_{b,j} + \bar{v}_{1,j+1} - \bar{v}_{1,j-2})}_{(M_3 u)_{1,j}} + \dots \quad (4.63)$$

$$= -3\bar{u}_{b,j}p_{b,j} + (\text{right boundary contribution}). \quad (4.64)$$

The Neumann condition  $p_{0,j} = p_{1,j}$  is necessary to write equation (4.62) as (4.63): the term in brackets in (4.63) then evaluates to 0 since it is the divergence of a coarse volume around  $p_{1,j}$  (see section 4.1.1.2). In short notation, the fourth order scheme satisfies

$$u^T G_3^u \bar{p} = -3\bar{u}_{b,j}p_{b,j} + (\text{right boundary contribution}) - p^T M_3 \bar{u}. \quad (4.65)$$

There are two important notes to be made regarding the condition  $p_{0,j} = p_{1,j}$ . Firstly, this condition *follows* from the discretization of the divergence operator and by requiring (4.65). Consequently, for the discretization of the interior points, *no explicit pressure boundary conditions need to be specified* separately from those for the velocity, as was the case for the second order discretization. Notice that pressure boundary conditions were not specified for the continuous problem either. Secondly, the ‘implied’ discrete Neumann condition is *not homogeneous*, in contrast to what  $p_{0,j} = p_{1,j}$  suggests. This will be detailed in section 4.1.6. We note that boundary condition (4.8), as suggested in [203], effectively leads to a Dirichlet condition for the pressure. Near solid walls this results into an artificial boundary layer in the pressure.

Lastly, we discuss the contribution of the diffusive terms to the energy equation. In appendix A.2.2 we show that for the fourth order scheme:

$$\begin{aligned}
 u^T D^u u = & - \left( \frac{\partial u}{\partial x} \right)_{b,j} (\alpha_1 - 9) u_{b,j} + \dots \\
 & - 72 \Delta x \Delta y \left[ \left( \frac{9}{8} \frac{u_{1,j} - u_{b,j}}{\Delta x} - \frac{1}{8} \frac{u_{2,j} + u_{1,j} - 2u_{b,j}}{3\Delta x} \right)^2 + \right. \\
 & \left. \left( \frac{9}{8} \frac{u_{2,j} - u_{1,j}}{\Delta x} - \frac{1}{8} \frac{u_{3,j} - u_{b,j}}{3\Delta x} \right)^2 + \dots \right]. \quad (4.66)
 \end{aligned}$$

This is the discrete equivalent of equation (4.39), with the first term representing the boundary integral, and the second term the volume integral.

#### 4.1.4 Extension to higher order

The boundary conditions that are derived in sections 4.1.1-4.1.3 are all based on first or second order extrapolations. This limits the local truncation error of the fourth order scheme, as will be shown in section 4.2. In this section we will investigate if the ghost values can be obtained by higher order extrapolations while still satisfying discrete equivalents of the continuous properties derived in chapter 2.2. We start with the divergence operator. Equation (4.5) can be generalized to

$$\bar{u}_{-1,j} = \delta_0 \bar{u}_{b,j} + \delta_1 \bar{u}_{1,j} + \delta_2 \bar{u}_{2,j} + \delta_3 \bar{u}_{3,j} + \dots, \quad (4.67)$$

hence the divergence operator becomes

$$M_3^u \bar{u} = \begin{bmatrix} -\delta_0 & -\delta_1 & 1 - \delta_2 & -\delta_3 & & \\ -1 & & & 1 & & \\ & -1 & & & 1 & \\ & & \ddots & \ddots & \ddots & \\ & & & & & \ddots \end{bmatrix} \begin{bmatrix} \bar{u}_{b,j} \\ \bar{u}_{1,j} \\ \bar{u}_{2,j} \\ \bar{u}_{3,j} \\ \vdots \end{bmatrix}. \quad (4.68)$$

The only solution that satisfies the summation-by-parts identity (4.7) is  $\delta_0 = 2$ ,  $\delta_1 = -1$ ,  $\delta_2 = \delta_3 = \dots = 0$ , which equals boundary condition (4.5). The other boundary conditions derived in section 4.1.1, equations (4.10)-(4.13), cannot be changed following similar reasoning.

We can also try more general formulations for  $u_{-1/2,j}$  and  $u_{1/2,j}$  (see equations (A.11)-(A.12)):

$$u_{-1/2,j} = \beta_0 u_{b,j} + \beta_1 u_{1,j} + \beta_2 u_{2,j} + \beta_3 u_{3,j} + \dots, \quad (4.69)$$

$$u_{1/2,j} = \gamma_0 u_{b,j} + \gamma_1 u_{1,j} + \gamma_2 u_{2,j} + \gamma_3 u_{3,j} + \dots \quad (4.70)$$

The coefficients  $\beta$  and  $\gamma$  are to be determined from order and symmetry requirements. Requiring skew-symmetry in case of  $\bar{\bar{u}}_{b,j} = 0$  leads to

$$\delta_0 - \delta_1 + \beta_1 = 0, \quad (4.71)$$

$$-\delta_2 - \beta_0 + \beta_2 = 0, \quad (4.72)$$

$$\delta_2 - \beta_1 = 0, \quad (4.73)$$

$$\delta_1 = \beta_2 = \frac{1}{2}, \quad (4.74)$$

$$\delta_3 = \beta_3 = 0. \quad (4.75)$$

Requiring a consistent (first-order) interpolation gives the conditions  $\delta_0 + \delta_1 + \delta_2 = 1$  and  $\beta_0 + \beta_1 + \beta_2 = 1$ . The solution to these equations leads to interpolations (A.11)-(A.12). The resulting coefficients also satisfy the conditions for second order interpolation, but not higher order conditions. Skew-symmetry and higher order are therefore conflicting requirements. Higher-order treatment of boundary conditions, as in [116] for instance, violates the skew-symmetry of the convective operator.

An important starting point in our analysis is that (skew-)symmetry is defined in terms of the standard inner product, equation (4.1). When generalizing the inner product higher order can be combined with symmetry properties, see e.g. [184, 106] for hyperbolic and parabolic problems, respectively. Perhaps also for the incompressible Navier-Stokes equations such an adapted inner product can be found. However, the pressure will then contribute to the energy equation. This is a subject of further research.

#### 4.1.5 Summary

In table 4.1 we summarize the old and new boundary conditions for the fourth order scheme. The old boundary conditions (4.79) form a rather unclear mix of Dirichlet and Neumann conditions for both  $u$ - and  $v$ -component. The new boundary conditions (4.80) feature Dirichlet conditions for the  $u$ -component, whereas Neumann type conditions are prescribed for the  $v$ -component. Furthermore, our new discretization of the  $v$ -component near the wall does not require the prescription of  $v_{-2,j}$ .

In case of no-slip or periodic boundary conditions, the symmetry properties of the discrete operators can be summarized as

$$(Gp, u) = -(p, Mu), \quad (4.76)$$

$$(C(c)u, u) = 0, \quad (4.77)$$

$$(Du, v) = (u, Dv), \quad (Du, u) \leq 0. \quad (4.78)$$

The first property is the relation between the divergence and gradient operator. The second property indicates that the convective terms do not change the total energy

of the flow (if  $Mc = 0$ ). The third property shows that the diffusive terms dissipate energy.

Old [203]		New	
$u_{-1,j} = 2u_{b,j} - u_{1,j}$		$u_{-1,j} = 2u_{b,j} - u_{1,j}$	
$u_{-2,j} = u_{2,j}$		$u_{-2,j} = 2u_{b,j} - u_{2,j}$	
$\bar{u}_{-1,j} = \bar{u}_{1,j}$		$\bar{u}_{-1,j} = 2\bar{u}_{b,j} - \bar{u}_{1,j}$	
$\bar{\bar{u}}_{-1,j} = \bar{\bar{u}}_{1,j}$		$\bar{\bar{u}}_{-1,j} = 2\bar{\bar{u}}_{b,j} - \bar{\bar{u}}_{1,j}$	
$\bar{\bar{u}}_{-2,j} = \bar{\bar{u}}_{2,j}$	(4.79)	$\bar{\bar{u}}_{-2,j} = 2\bar{\bar{u}}_{b,j} - \bar{\bar{u}}_{2,j}$	(4.80)
$v_{0,j} = 2v_{b,j} - v_{1,j}$		$v_{0,j} = v_{1,j}$	
$\bar{v}_{0,j} = 2\bar{v}_{b,j} - \bar{v}_{1,j}$		$\bar{v}_{0,j} = \bar{v}_{1,j}$	
$\bar{\bar{v}}_{0,j} = 2\bar{\bar{v}}_{b,j} - \bar{\bar{v}}_{1,j}$		$\bar{\bar{v}}_{0,j} = \bar{\bar{v}}_{1,j}$	
$v_{-2,j} = v_{3,j}$			

Table 4.1: Old and new boundary conditions for fourth order scheme.

#### 4.1.6 Pressure

In the incompressible Navier-Stokes equations the pressure is a Lagrange multiplier which makes the velocity field divergence-free. There is no pressure evolution equation and as such it is not a conserved quantity like mass, momentum and energy. Furthermore, the original PDEs (2.2)-(2.3) do not contain boundary conditions for the pressure. However, an equation for the pressure with associated boundary conditions can be *derived*, and it is important to see if a discretization of the incompressible Navier-Stokes equations inherits the continuous properties of the pressure in a discrete sense. The pressure Poisson equation (PPE) is obtained by taking the time-derivative of the continuity equation and substituting the momentum equation [59]:

$$\nabla^2 p = \nabla \cdot N(\mathbf{u}), \quad (4.81)$$

where  $N(\mathbf{u}) = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu \nabla^2 \mathbf{u}$ . On no-slip boundaries, where  $\mathbf{u} = \mathbf{u}_b$ , the boundary condition for the pressure is found by taking the normal component of the momentum equations at the boundary, i.e.

$$\frac{\partial p}{\partial n} = \left( -\frac{\partial \mathbf{u}_b}{\partial t} + N(\mathbf{u}) \right) \cdot \mathbf{n}. \quad (4.82)$$

Like in the continuous case, in a discrete setting the pressure Poisson equation is formed by differentiating the divergence-free constraint (in time) and substituting the momentum equations:

$$M\Omega^{-1}Gp = M\Omega^{-1}(-C(c)u + Du) = MN(u). \quad (4.83)$$

The boundary conditions are 'built' into the operators  $M$  and  $G$ ; see the discussion at the end of section 4.1.3. To obtain the effective boundary conditions for the pressure, we compare the Laplace operator at the boundary to the Laplace operator at an inner point, and subtract these two [197, 59]. For the second order method it is known that this gives a boundary condition which is the discrete equivalent of equation (4.82). In appendix A.1 we detail such an analysis for the fourth order method. The resulting effective boundary conditions at  $i = 1$ ,  $i = 2$  and  $i = 3$  are

$$\begin{aligned} \frac{1}{\Delta x} \left( \frac{1}{8}(p_{3,j} - p_{-2}) - \frac{27}{4}(p_{2,j} - p_{-1,j}) + \frac{783}{8}(p_{1,j} - p_{0,j}) \right) &= \alpha_1 N_{b,j}^u \\ &\quad - \alpha_2 (N_{-1,j}^u + N_{1,j}^u) + (-\alpha_1 + 2\alpha_2) \dot{u}_{b,j}, \end{aligned} \quad (4.84)$$

$$\frac{1}{\Delta x} \left( \frac{1}{8}(p_{2,j} - p_{-1,j}) - \frac{27}{4}(p_{1,j} - p_{0,j}) \right) = -\alpha_2 N_{b,j}^u + \alpha_2 \dot{u}_{b,j}, \quad (4.85)$$

$$\frac{1}{\Delta x} \frac{1}{8}(p_{1,j} - p_{0,j}) = 0. \quad (4.86)$$

The effective pressure boundary conditions at the three points  $i = 1, 2$ , and  $3$  are *not* consistent approximations to the continuous boundary conditions (4.82). This appears to be in line with a remark made by Sanjiv et al. [156], who state 'Whenever the discrete PPE is generated from a consistent (*but low order*) discretization of the Navier-Stokes equations, the resulting PPE will always converge to (4.82)'. *On average* the pressure boundary conditions are still correct. This follows by summing the effective boundary conditions for  $i = 1, i = 2$  and  $i = 3$ , leading to

$$\begin{aligned} \frac{1}{\Delta x} \left( \frac{1}{8}(p_{3,j} - p_{-2}) - \frac{53}{8}(p_{2,j} - p_{-1,j}) + \frac{365}{4}(p_{1,j} - p_{0,j}) \right) &= \\ (\alpha_1 - \alpha_2) N_{b,j}^u - \alpha_2 (N_{1,j}^u + N_{-1,j}^u) + (-\alpha_1 + 3\alpha_2) \dot{u}_{b,j}. \end{aligned} \quad (4.87)$$

Expanding the left side in Taylor expansions around the boundary gives

$$\begin{aligned} \frac{1}{\Delta x} \left( \frac{1}{8}(p_{3,j} - p_{-2}) - \frac{53}{8}(p_{2,j} - p_{-1,j}) + \frac{365}{4}(p_{1,j} - p_{0,j}) \right) &= \\ 72(p_x)_b - 3\Delta x^2 p_{xxx} + \mathcal{O}(\Delta x^4). \end{aligned} \quad (4.88)$$

Similarly, expansion of the right side gives

$$(\alpha_1 - \alpha_2) N_{b,j}^u - \alpha_2 (N_{1,j}^u + N_{-1,j}^u) + (-\alpha_1 + 3\alpha_2) \dot{u}_{b,j} =$$

$$-72\dot{u}_{b,j} + 72N_{b,j}^u - 3\Delta x^2(N_{xx}^u)_{b,j} + \mathcal{O}(\Delta x^4). \quad (4.89)$$

Upon division by the effective (pressure) volume size we find that the fourth order method leads to a consistent effective boundary condition for the pressure:

$$(p_x)_b = N_{b,j}^u - \dot{u}_{b,j} + \mathcal{O}(\Delta x^2), \quad (4.90)$$

which is the discrete approximation to equation (4.82).

#### 4.2 ORDER OF ACCURACY ANALYSIS

In this section we investigate the effect of symmetry-preserving boundary conditions on local and global order behavior. We consider the linear convection-diffusion equation as a special case of the Navier-Stokes equations (2.3) in one dimension:

$$c \frac{d\hat{u}}{dx} = \nu \frac{d^2\hat{u}}{dx^2}, \quad 0 \leq x \leq 1, \quad c < 0, \quad \nu > 0, \quad (4.91)$$

where  $c$  is the convecting velocity,  $\hat{u}$  the convected velocity and  $\nu$  the viscosity coefficient. To study the effect of no-slip conditions, equation (4.91) is supplemented with Dirichlet boundary conditions:

$$\hat{u}(0) = u_b = 0, \quad \hat{u}(1) = 1. \quad (4.92)$$

In operator notation equation (4.91) can be concisely written as

$$\hat{L}\hat{u} = \hat{f}, \quad (4.93)$$

where

$$\hat{L} \equiv c \frac{d}{dx} - \nu \frac{d^2}{dx^2}, \quad \hat{f} = 0. \quad (4.94)$$

For small values of  $\nu$  a boundary layer develops at  $x = 0$ . Defining  $\varepsilon = \nu/c$  ( $\varepsilon < 0$ ), the exact solution to equation (4.91) is given by

$$\hat{u}(x) = \frac{e^{x/\varepsilon} - 1}{e^{1/\varepsilon} - 1}. \quad (4.95)$$

##### 4.2.1 Discretization

The discretization of the convective term, equation (3.15), is restricted to 1D by taking  $c = c_x = \text{constant}$  and  $c_y = 0$ :

$$(Cu)_i = \alpha_1 \left( \frac{u_{i+1} + u_i}{2} - \frac{u_i + u_{i-1}}{2} \right) - \left( \frac{u_{i+3} + u_i}{2} - \frac{u_i + u_{i-3}}{2} \right), \quad (4.96)$$

where  $\alpha_1 = 3^3 = 27$ . Without loss of generality we consider the discretization at the left boundary (where the boundary layer is located). Following section 4.1.5, these boundary conditions are given by equation (4.80):

$$u_0 = u_b, \quad u_{-1} = 2u_b - u_1, \quad u_{-2} = 2u_b - u_2. \quad (4.97)$$

Restricting the convective discretization for the coarse grid, equation (A.13), to one dimension ( $\bar{u} = c, \bar{v} = 0$ ) leads to

$$C_3 = \begin{bmatrix} -\frac{1}{2} & 0 & 0 & \frac{1}{2} \\ -1 & 0 & \frac{1}{2} & 0 & \ddots \\ -1 & \frac{1}{2} & 0 & 0 & \\ -\frac{1}{2} & 0 & 0 & \ddots \\ \ddots & & & \ddots & \end{bmatrix}. \quad (4.98)$$

The lower right part of the matrix is the discretization of the interior points, which is used in practical implementation. We note that there is a fundamental difference between the 2D nonlinear case and the 1D linear case. In the 2D nonlinear case  $c = u$ , and  $C_3$  is skew-symmetric for  $u_b = 0$ . A homogeneous Dirichlet condition for  $u$  implies a homogeneous Dirichlet condition for  $c$ . In the 1D linear case,  $c = \text{constant}$  is prescribed, which is different from the boundary condition for  $u$ . The convective matrix is therefore *not* skew-symmetric.

The resulting scheme is written as follows:

$$Lu = f, \quad (4.99)$$

where

$$L = cC - vD, \quad (4.100)$$

and

$$f = -c f^c + v f^d. \quad (4.101)$$

The upper-left corner of the convective part reads (considering only interior points):

$$C = \frac{1}{2} \begin{bmatrix} 0 & \alpha_1 - 1 & 0 & -1 \\ -\alpha_1 - 1 & 0 & \alpha_1 & 0 & -1 \\ 0 & -\alpha_1 & 0 & \alpha_1 & 0 & \ddots \\ 1 & 0 & -\alpha_1 & 0 & \alpha_1 \\ \ddots & & & \ddots & & \ddots \end{bmatrix}. \quad (4.102)$$

The diffusive part reads

$$D = \frac{1}{24h} \begin{bmatrix} -2\alpha_1^2 + 2\alpha_1 - 2 & \alpha_1^2 + 2\alpha_1 - 1 & -2\alpha_1 & 1 & & \\ \alpha_1^2 + 2\alpha_1 - 1 & -2\alpha_1^2 - 2 & \alpha_1^2 + 2\alpha_1 & -2\alpha_1 & \ddots & \\ -2\alpha_1 & \alpha_1^2 + 2\alpha_1 & -2\alpha_1^2 - 2 & \alpha_1^2 + 2\alpha_1 & & \\ 1 & -2\alpha_1 & \alpha_1^2 + 2\alpha_1 & -2\alpha_1^2 - 2 & \ddots & \\ & \ddots & & & \ddots & \end{bmatrix}, \quad (4.103)$$

and the right-hand side vectors read

$$f^c = \frac{u_b}{2} \begin{bmatrix} -(\alpha_1 - 2) \\ 2 \\ 1 \\ 0 \\ \vdots \end{bmatrix}, \quad f^d = \frac{u_b}{24h} \begin{bmatrix} \alpha_1^2 - 2\alpha_1 + 2 \\ -2\alpha_1 + 2 \\ 1 \\ 0 \\ \vdots \end{bmatrix}. \quad (4.104)$$

The discretization at the right boundary is done in a similar fashion.

#### 4.2.2 Local truncation error

We first study the local truncation error and then relate it to the global truncation error. The local truncation error is defined as

$$\tau \equiv L\hat{u} - f. \quad (4.105)$$

Note that strictly speaking a restriction operator is necessary to map the continuous function  $\hat{u}$  to the space of discrete functions in order to be able to apply  $L$ ; in this case the restriction operator is a simple injection:  $\hat{u}_i = \hat{u}(x_i)$ . The global error is defined as

$$e \equiv \hat{u} - u, \quad (4.106)$$

which is related to  $\tau$  as

$$Le = \tau. \quad (4.107)$$

First we discuss the local truncation error of the convective terms, on a uniform grid. We Taylor expand the exact solution  $\hat{u}$  around a point  $x_i$ :

$$\hat{u}(x_i + h) = \hat{u}(x_i) + h\hat{u}_i^{(1)} + \frac{1}{2}h^2\hat{u}_i^{(2)} + \mathcal{O}(h^3), \quad (4.108)$$



where  $\hat{u}_i^{(n)} = \left(\frac{d^n \hat{u}}{dx^n}\right)_i$ , and we apply operator  $C$ :

$$\tau_i^c = -\frac{9}{5}h^5\hat{u}_i^{(5)} + \mathcal{O}(h^7). \quad (4.109)$$

The truncation error is fifth order instead of fourth order due to a factor  $h$  hidden in  $C$  (due to the integration over a finite volume). To find  $\tau$  at  $i = 1$  and  $2$ , where the stencil changes due to boundary conditions, we also employ a Taylor expansion for  $\hat{u}_b$ :

$$\hat{u}_b = \hat{u}_1 - h\hat{u}_1^{(1)} + \frac{1}{2}h^2\hat{u}_1^{(2)} + \mathcal{O}(h^3). \quad (4.110)$$

This leads to the following truncation errors:

$$\tau_1^c = -2h^2\hat{u}_1^{(2)} + \mathcal{O}(h^3), \quad (4.111)$$

$$\tau_2^c = -\frac{1}{2}h^2\hat{u}_2^{(2)} + \mathcal{O}(h^3). \quad (4.112)$$

It turns out that the local truncation error of the convective discretization at  $i = 1$  and  $i = 2$  is limited to first order.

The same procedure is followed to analyze the truncation error of the diffusive terms:

$$\tau_i^d = -\frac{9}{40}h^5\hat{u}_i^{(6)} + \mathcal{O}(h^7), \quad (4.113)$$

$$\tau_1^d = \frac{25}{12}h\hat{u}_1^{(2)} - \frac{25}{12}h^2\hat{u}_1^{(3)} + \mathcal{O}(h^3), \quad (4.114)$$

$$\tau_2^d = -\frac{1}{24}h\hat{u}_2^{(2)} + \frac{1}{12}h^2\hat{u}_2^{(3)} + \mathcal{O}(h^3). \quad (4.115)$$

The diffusive discretization is *inconsistent* at both  $i = 1$  and  $i = 2$ .

### 4.2.3 Global truncation error

Although the discretization at the boundaries is inconsistent, it is possible to have convergence of the global error. Several techniques exist to study the relation between the local and global error, see for example [105, 74, 117]. We use the approach outlined in Wesseling [207]. Recall the relation between the global and local truncation error, equation (4.107):

$$Lv = \tau. \quad (4.116)$$

We assume that  $L$  is a monotone operator, i.e.

$$Lv \geq 0 \quad \text{implies} \quad v \geq 0. \quad (4.117)$$

Equivalently one can say that  $L$  is monotone iff  $L^{-1} \geq 0^1$ . If we can construct a function  $E$ , a so-called *barrier function*, such that<sup>2</sup>

$$LE \geq |\tau|, \quad (4.118)$$

then, subtracting this equation from (4.116), we obtain

$$L(e - E) \leq 0. \quad (4.119)$$

Since  $L$  is assumed monotone, this means that

$$e - E \leq 0 \quad \rightarrow \quad e \leq E. \quad (4.120)$$

The art is to construct a barrier function  $E$  of a certain order  $p$ ,  $E \sim h^p$ , such that (4.118) holds. Equation (4.120) then proves that the global error  $e$  is also of order  $p$ . Since any sufficiently differentiable continuous function can be expressed as a polynomial expansion, we take  $E = h^p \psi(x)$  with

$$\psi(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4. \quad (4.121)$$

It is possible to include more terms in this expansion, but this is not necessary.

Note that proving monotonicity of  $L$  for the fourth-order discretization is not trivial. Normally, monotonicity can be proven by showing that  $L$  is positive<sup>3</sup>. For second-order central discretizations positivity is proven when the mesh Péclet number  $h/\nu$  is smaller than 2. For the fourth-order scheme this approach fails, because  $L$  can never be positive, as can be observed by inspecting the signs of the diagonals of  $L$ . However, a non-positive operator can still be monotone (see for example [100, 10]). We have not yet found a sufficient condition for monotonicity of the fourth-order scheme, but it suffices to say that we have found, by explicitly computing  $L^{-1}$ , that  $L$  is monotone if  $h/\nu$  is small enough.

#### 4.2.3.1 Interior

In a general interior point  $i$  we can write for  $LE$ :

$$L(h^p \psi(x))_i = 24h^{p+1} \left( c(4a_4x_i^3 + 3a_3x_i^2 + 2a_2x_i + a_1) - \nu(12a_4x_i^2 + 6a_3x_i + 2a_2) \right). \quad (4.122)$$

We recognize the exact first and second derivatives of  $\psi(x)$ . This is because  $\tau_i^e$  and  $\tau_i^d$  only contain derivatives of fifth order and higher:  $L$  can differentiate a fourth-

<sup>1</sup>  $a \geq 0$  means that  $a_i \geq 0 \forall i$ ;  $A \geq 0$  means that  $A_{ij} \geq 0 \forall i, j$

<sup>2</sup>  $|a|$  is the grid function with value  $|a_i|$

<sup>3</sup> positivity implies monotonicity, but not vice versa

order polynomial *exactly*. Of course one can take more terms in expansion (4.121); these will not be exactly differentiated. Since  $\tau_i$  is

$$\tau_i = c \tau_i^c - v \tau_i^d = -c \frac{9}{5} h^5 \hat{u}_i^{(5)} + v \frac{9}{40} h^5 \hat{u}_i^{(6)} + \mathcal{O}(h^7), \quad (4.123)$$

we find that  $LE - |\tau|$  can be written as

$$\begin{aligned} LE_i - |\tau_i| &= 24 h^{p+1} \left( c(4a_4 x_i^3 + 3a_3 x_i^2 + 2a_2 x_i + a_1) - v(12a_4 x_i^2 + 6a_3 x_i + 2a_2) \right) \\ &\quad - h^5 \left| -c \frac{9}{5} \hat{u}_i^{(5)} + v \frac{9}{40} \hat{u}_i^{(6)} \right| + \mathcal{O}(h^7). \end{aligned} \quad (4.124)$$

For  $p \leq 4$  it is possible to construct the coefficients  $a_n$  such that  $LE - |\tau| \geq 0$ . This indicates that a fourth-order accurate local truncation error leads to a fourth-order accurate global error, as expected.

#### 4.2.3.2 Boundary, $i = 1$

The analysis is now repeated for the boundary points. At  $i = 1$  ( $x_1 = h$ ) the local error is

$$\tau_1 = c \tau_1^c - v \tau_1^d = -v \frac{25}{12} h \hat{u}_1^{(2)} - c 2h^2 \hat{u}_1^{(2)} + v \frac{25}{12} h^2 \hat{u}_1^{(3)} + \mathcal{O}(h^3), \quad (4.125)$$

so we obtain

$$L(h^p \psi(x))_1 - |\tau_1| = h^p \left( c \frac{27}{2} a_0 + v \frac{677}{24h} a_0 + \mathcal{O}(h) \right) - h \left| -v \frac{25}{12} \hat{u}_1^{(2)} + \mathcal{O}(h) \right|. \quad (4.126)$$

To make sure that the order of the first term does not exceed the order of the truncation error (for  $h \rightarrow 0$ ), we require  $p \leq 2$ . For example, in case  $p = 2$ , the leading order term condition is

$$h \left( \frac{677}{24} v a_0 - \left| -v \frac{25}{12} \hat{u}_1^{(2)} \right| \right) \geq 0. \quad (4.127)$$

This means that with a sufficiently large positive value of  $a_0$  we have a barrier function that is always larger than  $|\tau|$  for sufficiently small  $h$ . Only the first term of the polynomial expansion (4.121) is of importance here; increasing the polynomial order will not change equation (4.127).

#### 4.2.3.3 Boundary, $i = 2$

At  $i = 2$  ( $x_2 = 2h$ ), we have

$$\tau_2 = c \tau_2^c - v \tau_2^d = v \frac{1}{24} h \hat{u}_2^{(2)} - c \frac{1}{2} h^2 \hat{u}_2^{(2)} - v \frac{1}{12} h^2 \hat{u}_2^{(3)} + \mathcal{O}(h^3), \quad (4.128)$$

and we obtain

$$L(h^p \psi(x))_2 - |\tau_2| = h^p \left( -c a_0 - v \frac{13}{6h} a_0 \right) - h \left| v \frac{1}{24} \hat{u}_2^{(2)} + \mathcal{O}(h) \right|. \quad (4.129)$$

This indicates again that the maximum order of the global error is  $p = 2$  if the following condition can be fulfilled:

$$h \left( -\frac{13}{6} v a_0 - \left| v \frac{1}{24} \hat{u}_2^{(2)} + \mathcal{O}(h) \right| \right) \geq 0. \quad (4.130)$$

However, this requires  $a_0 < 0$ , which contradicts the requirement for  $i = 1$ . For the current boundary layer problem we can escape from this by noting that  $\tau_1 > 0$ ,  $\tau_2 < 0$  and  $|\tau_1| > |\tau_2|$ . This means that the absolute value signs are too restrictive; a range of positive values of  $a_0$  exists for which both (4.127) and (4.130) are satisfied.

Summarizing the results for  $i = 1$  and  $i = 2$ , we find that *no polynomial barrier function exists with which we can prove a convergence rate higher than 2*. We expect that this result holds for other barrier functions, given that any sufficiently differentiable function on the domain  $x \in [0, 1]$  can be expressed as a polynomial. Consequently, *the global error of the fourth order scheme is limited to second order due to boundary conditions*.

### 4.3 NUMERICAL EXPERIMENTS

A number of numerical experiments are carried out to support the theoretical results from sections 4.1 and 4.2. First, we study the order of accuracy of the proposed boundary conditions for a one-dimensional convection-diffusion equation (section 4.3.1) and a two-dimensional lid-driven cavity (section 4.3.2), and how fourth order accuracy can be obtained on properly designed non-uniform grids. Secondly, the energy conservation properties of the new boundary conditions are demonstrated in section 4.3.4 by considering inviscid flow in a cavity with non-zero tangential boundary conditions. Lastly, in section 4.3.5 we consider three-dimensional turbulent channel flow for which we show that the new boundary conditions improve both the velocity and pressure field near the wall, compared to the old conditions.

#### 4.3.1 1D convection-diffusion equation

##### 4.3.1.1 Uniform grid

First we verify the fourth-order scheme by using ‘exact’ boundary conditions, i.e. we substitute the exact solution (4.95) for the ghost points, like in Veldman [198]. For  $v = 1/100$  and  $h = 1/200$  ( $|h/\varepsilon| = 1/2$ ) the discrete and exact solution are shown in figure 4.4a. The global truncation error  $e = \hat{u} - u$  and local truncation error, obtained from  $\tau = Le$ , are shown in figure 4.4b. The leading error term of

the theoretical local truncation error, equation (4.123), is found by using the exact solution, equation (4.95):

$$\tau_i = \left(-\frac{9}{5} + \frac{9}{40}\right) \left(\frac{h}{\varepsilon}\right)^5 \frac{e^{x_i/\varepsilon}}{e^{1/\varepsilon} - 1}, \quad (4.131)$$

and is also shown in figure 4.4b. Because equation (4.131) contains only the leading error term, there is a small deviation between the two local truncation errors. The local truncation error always attains its maximum (in absolute sense) at  $i = 1$ , because  $e^{x/\varepsilon}$  is largest here (note  $\varepsilon < 0$ ). The global error, on the other hand, does not necessarily attain its maximum value in the first grid point: this depends on  $L^{-1}$ . The structure of  $L^{-1}$  is shown in figure 4.4c; all entries are  $\geq 0$ , so  $L$  is indeed monotone in this case. It can be seen that  $L^{-1}$  distributes the local error  $\tau$  over the entire domain in a particular way. For instance,  $e_1$  is influenced by  $\tau$  in the entire domain, whereas  $e_N$  is mainly influenced by  $\tau$  at the right side of the domain. This is because the ‘flow’ is convection-dominated and information travels from right to left ( $c < 0$ ).

A plot of  $\|e\|_\infty$  as a function of  $h$  displays clear fourth-order convergence, see figure 4.4d. The local error  $\|\tau\|_\infty$  shows fifth-order convergence, as expected from (4.131).

Now we turn to the results for scheme (4.99)-(4.104) where the boundary conditions are chosen such that symmetry properties of the discrete operators are retained. Figure 4.5a shows that the behavior of the local truncation error near the boundary is much more irregular, as expected. As predicted by equations (4.125) and (4.128), the convergence of the local error at the boundary is first order for sufficiently small  $h$ . The global error shows second-order convergence, confirming the barrier function analysis from section 4.2.3. Note that numerical experiments showed that (at  $\nu = 1/100$ )  $L$  is monotone for  $h < 1/60$ , approximately.

The ‘kink’ in the global and local error around  $h = 10^{-2}$  is due to the fact that the local truncation errors in the convective and diffusive terms almost cancel each other at  $i = 1$ . This can also be seen from figure 4.5c, which shows that the position of maximum  $\tau$  shifts from  $i = 1$  to  $i = 2$ .

Considering that both theoretical arguments and numerical experiments show that the fourth-order symmetry-preserving method is only second-order accurate for boundary layer problems with Dirichlet boundary conditions, one might wonder if it is useful to apply a fourth-order method anyway. We therefore compare the results presented above with the second-order method, which does not require boundary conditions for  $u_{-1}$  and  $u_{-2}$ . The global error behavior of the second- and fourth-order schemes is shown in figure 4.5d. It is observed that for most meshes, except the very coarse ones, the second-order method is even more accurate than the fourth-order method. The explanation lies in the fact that in this test case the largest gradients in the solution are near the boundary, where the local truncation

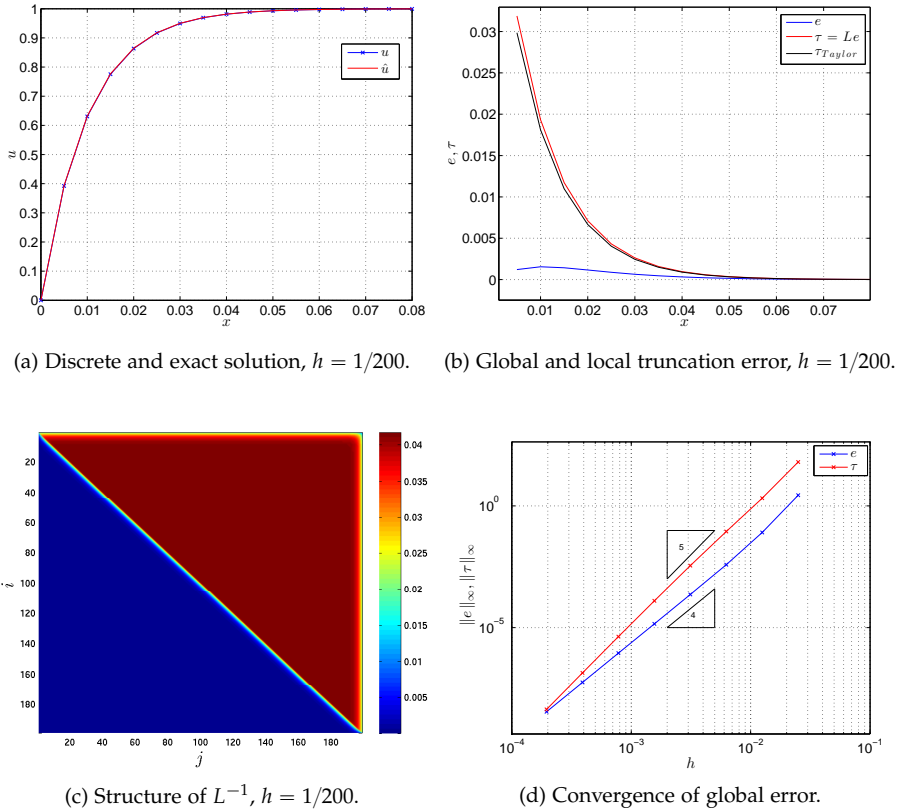


Figure 4.4: Solution and error behavior for uniform grid, exact boundary conditions,  $\nu = 1/100$ .

error has unfavorable properties. On uniform grids with boundary-layer like problems the fourth-order method is not preferred over the second-order method.

We will continue to study the effect of the boundary conditions and the difference between the second and fourth order scheme by considering non-uniform grids. Such grids are more suitable to handle boundary-layer like problems than uniform grids.

#### 4.3.1.2 Non-uniform grid

In practice, thin boundary layers should be calculated by employing non-uniform grids. In this section we investigate if the conclusions for uniform grids carry over to non-uniform grids. We investigate an exponential grid, i.e. a grid where each cell

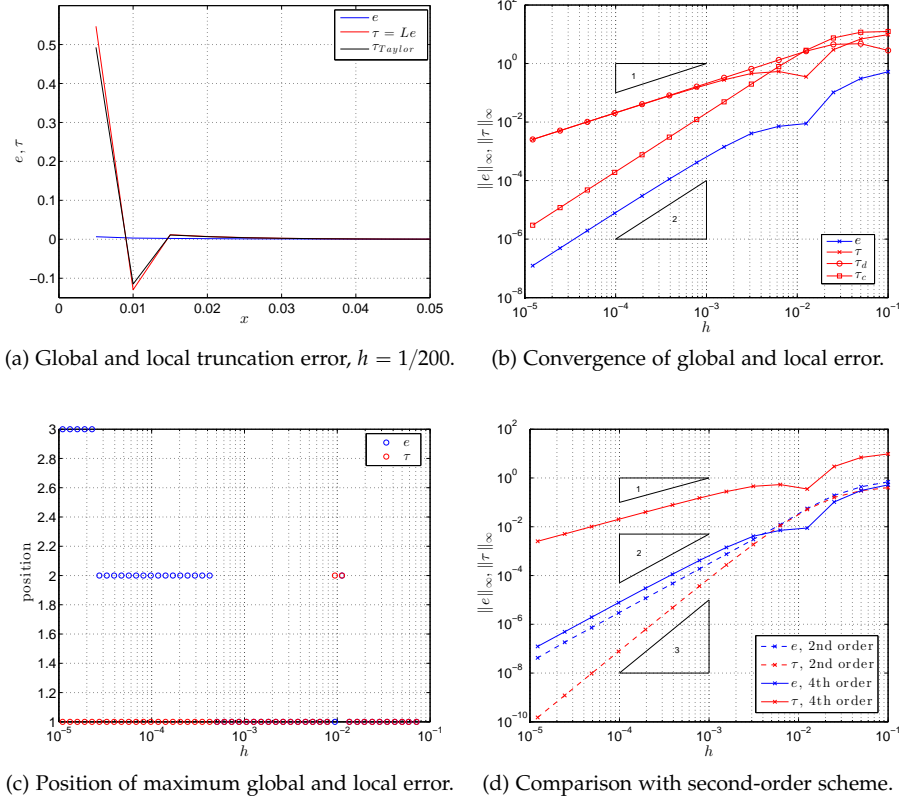


Figure 4.5: Error behavior for uniform grid, symmetry-preserving boundary conditions,  $\nu = 1/100$ .

size is a constant factor times the neighboring grid cell size. The grid is written as a mapping of a uniform grid,

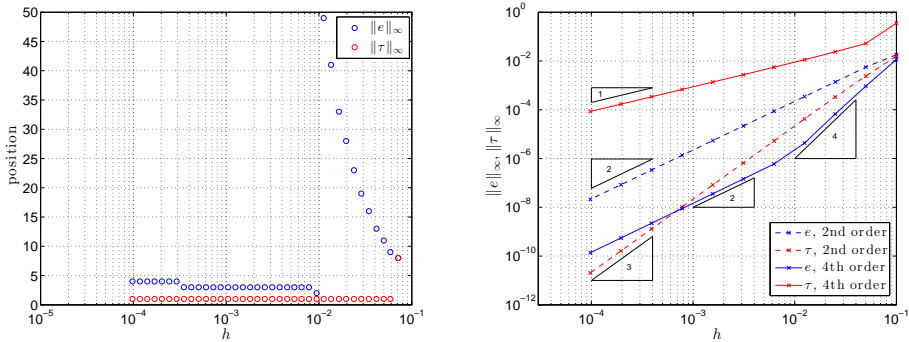
$$x(\xi) = (1 - s^{\xi})/(1 - s), \quad (4.132)$$

where  $\xi$  is uniformly distributed over  $[0, 1]$ , and  $s$  is the stretch factor. The stretch factor is determined by choosing a refinement region  $\delta$  such that in both  $[0, \delta]$  and  $[\delta, 1]$   $N/2$  volumes are located. The stretch factor stays constant during mesh refinement. Defining the boundary layer edge as the point where  $\hat{u} = p \hat{u}(1)$ , we find for sufficiently small  $\varepsilon$ :

$$\delta_p = \varepsilon \ln(1 - p). \quad (4.133)$$

For example, if the boundary layer edge is defined at the point where  $\hat{u} = 0.9$ , then  $\delta_{0.9} \approx 2.3 \cdot 10^{-2}$  ( $\varepsilon = -1/100$ ). Looking at the convergence and position of the local and global error for  $\delta_{0.9}$  (figures 4.6a and 4.6b) it can be seen that the

refinement region can make sure that the position of maximum global error is *not at the boundary*;  $u(\xi)$  does not have its largest gradient at the boundary. However, for sufficiently small  $h$  the global error moves to the boundary, and the slope changes to second order ( $h \approx 10^{-2}$ ), like in the uniform case. The position of the kink,  $h_k$ , is mainly determined by  $\delta$ : the *smaller*  $\delta$ , the *smaller* the value of  $h_k$ . If the refinement zone is chosen properly, the error will exhibit a region of fourth order convergence. For these non-uniform grids, the fourth-order scheme is more accurate than its second-order counterpart.



(a) Position of maximum global and local error,  $\delta_{0.9}$ . (b) Convergence of global error, second- and fourth-order scheme,  $\delta_{0.9}$ .

Figure 4.6: Error behavior for non-uniform grid,  $\nu = 1/100$ .

An important question that remains is: *can we design the mesh such that the global error keeps its fourth order behavior upon mesh refinement?* We concluded in section 4.2 that the local truncation error near boundaries is  $\mathcal{O}(1)$ , resulting in a global error of  $\mathcal{O}(h^2)$ . If we construct the mesh near the boundary such that  $h_{bnd} = \mathcal{O}(h^2)$  (where  $h = 1/N$ ), then one might expect heuristically a contribution to the global error of  $\mathcal{O}(h^4)$ . Such a mesh can be constructed by making the stretch factor  $h$ -dependent. The smallest finite volume is given by (equation (4.132)):

$$h_1 = \frac{1 - s^h}{1 - s} = -\frac{\ln(s)}{1 - s}h + \mathcal{O}(h^2). \tag{4.134}$$

With  $s \sim 1/h$  we obtain  $h_1 = \mathcal{O}(h^2)$ . In figure 4.7 we show the convergence of the global error for the case  $s = 10/h$ . It is clearly seen that the error now converges with  $\mathcal{O}(h^4)$ , for all mesh sizes. The fourth order method is now much more accurate than the second order method.

To summarize, the fourth order scheme attains fourth order convergence on properly chosen non-uniform meshes. These meshes should be designed such that the global error is not at the boundary but in the interior of the domain: the smallest cells near the boundary should scale with  $\mathcal{O}(h^2)$  upon mesh refinement. Furthermore, the mesh should be ‘smooth’ enough to exhibit fourth order behavior. Recall-



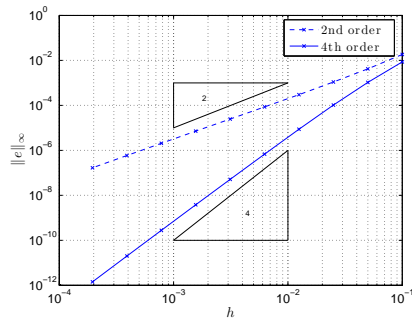


Figure 4.7: Error behavior for non-uniform grid,  $s = 10/h$ .

ing that the relation between the derivative on a uniform and non-uniform grid is given by

$$\frac{du}{dx} = \frac{du}{d\xi} \frac{d\xi}{dx}, \quad (4.135)$$

we observe that an  $\mathcal{O}(h^4)$  approximation for both  $\frac{du}{d\xi}$  and  $\frac{d\xi}{dx}$  is required. For example, Shishkin meshes (see e.g. [207, 198]), which consist of two uniform meshes with a different mesh width, lack this smoothness and fourth order accuracy is not obtained, not even when the point of maximum error is in the interior.

#### 4.3.2 Lid-driven cavity

We continue the study of the order behavior of the fourth-order scheme by studying the steady lid-driven cavity flow at a Reynolds number of 1000. In this benchmark problem [18] for the incompressible Navier-Stokes equations we introduce two major differences with the 1D convection-diffusion equation: (i) the Navier-Stokes equations are non-linear, (ii) the Navier-Stokes equations include a pressure term to satisfy the incompressibility constraint. To obtain steady solutions the full non-linear system of equations is solved with Newton linearization of the convective term. The resulting saddle-point matrix problem is solved with a direct solver. The iterative procedure is stopped when the residual (in the maximum norm) drops below  $10^{-8}$ , which requires approximately 6 iterations.

Similar to the one-dimensional boundary layer study, we have investigated both uniform and non-uniform exponential grids. The exponential grids are given by (4.132), with a stretching factor equal to the effective stretching factor of the cosine grid used by [18]:

$$s = \frac{2}{\pi h}. \quad (4.136)$$

The number of grid points ranges from  $8 \times 8$  to  $128 \times 128$ .

Figures 4.9 and 4.10 show some examples of solution profiles through the center-lines of the cavity, compared to the benchmark data of Botella and Peyret [18] (in

the figures abbreviated as BP). The pressure is determined up to a constant, which has been chosen such that  $p = 0$  in the center of the cavity. It can be seen that the pressure converges to the correct values at the boundary, and that the boundary condition for the pressure is certainly *not*  $\partial p / \partial n = 0$ . Note that the boundary conditions used by VV lead to a (numerical) boundary layer in the pressure, as shown in figures 4.11a-4.11b. The velocity profiles along the centerlines are used to perform a quantitative comparison of the second and fourth order discretization on both the uniform and exponential grid. In order to measure the order of accuracy we compare the minimum value of  $u$  along  $x = 0.5$  and maximum and minimum values of  $v$  along  $y = 0.5$  with the benchmark data. The resulting error plots are shown in figure 4.8. The non-uniform grids are better able to capture the boundary layers and therefore lead to more accurate prediction of the minimum and maximum velocity values along the centerline. Comparing the second and fourth order results on the uniform grid, figure 4.8a, we conclude that the fourth order method has a somewhat smaller error than the second order method, but converges only with second order upon grid refinement. This is in agreement with the error analysis from section 4.2 and the 1D results from section 4.3.1 - in particular figure 4.5d.

On non-uniform grids we expect better convergence properties since the exponential grid satisfies  $h_1 = \mathcal{O}(h^2)$ . Indeed, fourth order convergence of  $u_{max}$  and  $v_{max}$  is shown in figure 4.8b. The convergence order of  $v_{min}$  is apparently lower; this can possibly be improved by adapting (4.136). Since this is problem specific we do not investigate this in detail here.

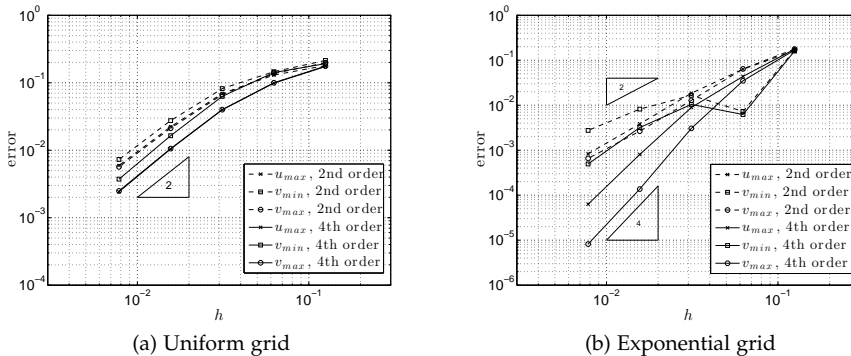


Figure 4.8: Errors in velocity profiles.

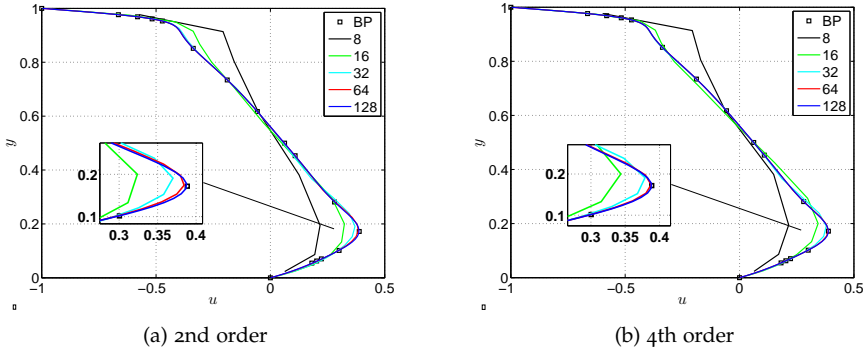


Figure 4.9: Velocity profiles  $u(y)$  at  $x = 0.5$ , 2nd and 4th order method on an exponential grid.

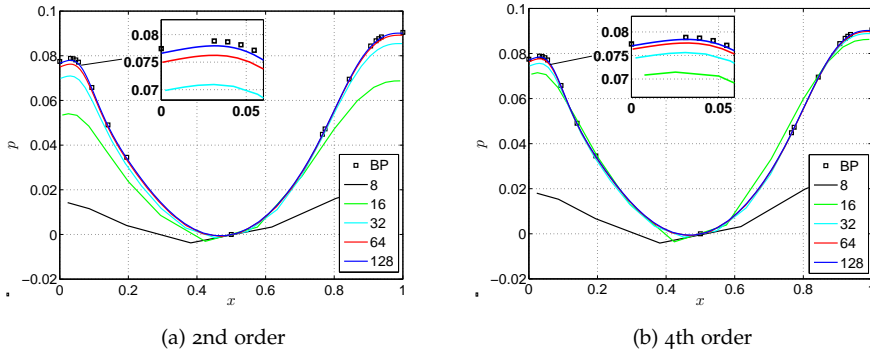


Figure 4.10: Pressure profiles  $p(x)$  at  $y = 0.5$ , 2nd and 4th order method on an exponential grid.

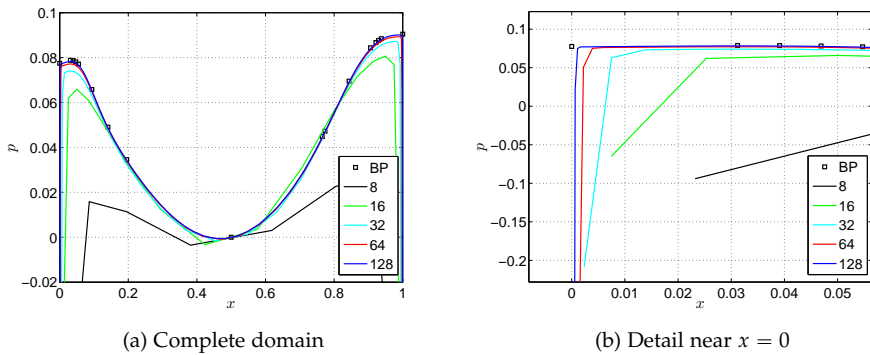


Figure 4.11: Pressure profiles  $p(x)$  at  $y = 0.5$  with 4th order method and old boundary conditions. Compare with figure 4.10b.

## 4.3.3 Taylor-Green vortex

We repeat the Taylor-Green test case from section 3.3.1, but now with time-dependent Dirichlet conditions. Figure 4.12 shows the spatial error in the velocity as function of mesh size for both the second and fourth order method. The second order method converges with second order accuracy, as for the case of periodic boundary conditions, but the convergence of the fourth order method is also only second order. This confirms the results from sections 4.3.1 and 4.3.2: the accuracy of the fourth order method drops to second order on uniform meshes when boundaries are present.

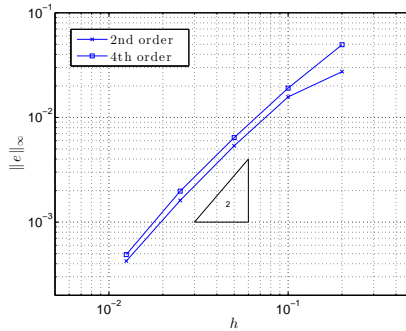


Figure 4.12: Convergence of spatial error with unsteady Dirichlet boundary conditions.

## 4.3.4 Energy conservation in an inviscid cavity

In this section we will show that the newly proposed boundary conditions for the fourth order method exactly conserve energy in case of no-penetration conditions. Desjardins et al. [42] considered a two-dimensional inviscid channel flow with periodic boundary conditions in one direction and no-slip conditions in the other. We increase the difficulty of their test case by prescribing non-zero tangential conditions on *all* boundaries, resulting in an ‘inviscid cavity’:

$$u(x, 0, t) = -1, \quad u(x, 1, t) = 16x^2(1-x)^2, \quad (4.137)$$

$$v(0, y, t) = 1, \quad v(1, y, t) = -1. \quad (4.138)$$

See figure 4.13a. The velocity field is discontinuous in the corners. Furthermore, in contrast to [42], we use the implicit midpoint method for time integration instead of the Crank-Nicolson method, since the latter is not truly energy-conserving (this will be discussed in chapter 7).

The non-uniform grid that we have used consists of  $20 \times 20$  volumes and is shown in figure 4.13b. In  $x$ -direction the stretch factor (ratio between largest and smallest volume) is 2, in  $y$ -direction it is 10. The initial velocity field consists of uniform

random numbers between  $-1$  and  $1$  and is made divergence-free by a projection step involving the solution of a Poisson equation. This velocity field is subsequently advanced in time from  $t = 0$  to  $t = 1$  with  $\Delta t = 1/100$ . In figure 4.14 we show the relative error in kinetic energy,

$$e_k = \frac{K(t) - K(t=0)}{K(t=0)}, \quad (4.139)$$

for both the new and old boundary condition treatment. With the new boundary conditions the energy error stays at machine precision ( $\mathcal{O}(10^{-14})$ ), whereas the energy grows in time with the old boundary conditions.

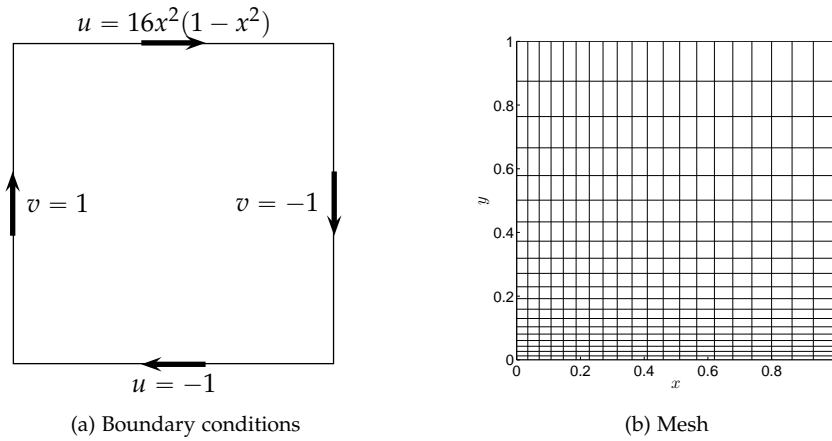


Figure 4.13: Inviscid cavity.

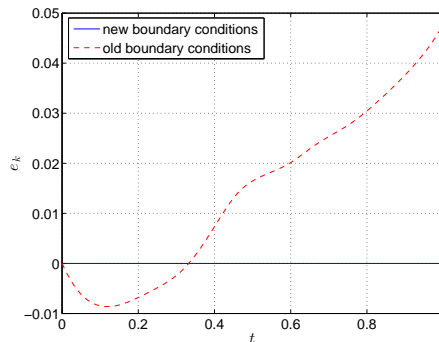


Figure 4.14: Energy evolution in inviscid cavity as a function of time.

4.3.5 *Turbulent channel flow*

In this section<sup>4</sup>, the two discrete boundary conditions given by (4.79) and (4.80) are compared for a turbulent channel flow. This flow can be seen as a prototype for near-wall turbulence. The Reynolds number in our simulation is based on the channel width and the bulk velocity. It is set equal to  $Re = 22,000$ ; in terms of the channel half width and the friction velocity the Reynolds number becomes approximately  $Re_\tau = 590$ . Moser et al. [118] have performed a direct numerical simulation (DNS) of this channel flow. Their DNS data is often used to evaluate the accuracy of large eddy simulations (LES) of near-wall turbulence. The DNS was done on a  $384 \times 257 \times 384$  grid; large eddy simulations typically use about  $64^3$  points, i.e., they have about 144 times less points. The present test considers an LES with a subgrid model that is based on the two non-zero invariants of the rate of strain tensor, see [201]. As usual the flow is assumed to be periodic in the stream- and spanwise direction. Consequently, the computational domain may be confined to a channel unit. The dimension of the considered unit is taken identical to that of the DNS. The LES uses 64 uniformly distributed streamwise points and 32 (uniformly distributed) spanwise points. In the lower half of the channel, the wall-normal grid points are computed according to

$$y_j = \frac{\sinh(\gamma j / N_y)}{2 \sinh(\gamma / 2)} \quad \text{with } j = 0, 1, \dots, N_y / 2, \quad (4.140)$$

where  $N_y$  denotes the number of grid points in the wall-normal direction. The stretching parameter  $\gamma$  is taken equal to 8. The grid points in the upper half are computed by means of symmetry. The time step is set equal to  $\Delta t = 10^{-3}$  (non-dimensionalized by the bulk velocity and the channel width). Mean values of the velocity are computed by averaging the instantaneous velocity over the directions of periodicity, the two symmetrical halves of the channel, and over time. The averaging over time starts after a start-up period of 90,000 time steps. The averaging is performed over 90,000 time steps too. Figure 4.15 shows a comparison of the two mean velocity profiles as obtained from our LES with the fourth-order symmetry-preserving discretization method, where we have applied the boundary treatment given by equations (4.80) as well as that given by equations (4.79). The difference between the resulting wall shear stresses is about 10%. Equations (4.80) result into  $\tau_w = 2.9 \cdot 10^{-3}$ , which is in excellent agreement with the DNS of Moser et al., whereas equations (4.79) lead to  $\tau_w = 3.2 \cdot 10^{-3}$ , which is too high. All results shown in figure 4.15 are scaled with the help of  $\tau_w = 2.9 \cdot 10^{-3}$ . As can be seen, the main differences between the two wall treatments are located in the region near the wall. The differences are very small away from the wall (for  $y^+ > 20$ ), whereas near the wall substantial differences, up to 10%, can be observed.

Figure 4.16 illustrates the differences between the resulting pressures. Away from the boundary ( $y^+ = 0$ ) the two pressures profiles differ by a constant (approxim-

<sup>4</sup> This section has been contributed by R. Verstappen (University of Groningen).

ately). Such a difference by a constant is not relevant, since the pressure gradients are (almost) the same. Near the boundary ( $0 < y^+ < 20$ ), on the other hand, the pressure profiles are different in the first few (4) grid points counted from the wall. The old boundary treatment described by equations (4.79) results into a thin - numerical - boundary layer: the pressure is solved from a Poisson equation that is effectively supplied with Dirichlet conditions. The new boundary treatment, which effectively imposes a Neumann condition, leads to the proper boundary behavior of the pressure near the wall.

In conclusion, the boundary treatment given by (4.80) yields better results for both velocity and pressure in the boundary layer of a turbulent channel flow.

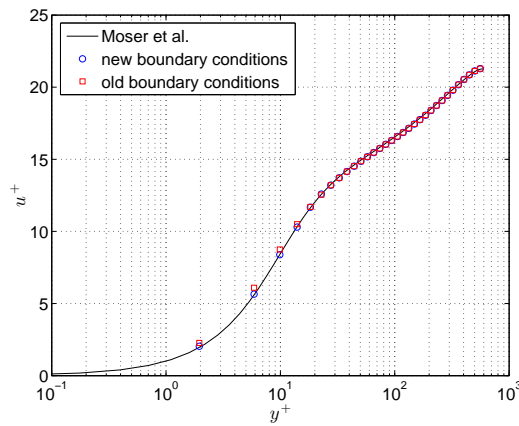
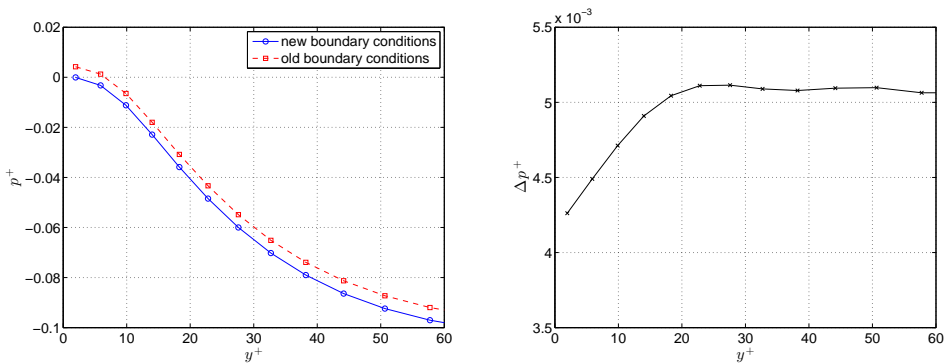


Figure 4.15: Comparison of the mean streamwise velocity  $u^+$  as a function of  $y^+$ .



(a) Mean pressure  $p^+$  as a function of  $y^+$

(b) The difference of the two pressures shown in figure 4.16a

Figure 4.16: Comparison of pressure profiles.

#### 4.4 CONCLUSIONS

Symmetry-preserving or energy-conserving discretization methods have become popular methods for simulating turbulent flows governed by the incompressible Navier-Stokes equations. In this chapter we have proposed a new consistent boundary treatment for the symmetry-preserving discretization method described in [203]. The new boundary conditions are such that continuous integral identities (expressing conservation of mass, momentum and kinetic energy) are mimicked in a discrete sense, including the contributions from the boundaries. To derive this boundary treatment, we have used a new technique in which the ‘implied’ discretization of boundary volumes is derived based on global conservation of momentum. Upon requiring conservation of energy it turns out that the ghost values of the normal velocity component at a boundary should be set according to a usual Dirichlet condition, whereas the ghost values of the tangential velocity component at a boundary should be set according to a Neumann type condition. With these boundary conditions the convective operator is also energy-conserving if non-zero tangential boundary conditions are prescribed. The newly derived boundary conditions furthermore imply the correct boundary conditions for the pressure.

The use of these conditions limits the global order of accuracy of the fourth order discretization method to second order on uniform grids, because the requirements which (skew-)symmetry pose on the coefficients of the matrices are not compatible with the requirements resulting from Taylor expansions to reach higher order accuracy. We have shown that by using properly refined non-uniform grids near the boundary, high-order accuracy of the global error can still be obtained.

The improved boundary conditions are important for wind-turbine wake simulations. No-slip boundary conditions appear at the Earth’s surface, leading to an atmospheric boundary layer in which wind turbines operate. Combining the new boundary conditions with proper grid refinement makes the fourth order method ready to be used for wind-turbine wake aerodynamics.



## Part II

### TEMPORAL DISCRETIZATION

This part describes the application of Runge-Kutta methods for time integration of the semi-discrete incompressible Navier-Stokes equations. Firstly, new *explicit* methods that lead to higher-order temporally accurate velocity and pressure approximations are proposed. Secondly, *implicit* methods that are energy-conserving in the inviscid limit are developed.



## RUNGE-KUTTA METHODS FOR THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

---

### 5.1 INTRODUCTION

In this part of the thesis we discuss the application of Runge-Kutta methods to the time discretization of the incompressible Navier-Stokes equations. Runge-Kutta methods can combine good stability properties with high order, do not have a start-up problem, and easily allow for adaptive time stepping. The main reason for studying Runge-Kutta methods in this thesis is because they can be constructed such that they conserve energy, and as such extend the energy-conserving property of the spatial discretization when marching in time. However, as we will see, such methods are necessarily implicit and therefore relatively expensive. As an alternative, we will therefore also consider explicit Runge-Kutta methods.

The application of (both explicit and implicit) Runge-Kutta methods to the incompressible Navier-Stokes equations is not straightforward because of the differential-algebraic nature of the equations, caused by the divergence-free constraint. It is common practice to advance the velocity at each stage as if the discretized equations are a system of ordinary differential equations, and subsequently solve a Poisson equation for the pressure to make the velocity field divergence-free. However, it is not clear if and how this approach influences the temporal order of accuracy of the velocity and pressure. The accuracy of the velocity is often silently assumed to be unaffected by the differential-algebraic nature of the incompressible Navier-Stokes equations, and the temporal accuracy of the pressure is often not reported. A temporally accurate pressure is however of interest in many flow simulations, such as those involving unsteady lift and drag computations (forces on wind-turbine blades), or fluid-structure interactions. We will analyze the accuracy of both velocity and pressure by applying the convergence theory developed for index 2 differential-algebraic equations [60, 63] to the incompressible Navier-Stokes equations.

The outline of this part of the thesis is as follows. First, in section 5.2 we consider the incompressible Navier-Stokes equations as a system of differential-algebraic equations, for which we write down the Runge-Kutta method in section 5.3. In chapter 6 we focus on explicit Runge-Kutta methods, and in chapter 7 on implicit (energy-conserving) Runge-Kutta methods.

### 5.2 DIFFERENTIAL-ALGEBRAIC EQUATIONS

The semi-discrete equations (3.1)-(3.2) obtained after spatial discretization form a non-autonomous differential-algebraic equation (DAE) system of index 2 (see e.g.

[60, 63, 8]), where  $u$  plays the role of the differential variable, and  $p$  the role of the algebraic variable. Following the literature on DAEs, they can be written as

$$0 = g(u, t), \quad (5.1)$$

$$\dot{u} = f(u, p, t), \quad (5.2)$$

where

$$g(u, t) = Mu - r_1(t), \quad (5.3)$$

$$f(u, p, t) = F(u, t) - Gp, \quad (5.4)$$

with  $F(u, t) = -C(u) + \nu Du + r_2(u, t)$ .  $\Omega^{-1}$  has been absorbed in the definition of  $C$ ,  $D$ ,  $G$  and  $r_2$ . The explicit presence of unsteady boundary conditions for the divergence equation,  $r_1(t)$ , is often omitted in literature, but it will be shown to be important when discussing explicit Runge-Kutta methods. An example of a nonzero  $r_1(t)$  is a time-varying inflow condition such as a turbulent inflow field. We assume that

$$L = -g_u(u, t)f_p(u, p, t) \quad \text{is non-singular,} \quad (5.5)$$

so that the problem is indeed of index 2 [63, 8]. For the incompressible Navier-Stokes equations

$$L = MG \quad (5.6)$$

is recognized as the Laplacian operator (independent of  $u$  and  $p$ ), which is actually singular in case of Dirichlet or periodic conditions for the velocity on the entire boundary. A possible remedy against the singular nature is to impose an additional constraint (e.g. setting the average pressure value by replacing one row of the Laplace matrix by ones). An instantaneous equation for the algebraic variable, the pressure, is found by differentiating the divergence-free constraint in time and substituting the momentum equation:

$$Lp = MF(u, t) - \dot{r}_1(t), \quad (5.7)$$

where we have used that  $M$  is not depending on  $t$ . Upon differentiation in time once more one obtains an ODE for the pressure; the index of the DAE is therefore 2 (the index is the number of differentiations necessary to obtain an ODE). We note that this equation can be derived similarly if  $M$ ,  $G$  and other operators depend on time, as in the case of time-varying meshes; an additional term  $M(t)u$  will appear on the right-hand side. The pressure can be eliminated from the system of equations, by solving equation (5.7) and inserting it into (5.2):

$$\dot{u} = PF(u, t) + GL^{-1}\dot{r}_1(t), \quad (5.8)$$

where the projection operator  $P$ , defined by

$$P = I - GL^{-1}M, \quad (5.9)$$

is a square matrix that projects velocity fields on the space of divergence-free fields; the divergence of this projection is zero ( $MP = 0$ ). However, when discretizing in time one should *not* start with equation (5.8). The differentiation of the constraint, necessary to arrive at (5.8), has lowered the index of the DAE system, and upon discretization its solutions do not necessarily satisfy the constraint (5.1). It is therefore desirable to use the original DAE system (5.1)-(5.2) (the one with highest index), because its solutions will satisfy all the derived lower index systems [59].

The initial conditions at  $t = t_0$  should be consistent with equations (5.1)-(5.2) and (5.7):

$$Mu_0 = r_1(t_0), \quad (5.10)$$

$$Lp_0 = MF(u_0, t_0) - \dot{r}_1(t_0). \quad (5.11)$$

Equation (5.11) expresses that the initial pressure cannot be chosen freely, but has to be calculated based on  $u_0$ .

## 5.3 RUNGE-KUTTA METHODS

### 5.3.1 General formulation

The solution to the semi-discrete equations will be approximated by the fully discrete solution  $u_n, p_n$  by means of a Runge-Kutta method. A general Runge-Kutta method applied to the index 2 DAE (5.1)-(5.2) reads [63, 149]:

$$U_i = u_n + \Delta t \sum_{j=1}^s a_{ij}(F_j - G\psi_j), \quad MU_i = r_1(t_i), \quad (5.12)$$

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^s b_i(F_i - G\psi_{n+1}), \quad Mu_{n+1} = r_1(t_{n+1}), \quad (5.13)$$

with  $t_i = t_n + c_i\Delta t$  and  $s$  is the number of stages. Here  $U_i$  and  $u_n$  are approximations to the exact values  $u(t_i)$  and  $u(t_n)$ , respectively, and  $F_j = F(U_j, t_j)$ . The pressure-like variable  $\psi$  is introduced to explicitly distinguish it from  $p$ , since  $\psi$  generally does not satisfy (5.7).

The 'classical' order of the method (when applied to non-stiff ODEs) is called  $p$  (from the context confusion with the pressure will be avoided). The set of coeffi-

icients  $a_{ij}$ ,  $b_i$  and  $c_i$  will be denoted collectively by  $\mathcal{A}$  and can be written in compact form using a Butcher tableau:

$$\begin{array}{c|cccc}
 c_1 & a_{11} & a_{12} & \dots & a_{1s} \\
 c_2 & a_{21} & a_{22} & \dots & \vdots \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 c_s & a_{s1} & \dots & \dots & a_{ss} \\
 \hline
 & b_1 & \dots & \dots & b_s
 \end{array} \tag{5.14}$$

with the convention

$$c_i = \sum_{j=1}^s a_{ij}, \tag{5.15}$$

not to be confused with the convecting velocity in (3.2). The following so-called ‘simplifying conditions’ will be used in studying order conditions for Runge-Kutta methods [24]:

$$B(p) : \quad \sum_{i=1}^s b_i c_i^{q-1} = \frac{1}{q}, \quad q = 1, \dots, p; \tag{5.16}$$

$$C(\eta) : \quad \sum_{j=1}^s a_{ij} c_j^{q-1} = \frac{c_i^q}{q}, \quad i = 1, \dots, s, \quad q = 1, \dots, \eta; \tag{5.17}$$

$$D(\zeta) : \quad \sum_{i=1}^s b_i c_i^{q-1} a_{ij} = \frac{b_j}{q} (1 - c_j^q), \quad j = 1, \dots, s, \quad q = 1, \dots, \zeta. \tag{5.18}$$

The first condition relates to the fact that the last step of the Runge-Kutta method can be seen as a quadrature method with abscissas  $c_i$  and weights  $b_i$ ;  $B(p)$  indicates that the quadrature method is exact for polynomials of degree  $\leq p - 1$ . Similarly, the second condition  $C(\eta)$  indicates the minimum order of quadrature of all the stages of the Runge-Kutta method; sometimes we will use  $C_i(\eta)$  to indicate the quadrature order of stage  $i$ . If  $C_i(\eta)$  holds then polynomials of degree lower than  $q$  are exactly interpolated at stage  $i$ .  $C_i(1)$  is equivalent to condition (5.15).

### 5.3.2 *Playing with the pressure*

Now that the Runge-Kutta method has been applied, one can obtain a more compact formulation by eliminating the pressure:

$$U_i = u_n + \Delta t \sum_{j=1}^s a_{ij} P F_j + GL^{-1}(r_1(t_i) - r_1(t_n)), \quad i = 1, \dots, s. \tag{5.19}$$

Since  $MP = 0$ , equation (5.19) satisfies  $MU_i = r_1(t_i)$  at all intermediate stages. When first eliminating the pressure, and then applying the Runge-Kutta method, the last term in (5.19) would change to  $GL^{-1} \sum_j a_{ij} \dot{r}_1(t_j)$ , and the constraint is only satisfied if  $\dot{r}_1(t) = 0$  (so both formulations are equal in the case of steady boundary conditions). The velocity at the new time step follows as:

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^s b_i P F_i + GL^{-1}(r_1(t_{n+1}) - r_1(t_n)). \quad (5.20)$$

The presence of the inverse of the Laplace operator (in the two last terms) makes the computation of  $U_i$  unattractive from a practical point of view. Therefore we rewrite equation (5.19) back into a two-step formulation by introducing a variable which 'looks' like the pressure  $p$ . First we substitute  $P = I - GL^{-1}M$  in (5.19), leading to

$$U_i = u_n + \Delta t \sum_{j=1}^i a_{ij} F_j - GL^{-1} \left( \Delta t \sum_{j=1}^i a_{ij} M F_j - (r_1(t_i) - r_1(t_n)) \right). \quad (5.21)$$

Comparing with the 'exact' equation for the pressure at each stage,

$$Lp_i = MF_i - \dot{r}_1(t_i), \quad (5.22)$$

it seems natural to introduce a pressure-like variable,  $\phi$ , and the  $c$  coefficients and rewrite (5.21) as the following two steps:

$$U_i = u_n + \Delta t \sum_{j=1}^i a_{ij} F_j - c_i \Delta t G \phi_i, \quad (5.23)$$

with  $\phi_i$  defined by

$$L\phi_i = \sum_{j=1}^i \frac{1}{c_i} a_{ij} M F_j - \frac{r_1(t_i) - r_1(t_n)}{c_i \Delta t}. \quad (5.24)$$

Equation (5.24) is simply the divergence of (5.23) supplemented with the additional information  $MU_i = r_1(t_i)$ . Each  $\phi_i$  is a Lagrange multiplier to make  $U_i$  divergence free and each  $U_i$  is independent of the value of  $\phi_j$  for  $j \neq i$ ; this is the advantage of using  $\phi$  instead of  $\psi$ . It should be stressed that the presence of the  $c$  coefficients in the pressure term is *not* necessary to obtain the correct velocity field. The reason to introduce the  $c$  coefficients is that for explicit methods it yields a  $\phi_i$  which is a consistent approximation to  $\tilde{p}_i$ . In case of implicit methods (e.g. methods based on Radau or Lobatto quadrature) it is possible to have a non-trivial first stage ( $a_{1j} \neq 0$ ) with  $c_1 = 0$  and then the  $c$ -coefficients should *not* be introduced. The most general approach is then to define a pressure-like variable  $\rho_i (= c_i \phi_i)$  and not introduce the  $c$  coefficients at all. As will be seen in sections 6.3 and 7.6, this does not complicate

obtaining high-order accurate pressure estimates, because the combination  $c_i\phi_i = \rho_i$  also occurs in that context. These  $\rho_i$  values are still Lagrange multipliers whose value is independent of previous time steps, and consequently the velocity field remains independent of the history of the pressure. On the other hand, this is no longer true when the solution of velocity and pressure is decoupled, for instance by employing a pressure correction method, see e.g. [194, 5, 87]. In that case the  $c$  coefficients appear in the pressure term of the initial velocity estimate and both velocity and pressure depend on pressure values from previous time steps.

When comparing equation (5.24) to equation (5.22), the first term on the right side of (5.24) is recognized as an approximation to  $MF_i$  and the second term as an approximation to  $\dot{r}_1(t_i)$ . The second term is clearly a first-order approximation, since

$$\dot{r}_1(t_i) = \frac{r_1(t_i) - r_1(t_n)}{c_i\Delta t} + \mathcal{O}(\Delta t). \quad (5.25)$$

The first term is also a first-order approximation, which we show by following an argument employed in [129, 130].  $F_i$  in (5.22) is  $F$  evaluated at  $(U_i, t_i)$ , whereas  $\sum_{j=1}^i \frac{1}{c_i} a_{ij} F_j$  is an approximation to the *average value* of  $F$  from  $t_n$  to  $t_i$ . Assuming that  $F$  is continuous over the interval  $[t_n, t_i]$ , this average equals the value of  $F$  at some point  $\hat{t} \in [t_n, t_i]$  (according to the integral version of the mean value theorem) and as such is an  $\mathcal{O}(\Delta t)$  approximation to  $F_i$ :

$$MF_i = \sum_{j=1}^i \frac{1}{c_i} a_{ij} MF_j + \mathcal{O}(\Delta t). \quad (5.26)$$

As a consequence the Lagrange multiplier  $\phi$  is a first-order approximation to the pressure  $p$ :

$$\phi_i = p_i + \mathcal{O}(\Delta t). \quad (5.27)$$

Of course, by virtue of the midpoint method,  $\phi_i$  is a second-order approximation to the pressure at  $t_n + \frac{1}{2}c_i\Delta t$ , as long as the stage order of the method is at least 2 (this will be detailed in section 6.3). We will call the approach, where one uses  $\phi_s$  as approximation to  $p_{n+1}$ , the ‘standard’ approach. The first-order accuracy of this approach is independent of the particular coefficients of the Runge-Kutta method. It results from the fact that (5.12) contains an approximation to the integral  $\int F dt$ , whereas (5.24) contains  $F$  evaluated at a certain time instance. This is because the pressure has an *instantaneous* character: its value is such that the velocity field is divergence free at each time instant, and is independent of the pressure at any previous time. The equation for the velocity is, on the contrary, an evolution equation.

An alternative formulation of equations (5.23) and (5.24) that can avoid the first-order behavior of the pressure is as follows. Instead of taking a single pressure-like



variable at each stage, we take a combination by introducing a Butcher tableau  $A^p$  for the pressure term such that:

$$U_i = u_n + \Delta t \sum_{j=1}^i a_{ij} F_j - \Delta t \sum_{j=1}^i a_{ij}^p G \psi_j, \quad (5.28)$$

with

$$\sum_{j=1}^i a_{ij}^p L \psi_j = \sum_{j=1}^i a_{ij} M F_j - \frac{r_1(t_i) - r_1(t_n)}{\Delta t}. \quad (5.29)$$

$A^p$  is, like  $A$ , lower triangular and should have  $a_{i,i}^p \neq 0$  in order to guarantee a unique  $\psi$  at each stage. Equations (5.24) and (5.29) are related by

$$\psi = (A^p)^{-1} \text{diag}(c_1, \dots, c_s) \phi. \quad (5.30)$$

One can therefore obtain  $\psi$  from  $\phi$ , as long as  $A^p$  is invertible. For the choice

$$A^p = \text{diag}(c_1, \dots, c_s), \quad (5.31)$$

we obtain formulation (5.23)-(5.24) with  $\psi = \phi$ , and the accuracy of the pressure is limited to first order. A possible way to obtain a higher order pressure is to choose  $A^p = A$  (so that there is just a single Butcher tableau). This is simply the original formulation (5.12)-(5.19), of which the accuracy can be evaluated with the theory of Hairer et al. [60, 63].



*Because the treatment of incompressible flow is so unforgiving of imprecise ideas, such flows still remain a fertile ground. [59]*

*adapted  
from [149]*

In this chapter we discuss explicit Runge-Kutta methods. In section 6.1 we rewrite the general Runge-Kutta method in terms of the shifted Butcher tableau. Subsequently, in section 6.2 we investigate the order conditions for velocity and pressure, and in section 6.3 we propose different methods to compute a high-order accurate pressure. In section 6.4 we show the results of two test cases which confirm our theoretical findings.

## 6.1 INTRODUCTION

A simpler representation for explicit methods is obtained by introducing the shifted matrix  $\tilde{A}$  and vector  $\tilde{c}$ :

$$\tilde{A} = \begin{pmatrix} a_{21} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{s1} & \dots & a_{s,s-1} & 0 \\ b_1 & \dots & b_{s-1} & b_s \end{pmatrix}, \quad \tilde{c} = \begin{pmatrix} \tilde{c}_1 \\ \vdots \\ \tilde{c}_{s-1} \\ \tilde{c}_s \end{pmatrix} = \begin{pmatrix} c_2 \\ \vdots \\ c_s \\ 1 \end{pmatrix}, \quad (6.1)$$

and the shifted vectors

$$\tilde{U} = \begin{pmatrix} \tilde{U}_1 \\ \vdots \\ \tilde{U}_{s-1} \\ \tilde{U}_s \end{pmatrix} = \begin{pmatrix} U_2 \\ \vdots \\ U_s \\ u_{n+1} \end{pmatrix}, \quad \tilde{p} = \begin{pmatrix} \tilde{p}_1 \\ \vdots \\ \tilde{p}_{s-1} \\ \tilde{p}_s \end{pmatrix} = \begin{pmatrix} p_2 \\ \vdots \\ p_s \\ p_{n+1} \end{pmatrix}, \quad (6.2)$$

so that equations (5.12) and (5.13) can be written as

$$\tilde{U}_i = u_n + \Delta t \sum_{j=1}^i \tilde{a}_{ij} P F_j + G L^{-1} (r_1(\tilde{t}_i) - r_1(t_n)), \quad (6.3)$$

with  $\tilde{t}_i = t_n + \Delta t \tilde{c}_i$ .

The solution algorithm that we use in practice is obtained by writing (6.3) as:

$$\tilde{V}_i = u_n + \Delta t \sum_{j=1}^i \tilde{a}_{ij} F_j, \quad (6.4)$$

$$L\tilde{\phi}_i = \frac{1}{\tilde{c}_i \Delta t} (M\tilde{V}_i - r_1(\tilde{t}_i)), \quad (6.5)$$

$$\tilde{U}_i = \tilde{V}_i - \tilde{c}_i \Delta t G \tilde{\phi}_i, \quad i = 1, 2, \dots, s, \quad (6.6)$$

optionally followed by equation (5.30). This sequence of first computing a tentative velocity, then the pressure, and finally correcting the tentative velocity is similar to fractional step methods (see e.g. [129]). However, in fractional step methods the diffusive and/or convective terms are taken implicitly, and a splitting error results from uncoupling the solution of velocity and pressure. Here all terms are handled explicitly (except the pressure) and consequently there is *no* splitting error involved. It is therefore unnecessary to solve a coupled system for  $\tilde{U}_i$  and  $\tilde{\phi}_i$ , as is done for example in Pereira et al. [125]. Hairer et al. [60] call these methods *half-explicit*: the differential variable is advanced with an explicit method (equations (6.4) and (6.6)) while the algebraic variable is handled implicitly (equation (6.5)). The implicit equation for the pressure has to be solved at each stage, resulting in  $s$  Poisson equations. The resulting  $\tilde{U}_s = u_{n+1}$  and  $\tilde{\phi}_s = \phi_{n+1}$  (or  $\tilde{\psi}_s$ ) are approximations to  $u(t_{n+1})$  and  $p(t_{n+1})$ . The order of accuracy of  $\tilde{\psi}_s$  and  $\tilde{U}_s$  will be considered next.

## 6.2 ORDER CONDITIONS

### 6.2.1 Local and global error

For general index 2 DAEs of the form (5.1)-(5.2) the classical order conditions ('classical' referring to non-stiff ODEs, see e.g. [25]) for the coefficients of the Butcher tableau are not sufficient to guarantee the correct order of accuracy for both the differential and algebraic variable. The work of Hairer et al. [60] and Hairer and Wanner [63] provides local and global error analyses for index 2 DAEs and identifies in which cases *order reduction* can occur. We focus on the local error, because for half-explicit methods the error propagation from local to global error is the same as for non-stiff ODEs. For the velocity (the differential variable) this is expressed by the following theorem:

**Theorem 6.2.1.** *Convergence - Brasey and Hairer [20]. Suppose that (5.5) holds in a neighborhood of the solution  $(u(t), p(t))$  of equations (5.1)-(5.2) and that the initial values satisfy (5.10)-(5.11). If the coefficients of the half-explicit Runge-Kutta method (6.4)-(6.6) satisfy  $\tilde{a}_{i,i}^p \neq 0$  and if the local error satisfies*

$$\delta u(t) = \mathcal{O}(\Delta t^{r+1}), \quad (6.7)$$

then the method is convergent of order  $r$ , i.e.,

$$u_n - u(t_n) = \mathcal{O}(\Delta t^r) \quad \text{for } t_n - t_0 = n\Delta t \leq T, \quad (6.8)$$

with  $T$  finite.

The pressure-like variable  $\tilde{\psi}_i$  (including  $\tilde{\psi}_s = \psi_{n+1}$ ) is independent of  $\psi_n$  and the order of accuracy of the global error in  $\psi$  is therefore given by the order of accuracy of the local error  $\delta\psi(t)$ , provided that  $\delta u(t)$  has at least the same order [60]. The focus of the rest of this section is therefore on the local error of both the  $u$ - and  $p$ -component.

### 6.2.2 A short introduction to trees

For Runge-Kutta methods applied to ODEs of the form  $\dot{u} = f(u)$ , the local error can be investigated by expanding both the exact and numerical solution in a Taylor series and comparing until which order they agree. This requires that  $\ddot{u}$ ,  $\ddot{\psi}$ , etc. are written in terms of  $f$  and its derivatives:

$$\dot{u} = f, \quad \ddot{u} = f_u f, \quad \ddot{\psi} = f_u f_u f + f_{uu}(f, f). \quad (6.9)$$

Since  $f$  and  $u$  are vectors, the first derivatives in this expression should be interpreted as Jacobian matrices, the second derivatives as bilinear maps, and  $(f, f)$  as a tensor product. The number of elementary differentials that appear in this process grows rapidly when high orders are compared. With each differential there is an associated order condition. An efficient way to handle the order conditions for ODEs was introduced by Butcher with the concept of rooted trees [25, 23]. Given a certain tree the elementary differential and the order condition corresponding to it can be easily written down. For example, (6.9) becomes in terms of trees

$$\dot{u} = \bullet \quad \ddot{u} = \begin{array}{c} \bullet \\ | \\ \bullet \end{array} \quad \ddot{\psi} = \begin{array}{c} \bullet \\ | \\ \bullet \\ | \\ \bullet \end{array} + \begin{array}{c} \bullet \quad \bullet \\ \diagdown \quad / \\ \bullet \end{array} \quad (6.10)$$

The order conditions for these trees are (in order of appearance):

$$\sum b_i = 1, \quad \sum b_i c_i = \frac{1}{2}, \quad \sum b_i a_{ij} c_j = \frac{1}{6}, \quad \sum b_i c_i^2 = \frac{1}{3}. \quad (6.11)$$

In all cases, the summation is over all indices present in the summand.

The extension of the analysis with trees to DAEs was done by Hairer et al. [60] and will be used here. Hairer et al. [60] consider the autonomous index 2 DAE

$$0 = g(u), \quad (6.12)$$

$$\dot{u} = f(u, p). \quad (6.13)$$

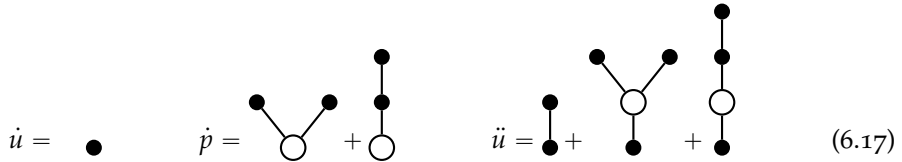
The non-autonomous system (5.1)-(5.2) can be written in this form by adding  $\dot{t} = 1$  so that equations (6.12)-(6.13) hold by redefining  $u := \begin{pmatrix} u \\ t \end{pmatrix}$  and  $f := \begin{pmatrix} f \\ 1 \end{pmatrix}$ . In the Taylor expansion of the exact and numerical solution,  $\dot{p}, \ddot{p}, \dots$  appear next to  $\dot{u}, \ddot{u}, \dots$ . The first few derivatives read (see [60, 63]):

$$\dot{u} = f, \tag{6.14}$$

$$\dot{p} = (-g_u f_p)^{-1}(g_{uu}(f, f) + g_u f_u f), \tag{6.15}$$

$$\ddot{u} = f_u f + f_p (-g_u f_p)^{-1}(g_{uu}(f, f) + g_u f_u f). \tag{6.16}$$

For DAEs, the number of differentials grows even more rapidly for higher order derivatives. Trees still provide a compact way to represent these derivatives, when extended to contain both meagre (solid) and fat (open) vertices:



The *order* of a tree is the number of meagre vertices minus the number of fat vertices [60]. To have a local error of  $\mathcal{O}(\Delta t^{r+1})$  (global error  $\mathcal{O}(\Delta t^r)$ ) the order conditions should be satisfied up to and including tree order  $r$ . (An exception is the case where a differential corresponding to a tree is of the form  $f_p \cdot (\dots)$ . The corresponding order condition then has to be considered for order  $r + 1$  instead of  $r$  [60].) These order conditions can be read again from the trees, as is outlined in Hairer et al. [60] and Brasey and Hairer [20]. As an example, the above trees correspond to the following order conditions:

$$\dot{u} : \sum b_i = 1, \tag{6.18}$$

$$\dot{p} : \sum b_i \omega_{ij} \omega_{jk} c_k^2 = 2, \quad \sum b_i \omega_{ij} \omega_{jk} a_{kl} c_l = 1, \tag{6.19}$$

$$\ddot{u} : \sum b_i c_i = \frac{1}{2}, \quad \sum b_i \omega_{ij} c_j^2 = 1, \quad \sum b_i \omega_{ij} a_{jk} c_k = \frac{1}{2}, \tag{6.20}$$

the summation being again over all indices. In this example we see that next to the classical order conditions (represented by trees with only meagre vertices) additional order conditions appear, corresponding to trees with fat vertices, which include the inverse of matrix  $A$  ( $\omega_{ij}$  denotes the entries of  $A^{-1}$ ). The order conditions for the  $p$ -component are especially difficult due to the presence of  $(A^{-1})^2$ . Fortunately, some of these additional trees do not pose additional constraints on

the coefficients because they reduce to classical order conditions. For example, the last condition in equation (6.20) can be written as

$$\sum b_i \omega_{ij} a_{jk} c_k = b^T A^{-1} A c = \sum b_i c_i = \frac{1}{2}, \quad (6.21)$$

so it reduces to the classical second-order condition. The additional order conditions that cannot be simplified to classical order conditions are of interest to us. To find these remaining conditions we used the software described in [30]. We found that for the  $u$ -component there is no additional tree for order 1, but there is 1 for order 2, there are 4 for order 3 and 17 for order 4. For the  $p$ -component there are 2 trees for order 1, 6 for order 2, 21 for order 3, and 81 for order 4.

This large number of additional order conditions can still be considerably reduced when taking into account two important facts, namely that (i) we are considering half-explicit methods and (ii) we are applying these to the Navier-Stokes equations.

### 6.2.3 Application to the incompressible Navier-Stokes equations

For the Navier-Stokes equations we know that  $f(u, p, t) = F(u, t) - Gp$ , which means that  $f_p = G$  is a constant matrix and thus all derivatives of  $f_p$ , such as  $f_{pu}$ ,  $f_{pp}$ , etc., are zero. Therefore trees which have a meagre vertex as root or as branch and connected to it a fat vertex and at least one other meagre or fat vertex need not be considered. In case  $G$  is a function of time, one cannot remove trees with derivatives of the form  $f_{pu}$ ,  $f_{puu}$ , but only those of the form  $f_{pp}$ ,  $f_{ppp}$ , etc.; we will treat this in section 6.2.5.

We note that in the case of non-autonomous systems,  $g_{uu}$  in equations (6.15) and (6.16) consists of  $g_{uu}$ ,  $g_{ut}$  and  $g_{tt}$ .  $g_{uu}$  and  $g_{ut}$  are zero for the Navier-Stokes equations, but  $g_{tt} = \ddot{r}_1(t)$  is in general not, and therefore trees that have a fat vertex with more than one meagre vertex connected to it do not vanish. This also covers the case of a time-dependent  $M$  matrix. The special case of  $\dot{r}_1(t) = 0$ , so that  $g_{tt} = 0$ , will be discussed in section 6.3.3.

### 6.2.4 Half-explicit methods

For half-explicit methods the construction of the order conditions changes slightly: if a meagre vertex follows a fat vertex then the index changes from  $a_{jk}$  to  $\tilde{a}_{jk}$  (or from  $c_j$  to  $\tilde{c}_j$ ). The trees and order conditions that result with this notation, and after removing all trees containing a derivative of  $f_p$ , are shown in tables 6.1 and 6.2. Contrary to the  $u$ -component, the order of accuracy of the global error of the  $p$ -component is equal to the order of the tree plus 1, because there is no power lost when going from local error to global error.

Table 6.1 shows that for the  $u$ -component up to and including order 4 only 3 trees with associated additional order conditions remain. The ‘shift’ in order indicated by

an arrow  $r \rightarrow r + 1$  is due to the form  $f_p \cdot (\dots)$  of the remaining trees, as was mentioned before. The order conditions corresponding to these trees can be simplified for the explicit methods under consideration. Considering that  $b$  is the last row of  $\tilde{A}$ , we can write

$$b^T = \begin{pmatrix} 0 & 0 & \dots & 1 \end{pmatrix} \tilde{A}. \tag{6.22}$$

As an example, the order condition for tree number 3 can be simplified to

$$\sum b_i \tilde{\omega}_{ij} \tilde{c}_j \tilde{a}_{jk} c_k = (00 \dots 1) \tilde{A} \tilde{A}^{-1} (\tilde{c} \circ (\tilde{A}c)) = \tilde{c}_s \sum b_i c_i = \sum b_i c_i = \frac{1}{2}, \tag{6.23}$$

where  $\circ$  denotes the elementwise product, i.e.,  $c = a \circ b$  means  $c_i = a_i b_i$ . If the Runge-Kutta method satisfies the classical second-order condition  $\sum b_i c_i = \frac{1}{2}$  this additional order condition is satisfied. A similar simplification of trees 1 and 2 yields the conditions  $\tilde{c}_s^2 = 1$  and  $\tilde{c}_s^3 = 1$ , which are automatically satisfied, see equation (6.2). In conclusion, the specific form of the pressure term in the Navier-Stokes equations *and* the use of explicit methods leads to the observation that all additional order conditions are trivially satisfied, at least up to and including order 4. This is also true for order 5, for which 6 stages are needed (although methods with order higher than 4 are hardly used for the time integration of the incompressible Navier-Stokes equations). We conjecture that this is true for any order, i.e., *when applying an explicit Runge-Kutta method to the incompressible Navier-Stokes equations no additional order conditions appear for the u-component, and consequently no order reduction occurs.*

On the other hand, order reduction will occur if the continuity equation  $M\tilde{U}_i = r_1(\tilde{t}_i)$  is not satisfied at all intermediate stages. Since this does not affect the stability domain of the method, this method can still be of interest to compute steady flows with a time stepping technique, because stability is then much more important than temporal accuracy. The effect on the order of accuracy of the velocity requires the study of the influence of perturbations in the constraint on the velocity at the end of the time step. Such a study is left as a suggestion for further research; here we focus on methods that satisfy the constraint at each stage, so  $s$  Poisson equations are solved for an  $s$ -stage method.

For the  $p$ -component the above simplification of the additional order conditions is not possible and we have to look in more depth in whether the remaining order conditions can be satisfied. This will be detailed in section 6.3.

### 6.2.5 Time-dependent operators

The interesting case of a time-dependent gradient operator  $G(t)$  can be treated in an analogous fashion. The additional trees that result when  $f_p$  is not constant but depends on time leads to the trees shown in table 6.3. In contrast to table 6.1, the order conditions associated with these trees do *not* reduce to classical order conditions. This table is similar to table 1 in [20], but with the difference that in that work trees containing  $f_{pp}$ ,  $f_{ppp}$  and  $f_{ppu}$  are also present. We present this table here



as a reference for practitioners of Runge-Kutta methods for time integration of the incompressible Navier-Stokes equations on time-varying meshes.

For a third order method there is one additional tree, denoted by number 10. Evaluating the condition associated with this tree for a three-stage method, together with the four classical order conditions for third order methods, leads to a solution family with  $c_3 = 1$ , and  $c_2$  as a free parameter ( $c_2 \neq 0$ ,  $c_2 \neq \frac{2}{3}$ ,  $c_2 \neq 1$ ):

$$a_{21} = c_2 \quad a_{31} = \frac{3c_2 - 3c_2^2 - 1}{c_2(2 - 3c_2)} \quad a_{32} = \frac{1 - c_2}{c_2(2 - 3c_2)}, \quad (6.24)$$

$$b_1 = \frac{3c_2 - 1}{6c_2} \quad b_2 = \frac{1}{6c_2(1 - c_2)} \quad b_3 = \frac{2 - 3c_2}{6(1 - c_2)}. \quad (6.25)$$

This family excludes Wray's popular third-order method [94]. Wray's method reduces to second order for time-dependent operators, such as moving meshes, and is therefore *not* recommended in this case.

For a fourth order method six additional trees appear due to the time dependency of  $G$ . It is proven in [20] that the order condition corresponding to tree number 15 cannot be satisfied with a four-stage, fourth-order method. An example of a five-stage, fourth-order method that satisfies the conditions corresponding to trees 10-15 is the HEM4 method [20]. It does not satisfy the conditions corresponding to trees 6 and 7, so it is second order accurate for the pressure.

Note that the time-dependence of  $f_p$  also leads to additional trees for the pressure, next to those already mentioned in table 6.2. They are of order 2 or higher.

### 6.2.6 A note on space-time errors

At this point it is worthwhile to mention that, apart from the order reduction mechanism discussed above (a result of the differential-algebraic nature of the Navier-Stokes equations), another mechanism for order reduction exists. This mechanism, analyzed for example in [31, 1, 74, 124], can result when Runge-Kutta methods are applied to PDEs with time-dependent inflow boundary conditions and the exact boundary values are imposed for the intermediate stages (as we do in the current work, see equation (5.12)). Order reduction then appears when studying the full space-time error, i.e., when simultaneously refining mesh and time step (for example mesh refinement at a fixed CFL number). A solution to this problem is to not impose any intermediate boundary values but instead obtain these values by integrating the semi-discrete equations at the boundary, using one-sided difference stencils to approximate the spatial derivatives (perhaps using the boundary discretizations from chapter 4). It is questionable if this is a mathematically valid approach for the incompressible Navier-Stokes equations, but in any case it significantly reduces the allowable time step for stability [124]. Another fix, for linear and non-linear hyperbolic PDEs, is presented in [31, 1], which boils down to repeated differentiation of the boundary condition and then cleverly integrating it along with the Runge-Kutta method for the interior points. However, it is questionable if such

a fix also works for the incompressible Navier-Stokes equations (a system of DAEs of mixed parabolic-elliptic type), because a change in boundary conditions for the intermediate stages affects the divergence equation. As mentioned before, this requires a study of the effect of perturbations in the divergence-free constraint on the order of accuracy. Furthermore, the fix of [31, 1] requires that  $r_1$  can be differentiated with respect to time; this derivative is not always available (as in the case of a turbulent inflow field) or might not even exist (e.g. a boundary condition of the form  $e^{1-1/t}$  for  $t \geq 0$  does not have a proper derivative at  $t = 0$ ). Note that this differentiability of  $r_1$  with respect to time will be encountered again in the next section when discussing the accuracy of the pressure. A cure for DAEs (or even for the specific case of the incompressible Navier-Stokes equations) is, to the authors' knowledge, not yet available. Fortunately, the order reduction from unsteady boundary conditions as described above, generally only manifests itself at very fine grids due to a small coefficient of the leading error term [31]. For the numerical experiments in this chapter we use the second order accurate discretization, so that a possible order reduction in the global error is likely to be overwhelmed by the spatial error. Therefore, in our current work we focus on the temporal error only; we fix the mesh and then refine the time step. We leave the possible interplay of temporal and spatial errors as suggestion for future research.

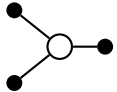
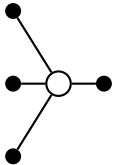
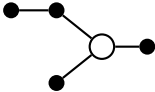
Tree	Order	Condition	Simplifies to	Differential
	2 → 3	$\sum b_i \tilde{\omega}_{ij} \tilde{c}_j^2 = 1$	$\tilde{c}_s^2 = 1$	$f_p(-g_u f_p)^{-1} g_{uu}(f, f)$
	3 → 4	$\sum b_i \tilde{\omega}_{ij} \tilde{c}_j^3 = 1$	$\tilde{c}_s^3 = 1$	$f_p(-g_u f_p)^{-1} g_{uuu}(f, f, f)$
	3 → 4	$\sum b_i \tilde{\omega}_{ij} \tilde{c}_j \tilde{\alpha}_{ijk} c_k = \frac{1}{2}$	$\sum b_i c_i = \frac{1}{2}$	$f_p(-g_u f_p)^{-1} g_{uu}(f, f, f)$

Table 6.1: Additional trees and order conditions for  $u$ -component up to and including order 4 when  $f_p = \text{constant}$ .





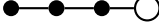
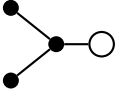
Tree	Order	Condition	Simplifies to	Differential
	1	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{c}_k^2 = 2$	$\sum \tilde{\omega}_{si} \tilde{c}_i^2 = 2$	$(-g_{ufp})^{-1} g_{uu}(f, f)$
	1	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{a}_{kl} c_l = 1$	$c_s = 1$	$(-g_{ufp})^{-1} g_{ufuf}$
	2	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{c}_k \tilde{a}_{kl} c_l = \frac{3}{2}$	$\sum \tilde{\omega}_{si} \tilde{c}_i \tilde{a}_{ij} c_j = \frac{3}{2}$	$(-g_{ufp})^{-1} g_{uu}(f, f, f)$
	2	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{c}_k^3 = 3$	$\sum \tilde{\omega}_{si} \tilde{c}_i^3 = 3$	$(-g_{ufp})^{-1} g_{uuu}(f, f, f)$
	2	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{a}_{kl} a_{lm} c_m = \frac{1}{2}$	$\sum a_{si} c_i = \frac{1}{2}$	$(-g_{ufp})^{-1} g_{ufufuf}$
	2	$\sum b_i \tilde{\omega}_{ij} \tilde{\omega}_{jk} \tilde{a}_{kl} c_l^2 = 1$	$c_s^2 = 1$	$(-g_{ufp})^{-1} g_{ufuu}(f, f)$

Table 6.2: Trees and order conditions for  $p$ -component up to and including order 2 when  $f_p = \text{constant}$ .

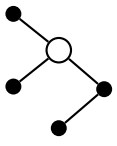
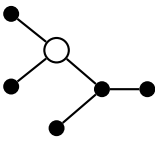
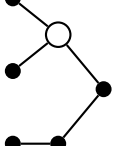
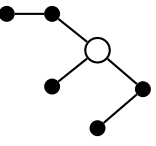
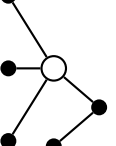
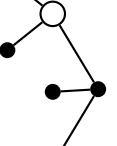
Tree	Order	Condition	Differential
	3	$\sum b_i c_i \tilde{\omega}_{ij} \tilde{c}_j^2 = \frac{2}{3}$	$f_{pu}(f, (-g_{ufp})^{-1} g_{uu}(f, f))$
	4	$\sum b_i a_{ij} c_i \tilde{\omega}_{jk} \tilde{c}_k^2 = \frac{1}{6}$	$f_{ufpu}(f, (-g_{ufp})^{-1} g_{uu}(f, f))$
	4	$\sum b_i a_{ij} c_i \tilde{\omega}_{jk} \tilde{c}_k^2 = \frac{1}{4}$	$f_{pu}(f_{uf}, (-g_{ufp})^{-1} g_{uu}(f, f))$
	4	$\sum b_i c_i \tilde{\omega}_{ij} \tilde{c}_j \tilde{a}_{jk} c_k = \frac{3}{8}$	$f_{pu}(f, (-g_{ufp})^{-1} g_{uu}(f, f_{uf}))$
	4	$\sum b_i c_i \tilde{\omega}_{ij} \tilde{c}_j^3 = \frac{3}{4}$	$f_{pu}(f, (-g_{ufp})^{-1} g_{uuu}(f, f, f))$
	4	$\sum b_i c_i^2 \tilde{\omega}_{ij} \tilde{c}_j^2 = \frac{1}{2}$	$f_{puu}(f, f, (-g_{ufp})^{-1} g_{uuu}(f, f, f))$

Table 6.3: Additional trees and order conditions for  $u$ -component up to and including order 4 when  $f_p$  is a function of  $t$ .

## 6.3 THE ACCURACY OF THE PRESSURE

## 6.3.1 Single Butcher tableau for velocity and pressure (Method 1)

We continue with an order study of the pressure when formulation (6.4)-(6.6) with (5.31) is employed. As was the case in table 6.1, the order conditions in table 6.2 can also be simplified in certain cases by employing equation (6.22). However, in contrast to the  $u$ -component, additional order conditions remain, even for a second-order method. This is not a surprise when considering that the Lagrange multipliers  $\phi$  and  $\psi$  are of a different nature than the pressure  $p$  (integral versus point value). Here we concentrate on obtaining an accurate point value for the pressure. Such a point value is of interest when comparing, for example, a pressure distribution at a certain time instant with an experimentally obtained pressure distribution at the same time instant. However, in other cases, such as computing the displacement of a body due to aerodynamic forces in a CFD code for fluid-structure interaction, the integral value can be a better quantity to use.

We should note that all additional order conditions for the pressure can be circumvented *entirely* by solving an additional Poisson equation, equation (5.7), at  $t_{n+1}$ :

$$Lp_{n+1} = MF_{n+1} - \dot{r}_1(t_{n+1}). \quad (6.26)$$

Given an  $r$ -th order accurate velocity field  $u_{n+1}$ , the resulting pressure  $p_{n+1}$  is of the same order of accuracy. However, there are two issues in solving equation (6.26). Firstly it is required that  $r_1(t)$  can be differentiated (analytically or numerically), something which is not required in the computation of  $u$  and  $\phi$ . In many practical computations, for example involving a prescribed turbulent inflow,  $\dot{r}_1(t)$  might not be available. Secondly, solving equation (6.26) amounts to the solution of an additional Poisson equation, which is computationally costly. We will therefore look at the additional conditions of table 6.2, which, when satisfied, give a higher order accurate pressure without solving equation (6.26).

## 6.3.1.1 Two-stage methods

For two-stage, second-order methods we have the classical conditions  $b_1 + b_2 = 1$  and  $b_2c_2 = \frac{1}{2}$ . For a second-order accurate pressure the conditions corresponding to trees 4 and 5 have to be satisfied as well. The additional order condition corresponding to tree 4 is

$$\sum_{i=1}^2 \tilde{\omega}_{2i} \tilde{c}_i^2 = 2, \quad (6.27)$$

where

$$(\omega_{ij}) = A^{-1} = \begin{pmatrix} \frac{1}{a_{21}} & 0 \\ -\frac{b_1}{a_{21}b_2} & \frac{1}{b_2} \end{pmatrix}. \quad (6.28)$$

This results in the condition

$$\frac{a_{21} - b_1 c_2}{a_{21} b_2} = 2. \quad (6.29)$$

The order condition corresponding to tree 5 is

$$c_2 = 1. \quad (6.30)$$

This latter condition results in  $b_1 = b_2 = \frac{1}{2}$  after applying the classical order conditions. However, condition (6.29) can then not be satisfied, because it reduces to  $a_{21} - \frac{1}{2} = a_{21}$ . It is therefore not possible to obtain better than first-order accuracy for the pressure with a two-stage explicit method.

### 6.3.1.2 Three-stage methods

Butcher [25] lists three cases for which a three-stage, third-order explicit method exists. Only in the 'case I' family there is a solution that allows  $c_3 = 1$  (the condition corresponding to tree 5), which is the same as solution family (6.24)-(6.25). Evaluating the order condition corresponding to tree 4 for this family leads to

$$\frac{3c_2^2 - 7c_2 + 4}{3c_2 - 2} = 2, \quad (6.31)$$

which has only one valid solution, being  $c_2 = \frac{1}{3}$ . The resulting Butcher tableau is

$$\begin{array}{c|cc} 0 & 0 & \\ \frac{1}{3} & \frac{1}{3} & \\ 1 & -1 & 2 \\ \hline & 0 & \frac{3}{4} & \frac{1}{4} \end{array} \quad (6.32)$$

which satisfies indeed trees 4 and 5. Evaluating equation (5.30) gives

$$p_{n+1} = \tilde{\psi}_3 = -\frac{3}{2}\tilde{\phi}_1 - \frac{3}{2}\tilde{\phi}_2 + 4\tilde{\phi}_3. \quad (6.33)$$

The conditions corresponding to trees 6, 7 and 8 are not satisfied, and the pressure is at best second-order accurate.

We remark that the occurrence of negative coefficients in (6.32) and (6.33) will in general not lead to spurious, non-positive solution behavior, as long as the incompressible Navier-Stokes solutions are 'smooth' enough. In chapter 8 we will encounter situations where this is not the case. No measures will be taken to make (6.32), (6.33), or any following method positive.

6.3.1.3 *Four-stage methods*

For explicit four-stage, fourth-order methods the classical order conditions require  $c_4 = 1$  (see e.g. [25]), which tree 5 (and 9) automatically satisfy. We found three methods that also satisfy the condition corresponding to tree number 4; they read:

$$\begin{array}{c|ccc}
 0 & 0 & & \\
 1 & 1 & & \\
 \frac{1}{2} & \frac{3}{8} & \frac{1}{8} & \\
 1 & -\frac{1}{8} & -\frac{3}{8} & \frac{3}{2} \\
 \hline
 & \frac{1}{6} & -\frac{1}{18} & \frac{2}{3} \quad \frac{2}{9}
 \end{array}
 \quad
 \begin{array}{c|ccc}
 0 & 0 & & \\
 \frac{2}{3} & \frac{2}{3} & & \\
 \frac{7}{12} & \frac{91}{192} & \frac{7}{64} & \\
 1 & \frac{1}{7} & -2 & \frac{20}{7} \\
 \hline
 & \frac{5}{28} & -\frac{3}{4} & \frac{48}{35} \quad \frac{1}{5}
 \end{array}
 \quad
 \begin{array}{c|ccc}
 0 & 0 & & \\
 \frac{3}{4} & \frac{3}{4} & & \\
 \frac{5}{9} & \frac{100}{243} & \frac{35}{243} & \\
 1 & \frac{4}{75} & -\frac{19}{21} & \frac{324}{175} \\
 \hline
 & \frac{8}{45} & -\frac{16}{63} & \frac{243}{280} \quad \frac{5}{24}
 \end{array}
 \tag{6.34}$$

For example, evaluating equation (5.30) for the left tableau gives

$$p_{n+1} = \tilde{\psi}_4 = \frac{1}{2}\tilde{\phi}_1 - 2\tilde{\phi}_2 - 2\tilde{\phi}_3 + \frac{9}{2}\tilde{\phi}_4. \tag{6.35}$$

As can be readily calculated, none of the above methods satisfies the conditions corresponding to trees 6, 7 and 8. Therefore, with a four-stage, fourth-order explicit method the pressure is, again, at best second-order accurate.

6.3.2 *Reconstructing instantaneous pressure values from time averages (Method 2)*

We mentioned in section 5.3.2 that  $\tilde{\phi}_i$  defined by (5.24) is only first-order accurate in time but that higher order accurate pressures  $\tilde{\psi}_i$  are possible with the generalized formulation (5.29). With the choice  $\tilde{A}^p = \tilde{A}$  this led to the order conditions outlined in section 6.3.1, and it appeared that only a limited number of three-stage and four-stage methods lead to a second-order accurate pressure. In this section we take a different approach by relating the integral averages  $\tilde{\phi}_i$  to the point value  $p_{n+1}$ . In fact, this boils down to taking  $\tilde{A}^p \neq \tilde{A}$ , although it is not necessary to explicitly derive  $\tilde{A}^p$ .

First we consider the *exact* integration of equation (5.7) from  $t_n$  to  $\tilde{t}_i$ , which reads

$$L \int_{t_n}^{\tilde{t}_i} p(t) dt = M \int_{t_n}^{\tilde{t}_i} F(t) dt - (r_1(\tilde{t}_i) - r_1(t_n)), \tag{6.36}$$

and we denote the exact average of  $p$  over this interval by  $\phi(\tilde{t}_i)$  (the exact counterpart of the approximation  $\tilde{\phi}_i$ ):

$$\phi(\tilde{t}_i) = \frac{1}{\tilde{c}_i \Delta t} \int_{t_n}^{\tilde{t}_i} p(t) dt. \tag{6.37}$$



The challenge is to find a higher order accurate point value  $p_{n+1}$  from the time average values  $\phi(\tilde{t}_i)$ . This is schematically shown in figure 6.1. Such an approximation of point values from integral averages is well-known in the field of Essentially Non-Oscillatory (ENO) conservative finite difference schemes and is called *reconstruction*. We follow [165] to perform this reconstruction, and refer to that work for more details. Denoting the primitive function of  $p(t)$  by  $P(t)$ , we can write

$$\phi(\tilde{t}_i) (\tilde{c}_i \Delta t) = \int_{t_n}^{\tilde{t}_i} p(t) dt = P(\tilde{t}_i) - P(t_n), \quad i = 1, 2, \dots, s. \quad (6.38)$$

We then construct a polynomial  $H(t)$  that interpolates  $P(t)$  at the following points:

$$\tilde{t}_{k_1}, \tilde{t}_{k_2}, \dots, \tilde{t}_{k_m}, \quad (6.39)$$

where  $K = \{k_1, \dots, k_m\}$  is the set of points that will be used in the interpolation. We always take  $k_1 = 0$  and  $k_m = s$  (using the convention that  $\tilde{t}_0 = t_n$ ). The other values of  $k$  depend on which of the intermediate stages are used in the interpolation. One could take all points, i.e.,  $K = \{0, 1, \dots, s\}$ , but this is in general not necessary, as we will show later. The derivative of  $H(t)$  is denoted by  $h(t)$ . Then  $h(t)$  is an approximation to  $p(t)$ , and their integrals are the same:

$$\begin{aligned} \frac{1}{\tilde{c}_i \Delta t} \int_{t_n}^{\tilde{t}_i} h(t) dt &= \frac{1}{\tilde{c}_i \Delta t} \int_{t_n}^{\tilde{t}_i} H'(t) dt = \frac{1}{\tilde{c}_i \Delta t} (H(\tilde{t}_i) - H(t_n)), \\ &= \frac{1}{\tilde{c}_i \Delta t} (P(\tilde{t}_i) - P(t_n)) = \frac{1}{\tilde{c}_i \Delta t} \int_{t_n}^{\tilde{t}_i} p(t) dt = \phi(\tilde{t}_i). \end{aligned} \quad (6.40)$$

The crucial point in this derivation is that  $H(t)$  interpolates  $P(t)$  exactly at the points used for the construction of the polynomial. We employ the Lagrange form of the interpolation polynomial, i.e., we write

$$H(t) = \sum_{k \in K} P(\tilde{t}_k) \ell_k(t), \quad (6.41)$$

where

$$\ell_k(t) = \prod_{j \in K, j \neq k} \frac{t - \tilde{t}_j}{\tilde{t}_k - \tilde{t}_j}. \quad (6.42)$$

For a well-posed polynomial we require that all  $\tilde{t}_k$  are distinct.  $H(t)$  can be written in terms of the integral values  $\phi(\tilde{t}_i)$  by subtracting  $P(t_n)$  from both sides and using  $\sum_{k \in K} \ell_k(t) = 1$ :

$$H(t) - P(t_n) = \sum_{k \in K'} \phi(\tilde{t}_k) \tilde{c}_k \Delta t \ell_k(t), \quad (6.43)$$

where  $K' = \{k_2, \dots, k_m\}$ . Differentiation then leads to

$$h(t) = \sum_{k \in K'} \phi(\tilde{t}_k) \tilde{c}_k \Delta t \ell'_k(t). \quad (6.44)$$

Given the values  $\phi(\tilde{t}_k)$  and the points  $\tilde{t}_k$  this expression can be evaluated at  $t_{n+1}$ , which provides the approximation we are looking for:

$$p_{n+1} = h(t_{n+1}). \quad (6.45)$$

Using  $m - 1$   $\phi$  values (the number of elements in  $K'$ ) this results in an  $(m - 1)$ -th order accurate pressure. If all  $\phi$  values are used, then  $m - 1 = s$  and one could, in theory, obtain an  $s$ -th order accurate pressure. Note that, in contrast to ENO schemes, we do not choose the reconstruction points in such a way that the smoothest stencil results.

In practice we cannot use the *exact* average  $\phi(\tilde{t}_i)$  to find  $p_{n+1}$ .  $\phi(\tilde{t}_i)$  is approximated by  $\tilde{\phi}_i$ , whose order of accuracy depends on the *stage order* of the method. This stage order can be expressed by making use of simplifying condition  $C_i(\eta)$ , equation (5.17). We remark that with the notation of the shifted tableau,  $\tilde{C}_s(p)$  is equal to simplifying condition  $B(p)$ . Assuming that  $\tilde{C}_i(q)$  holds, the equation for  $\phi(\tilde{t}_i)$  can be written as

$$L\phi(\tilde{t}_i) = \frac{1}{\tilde{c}_i} \sum_{j=1}^i \tilde{a}_{ij} M F_j - \frac{r_1(\tilde{t}_i) - r_1(t_n)}{\tilde{c}_i \Delta t} + \mathcal{O}(\Delta t^q), \quad (6.46)$$

so that the difference between the exact integral and its numerical approximation is  $\mathcal{O}(\Delta t^q)$ :

$$\phi(\tilde{t}_i) = \tilde{\phi}_i + \mathcal{O}(\Delta t^q). \quad (6.47)$$

To summarize, it is possible to obtain a higher order accurate pressure at  $t_{n+1}$  by combining the average values  $\tilde{\phi}_i$  from the different stages. To attain a certain order  $r$  requires at least  $r$  distinct stages (i.e. with different  $c_i$ ), and each individual stage  $i$  should have stage order  $r$ , i.e., satisfy  $C_i(r)$ . We will now check if this is possible for methods with two, three and four stages.

### 6.3.2.1 Two-stage methods

For two-stage, second-order methods  $\tilde{C}_2(2) = B(2)$  is obviously satisfied, but  $\tilde{C}_1(2)$  cannot be satisfied because the equation for  $\tilde{U}_1$  is simply a Forward Euler step, which is first-order accurate (this is always the case for explicit methods).

### 6.3.2.2 Three-stage methods

Since  $\tilde{C}_1(2)$  cannot be satisfied, we require  $\tilde{C}_2(2)$  to be satisfied, i.e.,

$$a_{32}c_2 = \frac{1}{2}c_3^2, \quad (6.48)$$

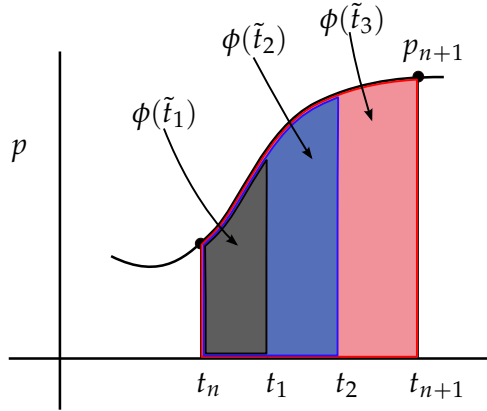


Figure 6.1: Combining  $\phi(t_i)$  to obtain a higher order accurate  $p_{n+1}$ .

together with the condition  $c_3 \neq 1$  to have distinct  $c$ 's. Using the third-order conditions

$$b_3 a_{32} c_2 = \frac{1}{6}, \quad b_2 c_2^2 + b_3 c_3^2 = \frac{1}{3}, \tag{6.49}$$

this leads to  $b_1 = \frac{1}{4}$ ,  $b_2 = 0$ ,  $b_3 = \frac{3}{4}$  and  $c_3 = \frac{2}{3}$ .  $c_2$  can be chosen freely ( $\neq 0$ ), and then determines  $a_{31}$  and  $a_{32}$ . Wray's popular third-order method [94] falls in this category, with  $c_2 = \frac{8}{15}$  - see equation (6.50).

$\begin{array}{c cc} 0 & 0 & \\ \hline \frac{8}{15} & \frac{8}{15} & \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} \\ \hline & \frac{1}{4} & 0 \quad \frac{3}{4} \end{array}$	(6.50)	$\begin{array}{c ccc} 0 & 0 & \\ \hline \frac{2}{3} & \frac{2}{3} & \\ \frac{2}{3} & \frac{1}{3} & \frac{1}{3} \\ \hline & \frac{1}{4} & 0 \quad \frac{3}{4} \end{array}$	(6.51)
---	--------	---	--------

Another possibility is to take  $c_2 = c_3$ , which saves an evaluation of boundary conditions and forcing terms; this gives tableau (6.51). The interpolation polynomial  $h(t)$  from equation (6.44) is independent of  $c_2$  and given by:

$$h(t) = - \left( \frac{2 \frac{t-t_n}{\Delta t} - 1}{1 - c_3} \right) \tilde{\phi}_2 + \left( \frac{2 \frac{t-t_n}{\Delta t} - c_3}{1 - c_3} \right) \tilde{\phi}_3, \tag{6.52}$$

and  $p_{n+1}$  follows with  $c_3 = \frac{2}{3}$  as

$$p_{n+1} = h(t_{n+1}) = -3\tilde{\phi}_2 + 4\tilde{\phi}_3. \tag{6.53}$$

This equation provides a new way to obtain a second-order accurate pressure by combining two first-order accurate pressures of a three-stage method. It is also

valid when Wray’s method is used only for the convective terms and an appropriate implicit method for the diffusive terms, such as the method from [181].

If one wants to maximize stability instead of order of accuracy (second-order accuracy is sufficient in many practical applications), one can use three-stage methods that are second order for the velocity. Combining the condition for maximum stability along the imaginary axis ( $b_3 a_{32} c_2 = \frac{1}{4}$ ) with condition (6.48) yields a family of methods with  $c_2$  and  $c_3$  as free parameters. An example of a low-storage method satisfying these conditions is presented in Perot and Nallapati [127], but it has  $c_3 = 1$  so a second-order accurate pressure cannot be obtained. We propose the following alternative method

$$\begin{array}{c|ccc}
 0 & 0 & & \\
 \frac{1}{2} & \frac{1}{2} & & \\
 \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \\
 \hline
 & 0 & -1 & 2
 \end{array} \tag{6.54}$$

which we obtained by requiring  $b_1 = 0$  and  $c_2 = c_3$ . The requirement  $b_1 = 0$  leads to the same storage requirements as [127], and  $c_2 = c_3$  has the advantage that only one intermediate boundary condition evaluation is needed.

6.3.2.3 Four-stage methods

Four-stage, fourth-order methods have  $c_4 = 1$ , so that even if  $\tilde{C}_3(2)$  would hold, it cannot be used. We therefore look again for methods that satisfy  $\tilde{C}_2(2)$ . Combining condition (6.48) with the fourth-order conditions leads to  $c_3 = \frac{1}{2}$  and two families of solutions result, corresponding to the ‘case II’ and ‘case IV’ solutions found by Kutta [25]:

$$\begin{array}{c|ccc}
 0 & 0 & & \\
 c_2 & c_2 & & \\
 \frac{1}{2} & \frac{1}{2} - \frac{1}{8c_2} & \frac{1}{8c_2} & \\
 1 & \frac{1}{2c_2} - 1 & -\frac{1}{2c_2} & 2 \\
 \hline
 & \frac{1}{6} & 0 & \frac{2}{3} \quad \frac{1}{6}
 \end{array} \tag{6.55}$$

$$\begin{array}{c|ccc}
 0 & 0 & & \\
 1 & 1 & & \\
 \frac{1}{2} & \frac{3}{8} & \frac{1}{8} & \\
 1 & 1 - \frac{1}{4b_4} & -\frac{1}{12b_4} & \frac{1}{3b_4} \\
 \hline
 & \frac{1}{6} & \frac{1}{6} - b_4 & \frac{2}{3} \quad b_4
 \end{array} \tag{6.56}$$

The left tableau with  $c_2 = c_3 = \frac{1}{2}$  is attractive because it requires only one intermediate evaluation of boundary conditions and forcing terms. In a similar fashion as the three-stage method (equation (6.52) with  $\tilde{\phi}_3$  replaced by  $\tilde{\phi}_4$ ),  $p_{n+1}$  follows as

$$p_{n+1} = -2\tilde{\phi}_2 + 3\tilde{\phi}_4. \tag{6.57}$$

Again, this equation provides a new way to obtain a second-order accurate pressure by combining two first-order accurate pressures of a four-stage method.

### 6.3.3 Steady boundary conditions for the continuity equation (Method 3)

An important case for the incompressible Navier-Stokes equations is when the boundary conditions for the continuity equation are steady, i.e., equation (3.1) can be written as

$$Mu = r_1, \quad (6.58)$$

where  $r_1$  is independent of  $t$ . Equation (5.1) then reads  $g(u) = 0$ , with  $g$  linear in  $u$ . This means that  $g_u = \text{constant}$  and all partial derivatives of  $g_u$  ( $g_{uu}$ ,  $g_{ut}$ ,  $g_{tt}$ , etc.) are zero. As before, all additional trees for the  $u$ -component (table 6.1) disappear, but most of the trees for the  $p$ -component (table 6.2) also vanish. Only trees 5, 8 and 9 remain, and it is possible to find higher order accurate methods for the pressure. For example, with two stages a second-order accurate pressure is possible ( $c_2 = 1$ ), and with four stages a third-order accurate pressure is possible. However, it is not necessary to consider such methods, because in the case  $g_u = \text{constant}$  the pressure can be computed to the same order of accuracy as the velocity, without additional cost. This can be seen by comparing equation (5.24) for  $i = 1$  with equation (6.26):

$$Lp_n = MF_n - \dot{r}_1(t_n), \quad (6.59)$$

$$L\tilde{\phi}_1 = \frac{\tilde{a}_{11}}{\tilde{c}_1} MF_1 - \frac{r_1(\tilde{t}_1) - r_1(t_n)}{\tilde{c}_1 \Delta t}. \quad (6.60)$$

Considering that  $F_1$  is equal to  $F_n$  (determined at the end of the previous time step) and that  $\tilde{a}_{11} = \tilde{c}_1$ , these expressions are equal if  $r_1$  is independent of time (or a linear function of  $t$ ). This means that  $\tilde{\phi}_1$  is actually the  $r$ -th order accurate pressure at  $t_n$ , and the additional Poisson solve associated with equation (6.26) can be avoided. An existing implementation could remain unaltered; instead of taking  $\tilde{\phi}_s = \phi_{n+1}$ , the pressure that makes  $u_{n+1}$  divergence free, one should take  $\tilde{\phi}_1$  from the next time step to have a higher order accurate pressure at the end of the current time step. We prefer to compute  $p_{n+1}$  and then skip the computation of  $\tilde{\phi}_1$  in the next time step. This works for any explicit Runge-Kutta method with at least two stages, thus providing a simple way to improve the temporal accuracy of the pressure without increasing computational cost.

## 6.4 RESULTS

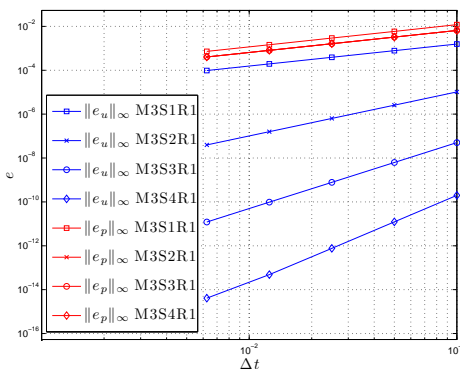
We classify the methods described in the previous sections as follows. The methods that were derived in section 6.3.1 (with  $\tilde{A}^p = \tilde{A}$ ) will be indicated by M1, the methods derived in section 6.3.2 are indicated by M2, and in case section 6.3.3 applies we write M3. We then write  $MmSsRr$  to indicate that an  $s$ -stage explicit Runge-Kutta method of type  $m$  is used with order  $s$  for the velocity and order  $r$  for the pressure. For example, there are three methods of type M1S4R2 and they are given by the tableaux in (6.34). In case  $m = 3$ , we can always make  $r$  equal to  $s$  with the approach of section 6.3.2, and any existing  $s$ -stage,  $s$ -th order method can be

used. In case  $r = 1$  we have the standard approach with  $p_{n+1} = \tilde{\phi}_s$ , which can be used with any method.

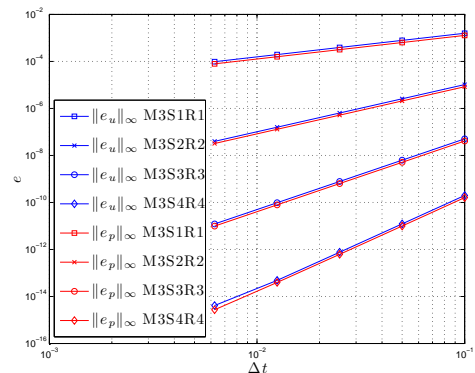
#### 6.4.1 Taylor-Green vortex

We use the Taylor-Green vortex from section 3.3.1 again, this time to investigate the temporal accuracy. We take a coarse mesh with  $20 \times 20$  volumes and vary the time step. The spatial error clearly overwhelms the temporal error and in order to compute the latter, we subtract the solution from a simulation with a small time step ( $\Delta t = 10^{-3}$ ), so that the spatial error is effectively eliminated.

The first test concerns periodic boundary conditions, such that the observations from section 6.3.3 apply and the methods are characterized as M<sub>3</sub>. Four  $s$ -stage,  $s$ th-order Runge-Kutta methods are tested, with  $s = 1, 2, 3, 4$ . For  $s = 1$  we take Forward Euler, for  $s = 2$  modified Euler (explicit trapezoidal, Heun's method), for  $s = 3$  Wray's method, and for  $s = 4$  the classical fourth-order method. In all cases the number of Poisson solves is the same as the number of stages. Figure 6.2a shows that with our current approach both pressure and velocity attain the classical order of convergence, whereas the standard method ( $p_{n+1} = \tilde{\phi}_s$ ) leads to only first order convergence of the pressure, see figure 6.2b. The velocity error is unaffected by the accuracy of the pressure. For this very smooth test case the error of higher order methods does not only converge faster upon time step refinement (as predicted by theory), but the magnitude of the error for the largest time step is also much smaller. In this particular example the two-stage, second-order method is stable and accurate enough and is to be preferred over the more expensive three- and four-stage methods.



(a) Standard method



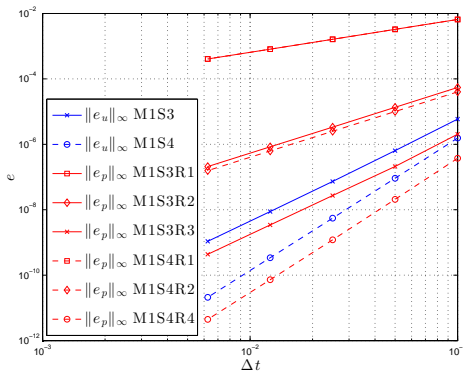
(b) New method for steady boundary conditions

Figure 6.2: Convergence of temporal error for Taylor-Green problem with periodic boundary conditions.

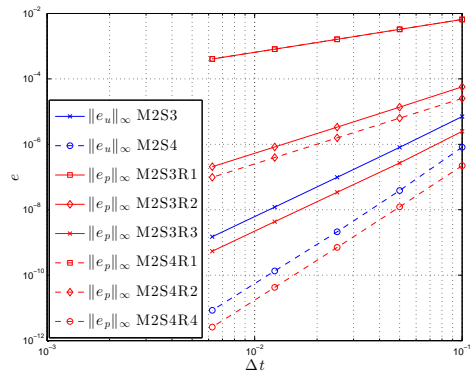
The second test concerns unsteady Dirichlet conditions with methods of type M1. For  $s = 1$  and  $s = 2$  only  $r = 1$  is possible, so we focus on  $s = 3$  and  $s = 4$  with  $r = 2$ . For M1S3R2 the only solution is (6.32) with (6.33) for the pressure, for M1S4R2 we take the first tableau in (6.34) (because it has the simplest coefficients) and (6.35) for the pressure.

The third test concerns unsteady Dirichlet conditions with methods of type M2. As for methods of type M1, we focus on  $s = 3$  and  $s = 4$ . For M2S3R2 we take Wray's method with (6.53) for the pressure and for M2S4R2 we take (6.55) with  $c_2 = \frac{1}{4}$  and (6.57) for the pressure.

Figures 6.3a and 6.3b then show the order of accuracy of the velocity and pressure for these two methods, in case of the 'standard' approach (R1), our approach (R2) and in case of an additional Poisson solve (R3 or R4). The additional Poisson solve can be performed because the explicit dependence of  $r_1$  on  $t$  is known so that  $\dot{r}_1(t)$  can be calculated. The velocity error is in all cases again independent of the particular approach for the pressure. Both methods M1 and M2 indeed lead to a second-order accurate pressure when the proper Butcher tableaux are chosen. The difference in accuracy between the results of M1 and M2 is small, but this depends on the test case under consideration. The computational effort is for both very similar. These second-order schemes greatly improve the accuracy with respect to the standard first-order approach without additional cost. The effort of an additional Poisson solve can only be justified in case higher-order accurate (third or fourth order) pressure solutions are required.



(a) New method, with single Butcher array for velocity and pressure (M1)



(b) New method, with reconstruction of instantaneous pressure values from time averages (M2)

Figure 6.3: Convergence of temporal error for Taylor-Green problem with unsteady Dirichlet boundary conditions.

6.4.2 An actuator disk in an unsteady inflow field

A practically relevant situation with a temporally varying inflow appears when simulating the flow of air through wind turbines operating in a turbulent atmospheric wind field. The wind turbine is modeled following the generalized actuator approach, introduced in section 1.4.1, and to be discussed in detail in chapter 8.

Here we study a simplified case of an actuator disk in a laminar flow. The domain is  $[0, 10] \times [-2, 2]$ , the Reynolds number is 100 and the thrust coefficient of the turbine is  $C_T = \frac{1}{2}$ . The actuator disk is located at  $x = 2$  and has unit length, see figure 6.4. On all boundaries, except the inflow boundary at  $x = 0$ , we prescribe outflow conditions (see e.g. [207]):

$$y = -2, 2 : \quad \frac{\partial u}{\partial y} = 0, \quad p - \frac{1}{\text{Re}} \frac{\partial v}{\partial y} = p_\infty, \quad (6.61)$$

$$x = 10 : \quad p - \frac{1}{\text{Re}} \frac{\partial u}{\partial x} = p_\infty, \quad \frac{\partial v}{\partial x} = 0. \quad (6.62)$$

For a verification study of the actuator disk in a laminar flow with steady inflow and these boundary conditions we refer to [145]. In the current test, the inflow conditions are given by:

$$x = 0 : \quad u_b(t) = \cos \alpha(t), \quad v_b(t) = \sin \alpha(t), \quad (6.63)$$

where  $\alpha(t) = \frac{\pi}{6} \sin(t/2)$ . This describes a time-varying inflow with constant magnitude but changing direction, see figure 6.5a.

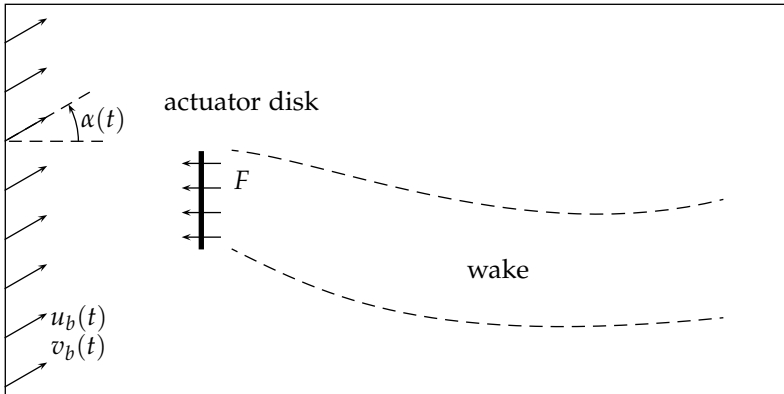


Figure 6.4: Actuator disk in an unsteady inflow field.

First we perform a simulation from  $t = 0$  to  $t = 4\pi$ , with a uniform mesh having  $200 \times 80$  volumes and 10,000 time steps, using the M2S4R4 method of equation (6.55), again with  $c_2 = \frac{1}{4}$ . We focus on methods of type M2, because they still allow for some freedom in the choice of the coefficients of the Butcher tableau, in contrast to methods of type M1. In figure 6.5b the normalized kinetic energy of the flow



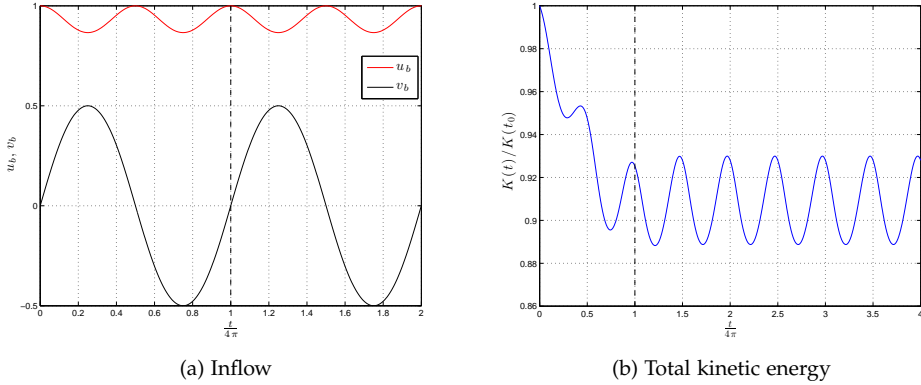


Figure 6.5: Inflow and total kinetic energy as a function of time.

(integrated over the entire domain) is shown, from which it can be concluded that the flow becomes periodic with period  $2\pi$  after approximately  $t = 4\pi$ . The velocity and pressure field at this time instant are shown in figures 6.6 and 6.7. The wake has been deflected downwards due to the inflow with negative  $v_b$  that was present from  $t = 2\pi$  to  $4\pi$ . The presence of the actuator disk is clearly seen in the pressure contours; they are discontinuous across the disk.

Due to the very small time step, these velocity and pressure fields have a negligible temporal error compared to the spatial error, and are therefore used to compute the temporal error in the velocity and pressure field for larger time steps. The resulting convergence of the velocity and pressure error is shown in figure 6.8, for methods (6.50) and (6.55). As before, we see that the velocity attains its classical order of accuracy, i.e., third order for the three-stage method, and fourth order for the four-stage method. The pressure can be computed to the same order as the velocity, but this requires an additional Poisson solve and an expression for  $\dot{r}_1(t)$ . Since  $\dot{r}_1(t)$  contains only the normal velocity component on the boundary, it is sufficient to derive the expression for  $\dot{u}_b(t)$ :

$$\dot{u}_b(t) = -\frac{\pi}{12} \sin(\alpha(t)) \cos(t/2). \quad (6.64)$$

On the other hand, the standard approach is only first order and starts with a large error at large time steps. Our proposed approach, corresponding to the lines M2S3R2 and M2S4R2, does not require any significant additional computational effort (no additional Poisson solve, no evaluation of  $\dot{r}_1(t)$ ), it clearly shows second-order accuracy and starts with a small error already at the largest time step considered. This time step,  $\Delta t = 4\pi/200$ , is the largest step for which stable solutions could be obtained. It is determined by the convective terms, showing the benefit of explicit Runge-Kutta methods for this test case.

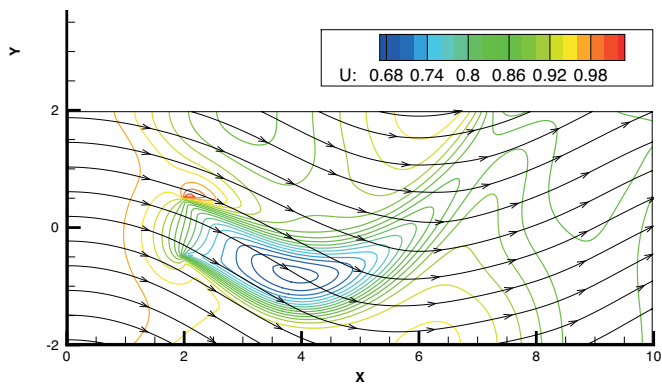


Figure 6.6: Streamlines and  $u$ -contour lines at  $t = 4\pi$ .

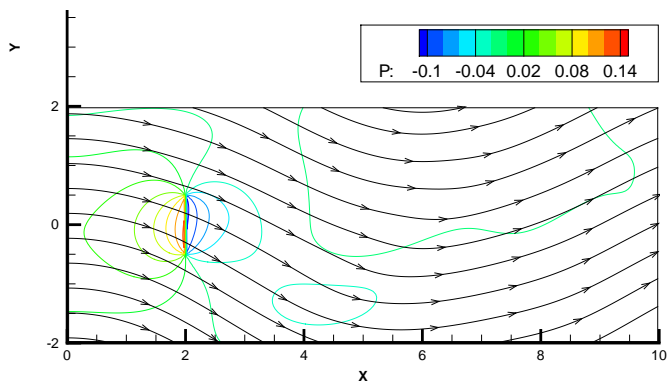


Figure 6.7: Streamlines and  $p$ -contour lines at  $t = 4\pi$ .

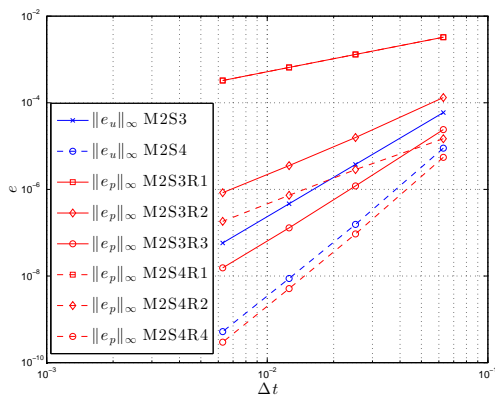


Figure 6.8: Velocity and pressure error at  $t = 4\pi$  for a selection of methods.

## 6.5 CONCLUSIONS

In this chapter we have analyzed the temporal order of accuracy of the velocity and pressure when explicit Runge-Kutta methods are applied to the incompressible Navier-Stokes equations. It is shown that the order of accuracy of the velocity is not affected by the differential-algebraic nature of the incompressible Navier-Stokes equations and is therefore the same as for non-stiff ordinary differential equations. However, if the semi-discrete equations involve time-dependent operators, then additional order conditions appear for orders higher than two. These conditions restrict three-stage, third-order methods to a one-parameter family of methods, to which the popular method of Wray does not belong. Four-stage, fourth-order methods for time-varying operators *do not exist*, and one has to resort to five stages to achieve fourth order.

In any case (time-dependent and time-independent operators) the pressure suffers from the problem that upon time stepping a time-average pressure is computed, instead of a point value. Therefore, achieving higher than first order accuracy for the pressure imposes additional conditions on the coefficients of the Runge-Kutta method compared to the classical order conditions. Fortunately, if the boundary conditions for the continuity equation are independent of time, then the pressure can be determined to the same order of accuracy as the velocity, without requiring an additional solution of a Poisson problem. However, if the boundary conditions for the continuity equation depend on time, then additional order conditions for the pressure appear. These are not satisfied by most existing explicit Runge-Kutta methods, so that the pressure is typically only first-order accurate in time. Using the same Butcher tableau for velocity and pressure, second-order accuracy can be achieved by only one three-stage, and only three four-stage methods.

A new approach is to reconstruct instantaneous pressure values from time-average values. We showed that this reconstruction, based on Lagrange polynomials, can be of the same order as the number of stages, but that the stage order of the method limits the accuracy of the pressure. These methods can be interpreted as having a different Butcher tableau for velocity and pressure, in contrast to the foregoing single-Butcher array approach. Three- and four-stage methods with second-order stage order were derived, leading to a much larger class of methods that have second-order accuracy for the pressure. Furthermore, a distinct advantage of this new class of methods is that they can be directly applied to implicit and implicit-explicit (IMEX) Runge-Kutta methods as well.

In all cases considered here third-order accuracy could not be obtained with a three- or four-stage method without resorting to an additional Poisson solve. Such an additional solve is not always straightforward in practical computations, because it requires the derivative of the boundary conditions for the continuity equation with respect to time. If the additional Poisson solve is to be performed anyway, the proposed second-order accurate methods can be used to provide an accurate and cheap initial guess for iterative methods (such as the conjugate gradient method) to solve this Poisson equation.

Runge-Kutta methods with more than four stages might perhaps lead to higher-order accurate pressures, but such methods are not very relevant from a practical point of view. For example, when considering five-stage methods (requiring five Poisson solves), to obtain third-order accuracy for the pressure, it would be better to employ an additional Poisson solve in a four-stage method, leading to fourth-order accuracy of the pressure (assuming that  $\dot{r}_1(t)$  is available). However, methods with more stages than the classical order of accuracy (e.g. a four-stage third-order method) can be of interest from a stability point of view. Such methods can possibly be used with less Poisson solves than the number of stages.

To conclude, we think that the ‘best’ explicit Runge-Kutta method for many incompressible Navier-Stokes problems is a three-stage method of type M2, that is third order for the velocity and second order for the pressure. It combines stability (includes the imaginary axis), sufficient accuracy (temporal error is in general sufficiently small) and flexibility ( $c_2$  can still be chosen, in contrast to the three-stage method of type M1). A fourth-order method might lead to unnecessarily accurate solutions for the velocity without improving the order of accuracy of the pressure. For time-dependent operators, the three-stage method derived in section 6.3.1.2 is to be preferred: it maintains third-order accuracy on time-varying meshes, and is second-order accurate for the pressure. Of course, the coefficients of a Runge-Kutta method can be chosen on other grounds than accuracy only, for example low storage, low dispersion or built-in error estimation with adaptive step-size control. Such arguments have not been considered in this chapter.

---

*With regard to the issue of secondary conservation, prior investigators have focused almost exclusively on the spatial discretization scheme. [128]*

*adapted  
from [146]*

## 7.1 INTRODUCTION

In the previous chapter we have analyzed explicit Runge-Kutta methods for the incompressible Navier-Stokes equations. In this chapter we focus on implicit Runge-Kutta methods that extend the energy-conservation property of the spatial discretization when marching in time. The advantages of energy conservation were discussed in section 2.1. Typically, time-integration methods used in conjunction with spatially energy-conserving methods are explicit methods, like explicit Runge-Kutta or Adams-Bashforth (AB) methods, possibly with an implicit Crank-Nicolson (CN) scheme for the diffusive term in case of wall-bounded flows. The divergence-free constraint is imposed with a fractional step approach, that decouples the treatment of velocity and pressure when marching in time. Examples are Wray's third order method (with or without CN) [116, 87, 181, 94], AB2 (with or without CN) [103, 86] or a generalization of AB2 [203]. The AB2 method is unstable for pure convection problems, and the explicit Runge-Kutta methods of order higher than 2 usually add a small amount of dissipation, because their amplification factors are smaller than unity inside the linear stability domain on the imaginary axis. In general such explicit methods are efficient if the time step is dictated by accuracy, and not by stability. In that case the errors associated with the spatial discretization are typically much larger than the errors associated with the temporal discretization [58].

Relatively little research has been performed on the use of energy-conserving methods for the Navier-Stokes equations. One example is the work of Ham et al. [65], who employ the implicit midpoint method and solve pressure and velocity coupledly. They find that the statistics of the DNS of turbulent channel flow are much less sensitive to the time step than results obtained by Choi and Moin [37] (who use a Crank-Nicolson scheme with a fractional step method), although the CFL number was still limited to approximately 4 in order to have solutions that were accurate enough. Mullen et al. [120] also apply the implicit midpoint method and indicate how to accelerate the solution of the non-linear system. It is found that the Crank-Nicolson method is not exactly energy-conserving, but oscillates around the initial energy. Simo et al. [168] recognized that it is possible to conserve energy in time (and therefore obtain non-linear stability) with a linear implicit scheme, since the time level of the convecting quantity does not affect the secondary conservation properties of the convected quantity. For the laminar flow over a circular cylinder

it is shown that the linear implicit scheme is as accurate as the non-linear implicit scheme. Mullen et al. [120] take a similar approach in their hybrid time integration, taking vorticity evaluated at the previous time level. Turek [193] also discusses such linear implicit schemes, but not in conjunction with energy conservation.

A drawback of these energy-conserving methods is that they are necessarily implicit, leading to a higher computational cost per time step as compared to explicit methods (namely the solution of a non-linear saddle point problem at each time step instead of a Poisson equation for the pressure). A common view is that the time step for the simulation of turbulent flows is dictated by accuracy, and not by (convective) stability, so that an implicit treatment of the convective terms is not efficient. This, however, depends on the specific flow problem [58]. For example, Verstappen et al. [200] argue that in many turbulent flows the time step imposed by the convective stability limit can be much smaller than the characteristic time-scale of the smallest resolved eddies. Similarly, Vreman [204] mentions that the time step determined by the stability restriction of the numerical scheme is considerably smaller than the shortest turbulent time-scale. Consequently, the truncation errors from the spatial discretization method are more important than truncation errors resulting from the discretization in time, when using explicit methods. This opens the perspective for efficient application of implicit methods. Turek [193] also suggests to use implicit methods, for both convection and diffusion, to allow for larger time steps.

Another potential drawback of energy-conserving methods is that they are not well-suited for stiff problems. Methods like implicit midpoint and Crank-Nicolson are not  $L$ -stable, meaning that they hardly damp stiff components (arising for example from initial and/or boundary conditions), and oscillatory solutions can result. One of the issues we address in this chapter is the construction of new methods that are both energy-conserving and  $L$ -stable.

Related to energy conservation is the time-reversibility property of the inviscid equations, since both properties are destroyed as soon as viscosity is introduced. These properties can therefore be used to assess if a discretization introduces numerical diffusion, as suggested by Duponcheel et al. [45], although in fact no fully reversible time integration methods are used in that work. A time-reversibility test was performed by Ham et al. [65], showing that the implicit midpoint method is time-reversible in the inviscid limit, until round-off errors contaminate the solution. However, questions such as 'are there energy-conserving methods which are not time-reversible?', 'are there time-reversible methods which are not energy-conserving?', and 'is one of these properties more important than the other for fluid simulations?' have apparently not yet been assessed for the incompressible Navier-Stokes equations.

In this chapter we will investigate the use of Runge-Kutta methods for high-order, energy-conserving time integration of the incompressible Navier-Stokes equations. Runge-Kutta methods can combine high-order with good stability and efficiency, they allow easy adaptive step size selection and are self-starting; multi-step methods, although less costly, do not have these properties, and it is questionable if they

can be energy-conserving [61]. We start with the general framework for Runge-Kutta methods for incompressible flows from chapter 5. We investigate higher order methods that are energy-conserving and/or time-reversible. Subsequently we propose a new class of additive Runge-Kutta methods that combine energy-conservation properties with  $L$ -stability. High-order accurate pressures are obtained with the techniques from chapter 6. Numerical experiments on different test cases are carried out to show the performance of the methods.

## 7.2 FORMULATION FOR IMPLICIT METHODS

Following equations (5.23)-(5.24) we write for the implicit Runge-Kutta method

$$U = u_n + \Delta t A F - \Delta t C^p G \phi, \quad (7.1)$$

with  $\phi$  defined by

$$C^p L \phi = A M F - \frac{r_1 - r_1(t_n)}{\Delta t}, \quad (7.2)$$

and  $C^p = \text{diag}(c_1, \dots, c_s)$ . This gives  $\phi = (C^p)^{-1} A \psi$ . We have used the following notation:

$$U = \begin{pmatrix} U_1 \\ \vdots \\ U_s \end{pmatrix}, \quad F = \begin{pmatrix} F_1 \\ \vdots \\ F_s \end{pmatrix}, \quad \phi = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_s \end{pmatrix}, \quad r_1 = \begin{pmatrix} r_1(t_1) \\ \vdots \\ r_1(t_s) \end{pmatrix}, \quad (7.3)$$

and we have written  $M$  instead of  $I_s \otimes M$ ,  $G$  instead of  $I_s \otimes G$ , etc. The size of these matrices should be clear from the context. By using  $\phi$  instead of  $\psi$  equation (7.2) resembles equation (5.7), and the total gradient matrix is block-diagonal. In terms of  $\phi$  a single Lagrange multiplier  $\phi_i$  makes each velocity field  $U_i$  divergence-free, while in terms of  $\psi$  a sum of Lagrange multipliers would be required.

We note that, in contrast to the explicit methods discussed in chapter 6, implicit methods can have  $c_1 = 0$  (e.g. methods based on Lobatto quadrature) so that  $C^p$  is singular. In that case we define a new pressure-like variable  $\rho = C^p \phi$  and equations (7.1)-(7.2) can be written without the  $c$  coefficients.

Once the stage values  $U_i$  are determined,  $u_{n+1}$  follows by introducing the Lagrange multiplier  $\phi_{n+1}$  and rewriting equation (5.13) as the following equivalent uncoupled system:

$$\hat{u}_{n+1} = u_n + \Delta t \sum_{i=1}^s b_i F_i, \quad (7.4)$$

$$L \phi_{n+1} = \sum_{i=1}^s (b_i M F_i) - \frac{r_1(t_{n+1}) - r_1(t_n)}{\Delta t}, \quad (7.5)$$

$$u_{n+1} = \hat{u}_{n+1} - \Delta t G \phi_{n+1}. \quad (7.6)$$

This projection of  $\hat{u}_{n+1}$  onto the space of divergence-free velocity fields is known as a projected Runge-Kutta method [7, 63]. It leads to the ‘classical’ order of accuracy of the Runge-Kutta method (the accuracy for ODEs), as will be detailed in the next section. An alternative to this projection step is described by Jay [79], but requires the solution of a larger system of coupled non-linear equations, and is therefore not considered here.

There are two important cases for which the additional projection step can be avoided, making the solution of the Poisson equation (7.5) unnecessary:

- If the implicit method is stiffly accurate, i.e.,  $a_{si} = b_i$  and  $c_s = 1$ , then  $u_{n+1} = U_s$ ,  $\phi_{n+1} = \phi_s$  and the last projection step is redundant. This is for example the case for the Radau IIA, Lobatto IIIA and IIIC methods.
- If  $A$  is invertible and the boundary conditions  $r_1(t)$  are not a function of time,  $\phi_{n+1}$  can be expressed as a linear combination of  $\phi$ 's, so that a Poisson solve for  $\phi_{n+1}$  (equation (7.5)) is not necessary. The equation for  $u_{n+1}$  then reads

$$u_{n+1} = u_n + b^T A^{-1}(U - u_n), \quad (7.7)$$

which satisfies the divergence-free constraint. In practice we therefore use formulation (7.7) instead of (7.4) and in case of unsteady boundary conditions equation (7.7) should be read for  $\hat{u}_{n+1}$  and the additional projection is performed. This formulation also saves an additional evaluation of  $F(U_i)$  when the nonlinear iterative process has converged.

Well-known examples of implicit Runge-Kutta methods are methods based on high-order quadrature formulas (like the Gauss, Lobatto and Radau methods, see e.g. [41, 63]), and (single) diagonally implicit Runge-Kutta ((S)DIRK) methods, which have zeros in the upper triangle of the Butcher tableau. Each of these methods has certain advantages for integrating the incompressible Navier-Stokes equations. The Gauss methods, for example, have the highest possible order ( $p = 2s$ ) given a certain number of stages; a well-known member is the implicit midpoint method ( $s = 1$ ,  $p = 2$ ). On the other hand, certain Lobatto and Radau methods have the advantage that they are stiffly accurate ( $a_{si} = b_i$ ), which is an important property for differential-algebraic equations [7, 63]: (i) it avoids the additional projection step mentioned above, and (ii) it leads to  $L$ -stable methods (provided that  $A$  is non-singular).  $L$ -stable methods are robust, give stable results for problems at any Reynolds number, and damp perturbations that originate from rough initial or boundary conditions. The (S)DIRK methods have the advantage that the stages of the Runge-Kutta method can be solved sequentially. The advantage of energy-conserving (and time-reversible) methods were already outlined in section 2.1. In our opinion, an ideal method would combine all these properties, resulting in a high-order, energy-conserving, time-reversible, stiffly-accurate/ $L$ -stable (S)DIRK scheme.

In the next sections we formulate the order conditions for implicit Runge-Kutta methods and the conditions for energy conservation, time-reversibility and  $L$ -sta-



bility. The  $W$ -transformation of Hairer and Wanner [63] will appear in this context and is therefore summarized in appendix B.1.

### 7.3 ORDER CONSIDERATIONS

The order conditions for the *local error* of Runge-Kutta methods can be obtained by using Butcher's analysis based on trees as introduced in section 6.2. For high-order methods, say order higher than 4, it becomes cumbersome to derive methods that satisfy these (non-linear) conditions. For implicit Runge-Kutta methods one can fortunately make use of the simplifying conditions (5.16)-(5.18) and a theorem of Butcher [24]: If  $\mathcal{A}$  satisfies  $B(p)$ ,  $C(\eta)$ ,  $D(\zeta)$  with  $p \leq \eta + \zeta + 1$  and  $p \leq 2\eta + 2$ , then the method is of order  $p$ . However, there are cases in which this 'classical' or 'ODE' order of accuracy is not obtained. We group these in three categories.

Firstly, in case of very stiff ODEs order reduction occurs if a method is not stiffly accurate [63, 25, 8]. The stage order  $\eta$  of the method then determines the convergence of the error, instead of the classical order. The value of the linear stability function  $R(z)$  at infinity determines the convergence: if  $R(\infty) = 0$  ( $L$ -stable methods) or  $R(\infty) = -1$  (e.g. Gauss with  $s$  odd), the global error behaves as  $\mathcal{O}(\Delta t^{\eta+1})$ , whereas for methods with  $R(\infty) = 1$  (e.g. Gauss with  $s$  even) the global error behaves as  $\mathcal{O}(\Delta t^\eta)$  [25]. This effect will be illustrated in section 7.8.3.

Secondly, order reduction can occur if index 2 DAEs are considered instead of ODEs. In some sense this is related to the order reduction for very stiff ODEs, by regarding the divergence-free constraint (3.1) as an infinitely stiff equation. For explicit methods we showed in chapter 6 that no order reduction occurs for the velocity, up to and including order 4, if the mesh is not moving in time. For implicit methods such an analysis is not necessary; no order reduction occurs according to a number of theorems for which we refer to Hairer and Wanner [63] (Theorems 4.12, 5.10, 5.13). On the other hand, the pressure suffers in general from order reduction, like in the explicit case. This will be detailed in section 7.6. Convergence of the *global error* follows from Theorems 4.4 and 4.9 in [60], which require invertibility of the  $A$ -matrix and consistency of the initial values.

We note that recently implicit Runge-Kutta methods have been applied to the incompressible Navier-Stokes equations by Montlaur et al. [115]. In that work the Gauss methods are dismissed as useful 'because they present higher order reduction [than Radau] when applied to DAEs with respect to ODEs'. *This is a misconception: the Gauss methods keep their ODE accuracy when projected Runge-Kutta methods are used* [63]. In section 7.8 this will be supported by numerical experiments.

Thirdly, order reduction can occur upon simultaneously refining mesh and time step combined with time-dependent inflow boundary conditions, due to so-called 'space-time errors'. This phenomenon will not be investigated here (for the same reasons mentioned in section 6.2.6).

7.4 ENERGY CONSERVATION AND TIME REVERSIBILITY

7.4.1 Energy conservation

In this section we derive conditions for the coefficients of the Butcher tableau such that the resulting Runge-Kutta method is energy-conserving. In all cases we consider inviscid flow ( $\nu = 0$ ) and periodic boundary conditions. The monotonicity property of the fully continuous equations, equation (2.23), is shared by the semi-discrete system of equations (5.1)-(5.2):

$$\|u(t)\| \leq \|u(t_0)\|, \quad t \geq t_0, \tag{7.8}$$

due to properties (4.76)-(4.78). Equality holds in case  $\nu = 0$ . For the fully discrete equations, this monotonicity or energy-conservation property is investigated by taking the inner product of the fully discrete momentum equation, equation (5.13), with itself. First we write equation (5.13) with  $r_1 = 0$  as

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^s b_i P F_i, \tag{7.9}$$

where  $P = I - GL^{-1}M$  is a projection operator that projects velocity fields onto the space of divergence-free velocity fields. Taking the norm of  $u_{n+1}$  and substituting the equation for the stages leads to

$$\|u_{n+1}\|^2 = \|u_n\|^2 + 2\Delta t \sum_{i=1}^s b_i (P F_i, U_i) - \Delta t^2 \sum_{i,j=1}^s (b_i a_{ij} + b_j a_{ji} - b_i b_j) (P F_i, P F_j). \tag{7.10}$$

*This is one of the key results of this chapter.* Conservation of energy requires that the two last terms vanish. The term  $(P F_i, U_i)$ , is zero if  $(C(U_i), U_i, U_i) = 0$  and  $(G p_i, U_i) = -(p_i, M U_i) = 0$ . For the last term to be zero the coefficients  $A$  and  $b$  from the Runge-Kutta tableau should satisfy

$$e_{ij} \equiv b_i a_{ij} + b_j a_{ji} - b_i b_j = 0. \tag{7.11}$$

Note that we define the energy ‘pointwise’ in time; at  $n, n + 1$ , etc. Other definitions, such as the average value over a time step, can also be employed and might lead to different requirements on the coefficients of the Runge-Kutta method - this is the subject of future work.

In the literature Runge-Kutta methods that satisfy  $E = (e_{ij}) = 0$  are known as *symplectic*, since this condition is obtained when applying a Runge-Kutta method to symplectic systems and requiring that the symplectic structure is preserved. In general, symplectic methods are *not* conserving energy, but in the case of the incompressible Navier-Stokes equations the energy is a quadratic invariant, and every Runge-Kutta method that preserves quadratic invariants is a symplectic method

[61]. To summarize: iff a method (i) satisfies equation (7.11), (ii) has a skew-symmetric convective operator ( $(C(U_i, U_i), U_i) = 0$ ), and (iii) has a pressure gradient that does not contribute to the energy ( $(Gp_i, U_i) = -(p_i, MU_i) = 0$ ), then the total kinetic energy is exactly conserved for inviscid flows:

$$\|u_{n+1}\|^2 = \|u_n\|^2. \quad (7.12)$$

For viscous flows the second term in (7.10) also contains the viscous contribution ( $DU_i, U_i) \leq 0$ , and then equation (7.10) reads (for  $E = 0$ )

$$\|u_{n+1}\|^2 = \|u_n\|^2 + 2\Delta t \sum_{i=1}^s b_i (DU_i, U_i), \quad (7.13)$$

assuming  $MU_i = 0$ . This is the discrete equivalent of equation (2.22).

Energy-conserving methods ( $E = 0$ ) are a particular example of algebraically stable methods [63]: methods having a nonnegative definite  $E$  matrix and nonnegative  $b$  coefficients. Such methods lead to energy-stable time-integration, for *any time step*.

The energy-conservation condition (7.11) cannot be satisfied by explicit methods. Implicit energy-conserving Runge-Kutta methods can be constructed with the  $W$ -transformation of Hairer and Wanner [63], explained in appendix B.1, in particular condition (B.6). This transformation shows that the Gauss methods are the methods with the highest possible order ( $p = 2s$ ) that satisfy (7.11). For one order lower,  $p = 2s - 1$ , a one-parameter family of methods results, of which the Radau IB and IIB methods [187] are two examples. For  $p = 2s - 2$  a two-parameter family of methods results, which includes the Lobatto IIIE methods (sometimes denoted by IIID) [33, 99]. Some examples are shown in table 7.1. Of particular interest are diagonally-implicit energy-conserving methods, since the stages of the Runge-Kutta method are uncoupled. For  $s = p = 2$  there is a one-parameter family which reads

$$\begin{array}{c|cc} c_1 & c_1 & 0 \\ c_1 + \frac{1}{2} & 2c_1 & \frac{1}{2} - c_1 \\ \hline & 2c_1 & 1 - 2c_1 \end{array} \quad (7.14)$$

with  $c_1 \in (0, \frac{1}{2})$  to keep the  $b$  coefficients positive. This family is recognized as the implicit midpoint applied twice over intervals of size  $2c_1$  and  $1 - 2c_1$ , and is limited to second order; we call it DIRK E. As is indicated in appendix B.2, it is not possible to construct algebraically stable or  $A$ -stable symplectic DIRK methods with  $p > 2$ .

An alternative for constructing energy-conserving methods with the  $W$ -transformation is to use the following transformation [188]

$$\mathcal{A}^* = \mathcal{E}(\mathcal{A}), \quad (7.15)$$

where  $\mathcal{E}$  is such that

$$a_{ij}^* = \frac{1}{2} \left( a_{ij} + b_j \left( 1 - \frac{a_{ji}}{b_i} \right) \right), \quad b_i^* = b_i, \quad c_i^* = \sum_j a_{ij}^*. \quad (7.16)$$

In this way an energy-conserving method can be obtained from *any* existing Runge-Kutta method. This does not necessarily lead to useful methods, because  $\mathcal{E}$  is a non-linear transformation, which in general changes the order of the method. However, in case the original method  $\mathcal{A}$  satisfies simplifying conditions  $B(p)$ ,  $C(\eta)$  and  $D(\zeta)$ , then the transformed method  $\mathcal{E}(\mathcal{A})$  satisfies  $B(p)$ ,  $C(\xi)$  and  $D(\xi)$ , with  $\xi = \min(\eta, \zeta)$  [188]. The transformation  $\mathcal{E}$  relates certain quadrature methods. For example, the Radau IIB methods follow by applying  $\mathcal{E}$  on the Radau IIA methods and the Lobatto IIIE methods follow from the Lobatto IIIC methods. These relations will be used to construct additive Runge-Kutta methods in section 7.5.

In table 7.3 we list a number of energy-conserving methods, together with other properties that will be discussed in subsequent sections.

<table style="border-collapse: collapse; margin: auto;"> <tr> <td style="border-right: 1px solid black; padding: 5px;"><math>\frac{1}{2}</math></td> <td style="padding: 5px;"><math>\frac{1}{2}</math></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;"></td> <td style="padding: 5px;">1</td> </tr> </table> <p>(a) Gauss, <math>p = 2</math></p>	$\frac{1}{2}$	$\frac{1}{2}$		1	<table style="border-collapse: collapse; margin: auto;"> <tr> <td style="border-right: 1px solid black; padding: 5px;"><math>\frac{1}{2} - \frac{\sqrt{3}}{6}</math></td> <td style="padding: 5px;"><math>\frac{1}{4}</math></td> <td style="padding: 5px;"><math>\frac{1}{4} - \frac{\sqrt{3}}{6}</math></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;"><math>\frac{1}{2} + \frac{\sqrt{3}}{6}</math></td> <td style="padding: 5px;"><math>\frac{1}{4} + \frac{\sqrt{3}}{6}</math></td> <td style="padding: 5px;"><math>\frac{1}{4}</math></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;"></td> <td style="padding: 5px;"><math>\frac{1}{2}</math></td> <td style="padding: 5px;"><math>\frac{1}{2}</math></td> </tr> </table> <p>(b) Gauss, <math>p = 4</math></p>	$\frac{1}{2} - \frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4} - \frac{\sqrt{3}}{6}$	$\frac{1}{2} + \frac{\sqrt{3}}{6}$	$\frac{1}{4} + \frac{\sqrt{3}}{6}$	$\frac{1}{4}$		$\frac{1}{2}$	$\frac{1}{2}$												
$\frac{1}{2}$	$\frac{1}{2}$																									
	1																									
$\frac{1}{2} - \frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4} - \frac{\sqrt{3}}{6}$																								
$\frac{1}{2} + \frac{\sqrt{3}}{6}$	$\frac{1}{4} + \frac{\sqrt{3}}{6}$	$\frac{1}{4}$																								
	$\frac{1}{2}$	$\frac{1}{2}$																								
<table style="border-collapse: collapse; margin: auto;"> <tr> <td style="border-right: 1px solid black; padding: 5px;"><math>\frac{1}{3}</math></td> <td style="padding: 5px;"><math>\frac{3}{8}</math></td> <td style="padding: 5px;"><math>-\frac{1}{24}</math></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">1</td> <td style="padding: 5px;"><math>\frac{7}{8}</math></td> <td style="padding: 5px;"><math>\frac{1}{8}</math></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;"></td> <td style="padding: 5px;"><math>\frac{3}{4}</math></td> <td style="padding: 5px;"><math>\frac{1}{4}</math></td> </tr> </table> <p>(c) Radau IIB, <math>p = 3</math></p>	$\frac{1}{3}$	$\frac{3}{8}$	$-\frac{1}{24}$	1	$\frac{7}{8}$	$\frac{1}{8}$		$\frac{3}{4}$	$\frac{1}{4}$	<table style="border-collapse: collapse; margin: auto;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">0</td> <td style="padding: 5px;"><math>\frac{1}{12}</math></td> <td style="padding: 5px;"><math>-\frac{1}{6}</math></td> <td style="padding: 5px;"><math>\frac{1}{12}</math></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;"><math>\frac{1}{2}</math></td> <td style="padding: 5px;"><math>\frac{5}{24}</math></td> <td style="padding: 5px;"><math>\frac{1}{3}</math></td> <td style="padding: 5px;"><math>-\frac{1}{24}</math></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">1</td> <td style="padding: 5px;"><math>\frac{1}{12}</math></td> <td style="padding: 5px;"><math>\frac{5}{6}</math></td> <td style="padding: 5px;"><math>\frac{1}{12}</math></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;"></td> <td style="padding: 5px;"><math>\frac{1}{6}</math></td> <td style="padding: 5px;"><math>\frac{2}{3}</math></td> <td style="padding: 5px;"><math>\frac{1}{6}</math></td> </tr> </table> <p>(d) Lobatto IIIE, <math>p = 4</math></p>	0	$\frac{1}{12}$	$-\frac{1}{6}$	$\frac{1}{12}$	$\frac{1}{2}$	$\frac{5}{24}$	$\frac{1}{3}$	$-\frac{1}{24}$	1	$\frac{1}{12}$	$\frac{5}{6}$	$\frac{1}{12}$		$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$
$\frac{1}{3}$	$\frac{3}{8}$	$-\frac{1}{24}$																								
1	$\frac{7}{8}$	$\frac{1}{8}$																								
	$\frac{3}{4}$	$\frac{1}{4}$																								
0	$\frac{1}{12}$	$-\frac{1}{6}$	$\frac{1}{12}$																							
$\frac{1}{2}$	$\frac{5}{24}$	$\frac{1}{3}$	$-\frac{1}{24}$																							
1	$\frac{1}{12}$	$\frac{5}{6}$	$\frac{1}{12}$																							
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$																							

Table 7.1: Examples of energy-conserving methods.

### 7.4.2 Time reversibility

Time reversibility is investigated by changing  $u_{n+1}$  to  $-u_n$ ,  $u_n$  to  $-u_{n+1}$  and  $U_i = u_{n+c_i}$  to  $-u_{n+1-c_i}$  in equations (5.12)-(5.13) and checking if the resulting equations are equivalent to the original ones. This leads to the following conditions [206, 61]:

$$a_{ij} + a_{s+1-i, s+1-j} = b_j, \quad (7.17)$$

$$b_i = b_{s+1-i}, \quad (7.18)$$

$$c_i = 1 - c_{s+1-i}. \quad (7.19)$$

Runge-Kutta methods satisfying these conditions are called *symmetric*. In terms of the permutation matrix  $\hat{P}$  they are written as  $A + \hat{P}A\hat{P} = eb^T$ ,  $\hat{P}b = b$ ,  $\hat{P}c = e - c$ , where  $e = (1, \dots, 1)^T$ . Like the symplecticity condition, explicit Runge-Kutta methods cannot satisfy the symmetry conditions.

Implicit methods, on the other hand, can satisfy the symmetry conditions. For  $s = 1$  there is one solution to these conditions, being the 1-stage Gauss method (better known as the implicit midpoint method), see table 7.1a. For  $s = 2$  condition (7.18) fixes the  $b$ -coefficients:  $b_1 = b_2 = \frac{1}{2}$ . Examples are the Gauss, Lobatto IIIA, IIIB and IIIE methods, shown in tables 7.1a-7.1b, table 7.1d and 7.2a-7.2b. The 2-stage Lobatto IIIA method is better known as the trapezoidal rule or Crank-Nicolson method. General (higher order) symmetric methods can be constructed with the  $W$ -transformation in a similar way as the energy-conserving methods, now by satisfying condition (B.8). Table 7.3 lists for some methods whether they are symmetric or not.

0	0	0	0	$\frac{1}{6}$	$-\frac{1}{6}$	0
1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$	0
	$\frac{1}{2}$	$\frac{1}{2}$		$\frac{1}{6}$	$\frac{5}{6}$	0
(a) Lobatto IIIA, $p = 2$				$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Table 7.2: Examples of time-reversible methods.

The symmetry conditions do not imply stability, so it is of interest which symmetric methods are algebraically stable, i.e., have a nonnegative definite  $E$  matrix and  $b \geq 0$ . By using the  $X$ -matrix, introduced in appendix B.1, algebraic stability requires that

$$Y = X + X^T - e_1 e_1^T \quad \text{is nonnegative definite.} \quad (7.20)$$

From the  $W$ -transformation we know that symmetric methods have an  $X$ -matrix with a zero diagonal (except for  $x_{1,1} = \frac{1}{2}$ ). As a result  $Y$  is a symmetric matrix with a zero diagonal, and consequently it cannot be nonnegative definite, except if it is completely zero. Since  $Y = W^T E W$  this implies that *of all symmetric methods, only the energy-conserving ( $E = 0$ ) ones are algebraically stable*. A similar conclusion was drawn in [45], where it was stated that energy conservation is a practical requirement for schemes to be time-reversible. However, algebraic stability is a strong stability requirement and in many situations  $A$ -stability suffices, so this ‘practical’ requirement might be too restrictive. An important example of an  $A$ -stable time-reversible method which is not energy-conserving is the Crank-Nicolson method (member of the Lobatto IIIA family). In fact this method is conserving another quantity (see e.g. [74]),

$$\|u_{n+1}\|^2 + \frac{1}{4}\Delta t^2 \|PF_{n+1}\|^2 = \|u_n\|^2 + \frac{1}{4}\Delta t^2 \|PF_n\|^2, \quad (7.21)$$

with  $PF_n = F_n - GL^{-1}MF_n = F_n - Gp_n$  (and similarly for  $n + 1$ ). This statement is however not as strong as energy conservation, because  $\|u_{n+1}\|^2$  can grow in time due to the contribution of  $\|PF_n\|^2$ .

Type	Energy-cons.	Time-rev.	Stiffly acc.	$L$ -stable	Alg. stable	Order	$\eta$	$\zeta$
Gauss	Y	Y	N	N	Y	$2s$	$s$	$s$
Rad. IIA	N	N	Y	Y	Y	$2s - 1$	$s$	$s - 1$
Rad. IIB	Y	N	N	N	Y	$2s - 1$	$s - 1$	$s - 1$
Lob. IIIA	N	Y	Y	N	N	$2s - 2$	$s$	$s - 2$
Lob. IIIB	N	Y	N	N	N	$2s - 2$	$s - 2$	$s$
Lob. IIIC	N	N	Y	Y	Y	$2s - 2$	$s - 1$	$s - 1$
Lob. IIIE(D)	Y	Y	N	N	Y	$2s - 2$	$s - 1$	$s - 1$
DIRK E	Y	Y	N	N	Y	2	1	1

Table 7.3: Properties of implicit Runge-Kutta methods investigated in this work.

### 7.4.3 Other properties

In the previous sections we derived the conditions for energy conservation and time reversibility and we mentioned some high-order methods that satisfy these conditions, see table 7.3. Other desirable properties, such as  $L$ -stability and algebraic stability have also been listed in this table. The table reveals that all energy-conserving or time-reversible method are not stiffly accurate or  $L$ -stable (with the exception of the Lobatto IIIA methods which have singular  $A$ ). This is not a coincidence. It can be shown that energy-conserving and time-reversible methods have  $R = 1$  on the imaginary axis of the stability domain by applying them on a linear test equation. Since the stability function  $R(z)$  is a rational function, it attains the same value when approaching infinity on either the negative real axis or the imaginary axis, and consequently  $|R(\infty)| = 1$ . Energy-conserving or time-reversible methods can therefore not satisfy the condition  $R(\infty) = 0$ , necessary for  $L$ -stability. An alternative way of seeing this, which provides insight in how to construct  $L$ -stable methods, is to write the stability function as

$$R(z) = \frac{\text{Det}(I - zQ)}{\text{Det}(I - zA)}, \tag{7.22}$$

where  $Q = A - eb^T$ .  $L$ -stable methods require  $Q$  to be singular. Energy-conserving and time-reversible methods have  $Q = -B^{-1}A^TB$  and  $Q = -\hat{P}A\hat{P}$  (note  $B = \text{diag}(b_1, \dots, b_s)$ ), respectively, meaning that in both cases the eigenvalues of  $Q$  are  $-1 \times$  the eigenvalues of  $A$ . The degree of the numerator and denominator of  $R(z)$  are therefore the same, so such methods cannot be  $L$ -stable.

This negative result leads us to consider *additive* Runge-Kutta methods, where the convective and diffusive terms are treated with different tableaux, in order to arrive at methods that possess energy-conservation, time-reversibility and  $L$ -stability.

## 7.5 NEW ADDITIVE RUNGE-KUTTA METHODS

The conditions for energy conservation and time reversibility are derived by considering time integration of the non-linear convective terms. As such, it is not surprising that these methods do not have ideal properties for integrating the diffusive terms. It seems therefore logical to take two different Runge-Kutta methods for the convective and the diffusive terms. We propose to use *additive Runge-Kutta methods* (ARK), a class of methods that includes the well-known implicit-explicit (IMEX) Runge-Kutta methods [9]. In implicit-explicit methods stiff terms (such as diffusion terms) are handled implicitly, whereas other terms (such as non-linear convective terms) are handled explicitly, both with a different Butcher tableau. We keep the idea of two different Butcher tableaux,  $\mathcal{A}$  and  $\hat{\mathcal{A}}$ , but we take all terms implicitly:

$$U_i = u_n + \Delta t \sum_{j=1}^s (a_{ij}PF_j + \hat{a}_{ij}P\hat{F}_j) + GL^{-1}(r_1(t_i) - r_1(t_n)), \quad (7.23)$$

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^s (b_iPF_i + \hat{b}_iP\hat{F}_i) + GL^{-1}(r_1(t_{n+1}) - r_1(t_n)). \quad (7.24)$$

The diffusive terms are denoted by  $F$  and the convective terms by  $\hat{F}$ . We require that the Butcher tableau for the convective terms,  $\hat{\mathcal{A}}$ , is energy-conserving. The Butcher tableau for the diffusive terms,  $\mathcal{A}$ , is chosen such that an  $L$ -stable method results, so that stiff problems are damped sufficiently. A natural way to obtain such properties is to use an algebraically stable and stiffly accurate method for  $\mathcal{A}$  and its energy-transformed counterpart  $\hat{\mathcal{A}} = \mathcal{E}(\mathcal{A})$  for the convective terms. From an implementation point of view, the solution of equations (7.23)-(7.24) is not more expensive than for the 'standard' Runge-Kutta method (which is a special case of an ARK with  $A = \hat{A}$ ), except that in case of steady boundary conditions for the continuity equation we cannot employ formulation (7.7); we always need an additional Poisson solve to make  $u_{n+1}$  divergence-free.

In the next sections we will investigate the order conditions and stability properties of such ARK methods, and subsequently we will propose three new classes of methods: one based on Radau quadrature, one based on Lobatto quadrature, and one which is of DIRK type (see table 7.4).

### 7.5.1 Order conditions

The transformation  $\mathcal{E}$  leaves the  $b$ -coefficients unchanged, leading to  $b = \hat{b}$ . It is also desirable that the  $c$ -coefficients remain unchanged. Firstly, having  $b = \hat{b}$  and  $c = \hat{c}$

means that no additional order conditions (so-called coupling conditions) appear for orders less than four [85, 123], and for higher orders the number of additional order conditions is much smaller than for the case  $c \neq \hat{c}$  and  $b \neq \hat{b}$ . Secondly,  $c = \hat{c}$  is required to evaluate the divergence-free constraint  $MU_i = r_1(t_i)$ . We use the following theorem to show when  $c = \hat{c}$ .

**Theorem 7.5.1.** *If a Runge-Kutta method is transformed into an energy-conserving method according to (7.16), then the  $c$ -coefficients remain unchanged if the original method satisfies  $D(1)$ .*

*Proof.* Let the original method be given by  $\mathcal{A}$ , and the transformed method by  $\hat{\mathcal{A}} = \mathcal{E}(\mathcal{A})$ . Then we have

$$\hat{c}_i = \sum_{j=1}^s \hat{a}_{ij} = \frac{1}{2} \sum_{j=1}^s a_{ij} + \frac{1}{2} \sum_{j=1}^s b_j \left(1 - \frac{a_{ji}}{b_i}\right) = \frac{1}{2}c_i + \frac{1}{2} - \frac{1}{2} \frac{1}{b_i} \sum_{j=1}^s b_j a_{ji}, \quad (7.25)$$

which can be written as

$$\sum_{j=1}^s b_j a_{ji} = b_i(1 + c_i - 2\hat{c}_i). \quad (7.26)$$

Condition  $D(1)$  can be written as

$$\sum_{j=1}^s b_j a_{ji} = b_i(1 - c_i). \quad (7.27)$$

Therefore, if a method satisfies both (7.26) and (7.27), then  $c_i - 2\hat{c}_i = -c_i$ , so  $\hat{c}_i = c_i$ .  $\square$

By considering an additive Runge-Kutta method as a partitioned Runge-Kutta method (see [9]), the order of the pair is [63, 188]

$$\min(p, 2\bar{\zeta} + 1), \quad (7.28)$$

where  $\bar{\zeta} = \min(\eta, \zeta)$ , and  $\eta$  and  $\zeta$  correspond to method  $\mathcal{A}$ . It remains to be proven that the order of the additive Runge-Kutta method is not affected when the method is applied to DAEs instead of ODEs. This seems to be an open question, but theoretical [78] and numerical results, e.g. [94], indicate that, like for the standard Runge-Kutta methods, no order reduction occurs for the velocity in case of the incompressible Navier-Stokes equations. We will also confirm this with numerical experiments in section 7.8.

### 7.5.2 Stability

The algebraic stability properties of this new class of methods are investigated by assessing the energy conservation properties of the method, similar to equation (7.10).



Although both  $\mathcal{A}$  and  $\hat{\mathcal{A}}$  are chosen to be algebraically stable, their combination in an ARK method requires additional conditions to be satisfied. The expression for  $\|u_{n+1}\|^2$  reads, after employing equations (7.23)-(7.24) with  $b = \hat{b}$ :

$$\|u_{n+1}\|^2 = \|u_n\|^2 + 2\Delta t \sum_{i=1}^s b_i ((U_i, PF_i) + (U_i, P\hat{F}_i)) - \Delta t^2 \left( \sum_{i,j=1}^s e_{ij}(PF_i, PF_j) + \sum_{i,j=1}^s \hat{e}_{ij}(P\hat{F}_i, P\hat{F}_j) + 2 \sum_{i,j=1}^s \tilde{e}_{ij}(PF_i, P\hat{F}_j) \right), \quad (7.29)$$

where  $e_{ij}$  is given by (7.11),  $\hat{e}_{ij}$  is given by (7.11) with  $a$  replaced by  $\hat{a}$ , and  $\tilde{e}_{ij}$  is given by

$$\tilde{e}_{ij} = b_i \hat{a}_{ij} + b_j a_{ji} - b_i b_j. \quad (7.30)$$

Since  $\hat{\mathcal{A}}$  is energy-conserving we have  $\hat{E} = 0$ . In appendix B.3.1 we prove that as a result the additive Runge-Kutta method cannot be algebraically stable - even though the two constituents that form the method are in itself algebraically stable.

We therefore investigate a weaker (linear) stability concept by applying an additive Runge-Kutta method to the linear scalar test equation [9]

$$\dot{u} = (\lambda + \hat{\lambda})u, \quad (7.31)$$

where  $\lambda$  is an eigenvalue of the diffusion operator ( $\lambda \in \mathbb{R}$ ,  $\lambda \leq 0$ ), and  $\hat{\lambda}$  an eigenvalue of the (linearized) convection operator ( $\hat{\lambda} \in i\mathbb{R}$ ). When an additive Runge-Kutta method is applied to (7.31), the expression for the amplification factor  $R$  becomes [85, 29]:

$$R(z, \hat{z}) \equiv \frac{u_{n+1}}{u_n} = \frac{\text{Det}(I - zA - \hat{z}\hat{A} + (z + \hat{z})eb^T)}{\text{Det}(I - zA - \hat{z}\hat{A})} = \frac{N(z, \hat{z})}{D(z, \hat{z})}, \quad (7.32)$$

where  $z = \lambda\Delta t$ ,  $\hat{z} = \hat{\lambda}\Delta t$ ,  $I$  is the  $s \times s$  identity matrix and  $e$  is  $(1, \dots, 1)^T$ . The requirement for linear stability, which we will loosely call 'A-stability', is

$$|R(z, \hat{z})| \leq 1 \quad \text{for } z \leq 0. \quad (7.33)$$

As for algebraic stability, A-stability cannot be concluded from the A-stability of the two methods, and has to be investigated for each method separately. On the other hand, the L-stability of  $\mathcal{A}$  is inherited by the additive Runge-Kutta method, see appendix B.3.2.

### 7.5.3 Radau IIA/B pair

The Radau IIA methods are a suitable choice for  $\mathcal{A}$ , because they give the highest possible order  $(2s - 1)$  combined with stiff accuracy (and therefore L-stability). For

Type	Energy- cons.	Time- rev.	$L$ -stable	Alg. stable	Order
Radau IIA/B	Y	N	Y	N	$2s - 1$
Lobatto IIIC/E	Y	Y	Y	N	$2s - 2$
DIRK L/E	Y	Y	Y	N	2

Table 7.4: Properties of some implicit additive Runge-Kutta methods.

$\hat{\mathcal{A}}$  we then use  $\mathcal{E}(\mathcal{A})$ , the Radau IIB methods, and their combination results in a method that has many of the required properties (table 7.4): it is of order  $2s - 1$ , energy-conserving,  $L$ -stable, and has  $c = \hat{c}$  and  $b = \hat{b}$ . We note that, instead of using equation (7.28) it is also possible to evaluate the additional order conditions originating from the use of additive methods [85]. For  $s = 2$  there are no additional order conditions, for  $s = 3$  there are 16 additional conditions, and they are indeed satisfied by the Radau IIA/B pair.

Evaluating (7.32) for the new scheme results in

$$R(z, \hat{z}) = \frac{12 + 6\hat{z} + 4z + z\hat{z} + \hat{z}^2}{12 - 8z - 6\hat{z} + 2z^2 + 3z\hat{z} + \hat{z}^2}. \quad (7.34)$$

This amplification factor is shown in figure 7.1b.  $A$ -stability can be proven as follows. Let  $\hat{z} = \alpha i$ ,  $z = \beta$ , then the requirement for  $A$ -stability can be written as

$$|R(\beta, \alpha i)|^2 \leq 1 \quad \text{for } \beta \leq 0, \quad (7.35)$$

which becomes

$$-\beta(\beta - 6)(\beta^2 - 2\beta + \alpha^2 + 12) < 0. \quad (7.36)$$

This expression is indeed satisfied for all  $\alpha$  and  $\beta < 0$  and we can conclude that the two-stage Radau IIA/B method is  $A$ -stable. For pure convection problems we have  $z = 0$  and (7.34) reduces to

$$R(z = 0, \hat{z}) = \frac{12 + 6\hat{z} + \hat{z}^2}{12 - 6\hat{z} + \hat{z}^2}, \quad (7.37)$$

so  $|R(z = 0, \hat{z})| = 1$ , and there is no damping on the imaginary axis, as expected. One recognizes the stability function for Radau IIB which is in fact the same as the stability function for the two-stage Gauss method. Consequently, the Radau IIB method is fourth order accurate for linear equations, instead of third order. For diffusion dominated problems where  $z \rightarrow -\infty$ , we find (for fixed  $\hat{z}$ ):

$$\lim_{z \rightarrow -\infty} R(z, \hat{z}) = 0. \quad (7.38)$$

For the fifth order, 3-stage method,  $A$ -stability can be proven in a similar way.

$$\begin{array}{c|cc}
 \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\
 1 & \frac{3}{4} & \frac{1}{4} \\
 \hline
 & \frac{3}{4} & \frac{1}{4} \\
 \text{(a) } \mathcal{A}_{\text{RadIIA}} & & 
 \end{array}
 \quad
 \begin{array}{c|cc}
 \frac{1}{3} & \frac{3}{8} & -\frac{1}{24} \\
 1 & \frac{7}{8} & \frac{1}{8} \\
 \hline
 & \frac{3}{4} & \frac{1}{4} \\
 \text{(b) } \mathcal{A}_{\text{RadIIB}} & = & \\
 \mathcal{E}(\mathcal{A}_{\text{RadIIA}}) & & 
 \end{array}$$

Table 7.5: Energy-conserving,  $L$ -stable, 2-stage, 3rd order Radau IIA/B pair.

#### 7.5.4 Lobatto IIIC/E pair

Another additive Runge-Kutta method combining stiff accuracy and energy conservation can be constructed by applying the transform  $\mathcal{E}$  on the Lobatto IIIC methods, leading to the IIIE methods. These are of lower order ( $2s - 2$ ) than the Radau pair, but keep the time-reversibility property for inviscid flows. We will investigate numerically if this additional property outweighs the loss in order compared to the Radau pair. From the one-parameter family of stiffly accurate Lobatto methods that are of order  $2s - 2$  [99], the Lobatto IIIC method is the only one that satisfies  $D(1)$  for  $s = 2$  (necessary for  $c = \hat{c}$  according to theorem 7.5.1). The degree of the numerator of the stability function  $R$  is at most  $s - 2$  for the Lobatto IIIC method [63], leading to a very fast decay along the negative real axis of the stability domain, and making them very suitable for stiff problems. Table 7.6 shows the Butcher tableaux for the 2-stage methods.

$A$ -stability for the 2-stage method follows by calculating the stability function as before:

$$R(z, \hat{z}) = \frac{4 + 2\hat{z} + \hat{z}^2 + z\hat{z}}{4 - 4z - 2\hat{z} + 2z^2 + 3z\hat{z} + \hat{z}^2}, \quad (7.39)$$

shown in figure 7.1c.  $|R(\beta, \alpha i)|^2 \leq 1$  leads to

$$-\beta(-2 + \beta)(\beta^2 - 2\beta + \alpha^2 + 4) \leq 0, \quad (7.40)$$

which is indeed satisfied for  $\beta \leq 0$ . For the 3-stage method,  $A$ -stability follows in a similar fashion.

We note that, apart from the Lobatto IIIC/E method, other combinations of Lobatto methods can lead to interesting methods. For example, the Lobatto IIIA/B pair has the property that  $\tilde{E} = 0$ , but is not energy-conserving in the inviscid limit. We refer to [92] for an extensive discussion on additive Lobatto methods.

#### 7.5.5 DIRK pair

We mentioned before that the maximum order for an energy-conserving DIRK method is two, and for a two-stage method there is one free parameter  $c_1$ , see equation (7.14). We combine this method with a two stage, second order  $L$ -stable DIRK

$$\begin{array}{c|cc}
 0 & \frac{1}{2} & -\frac{1}{2} \\
 1 & \frac{1}{2} & \frac{1}{2} \\
 \hline
 & \frac{1}{2} & \frac{1}{2}
 \end{array}
 \quad
 \begin{array}{c|cc}
 0 & \frac{1}{4} & -\frac{1}{4} \\
 1 & \frac{3}{4} & \frac{1}{4} \\
 \hline
 & \frac{1}{2} & \frac{1}{2}
 \end{array}$$

(a)  $\mathcal{A}_{\text{LobIII C}}$       (b)  $\mathcal{A}_{\text{LobIII E}} = \mathcal{E}(\mathcal{A}_{\text{LobIII C}})$

Table 7.6: Energy-conserving, time-reversible,  $L$ -stable, 2-stage, 2nd order, Lobatto III C/E pair.

method to arrive at an additive Runge-Kutta method of DIRK type which combines energy conservation, time reversibility and  $L$ -stability. Again, we look for a method with  $c = \hat{c}$ , which implies  $b = \hat{b}$  due to the second order coupling condition. When we furthermore adhere to convention (5.15), the condition for  $L$ -stability (namely a singular  $Q$ -matrix), leads to the following family of  $L$ -stable DIRK methods:

$$\begin{array}{c|cc}
 c_1 & c_1 & 0 \\
 c_1 + \frac{1}{2} & \frac{1}{2} \frac{c_1(2c_1-3)}{c_1-1} & \frac{1}{2} \frac{2c_1-1}{c_1-1} \\
 \hline
 & 2c_1 & 1 - 2c_1
 \end{array}
 \tag{7.41}$$

In contrast to the previous tableaux for the diffusive terms,  $\mathcal{A}$  is not algebraically stable, because  $E$  is not nonnegative definite. It is not possible to choose  $c_1$  such that both the convective and diffusive tableau are of SDIRK type. In the sequel we take  $c_1 = \frac{1}{4}$ , so that for pure convection problems the method is of SDIRK type; the tableaux are then given in table 7.7. Note that here  $\hat{\mathcal{A}} \neq \mathcal{E}(\mathcal{A})$ . The amplification factor is given by

$$R(z, \hat{z}) = \frac{48 + 20z + 24\hat{z} + 3z\hat{z} + 3\hat{z}^2}{(-4 + z + \hat{z})(-12 + 4z + 3\hat{z})},
 \tag{7.42}$$

and is shown in figure 7.1d. The method is  $A$ -stable, because

$$-\beta(-12 + \beta)(\beta^2 - 2\beta + 24 + \alpha^2) \leq 0
 \tag{7.43}$$

is satisfied for  $\beta \leq 0$ .

$$\begin{array}{c|cc}
 \frac{1}{4} & \frac{1}{4} & 0 \\
 \frac{3}{4} & \frac{5}{12} & \frac{1}{3} \\
 \hline
 & \frac{1}{2} & \frac{1}{2}
 \end{array}
 \quad
 \begin{array}{c|cc}
 \frac{1}{4} & \frac{1}{4} & 0 \\
 \frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\
 \hline
 & \frac{1}{2} & \frac{1}{2}
 \end{array}$$

(a)  $\mathcal{A}_{\text{DIRK L}}$       (b)  $\mathcal{A}_{\text{DIRK E}}$

Table 7.7: Energy-conserving, time-reversible,  $L$ -stable, 2-stage, 2nd order DIRK pair.

### 7.5.6 Summary

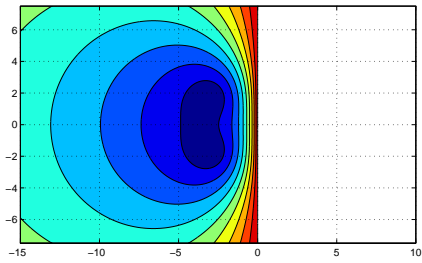
To summarize, we have a hierarchy of methods, listed in table 7.4. The Gauss methods have highest possible order,  $p = 2s$ , but lack  $L$ -stability. Sacrificing one order,  $p = 2s - 1$ , leads to the additive Radau IIA/B methods, which have both energy conservation and  $L$ -stability. Sacrificing another order,  $p = 2s - 2$ , leads to the additive Lobatto IIIC/E methods, which have energy conservation, time reversibility and  $L$ -stability. Dropping the requirement on algebraic stability of the diffusive tableau leads to an additive DIRK L/E method, also energy-conserving, time-reversible and  $L$ -stable. These methods will be numerically tested in section 7.8, together with the Lobatto IIIA (Crank-Nicolson) method, because this is a widely used method and an example of a method which is time-reversible but not energy-conserving.

All these methods are two-stage methods, with the exception of the 1-stage Gauss method. The stability domains of the methods,  $|R(\beta, \alpha i)| \leq 1$ , are shown in figures 7.1a-7.1d. The stability domain of the 2-stage Lobatto IIIA method (Crank-Nicolson) coincides with the Gauss 1-stage method (implicit midpoint) and can be found in any standard textbook, e.g. [63]. One can observe that the proposed methods satisfy both  $|R| = 1$  on the imaginary axis and  $|R| = 0$  for  $\beta \rightarrow -\infty$ . However, when compared with the exact stability function (figure 7.1e), or with a stiffly accurate method such as Radau IIA (figure 7.1f), the proposed methods possess a much slower decay of  $R$  to zero along the negative real axis.

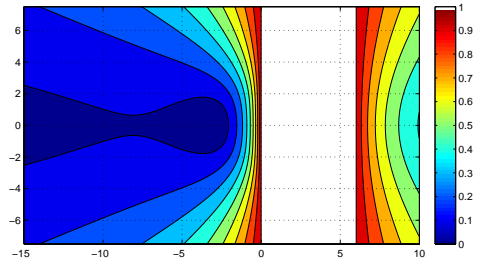
We note that all additive methods considered here have a region  $|R| < 1$  in the right half of the complex plane. For example, equation (7.36) is also satisfied for  $\beta \geq 6$ , and equation (7.43) for  $\beta > 12$ . This is an undesirable property when one wants to integrate physically ‘unstable’ phenomena. However, compared to the ‘original’ stiffly accurate /  $L$ -stable methods, such as the Radau IIA method depicted in figure 7.1f, the behavior is much better. From the methods considered here the Gauss methods are the only ones that have  $|R| > 1$  in the entire right half of the complex plane.

## 7.6 THE ACCURACY OF THE PRESSURE

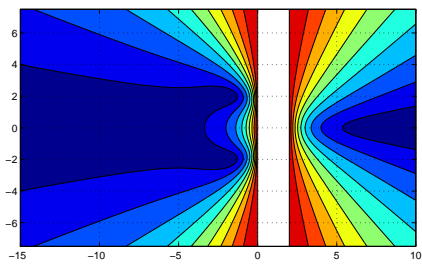
The velocity at the new time step follows from the sequence (7.4)-(7.6). The ‘pressure’  $\phi_{n+1}$  necessary to obtain a divergence-free  $u_{n+1}$  is only a first order approximation to  $p_{n+1}$ , as was explained in section 5.3.2. A higher order accurate  $p_{n+1}$  can be computed by solving an additional Poisson equation, equation (6.26). The resulting pressure has the same temporal accuracy as the velocity field  $u_{n+1}$ . If  $\dot{r}_1(t)$  is not available or not well-defined, or to avoid the additional Poisson equation, one can construct higher order approximations by making linear combinations of the stage values  $\phi_i$ . For the Runge-Kutta methods based on Gauss, Radau and Lobatto quadrature we apply the techniques mentioned in [63, 60] to the Navier-Stokes equations, and subsequently we extend and unify these techniques in a new approach for additive Runge-Kutta methods proposed in section 6.3.2. In contrast to what is reported for explicit methods in that section, for implicit methods and



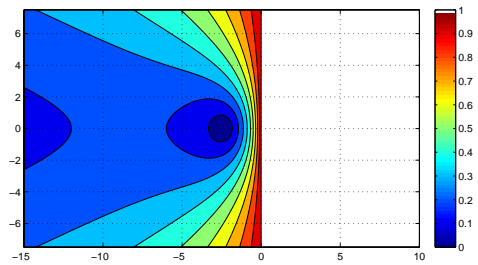
(a) Gauss - 2 stage



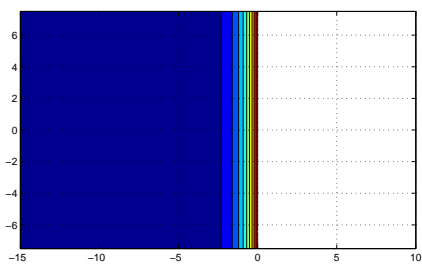
(b) Additive Radau II A/B - 2 stage



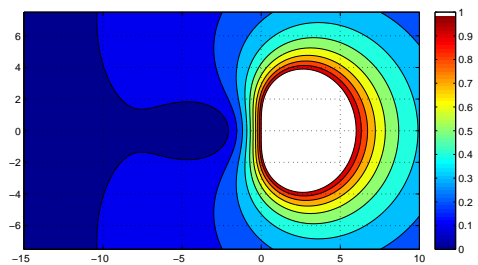
(c) Additive Lobatto III C/E - 2 stage



(d) Additive DIRK L/E - 2 stage



(e) Exact



(f) Radau IIA - 2 stage

Figure 7.1: Stability regions  $|R(\beta, \alpha i)| \leq 1$ .

steady boundary conditions for the continuity equation it is not possible to obtain a pressure of the same order as the velocity field without the additional Poisson solve mentioned above.

### 7.6.1 *Radau IIA, Lobatto IIIC*

If  $A$  is non-singular one can write [63]:

$$p_{n+1} = p_n + \sum_{i,j=1}^s b_i \omega_{ij} (\psi_j - p_n) = R(\infty) p_n + \sum_{i,j=1}^s b_i \omega_{ij} \psi_j, \quad (7.44)$$

where  $\psi = A^{-1} C^p \phi$  and  $\omega_{ij}$  are the entries of  $A^{-1}$ . For the Radau IIA and Lobatto IIIC methods this gives an approximation of order  $s$  and  $s - 1$ , respectively, corresponding to their stage orders. They both feature stiff accuracy and a non-singular  $A$ , leading to  $R(\infty) = 0$ , so  $p_{n+1}$  is not depending on  $p_n$  and the global error equals the local error. Furthermore, since

$$b^T = \begin{pmatrix} 0 & \dots & 0 & 1 \end{pmatrix} A, \quad (7.45)$$

we have

$$p_{n+1} = \psi_s = \sum_{i=1}^s \omega_{si} c_i \phi_i. \quad (7.46)$$

In case of Lobatto IIIC  $c_i \phi_i$  is replaced by  $\rho_i$  since  $c_1 = 0$ .

### 7.6.2 *Gauss*

For the Gauss methods  $|R(\infty)| = 1$  and (7.44) gives an approximation of order  $s - 1$  ( $s$  odd) or  $s - 2$  ( $s$  even), which is unsatisfactory, especially for the 1- and 2-stage methods under consideration. An alternative is to construct a polynomial  $v(t)$  that interpolates the  $\psi_i$  values. Evaluating  $v(t)$  at  $t_{n+1}$  then gives an  $s$ -th order approximation to the pressure [63, 6]:

$$p_{n+1} \approx v(t_{n+1}) = \sum_{i=1}^s \psi_i \prod_{j=1, j \neq i}^s \frac{1 - c_j}{c_i - c_j}. \quad (7.47)$$

With this expression the pressure  $p_{n+1}$  is independent of  $p_n$ , like for  $L$ -stable methods, and the global error is therefore the same as the local error.

### 7.6.3 *Lobatto IIIA*

The Lobatto IIIA methods have  $c_1 = 0$ , so the approach for the Gauss methods cannot be applied. Furthermore,  $A$  is singular, hence the approach for the Radau IIA

and Lobatto III C methods also fails. However, since the method is stiffly accurate and satisfies  $c_1 = 0$ , we can take  $p_{n+1} = \psi_s$ . By defining  $\tilde{A} = A_{2:s,2:s}$  (which is non-singular) and  $\tilde{\omega}_{ij} = (\tilde{A}^{-1})_{ij}$  the entries of its inverse,  $\psi$  can be written in terms of  $\rho$  as

$$\psi_i = \sum_{j=2}^s \tilde{\omega}_{ij}(\rho_j - a_{j1}\psi_1), \quad (7.48)$$

where  $\psi_1 = p_n$ . In this case  $p_{n+1}$  depends on  $p_n$  and the order is  $s$  ( $s$  even) or  $s - 1$  ( $s$  odd) [60].

#### 7.6.4 A unified approach

In this section we compare the results obtained above with the approach outlined for explicit methods in section 6.3.2. This method is based on the reconstruction of the point value  $p_{n+1}$  from the *average* pressure values  $\phi_i$ . If a Runge-Kutta method has at least  $\eta$  stages that satisfy  $C_i(\eta)$ , then it is possible to obtain an  $\mathcal{O}(\Delta t^\eta)$  accurate approximation to  $p(t_{n+1})$  by combining the  $\phi_i$  values. For the implicit methods under consideration we have  $C(\eta)$  (holding for all stages), with  $\eta = s$  or  $s - 1$  (except the DIRK methods), so such a construction is always possible.  $p_{n+1}$  then follows as (equation (6.45)):

$$p_{n+1} = \sum_{k \in K} \phi_k c_k \Delta t \dot{\ell}_k(t_{n+1}), \quad (7.49)$$

where  $\ell_k(t)$  is given by (6.42). The set  $K$  consists of (at least)  $\eta$  stages, which have to be chosen such that the  $c_k$  are distinct and nonzero.

Equation (7.49) appears to be equivalent to the formulations above for the Radau and Gauss methods:

- The Gauss methods satisfy  $C(s)$ , so  $\eta = s$  and  $K = \{1, 2, \dots, s\}$ . First we evaluate our approach, equation (7.49), for  $s = 2$ :

$$p_{n+1} = \phi_1 \frac{2 - c_2}{c_1 - c_2} + \phi_2 \frac{c_2 - 2}{c_1 - c_2}. \quad (7.50)$$

Secondly, we rewrite (7.47) by interpreting  $C(s)$ , see equation (5.17), as an equation for the  $a$ -coefficients in terms of the  $c$ -coefficients.  $\psi = A^{-1}C^p\phi$  can then be completely expressed in terms of the  $c$ -coefficients, and when evaluating the resulting expression, equation (7.50) is obtained. This also holds for  $s = 3$ , and we conjecture that our approach, equation (7.49), is equivalent to equation (7.47) for any  $s$ .

- For the Radau methods we proceed in a similar way. We evaluate (7.50) for  $c_s = 1$  and compare with (7.46), evaluated with  $C(s)$  and  $c_s = 1$ . Both formulations are again identical for  $s = 2$  and  $s = 3$ .



For the Lobatto IIIC methods our approach (7.49) differs from (7.46). Since  $C(s - 1)$  and  $c_1 = 0$  are satisfied, we take  $K = \{2, \dots, s\}$ , and (7.49) boils down to the following first order approximation:

$$p_{n+1} = \frac{\rho_2}{c_2}, \quad (7.51)$$

while (7.46) gives

$$p_{n+1} = \omega_{21}\rho_1 + \omega_{22}\rho_2, \quad (7.52)$$

which is also of first order. It turns out, somewhat surprisingly, that  $\omega_{22} = \frac{1}{c_2}$ . The presence of the  $\rho_1$  term in (7.52) is therefore not necessary to obtain the required order of accuracy. For  $s = 3$  a similar phenomenon occurs.

Lastly, we compare the pressure update for the Lobatto IIIA method with our approach. Equation (7.48) reads, for  $s = 2$ :

$$p_{n+1} = \frac{1}{a_{22}}(\rho_2 - a_{21}p_n), \quad (7.53)$$

which is a second-order accurate approximation. This is the only method where  $p_n$  appears in the formulation, and the only method for which our approach gives a lower order: first order instead of second order for  $s = 2$ .

An advantage of approach (7.49) is that it can be directly applied to additive Runge-Kutta methods, because it only depends on the  $c$ -coefficients, which are the same for the convective and diffusive tableaux. Table 7.8 summarizes the different methods. Although the Radau IIA/B method satisfies only  $C(s - 1)$  instead of  $C(s)$ , we still take the same approach as for Radau IIA, so that the Radau IIA method is simply a special case of the Radau IIA/B method. For the Lobatto IIIC/E pair we can directly use the formulation for Lobatto IIIC. The DIRK pair satisfies  $C(1)$  and the pressure is therefore limited to first order; we simply take

$$p_{n+1} = \phi_2. \quad (7.54)$$

Type	$s = 2$	general $s$	Order
Gauss, Radau IIA	(7.50)	(7.49)	$s$
Radau IIA/B	(7.50)	(7.49)	$s - 1$
Lobatto IIIC/IIIE	(7.51)	(7.49)	$s - 1$
Lobatto IIIA	(7.53)	(7.48)	$s/s - 1$ (even/odd)
DIRK	(7.54)	-	1

Table 7.8: Computation of the pressure for implicit (additive) Runge-Kutta methods.

## 7.7 IMPLEMENTATION ISSUES

## 7.7.1 System of equations

Equation (5.12) represents a full, implicit system of nonlinear equations. Introducing the iteration counter  $k$  we write for the stage values

$$U^{k+1} = U^k + \Delta U^k. \quad (7.55)$$

One iteration of Newton's method leads to the following linear system of equations:

$$\begin{bmatrix} T^k & \Delta t C^p G \\ M & 0 \end{bmatrix} \begin{bmatrix} \Delta U^k \\ \Delta \phi^k \end{bmatrix} = d^k, \quad (7.56)$$

where

$$d^k = \begin{bmatrix} d_u^k \\ d_\phi^k \end{bmatrix} = \begin{bmatrix} u_n - U^k + \Delta t A F^k - \Delta t C^p G \phi^k \\ -M U^k + r_1 \end{bmatrix}, \quad T^k = I - \Delta t A J^k. \quad (7.57)$$

$J = \frac{\partial F}{\partial U}$  represents the Jacobian of the system. With full Newton the exact Jacobian is evaluated at each iteration step. We stress again that by using  $\phi$  instead of  $\psi$ ,  $M$  and  $G$  are block-diagonal matrices ( $I_s \otimes M$ ,  $I_s \otimes G$ ). At each time step the non-linear system is solved until an absolute tolerance  $\delta_a$  or relative tolerance  $\delta_r$  on the residual is satisfied:

$$\|d^k\|_\infty \leq \delta_a \quad \text{or} \quad \|d^k\|_\infty \leq \delta_r \|d^1\|_\infty. \quad (7.58)$$

The presence of the convective terms in the Jacobian leads to an asymmetric system, which moreover changes every iteration, making the solution of the system expensive. A possible method to solve the system is the Newton-Krylov method proposed by Pereira et al. [125]; a general overview of methods is given in [16]. In the results presented in this chapter we have used small test problems for which the use of a direct solver was most efficient. However, to make the approach suitable for large-scale computations, we will give some directions for improvement.

One can use simplified Newton instead of full Newton: take the Jacobian at the start of each time step and keep it constant during the iterative process. If a direct method is used, one can even compute the LU-decomposition of  $I - \Delta t A J(U^0)$  and use it for all iterations. The convergence of the residual, given a sufficiently accurate starting value, is then linear instead of quadratic. A starting value can be obtained by fitting a polynomial of degree  $s + 1$  through the values of  $U_i$  of the previous time step and then extrapolate this to the stages of the new time step [63]. This provides in general an  $\mathcal{O}(\Delta t^s)$  approximation to  $U^k$ .

### 7.7.2 Splitting

Another technique which can significantly reduce computational costs is splitting the solution of velocity and pressure. Such a splitting can be seen as a block LU-decomposition of matrix (7.56) [129]:

$$\begin{bmatrix} T^k & G \\ M & 0 \end{bmatrix} = \begin{bmatrix} T^k & 0 \\ M & -M(T^k)^{-1}G \end{bmatrix} \begin{bmatrix} I & (T^k)^{-1}G \\ 0 & I \end{bmatrix}, \quad (7.59)$$

so one iteration of the nonlinear method is equivalent to the following sequence:

$$T^k \Delta \hat{U}^k = d_u^k, \quad (7.60)$$

$$M(T^k)^{-1}G\Delta\phi^k = M\Delta\hat{U}^k - d_\phi^k, \quad (7.61)$$

$$\Delta U^k = \Delta \hat{U}^k - (T^k)^{-1}G\Delta\phi^k. \quad (7.62)$$

This factorization does not introduce a splitting error yet. Although the large original matrix is not present in the formulation, the explicit computation of  $(T^k)^{-1}$ , necessary in the second equation (note that  $M$  and  $G$  are not square), makes the approach unattractive in practice. When  $(T^k)^{-1}$  in (7.61)-(7.62) is replaced by an approximation  $(\hat{T}^k)^{-1}$  one obtains an approximate factorization as proposed by Perot [129]. The simplest choice is to take the first order approximation  $T^k = I$ , which leads to a scheme similar to fractional step or projection methods. Higher order methods follow by more accurate approximations of  $(T^k)^{-1}$ , but result in a matrix  $M(T^k)^{-1}G$  which is not a Laplacian anymore. In any case, a splitting error is introduced, which destroys the energy-conserving property of the method (if present). This is due to the fact that the intermediate velocity field  $\hat{U}$  does not satisfy the divergence constraint and therefore contributes to the kinetic energy. Furthermore, the unconditional stability of methods associated with the energy conservation property is lost (although a weaker type of stability can still be proven [194]).

To retrieve the original conservation and stability properties we can proceed as follows. Taking  $(T^k)^{-1} = I$  one can repeatedly solve (7.60)-(7.62) until (7.58) is satisfied - the iterative method is now used to remove both linearization and splitting errors. From our experience it is most efficient to solve the linear problem (7.60) at each iteration  $k$  and to not introduce a subiteration to solve the nonlinear system at each  $k$ . A slightly better approximation to  $(T^k)^{-1}$  is to take  $(\text{diag}(T^k))^{-1}$  [120], but this only leads to a positive definite matrix  $M(T^k)^{-1}G$  if the time step is small enough. In any case, the quadratic or linear convergence of 'full' or 'simplified' Newton is lost.

A completely different approach is the exact fractional step method developed in Chang et al. [35] for staggered mesh methods, by applying a discrete rotation operator to the system of equations. This leads to a system of size approximately  $N_p \times N_p$  (2D) or  $N_u \times N_u$  (3D). Since there is no splitting error involved, the same number of iterations as for the original system is required. The matrix to be solved

has similar properties as matrix  $T^k$ , so the same (iterative) technique normally applied to that matrix can be used. The solution of the saddle-point system is avoided. For future work on large-scale problems this might be the method of choice.

### 7.7.3 Linearization

The energy-conservation property (7.12) has been derived for Runge-Kutta methods assuming the convective terms are given by  $C(U_i, U_i)$ . However, property (4.77) shows that  $(C(c_i, U_i), U_i) = 0$  holds *independent* of the time level of  $c$  as long as  $Mc = 0$  (see e.g. [168, 203, 195]): the time level of the *convecting* quantity is not important for the conservation of the norm of the *convected* quantity. This means that instead of taking  $c_i = U_i$  one can approximate  $c_i$  by using previous time levels, and then only a *linear* system has to be solved in order to achieve conservation of energy. For example, for the 1-stage Gauss method, a second order approximation is  $\tilde{c}_1 = \frac{3}{2}c_n - \frac{1}{2}c_{n-1}$ . The extension to higher order methods is detailed in [151].

If linearization is combined with the splitting of Chang et al. [35] no iteration is required. Although the total work involved at each time step is obviously higher than the solution of a simple Poisson equation (as done in fully explicit methods), the reduction in number of necessary iterations [35] and increase in allowable time step can make this approach competitive with explicit methods.

## 7.8 RESULTS

A number of test cases are considered to evaluate the practical performance of the (additive) Runge-Kutta methods presented in this chapter, which were summarized in section 7.5.6. It is interesting to compare the additive methods with the two methods by which they are formed (such as Radau IIA and IIB in case of IIA/B). In the last test we leave out the Lobatto IIIC/E method, because it will turn out not to be competitive. We consider 1-stage (Gauss 2) and 2-stage methods (all others). We have also numerically verified the orders of accuracy of the 3-stage variants of these methods. These will not be presented here, because the 2-stage methods are a better compromise between order of accuracy and computational cost. Note that we leave out the DIRK E method, because it is simply the Gauss 2 method but then applied to a time step twice as large.

### 7.8.1 Taylor-Green vortex

We repeat the Taylor-Green vortex test from section 6.4.1, now for implicit methods. The nonlinear system of equations is solved with full Newton, with a trivial initial guess and  $\delta_a = \delta_r = 10^{-14}$ . For the largest time step,  $\Delta t = 1/10$ , this requires 2 or 3 nonlinear iterations (depending on the method); for the smallest time steps 1 iteration suffices.

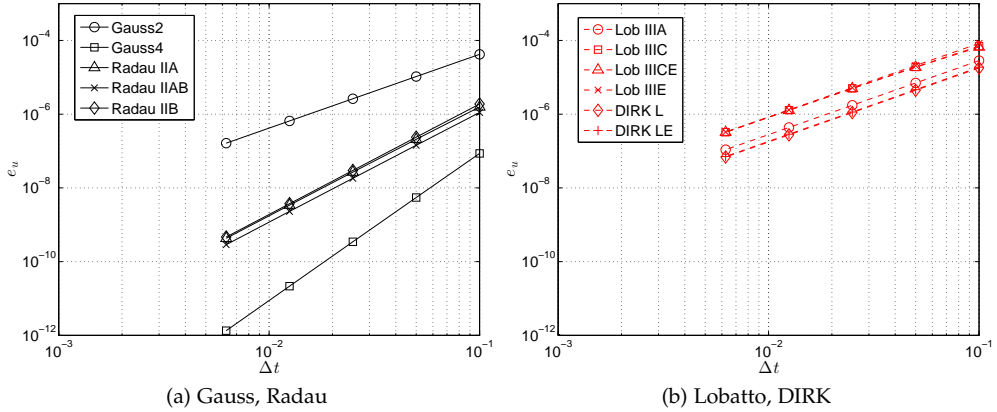


Figure 7.2: Error in velocity as function of time step.

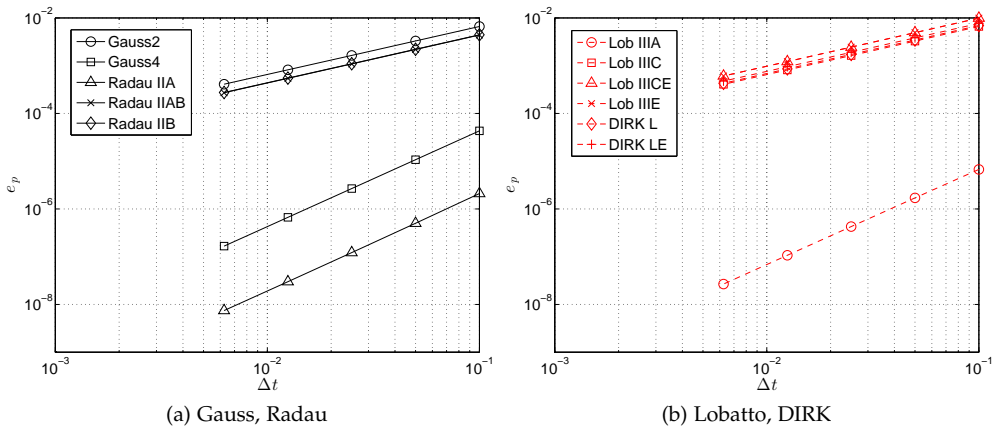


Figure 7.3: Error in pressure as function of time step.

Figures 7.2 and 7.3 show the temporal error in the  $u$ -velocity and pressure for all methods:

$$e_u = \|u - u_{\Delta t \rightarrow 0}\|_\infty, \quad e_p = \|p - p_{\Delta t \rightarrow 0}\|_\infty. \quad (7.63)$$

The error in the  $v$ -velocity component and the error measured in the  $L_2$ -norm give almost identical results and are not shown here. The methods are split over two figures to keep the presentation as clear as possible. All methods achieve their theoretical order of accuracy. The third order methods of the Radau family are almost indistinguishable, and similarly the second order methods of Lobatto and DIRK type are close. The results for the pressure error show that only Gauss 4, Radau IIA and Lobatto IIIA reach second order convergence, as predicted by table 7.8. The Lobatto IIIA stands out since it is the only method that achieves the same order in velocity and pressure, and its pressure error is much lower than Gauss 2. Although this corresponds to the theoretical results, it is still surprising considering that both methods are so closely related: implicit midpoint (Gauss 2) and trapezoidal rule (Lobatto IIIA), which are identical for linear equations. We remark that in all cases a pressure with the same temporal order as the velocity can be obtained at the expense of solving the additional Poisson equation (6.26).

## 7.8.2 Inviscid shear layer roll-up

### 7.8.2.1 Introduction

In order to study the energy conservation and time reversibility properties of the different Runge-Kutta methods we simulate again the roll-up of a shear layer, as was introduced in 3.3.2. For this test case (having periodic boundaries) equations (2.22) and (2.25) should hold in a discrete sense. We first investigate these (in)equalities when  $\nu$  approaches zero. Since the diffusive operator is a symmetric positive definite matrix, an analog of the continuous energy estimate (2.25) can be obtained:

$$K(t) \leq K(t_0)e^{-2\nu C_h t}. \quad (7.64)$$

In the continuous case the Poincaré constant reads  $C = (\pi/L)^2$ ; in the discrete case we have  $C_h = 8/L^2$  (for the second order discretization), with  $L = 2\pi$  the size of the domain. The continuous energy estimate is slightly stricter than the semi-discrete estimate.

Figure 7.4 shows the evolution of the normalized kinetic energy  $K(t)/K(t=0)$  as a function of time for different values of  $\nu = 1/\text{Re}$  and for both spatial discretizations (second and fourth order), using the Gauss methods. The energy is strictly decreasing for any value of  $\nu > 0$ , and for  $\nu = 0$  energy is *exactly conserved* (until machine precision). Equation (7.64) gives an upper bound; in this case the energy decreases much faster. We have taken a coarse mesh ( $40 \times 40$ ) and a large time step ( $\Delta t = \frac{1}{2}$ ) to stress that property (2.22) is discretely mimicked for *any mesh* and *any time step*.

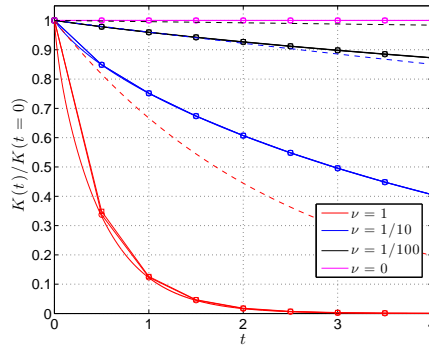


Figure 7.4: Energy as a function of time for different values of  $\nu$ . --: estimate (7.64), —: reference solution;  $\square$ : 2nd order space and time;  $\circ$ : 4th order space and time.

We continue with the case  $\nu = 0$ , i.e., the flow is completely inviscid. Noise ('wiggles') will be present on *any mesh* (if the simulation is run long enough), due to dispersive and aliasing errors and the absence of viscosity to damp the smallest scales. The associated 'bursts' in vorticity lead to an increase in the enstrophy. However, in contrast to what is mentioned in Brown and Minion [21], such oscillations do *not* lead to 'catastrophic blowup' with the energy-conserving time-integration methods under investigation (in fact, they *cannot* blow up due to their unconditional stability properties). Time-reversibility errors will be investigated by reverting the simulation at  $t = 2$  or  $t = 8$ : we change  $\Delta t$  to  $-\Delta t$  and march back to  $t = 0$ . Since we solve the inviscid Navier-Stokes equations, the additive methods reduce to their energy-conserving part: Radau IIA/B is Radau IIB, Lobatto IIIC/E is IIIE and DIRK L/E is DIRK E.

We define the following error measures. The error in the energy of the flow at any time instant  $t$  is defined as

$$e_k(t) = \frac{K(t) - K(t=0)}{K(t=0)}, \quad (7.65)$$

where  $K$  is the total discrete energy (equation (4.40)). To express the error in time reversibility of a method we compare the velocity fields at the start and the end of the simulation:

$$e_t = \|V(t_{\text{end}}) - V(t=0)\|_{\infty}. \quad (7.66)$$

Finally, it is common for this test case to study the total enstrophy of the flow,

$$\mathcal{E} = \frac{1}{2} \sum_{i,j} \Omega_{ij}^{\omega} \omega_{ij}^2, \quad (7.67)$$

with  $\Omega^{\omega}$  the sizes of the vorticity-centered finite volumes. The enstrophy is a quadratic quantity which is conserved by the continuous inviscid, incompressible, two-

dimensional Navier-Stokes equations, but not by the spatial discretization. The corresponding error,

$$e_\varepsilon(t) = \frac{\mathcal{E}(t) - \mathcal{E}(t = 0)}{\mathcal{E}(t = 0)}, \tag{7.68}$$

can be used to investigate temporal errors of the methods that have no energy and time-reversibility errors. It gives an indication of how well the flow field is resolved. The enstrophy error defined by (7.68) consists of both spatial and temporal errors. When performing order studies this spatial error is removed by subtracting a simulation at a very fine time step.

All simulations are performed with full Newton and  $\delta_a = 10^{-14}$ . We do not perform the additional pressure solve (6.26), but noticed that it helps to obtain a better initial guess for the next time step. However, extrapolating the velocities from the previous time step as prescribed in section 7.7 leads to more iterations of the Newton process if  $t > 4$ , especially for higher order methods ( $p > 2$ ). This is attributed to the under-resolution of the simulation: the flow field becomes noisy when the shear layer starts to roll up. For well-resolved simulations this issue will not occur, but in any case it can be solved by using the trivial initial guess. In order to investigate the sensitivity of energy conservation to the tolerance of the nonlinear method we performed a simulation with different tolerance levels,  $\delta_a = 10^{-12}, 10^{-8}$  and  $10^{-4}$ . In figure 7.5 the behavior of the energy error and the number of iterations are shown for the case of Gauss 2 with  $t = 10^{-1}$ . The fact that the flow field is more ‘difficult’ after  $t = 4$  manifests itself in the number of iterations (for  $\text{tol}=10^{-8}$ ) or in an increase in energy error ( $\text{tol}=10^{-12}$ ). After an initial increase the error stays approximately constant and remains on the order of the specified tolerance. This is important, since it shows that the energy-conserving methods are robust with respect to convergence of the nonlinear iteration, and keep their stability properties for larger tolerances. For smaller time steps or higher order methods the error levels are lower.

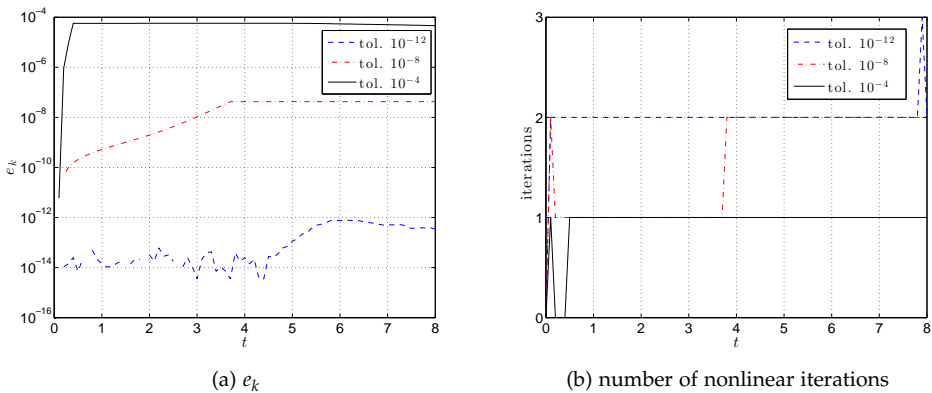


Figure 7.5: Influence of solver tolerance, Gauss 2,  $\Delta t = 10^{-1}$ .



## 7.8.2.2 Comparison

To show the beneficial effect of energy conservation we continue with the case  $\nu = 0$  and compare the behavior of  $e_k$  and  $e_\varepsilon$  for the Gauss 4 method (figures 7.6a-7.6b) and the Radau IIA method (figures 7.7a-7.7b). Time reversal occurs at  $t = 8$ . The Gauss methods behave as expected: energy is conserved until machine precision and the simulation is exactly time-reversible. Enstrophy is not conserved by the spatial discretization, as can be seen in figure 7.6b. The enstrophy error is dominated by the spatial errors, even for large time steps; at  $\Delta t = 1$  the CFL number is almost 10. Radau IIA has a small energy error (1-2% of the initial energy), but for longer time periods it will continue to grow in time; eventually smoothing and damping the entire flow. The sign of the error is negative, so the energy of the flow strictly decreases, as was expected based on the theoretical analysis from section 7.4.1. When the time step is reversed to  $-\Delta t$ , expression (7.10) still predicts a decrease in the total energy ( $\|u_{n+1}\|^2 \leq \|u_n\|^2$ ), which is in accordance with the results presented here. In contrast to Gauss 4, the (temporal) enstrophy error of Radau IIA, figure 7.7b, is approximately the same order as the spatial error. A much smaller time step is necessary to achieve results that are of the same quality as the Gauss methods.

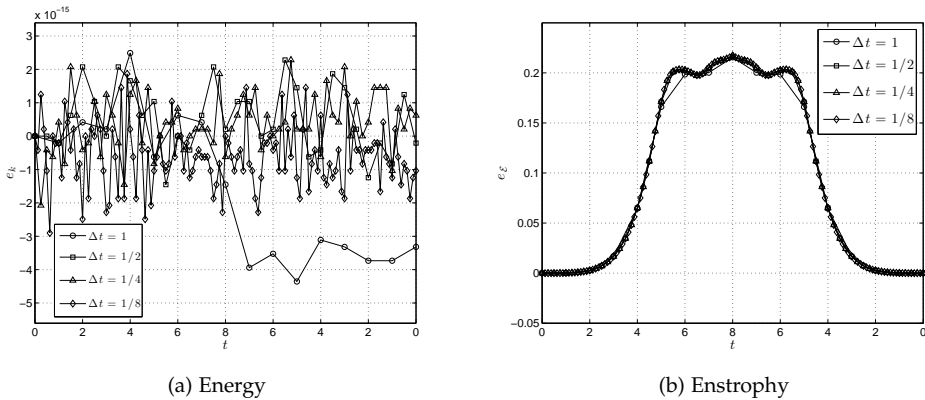


Figure 7.6: Energy and enstrophy errors for Gauss 4 applied to shear-layer problem.

The errors in energy, enstrophy and time reversibility of all methods have been collected in figures 7.8a-7.8b (time reversal at  $t = 2$ ). Figure 7.8a shows that all energy-conserving methods (Gauss, Radau IIB, Lobatto IIIE) have an error in energy which stays at machine precision, as expected. The error in the non-energy-conserving schemes (Radau IIA, Lobatto IIIC and DIRK) decreases upon time step refinement according to the order of the method, except for Lobatto IIIC, which shows third order instead of second order. Similarly, figure 7.8b shows that the enstrophy error at  $t = 2$  converges according to the order of the method, except for Lobatto IIIC, which shows again a higher order convergence rate. Most importantly, the energy-conserving methods have a much lower error constant than

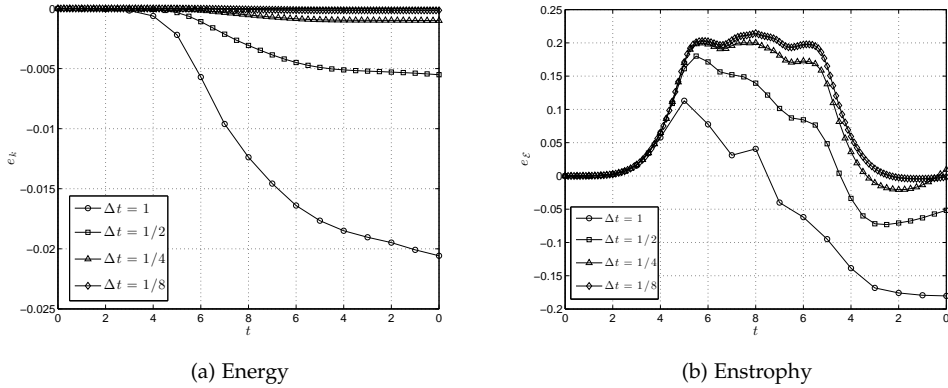


Figure 7.7: Energy and enstrophy errors for Radau IIA applied to shear-layer problem.

their non-energy-conserving counterparts. The order of accuracy alone is therefore not decisive. Figure 7.8c shows the time-reversibility error, which decreases with third order for all time-irreversible methods (Radau IIA, Radau IIB, Lobatto IIIC and DIRK) - this can probably be explained by fortuitous error cancellations in the backward simulation of the second order methods. Comparing figures 7.8a to 7.8c reveals that Radau IIB is energy-conserving but time-irreversible, and Lobatto IIIA (Crank-Nicolson) is time-reversible, but not energy-conserving. Although Crank-Nicolson is conceptually close to the implicit midpoint method, we see here that the energy-conserving properties of the implicit midpoint method make it clearly superior in predicting the enstrophy of the flow. Furthermore we see that the time-reversibility property of Lobatto IIIE (which was obtained by sacrificing one order compared to Radau IIB) is unimportant for accurate enstrophy prediction. Therefore, we conclude that *energy conservation is more important than time-reversibility*. We disagree with the conclusions stated by Duponcheel et al. [45], who mention that a ‘crucial factor for time-reversibility is the accuracy of the time stepping scheme and its interaction with the space-discretization’. The time-reversibility of a method is *not* depending on its accuracy; we clearly see that second-order methods like Crank-Nicolson or Gauss 2 can be time-reversible, while a third-order method like Radau IIA is not. For Runge-Kutta methods, time-reversibility is completely determined by satisfying conditions (7.17)-(7.19). The only relation between time-reversibility and order is that time-reversible schemes are of even order.

At  $t = 8$  similar conclusions can be drawn. The absolute error values are much higher than at  $t = 2$ , which is attributed to the lack of smoothness of the flow field compared to  $t = 2$ . Furthermore, the enstrophy errors at  $t = 8$  are more irregular than at  $t = 2$ , and not all methods are in the range of asymptotic convergence. Still, the energy-conserving methods show the best performance.

In practice, the choice for a method depends on efficiency: accuracy versus computational costs. Gauss 2 and Lobatto IIIA are the ‘cheapest’ methods, having

roughly the same cost, DIRK is about twice as expensive, and all other methods are much more expensive. For this test case the Gauss 2 or DIRK methods are favorable if second-order accuracy is sufficient. For highly accurate computations the Gauss 4 method is to be preferred, followed closely by the Radau IIB method. The Lobatto IIIC/E methods are not competitive for this test case due to their high computational costs and relatively low accuracy.

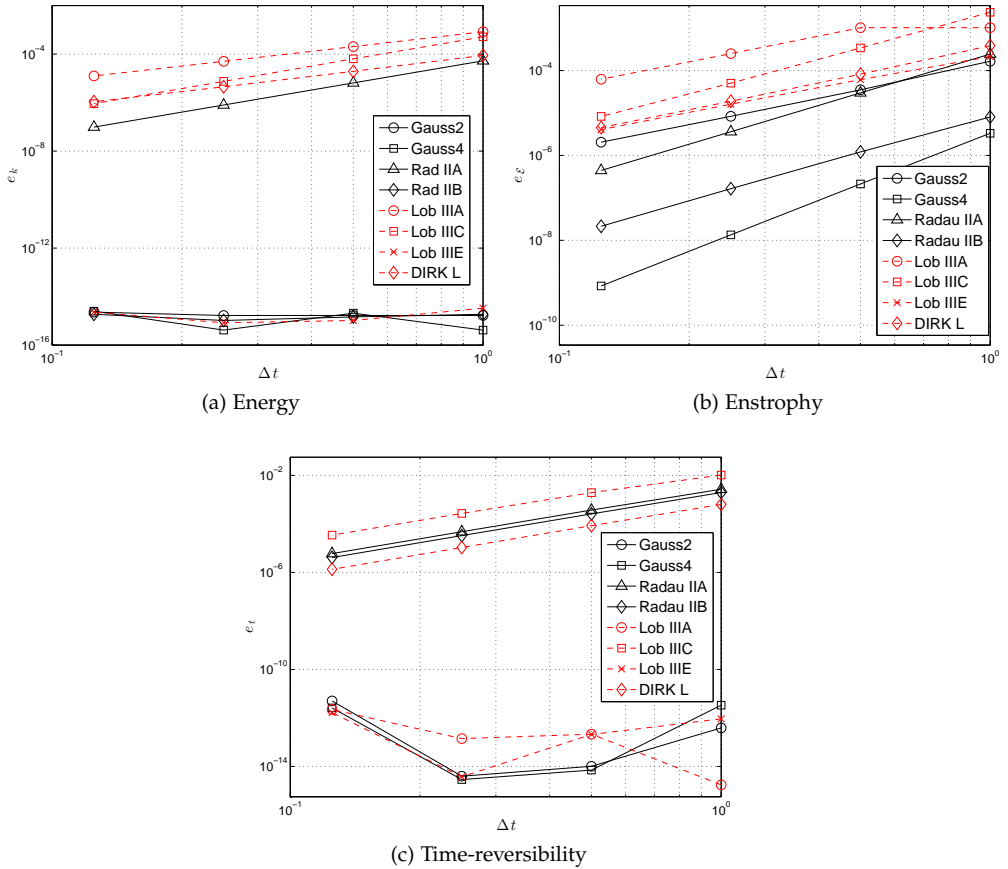


Figure 7.8: Energy, entropy and time-reversibility errors at  $t = 2$ .

7.8.3 Corner flow

In this section we investigate the performance of the different time-integration schemes for problems with varying stiffness and the role of  $L$ -stability for such problems. We consider a test case from van Kan [194]: unsteady flow through a corner, featuring unsteady inflow conditions on the top, no-slip conditions on the left and bottom, and outflow conditions on the right (figure 7.9). Although the flow in the lower-left corner is similar to the flow in lid-driven cavities, we deliberately do not compute that test case here, because it features steady boundary conditions for the continuity equation. Unsteady inflow conditions are a more severe test for methods to show the correct convergence order in both velocity and pressure (see the discussion in chapter 6). Boundary layers develop on the solid walls, and the stiffness of the problem can be controlled by varying the mesh size and Reynolds number (and as such the thickness of the boundary layers).

case	$N_x \times N_y$	$t_{end}$	Re	domain ( $L_x \times L_y$ )	grid
1	$20 \times 20$	1	10	$1 \times 1$	uniform
2	$80 \times 80$	1	10	$1 \times 1$	uniform
3	$80 \times 80$	1	10	$1 \times 1$	cosine
4	$80 \times 80$	1	1000	$1 \times 1$	cosine
5	$200 \times 100$	40	1000	$2 \times 1$	cosine

Table 7.9: Settings for corner flow.

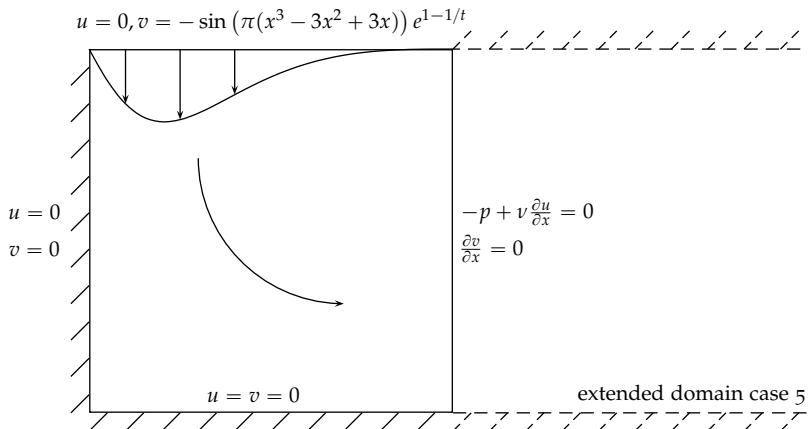


Figure 7.9: Corner flow.

We perform the five different test cases listed in table 7.9. In each case one of the parameters ( $N_x/N_y$ ,  $Re$ , domain or grid) is varied with respect to the previous case. The cosine grid is given by transforming a uniform grid  $\xi$  according to

$$x = L_x \left( 1 - \cos \left( \frac{\pi}{2} \frac{\xi}{L_x} \right) \right), \quad (7.69)$$

and similarly in  $y$ -direction. This transformation results in a fine grid near the solid walls  $x = 0$  and  $y = 0$ . For case 5 the transformation in  $y$ -direction is changed to  $y = L_y/2 \left( 1 - \cos \left( \pi \frac{\xi}{L_y} \right) \right)$  in order to have a fine grid at both  $y = 0$  and  $y = 1$ .

Qualitative pictures of the flow field are shown in figure 7.10a for case 1 (2 and 3 are similar), figure 7.10b for case 4, and figure 7.11 for case 5. In cases 1-3 the flow is diffusion-dominated and the mesh Péclet condition is satisfied in the entire domain. In cases 4 and 5 the flow is convection-dominated and the mesh Péclet condition is satisfied only near the walls, where the largest gradients appear. Inspection of the flow field did not show any significant wiggles.

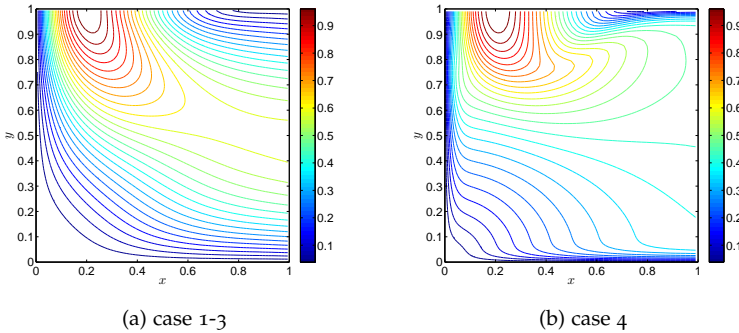


Figure 7.10: Contour lines of velocity magnitude at  $t = 1$ .

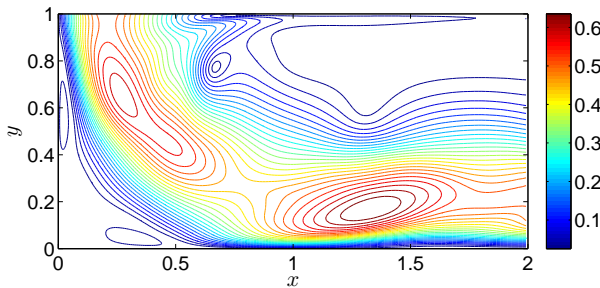


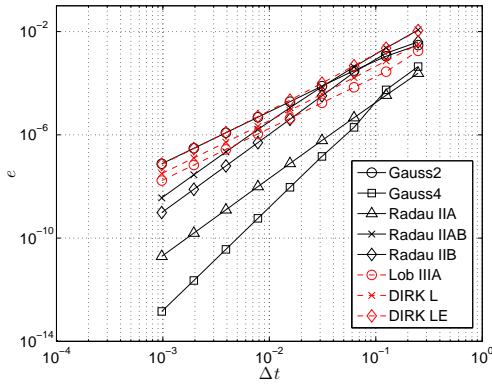
Figure 7.11: Contour lines of velocity magnitude, case 5 at  $t = 40$ .

Figures 7.12a-7.12d show the error in velocity for the different time-integration methods. The error is calculated with respect to a simulation with a very small time step, and the infinity norm is displayed. The error in pressure is left out, because it does not provide new insights with respect to the Taylor-Green test case. The Lobatto IIIC, C/E and E methods have been left out because their accuracy was again comparable to Gauss 2 and Lobatto IIIA, but at much higher computational costs. To interpret the convergence plots we will use the eigenvalues of the linearized operator (Jacobian)  $-C(u) + \nu D$  at  $t = 1$ . Figure 7.13 shows for each case the largest magnitude eigenvalue in the complex plane (they lie on the real axis) and the amplification factor of each method for  $\Delta t = 1$ . These eigenvalues scale as  $\lambda \sim \frac{1}{\text{Re}} \frac{1}{h^2}$ , with  $h$  the smallest mesh size and  $\text{Re} = 1/\nu$  the Reynolds number. If  $h$  is (locally) of the order as required by the mesh Péclet condition ( $h \sim 2/\text{Re}$ ), the eigenvalues scale as  $\lambda \sim \text{Re}$ .

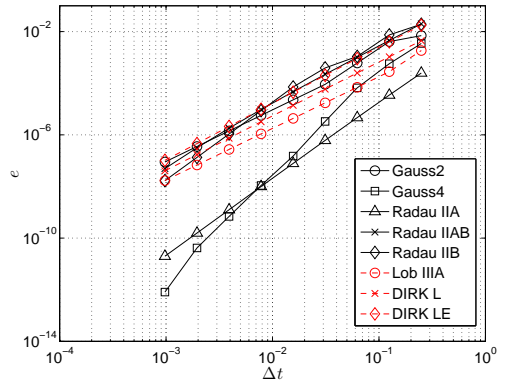
In case 1 the mesh is coarse and the position of the eigenvalues reveals that the problem is not very stiff. All methods obtain the theoretical rate of convergence, as was the case for the Taylor-Green vortex. Radau IIA/B is, somewhat surprisingly, less accurate than Radau IIB. Intuitively one might think that Radau IIA/B should lie in between Radau IIA and IIB, but this is not necessarily the case. The local truncation error of Radau IIA/B can be higher due to the presence of error terms that result from the additional (coupling) order conditions.

Upon refining the mesh, case 2, the problem becomes stiffer and the eigenvalues shift away from the imaginary axis. Gauss 4 still attains its theoretical convergence order, but shows a reduced order (namely 2) at large time steps, and is only more accurate than Radau IIA for sufficiently small time steps. Radau IIA/B is slowly converging towards its asymptotic rate, but is not more accurate than the second order methods over the range of time steps considered here.

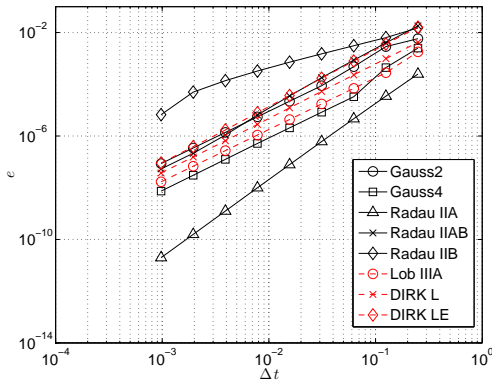
In case 3, the mesh is refined near the walls and the problem becomes very stiff. The ratio of the smallest mesh size between the uniform and cosine grid is approximately 65, and the eigenvalues shift with this factor from case 2 to case 3 as is observed in figure 7.13. All methods now converge according to the orders for stiff problems mentioned in section 7.3; Gauss 2:  $\mathcal{O}(\Delta t^{\eta+1}) = \mathcal{O}(\Delta t^2)$ , Gauss 4:  $\mathcal{O}(\Delta t^{\eta}) = \mathcal{O}(\Delta t^2)$ , Radau IIB:  $\mathcal{O}(\Delta t^{\eta}) = \mathcal{O}(\Delta t)$ , Radau IIA/B:  $\mathcal{O}(\Delta t^{\eta+1}) = \mathcal{O}(\Delta t^2)$ , DIRK L or L/E:  $\mathcal{O}(\Delta t^{\eta+1}) = \mathcal{O}(\Delta t^2)$ . Radau IIA and Lobatto IIIA are stiffly accurate and obtain their classical order  $p$ . Not surprisingly, Radau IIA is superior to Gauss 4. Radau IIA/B is now more accurate than Radau IIB (which has very poor behavior due to  $R(\infty) = 1$  and  $\eta = 1$ ), but it is not more accurate than Gauss 4 for the range of time steps considered here. Although Radau IIA/B is  $L$ -stable, its lower stage order compared to Gauss 4 prevents it to be more accurate. The second order methods all behave very similarly, making Gauss 2 and Lobatto IIIA most attractive due to the lowest computational cost. We stress once more that the observed order reduction is *not* a consequence of the fact that we are dealing with DAEs instead of ODEs, but a result of ‘ordinary’ stiffness that would also lead to order reduction if ODEs were considered.



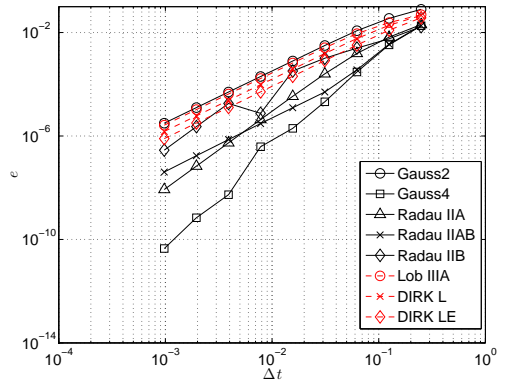
(a) case 1



(b) case 2



(c) case 3



(d) case 4

Figure 7.12: Velocity errors corner flow for cases 1-4.

A practically more important example of a convection-dominated flow, where the mesh spacing is close to ‘physical’ requirements, is obtained by increasing the Reynolds number to 1000, while keeping the grid fixed: case 4. The eigenvalues shift back towards the imaginary axis by a factor of 100. Lobatto IIIA loses its advantage over Gauss 2 compared to cases 1-3, where it was more accurate than Gauss 2. Figure 7.12d indicates that Gauss 4 and Radau IIA/B perform much better than in case 3. A ‘hump’ in the curves of Radau IIB and Gauss 4 around  $\Delta t = 10^{-2}$  indicates a transition from stage order to classical order (a similar hump appeared in case 2 at a roughly ten times larger  $\Delta t$ , corresponding to the ratio of eigenvalues of case 2 and 4). When increasing the Reynolds number further, with the mesh locally satisfying the Péclet condition, the eigenvalues will shift again away from the imaginary axis, increasing the stiffness of the problem. We expect therefore that for high Reynolds number flow the order reduction as observed in case 3 is likely to occur again. For practical computations it might be worthwhile to detect stiffness during running. For example, one can estimate the largest eigenvalue  $\lambda_{\max}$  of the Jacobian by Gershgorin’s theorem and compute the associated value of the stability function  $|R(\lambda_{\max}\Delta t)|$ . If this value is larger than some predefined threshold  $R_{\max}$ , e.g.  $R_{\max} = 0.999$ , then a code could switch to a stiffly-accurate method such as Radau IIA.

	case 1	case 2	case 3	case 4
Gauss 2, Lob IIIA	-0.98859	-0.99928	-1.00000	-0.99996
Gauss 4, Radau IIB	0.96617	0.99784	1.00000	0.99989
Radau IIA, IIA/B	$-5.6 \cdot 10^{-3}$	$-3.6 \cdot 10^{-4}$	$-1.8 \cdot 10^{-7}$	$-1.8 \cdot 10^{-5}$
DIRK L, L/E	$-1.4 \cdot 10^{-2}$	$-9.0 \cdot 10^{-4}$	$-4.4 \cdot 10^{-7}$	$-4.4 \cdot 10^{-5}$

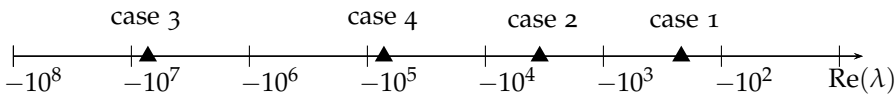


Figure 7.13: Largest magnitude eigenvalues on the real axis and corresponding amplification factors for cases 1-4.

Finally we consider a similar grid as in case 4, but change the inflow condition to a periodic one:

$$v(t) = -\sin\left(\pi(x^3 - 3x^2 + 3x)\right) e^{1-1/t^4} \left(\frac{1}{2} + \sin^2(t)\right) / 2e. \tag{7.70}$$

After an initial transient of approximately 5 time units, the inflow becomes periodic with period  $\pi$ , average 0.5 and amplitude 0.25. The domain is increased in size (see figure 7.9) to allow vortices to travel for some distance without being affected by the outflow boundary conditions. Based on the results of case 4 we select the Gauss



2, 4, Radau IIA and Radau IIA/B methods. In order to study the performance of the different methods, we look at the global kinetic energy  $K$  and the  $u$ -velocity component in point  $(0.2, 0.2)$  as function of time. The latter point lies inside an oscillating vortex and its value gives an indication how well the small scales of the flow are resolved in time. In all cases we use a very large time step,  $\Delta t = 1$ , meaning that *there are only 3 time steps to resolve a period of the oscillation*. Figures 7.14a-7.14b show  $K$  and  $u(0.2, 0.2)$  for these methods, together with reference results obtained at very small  $\Delta t$ . Surprisingly, even at this large time step, all methods agree reasonably well with the reference solution. When zooming in on the results of figure 7.14a, one finds that Gauss 4 is following the exact solution perfectly, followed closely by Radau IIA/B and Radau IIA. The good behavior of Gauss 4 can be explained by the fact that the problem is smooth in time, so that higher-order methods are effective. Gauss 2 is the least accurate, but this is not surprising since it is only second order in time. These conclusions become more pronounced when studying the small scale oscillations, figure 7.14b, where Gauss 2 starts to deviate strongly from the reference solution, while Gauss 4 and Radau IIA/B still perform very well. To obtain qualitatively similar results with Gauss 2 as with Gauss 4 the time step had to be reduced to approximately  $\Delta t = 1/4$ , increasing the number of time steps with a factor 4. Since the size of the matrix of the non-linear problem to be solved for Gauss 4 is  $(2(N_u + N_p))^2$  instead of  $(N_u + N_p)^2$ , a factor of 4 is generally not enough to offset the increase in computational costs, because the solution of the system is more expensive than  $\mathcal{O}(N_u + N_p)$ . Gauss 2 is then to be preferred. However, for a higher accuracy and smaller time steps this factor will rapidly increase and make Gauss 4 competitive.

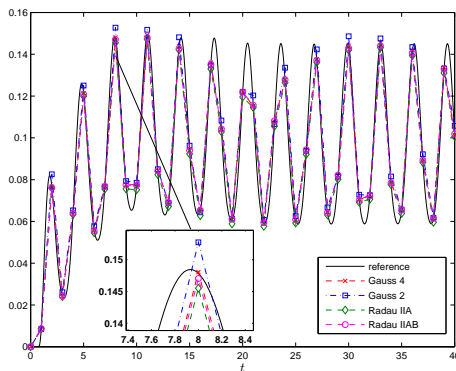
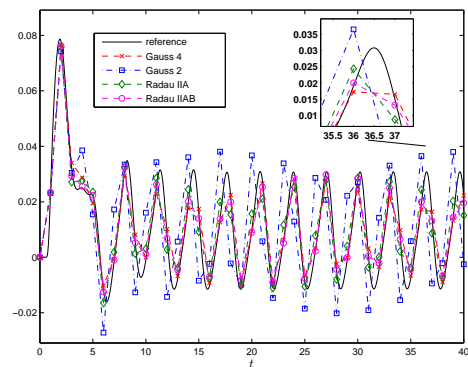
(a)  $K$ (b)  $u(0.2, 0.2)$ 

Figure 7.14: Periodic inflow, case 5.

## 7.9 CONCLUSIONS

In this chapter we have investigated energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations. Energy-conserving schemes are devoid of numerical viscosity, which is a favorable property for DNS and LES computations, where the Reynolds numbers are typically high and any viscosity should be as physical as possible. LES sub-grid models can then be tested without the interference of any artificial viscosity. Furthermore, the non-linear stability properties associated with energy conservation make these methods very appropriate for integrating over long periods of time, possibly with large time steps. The Gauss methods stand out since they are energy-conserving, time reversible and have the highest possible order given the number of stages. Methods based on Radau and Lobatto quadrature can also be transformed into energy-conserving methods (such as Radau IIB and Lobatto IIIE), but are of lower order.

The beneficial effect of energy conservation was shown by simulating the roll-up of an inviscid shear layer. The energy-conserving methods gave an accurate prediction of the enstrophy of the flow; in particular the two-stage Gauss and Radau IIB methods gave excellent results at very large time steps. Apparently, these methods have small error constants, which is at least as important as the order of accuracy of the method. Classical methods, such as explicit Runge-Kutta or multi-step methods, cannot combine high order with unconditional stability and inevitably add some artificial viscosity to the flow. The time-reversibility property of the time integration method was found to be unimportant by comparing Gauss 2 (time-reversible and energy-conserving) with Crank-Nicolson (time-reversible but not energy-conserving) and Radau IIB (energy-conserving but not time-reversible).

A disadvantage of all energy-conserving methods is that they lack  $L$ -stability, which makes them unsuitable if stiff components (such as boundary layers) are present in the semi-discrete equations. This leads to order reduction for energy-conserving methods of order higher than 2. We have therefore proposed a new class of high-order Runge-Kutta methods, that combine energy-conservation in the inviscid limit with  $L$ -stability in the stiff limit. We construct such methods by starting from a stiffly accurate ( $L$ -stable) Runge-Kutta method, transforming it with the method of Sun [188], and using these together to form a fully implicit additive Runge-Kutta (ARK) method. Although these new ARK methods lack the algebraic stability properties of their two constituents, they possess  $A$ -stability, which was sufficient for the test cases considered here. The highest possible order ( $p = 2s - 1$ ) ARK method with these properties is the Radau IIA/B method. Other additive methods derived in this chapter, being diagonally implicit or based on Lobatto quadrature, were not found to be competitive in terms of accuracy versus computational cost. The Radau IIA/B method gave excellent results for long time integration of a convection-dominated problem with considerable stiffness, employing large time steps. However, order reduction still appears for very stiff problems - not due to a lack of  $L$ -stability, but due to a lower stage order - and the asymptotic convergence rate is only reached for sufficiently small time steps. As a consequence,

the Radau IIA/B method does not outperform the Gauss method in the test cases considered here. For very stiff problems the issue of order reduction persists, and only stiffly-accurate methods perform well on such problems. *The construction of an 'all-round' high-order (additive) Runge-Kutta method, which is both energy-conserving and not suffering from order reduction for stiff problems, remains an open question.*

In the end, the choice for a method depends on the problem at hand and on the desired accuracy. We believe the Gauss 4 method is very suitable for sensitive flow phenomena that require high accuracy, such as DNS studies of the transition from laminar to turbulent flow, especially when long time intervals are considered. For many engineering problems, including wind-turbine wake studies, second order is probably sufficient and then the Gauss 2 method is the best candidate for turbulent flow simulations in which artificial viscosity should be avoided. This method has the additional advantage that it does not suffer from order reduction when applied to stiff problems.

For future work on large turbulent flow computations it is important to solve the non-linear saddle-point problem efficiently, which can be achieved by applying proper splitting and linearization techniques. Furthermore, it is crucial to perform adaptive time stepping in order to obtain efficient methods which adjust to the dynamics of the flow. This is not trivial, since variable step sizes can considerably deteriorate the good properties of symplectic methods (see e.g. [62] for an example, and [61] for remedies). We expect that the combined (space and time) fourth-order energy-conserving method leads to accurate simulation of turbulence, even with large time steps and coarse grids. In future work we will address such turbulent flows and compare the efficiency of the methods analyzed in this chapter with the current state-of-the-art methods, such as the IMEX Runge-Kutta method from [181].



## Part III

### ACTUATOR METHODS

This part describes actuator methods used to represent the action of bodies on the flow. A finite volume discretization of the immersed interface method is proposed as basis for these new actuator methods. The resulting actuator-disk, -line and -surface methods are applied to wind-turbine wake simulations.



---

*..the actuator disk, which we define as an artificial device producing sudden discontinuities in flow properties.. [71]*

*adapted  
from [147]*

## 8.1 INTRODUCTION

### 8.1.1 Background

In section 1.4.1 we described the generalized actuator disk approach as a method to model the effect of a wind turbine on the incoming flow. A forcing term represents the action of the body of the flow. As became clear from this section, in the wind energy community different researchers have found that upon introducing such a forcing term in a discrete setting care has to be taken to prevent unphysical instabilities and oscillations, see e.g. [175, 174]. Currently, the most common approach is based on the work of Sørensen et al. [177], who distribute the force on the actuator to the surrounding grid points by convolution with a regularization kernel (equation (1.14)). This technique is extended to an unsteady formulation employing actuator lines to represent rotating turbine blades [176]. Inherent to this approach is the choice of a parameter that controls the volume over which the forces are distributed. Computations with the actuator line method in [111] show that results can be sensitive to the value of this regularization parameter, especially near blade tips. Réthoré [140] shows that in a collocated variable arrangement a correction to the Rhie-Chow algorithm is necessary to avoid oscillations in pressure and velocity. The actuator surface approach of Sibuet-Watters et al. [167] uses surface forces instead of volume forces and does not require a regularization parameter. However, their approach is strongly rooted in inviscid, incompressible aerodynamic theory, having the downside that viscous effects are difficult, if not impossible, to take into account. Another issue is that the simple case of an actuator disk has to be modeled by prescribing the vorticity distribution of the slipstream surface (which in the inviscid case extends infinitely far downstream). Furthermore, the method requires the unknowns to be placed on the actuator surface, necessitating a ‘body-fitted’ mesh for the actuator surface. Rotating actuator surfaces are modeled by working in a rotating reference frame and introducing Coriolis forces in the equations.

To summarize, different types of forces (surface/volume forces) have been proposed with different discretization methods (finite difference/volume/element), different variables (velocity-pressure/velocity-vorticity etc.) and different variable layouts (collocated/staggered). Staggered grids and surface forces seem to be less susceptible to oscillations than other approaches.

### 8.1.2 Approach

We propose a discretization method that circumvents the issues mentioned above. The first important step is to realize that, from a physical point of view, *actuator forces are surface forces*. The second step is to realize that these *surface forces can introduce discontinuities* in flow variables and/or their derivatives. A finite volume discretization with a staggered grid layout, as described in chapter 3, is therefore well-suited to handle this problem: a finite volume approach is able to handle discontinuities, and the staggered grid has the advantage of strong pressure-velocity coupling.

The starting point for our discretization is the *immersed interface method* (IIM), first proposed by LeVeque and Li [96] for elliptic problems. Interfaces occur in different physical situations, such as problems involving membranes and materials with discontinuous properties. In our case the interface represents an actuator surface on which prescribed surface forces act. The IIM is a finite difference method that handles discontinuities arising from surface forces by taking jump conditions into the differencing stencil. The method has been extended to the incompressible Navier-Stokes equations and has been applied to a variety of problems, see e.g. [98]. The method is second-order accurate in the maximum norm and results in a sharp representation of an interface. In this chapter we propose a finite volume approach to the immersed interface method, instead of the commonly used finite difference approach. This is natural, since flows involving jumps should be tackled by using the weak (integral) formulation. We integrate the governing equations in space and time and show that this has a number of advantages over the finite difference approach.

Since in wind-turbine applications actuators generally have a prescribed motion and prescribed forcing, this work will focus on these situations. We are paying special attention to what happens at large time steps, since this is of practical interest when simulating rotating actuator lines. In those simulations the time step can be severely restricted by the fact that the tip of a blade should not cross more than a grid cell per time step. However, we stress that the methods described in this paper are also applicable to general situations in which the force on or the motion of an interface follows from the flow field.

The papers of Beyer and LeVeque [17] and LeVeque and Li [96] form the starting point for our work, providing a sound theoretical background and a number of one-dimensional test cases that we will repeat in this paper with our finite volume approach. The one-dimensional case, discussed in section 8.2, contains many ingredients that will be used for treating the two-dimensional incompressible Navier-Stokes equations, discussed in sections 8.3 and 8.4.



## 8.2 ONE-DIMENSIONAL PRELIMINARIES

8.2.1 *Steady*

As an introduction we consider, similar to LeVeque and Li [96], a one-dimensional elliptic equation with a singular source term:

$$u_{xx} = C\delta(x - \alpha). \quad (8.1)$$

The reason to study this equation as a model for the incompressible Navier-Stokes equations is explained in [96]. By integrating equation (8.1) from  $x = \alpha^-$  to  $x = \alpha^+$  we obtain the jump relation

$$[u_x] = C. \quad (8.2)$$

The solution is also allowed to jump, i.e.

$$[u] = \hat{C}. \quad (8.3)$$

This can be taken into (8.1) as the derivative of a Dirac function, but here it is taken as an external constraint (like in [96]). The jumps  $C$  and  $\hat{C}$  are assumed to be known.

Consider finite volumes  $\Omega_i, \Omega_{i+1}$ , with centers  $x_i, x_{i+1}$ , etc. The discontinuity at  $x = \alpha$  lies inside finite volume  $\Omega_{j+1}$ , as depicted in 8.1. The discretization for a general (interior) volume reads

$$(u_x)_{i+1/2} - (u_x)_{i-1/2} = 0, \quad i \neq j + 1. \quad (8.4)$$

For the finite volume containing the discontinuity we have

$$(u_x)_{i+1/2} - (u_x)_{i-1/2} = C, \quad i = j + 1. \quad (8.5)$$

The finite volume formulation has the advantage that the singular Dirac function is not present. Note that there is no approximation in these equations yet. Errors are introduced when approximating  $u_x$  at the faces in terms of  $u_i$ , the point values in the centers of the finite volumes (not volume average values). At an interior face we use the central approximation

$$(u_x)_{i+1/2} = \frac{u_{i+1} - u_i}{\Delta x}, \quad i \neq j, i \neq j + 1. \quad (8.6)$$

This approximation is second order accurate provided the underlying function  $u(x)$  is ‘smooth’ enough. If jumps are present in the solution or its derivative(s), additional terms are required to guarantee second order accuracy. These extra terms are found by considering *generalized Taylor expansions*. In appendix A.3 we detail the derivation for  $(u_x)_{j+1/2}$ , resulting in:

$$(u_x)_{j+1/2} = \frac{u_{j+1} - \hat{C} - C(x_{j+1} - \alpha) - u_j}{\Delta x}. \quad (8.7)$$

The local truncation error of this approximation is  $\mathcal{O}(\Delta x)$ . This is sufficient to ensure that the global error is  $\mathcal{O}(\Delta x^2)$  in the maximum norm (see e.g. [96]). This can be derived in a similar way as the local-global error analysis from chapter 4. For completeness, an approximation with a local truncation error of  $\mathcal{O}(\Delta x^2)$  is shown in appendix A.3. Note that so far we assumed that  $\alpha$  is to the *right* of  $x_{j+1/2}$ . If it is to the left, then  $x_{j+1}$  should be changed to  $x_j$  (i.e., the finite volume center closest to  $\alpha$ ).

Upon inserting the expressions for  $u_x$  into (8.4)-(8.5) we obtain

$$\frac{u_{j+2} - u_{j+1}}{\Delta x} - \frac{u_{j+1} - u_j}{\Delta x} + \frac{\hat{C} + C(x_{j+1} - \alpha)}{\Delta x} = C, \tag{8.8}$$

$$\frac{u_{j+1} - u_j}{\Delta x} - \frac{\hat{C} + C(x_{j+1} - \alpha)}{\Delta x} - \frac{u_j - u_{j-1}}{\Delta x} = 0, \tag{8.9}$$

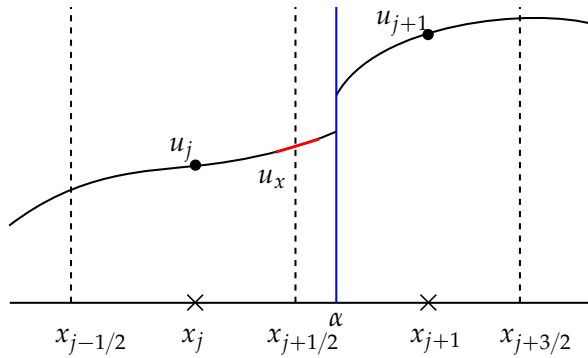
which can be rewritten as

$$\Delta_2 u_{j+1} = C - \underbrace{\frac{\hat{C}}{\Delta x} - \frac{C(x_{j+1} - \alpha)}{\Delta x}}_{\text{correction}}, \tag{8.10}$$

$$\Delta_2 u_j = \underbrace{\frac{\hat{C}}{\Delta x} + \frac{C(x_{j+1} - \alpha)}{\Delta x}}_{\text{correction}}, \tag{8.11}$$

where  $\Delta_2 u_j = \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x}$ . This discretization is the same as the finite difference discretization derived by LeVeque and Li [96]. We do not group the terms on the right-hand side in terms of  $C$  in order to stress the difference between the terms:  $C$  in equation (8.10) is the source term arising from the right-hand side of the differential equation;  $\frac{C(x_{j+1} - \alpha)}{\Delta x}$  is a correction term to make the evaluation of  $u_x$  more accurate. This distinction is less clear in the finite difference approach, where the entire right-hand side terms result from local truncation error considerations. Without the correction terms local truncation errors of  $\mathcal{O}(1)$  are introduced, reducing the global error to  $\mathcal{O}(\Delta x)$ . Note that the terms involving  $\hat{C}$  should not be regarded as corrections; they can be seen as discretization of the derivative of a Dirac function, and are the way in which the discrete system notices the jump in  $u$ . Without these terms the discrete solution will not converge to the correct solution. Upon summing the right-hand side of the discrete equations over the entire domain all correction terms cancel and we get  $C$ , like in the continuous case:

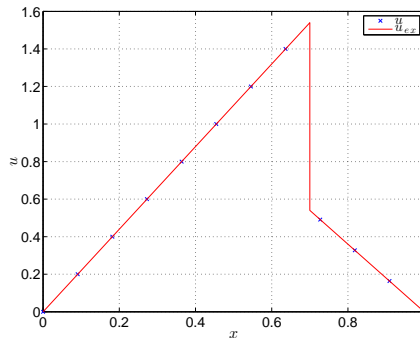
$$\text{continuous: } \int u_{xx} \, dx = C, \quad \text{discrete: } \sum_j \Delta_2 u_j = C. \tag{8.12}$$

Figure 8.1: Function  $u(x)$  with discontinuity at  $x = \alpha$ .

As a test case, we take a domain  $[0, 1]$  with homogeneous Dirichlet boundary conditions, with the exact solution:

$$u(x) = \begin{cases} ((\alpha - 1)C - \hat{C})x, & 0 < x < \alpha, \\ (\hat{C} - \alpha C)(1 - x), & \alpha < x < 1. \end{cases} \quad (8.13)$$

We take  $\alpha = 0.7$ ,  $\hat{C} = -1$ ,  $C = -4$ , and show the results in figure 8.2. The numerical solution and exact solution are equivalent; the error in the maximum norm is at machine precision. Without the correction terms in (8.10)-(8.11) the solution will be first-order accurate (in the maximum norm).

Figure 8.2: Exact and numerical solution to equation (8.1) with jumps in  $u$  and in  $u_x$ .

We note that the more general 1D problem discussed in [96]

$$(\beta u_x)_x + \kappa u = f + C\delta(x - \alpha), \quad (8.14)$$

can be handled similarly. Upon integration over a finite volume we need an expression for  $\int \kappa u \, dx$  in terms of the point values  $u_i$  and jumps  $C$  and  $\hat{C}$ . This is detailed in appendix A.3.

### 8.2.2 Unsteady, fixed source

We proceed with the unsteady case

$$u_t = u_{xx} + C(t)\delta(x - \alpha). \quad (8.15)$$

The strength of the singular source  $C(t)$  depends on time, but its position  $\alpha$  is still fixed. The jump in the derivative is given by  $[u_x(t)] = -C(t)$ , and the jump in the solution by  $[u(t)] = \hat{C}(t)$ . Instead of first integrating in space and subsequently discretizing the resulting equation in time (method of lines), (8.15) is integrated in *space and time* and then discretized. For finite volume  $\Omega_{j+1}$ , which contains the discontinuity, we get

$$\int_{t^n}^{t^{n+1}} \int_{\Omega_{j+1}} u_t \, d\Omega \, dt = \int_{t^n}^{t^{n+1}} \left( (u_x)_{j+3/2} - (u_x)_{j+1/2} \right) dt + \int_{t^n}^{t^{n+1}} C(t) \, dt. \quad (8.16)$$

For all other finite volumes ( $i \neq j + 1$ ) the integral involving  $C(t)$  does not contribute. Equation (8.16) is still exact. We will discuss how to approximate each term individually.

The first integral can be written as

$$\int_{t^n}^{t^{n+1}} \int_{\Omega_{j+1}} u_t \, d\Omega \, dt = \int_{t^n}^{t^{n+1}} \frac{d}{dt} \int_{\Omega_{j+1}} u \, d\Omega \, dt = \left[ \int_{\Omega_{j+1}} u \, d\Omega \right]_{t^n}^{t^{n+1}}. \quad (8.17)$$

Exchanging differentiation and integration is valid as long as the finite volume *and any discontinuities inside the finite volume* are not moving in time. The approximation of the spatial integral in the last term is detailed in appendix A.3, equation (A.44):

$$\int_{\Omega_{j+1}} u \, d\Omega = \Delta x u_{j+1} + \hat{C}(t)\Delta\alpha, \quad (8.18)$$

which constitutes an  $\mathcal{O}(h)$  approximation. The complete term becomes

$$\left[ \int_{\Omega_{j+1}} u \, d\Omega \right]_{t^n}^{t^{n+1}} = \left( \Delta x u_{j+1}^{n+1} + \hat{C}(t^{n+1})\Delta\alpha \right) - \left( \Delta x u_{j+1}^n + \hat{C}(t^n)\Delta\alpha \right) \quad (8.19)$$

$$= \Delta x (u_{j+1}^{n+1} - u_{j+1}^n) + (\hat{C}(t^{n+1}) - \hat{C}(t^n))\Delta\alpha. \quad (8.20)$$

The approximation of the second term is straightforward (but will become more complicated when the discontinuity is moving):

$$\int_{t^n}^{t^{n+1}} \left( (u_x)_{j+3/2} - (u_x)_{j+1/2} \right) dt = \int_{t^n}^{t^{n+1}} \left( \Delta_2 u_{j+1} + \frac{\hat{C}(t) - C(t)(x_{j+1} - \alpha)}{\Delta x} \right) dt. \quad (8.21)$$

The first term in this expression can be evaluated with any standard time-integration method; we use Crank-Nicolson:

$$\int_{t^n}^{t^{n+1}} \Delta_2 u_{j+1} dt = \frac{1}{2} \Delta t (\Delta_2 u_{j+1}^n + \Delta_2 u_{j+1}^{n+1}). \quad (8.22)$$

The second term can also be treated this way:

$$\frac{1}{h} \int_{t^n}^{t^{n+1}} \hat{C}(t) - C(t)(x_{j+1} - \alpha) dt = \frac{1}{2} \frac{\Delta t}{\Delta x} \left( \hat{C}(t^n) + \hat{C}(t^{n+1}) - (C(t^n) + C(t^{n+1}))(x_{j+1} - \alpha) \right). \quad (8.23)$$

The third term in (8.16) should be similarly approximated with a suitable numerical quadrature method; we use the trapezoidal rule:

$$\int_{t^n}^{t^{n+1}} C(t) dt = \frac{1}{2} \Delta t (C(t^n) + C(t^{n+1})). \quad (8.24)$$

The fully discrete system can thus be written as

$$\begin{aligned} \Delta x (u_{j+1}^{n+1} - u_{j+1}^n) + \underbrace{(\hat{C}(t^{n+1}) - \hat{C}(t^n)) \Delta \alpha}_{\text{corrections}} = \\ \frac{1}{2} \Delta t (\Delta_2 u_{j+1}^n + \Delta_2 u_{j+1}^{n+1}) + \frac{1}{2} \frac{\Delta t}{\Delta x} \left( \hat{C}(t^n) + \hat{C}(t^{n+1}) \right) + \frac{1}{2} \Delta t (C(t^n) + C(t^{n+1})) \\ - \underbrace{\frac{1}{2} \frac{\Delta t}{\Delta x} \left( C(t^n) + C(t^{n+1}) \right) (x_{j+1} - \alpha)}_{\text{corrections}}. \end{aligned} \quad (8.25)$$

For the case  $\hat{C} = 0$ , our method is equivalent to the discrete Dirac approach from Beyer and LeVeque [17] (with a hat function as discrete Dirac function) and the finite difference IIM in Li and Ito [97]. For  $\hat{C} \neq 0$ , our approach differs from the approach of Li and Ito [97] (this case is not discussed in [17]). Their approach requires  $\frac{d\hat{C}}{dt}$  ([97], p. 190), which is not needed in the current finite volume formulation. Our proposed method has therefore the advantage that less differentiability of the source terms is required.

We use a test case from Beyer and LeVeque to test our discretization (see [17], section 5.2). The exact solution is

$$u(x, t) = \begin{cases} \sin(3\pi x)e^{-9\pi^2 t}, & 0 \leq x \leq \alpha, \\ \sin(\frac{3}{2}\pi(1-x))e^{-\frac{9}{4}\pi^2 t}, & \alpha \leq x \leq 1. \end{cases} \quad (8.26)$$

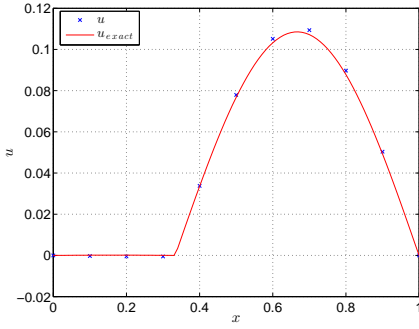
Beyer and LeVeque take  $\alpha = \frac{1}{3}$ . There is no jump in  $u$  ( $\hat{C} = 0$ ), but there is a jump in  $u_x$ :

$$C(t) = -[u_x] = -\frac{3}{2}\pi e^{-\frac{9}{4}\pi^2 t} - 3\pi e^{-9\pi^2 t}. \quad (8.27)$$

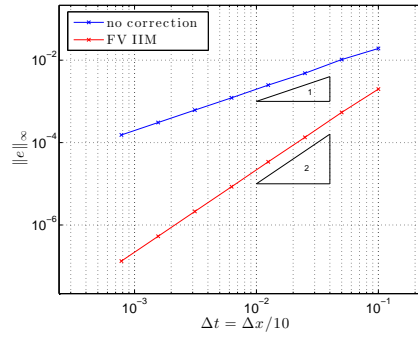
We integrate from  $t = 0$  to  $t = 0.1$  with  $\Delta t = \Delta x/10$ , so that we evaluate both the temporal and spatial order of accuracy at the same time. An example solution is shown in figure 8.3a. Figure 8.3b displays the error in the maximum norm, clearly indicating second order convergence. We also show what happens if we do not include correction terms: the accuracy drops to first order.

This test case is, however, not very stringent in the sense that both  $[u] = 0$  and  $[u_{xx}] = 0$ . We repeat the test with  $\alpha = \frac{1}{6}$ . Solution and error behavior, shown in figures 8.4a-8.4b, indicate again second order convergence. This confirms that the use of  $\frac{d\hat{C}}{dt}$  terms is not necessary.

As a prelude to the next section, which discusses *moving* sources, we investigate what happens when larger time steps are taken. Figure 8.5 shows for both  $\alpha = \frac{1}{3}$  and  $\alpha = \frac{1}{6}$  the solution behavior when  $\Delta t$  is increased to  $\Delta t = \Delta x$  and  $\Delta t = 2\Delta x$ . Apart from the (expected) fact that the numerical solution becomes less accurate, it also becomes oscillatory. This is especially evident in figure 8.5b.

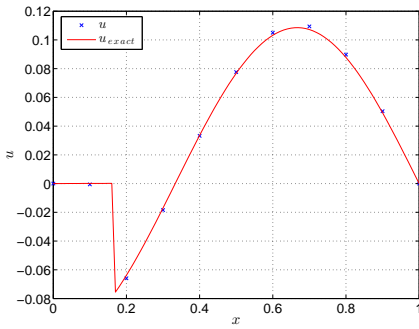


(a) Example solution,  $\Delta x = 1/10, t = 0.1$

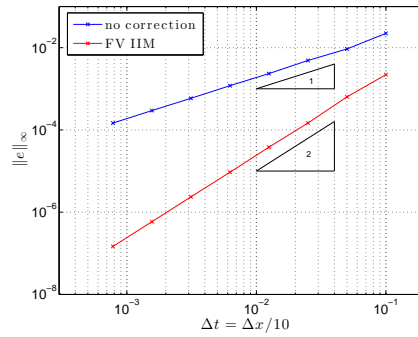


(b) Error behavior

Figure 8.3: Results for problem 5.2 in Beyer and LeVeque,  $\alpha = \frac{1}{3}$ , [17].

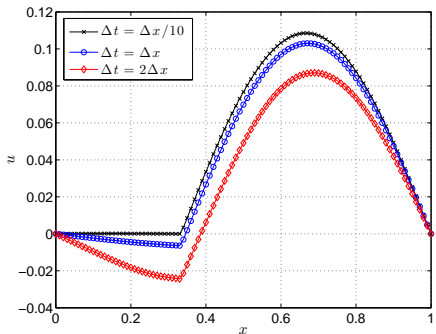


(a) Example solution,  $\Delta x = 1/10, t = 0.1$

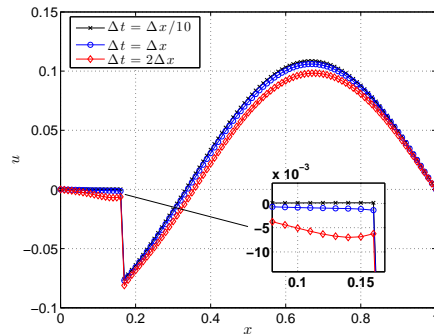


(b) Error behavior

Figure 8.4: Results for problem 5.2 in Beyer and LeVeque,  $\alpha = \frac{1}{6}$ , [17].



(a)  $\alpha = \frac{1}{3}$



(b)  $\alpha = \frac{1}{6}$

Figure 8.5: Qualitative behavior at large time steps,  $\Delta x = 1/100$ .

### 8.2.3 Unsteady, moving source

Finally, we conclude this one-dimensional analysis by considering a moving source:

$$u_t = u_{xx} + C(t)\delta(x - \alpha(t)). \tag{8.28}$$

An example of the movement of the source during one time step is shown in figure 8.6. As will become clear, the moving source problem is significantly more difficult than the stationary problem discussed in section 8.2.2. We consider again a stationary finite volume, and integrate (8.25) in *space and time*:

$$\int_{t^n}^{t^{n+1}} \int_{\Omega_{j+1}} u_t \, d\Omega \, dt = \int_{t^n}^{t^{n+1}} \left( (u_x)_{j+3/2} - (u_x)_{j+1/2} \right) dt + \int_{t^n}^{t^{n+1}} \int_{\Omega_{j+1}} C(t)\delta(x - \alpha(t)) \, d\Omega \, dt. \tag{8.29}$$

The discretization of each term will be discussed again separately.

#### 8.2.3.1 Unsteady term

We start with the first term in equation (8.29). It is tempting to follow equation (8.17). Care must be taken, however, when a moving discontinuity is present. Suppose the discontinuity at  $\alpha(t)$  splits the solution in two parts,  $u_1(x, t)$  ( $x_{j+1/2} \leq x \leq \alpha(t)$ ) and  $u_2(x, t)$  ( $\alpha(t) \leq x \leq x_{j+3/2}$ ), and  $[u(t)] = u_2(\alpha(t), t) - u_1(\alpha(t), t)$ . Then we define the volume integral

$$V(\alpha(t), t) = \int_{x_{j+1/2}}^{x_{j+3/2}} u \, d\Omega = \int_{x_{j+1/2}}^{\alpha(t)} u_1 \, d\Omega + \int_{\alpha(t)}^{x_{j+3/2}} u_2 \, d\Omega. \tag{8.30}$$

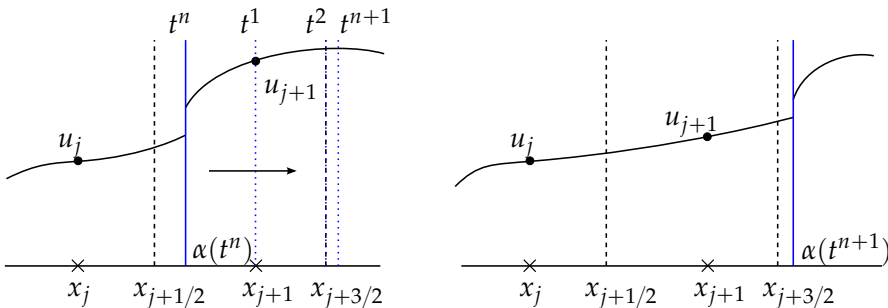


Figure 8.6: Moving discontinuity crossing a finite volume center (at  $t = t^1$ ) and a finite volume face (at  $t = t^2$ ). Left: solution at  $t^n$ , right: solution at  $t^{n+1}$ .



We are interested in the time derivative of this volume integral, which should be written as *partial* derivative, since the volume integral depends both on  $t$  and  $\alpha(t)$ . The partial derivative is related to the total derivative as follows. Let  $U_1(x, t)$  be the primitive of  $u_1(x, t)$ , and  $U_2(x, t)$  be the primitive of  $u_2(x, t)$ . Then

$$\begin{aligned} \frac{d}{dt} \int_{x_{j+1/2}}^{x_{j+3/2}} u \, d\Omega &= \frac{d}{dt} (U_1(\alpha(t), t) - U_1(x_{j+1/2}, t)) + \frac{d}{dt} (U_2(x_{j+3/2}, t) - U_2(\alpha(t), t)) \\ &= \dot{U}_1(\alpha(t), t) + \dot{\alpha}(t)u_1(\alpha(t), t) - \dot{U}_1(x_{j+1/2}, t) + \\ &\quad \dot{U}_2(x_{j+3/2}, t) - \dot{U}_2(\alpha(t), t) - \dot{\alpha}(t)u_2(\alpha(t), t) \\ &= \frac{\partial}{\partial t} \int_{x_{j+1/2}}^{\alpha(t)} u_1 \, d\Omega + \frac{\partial}{\partial t} \int_{\alpha(t)}^{x_{j+3/2}} u_2 \, d\Omega - \dot{\alpha}(t)[u(t)]. \end{aligned} \quad (8.31)$$

Upon time integration one obtains

$$\int_{t^n}^{t^{n+1}} \frac{\partial}{\partial t} \int_{\Omega_{j+1}} u \, d\Omega \, dt = \left[ \int_{\Omega_{j+1}} u \, d\Omega \right]_{t^n}^{t^{n+1}} + \int_{t^n}^{t^{n+1}} \dot{\alpha}(t) \hat{C}(t) \, dt, \quad (8.32)$$

revealing the presence of an additional term (compared to the case of a stationary discontinuity), involving the velocity of the discontinuity.

For the first integral on the right-hand side, equation (8.18) remains valid, so the approximation reads

$$\left[ \int_{\Omega_{j+1}} u \, d\Omega \right]_{t^n}^{t^{n+1}} = \Delta x (u_{j+1}^{n+1} - u_{j+1}^n) + \hat{C}(t^{n+1}) \Delta \alpha^{n+1} - \hat{C}(t^n) \Delta \alpha^n. \quad (8.33)$$

This assumes that  $\alpha(t^{n+1})$  and  $\alpha(t^n)$  are contained in  $\Omega_{j+1}$ . If  $\alpha$  has moved into a neighboring volume (like in figure 8.6) the corresponding correction term involving  $\hat{C}$  belongs to that volume. Equation (8.33) does essentially not contain a temporal discretization error; it contains a spatial discretization error, which is  $\mathcal{O}(\Delta x)$  as discussed in equation (A.43). The second integral on the right-hand side of equation (8.32) is discretized as

$$\int_{t^n}^{t^2} \dot{\alpha}(t) \hat{C}(t) \, dt = \frac{1}{2} (t^2 - t^n) \left( \dot{\alpha}(t^2) \hat{C}(t^2) + \dot{\alpha}(t^n) \hat{C}(t^n) \right), \quad (8.34)$$

which is an  $\mathcal{O}(\Delta t^2)$  approximation.

## 8.2.3.2 Fluxes

For the second term in equation (8.29) we focus on  $(u_x)_{j+3/2}$ . This term requires corrections when  $\alpha(t) \in [x_{j+1}, x_{j+2}]$ , so for  $t \in [t^1, t^{n+1}]$ :

$$\int_{t^n}^{t^{n+1}} (u_x)_{j+3/2} dt = \int_{t^n}^{t^{n+1}} \frac{u_{j+2} - u_{j+1}}{\Delta x} dt - \int_{t^1}^{t^2} \frac{\hat{C}(t) - C(t)(x_{j+1} - \alpha(t))}{\Delta x} dt - \int_{t^2}^{t^{n+1}} \frac{\hat{C}(t) - C(t)(x_{j+2} - \alpha(t))}{\Delta x} dt. \quad (8.35)$$

The first integral on the right-hand side should be treated carefully. When the discontinuity passes through point  $x_{j+1}$  the solution  $u_{j+1}$  jumps *in time*, and this has to be taken into account when approximating the integral. Following appendix A.3, equation (A.47):

$$\int_{t^n}^{t^{n+1}} u_{j+1} dt = \frac{1}{2} \Delta t (u_{j+1}^n + u_{j+1}^{n+1}) + \frac{1}{2} \llbracket u(t^1) \rrbracket ((t^{n+1} - t^1) - (t^1 - t^n)) - \frac{1}{2} \llbracket u_t(t^1) \rrbracket (t^{n+1} - t^1)(t^1 - t^n). \quad (8.36)$$

The *temporal jumps*  $\llbracket u \rrbracket$  and  $\llbracket u_t \rrbracket$  are given by (see e.g. [17])

$$\llbracket u \rrbracket = -\operatorname{sgn}(\dot{\alpha}(t)) [u], \quad (8.37)$$

and

$$\llbracket u_t \rrbracket = -\operatorname{sgn}(\dot{\alpha}(t)) [u_t] = -\operatorname{sgn}(\dot{\alpha}(t)) [u_{xx}]. \quad (8.38)$$

The second integral on the right-hand side of equation (8.35) can be approximated with the trapezoidal rule:

$$\int_{t^1}^{t^2} \frac{\hat{C}(t) - C(t)(x_{j+1} - \alpha(t))}{\Delta x} dt = \frac{1}{2} \frac{t^2 - t^1}{\Delta x} \left( \hat{C}(t^1) + \hat{C}(t^2) - C(t^1)(x_{j+1} - \alpha(t^1)) - C(t^2)(x_{j+1} - \alpha(t^2)) \right), \quad (8.39)$$

and the third integral follows in a similar fashion.

## 8.2.3.3 Source term

Finally we discuss the third term in (8.29). If  $\alpha(t)$  is contained in  $\Omega_{j+1}$  in interval  $[t^n, t^{n+1}]$ , then the double integral simplifies to

$$\int_{t^n}^{t^{n+1}} C(t) dt. \quad (8.40)$$

If  $\alpha(t)$  crosses a face of  $\Omega_{j+1}$ , as in figure 8.6, we proceed as follows:

$$\int_{t^n}^{t^{n+1}} \int_{\Omega_{j+1}} C(t)\delta(x - \alpha(t)) \, d\Omega \, dt = \int_{t^n}^{t^1} \int_{\Omega_{j+1}} C(t)\delta(x - \alpha(t)) \, d\Omega \, dt + \int_{t^1}^{t^{n+1}} \int_{\Omega_{j+1}} C(t)\delta(x - \alpha(t)) \, d\Omega \, dt. \quad (8.41)$$

At  $t^2$   $\alpha(t)$  crosses the right face of  $\Omega_{j+1}$  ( $\alpha(t^2) = x_{j+3/2}$ ). The second integral vanishes because  $\alpha(t)$  is not contained in  $\Omega_{j+1}$  - it will have a contribution to  $\Omega_{j+2}$ . The first integral simplifies to

$$\int_{t^n}^{t^{n+1}} \int_{\Omega_{j+1}} C(t)\delta(x - \alpha(t)) \, d\Omega \, dt = \int_{t^n}^{t^2} C(t) \, dt. \quad (8.42)$$

This can be discretized as

$$\int_{t^n}^{t^2} C(t) \, dt \approx \frac{1}{2}(t^2 - t^n) (C(t^n) + C(t^2)). \quad (8.43)$$

In our method each finite volume that is traced out during the movement of the source gets a contribution from the source. This idea will be extended to two dimensions in section 8.4, to develop a new discretization for moving interfaces in fluids.

#### 8.2.3.4 Space-time discretization and the method of lines

By discretizing the (in space and time) integrated form of the equations, we can handle the contribution of the moving source term in the equations accurately (even exactly for simple source functions). A consequence is, however, that the spatial and temporal discretization cannot be clearly separated: there is no semi-discrete system resulting from spatial discretization that can subsequently be integrated in time (i.e. the method of lines). Strictly speaking, the time discretization that we employ (trapezoidal rule / Crank-Nicolson) is therefore not a Runge-Kutta method. One might wonder if other Runge-Kutta methods (for example with better stability properties or higher order accuracy) can still be used in conjunction with our method, if there is no semi-discrete system to integrate. Fortunately, classic Runge-Kutta methods can still be applied in our situation. When using Runge-Kutta methods to integrate only part of the right-hand side of the equation (namely the flux term), we obtain a 'partial' Runge-Kutta method. Appendix B.4 shows that the order of accuracy of the partial Runge-Kutta method is the same as of the original Runge-Kutta method. This means that we can use, besides the Crank-Nicolson method used in the previous sections, any other Runge-Kutta method to discretize the equations in time.

### 8.2.3.5 Numerical experiments

We repeat example 5.4 from Beyer and LeVeque [17]. The exact solution is given by

$$u(x, t) = \begin{cases} \sin(\omega_1 x) e^{-\omega_1^2 t}, & x \leq \alpha(t), \\ \sin(\omega_2(1-x)) e^{-\omega_2^2 t}, & x \geq \alpha(t). \end{cases} \quad (8.44)$$

We take  $\omega_1 = 5\pi/4$  and  $\omega_2 = 7\pi/4$ , and integrate from  $t = 0$  to  $t = 0.1$ . We take a relatively coarse time step  $\Delta t = \Delta x/2$ , which is quite severe in the sense that during each time step finite volume faces or finite volume midpoints are traced out.

In a first test, case 1, we take  $\alpha(t)$  such that  $C(t) = -[u_x(t)] = 0$  (using a numerical root-finding algorithm). The jump in the solution can then be prescribed. The velocity of the interface at  $t = 0$  is  $\dot{\alpha}(0) \approx 0.93$  and decreases in time. An example solution for  $\Delta x = 1/25$  is shown in figure 8.7a. Figure 8.8 shows the error as a function of the time step, which clearly converges with second order upon mesh refinement.

In a second test, case 2, we take  $\alpha(t)$  such that  $\hat{C}(t) = [u(t)] = 0$ , and  $C(t) = -[u_x(t)]$  follows from the exact solution. The velocity of the interface at  $t = 0$  is  $\dot{\alpha}(0) \approx 1.8$  and decreases in time. An example solution and the error behavior are shown in figures 8.7b and 8.8.

In a third test, we investigate the effect of ‘sweeping’ out multiple volumes per time step by taking even larger time steps,  $\Delta t = \Delta x$  and  $\Delta t = 2\Delta x$ . With the latter time step as much as 4 finite volumes are traced out during a single time step. We take 100 finite volumes so the error is dominated by its temporal part. Like in section 8.2.2, the results in figure 8.9 show that our proposed method (with Crank-Nicolson as basic time integration scheme) leads to oscillatory results. Oscillations appear at the left side of the discontinuity; this is the region where the interface has passed and correction terms have acted. Three concepts are important in understanding these oscillations and reducing them: *monotonicity*, *accuracy* and *stability*.

Regarding monotonicity, oscillations can be avoided by taking  $\Delta t \leq \Delta x^2$ ; this is a *sufficient* condition for Crank-Nicolson to be monotone for the heat equation (see e.g. [74]). Basically, any implicit time integration method suffers from such a (stringent) condition on the time step (an exception is Backward Euler, which is monotone for any time step). However, this condition is often too strict in the sense that it is not always *necessary*. It depends on the smoothness of the problem. In this test case we experimentally observed that as long as  $\Delta t < \Delta x/\dot{\alpha}$  the solution remains essentially free of oscillations.

Regarding accuracy, a condition  $\Delta t = \mathcal{O}(\Delta x)$  is natural if the spatial and temporal discretizations are  $\mathcal{O}(\Delta x^2)$  and  $\mathcal{O}(\Delta t^2)$ , respectively. If the temporal accuracy is only  $\mathcal{O}(\Delta t)$ , as is the case with Backward Euler, one would require  $\Delta t = \mathcal{O}(\Delta x^2)$  to have spatial and temporal errors of similar magnitude.

Regarding (linear) stability, Crank-Nicolson does not have a time step restriction. This means that, even though oscillations can appear, they do not blow up in time.

Note that explicit methods would have a stability time step restriction of  $\Delta t = \mathcal{O}(\Delta x^2)$ .

#### 8.2.4 *Summary*

In this section we have developed a finite volume discretization in space and time for treating moving discontinuities in the context of the heat equation. By properly deriving spatial and temporal correction terms for each term second order accuracy in the maximum norm is obtained. It is sufficient that these correction terms have a local truncation error of  $\mathcal{O}(\Delta x)$  near the interface. The advantage of our finite volume formulation is that less differentiability of the problem is required than in a finite difference formulation (see [17, 97]). Furthermore, the method is intuitive since it tracks the finite volumes that are intersected by the moving interface. Lastly, the method is somewhat more straightforward than that presented in [17, 97], since we derive  $\mathcal{O}(\Delta x)$  approximations for each term in the differential equation near the interface, instead of deriving correction terms such that the total local truncation is first order accurate (as in [17]).

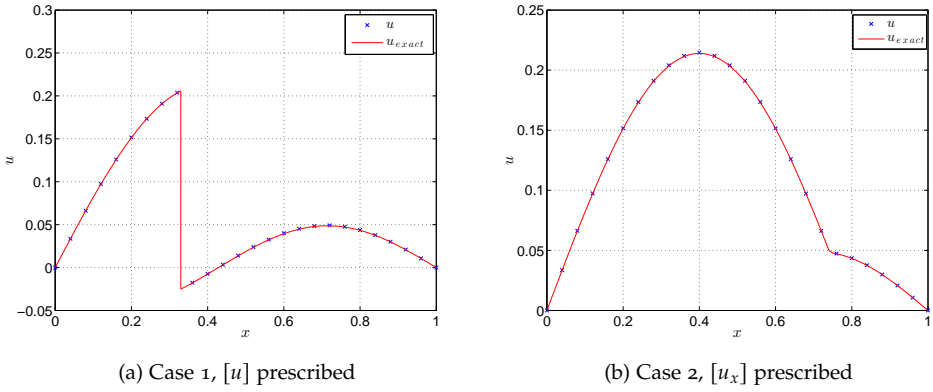


Figure 8.7: Example solutions,  $\Delta x = 1/25$ ,  $t = 0.1$ .

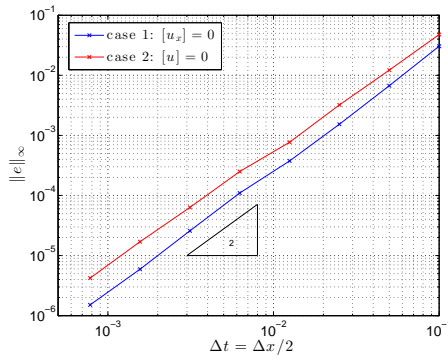


Figure 8.8: Error in numerical solution for case 1 and case 2.

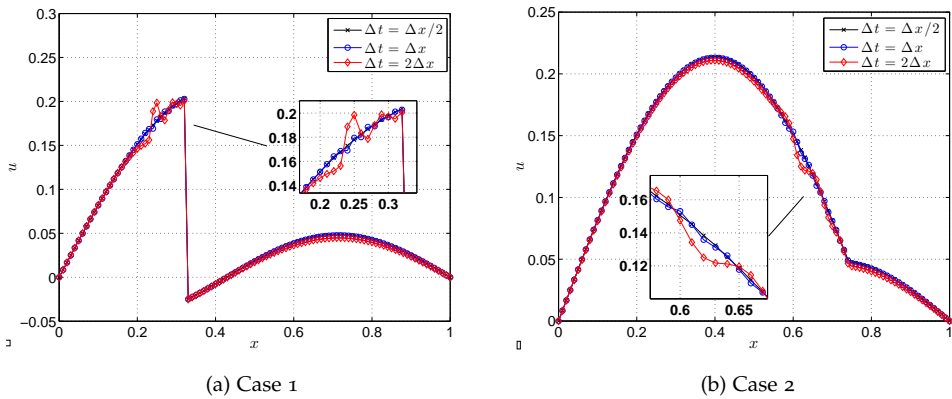


Figure 8.9: Qualitative behavior at large time steps,  $\Delta x = 1/100$ .

## 8.3 INCOMPRESSIBLE NAVIER-STOKES, STATIONARY FORCING TERMS

In this section we extend the methodology presented for the one-dimensional unsteady heat equation in section 8.2 to the two-dimensional incompressible Navier-Stokes equations.

8.3.1 *Continuous*

We consider the incompressible Navier-Stokes equations with a source term (a body force  $F$ ) in *integral form*,

$$\int_{\Gamma} \mathbf{u} \cdot \mathbf{n} \, d\Gamma = 0, \quad (8.45)$$

$$\int_{\Omega} \rho \frac{\partial \mathbf{u}}{\partial t} \, d\Omega + \int_{\Gamma} \rho \mathbf{u} \mathbf{u} \cdot \mathbf{n} \, d\Gamma = \int_{\Gamma} -p \mathbf{n} \, d\Gamma + \int_{\Gamma} \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \cdot \mathbf{n} \, d\Gamma + F, \quad (8.46)$$

where  $\Omega$  is a control volume with boundary  $\Gamma$ , see figure 8.10a. As before, we use the integral form as starting point since we will be dealing with discontinuities. We consider  $F$  to be a *surface force* that acts on an immersed interface  $S$ . It can be written as a surface integral of stresses (e.g. pressure or shear stresses):

$$F = \int_{S \cap \Omega} \mathbf{f}^b(\boldsymbol{\xi}(s)) \, dA \left( = \int_{S \cap \Omega} \mathbf{f}^b(\boldsymbol{\xi}(s)) \left| \frac{d\boldsymbol{\xi}}{ds} \right| ds \right), \quad (8.47)$$

where  $\mathbf{f}^b$  is a *surface stress* (force per unit area) which depends on the position on the surface  $\boldsymbol{\xi}$  parameterized by  $s$ ; see figure 8.10a.

Discontinuities arise due to the presence of surface forces. Decomposing  $\mathbf{f}^b = (\mathbf{f}^b \cdot \mathbf{n})\mathbf{n} + (\mathbf{f}^b \cdot \boldsymbol{\tau})\boldsymbol{\tau} = f_n^b \mathbf{n} + f_{\tau}^b \boldsymbol{\tau}$ , the resulting jump conditions read [98]:

$$[\mathbf{u}] = \mathbf{0}, \quad (8.48)$$

$$[p] = f_n^b, \quad (8.49)$$

$$\left[ \mu \frac{\partial \mathbf{u}}{\partial n} \right] = -f_{\tau}^b \boldsymbol{\tau}. \quad (8.50)$$

The velocity field through the interface is continuous; the pressure jumps due to normal forces on the interface, and the normal derivative of the tangential component of the velocity jumps due to tangential (shear) forces. Higher-order jump conditions can also be derived, see for example [213]. We will limit ourselves here to (prescribed) normal forces, so that only the pressure jumps. When drawing an analogy with the one-dimensional situation, the pressure compares to  $u_x$ , and  $\mathbf{f}^b$  compares to  $C$ .

In most existing work on immersed interface methods (and immersed boundary methods) the *differential form* of the incompressible Navier-Stokes equations is used:

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}^{fl}. \quad (8.51)$$

The (singular) force term  $\mathbf{f}^{fl}$  is related to  $\mathbf{f}^b$  via

$$\mathbf{f}^{fl}(\mathbf{x}) = \int_S \mathbf{f}^b(\boldsymbol{\zeta}(s)) \delta(\boldsymbol{\zeta}(s) - \mathbf{x}) \, dA. \quad (8.52)$$

In immersed boundary methods (e.g. [131]), the force  $\mathbf{f}^b$  is called the force on the body, and  $\mathbf{f}^{fl}$  the force on the fluid; the mapping between these two is performed by a convolution with a Dirac function  $\delta(\mathbf{x})$ . Integrating  $\mathbf{f}^{fl}(\mathbf{x})$  over the domain  $\Omega$  leads to

$$\int_{\Omega} \mathbf{f}^{fl}(\mathbf{x}) \, d\Omega = \int_S \mathbf{f}^b(\boldsymbol{\zeta}(s)) \, dA = \mathbf{F}. \quad (8.53)$$

So, the total force is the same in both formulations (as long as  $S$  is fully contained in  $\Omega$ ). By basing our finite volume discretization on equation (8.46), no discrete Dirac function will be required.

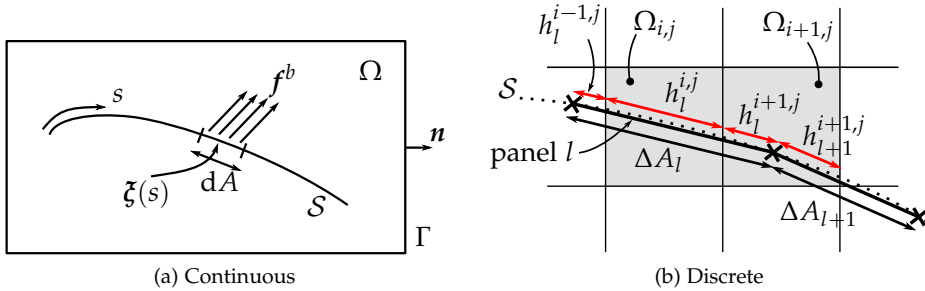


Figure 8.10: Surface force acting on an interface  $S$  in a fluid in a volume  $\Omega$ .

### 8.3.2 Discrete

First we discretize an interface by representing it by  $N$  piecewise linear segments ('panels'). The integral (8.47) changes into a summation over these panels:

$$\mathbf{F} = \sum_{l=1}^N \mathbf{f}_l^b \Delta A_l, \quad (8.54)$$

where  $\Delta A_l$  is the area of panel  $l$  and  $\mathbf{f}_l^b$  the surface stress acting on it.



With the interface discretized in panels, we apply equation (8.46) to a finite volume. Exactly like in the continuous case, a volume  $\Omega_{i,j}$  gets a contribution from  $f_l^b$  if  $\Omega_{i,j}$  is intersected by the body  $\mathcal{S}$  (see figure 8.10b). The force on finite volume  $\Omega_{i,j}$  is simply given by

$$F_{i,j} = \sum_{l=1}^N f_l^b h_l^{i,j}, \quad (8.55)$$

where  $h_l^{i,j}$  is the area of panel  $l$  contained in volume  $\Omega_{i,j}$ .

Note again, that when employing the differential form of the Navier-Stokes equations (equation (8.51)), the force is introduced based on a discrete version of equation (8.52):

$$f_{i,j}^{fl} = \sum_{l=1}^N f_l^b \Delta A_l d(\xi_l - x_{i,j}), \quad (8.56)$$

where  $d(x)$  is a *discrete* Dirac function;  $\xi_l$  is the midpoint of panel  $l$  and  $x_{i,j}$  is the midpoint of finite volume  $(i,j)$ . The Dirac function distributes the force from the interface to the surrounding mesh. Different discrete Dirac functions have been proposed in the literature, such as ‘hat’, ‘cosine’, or ‘Gaussian’ distributions, see e.g. [17]. These discrete Dirac functions generally have a free parameter  $\varepsilon$  (typical values of  $\varepsilon$  are in the range  $[\Delta x, 3\Delta x]$ ), that determines the width of the Dirac distribution and as such the amount of ‘smearing’ in transferring the force on the body to the fluid. Our formulation (8.55) is free of such a parameter.

### 8.3.3 Handling discontinuities

Equation (8.55) is, like equation (8.5) for the one-dimensional case, exact (except for the error introduced due to discretizing the interface in piecewise linear segments). However, like in the one-dimensional case, the presence of discontinuities has to be properly taken into account in the finite volume discretization of the other terms in the incompressible Navier-Stokes equations.

We focus on the important case of a force  $f_n^b$  normal to the interface. This normal force leads to a jump in the pressure as indicated by equation (8.49), so the discretization of the pressure gradient term in equation (8.46) has to be adapted. Consider an interface crossing a finite volume as illustrated in figure 8.11a. We use a staggered grid, with the pressure points at the east and west faces of the finite volume for the  $u$ -component. Without the presence of the interface (so without discontinuities) the pressure force acting on the left side of this volume would be given by the second-order approximation

$$\int p n_x dy = p_{i,j} \Delta y. \quad (8.57)$$

However, if the pressure is discontinuous on the face this approximation is not very accurate. Like in the one-dimensional case (equation (A.44)), a more accurate approximation is obtained by including the pressure jump  $[p]$  into the discretization:

$$\int p n_x dy = p_{i,j} \Delta y + \underbrace{[p] \Delta g}_{\text{correction}}, \tag{8.58}$$

with  $\Delta g$  defined in figures 8.11a-8.11c. As shown by equation (A.43) this is a first-order accurate approximation. Based on the results from section 8.2, and results from literature [96], it is expected that this is sufficient for global second-order accuracy in the maximum norm.

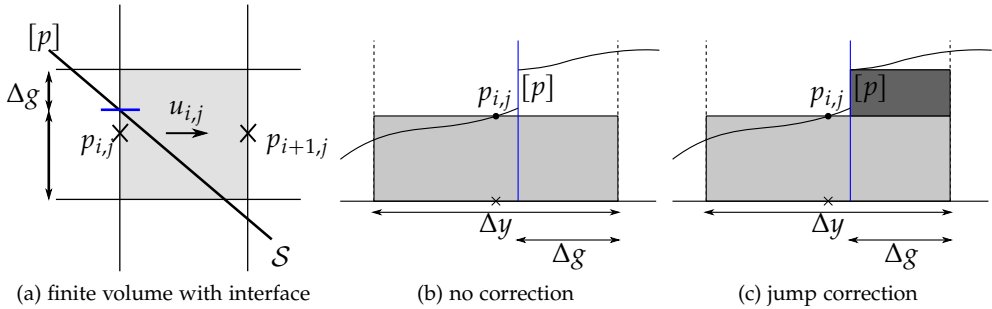


Figure 8.11: Pressure gradient correction.

It is instructive to interpret the correction term in (8.58) as a correction to (the  $x$ -component of) the force  $F_{i,j}$ . The effective force is:

$$(\tilde{F}_x)_{i,j} = (F_x)_{i,j} + [p] \Delta g. \tag{8.59}$$

The sum of the correction terms over the entire domain is zero, so that the total applied force remains unchanged.

### 8.3.4 Extension to fourth order

The discretization derived in sections 8.3.2-8.3.3 was based on a second order staggered discretization. The extension to the fourth order method introduced in chapter 3 is as follows. Considering that this fourth order method is a combination of two second order methods (on a fine and coarse finite volume), the total force in the fourth order method should also be constructed as such:

$$\mathbf{F}_{i,j} = \alpha_1 (\mathbf{F}_1)_{i,j} - (\mathbf{F}_3)_{i,j} = \sum_{l=1}^N \mathbf{f}_l^b (\alpha_1 (h_1)_l^{i,j} - (h_3)_l^{i,j}), \tag{8.60}$$

where  $(h_1)_l^{i,j}$  and  $(h_3)_l^{i,j}$  are the areas of panel  $l$  contained in volumes  $\Omega_1$  and  $\Omega_3$ , respectively. Figure 8.12 shows for the simple case of a vertical interface how  $(\Omega_3^u)_{i,j}$  and  $(\Omega_3^u)_{i+1,j}$  get a force contribution from the interface passing through  $(\Omega_1^u)_{i,j}$ . Consequently, when plotting the force as function of  $x$ , it is not concentrated in a single volume (as in the second order case), but distributed over three ‘effective’ volumes, see figure 8.13. This does not smear the solution, because the pressure gradient operator also has a wider stencil than in the second order case. The discontinuity in the pressure is therefore also exact for the fourth order method, as can be seen from figure 8.13b. Note that the pressure and force are defined at different positions due to the staggering of the grid.

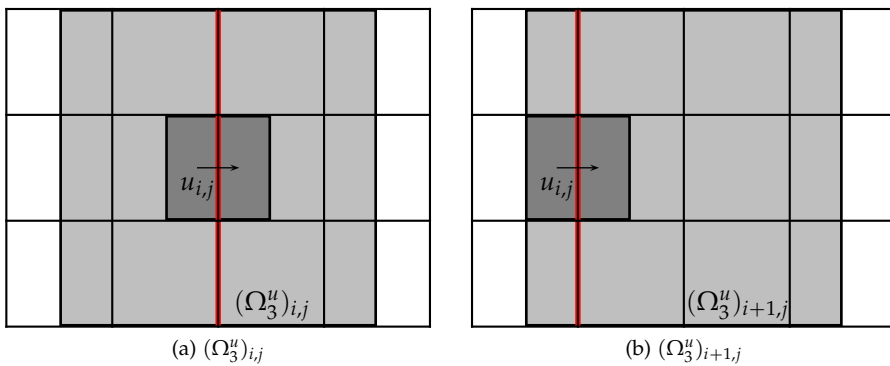


Figure 8.12: Coarse finite volumes obtaining a contribution from the force in  $u_{i,j}$ . Red line denotes the interface.

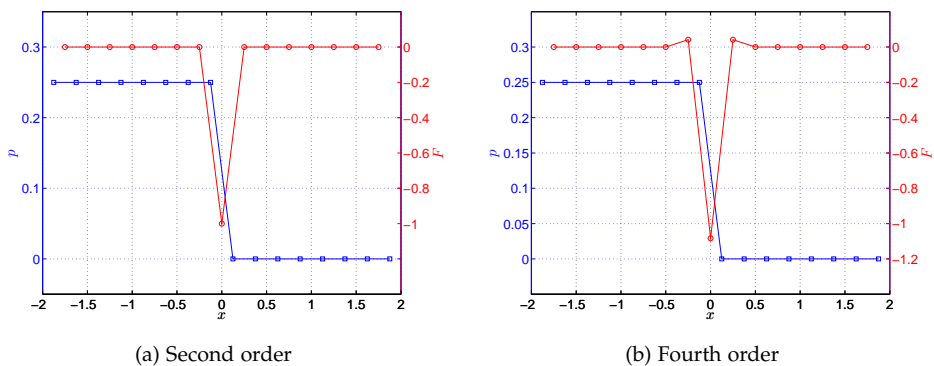


Figure 8.13: Pressure and force behavior for one-dimensional test problem.

### 8.3.5 Contribution to the energy equation

The immersed interface method is implemented in our energy-conserving spatial discretization method. The body forces present in the momentum equation give a contribution to the energy of the flow as shown by equation (2.26). For the case of immersed interfaces with surface forces, this contribution becomes

$$(\mathbf{f}^{fl}, \mathbf{u}) = \int_{\Omega} \left( \int_S \mathbf{f}^b(\boldsymbol{\zeta}(s)) \delta(\boldsymbol{\zeta}(s) - \mathbf{x}) dA \right) \cdot \mathbf{u}(\mathbf{x}) d\Omega = \int_S \mathbf{f}^b(\boldsymbol{\zeta}(s)) \cdot \mathbf{u}(\boldsymbol{\zeta}(s)) dA. \quad (8.61)$$

The energy remains strictly decreasing if  $(\mathbf{f}^{fl}, \mathbf{u}) < 0$ . For actuator methods, such as an actuator disk,  $\mathbf{f}$  is generally in the opposite direction of the velocity, and  $(\mathbf{f}^{fl}, \mathbf{u}) < 0$  is valid. An upper bound for the energy is given by equation (2.27). However, since actuators *extract* energy of the flow, this bound is not very sharp.

### 8.3.6 Results

#### 8.3.6.1 An elliptical membrane with a pressure jump

We repeat example 2 from Li and Lai [98]: a fixed interface defined by the ellipse

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1, \quad (8.62)$$

on which the normal ‘stress’  $f_n = -1$  acts. The exact solution to this problem is

$$u = v = 0, \quad (8.63)$$

$$p = \begin{cases} C - 1, & \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 > 1, \\ C, & \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 < 1, \end{cases} \quad (8.64)$$

where  $C$  is an arbitrary constant which we set at  $C = 1$ . Like in [98] we take  $a = 0.35$  and  $b = 0.25$ . The domain is  $[-1, 1] \times [-1, 1]$ . The pressure gradient correction terms from equation (8.58) are shown in figure 8.14; they can be interpreted as additional body forces. The solution for the pressure on a  $40 \times 40$  mesh is shown in figure 8.15. In contrast to the method of [98] our discretization is *exact* for this problem. This is independent of the number of finite volumes used, independent of the number of points used to describe the ellipse (here we used 100), independent of the values of  $a$  and  $b$ , and also holds on non-uniform grids. It can be explained by the fact that approximation (8.58) is exact when the solution is piecewise constant. Without the pressure gradient correction terms the solution is not exact and behaves oscillatory near the interface.

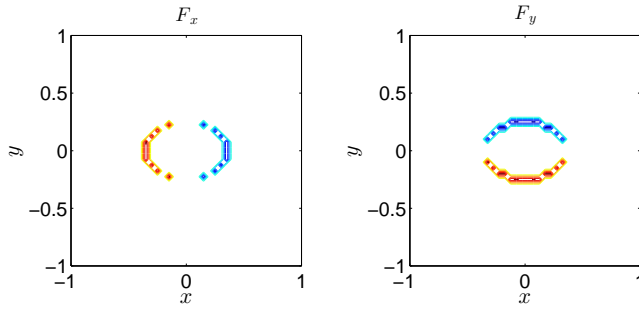


Figure 8.14: Pressure gradient correction terms for elliptical interface problem. Contour lines  $-0.05 \dots 0.05$  with steps 0.01.

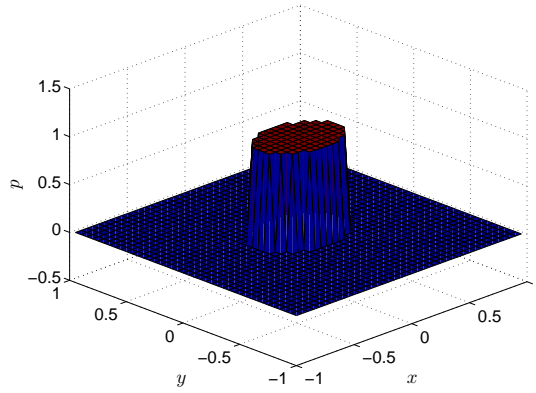


Figure 8.15: Discontinuous pressure solution for elliptical interface problem.

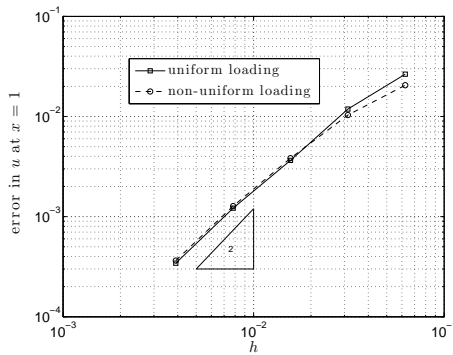


Figure 8.16: Error in velocity profile at  $x = 1$  for flow through actuator disk.

## 8.3.6.2 An actuator disk

As a second example we consider the flow through an actuator disk, a practically relevant test for simulating the flow of air in wind turbine wakes.

First we perform an order of accuracy study of the immersed interface approach for an actuator disk perpendicular to the flow, either uniformly or non-uniformly loaded. In the uniformly loaded case we take  $f_n(r) = -\frac{1}{4}$ , in the non-uniformly loaded case we take [141]:

$$f_n(r) = -\frac{3}{2} \frac{r(r-R)}{R^2}, \quad (8.65)$$

where  $R = \frac{1}{2}$  is the radius of the disk, and  $r$  the distance from the disk center. The total loading is in both cases the same; the effective thrust coefficient is  $C_T = \frac{1}{2}$ . Figure 8.18 shows a flow field example obtained with  $Re = 100$  and  $h = 1/32$ . To compute the error we compare the velocity profile in the wake (at  $x = 1$ ) with the velocity profile computed on a fine mesh. The wake velocity profile is an important quantity, for example in the study of wind turbine wake interactions. The error behavior (in the maximum norm) as function of mesh size is shown in figure 8.16. The rate of convergence is approaching 2 upon mesh refinement.

Secondly, we do a qualitative comparison of the immersed interface approach with the discrete Dirac approach for an actuator disk at an angle of  $30^\circ$  with respect to the flow. Figure 8.19 shows the velocity and pressure contours. The wake behind the disk is deflected *downward*, since momentum is only extracted in a direction normal to the disk surface. The jump in pressure is again sharply captured. In figure 8.17 the two approaches are compared. We used a Gaussian as discrete Dirac function with  $\varepsilon = 2\Delta x$ ; this type of Dirac function and value of  $\varepsilon$  are typically used in wind-turbine wake computations, see e.g. [111]. Figure 8.17 shows that the immersed interface method is *much more accurate* than the discrete Dirac approach: the results at  $h = 1/4$  with the IIM are better than the results with the Dirac approach at  $h = 1/8$ . For this case we need a four times finer mesh when using Dirac functions to obtain an accuracy comparable to the IIM.

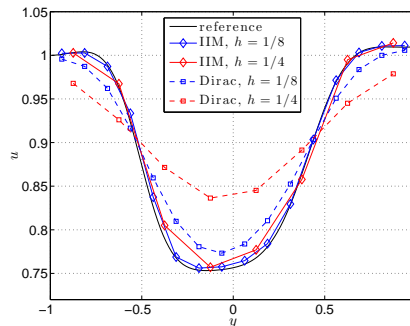


Figure 8.17: Comparison of IIM with discrete Dirac approach. Velocity profiles at  $x = 1$  for actuator at an angle of  $30^\circ$ .

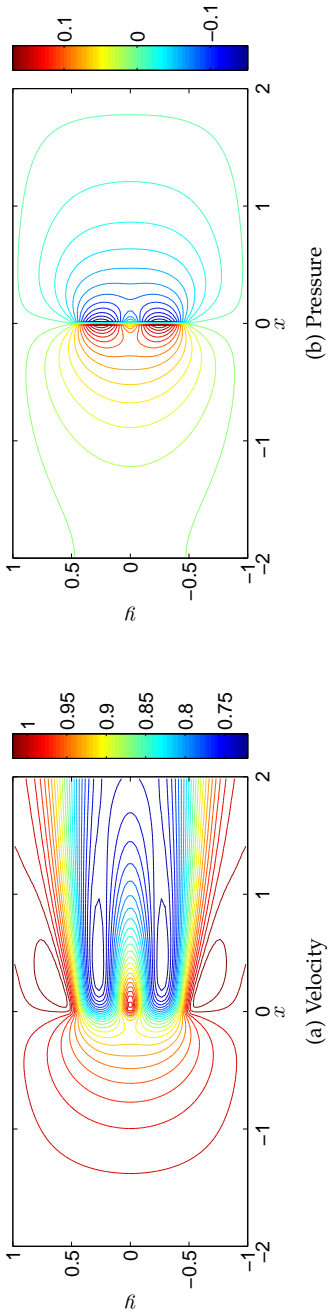


Figure 8.18: Velocity and pressure for non-uniformly loaded actuator disk.

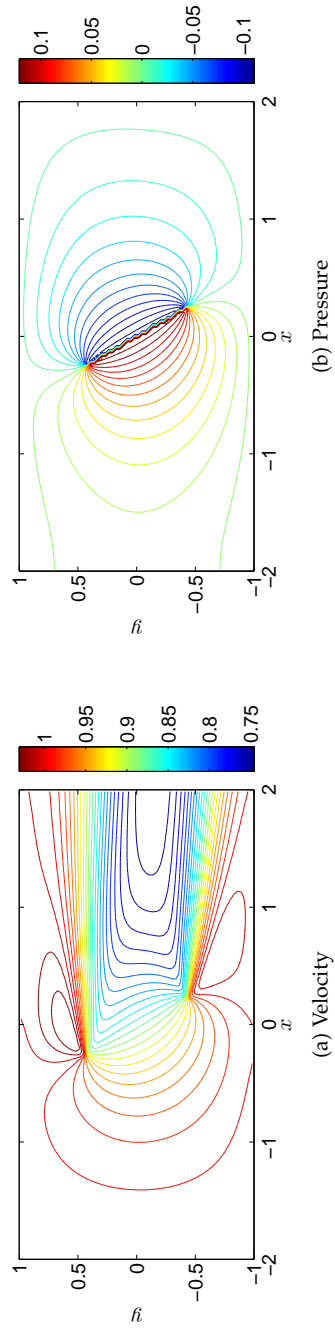


Figure 8.19: Velocity and pressure for actuator disk at angle of  $30^\circ$ .

## 8.4 INCOMPRESSIBLE NAVIER-STOKES, MOVING FORCING TERMS

8.4.1 *Continuous*

In the unsteady case, we proceed as in section 8.2.2 and integrate equation (8.46) in both space and time:

$$\left[ \int_{\Omega} \rho \mathbf{u} \, d\Omega \right]_{t^1}^{t^2} = \int_{t^n}^{t^{n+1}} \left( \int_{\Gamma} \left( -\rho \mathbf{u} \mathbf{u} + \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \right) \cdot \mathbf{n} - p \mathbf{n} \, d\Gamma \right) dt + \int_{t^n}^{t^{n+1}} \mathbf{F}(t) \, dt. \quad (8.66)$$

Exchanging differentiation and integration in the unsteady term is, in contrast to the one-dimensional case, valid, because no discontinuities in the velocity are present (see equation (8.48)). Treating discontinuities will be further discussed in section 8.4.3. In this section we focus on the last term in (8.66). We consider a rigid interface  $\mathcal{S}(t)$  moving in time. In two dimensions such an interface traces out an area  $V$  in time. This area is parameterized as  $\boldsymbol{\zeta}(s, t)$ , see figure 8.20a. The total impulse exerted by the actuator on the flow is given by

$$\mathbf{I} = \int_{t^n}^{t^{n+1}} \mathbf{F}(t) \, dt = \int_{t^n}^{t^{n+1}} \int_{\mathcal{S}} \mathbf{f}^b(\boldsymbol{\zeta}(s, t), t) \, dA \, dt = \int_{t^n}^{t^{n+1}} \int_{\mathcal{S}} \mathbf{f}^b(\boldsymbol{\zeta}(s, t), t) \left| \frac{\partial \boldsymbol{\zeta}}{\partial s} \right| ds \, dt. \quad (8.67)$$

The integral in terms of the parametric representation  $(s, t)$  can be written in terms of an integral in space. Noting that a differential element of size  $ds \, dt$  is related to an element  $dV = dx \, dy$  by

$$dx \, dy = \left| \frac{\partial \boldsymbol{\zeta}}{\partial s} \times \frac{\partial \boldsymbol{\zeta}}{\partial t} \right| ds \, dt, \quad (8.68)$$

the integral is rewritten as

$$\int_{t^n}^{t^{n+1}} \int_{\mathcal{S}} \mathbf{f}^b(\boldsymbol{\zeta}(s, t), t) \left| \frac{\partial \boldsymbol{\zeta}}{\partial s} \right| ds \, dt = \iint_V \mathbf{f}^b(\boldsymbol{\zeta}(s, t), t) \frac{\left| \frac{\partial \boldsymbol{\zeta}}{\partial s} \right|}{\left| \frac{\partial \boldsymbol{\zeta}}{\partial s} \times \frac{\partial \boldsymbol{\zeta}}{\partial t} \right|} dV. \quad (8.69)$$

Here  $\frac{\partial \boldsymbol{\zeta}}{\partial t} = \mathbf{w}(s, t)$  is the local velocity of the interface. In the cases that we consider,  $\mathbf{w}(s, t)$  will be *prescribed*, but this is not necessary. At any time  $t$  the derivative  $\frac{\partial \boldsymbol{\zeta}}{\partial s}$  is tangential to the actuator contour, so it can be written as

$$\frac{\partial \boldsymbol{\zeta}}{\partial s} = \left| \frac{\partial \boldsymbol{\zeta}}{\partial s} \right| \boldsymbol{\tau}(s, t). \quad (8.70)$$

Decomposing  $\mathbf{w}(s, t)$  in directions locally parallel and perpendicular to the actuator contour gives

$$\mathbf{w}(s, t) = w_{\tau}(s, t) \boldsymbol{\tau}(s, t) + w_n(s, t) \mathbf{n}(s, t), \quad (8.71)$$



so equation (8.67) becomes ( $\boldsymbol{\tau} \times \boldsymbol{\tau} = 0, |\boldsymbol{\tau} \times \mathbf{n}| = 1$ ):

$$I = \iint_V \frac{f^b(\mathbf{x})}{|w_n(\mathbf{x})|} dV. \tag{8.72}$$

In this way the integral in time has been transformed into an integral in space; this form will be employed in section 8.4.2 to derive a new discretization method. Equation (8.72) is also valid in three dimensions when  $V$  is interpreted as the volume swept out by the immersed interface in time.

Note that for  $w_n$  approaching zero (e.g. the steady case) this integral does not tend to infinity, because the area of integration  $V$  tends to zero. This can be observed directly from the original form  $\int F dt$  which reduces to  $F \Delta t$  for a steady actuator.

In one dimension we also integrated the source term in time, see e.g. equation (8.29). In that case the time integral can also be transformed according to (8.72). However, there is no apparent advantage to do so since both time and space are ‘one-dimensional’.

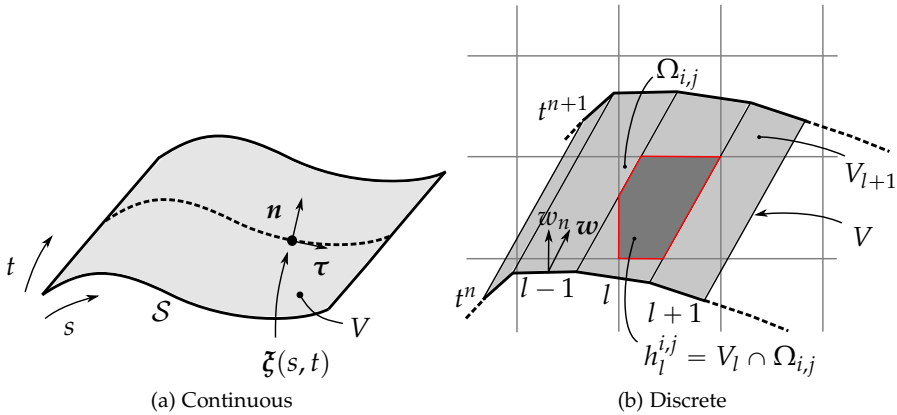


Figure 8.20: An interface moving in time (in two dimensions).

### 8.4.2 Discrete

The discrete version of equation (8.72) reads

$$I_{i,j} = \sum_l \iint_{V_l \cap \Omega_{i,j}} \frac{f_l(\mathbf{x})}{|w_n(\mathbf{x})|} dV, \tag{8.73}$$

where  $V_l$  is the volume swept out by panel  $l$  during a time step. In general,  $V_l \cap \Omega_{i,j} = \emptyset$  for most  $l$ . For example, figure 8.20b shows that  $\Omega_{i,j}$  is only intersected by panels  $l - 1, l$  and  $l + 1$ . Equation (8.73) is still *exact* since no discretization er-

rors have been introduced (except in discretizing the interface in panels). However, in general  $w_n$  and  $f$  are varying in time and space, and  $V_l \cap \Omega_{i,j}$  has a complex shape, so that the integral should be approximated numerically (this introduces a discretization error in conservation of total impulse added to the flow).

If the force and the normal velocity are constant during a time step and uniform over a panel, equation (8.73) becomes

$$\mathbf{I}_{i,j} = \sum_l \frac{f_l}{|w_n|} \iint_{V_l \cap \Omega_{i,j}} dV = \sum_l \frac{f_l}{|w_n|} h_l^{i,j}, \quad (8.74)$$

where  $h_l^{i,j}$  is the part of  $\Omega_{i,j}$  swept out by panel  $l$ , see figure 8.20b. In order to determine  $h_l^{i,j}$  we transform the volume integral into a surface integral with the divergence theorem:

$$h_l^{i,j} = \iint_{V_l \cap \Omega_{i,j}} dV = \iint_{V_l \cap \Omega_{i,j}} \nabla \cdot \mathbf{x} dV = \oint_{\partial(V_l \cap \Omega_{i,j})} \mathbf{x} \cdot \mathbf{n} dS. \quad (8.75)$$

The intersection between  $V_l$  and  $\Omega_{i,j}$  is a polygon (if the interface is only translating, and not rotating), and with (8.75) its area can be expressed by using the coordinates of the vertices of the polygon. Assuming a rigid actuator, the boundaries of  $V_l$  are found by tracking the begin and end points  $\mathbf{x}_l^1$  and  $\mathbf{x}_l^2$  of each panel:

$$\frac{d\mathbf{x}_l^1(t)}{dt} = \mathbf{w}_l^1(t), \quad \frac{d\mathbf{x}_l^2(t)}{dt} = \mathbf{w}_l^2(t), \quad (8.76)$$

and  $\mathbf{w}_l^1(t)$ ,  $\mathbf{w}_l^2(t)$  are assumed to be prescribed. (Note that  $\mathbf{x}_l^1(t) = \mathbf{x}_{l-1}^2(t)$ .)

Discretization (8.73) has a number of attractive properties compared to a ‘naive’ discretization of equation (8.66). Such a naive discretization could read, for example:

$$\int_{t^n}^{t^{n+1}} \mathbf{F}(t) dt = \frac{1}{2} \Delta t \left( \mathbf{F}(t^n) + \mathbf{F}(t^{n+1}) \right). \quad (8.77)$$

With this discretization only the finite volumes that are intersected by  $\mathbf{F}(t^n)$  and  $\mathbf{F}(t^{n+1})$  get a contribution. Our proposed discretization (8.73) tracks the movement of the interface, and each intersected finite volume will obtain a force contribution. We expect therefore that it is more suitable for large time steps.

### 8.4.3 Handling discontinuities

In section 8.2.3 we outlined how to treat moving discontinuities for the one-dimensional heat equation by properly taking into account spatial and temporal jump corrections. Here we propose a similar technique for the incompressible Navier-Stokes equations. Note that the jump relations (8.48)-(8.49) still hold for moving

discontinuities. We focus again on discontinuities in the pressure caused by forces normal to an interface. In particular, we will discuss the discretization of

$$\int_{t^n}^{t^{n+1}} \int_{\Gamma} p \mathbf{n} \, d\Gamma \, dt, \quad (8.78)$$

for the case that  $p$  has a discontinuity in space which is moving in time and possibly changing its magnitude.

We consider the simplest case in which the interface is aligned with the  $y$ -axis and translating in  $x$ -direction. The spatial integral in equation (8.78) becomes, for volume  $\Omega_{i,j}$ :

$$\int_{t^n}^{t^{n+1}} \int_{\Gamma} p \mathbf{n} \, d\Gamma \, dt = \Delta y \int_{t^n}^{t^{n+1}} (p_{i+1,j}(t) - p_{i,j}(t)) \, dt. \quad (8.79)$$

When the interface passes through one of the faces of the finite volume, the pressure on this face will jump *in time*. We assume this happens at  $t = t^1$ . The discretization of the right face reads, like equation (8.36) for the one-dimensional heat equation,

$$\int_{t^n}^{t^{n+1}} p_{i+1,j}(t) \, dt = \frac{1}{2} \Delta t (p_{i+1,j}^{n+1} + p_{i+1,j}^n) + \frac{1}{2} \llbracket p(t^1) \rrbracket ((t^{n+1} - t^1) - (t^1 - t^n)), \quad (8.80)$$

with the temporal jump given by  $\llbracket p \rrbracket = -\text{sgn}(w_n)[p]$ .

## 8.4.4 Results

### 8.4.4.1 A translating interface

In this section we consider a translating interface. We propose to evaluate the accuracy of our method by comparing the solution of the translating interface to the solution of a stationary interface in a moving reference frame, whose accuracy was tested in section 8.3.6.2. Figure 8.21 provides a sketch of the two situations. The domain is  $[-2, 2] \times [-2, 2]$ , and all boundary conditions are periodic.

In the *moving reference frame*, in which the interface is stationary, we compute a reference solution. With the movement of the interface prescribed as  $x_{AD}(t) = \sin(t)$ , the position, velocity and acceleration of the moving reference frame are given by

$$x'(t) = x(t) - \sin(t), \quad u'(t) = u(t) - \cos(t), \quad a'(t) = a(t) + \sin(t). \quad (8.81)$$

The acceleration of the moving reference frame,  $\sin(t)$ , enters as a fictitious force in the finite volume formulation. We employ a fourth-order explicit Runge-Kutta method, see chapter 6, with a sufficiently small time step, so that temporal errors are negligible.

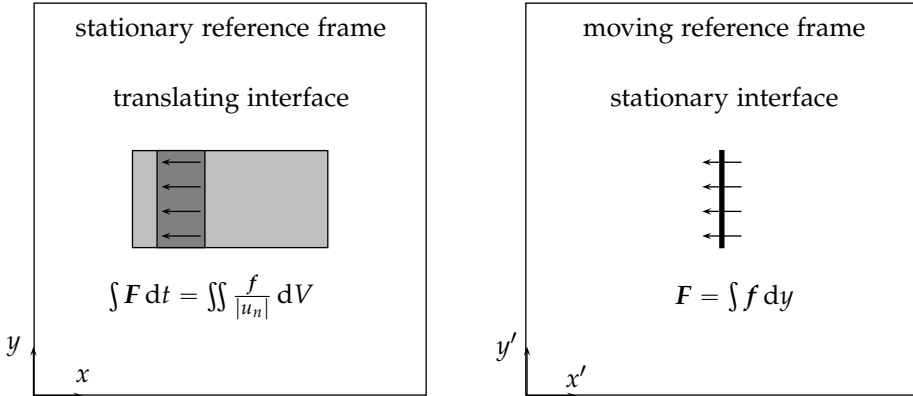


Figure 8.21: Interface (actuator disk) in stationary and moving reference frame.

In the *stationary reference frame* we compute a solution by tracking the movement of the interface as described in sections 8.4.2-8.4.3. The total impulse exerted on the flow during a time interval  $\Delta t$  is given by

$$\int_{t^n}^{t^{n+1}} \int f \, dy \, dt = \int_V \frac{f}{|u_n|} dV = -f_x \Delta y \Delta t e_x. \tag{8.82}$$

Time integration is performed with the Crank-Nicolson method. The pressure that results from the Crank-Nicolson method is an *average* pressure over the time interval, as discussed in chapter 5.3. To obtain a temporally accurate pressure, an additional Poisson equation at the end of a time step should be solved (this can be done as post-processing step). This Poisson equation contains the force *at* the new time level (obtained using the method described in section 8.3, so not the time integrated force given by equation (8.67)). For comparison we also obtain a solution with the discrete Dirac approach and Crank-Nicolson as time integration (see equation (8.77)), using the same time step.

Plots of velocity and pressure at  $t = 2\pi$  are shown in figures 8.22-8.23 ( $Re = 500$ ). At this time instant vortices are forming at the upper and lower edge of the disk, as can be seen from the low pressure areas. Like in the steady case, the immersed interface method results in a much sharper representation of the pressure jump across the interface, and consequently in a more accurate representation of the vortices shed from the disk.

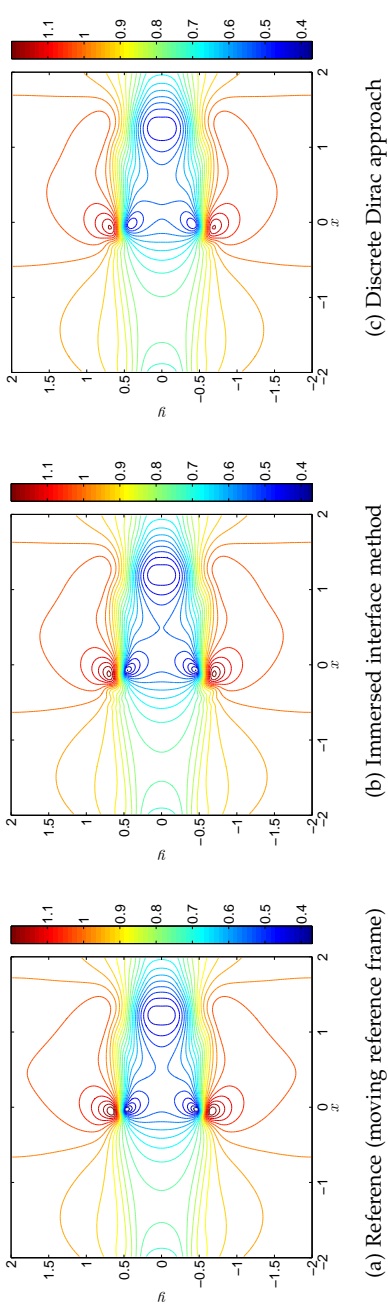


Figure 8.22: Velocity contours at  $t = 2\pi$ , translating interface problem.

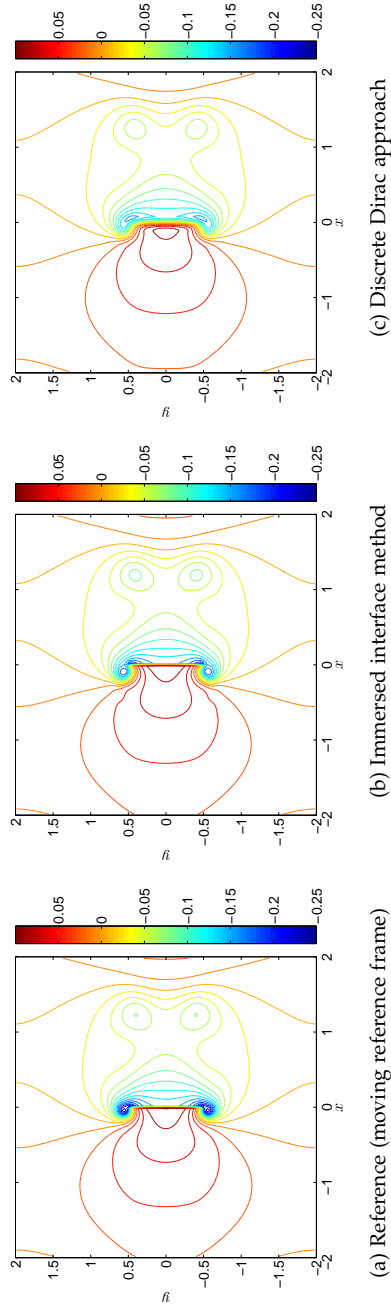


Figure 8.23: Pressure contours at  $t = 2\pi$ , translating interface problem.

### 8.4.4.2 A rotating interface

In this section we consider an immersed interface which is rotating instead of translating. This is, for example, a simplified model of a wind-turbine constantly yawing in the wind, but can also be seen as a model of an insect moving its wings back and forth [48]. This test contains some important ingredients necessary for rotating wind-turbine blades in three dimensions.

We assume that the force on the interface is always normal to its surface, i.e.,  $\mathbf{f} = f_\theta \mathbf{e}_\theta$ . The motion of a point on the disk is given by  $x(t) = r \cos \theta(t)$ ,  $y(t) = r \sin \theta(t)$ , where  $\theta(t)$  is prescribed by a triangle wave with period 2 and amplitude  $\pi/4$ ; the normal velocity of a point on the disk is then given by  $|w_n(r)| = \Omega r$ , with  $\Omega$  the rotational frequency. The disk can be described by a single panel, so equation (8.73) reads

$$\mathbf{I}_{i,j} = \iint_{V \cap \Omega_{i,j}} \frac{\mathbf{f}(\mathbf{x})}{|w_n(\mathbf{x})|} dV = \frac{f_\theta}{|\Omega|} \iint \frac{1}{x^2 + y^2} (-y \mathbf{e}_x + x \mathbf{e}_y) dx dy, \quad (8.83)$$

with  $V$  the volume traced out by the actuator during a time step. The intersection of  $V$  with  $\Omega_{i,j}$  is approximated by an (arbitrarily accurate) polygon. The integrand is expressed as the divergence of a vector field,

$$\iint \frac{1}{x^2 + y^2} (-y \mathbf{e}_x + x \mathbf{e}_y) dx dy = \mathbf{e}_x \iint \nabla \cdot \begin{pmatrix} 0 \\ -\frac{1}{2} \ln(x^2 + y^2) \end{pmatrix} dV + \mathbf{e}_y \iint \nabla \cdot \begin{pmatrix} \frac{1}{2} \ln(x^2 + y^2) \\ 0 \end{pmatrix} dV, \quad (8.84)$$

which can subsequently be rewritten as a contour integral by employing the divergence theorem, yielding:

$$\mathbf{I}_{i,j} = \frac{f_\theta}{|\Omega|} \left[ \mathbf{e}_x \oint_{\partial(V \cap \Omega_{i,j})} -\frac{1}{2} \ln(x^2 + y^2) dx + \mathbf{e}_y \oint_{\partial(V \cap \Omega_{i,j})} \frac{1}{2} \ln(x^2 + y^2) dy \right]. \quad (8.85)$$

These terms can be evaluated by straightforward integration along the sides of a polygon, which are of the form  $\mathbf{x} \cdot \mathbf{n} = \text{constant}$ . The advantage of this approach is that no approximation is introduced in evaluating the volume integrals. For more general force distributions it might not be possible to compute the resulting integrals exactly. In that case it is recommended to use a numerical integration technique, such as Gaussian quadrature.

We use a domain of  $8 \times 4$  with smallest grid size  $1/80$  in the region of the actuator, which is located at  $x = 4$ ,  $y = 2$ . Away from the actuator the grid is mildly stretched with a factor of 1.02. The total number of volumes is  $336 \times 230$ . The Reynolds number is  $\text{Re} = 100$ , the thrust coefficient  $C_T = 0.5$  and the time step is  $\Delta t = 1/200$ .

A typical force distribution is shown in figure 8.24. The largest contribution is at the center of the disk due to the  $1/r$  dependency in equation (8.83). To check the nu-

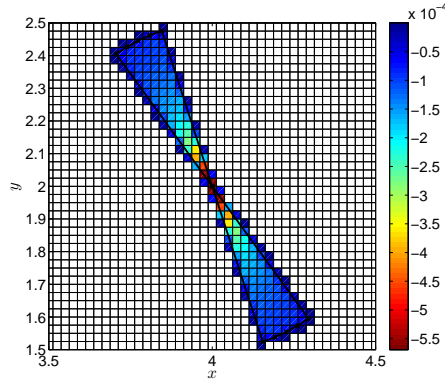


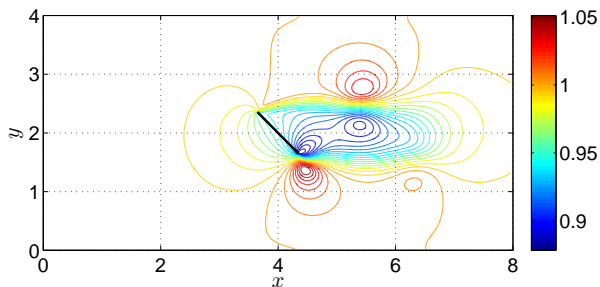
Figure 8.24: An example of the  $u$ -volumes traced out and the magnitude of  $F_x$ . The actual mesh and time step are finer.

merical implementation of equation (8.85), the sum of the force on all finite volumes is compared with the exact value obtained by integrating in polar coordinates:

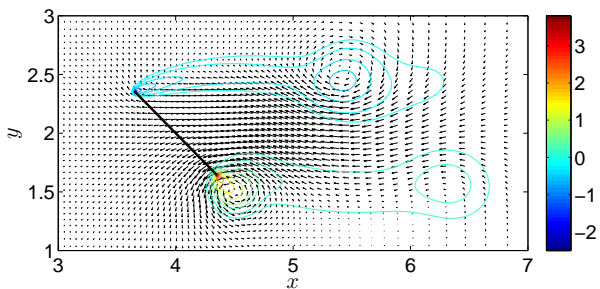
$$F_x = \iint \frac{-\sin \theta}{r} r \, dr \, d\theta = D(\cos \theta_2 - \cos \theta_1), \quad (8.86)$$

where  $\theta_1$  and  $\theta_2$  are the initial and final angle that the actuator makes with the  $x$ -axis, and  $D$  the diameter of the disk.

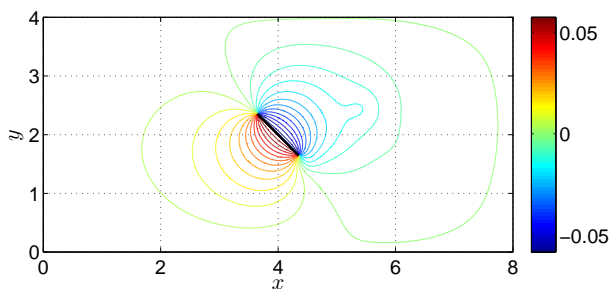
At  $t = 2.5$  the disk is at  $\theta = 3\pi/4$  and the direction of motion flips. Contour plots of velocity, pressure and vorticity are shown in figure 8.25. From the velocity contours it is clearly seen that a wake is forming behind the disk, which has travelled over a distance of approximately  $2.5D$  since the start of the simulation at  $t = 0$ . Upon reversal of the direction of motion a vortex is shed from the lower side of the disk as can be seen in figure 8.25c. The vortex previously shed from the upper side is positioned approximately at  $(5.5, 2.5)$  but has strongly reduced in strength due to the relatively high viscosity of the flow. The pressure is, as in the previous test cases, discontinuous across the disk.



(a) Velocity magnitude



(b) Vorticity and velocity vectors. The inflow is subtracted to clarify the motion.



(c) Pressure

Figure 8.25: Flow field just after reaching maximum deflection ( $t = 2.505$ ).



## 8.5 NEW ACTUATOR METHODS FOR WIND-TURBINE WAKE SIMULATIONS

Although the results presented so far are for relatively simple geometries and simple force expressions, the proposed methods are valid for arbitrary shaped interfaces on which surface forces act. For wind-turbine wake computations a hierarchy of actuator models can be constructed based on these ideas.

The most advanced and complete actuator model of a rotating wind-turbine blade is obtained by prescribing shear stresses and pressure stresses along the entire blade. We call this approach the ‘complete’ *actuator surface method*. This requires detailed knowledge of the geometry and the force distribution, which is generally not available. Several simplifications are possible based on reducing the complexity of the geometry and of the forcing terms. A first (geometric) simplification is to collapse the blade onto the chord or camber line, and prescribe the pressure difference and sum of shear forces along this surface; this is the ‘thin’ actuator surface method. A second (force) simplification is to prescribe average lift and drag forces along this collapsed surface. We call this the *actuator line method*, since it uses the same forcing (lift and drag) as the original actuator line method proposed in Sørensen and Shen [176]. However, in their method the force is applied along a line (a point in 2D), which leads to a singularity in the flow that has to be ‘repaired’ by smoothing with discrete Dirac functions. We propose to keep the force associated with a surface (the chord in 2D); this removes the singularity and gives consistent results upon mesh refinement.

The different approaches are illustrated in figure 8.26 for a two-dimensional situation. The ‘reference’ flow field is obtained with an immersed boundary method [113] and shown in figure 8.26a. In figure 8.26b the body is replaced by the distribution of pressure and shear forces along the surface, which have been obtained from Xfoil [44]. In figure 8.26c these forces are projected onto the airfoil chord, and in figure 8.26d the lift and drag coefficients are prescribed along the chord. We observe that with every simplification some detail of the flow field is lost. However, the velocity profiles of the different methods in the wake are very similar. For the far wake it is most important that the total imposed force is correct, and the details of geometry and force distribution have a smaller influence. This makes these approaches suitable for wind turbine wake simulations, where the detailed flow near the blade is less important.

Note that a further simplification of the actuator line can be obtained by time averaging. The actuator line sweeps out a circle in time: this is the *actuator disk method* (possibly with both normal and tangential loading).

An important feature of our immersed interface method in finite volume setting is that the above suggested reductions in complexity of geometry or forcing terms follow automatically in a discrete sense when taking large time steps or coarse meshes. For example, the actuator surface model reduces to the actuator line model on a coarse mesh, and the actuator line model reduces to the actuator disk model when time steps corresponding to a full rotor revolution are taken.

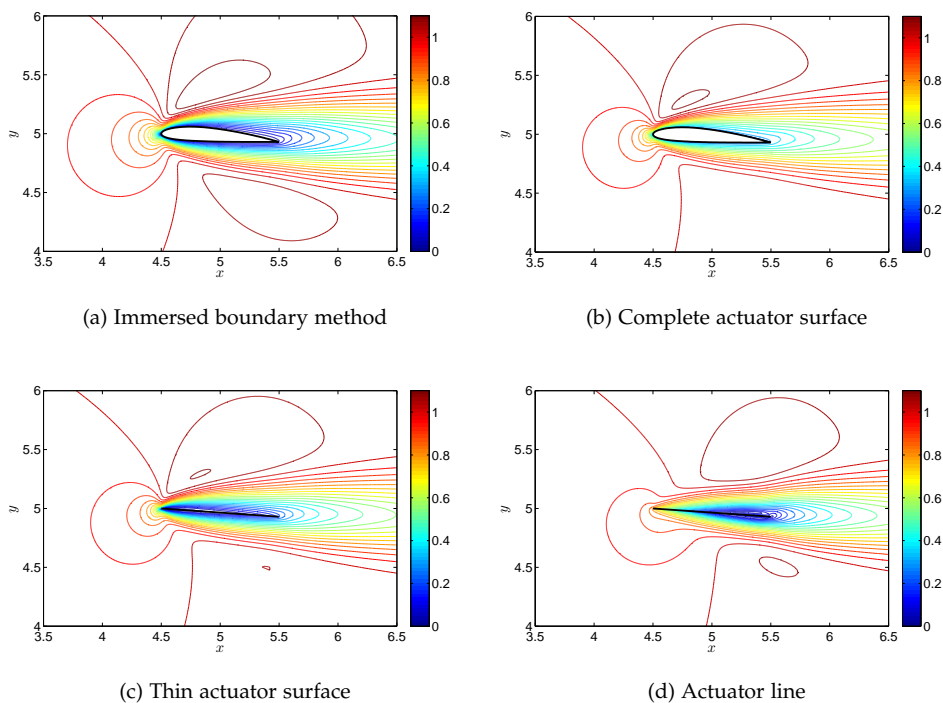


Figure 8.26: Hierarchy of actuator models for the flow over an airfoil. Contour lines of velocity magnitude.

## 8.6 CONCLUSIONS

In this chapter we have proposed a new method for introducing actuator forces in finite volume methods for the incompressible Navier-Stokes equations. Such actuator forces lead to discontinuities in the flow field that should be treated carefully. The background for our approach is the immersed interface method (IIM), a finite difference method first proposed by LeVeque and Li [96]. We have extended the IIM to a finite volume setting. The finite volume approach has the advantage that less differentiability of the problem is required. For a one-dimensional test problem we show how correction terms are derived. These correction terms involve both spatial and temporal jumps, and are taken such that the local truncation error is  $\mathcal{O}(\Delta x)$  near the interface. This is sufficient for global second order accuracy in the maximum norm.

The method has been applied to two-dimensional flows governed by the incompressible Navier-Stokes equations. For stationary interfaces our method is the finite volume equivalent of the continuous equations in integral form: a surface force acts on the finite volumes which it intersects, and the contribution to the finite volume depends on the area of intersection. The method sharply captures discontinuities in pressure that arise from the imposition of forcing terms normal to the interface. Compared to the conventional ‘Dirac’ approach, both velocity and pressure are more accurate (less smeared out), and there is no need to choose a discrete Dirac function and regularization parameter.

For moving interfaces, we have integrated the continuous equations in both space and time. By rewriting the time-integrated force in terms of a volume integral, a new discretization method is obtained: a finite volume gets a force contribution depending on the intersectional area with the moving actuator. This opens up the opportunity to take large time steps. However, as illustrated by one-dimensional test cases, care should be taken to avoid unphysical oscillations arising from the time integration method.

In order to apply the methods described in this chapter to the computation of wind turbine wakes, a number of steps should be taken. Firstly, the method should be implemented for more complicated geometries (turbine blades), and in three dimensions. This requires an efficient method to trace the intersected finite volumes during a time step; a possibility is to use a level-set representation of the interface [98]. Secondly, the forcing terms should not be prescribed, but follow from the flow field. For a rotating turbine blade, the forcing can be computed for example with the blade element momentum (BEM) method based on the local angle of attack. Thirdly, the current work has focused on the (most important) case of forcing normal to the interface. However, tangential forces will introduce jumps in the derivative of the velocity that have to be taken into account properly.



In this chapter we show a number of simulation results that have been obtained with the ECNS code as developed during this PhD project. Ingredients from all parts of this thesis are combined: the staggered grid spatial discretization described in part I, time integration with Runge-Kutta methods as described in part II, and the representation of actuators as immersed interfaces as described in this part, part III. The purpose of this chapter is not, in contrast to previous chapters, to develop and verify numerical methods, but rather to show that our approach has resulted in a code that is ready to be used for wind-turbine wake aerodynamics. As we mentioned in section 1.5, there is a fourth ingredient - LES turbulence modeling - necessary to complete the current work. Development, implementation and testing of an LES model for wind-turbine wakes constitutes a PhD research in itself, and is left as suggestion for future research. The simulations in this chapter therefore consider laminar flow, which already reveal some important physical phenomena occurring in wakes.

### 9.1 PARALLELIZATION

In order to use the ECNS code for large-scale problems it had to be parallelized<sup>1</sup>. The approach that has been taken is to write all discrete operators (convection, diffusion, pressure gradient) as matrices. These matrices are saved in a sparse, parallel format. Evaluating convective or diffusive terms then amounts to matrix-vector products. A clear advantage of this approach is that implicit time integration methods, such as described in section 7, can be implemented with relative ease. A drawback is that storing the discretization matrices can be memory intensive.

The framework that has been used for parallelization is PETSc (Portable Extensible Toolkit for Scientific Computation), ‘a suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations’ [11]. After building the discretization matrices using PETSc routines, the user can perform matrix computations (such as matrix-vector and matrix-matrix multiplications, dot products, etc.) without being concerned about the specifics of the parallel implementation. A large set of matrix solvers is available in PETSc to solve the resulting system of equations.

The results shown in this chapter have been obtained with the explicit Runge-Kutta methods from chapter 6, so that a solution method for the Poisson equation is necessary. Since the Laplace operator in the Poisson equation is symmetric positive definite (on cartesian grids), a good candidate to solve the Poisson equation is the

---

<sup>1</sup> This has been performed in cooperation with Margreet Nool (CWI, Amsterdam) and Steven van Haren (NRG, Petten) under an NCF Research Grant, grant number NRG-2011.06.

preconditioned conjugate gradient method. As preconditioner algebraic multigrid was chosen (ML, see [57]). The same preconditioner can be used throughout the time stepping process (the matrix for the Laplace operator is constant in time).

## 9.2 FLOW OVER A WING

As a first application of actuator methods in three dimensions we consider the flow over a wing. The wing is modeled as an actuator surface that carries a uniformly distributed load normal to its surface; the effective lift coefficient is  $C_L = 2$ . The computational domain has size  $6 \times 3 \times 3$  and is discretized into  $120 \times 60 \times 60$  finite volumes. The wing is positioned at  $x = 1$ , has a span equal to 1 and a chord equal to 0.2. A steady solution is obtained by marching in time. Figure 9.1 shows the solution in terms of streamlines and contours of the vertical velocity component. The upwash in front of the wing and downwash behind it are clearly visible. The prescribed loading induces a pressure discontinuity in  $y$ -direction, as can be seen from figure 9.2a. At the wing tips this pressure difference leads to the formation of a tip vortex, and the roll-up of the wake. This is visualized in figure 9.2b.

Upon rotating this actuator surface in time (around the  $x$ -axis) using the method described in section 8.4, and taking a non-uniformly distributed load over the surface, a model for a rotating wind-turbine blade results.

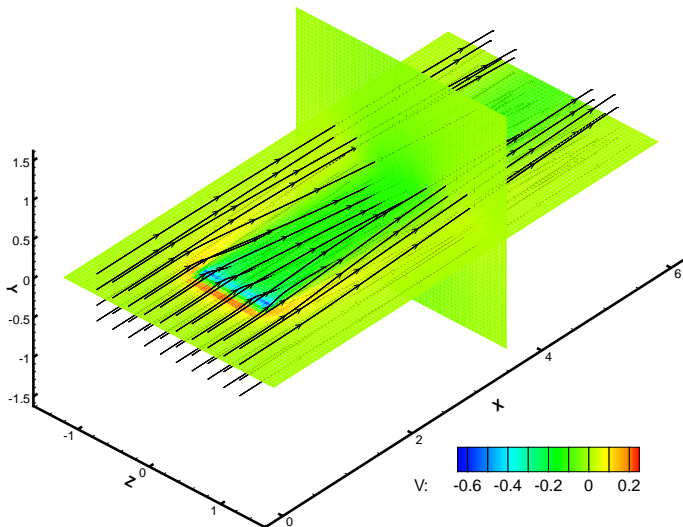


Figure 9.1: Flow over a wing, contours of  $v$ -velocity.

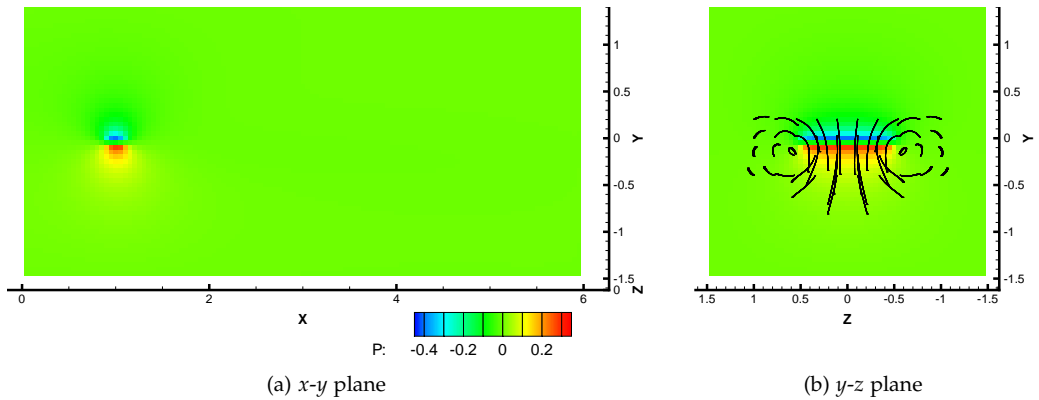


Figure 9.2: Flow over a wing, contours of pressure.

### 9.3 FLOW THROUGH AN ACTUATOR DISK

We continue with the flow through a uniformly loaded actuator disk. The forcing is now parallel (but in opposite direction) to the incoming flow, instead of perpendicular to it (as in the previous section). The representation of the circular disk on a cartesian mesh is done following section 8.3, leading to the force distribution shown in figure 9.3. The disk is placed in the center of a domain of  $6 \times 3 \times 3$ . We employ 20 finite volumes per rotor diameter (this is quite common, even in LES computations). The resulting velocity field, shown in figure 9.4a, clearly reveals the expansion of the wake. The high loading of the disk ( $C_T = 1$ ) causes the velocity in the far wake to approach zero. Due to the relatively high Reynolds number ( $Re = 1000$ ) the wake interface stays quite sharp - in reality turbulent mixing will diffuse the interface and will lead to recovery of the velocity field. The wake interface can also be visualized by considering isosurfaces of the vorticity magnitude. Figure 9.4b shows how vorticity is created at the edge of the disk and subsequently transported downstream. The coloring by means of pressure values reveals the jump in pressure across the disk.

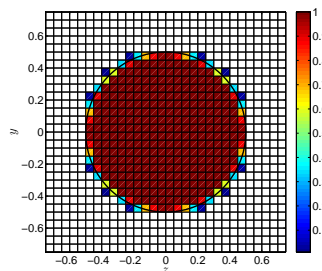


Figure 9.3: Uniformly loaded actuator disk. Force distribution based on intersectional area.

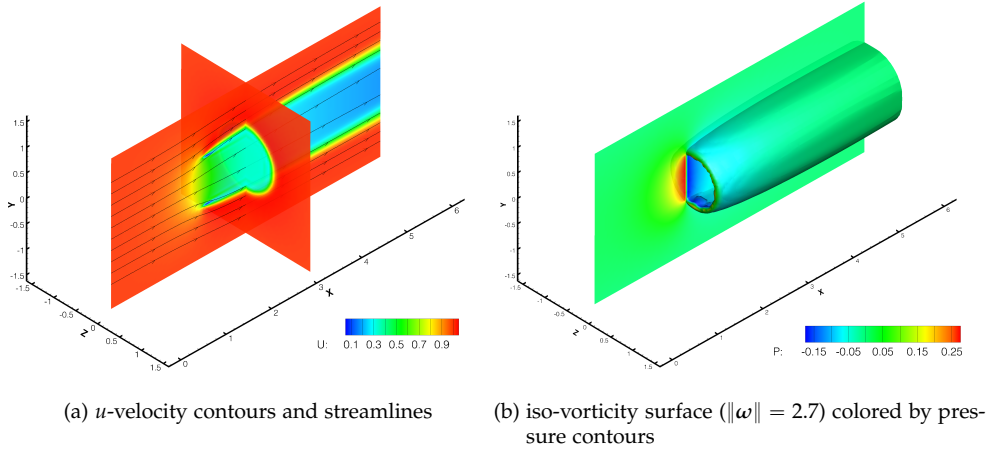


Figure 9.4: Flow through an actuator disk.

#### 9.4 FLOW IN A WIND FARM

We consider the flow in a wind farm array of 9 turbines, with spacing of 4 D and 3 D in streamwise and spanwise direction, respectively (the Lillgrund wind farm in Sweden is an example of a wind farm with such a small spacing). The turbine rows are at an angle of  $30^\circ$  with respect to the incoming flow. Instead of prescribing an inflow velocity at this angle, we have changed the position of the turbines. This has the advantage that the actuator disks are aligned with the mesh. The domain has size  $13 \times 3 \times 10$ . The same resolution as for the single wake computation is used, leading to a total of  $3.12 \cdot 10^6$  finite volumes.

First we simulate, like in the previous example, wakes without presence of the ground. All boundaries are of outflow (pressure) type, except the inflow boundary at  $x = 0$ . With these outflow conditions it is possible to place the computational boundaries close to the turbines without any significant effect on the solution. The thrust coefficient of the turbines is taken equal to  $C_T = \frac{1}{2}$  (in reality the thrust coefficient of the turbines depends on the local velocity field, see 1.4.1.1). The Reynolds number is  $Re = 200$ , which (compared to the previous section) leads to a more realistic recovery of the velocity field in the wake. Even though the flow is laminar, there are important physical effects present in these computations. Figure 9.5 shows vorticity isosurfaces colored by the  $u$ -component of the velocity field. The high vorticity areas close to the disks get diffused further downstream. In terms of vorticity, the wakes of the downstream turbines are longer than the upstream ones; the highest vorticity regions are encountered at the downstream turbines. Significant wake effects can be observed for these downstream turbines, where the velocity drops to less than 50% of the freestream value. In contrast to what would happen in



a two-dimensional simulation, there is hardly any ‘speed-up’ of the flow in between the wakes, because the flow is able to expand in the third dimension.

Next we add the presence of the ground to the simulation. We prescribe a boundary-layer type inflow profile of the form [22]

$$u(y) = y^{0.14}, \quad (9.1)$$

and no-slip boundary conditions at  $y = 0$ . The thickness of the resulting boundary layer is still growing as function of downstream position. The presence of the ground leads to a loss of axi-symmetry in the wake velocity profiles, and to an upward movement of the wakes (see figure 9.6). The wake deficits are clearly visualized in figure 9.7.

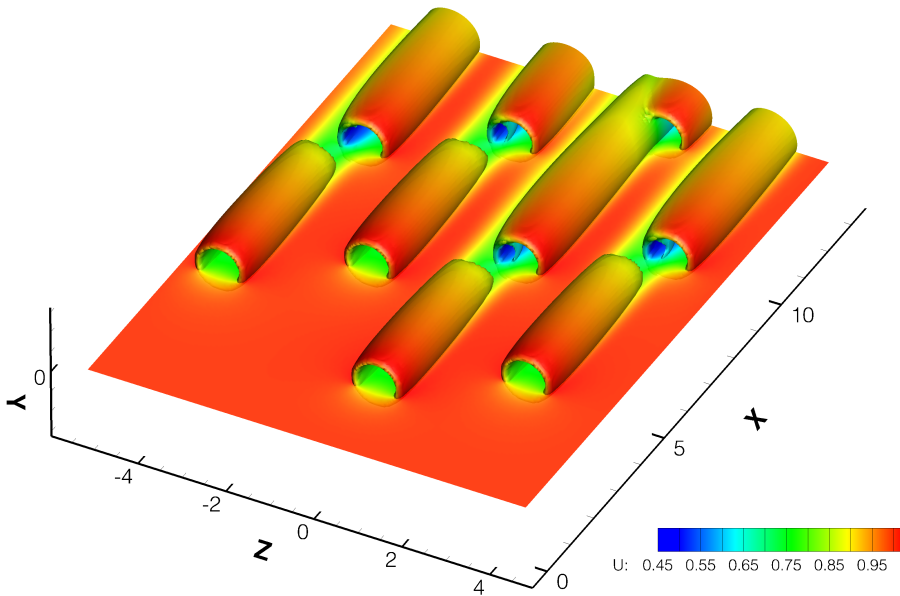


Figure 9.5: Flow through a farm of 9 turbines. Vorticity isosurfaces colored by velocity field.

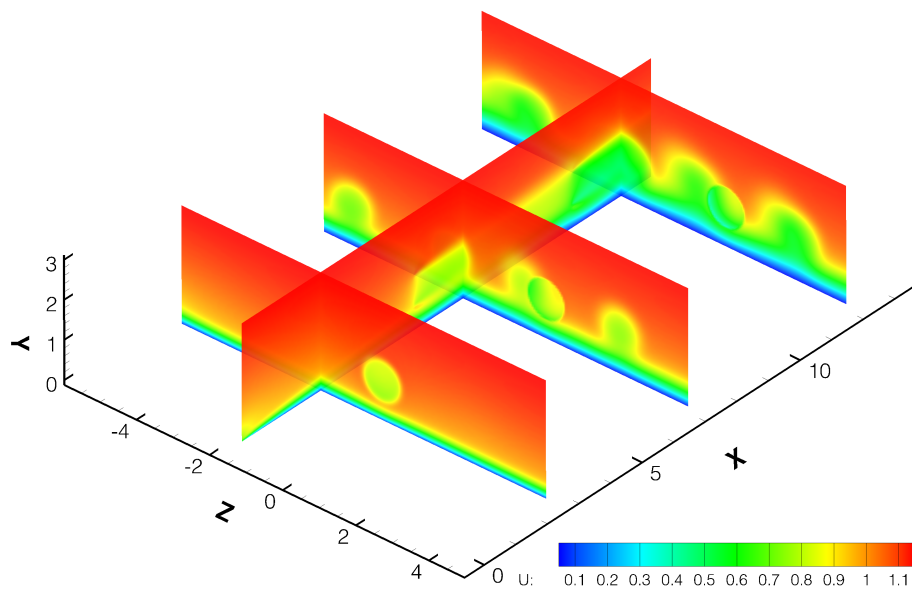


Figure 9.6: As figure 9.5, but with presence of ground. Slices of velocity field.

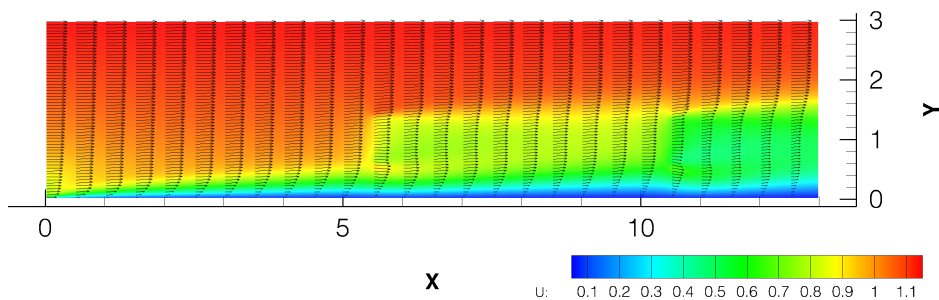


Figure 9.7: Velocity vectors along  $x - y$  slice of figure 9.6.

CONCLUSIONS AND RECOMMENDATIONS

---

*As in all research we have had some real success, some partial success, some partial failure and some real failure, and there is obviously still plenty to do.*<sup>1</sup>

## 10.1 CONCLUSIONS

As stated in chapter 1, the goal of this thesis was to develop accurate and efficient numerical simulation methods for turbulent flows, and in particular for wind-turbine wake aerodynamics. In the three parts of this thesis we have investigated and proposed new numerical methods for the spatial and temporal discretization of the incompressible Navier-Stokes equations, as well as for the representation of the wind turbine model with actuator methods. The main conclusion is that: the use of (i) finite volume methods on staggered cartesian grids, (ii) time discretization with Runge-Kutta methods, and (iii) immersed interface methods to represent wind turbines, constitute a very appropriate approach to perform wind-turbine wake simulations with LES as turbulence model. The reason for this is the combination of *stability, low artificial viscosity and high-order accuracy in space and time*. Since these properties are generally valuable for simulating turbulent flows governed by the incompressible Navier-Stokes equations, we expect that the proposed discretization methods are also widely applicable outside the wind energy community. Below we will detail the conclusions for each of the parts of this thesis.

10.1.1 *Spatial discretization*

We have proposed the use of second and fourth order energy-conserving finite volume methods on staggered cartesian grids to simulate wind-turbine wake aerodynamics. These methods are stable for any mesh, independent of viscosity, and do not introduce artificial diffusion. For both the second and fourth order method an analysis of the boundary contribution to the energy equation was performed, leading to new ‘energy-conserving’ boundary conditions. A theoretical one-dimensional analysis showed that the global error of high-order methods is limited to second order when requiring energy conservation. Therefore, near solid boundaries, non-uniform grids should be applied in order for the fourth order scheme to be efficient.

---

<sup>1</sup> L. Fox, Some Experiments with Singularities in Linear Elliptic Partial Differential Equations.

### 10.1.2 *Temporal discretization*

For the temporal discretization we have considered Runge-Kutta methods. Runge-Kutta methods have been widely applied to the spatially discretized incompressible Navier-Stokes equations, but order of accuracy proofs that address both velocity and pressure are missing. By viewing the spatially discretized Navier-Stokes equations as a system of differential-algebraic equations the order conditions for velocity and pressure were derived. It turned out that for implicit methods (at least the ones based on Gauss, Radau or Lobatto quadrature), the velocity attains its classical order of accuracy, while for explicit methods this is true as long as the mesh is not moving. The pressure is in general only first order accurate, unless an additional Poisson equation is solved. We have proposed new methods that lead to a second order pressure for a range of Runge-Kutta methods.

An important property of Runge-Kutta methods is that the coefficients can be chosen such that the energy-conserving property of the spatial discretization is kept when marching in time. The most prominent class of methods is based on Gauss quadrature, which have the highest possible order of all Runge-Kutta methods. The resulting second and fourth order energy-conserving discretizations (in space and time) are stable for any mesh and any time step. In a practical computation this resulted in the accurate simulation of the roll-up of a shear layer, and especially in the accurate prediction of the enstrophy of the flow.

The energy-conserving property of Runge-Kutta methods comes at a price: lack of  $L$ -stability. For stiff problems order reduction will occur. Energy-conserving Runge-Kutta methods should therefore preferably be applied to the convective terms only, and not to the diffusive terms (which can be stiff). As a remedy, an extension of energy-conserving Runge-Kutta methods to additive Runge-Kutta methods was proposed. In these additive methods the diffusive terms are integrated with an  $L$ -stable method, and the convective terms with an energy-conserving method, both based on the same quadrature points. Unfortunately, the resulting method has lost the original unconditional (non-linear) stability properties, and due to low stage order, it is still not competitive for stiff problems.

### 10.1.3 *Actuator methods*

The action of wind turbines on the flow is represented by using actuator methods. These actuators can lead to discontinuities in the flow which should be handled with care. In contrast to the commonly used discrete Dirac functions, we propose to use immersed interface methods, which take into account the presence of discontinuities in the discretization method. We have extended the immersed interface methodology to a finite volume setting, which is consistent with the presence of discontinuities. Simulation of the flow through actuator disks shows that the method sharply captures pressure discontinuities and results in a velocity field that is less diffusive than the discrete Dirac approach. For instationary interfaces a new discretization method was derived based on the volume swept by the moving interface.

## 10.2 RECOMMENDATIONS

As mentioned in the introductory chapter, the state-of-the-art in wind-turbine wake simulations is to use LES as turbulence model. The main extension of the current work, and as such the most important recommendation, is to include an LES model in the ECNS code. The code is then ready to perform simulations of turbulent flow. The non-dissipative discretization methods developed in this thesis are suitable to test and compare different LES models, since all diffusion will be coming from the turbulence model, and not from the numerics (the contribution from the laminar viscosity is often negligible in wind-turbine wakes). Such a comparison is very valuable, also outside the wind energy community.

Other recommendations - or: areas for further research - are listed below, grouped by each part of the thesis.

### 10.2.1 *Spatial discretization*

It was shown in this thesis that the order of accuracy of the energy-conserving fourth order spatial discretization is limited to second order when boundaries are involved. The starting point in this analysis was the definition of the energy, obtained by the choice for an inner product. It is possible to generalize the analysis by choosing a ‘weighted’ or ‘adapted’ inner product, e.g.

$$(\mathbf{u}, \mathbf{v})_A = \int_{\Omega} \mathbf{u} \cdot (\mathbf{A} \cdot \mathbf{v}) \, d\Omega, \quad (10.1)$$

where  $\mathbf{A}$  is a symmetric positive definite rank-2 tensor. Deriving both the continuous and discrete energy equation for this inner product shows that the requirements on the convective and diffusive matrices are relaxed with respect to the standard inner product: instead of (skew-) symmetry, a condition on the eigenvalues of the matrices results (purely imaginary for convection, negative and real for diffusion). This constitutes a broader class of matrices, opening the room for higher-order boundary discretizations. See for example [184] for a weighted inner product approach to hyperbolic partial differential equations.

This thesis has furthermore focused on (uniform and non-uniform) cartesian grids. For many (unsteady) turbulent flows, such as in wind-turbine wakes, it might be efficient to consider local grid refinement, for example to accurately track tip vortices. Combining local grid refinement with energy conservation and possibly high order constitutes a formidable challenge; some work in this direction is described in [69].

The order of accuracy of the pressure in case of boundary conditions was not reported in this thesis. Hou and Wetton [72] proved second order convergence for a case of no-slip boundary conditions in one direction and periodic boundary conditions in the other direction. It appears, however, that second order is not obtained in domains with no-slip boundary conditions on all sides (i.e. domains involving

corners). We found that the reason for this lies in the local truncation error of the diffusive terms, which is inconsistent at the boundary of a staggered grid. A fix can be found by changing the discretization of the diffusive term near the boundary, which restores the second order accuracy of the pressure. Further research is needed to investigate if this can be applied to the fourth order scheme as well.

### 10.2.2 *Temporal discretization*

Like for the spatial discretization, it is possible to define the kinetic energy in time in different ways. In this thesis the energy was defined ‘pointwise’ in time. An alternative definition could be based on the average kinetic energy during a time interval. This could lead to other interesting ‘energy’-conserving time integration schemes.

The implicit Gauss Runge-Kutta methods proposed for energy-conserving time integration of the incompressible Navier-Stokes equations are relatively expensive (the cost per time step is high). Research into efficient solution of non-linear saddle-point systems is indispensable. Then, the performance of these methods should be evaluated for a variety of test cases, including turbulent flows. We expect that the Gauss methods are most effective for problems that are sensitive to the amount of numerical diffusion, such as problems involving transition.

The order conditions and stability properties of Runge-Kutta methods treated in this thesis were investigated by using existing theory on index-2 differential-algebraic equations. These theoretical results provide a very general and thorough framework to study time discretizations for the spatially discretized incompressible Navier-Stokes equations. It is recommended to continue further research on time integration methods along this line. This offers insight in order conditions and stability for situations such as moving meshes, implicit-explicit (IMEX) Runge-Kutta methods, (symmetric) projection methods, multi-step methods, etc. Some pointers to literature are [63, 61, 79].

The additive Runge-Kutta methods proposed in this work - aimed at being energy-conserving in the inviscid limit, and  $L$ -stable in the stiff limit - integrate the convective terms with a different Runge-Kutta method than the diffusive terms. Another idea, not tested in this thesis, is to integrate convection-dominated regions with an energy-conserving Runge-Kutta method, and diffusion-dominated regions with an  $L$ -stable method.

Another issue with Runge-Kutta methods are ‘space-time errors’, which can result from unsteady boundary conditions (as mentioned in section 6.2.6). This is an important case for wind-turbine wake simulations, where the inflow is generally unsteady. An order of accuracy study with simultaneous mesh and time step refinement is necessary to investigate if this is an issue.

### 10.2.3 *Actuator methods*

The immersed interface approach developed in this paper has been successfully applied to actuators with prescribed loading and prescribed motion. The most important extension will be to make the loading depending on the local velocity field, as is required when modeling wind turbine blades. The forcing then becomes an unknown (as function of the velocity) which has to be taken into account in the solution of the (non-linear) system of equations at each time step. Furthermore, an efficient method to determine intersections between a (moving) actuator and the mesh is required, possibly by using level-set methods.





## DETAILS OF SPATIAL DISCRETIZATION

---

### A.1 BOUNDARY STENCIL FOR THE POISSON EQUATION

In the interior the  $u$ -momentum equation reads:

$$\dot{u}_{i,j} = N_{i,j}^u - (\Omega_{i,j}^u)^{-1} (G^u p)_{i,j}, \quad (\text{A.1})$$

where  $N^u$  includes the convective and diffusive terms of the  $u$ -component, divided by the effective finite volume size (see equation (3.13)). The time derivative of the divergence equation reads

$$\alpha_1(\dot{u}_{i,j} - \dot{u}_{i-1,j}) - \alpha_2(\dot{u}_{i+1,j} - \dot{u}_{i-2,j}) + (v\text{-component}) = 0. \quad (\text{A.2})$$

Inserting (A.1) gives the Laplace operator for a uniform grid:

$$\frac{1}{\Delta x} \left( \frac{1}{8} p_{i-3,j} - \frac{27}{4} p_{i-2,j} + \frac{783}{8} p_{i-1,j} - \frac{365}{2} p_{i,j} + \frac{783}{8} p_{i+1,j} - \frac{27}{4} p_{i+2,j} + \frac{1}{8} p_{i+3,j} \right) = \alpha_1(N_{i,j}^u - N_{i-1,j}^u) - \alpha_2(N_{i+1,j}^u - N_{i-2,j}^u) + (v\text{-component}). \quad (\text{A.3})$$

Near a boundary the stencil of the Laplace operator is found in a similar way. Taking the time derivative of the divergence-free constraint near the boundary (the first row of equation (4.6)) gives

$$(\alpha_1 - \alpha_2)\dot{u}_{1,j} - \alpha_2\dot{u}_{2,j} + (-\alpha_1 + 2\alpha_2)\dot{u}_{b,j} = 0, \quad (\text{A.4})$$

where we have ignored the  $v$ -component. Substituting the momentum equations for  $\dot{u}_{1,j}$  and  $\dot{u}_{2,j}$  gives

$$\frac{1}{\Delta x} \left( -\frac{677}{8} p_{1j} + \frac{729}{8} p_{2,j} - \frac{53}{8} p_{3,j} + \frac{1}{8} p_{4,j} \right) = (\alpha_1 - \alpha_2)N_{1,j}^u - \alpha_2 N_{2,j}^u + (-\alpha_1 + 2\alpha_2)\dot{u}_{b,j}. \quad (\text{A.5})$$

The effective ('implied') boundary condition for the pressure follows by subtracting equation (A.3) (for  $i = 1$ ) from the last equation:

$$\frac{1}{\Delta x} \left( \frac{1}{8} (p_{3,j} - p_{-2,j}) - \frac{27}{4} (p_{2,j} - p_{-1,j}) + \frac{783}{8} (p_{1,j} - p_{0,j}) \right) = \alpha_1 N_{b,j}^u - \alpha_2 (N_{-1,j}^u + N_{1,j}^u) + (-\alpha_1 + 2\alpha_2)\dot{u}_{b,j}. \quad (\text{A.6})$$

Repeating this procedure for  $i = 2$  gives the following stencil for the Poisson operator:

$$\frac{1}{\Delta x} \left( \frac{729}{8} p_{1,j} - \frac{1459}{8} p_{2,j} + \frac{783}{8} p_{3,j} - \frac{27}{4} p_{4,j} + \frac{1}{8} p_{5,j} \right) = \alpha_1 (N_{2,j}^u - N_{1,j}^u) - \alpha_2 N_{3,j}^u + \alpha_2 \dot{u}_{b,j}, \quad (\text{A.7})$$

so the effective boundary condition is

$$\frac{1}{\Delta x} \left( \frac{1}{8} (p_{2,j} - p_{-1,j}) - \frac{27}{4} (p_{1,j} - p_{0,j}) \right) = -\alpha_2 N_{b,j}^u + \alpha_2 \dot{u}_{b,j}. \quad (\text{A.8})$$

Similarly, for  $i = 3$ , the stencil of the Poisson operator is

$$\frac{1}{\Delta x} \left( -\frac{53}{8} p_{1,j} + \frac{783}{8} p_{2,j} - \frac{365}{2} p_{3,j} + \frac{783}{8} p_{4,j} - \frac{27}{4} p_{5,j} + \frac{1}{8} p_{6,j} \right) = \alpha_1 (N_{3,j}^u - N_{2,j}^u) - \alpha_2 (N_{4,j}^u - N_{1,j}^u), \quad (\text{A.9})$$

and the effective boundary condition is

$$\frac{1}{\Delta x} \frac{1}{8} (p_{1,j} - p_{0,j}) = 0. \quad (\text{A.10})$$

A.2 BOUNDARY CONTRIBUTIONS TO THE KINETIC ENERGY, FOURTH ORDER METHOD

A.2.1 *Convective terms*

The convective matrix  $C_3^u(\bar{u}, \bar{v})$  can be constructed once  $u_{-1/2,j}$  and  $u_{1/2,j}$  are expressed in terms of  $u_b$  and interior points:

$$u_{-1/2,j} = \frac{1}{2}(u_{-2,j} + u_{1,j}) = u_b + \frac{1}{2}u_{1,j} - \frac{1}{2}u_{2,j} \tag{A.11}$$

$$u_{1/2,j} = \frac{1}{2}(u_{-1,j} + u_{2,j}) = u_b - \frac{1}{2}u_{1,j} + \frac{1}{2}u_{2,j} \tag{A.12}$$

In section 4.1.4 we comment on the use of higher-order interpolation formulas to obtain  $u_{-1/2,j}$  and  $u_{1/2,j}$ . With these interpolations  $C_3^u(\bar{u}, \bar{v})$  reads

$$C_3^u(\bar{u}, \bar{v}) = \begin{bmatrix} \bar{u}_{-1/2,j} + \bar{u}_{1/2,j} + \frac{1}{2}\bar{u}_{3/2,j} - 3\bar{u}_{b,j} & \frac{1}{2}\bar{u}_{-1/2,j} - \frac{1}{2}\bar{u}_{1/2,j} & -\frac{1}{2}\bar{u}_{-1/2,j} + \frac{1}{2}\bar{u}_{1/2,j} & \frac{1}{2}\bar{u}_{3/2,j} \\ \frac{1}{2}(\bar{v}_{3/4,j+1} - \bar{v}_{3/4,j-2}) & \frac{1}{2}\bar{u}_{-1/2,j} & -\frac{1}{2}\bar{u}_{-1/2,j} & \frac{1}{2}\bar{u}_{3/2,j} \\ -\bar{u}_{-1/2,j} & -\frac{1}{2}\bar{u}_{-1/2,j} & \frac{1}{2}\bar{u}_{-1/2,j} & 0 \\ -\bar{u}_{1/2,j} & \frac{1}{2}\bar{u}_{1/2,j} & -\frac{1}{2}\bar{u}_{1/2,j} + \frac{1}{2}\bar{u}_{7/2,j} & 0 \\ -\frac{1}{2}\bar{u}_{3/2,j} & -\frac{1}{2}\bar{u}_{3/2,j} & \frac{1}{2}(\bar{v}_{5/2,j+1} - \bar{v}_{5/2,j-2}) & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{A.13}$$

The rows represent the discretizations for the finite volumes around  $u_b$ ,  $u_1$ ,  $u_2$ , etc., whereas the columns represent the contribution from  $u_b$ ,  $u_1$ ,  $u_2$ , etc. The lower right part of the matrix is the interior discretization which is in practice used for solving the system of equations. Conservation of momentum as derived in section 4.1.2 can be checked from the column sums

of the matrix. We apply interpolation (3.17) and boundary conditions (4.5) and (4.10) and rewrite  $C_3^u$ . This is straightforward, except for the upper left entry of the matrix:

$$\bar{u}_{-1/2,j} + \bar{u}_{1/2,j} + \frac{1}{2}\bar{u}_{3/2,j} - 3\bar{u}_{b,j} + \frac{1}{2}(\bar{v}_{3/4,j+1} - \bar{v}_{3/4,j-2}) = \quad (\text{A.14})$$

$$= -\frac{1}{2}\bar{u}_b + \frac{1}{2}(\bar{u}_{3/2,j} - \bar{u}_{b,j} + \bar{v}_{3/4,j+1} - \bar{v}_{3/4,j-2}) \quad (\text{A.15})$$

$$= -\frac{1}{2}\bar{u}_b + \frac{1}{4}(\beta_1\bar{u}_b + \beta_2\bar{u}_1 + \beta_2\bar{u}_2 + \beta_1\bar{u}_3) - \frac{1}{2}(\beta_1 + \beta_2)\bar{u}_b + \frac{1}{2}(\bar{v}_{3/4,j+1} - \bar{v}_{3/4,j-2}) \quad (\text{A.16})$$

$$= -\frac{1}{2}\bar{u}_b + \frac{1}{4}(\beta_1(\bar{u}_{3,j} - \bar{u}_{b,j} + \bar{v}_{2,j+1} - \bar{v}_{2,j-2}) + \beta_2(\bar{u}_{2,j} + \bar{u}_{1,j} - 2\bar{u}_{b,j} + \bar{v}_{1,j+1} - \bar{v}_{1,j-2})) \quad (\text{A.17})$$

$$= -\frac{1}{2}\bar{u}_b. \quad (\text{A.18})$$

To arrive from equation (A.16) at equation (A.17) it is required to interpolate the vertical fluxes near the boundary as

$$\bar{v}_{3/4,j+1} = \frac{1}{2}\beta_1\bar{v}_{2,j+1} + \frac{1}{2}\beta_2\bar{v}_{1,j+1}. \quad (\text{A.19})$$

This leads to

$$C_3^u(\bar{u}, \bar{v}) = \begin{bmatrix} -\frac{1}{2}\bar{u}_{b,j} & \bar{u}_{b,j} - \bar{u}_{1/2,j} & -\bar{u}_{b,j} + \bar{u}_{1/2,j} & \frac{1}{2}\bar{u}_{3/2,j} \\ -2\bar{u}_{b,j} + \bar{u}_{1/2,j} & \bar{u}_{b,j} - \frac{1}{2}\bar{u}_{1/2,j} & 0 & \ddots \\ -\bar{u}_{1/2,j} & \frac{1}{2}\bar{u}_{1/2,j} & 0 & 0 \\ -\frac{1}{2}\bar{u}_{3/2,j} & 0 & 0 & \ddots \end{bmatrix}. \quad (\text{A.20})$$

In case  $\bar{u}_{b,j} = 0$  the matrix is completely skew-symmetric, and  $u^T C_3^u u = 0$ : the discretization mimics exactly the property that the convective terms do not contribute to the global energy balance. For  $\bar{u}_{b,j} \neq 0$  we obtain

$$u^T C_3^u u = -\frac{1}{2}\bar{u}_{b,j} \left( u_{b,j}^2 + 2u_{b,j}(u_{1,j} + u_{2,j}) - 2u_{1,j}u_{2,j} \right). \quad (\text{A.21})$$

Taylor expanding  $u_{1,j}$  and  $u_{2,j}$  around  $u_{b,j}$  gives

$$u^T C_3^u u = -\frac{3}{2} \bar{u}_{b,j} u_{b,j}^2 + \mathcal{O}(h^2). \quad (\text{A.22})$$

A similar analysis can be carried out for the  $v$ -component near a vertical boundary. The discretization of the volume surrounding  $i = 2$  requires an expression for  $v_{0,j}$ , since  $v_{3/2,j} = \frac{1}{2}(v_{0,j} + v_{3,j})$ . We write the general expression

$$v_{3/2,j} = \delta_0 v_{b,j} + \delta_1 v_{1,j} + \delta_2 v_{2,j} + \delta_3 v_{3,j}, \quad (\text{A.23})$$

and determine the coefficients  $\delta_i$  based on symmetry requirements. From this the boundary condition for  $v_{0,j}$  can be derived. The convective matrix becomes

$$C_3^v(\bar{u}, \bar{v}) = \begin{bmatrix} -2\bar{u}_{b,j+1/2} + \delta_0 \bar{u}_{1,j} & \delta_1 \bar{u}_{1,j+1/2} + \frac{1}{2} \bar{u}_{2,j+1/2} & \delta_2 \bar{u}_{1,j+1/2} & \delta_3 \bar{u}_{1,j+1/2} & \frac{1}{2} \bar{u}_{2,j+1/2} \\ -\bar{u}_{b,j+1/2} & \frac{1}{2} (\bar{v}_{1,j+3/2} - \bar{v}_{1,j-3/2}) & 0 & \frac{1}{2} \bar{v}_{2,j+3/2} - \bar{v}_{2,j-3/2} & 0 \\ -\delta_0 \bar{u}_{1,j+1/2} & 0 & \frac{1}{2} \bar{u}_{3,j+1/2} & 0 & \ddots \\ 0 & -\delta_1 \bar{u}_{1,j+1/2} & -\delta_2 \bar{u}_{1,j+1/2} & \frac{1}{2} \bar{u}_{4,j+1/2} - \delta_3 \bar{u}_{1,j+1/2} & \vdots \\ 0 & -\frac{1}{2} \bar{u}_{2,j+1/2} & 0 & \frac{1}{2} (\bar{v}_{2,j+3/2} - \bar{v}_{2,j-3/2}) & 0 \end{bmatrix}. \quad (\text{A.24})$$

The rows represent the discretizations for the finite volumes around  $v_1, v_2$ , etc., whereas the columns represent the contribution from  $v_b, v_1, v_2$ , etc. Requiring that the convective discretization does not contribute to the kinetic energy in case  $\bar{u}_b = 0$  gives the following solution for the coefficients:  $\delta_1 = \delta_3 = \frac{1}{2}, \delta_0 = \delta_2 = 0$ . This means that the interpolation for  $v_{3/2,j}$  reads

$$v_{3/2,j} = \frac{1}{2}(v_{1,j} + v_{3,j}), \quad (\text{A.25})$$

and the implied boundary condition is

$$v_{0,j} = v_{1,j}. \quad (\text{A.26})$$

So, like the fluxes  $\bar{v}$  and  $\bar{v}$  (equations (4.12)-(4.13)), the ghost value  $v_{0,j}$  is obtained from a Neumann-type boundary condition. Rewriting the convective matrix using the divergence-free condition leads to

$$C_3^v(\bar{u}, \bar{v}) = \begin{bmatrix} -2\bar{u}_{b,j+1/2} & \bar{u}_{b,j+1/2} & 0 & \frac{1}{2}\bar{u}_{1,j+1/2} & \frac{1}{2}\bar{u}_{2,j+1/2} \\ -\bar{u}_{b,j+1/2} & 0 & \frac{1}{2}\bar{u}_{b,j+1/2} & 0 & \dots \\ 0 & -\frac{1}{2}\bar{u}_{1,j+1/2} & -\frac{1}{2}\bar{u}_{2,j+1/2} & 0 & \dots \\ 0 & 0 & -\frac{1}{2}\bar{u}_{2,j+1/2} & \dots & \dots \end{bmatrix}. \quad (\text{A.27})$$

For  $\bar{u}_{b,j+1/2} = 0$  the matrix is completely skew-symmetric, and  $v^T C_3^v v = 0$ . For nonhomogeneous boundary data the contribution to the energy equation is

$$v^T C_3^v v = \bar{u}_{b,j+1/2} \left( -2v_{b,j}v_{1,j} - v_{b,j}v_{2,j} + v_{1,j}^2 + \frac{1}{2}v_{2,j}^2 \right). \quad (\text{A.28})$$

Using Taylor expansions for  $v_1$  and  $v_2$  around  $v_b$  then gives

$$v^T C_3^v v = -\frac{3}{2}v_{b,j}^2 \bar{u}_{b,j+1/2} + \mathcal{O}(h^2). \quad (\text{A.29})$$

## A.2.2 Diffusive terms

The contribution of the diffusive terms to the energy equation reads

$$\begin{aligned}
u^T D^u u = & \\
& u_{b,j} \left[ \alpha_1 \left( \left( \frac{\partial \bar{u}}{\partial x} \right)_{1/2,j} - \left( \frac{\partial \bar{u}}{\partial x} \right)_{b,j} \right) \right. \\
& \quad \left. - \left( \left( \frac{\partial \bar{u}}{\partial x} \right)_{-1/2,j} + \left( \frac{\partial \bar{u}}{\partial x} \right)_{1/2,j} + \left( \frac{\partial \bar{u}}{\partial x} \right)_{3/2,j} - 3 \left( \frac{\partial \bar{u}}{\partial x} \right)_{b,j} \right) \right] + \\
& u_{1,j} \left[ \alpha_1 \left( \left( \frac{\partial \bar{u}}{\partial x} \right)_{3/2,j} - \left( \frac{\partial \bar{u}}{\partial x} \right)_{1/2,j} \right) - \left( \left( \frac{\partial \bar{u}}{\partial x} \right)_{5/2,j} - \left( \frac{\partial \bar{u}}{\partial x} \right)_{-1/2,j} \right) \right] + \\
& u_{2,j} \left[ \alpha_1 \left( \left( \frac{\partial \bar{u}}{\partial x} \right)_{5/2,j} - \left( \frac{\partial \bar{u}}{\partial x} \right)_{3/2,j} \right) - \left( \left( \frac{\partial \bar{u}}{\partial x} \right)_{7/2,j} - \left( \frac{\partial \bar{u}}{\partial x} \right)_{1/2,j} \right) \right] + \dots
\end{aligned} \tag{A.30}$$

Regrouping gives

$$\begin{aligned}
u^T D^u u = & - \left( \frac{\partial \bar{u}}{\partial x} \right)_{b,j} (\alpha_1 - 9) u_{b,j} + \left( \frac{\partial \bar{u}}{\partial x} \right)_{-1/2,j} 3(u_{1,j} - u_{b,j}) \\
& - \left( \frac{\partial \bar{u}}{\partial x} \right)_{1/2,j} (\alpha_1(u_{1,j} - u_{b,j}) - 3(u_{2,j} - u_{b,j})) \\
& - \left( \frac{\partial \bar{u}}{\partial x} \right)_{3/2,j} (\alpha_1(u_{2,j} - u_{1,j}) - 3(u_{3,j} - u_{b,j})) + \dots
\end{aligned} \tag{A.31}$$

The first term is recognized as the discrete counterpart of the boundary integral in (4.39). The last term is handled like the interior terms. Since the first order derivatives are discretized as

$$\left( \frac{\partial \bar{u}}{\partial x} \right)_{i+1/2,j} = \frac{\Delta y}{72\Delta x} (\alpha_1 (u_{i+1,j} - u_{i,j}) - 3(u_{i+2,j} - u_{i-1,j})), \tag{A.32}$$

it can be written as

$$\begin{aligned}
- \left( \frac{\partial \bar{u}}{\partial x} \right)_{3/2,j} (\alpha_1(u_{2,j} - u_{1,j}) - 3(u_{3,j} - u_{b,j})) = \\
- 72\Delta x \Delta y \left( \frac{9}{8} \frac{u_{2,j} - u_{1,j}}{\Delta x} - \frac{1}{8} \frac{u_{3,j} - u_{b,j}}{3\Delta x} \right)^2. \tag{A.33}
\end{aligned}$$

This is the discrete equivalent of  $\int \left( \frac{\partial u}{\partial x} \right)^2 dx$ , centered around  $(i + 3/2, j)$ . The second and third term in (A.31) should provide a similar approximation to a volume

centered around  $(i + 1/2, j)$ . Evaluating (A.32) for  $i = 0$  and  $i = -1$  and 'standard' boundary conditions

$$u_{-2,j} = 2u_{b,j} - u_{2,j}, \quad (\text{A.34})$$

$$u_{-1,j} = 2u_{b,j} - u_{1,j}, \quad (\text{A.35})$$

gives

$$\left(\frac{\partial \bar{u}}{\partial x}\right)_{1/2,j} = \frac{\Delta y}{72\Delta x} \left( \alpha_1 (u_{1,j} - u_{b,j}) - 3(u_{2,j} + u_{1,j} - 2u_{b,j}) \right), \quad (\text{A.36})$$

$$\left(\frac{\partial \bar{u}}{\partial x}\right)_{-1/2,j} = \left(\frac{\partial \bar{u}}{\partial x}\right)_{1/2,j}. \quad (\text{A.37})$$

Adding the second and third term in (A.31) gives

$$\begin{aligned} \left(\frac{\partial \bar{u}}{\partial x}\right)_{-1/2,j} 3(u_{1,j} - u_{b,j}) - \left(\frac{\partial \bar{u}}{\partial x}\right)_{1/2,j} (\alpha_1(u_{1,j} - u_{b,j}) - 3(u_{2,j} - u_{b,j})) \\ = - \left(\frac{\partial \bar{u}}{\partial x}\right)_{1/2,j} (\alpha_1(u_{1,j} - u_{b,j}) - 3(u_{2,j} + u_{1,j} - 2u_{b,j})) \\ = -72\Delta x \Delta y \left( \frac{9}{8} \frac{u_{1,j} - u_{b,j}}{\Delta x} - \frac{1}{8} \frac{u_{2,j} + u_{1,j} - 2u_{b,j}}{3\Delta x} \right)^2. \end{aligned} \quad (\text{A.38})$$

Taking all steps together, equation (A.30) can be written as

$$\begin{aligned} u^T D^u u = - \left(\frac{\partial \bar{u}}{\partial x}\right)_{b,j} (\alpha_1 - 9)u_{b,j} + \dots \\ - 72\Delta x \Delta y \left[ \left( \frac{9}{8} \frac{u_{1,j} - u_{b,j}}{\Delta x} - \frac{1}{8} \frac{u_{2,j} + u_{1,j} - 2u_{b,j}}{3\Delta x} \right)^2 + \right. \\ \left. \left( \frac{9}{8} \frac{u_{2,j} - u_{1,j}}{\Delta x} - \frac{1}{8} \frac{u_{3,j} - u_{b,j}}{3\Delta x} \right)^2 + \dots \right]. \end{aligned} \quad (\text{A.39})$$

This is the discrete equivalent of the right hand side of (4.39).

For the  $v$ -component a similar analysis can be carried out.



## A.3 GENERALIZED TAYLOR EXPANSIONS

We follow [213] for the generalized Taylor expansion of a function with discontinuities. Assume a function  $u(x)$  on the domain  $x \in [x_0, x_1]$  has a discontinuity at  $\alpha$ . Then  $u(x)$  can be Taylor expanded around  $x_0$  as

$$u(x) = \sum_{n=0}^{\infty} \frac{u^{(n)}(x_0)}{n!} (x - x_0)^n + H(x - \alpha) \sum_{n=0}^{\infty} \frac{[u^{(n)}(\alpha)]}{n!} (x - \alpha)^n, \quad (\text{A.40})$$

where  $H(x)$  is the Heaviside function. We will focus on the first 3 terms in this infinite sum.

## A.3.1 Approximating integrals

We express the integral of  $u(x)$  over a finite volume  $[x_1, x_2]$  in terms of the midpoint value  $u(x_0)$  ( $x_0 = \frac{1}{2}(x_1 + x_2)$ ) and its derivatives and in terms of  $[u(\alpha)]$  and its derivatives (assuming  $\alpha > x_0$ ):

$$\begin{aligned} \int_{x_1}^{x_2} u(x) dx &= \int_{x_1}^{x_2} \left( u(x_0) + u_x(x_0)(x - x_0) + \frac{1}{2}u_{xx}(x_0)(x - x_0)^2 + \mathcal{O}((x - x_0)^3) \right) dx + \\ &\int_{\alpha}^{x_2} \left( [u(\alpha)] + [u_x(\alpha)](x - \alpha) + \frac{1}{2}[u_{xx}(\alpha)](x - \alpha)^2 + \mathcal{O}((x - \alpha)^3) \right) dx \end{aligned} \quad (\text{A.41})$$

$$\begin{aligned} &= u(x_0)\Delta x + \frac{1}{24}u_{xx}(x_0)\Delta x^3 + [u(\alpha)]\Delta\alpha + \frac{1}{2}[u_x(\alpha)]\Delta\alpha^2 + \\ &\frac{1}{6}[u_{xx}(\alpha)]\Delta\alpha^3 + \mathcal{O}(\Delta x^5), \end{aligned} \quad (\text{A.42})$$

where  $\Delta\alpha = x_2 - \alpha$ . The integral can therefore be approximated to second order accuracy by taking the midpoint approximation

$$\int_{x_1}^{x_2} u(x) dx = u(x_0)\Delta x + [u(\alpha)]\Delta\alpha + \frac{1}{2}[u_x(\alpha)]\Delta\alpha^2 + \mathcal{O}(\Delta x^3). \quad (\text{A.43})$$

In practice, it is allowed to ‘loose’ one order of accuracy at the volumes near the interface (see e.g. [96]), so that for second-order accuracy it is sufficient to use the following adapted midpoint rule:

$$\int_{x_1}^{x_2} u(x) dx \approx u(x_0)\Delta x + [u(\alpha)]\Delta\alpha. \quad (\text{A.44})$$

Equations (A.43) and (A.44) also hold for the case  $\alpha < x_0$  if we define  $\Delta\alpha$  as

$$\Delta\alpha = \text{sgn}(\alpha - x_0) \min(\alpha - x_1, x_2 - \alpha). \quad (\text{A.45})$$

An alternative approximation to the integral can be constructed by employing the trapezoidal rule. Expanding  $u(x_1)$  and  $u(x_2)$  in terms of  $u(x_0)$  we get

$$\begin{aligned} \frac{1}{2}\Delta x(u(x_1) + u(x_2)) &= u(x_0)\Delta x + \frac{1}{8}u_{xx}(x_0)\Delta x^3 + \\ &\frac{1}{2}[u(\alpha)]\Delta x + \frac{1}{2}[u_x(\alpha)]\Delta x(x_2 - \alpha) + \mathcal{O}(\Delta x^5). \end{aligned} \quad (\text{A.46})$$

By comparison with the exact integral (A.42), the adapted trapezoidal rule can be derived to read

$$\begin{aligned} \int_{x_1}^{x_2} u(x) \, dx &\approx \frac{1}{2}\Delta x(u(x_1) + u(x_2)) + \\ &\frac{1}{2}[u(\alpha)]((x_2 - \alpha) - (\alpha - x_1)) - \frac{1}{2}[u_x(\alpha)](x_2 - \alpha)(\alpha - x_1). \end{aligned} \quad (\text{A.47})$$

### A.3.2 Approximating derivatives

In order to find an expression for  $u_x$  at a finite volume face  $x_{j+1/2}$ , we expand  $u(x_j)$  and  $u(x_{j+1})$  in terms of  $u(x_{j+1/2})$ . We assume that the discontinuity at  $x = \alpha$  lies right of  $x_{j+1/2}$  ( $\alpha > x_{j+1/2}$ ), see figure 8.1. Following (A.40) this results into

$$u(x_j) = u(x_{j+1/2}) - u_x(x_{j+1/2}) \left( \frac{1}{2}\Delta x \right) + \frac{1}{2}u_{xx}(x_{j+1/2}) \left( \frac{1}{2}\Delta x \right)^2 + \mathcal{O}(\Delta x^3), \quad (\text{A.48})$$

$$\begin{aligned} u(x_{j+1}) &= u(x_{j+1/2}) + u_x(x_{j+1/2}) \left( \frac{1}{2}\Delta x \right) + \frac{1}{2}u_{xx}(x_{j+1/2}) \left( \frac{1}{2}\Delta x \right)^2 + \\ &[u(\alpha)] + [u_x(\alpha)](x_{j+1} - \alpha) + \frac{1}{2}[u_{xx}(\alpha)](x_{j+1} - \alpha)^2 + \mathcal{O}(\Delta x^3). \end{aligned} \quad (\text{A.49})$$

A second order approximation to  $u_x(x_{j+1/2})$  then follows by taking

$$\begin{aligned} \frac{u(x_{j+1}) - [u(\alpha)] - [u_x(\alpha)](x_{j+1} - \alpha) - \frac{1}{2}[u_{xx}(\alpha)](x_{j+1} - \alpha)^2 - u(x_j)}{\Delta x} = \\ u_x(x_{j+1/2}) + \mathcal{O}(\Delta x^2). \end{aligned} \quad (\text{A.50})$$

Again, since a local truncation error of  $\mathcal{O}(\Delta x)$  is sufficient for the volumes near the interface, we take in practice

$$\frac{u(x_{j+1}) - [u(\alpha)] - [u_x(\alpha)](x_{j+1} - \alpha) - u(x_j)}{\Delta x} = u_x(x_{j+1/2}) + \mathcal{O}(\Delta x). \quad (\text{A.51})$$

In case  $\alpha < x_{j+1/2}$  one should change  $x_{j+1}$  to  $x_j$  in the last equation.

## DETAILS OF RUNGE-KUTTA METHODS

---

### B.1 W-TRANSFORMATION

Hairer and Wanner introduced the  $W$ -transformation in the study of the construction of high-order implicit Runge-Kutta methods [63]. The  $s \times s$  matrix  $W$  is a Vandermonde matrix defined by

$$w_{ij} = \tilde{P}_{j-1}(c_i), \quad i = 1, \dots, s, \quad j = 1, \dots, s, \quad (\text{B.1})$$

where  $\tilde{P}_k(x)$  are the normalized shifted Legendre polynomials defined by

$$\tilde{P}_k(x) = \sqrt{2k+1} \sum_{j=0}^k (-1)^{j+k} \binom{k}{j} \binom{j+k}{j} x^j, \quad k = 0, 1, \dots \quad (\text{B.2})$$

Application of  $W$  on the coefficient matrix  $A$  (the  $W$ -transformation) leads to a matrix,  $X^*$ , with a special structure:

$$X^* = W^{-1}AW. \quad (\text{B.3})$$

This matrix can be related to the simplifying conditions  $C(\eta)$  and  $D(\zeta)$  and is therefore linked to the order of the method ([63], Theorem 5.11). The matrix  $X$ , defined by

$$X = W^T B A W = J X^*, \quad \text{where } J = W^T B W, \quad (\text{B.4})$$

is closely resembling  $X^*$  and can be used to characterize symmetry, symplecticity and stability properties. In case the quadrature formula is of order  $\geq 2s - 1$ , then  $J = I$  and  $X = X^*$ . The special structure of matrix  $X$  is especially evident for the Gauss methods, for which it is denoted by  $X_G$ :

$$X_G = \begin{pmatrix} 1/2 & -\zeta_1 & & & \\ \zeta_1 & 0 & -\zeta_2 & & \\ & \zeta_2 & \ddots & \ddots & \\ & & \ddots & 0 & -\zeta_{s-1} \\ & & & \zeta_{s-1} & 0 \end{pmatrix}, \quad (\text{B.5})$$

where  $\zeta_k = \frac{1}{2\sqrt{4k^2-1}}$ . The conditions for symplecticity, algebraic stability and symmetry in terms of the  $X$  matrix read [33, 63]:

$$X + X^T - e_1 e_1^T = 0 \qquad \text{symplectic,} \qquad \text{(B.6)}$$

$$X + X^T - e_1 e_1^T \geq 0 \qquad \text{algebraically stable,} \qquad \text{(B.7)}$$

$$X_{kl} = 0 \text{ for } k + l \neq 2 \text{ is even, } \hat{P}c = e - c, \hat{P}b = b \qquad \text{symmetric,} \qquad \text{(B.8)}$$

where  $\hat{P}$  is the permutation matrix with elements  $\hat{p}_{ij} = \delta_{i,s+1-j}$ ,  $e = (1, \dots, 1)^T$  and  $e_1 = (1, 0, \dots, 0)^T$ .

### B.2 ENERGY-CONSERVING (S)DIRK METHODS

We are interested in algebraically stable energy-conserving (S)DIRK methods, i.e., having  $E = 0$  and  $b \geq 0$ . Energy-conserving DIRK methods can be derived by employing the  $W$ -transform of the previous section, but for ‘low’ order methods (say lower than 4) it is somewhat easier to take the following approach. Evaluating the symplecticity condition

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \qquad \text{(B.9)}$$

for DIRK methods ( $a_{ij} = 0$  for  $j > i$ ) leads to a Butcher tableau with the following structure:

$$\begin{array}{c|cccc}
 b_1/2 & & & & \\
 b_1 + b_2/2 & b_1 & b_2/2 & & \\
 b_1 + b_2 + b_3/2 & b_1 & b_2 & b_3/2 & \\
 \vdots & \vdots & \vdots & \vdots & \ddots \\
 b_1 + \dots + b_{s-1} + b_s/2 & b_1 & b_2 & b_3 & \dots & b_s/2 \\
 \hline
 & b_1 & b_2 & b_3 & \dots & b_s
 \end{array} \qquad \text{(B.10)}$$

A particular family which satisfies this form is given by  $b_i = 1/s$  ( $j < i$ ), but it is limited to second order (it is simply repeated application of the midpoint method). All high-order energy-conserving DIRK methods obtained in literature, e.g. [38, 135, 187, 53], have negative  $b$ -coefficients and are therefore not algebraically stable - they are only useful for pure convective-type problems, where all eigenvalues lie on the imaginary axis. This is confirmed by Chan et al. [34], who showed by using theory on order stars that *all algebraically stable energy-conserving DIRK methods are limited to second order*, independent of  $s$ . This negative result also holds for the weaker concept of  $A$ -stability. For energy-conserving DIRK methods the stability function reads (see equation (7.22))

$$R(z) = \frac{\prod_{i=1}^s (1 + \frac{1}{2} b_i z)}{\prod_{i=1}^s (1 - \frac{1}{2} b_i z)}, \qquad \text{(B.11)}$$

and its poles are given by  $z = 2/b_i$ .  $A$ -stability requires that all poles of the stability function  $R(z)$  lie in the positive half plane  $\mathbb{C}^+$  [63]. The condition  $b_i \geq 0$  is therefore not only necessary for algebraic stability, but also for  $A$ -stability, when energy-conserving DIRK methods are considered.

B.3 STABILITY OF ADDITIVE RUNGE-KUTTA METHODS

B.3.1 Algebraic stability

Consider equation (7.29), repeated here for convenience:

$$\begin{aligned} \|u_{n+1}\|^2 &= \|u_n\|^2 + 2\Delta t \sum_{i=1}^s b_i ((U_i, PF_i) + (U_i, P\hat{F}_i)) \\ &\quad - \Delta t^2 \left( \sum_{i,j=1}^s e_{ij}(PF_i, PF_j) + \sum_{i,j=1}^s \hat{e}_{ij}(P\hat{F}_i, P\hat{F}_j) + 2 \sum_{i,j=1}^s \bar{e}_{ij}(PF_i, P\hat{F}_j) \right). \end{aligned} \quad (\text{B.12})$$

The second term on the right-hand side of this equation,  $2\Delta t(\dots)$ , is zero for pure convection problems and negative if diffusion is present, as long as  $b_i \geq 0$ . The sign of the third term,  $-\Delta t^2(\dots)$ , depends on the matrices  $E$ ,  $\hat{E}$  and  $\bar{E}$ . The entire term can be written as

$$-\Delta t^2 \sum_{i,j=1}^{2s} \bar{e}_{ij}(x_i, x_j), \quad (\text{B.13})$$

by defining  $x = (PF_1, \dots, PF_s, P\hat{F}_1, \dots, P\hat{F}_s)^T$ , and

$$\bar{E} = \begin{pmatrix} E & \bar{E} \\ \bar{E} & \hat{E} \end{pmatrix}. \quad (\text{B.14})$$

Unconditional algebraic stability requires that  $\bar{E}$  is nonnegative definite (nonnegative definiteness of the individual blocks is not sufficient, see [56]). We now encounter the following negative result.

**Theorem B.3.1.** (*Algebraic stability of additive Runge-Kutta methods*) Consider an additive Runge-Kutta method (7.23)-(7.24) with  $b = \hat{b}$  and the Butcher tableau  $\hat{A}$  for the convective terms such that  $\hat{E} = 0$ . Then the method cannot be algebraically stable, except if it is trivial (i.e.  $A = \hat{A}$ ).

*Proof.* For algebraic stability  $\bar{E}$  should be nonnegative definite. Given that  $\hat{E} = 0$  this leads to the requirement  $\bar{E} = 0$ . However,  $\bar{E} = B\hat{A} + \hat{A}^T B - bb^T = 0$  leads to  $\bar{E} = B\hat{A} + A^T B - bb^T = B(\hat{A} - A)$ , which is zero only in case  $\hat{A} = A$ .  $\square$

In [56] it is shown (for general additive Runge-Kutta methods) how the term

$$2\Delta t \sum_{i=1}^s b_i ((U_i, PF_i) + (U_i, P\hat{F}_i)) \tag{B.15}$$

can be used to obtain *conditional* algebraic stability, by requiring a more strict condition than  $(F_i, U_i) \leq 0$ , namely

$$(F_i, U_i) \leq \mu_i \|F_i\|^2, \quad \mu_i < 0. \tag{B.16}$$

Such a condition permits one to add diagonal matrices to the diagonal blocks of  $\bar{E}$ , and with a condition on  $\Delta t$  this can lead to a nonnegative definite  $\bar{E}$ . In our case, (B.16) cannot hold for the convective terms because  $(C(U_i, U_i), U_i) = 0$ , and consequently the zero block in  $\bar{E}$  caused by  $\hat{E}$  will remain zero. Conditional algebraic stability can therefore not be proven either.

### B.3.2 Linear stability

Consider the expression for the amplification factor for an additive Runge-Kutta method, equation (7.32):

$$R(z, \hat{z}) \equiv \frac{u_{n+1}}{u_n} = \frac{\text{Det}(I - zA - \hat{z}\hat{A} + (z + \hat{z})eb^T)}{\text{Det}(I - zA - \hat{z}\hat{A})} = \frac{N(z, \hat{z})}{D(z, \hat{z})}, \tag{B.17}$$

Both  $N(z, \hat{z})$  and  $D(z, \hat{z})$  can be expressed as polynomials of the form:

$$N(z, \hat{z}) = \sum_{i=0}^s \left( \sum_{j=0}^{s-i} n_{ij} \hat{z}^j \right) z^i, \quad D(z, \hat{z}) = \sum_{i=0}^s \left( \sum_{j=0}^{s-i} d_{ij} \hat{z}^j \right) z^i, \tag{B.18}$$

where the coefficients  $n_{ij}$  and  $d_{ij}$  associated with the highest order terms are

$$n_{0s} = (-1)^s \text{Det}(\hat{Q}), \quad n_{s0} = (-1)^s \text{Det}(Q), \tag{B.19}$$

$$d_{0s} = (-1)^s \text{Det}(\hat{A}), \quad d_{s0} = (-1)^s \text{Det}(A). \tag{B.20}$$

If the original method  $\mathcal{A}$  is  $L$ -stable (i.e., it has  $\text{Det}(Q) = 0$  and  $\text{Det}(A) \neq 0$ ), then these expressions show that the resulting additive method is still  $L$ -stable, i.e., we have for fixed  $\hat{z}$ :

$$\lim_{z \rightarrow -\infty} R(z, \hat{z}) = 0, \tag{B.21}$$

provided that the additive method is  $A$ -stable. Unfortunately, as in the case of algebraic stability,  $A$ -stability of the additive method does not follow from the  $A$ -stability of the two separate methods due to the coupling terms in  $N(z, \hat{z})$  and  $D(z, \hat{z})$ .

## B.4 PARTIAL RUNGE-KUTTA METHODS

Consider an ODE of the form

$$\dot{u}(t) = f(u(t), t). \quad (\text{B.22})$$

A Runge-Kutta method applied to this ODE reads:

$$U_i = u_n + \Delta t \sum_{j=1}^s a_{ij} f(U_j, t_j), \quad (\text{B.23})$$

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^s b_i f(U_i, t_i). \quad (\text{B.24})$$

See section 5.3 for notation. The summations can be seen as approximations to the following integrals:

$$\Delta t \sum_{j=1}^s a_{ij} f(U_j, t_j) \approx \int_{t^n}^{t_i} f(u(t), t) dt, \quad (\text{B.25})$$

$$\Delta t \sum_{i=1}^s b_i f(U_i, t_i) \approx \int_{t^n}^{t^{n+1}} f(u(t), t) dt. \quad (\text{B.26})$$

We consider the case where  $f$  can be split as  $f(u(t), t) = h(u(t), t) + g(t)$ , and  $g(t)$  can be integrated exactly in time. We then adapt formulation (B.23)-(B.24) as follows:

$$\hat{U}_i = u_n + \int_{t^n}^{t_i} g(t) dt + \Delta t \sum_{j=1}^s a_{ij} h(\hat{U}_j, t_j), \quad (\text{B.27})$$

$$\hat{u}_{n+1} = u_n + \int_{t^n}^{t^{n+1}} g(t) dt + \Delta t \sum_{i=1}^s b_i h(\hat{U}_i, t_i). \quad (\text{B.28})$$

We call this a *partial* Runge-Kutta method. Since this is formally not a Runge-Kutta method, it is questionable if the classical order conditions are still valid. Intuitively one might expect that the partial RK method is more accurate than the standard RK method, since part of  $f(u(t), t)$  is integrated exactly. This can be investigated as follows. The Taylor expansion of the exact solution is given by

$$u(t + \Delta t) = u(t) + \Delta t \dot{u}(t) + \frac{1}{2} \Delta t^2 \ddot{u}(t) + \mathcal{O}(\Delta t^3), \quad (\text{B.29})$$

where  $\dot{u}(t) = f$  and  $\ddot{u}(t) = \dot{f} + f'f$  (we leave out the arguments  $(u(t), t)$  for notational convenience). A  $'$  denotes differentiation with respect to  $u$ . The Taylor ex-

pansion of the numerical solution is as follows. First write the partial Runge-Kutta method as

$$\hat{z}(\Delta t) = u(t) + G(t + \Delta t) - G(t) + \Delta t \sum_{i=1}^s b_i H_i, \quad (\text{B.30})$$

where  $G(t)$  is the primitive of  $g(t)$  and

$$H_i = h(\hat{U}_i, t_i), \quad (\text{B.31})$$

$$\hat{U}_i = u(t) + G(t + c_i \Delta t) - G(t) + \Delta t \sum_{j=1}^s a_{ij} H_j. \quad (\text{B.32})$$

$\hat{z}$ ,  $H_i$  and  $\hat{U}_i$  are functions of  $\Delta t$ , with  $u(t)$  fixed. The Taylor expansion of the numerical solution is given by

$$\hat{z}(\Delta t) = \hat{z}(0) + \Delta t \hat{z}'(0) + \frac{1}{2} \Delta t^2 \hat{z}''(0) + \mathcal{O}(\Delta t^3), \quad (\text{B.33})$$

where ' indicates here differentiation with respect to  $\Delta t$ . To find the first terms of (B.33) we proceed as follows. From (B.30) we have  $\hat{z}(0) = u(t)$ . Differentiating (B.30) with respect to  $\Delta t$  gives

$$\hat{z}'(\Delta t) = \sum_i b_i H_i + \Delta t \sum_i b_i H_i' + g + \Delta t \dot{g} + \frac{1}{2} \Delta t^2 \ddot{g} + \mathcal{O}(\Delta t^3), \quad (\text{B.34})$$

so

$$\hat{z}'(0) = \sum_i b_i H_i(\Delta t = 0) + g = \left( \sum_i b_i \right) h + g. \quad (\text{B.35})$$

Note that for a 'standard' Runge-Kutta method, equations (B.23)-(B.24), the  $g$  term would be inside the summation, i.e., one would have

$$z'(0) = \left( \sum_i b_i \right) (h + g) = \left( \sum_i b_i \right) f. \quad (\text{B.36})$$

In case the first order (consistency) condition  $\sum_i b_i = 1$  is satisfied,  $\hat{z}'(0) = z'(0) = f$ , and both the partial and the standard Runge-Kutta method are first order accurate (by comparing with (B.29)). The local truncation error is given by

$$\hat{\tau} = \Delta t \left( f - \left( \sum_i b_i \right) h - g \right) + \mathcal{O}(\Delta t^2) = \Delta t \left( 1 - \sum_i b_i \right) (f - g) + \mathcal{O}(\Delta t^2). \quad (\text{B.37})$$

Further differentiation gives

$$\hat{z}''(\Delta t) = 2 \sum_i b_i H_i' + \Delta t \sum_i b_i H_i'' + \dot{g} + \Delta t \ddot{g} + \mathcal{O}(\Delta t^2), \quad (\text{B.38})$$



where

$$H'_i = \frac{\partial H_i}{\partial \hat{U}_i} \frac{\partial \hat{U}_i}{\partial \Delta t} + \frac{\partial H_i}{\partial t_i} \frac{\partial t_i}{\partial \Delta t} \quad (\text{B.39})$$

$$= \frac{\partial H_i}{\partial \hat{U}_i} \left( c_i g + \frac{1}{2} c_i^2 \Delta t \dot{g} + \dots + \sum_j a_{ij} H_j + \Delta t \sum_j a_{ij} H'_j \right) + \frac{\partial H_i}{\partial t_i} c_i. \quad (\text{B.40})$$

Letting  $\Delta t \rightarrow 0$ , we obtain

$$\hat{z}''(0) = 2 \sum_i b_i \left( h' \left( c_i g + \sum_j a_{ij} h \right) + c_i \dot{h} \right) + \dot{g}, \quad (\text{B.41})$$

$$= 2 \left( \sum_i \sum_j b_i a_{ij} \right) (h'h + \dot{h} + h'g) + \dot{g}. \quad (\text{B.42})$$

Like in equation (B.35), the partial Runge-Kutta method has  $\dot{g}(t)$  'outside' the summation whereas in the standard Runge-Kutta method it is inside the summation:

$$z''(0) = \left( 2 \sum_i \sum_j b_i a_{ij} \right) (\dot{f} + f'f). \quad (\text{B.43})$$

Note that  $\dot{f} + f'f = h'h + \dot{h} + h'g + \dot{g}$ . Again, if the second-order condition  $\sum_i \sum_j b_i a_{ij} = \sum_i b_i c_i = \frac{1}{2}$  is satisfied,  $\hat{z}''(0) = z''(0) = \dot{f} + f'f$ , and both formulations are second order. This process can be repeated (also for systems of equations, instead of scalar equations), showing that each equation for  $\hat{z}^{(n)}(0)$  contains the  $(n-1)$ th derivative of  $g$ , which is not preceded by the coefficients of the Runge-Kutta method: *the order of accuracy of a partial Runge-Kutta method* (B.27)-(B.28) *is the same as the order of accuracy of the original Runge-Kutta method* (B.23)-(B.24).

However, the methods are not equivalent, since the local truncation errors are different. Imagine a first order method (like Forward Euler) that does not satisfy the second order condition  $\sum_i b_i c_i = \frac{1}{2}$ . The local truncation error is given by

$$\begin{aligned} \hat{\tau} &= \frac{1}{2} \Delta t^2 \left( \dot{f} + f'f - 2 \left( \sum_i b_i c_i \right) (h'h + \dot{h} + h'g) - \dot{g} \right) + \mathcal{O}(\Delta t^3), \\ &= \frac{1}{2} \Delta t^2 \left( 1 - 2 \sum_i b_i c_i \right) (\ddot{u} - \dot{g}) + \mathcal{O}(\Delta t^3). \end{aligned} \quad (\text{B.44})$$

The effect of the partial Runge-Kutta method is to change the local truncation error; whether it increases or decreases is problem specific (it depends on  $\dot{g}(t)$ ). In the special case  $f(u(t), t) = g(t)$ , then  $\dot{u} = g$ ,  $\ddot{u} = \dot{g}$ , etc.; the solution is exact and the local truncation error vanishes.



## BIBLIOGRAPHY

---

- [1] S. Abarbanel, D. Gottlieb, and M.H. Carpenter. On the removal of boundary errors caused by Runge-Kutta integration of nonlinear partial differential equations. *SIAM J. Sci. Comput.*, 17(3):777–782, 1996.
- [2] P.-H. Alfredsson and J.-Å. Dahlberg. A preliminary wind tunnel study of windmill wake dispersion in various flow conditions. Technical Note AU-1499, part 7, FFA, Stockholm, Sweden, 1979.
- [3] I. Ammara, C. Leclerc, and C. Masson. A viscous three-dimensional method for the aerodynamic analysis of wind farms. *J. Sol. Energy Eng.*, 124:345–356, 2002.
- [4] A. Arakawa. Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I. *J. Comput. Phys.*, 1(1):119–143, 1966.
- [5] S. Armfield and R. Street. An analysis and comparison of the time accuracy of fractional-step methods for the Navier-Stokes equations on staggered grids. *Int. J. Numer. Meth. Fluids*, 38:255–282, 2002.
- [6] U. Ascher. On numerical differential algebraic problems with application to semiconductor device simulation. *SIAM J. Numer. Anal.*, 26:517–538, 1989.
- [7] U.M. Ascher and L.R. Petzold. Projected implicit Runge-Kutta methods for differential-algebraic equations. *SIAM J. Numer. Anal.*, 28(4):1097–1120, 1991.
- [8] U.M. Ascher and L.R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, 1998.
- [9] U.M. Ascher, S.J. Ruuth, and R.J. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl. Num. Math.*, 25:151–167, 1997.
- [10] O. Axelsson and L. Kolotilina. Monotonicity and discretization error estimates. *SIAM J. Numer. Anal.*, 27(6):1591–1611, 1990.
- [11] S. Balay et al. PETSc Users Manual. Technical Report ANL-95/11 - Revision 3.3, Argonne National Laboratory, 2012.
- [12] R.J. Barthelmie et al. Flow and wakes in large wind farms in complex terrain and offshore. In *EWEC, Brussels*, 2008.
- [13] R.J. Barthelmie et al. Modelling the impact of wakes on power output at Nysted and Horns Rev. In *EWEC, Marseille*, 2009.

- [14] A. Bechmann. *Large-eddy simulation of atmospheric flow over complex terrain*. PhD thesis, Technical University of Denmark, 2006.
- [15] A. Bechmann and N.N. Sørensen. CFD simulation of the MEXICO rotor wake. In *EWEC, Marseille*, 2009.
- [16] M. Benzi, G.H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, pages 1–137, 2005.
- [17] R.P. Beyer and R.J. LeVeque. Analysis of a one-dimensional model for the immersed boundary method. *SIAM J. Numer. Anal.*, 29(2):332–364, 1992.
- [18] O. Botella and R. Peyret. Benchmark spectral results on the lid-driven cavity flow. *Comp. Fluids*, 27(4):421–433, 1998.
- [19] J. Boussinesq. Théorie de l'écoulement tourbillant. *Acad. Sci. Inst. Fr.*, 23:46–50, 1877.
- [20] V. Brasey and E. Hairer. Half-explicit Runge-Kutta methods for differential-algebraic systems of index 2. *SIAM J. Numer. Anal.*, 30(2):538–552, 1993.
- [21] D.L. Brown and M. Minion. Performance of under-resolved two-dimensional incompressible flow simulations, II. *J. Comput. Phys.*, 138:734–765, 1997.
- [22] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi. *Wind Energy Handbook*. John Wiley & Sons, 2001.
- [23] J.C. Butcher. Coefficients for the study of Runge-Kutta integration processes. *J. Australian Math. Soc.*, 3:185–201, 1963.
- [24] J.C. Butcher. Implicit Runge-Kutta processes. *Math. Comput.*, 18:50–64, 1964.
- [25] J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2003.
- [26] D. Cabezón, K. Hansen, and R.J. Barthelmie. Analysis and validation of CFD wind farm models in complex terrain. Wakes induced by topography and wind turbines. In *EWEC, Warsaw*, 2010.
- [27] D. Cabezón, J. Sanz, I. Marti, and A. Crespo. CFD modeling of the interaction between the surface boundary layer and rotor wake. Comparison of results obtained with different turbulence models and mesh strategies. In *EWEC, Marseille*, 2009.
- [28] M. Calaf, C. Meneveau, and J. Meyers. Large eddy simulation study of fully developed wind-turbine array boundary layers. *Phys. Fluids*, 22, 2010.
- [29] M.P. Calvo, J. de Frutos, and J. Novo. Linearly implicit Runge-Kutta methods for advection-reaction-diffusion equations. *Appl. Num. Math.*, 37:535–549, 2001.

- [30] F. Cameron. A Matlab package for automatically generating Runge-Kutta trees, order conditions, and truncation error coefficients. *ACM Trans. Math. Software*, 32(2):274–298, 2006.
- [31] M.H. Carpenter, D. Gottlieb, S. Abarbanel, and W.-S. Don. The theoretical accuracy of Runge-Kutta time discretizations for the initial boundary value problem: a study of the boundary error. *SIAM J. Sci. Comput.*, 16(6):1241–1252, 1995.
- [32] L.P. Chamorro and F. Porté-Agel. A wind-tunnel investigation of wind-turbine wakes: boundary-layer turbulence effects. *Boundary-Layer Meteorol.*, 132:129–149, 2009.
- [33] R.P.K. Chan. On symmetric Runge-Kutta methods of high order. *Computing*, 45:301–309, 1990.
- [34] R.P.K. Chan, H. Liu, and G. Sun. Efficient symplectic Runge-Kutta methods. *Appl. Math. Comput.*, 172:908–924, 2006.
- [35] W. Chang, F. Giraldo, and B. Perot. Analysis of an exact fractional step method. *J. Comput. Phys.*, 180:183–199, 2002.
- [36] Y.S. Chen and S.W. Kim. Computation of turbulent flow using an extended turbulence closure model. Technical Report CR-179204, NASA, 1987.
- [37] H. Choi and P. Moin. Effect of the computational time step on numerical simulation of turbulent flow. *J. Comput. Phys.*, 113:1–4, 1994.
- [38] G.J. Cooper. Stability of Runge-Kutta methods for trajectory problems. *IMA J. Numer. Anal.*, 7:1–13, 1987.
- [39] A. Crespo, J. Hernández, and S. Frandsen. Survey of modelling methods for wind turbine wakes and wind farms. *Wind Energ.*, 2:1–24, 1999.
- [40] P.A. Davidson. *Turbulence - an Introduction for Scientists and Engineers*. Oxford University Press, 2004.
- [41] K. Dekker and J.G. Verwer. *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations*. North-Holland, 1984.
- [42] O. Desjardins, G. Blanquart, G. Balarac, and H. Pitsch. High order conservative finite difference scheme for variable density low Mach number turbulent flows. *J. Comput. Phys.*, 227:7125–7159, 2008.
- [43] I. Dobrev, F. Massouh, and M. Rapin. Actuator surface hybrid model. In *The science of making torque from wind. Conference series*, volume 75, 2007.
- [44] M. Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In *Conference on Low Reynolds Number Aerodynamics, University of Notre Dame, Canada*, 1989.

- [45] M. Duponcheel, P. Orlandi, and G. Winckelmans. Time-reversibility of the Euler equations as a benchmark for energy conserving schemes. *J. Comput. Phys.*, 227:8736–8752, 2008.
- [46] P.A. Durbin. On the  $k - \epsilon$  stagnation point anomaly. *Int. J. Heat Fluid Fl.*, 17:89–90, 1996.
- [47] A. El Kasmi and C. Masson. An extended  $k - \epsilon$  model for turbulent flow through horizontal-axis wind turbines. *J. Wind Eng. and Ind. Aerodyn.*, 96:103–122, 2008.
- [48] C.P. Ellington. The aerodynamics of hovering insect flight. V. A vortex theory. *Phil. Trans. R. Soc. Lond. B*, 305:115–144, 1984.
- [49] G. España, S. Aubrun-Sanches, and P. Devinant. The meandering phenomenon of a wind turbine wake. In *EWEC, Marseille*, 2009.
- [50] F.N. Felten and T.S. Lund. Kinetic energy conservation issues associated with the collocated mesh scheme for incompressible flow. *J. Comput. Phys.*, 215:465–484, 2006.
- [51] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, 3rd revised edition, 2002.
- [52] C. Foias, O. Manley, R. Rosa, and R. Temam. *Navier-Stokes Equations and Turbulence*. Cambridge University Press, 2001.
- [53] J.M. Franco and I. Gómez. Fourth-order symmetric DIRK methods for periodic stiff problems. *Numer. Algorithms*, 32:317–336, 2003.
- [54] F. Freedman and M. Jacobson. Modification of the standard  $\epsilon$ -equation for the stable ABL through enforced consistency with Monin-Obukhov similarity theory. *Boundary-Layer Meteorol.*, 106:383–410, 2003.
- [55] R.E. Froude. On the part played in propulsion by differences of fluid pressure. *Transactions of the Institute of Naval Architects*, 30:390–405, 1889.
- [56] B. García-Celayeta, Inmaculada Higuera, and T. Roldán. Contractivity/monotonicity for additive Runge-Kutta methods: Inner product norms. *Appl. Num. Math.*, 56:862–878, 2006.
- [57] M.W. Gee et al. ML 5.0 Smoothed Aggregation User’s Guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.
- [58] B.J. Geurts. *Elements of Direct and Large-Eddy Simulation*. Edwards, 2004.
- [59] P.M. Gresho and R.L. Sani. *Incompressible Flow and the Finite Element Method. Volume 2: Isothermal Laminar Flow*. Wiley, 2000.

- [60] E. Hairer, C. Lubich, and M. Roche. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. Springer, 1989.
- [61] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2nd edition, 2006.
- [62] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer, 3rd edition, 2008.
- [63] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer, 2nd edition, 1996.
- [64] F. Ham, K. Mattsson, G. Iaccarino, and P. Moin. Towards time-stable and accurate LES on unstructured grids. In S.C. Kassinos, C.A. Langer, G. Iaccarino, and P. Moin, editors, *Complex Effects in Large Eddy Simulations*, volume 56 of *Lecture Notes in Computational Science and Engineering*, pages 235–249. Springer, 2007.
- [65] F.E. Ham, F.S. Lien, and A.B. Strong. A fully conservative second-order finite difference scheme for incompressible flow on nonuniform grids. *J. Comput. Phys.*, 177:117–133, 2002.
- [66] M.O.L. Hansen, J.N. Sørensen, S. Voutsinas, N.N. Sørensen, and H.A. Madsen. State of the art in wind turbine aerodynamics and aeroelasticity. *Prog. Aerospace Sci.*, 42:285–330, 2006.
- [67] M.O.L. Hansen, N.N. Sørensen, J.N. Sørensen, and J.A. Michelsen. Extraction of lift, drag and angle of attack from computed 3-D viscous flow around a rotating blade. In *EWEC, Dublin*, 1997.
- [68] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 8:2182–9, 1965.
- [69] J.E. Hicken, F.E. Ham, J. Militzer, and M. Koksal. A shift transformation for fully conservative methods: turbulence simulation on complex, unstructured grids. *J. Comput. Phys.*, 208:704–714, 2005.
- [70] A. Hokpunna and M. Manhart. Compact fourth-order finite volume method for numerical solutions of Navier-Stokes equations on staggered grids. *J. Comput. Phys.*, 229:7545–7570, 2010.
- [71] J.H. Horlock. *Actuator disk theory - Discontinuities in thermo-fluid dynamics*. McGraw-Hill, 1978.
- [72] T.Y. Hou and B.T.R. Wetton. Second-order convergence of a projection scheme for the incompressible Navier-Stokes equations with boundaries. *SIAM J. Numer. Anal.*, 30(3):609–629, 1993.

- [73] R.J.A. Howard and J.C.F Pereira. A study of wind turbine power generation and turbine-tower interaction using large eddy simulation. *Wind and Structures*, 9(2):95–108, 2006.
- [74] W. Hundsdorfer and J.G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer, 2003.
- [75] J. Hyman, M. Shashkov, and S. Steinberg. The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials. *J. Comput. Phys.*, 132:130–148, 1997.
- [76] S. Ivanell. *Numerical computation of wind turbine wakes*. PhD thesis, KTH Mechanics, Royal Institute of Technology, 2009.
- [77] S. Ivanell, J.N. Sørensen, R. Mikkelsen, and D. Henningson. Analysis of numerically generated wake structures. *Wind Energ.*, 12:63–80, 2009.
- [78] L. Jay. Symplectic partitioned Runge-Kutta methods for constrained Hamiltonian systems. *SIAM J. Numer. Anal.*, 33:368–387, 1996.
- [79] L.O. Jay. Specialized Runge-Kutta methods for index 2 differential-algebraic equations. *Math. Comput.*, 75:641–654, 2005.
- [80] A. Jimenez, A. Crespo, and E. Migoya. Application of a LES technique to characterize the wake deflection of a wind turbine in yaw. *Wind Energ.*, 2009.
- [81] A. Jimenez, A. Crespo, E. Migoya, and J. Garcia. Advances in large-eddy simulation of a wind turbine wake. In *The science of making torque from wind. Conference series*, volume 75, 2007.
- [82] A. Jimenez, A. Crespo, E. Migoya, and J. Garcia. Large-eddy simulation of spectral coherence in a wind turbine wake. *Environ. Res. Lett.*, 3, 2008.
- [83] J. Johansen and N.N. Sørensen. Aerofoil characteristics from 3D CFD rotor computations. *Wind Energ.*, 7:283–294, 2004.
- [84] N.A. Kampanis and J.A. Ekaterinaris. A staggered, high-order accurate method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 215:589–613, 2006.
- [85] C.A. Kennedy and M.H. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. Technical Report TM-2001-211038, NASA, 2001.
- [86] J. Kim and P. Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comput. Phys.*, 59:308–323, 1985.
- [87] R. Knikker. Study of a staggered fourth-order compact scheme for unsteady incompressible viscous flow. *Int. J. Numer. Meth. Fluids*, 59:1063–1092, 2009.



- [88] H.-O. Kreiss and G. Scherer. Finite element and finite difference methods for hyperbolic partial differential equations. In C. de Boor, editor, *Mathematical Aspects of Finite Elements in Partial Differential Equations*. Academic Press, New York, 1974.
- [89] G.C. Larsen et al. Dynamic wake meandering modeling. Technical Report R-1607, Risø National Laboratory, Technical University of Denmark, 2007.
- [90] G.C. Larsen, H.A. Madsen, K. Thomsen, and T.J. Larsen. Wake meandering: a pragmatic approach. *Wind Energ.*, 11:377–395, 2008.
- [91] B.E. Launder, G.J. Reece, and W. Rodi. Progress in the development of a Reynolds-stress turbulent closure. *J. Fluid Mech.*, 68(3):537–566, 1975.
- [92] J.O. Laurent. Structure preservation for constrained dynamics with super partitioned additive Runge-Kutta methods. *SIAM J. Sci. Comput.*, 20:416–446, 1998.
- [93] W.J. Layton and L.G. Rebholz. *Approximate Deconvolution Models of Turbulence*, volume 2042 of *Lecture Notes in Mathematics*. Springer, 2012.
- [94] H. Le and P. Moin. An improvement of fractional step methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 92(2):369–379, 1991.
- [95] C. Leclerc and C. Masson. Toward blade-tip vortex simulation with an actuator-lifting surface model. AIAA Paper 2004-0667, 2004.
- [96] R.J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 31:1019–1044, 1994.
- [97] Z. Li and K. Ito. *The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*. SIAM, 2006.
- [98] Z. Li and M.-C. Lai. The immersed interface method for the Navier-Stokes equations with singular forces. *J. Comput. Phys.*, 171:822–842, 2001.
- [99] H. Liu and G. Sun. Implicit Runge-Kutta methods based on Lobatto quadrature formula. *Int. J. Comput. Math.*, 82:77–88, 2005.
- [100] J. Lorenz. Zur Inversmonotonie diskreter Probleme. *Num. Math.*, 27:227–238, 1977.
- [101] H.A. Madsen. CFD analysis of the actuator disc flow compared with momentum results. In *10th IEA Meeting on Aerodynamics, University of Edinburgh*, 1996.
- [102] H.A. Madsen, C. Bak, M. Døssing, R. Mikkelsen, and S. Øye. Validation and modification of the blade element momentum theory based on comparisons with actuator disc simulations. *Wind Energ.*, 13:373–389, 2009.

- [103] K. Mahesh, G. Constantinescu, and P. Moin. A numerical method for large-eddy simulation in complex geometries. *J. Comput. Phys.*, 197:215–240, 2004.
- [104] C. Masson, A. Smaili, and C. Leclerc. Aerodynamic analysis of HAWTs operating in unsteady conditions. *Wind Energ.*, 4:1–22, 2001.
- [105] R.M.M. Mattheij, S.W. Rienstra, and J.H.M. ten Thije Boonkkamp. *Partial Differential Equations: Modeling, Analysis, Computation*. SIAM, 2005.
- [106] K. Mattson and J. Nordstrøm. Summation by parts operators for finite difference approximations of second derivatives. *J. Comput. Phys.*, 199:503–540, 2004.
- [107] J. Meyers, B.J. Geurts, and P. Sagaut. A computational error-assessment of central finite-volume discretizations in large-eddy simulation using a Smagorinsky model. *J. Comput. Phys.*, 227:156–173, 2007.
- [108] J. Meyers and C. Meneveau. Large eddy simulations of large wind-turbine arrays in the atmospheric boundary layer. In *48th AIAA Aerospace Sciences Meeting, Orlando, Florida, AIAA 2010-827*, 2010.
- [109] J.A. Michelsen. Basis3D - a platform for development of multiblock PDE solvers. Technical Report AFM 92-05, Technical University of Denmark, Lyngby, 1992.
- [110] J.A. Michelsen. Block structured multigrid solution of 2D and 3D elliptic PDEs. Technical Report AFM 94-06, Technical University of Denmark, Lyngby, 1994.
- [111] R. Mikkelsen. *Actuator disc methods applied to wind turbines*. PhD thesis, Technical University of Denmark, 2003.
- [112] R. Mikkelsen, J.N. Sørensen, and N. Troldborg. Prescribed wind shear modelling with the actuator line technique. In *EWEC, Milan*, 2007.
- [113] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, and A. von Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *J. Comput. Phys.*, 227:4825–4852, 2008.
- [114] R. Mittal and P. Moin. Suitability of upwind-biased finite difference schemes for large-eddy simulation of turbulent flows. *AIAA Journal*, 35(8), 1997.
- [115] A. Montlaur, S. Fernandez-Mendez, and A. Huerta. High-order implicit time integration for unsteady incompressible flows. *Int. J. Numer. Meth. Fluids*, 70:603–636, 2012.
- [116] Y. Morinishi, T.S. Lund, O.V. Vasilyev, and P. Moin. Fully conservative higher order finite difference schemes for incompressible flows. *J. Comput. Phys.*, 143:90–124, 1998.

- [117] K.W. Morton and D.F. Mayers. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, 2005.
- [118] R.D. Moser, J. Kim, and N.N. Mansour. Direct numerical simulation of turbulent channel flow up to  $Re_\tau=590$ . *Phys. Fluids*, 11:943–945, 1999.
- [119] F.T. Muijres, G.R. Spedding, Y. Winter, and A. Hedenström. Actuator disk model and span efficiency of flapping flight in bats based on time-resolved PIV measurements. *Exp. Fluids*, 51:511–525, 2011.
- [120] P. Mullen, K. Crane, D. Pavlov, Y. Tong, and M. Desbrun. Energy-preserving integrators for fluid animation. In *Proceedings of ACM SIGGRAPH*, volume 28, 2009.
- [121] S. Nagarajan, S.K. Lele, and J.H. Ferziger. A robust high-order compact method for large eddy simulation. *J. Comput. Phys.*, 191:392–419, 2003.
- [122] R.A. Nicolaides. The covolume approach to computing incompressible flow. In M.D. Gunzburger and R.A. Nicolaides, editors, *Incompressible Computational Fluid Dynamics. Trends and Advances*, pages 295–333. Cambridge University Press, 1993.
- [123] L. Pareschi and G. Russo. Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *J. Sci. Comput.*, 25(1):129–155, 2005.
- [124] D. Pathria. The correct formulation of intermediate boundary conditions for Runge-Kutta time integration of initial boundary value problems. *SIAM J. Sci. Comput.*, 18:1255–1266, 1997.
- [125] J.M.C. Pereira, M.H. Kobayashi, and J.C.F. Pereira. A fourth-order-accurate finite volume compact method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 167:217–243, 2001.
- [126] B. Perot. Conservation properties of unstructured staggered mesh schemes. *J. Comput. Phys.*, 159:58–89, 2000.
- [127] B. Perot and R. Nallapati. A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows. *J. Comput. Phys.*, 184:192–214, 2003.
- [128] J. B. Perot. Discrete conservation properties of unstructured mesh schemes. *Annu. Rev. Fluid Mech.*, 43:299–318, 2011.
- [129] J.B. Perot. An analysis of the fractional step method. *J. Comput. Phys.*, 180:51–58, 1993.
- [130] J.B. Perot. Comments on the fractional step method. *J. Comput. Phys.*, 121:190–191, 1995.

- [131] C.S. Peskin. Flow patterns around heart valves: a numerical method. *J. Comput. Phys.*, 10:252–271, 1972.
- [132] F. Porté-Agel, H. Lu, and Y. Wu. A large-eddy simulation framework for wind energy applications. In *Fifth International Symposium on Computational Wind Engineering, Chapel Hill, USA*, 2010.
- [133] J.M. Prospathopoulos, E.S. Politis, and P.K. Chaviaropoulos. Modelling wind turbine wakes in complex terrain. In *EWEC, Brussels*, 2008.
- [134] J.M. Prospathopoulos, E.S. Politis, P.K. Chaviaropoulos, and K.G. Rados. Enhanced CFD modeling of wind turbine wakes. In *Euromech 508 colloquium on wind turbine wakes, Madrid*, 2009.
- [135] M.-Z. Qin and M.-Q. Zhang. Symplectic Runge-Kutta algorithms for Hamiltonian systems. *J. Comput. Math.*, Supplementary Issue:205–215, 1992.
- [136] K.G. Rados, J.M. Prospathopoulos, N.Ch. Stefanatos, E.S. Politis, P.K. Chaviaropoulos, and A. Zervos. CFD modeling issues of wind turbine wakes under stable atmospheric conditions. In *EWEC, Marseille*, 2009.
- [137] R.G. Rajagopalan, T.L. Rickerl, and P.C. Klimas. Aerodynamic interference of vertical axis wind turbines. *J. Propulsion*, 6:645–653, 1990.
- [138] W.J. Rankine. On the mechanical principles of the action of propellers. *Transactions of the Institute of Naval Architects*, 6:13–39, 1865.
- [139] P.-E. Réthoré. *Wind turbine wake in atmospheric turbulence*. PhD thesis, Aalborg University, 2009.
- [140] P.-E. Réthoré and N.N. Sørensen. A discrete force allocation algorithm for modelling wind turbines in computational fluid dynamics. *Wind Energ.*, 15:915–926, 2011. DOI:10.1002/we.525.
- [141] P.-E. Réthoré, P. van der Laan, N. Troldborg, F. Zahle, and N.N. Sørensen. Verification and validation of an actuator disc model. *Wind Energ.*, 2013.
- [142] R. Sadourny. The dynamics of finite-difference models of the shallow-water equations. *J. Atmos. Sci.*, 32:680–689, 1975.
- [143] P. Sagaut. *Large Eddy Simulation for Incompressible Flows*. Springer, 2002.
- [144] B. Sanderse. Aerodynamics of wind turbine wakes. Technical Report ECN-E-09-016, Energy research Centre of the Netherlands, 2009.
- [145] B. Sanderse. ECNS: Energy-Conserving Navier-Stokes Solver. Verification of steady laminar flows. Technical Report ECN-E-11-042, Energy research Centre of the Netherlands, 2011.

- [146] B. Sanderse. Energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 233:100–131, 2013.
- [147] B. Sanderse. Immersed interface methods for finite volume discretizations. *Submitted to J. Comput. Phys.*, 2013.
- [148] B. Sanderse and B. Koren. Analysis of fourth-order accurate symmetry-preserving boundary conditions for the incompressible Navier-Stokes equations. In J.C.F. Pereira, A. Sequeira, and J.M.C. Pereira, editors, *Proceedings of the Fifth European Conference on Computational Fluid Dynamics (ECCOMAS CFD)*, 2010.
- [149] B. Sanderse and B. Koren. Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 231:3041–3063, 2012.
- [150] B. Sanderse and B. Koren. Immersed actuator methods for wind-turbine wake calculations. In *The science of making torque from wind. Journal of Physics: Conference series*, 2012.
- [151] B. Sanderse and B. Koren. Linearly implicit energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)*, Vienna, Austria, September 2012.
- [152] B. Sanderse and B. Koren. New explicit Runge-Kutta methods for the incompressible Navier-Stokes equations. In *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, Big Island, Hawaii, number ICCFD7-1403, July 2012.
- [153] B. Sanderse, M. Nool, S.W. van Haren, and B. Koren. ECNS: a new dedicated Navier-Stokes solver for wind-turbine wake simulations. In *The science of making torque from wind. Journal of Physics: Conference series*, 2012.
- [154] B. Sanderse, S.P. van der Pijl, and B. Koren. Review of computational fluid dynamics for wind turbine wake aerodynamics. *Wind Energ.*, 14:799–819, 2011.
- [155] B. Sanderse, R.W.C.P. Verstappen, and B. Koren. Boundary treatment for symmetry-preserving discretizations of the incompressible Navier-Stokes equations. *Submitted to J. Comput. Phys.*, 2012.
- [156] R.L. Sani, J. Shen, O. Pironneau, and P.M. Gresho. Pressure boundary condition for the time-dependent incompressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 50:673–682, 2006.
- [157] F.G. Schmitt. About Boussinesq’s turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity. *Comptes Rendus Mécanique*, 335:617–627, 2007.

- [158] U. Schumann. Realizability of Reynolds-stress turbulence models. *Phys. Fluids*, 20(5):721–725, 1977.
- [159] M. Shashkov. *Conservative Finite-Difference Methods on General Grids*. CRC Press, 1996.
- [160] W.Z. Shen, M.O.L. Hansen, and J.N. Sørensen. Determination of the angle of attack on rotor blades. *Wind Energ.*, 12:91–98, 2009.
- [161] W.Z. Shen, R. Mikkelsen, and J.N. Sørensen. Tip loss correction for actuator/Navier-Stokes computations. *J. Sol. Energy Eng.*, 127:209–213, 2005.
- [162] W.Z. Shen, J.N. Sørensen, and J.H. Zhang. Actuator surface model for wind turbine flow computations. In *EWEC, Milan*, 2007.
- [163] W.Z. Shen, J.H. Zhang, and J.N. Sørensen. The actuator-surface model: a new Navier-Stokes based model for rotor computations. *J. Sol. Energy Eng.*, 131, 2009.
- [164] T.-H. Shih, W.W. Liou, A. Shabbir, Z. Yang, and J. Zhu. A new  $k-\epsilon$  eddy viscosity model for high Reynolds number turbulent flows. *Comp. Fluids*, 24:227–238, 1995.
- [165] C.-W. Shu. Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory schemes for hyperbolic conservation laws. Technical Report NASA/CR-97-206253, ICASE Report no. 97-65, NASA Langley Research Center, 1997.
- [166] C. Sibuet Watters and C. Masson. Recent advances in modeling of wind turbine wake vortical structure using a differential actuator disk theory. In *The science of making torque from wind. Conference series*, volume 75, 2007.
- [167] C. Sibuet Watters and C. Masson. Modeling of lifting-device aerodynamics using the actuator surface concept. *Int. J. Numer. Meth. Fluids*, 2009.
- [168] J.C. Simo and F. Armero. Unconditional stability and long-term behavior of transient algorithms for the incompressible Navier-Stokes and Euler equations. *Comput. Methods Appl. Mech. Engrg.*, 111:111–154, 1994.
- [169] J. Smagorinsky. General circulation experiments with the primitive equations. I. The basic experiment. *Mon. Weather Rev.*, 91:99–164, 1963.
- [170] H. Snel. Review of the present status of rotor aerodynamics. *Wind Energ.*, 1:46–69, 1998.
- [171] H. Snel. Review of aerodynamics for wind turbines. *Wind Energ.*, 6:203–211, 2003.
- [172] A. Sogachev and O. Panferov. Modification of two-equation models to account for plant drag. *Boundary-Layer Meteorol.*, 121:229–266, 2006.

- [173] A. Sogachev and O. Panferov. A note on two-equation closure modelling of canopy flow. *Boundary-Layer Meteorol.*, 130:423–435, 2009.
- [174] J.N. Sørensen and C.W. Kock. A model for unsteady rotor aerodynamics. *J. Wind Eng. and Ind. Aerodyn.*, 58:259–275, 1995.
- [175] J.N. Sørensen and A. Myken. Unsteady actuator disc model for horizontal axis wind turbines. *J. Wind Eng. and Ind. Aerodyn.*, 39:139–149, 1992.
- [176] J.N. Sørensen and W.Z. Shen. Numerical modeling of wind turbine wakes. *J. Fluid. Eng.*, 124:393–399, 2002.
- [177] J.N. Sørensen, W.Z. Shen, and X. Munduate. Analysis of wake states by a full-field actuator disc model. *Wind Energ.*, 1:73–88, 1998.
- [178] N.N. Sørensen. General purpose flow solver applied to flow over hills. Technical Report Risø-R-827(EN), Risø National Laboratory, Roskilde, 1995.
- [179] P.R. Spalart. Strategies for turbulence modelling and simulations. *Int. J. Heat Fluid Fl.*, 21:252–263, 2000.
- [180] P.R. Spalart, W.H. Jou, M. Strelets, and S.R. Allmaras. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. In *Advances in DNS/LES, 1st AFOSR Int. Conf. on DNS/LES*, 1997.
- [181] P.R. Spalart, R.D. Moser, and M.M. Rogers. Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions. *J. Comput. Phys.*, 96(2):297–324, 1991.
- [182] R. Stoll and F. Porté-Agel. Dynamic subgrid-scale models for momentum and scalar fluxes in large-eddy simulations of neutrally stratified atmospheric boundary layers over heterogeneous terrain. *Water Resources Research*, 42, 2006.
- [183] T. Stovall, G. Pawlas, and P. Moriarty. Wind farm wake simulations in OpenFOAM. In *48th AIAA Aerospace Sciences Meeting, Orlando, Florida, AIAA 2010-825*, 2010.
- [184] B. Strand. Summation by parts for finite difference approximations for  $d/dx$ . *J. Comput. Phys.*, 110:47–67, 1994.
- [185] R.B. Stull. *An Introduction to Boundary Layer Meteorology*. Kluwer Academic Publishers, 1988.
- [186] V. Subramanian and J.B. Perot. Higher-order mimetic methods for unstructured meshes. *J. Comput. Phys.*, 219:68–85, 2006.
- [187] Geng Sun. Construction of high order symplectic Runge-Kutta methods. *J. Comput. Math.*, 11(3):250–260, 1993.

- [188] Geng Sun. A simple way constructing symplectic Runge-Kutta methods. *J. Comput. Math.*, 18(1):61–68, 2000.
- [189] D. Tafti. Comparison of some upwind-biased high-order formulations with a second-order central difference scheme for time integration of the incompressible Navier-Stokes equations. *Comp. Fluids*, 25(7):647–665, 1996.
- [190] N. Troldborg. *Actuator line modeling of wind turbine wakes*. PhD thesis, Technical University of Denmark, 2008.
- [191] N. Troldborg, J.N. Sørensen, and R. Mikkelsen. Actuator line simulation of wake of wind turbine operating in turbulent inflow. In *The science of making torque from wind. Conference series*, volume 75, 2007.
- [192] N. Troldborg, J.N. Sørensen, and R. Mikkelsen. Numerical simulations of wake characteristics of a wind turbine in uniform inflow. *Wind Energ.*, 2009.
- [193] S. Turek. A comparative study of some time-stepping techniques for the incompressible Navier-Stokes equations. from fully implicit nonlinear schemes to semi-implicit projection methods. *Int. J. Numer. Meth. Fluids*, 20:987–1011, 1996.
- [194] J. van Kan. A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM J. Sci. Stat. Comput.*, 7(3):870–891, 1986.
- [195] J.J.A.M. van Os and R.E. Uittenbogaard. Towards the ultimate variance-conserving convection scheme. *J. Comput. Phys.*, 197:197–214, 2004.
- [196] O.V. Vasilyev. High order finite difference schemes on non-uniform meshes with good conservation properties. *J. Comput. Phys.*, 157:746–761, 2000.
- [197] A.E.P. Veldman. "Missing" boundary conditions? Discretize first, substitute next, and combine later. *SIAM J. Sci. Stat. Comput.*, 11:82–91, 1990.
- [198] A.E.P. Veldman. High-order symmetry-preserving discretization of convection-diffusion equations on strongly stretched grids. In G. Lube and G. Rapin, editors, *Proceedings of the Int. Conf. on Boundary and Interior Layers*, 2006.
- [199] L.J. Vermeer, J.N. Sørensen, and A. Crespo. Wind turbine wake aerodynamics. *Prog. Aerospace Sci.*, 39:467–510, 2003.
- [200] R. Verstappen, J.G. Wissink, and A.E.P. Veldman. Direct numerical simulation of driven cavity flows. *Appl. Sci. Res.*, 51:377–381, 1993.
- [201] R.W.C.P. Verstappen. When does eddy viscosity damp subfilter scales sufficiently? *J. Sci. Comput.*, 49:94–110, 2011.



- [202] R.W.C.P. Verstappen and A.E.P. Veldman. Spectro-consistent discretisation of Navier-Stokes: a challenge to RANS and LES. *J. Engng. Math.*, 34:163–179, 1998.
- [203] R.W.C.P. Verstappen and A.E.P. Veldman. Symmetry-preserving discretization of turbulent flow. *J. Comput. Phys.*, 187:343–368, 2003.
- [204] B. Vreman. *Direct and large-eddy simulation of the compressible turbulent mixing layer*. PhD thesis, University of Twente, 1993.
- [205] M. Vysohlid. *Large Eddy Simulation of Crashback in Marine Propellers*. PhD thesis, University of Minnesota, 2007.
- [206] G. Wanner. Runge-Kutta methods with expansion in even powers of  $h$ . *Computing*, 11:81–85, 1973.
- [207] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer, 2001.
- [208] P. Wesseling, A. Segal, and C.G.M. Kassels. Computing flows on general three-dimensional nonsmooth staggered grids. *J. Comput. Phys.*, 149:333–362, 1999.
- [209] J. Whale, C.G. Anderson, R. Bareiss, and S. Wagner. An experimental and numerical study of the vortex structure in the wake of a wind turbine. *J. Wind Eng. and Ind. Aerodyn.*, 84:1–21, 2000.
- [210] D.C. Wilcox. *Turbulence Modeling in CFD*. DCW, 2006.
- [211] Y. Wu and F. Porté-Agel. Large-eddy simulation of wind-turbine wakes: Evaluation of turbine parameterizations. *Boundary-Layer Meteorol.*, 138:345–366, 2011.
- [212] J.C. Wyngaard. Atmospheric turbulence. *Annu. Rev. Fluid Mech.*, 24:205–233, 1992.
- [213] S. Xu and Z. Wang. Systematic derivation of jump conditions for the immersed interface method in three-dimensional flow simulation. *SIAM J. Sci. Comput.*, 27(6):1948–1980, 2006.
- [214] F. Zahle, N.N. Sørensen, and J. Johansen. Wind turbine rotor-tower interaction using an incompressible overset grid method. *Wind Energ.*, 12:594–619, 2009.
- [215] X. Zhang, D. Schmidt, and B. Perot. Accuracy and conservation properties of a three-dimensional unstructured staggered mesh scheme for fluid dynamics. *J. Comput. Phys.*, 175(2):764–791, 2002.



## INDEX

---

- actuator models
  - disk, 12, 106, 176, 187, 193
  - line, 15, 187
  - surface, 15, 187
  - wing, 192
- amplification factor, *see* stability function
- barrier function, 60
- boundary conditions, 37, 53, 57
- Boussinesq hypothesis, 5, 8
- convection-diffusion equation, 56, 62
- corner flow, 142
- differential-algebraic equations, 77
- discontinuities, 155, 169, 171
- discrete Dirac function, 159, 170
- DNS, 4, 21, 72
- energy, *see* kinetic energy
- generalized Taylor expansion, 155, 211
- immersed interface method, 154
- interface
  - rotating, 184
  - translating, 181
- jump relation, *see* discontinuities
- kinetic energy equation
  - continuous, 21, 25
  - discrete, 116
  - semi-discrete, 47
- LES, 7, 8, 21, 72
  - Smagorinsky model, 8
- lid-driven cavity, 67, 70
- membrane, 174
- monotone, 59, 166
- Navier-Stokes equations, 4, 22, 169
- numerical diffusion, 11, 21
- numerical dissipation, *see* numerical diffusion
- parallelization, 191
- PETSc, 191
- Poisson equation, *see* pressure
- positive, 60
- pressure, 54, 78, 96, 127, 203
- projection operator, 78
- quadrature, 114
  - Gauss, 114, 118
  - Lobatto, 114, 118, 119, 125
  - Radau, 114, 118, 123
- RANS, 5, 6
  - $k - \epsilon$ , 6
  - $k - \omega$ , 6
  - RSM, 7
- reconstruction, 99
- regularization, 14, 153
- Richardson extrapolation, 30
- rotor modeling, 11
  - direct, 16
  - generalized actuator, 11
- Runge-Kutta methods, 79
  - additive, 121
  - Butcher tableau, 80, 85
  - DIRK, 114, 117, 125, 214
  - explicit, 85
  - half-explicit, 89
  - implicit, 113
  - order conditions, 86, 115, 122, 127
  - partial, 217
  - simplifying conditions, 80, 115

- semi-discrete equations, 28
- shear layer roll-up, 33, 136
- source term
  - moving, 162, 178
  - stationary, 155, 169
- space-time error, 91
- stability, 166
  - $A$ -, 119, 216
  - algebraic, 117, 119, 120, 122, 214, 215
  - $L$ -, 120
- stability function, 120, 123, 216
- staggered grid, 27, 28
- stiffly accurate, 114, 120
- stiffness, 115, 142
- surface force, 169
- symmetric, *see* time-reversible
- symmetry properties
  - convection, 24, 53
  - diffusion, 24, 53
  - div-grad, 24, 53
- symplectic, 116, 214
  
- Taylor-Green vortex, 32, 70, 104, 134
- time-reversible, 118, 214
- trees, 87
- truncation error
  - global, 58, 59
  - local, 58, 156
- turbulence, 1, 2, 4
- turbulent channel flow, 72
  
- $W$ -transformation, 117, 213
- wake models, 3
- wind farm, 1, 194

## SUMMARY

---

### Energy-conserving discretization methods for the incompressible Navier-Stokes equations. Application to the simulation of wind-turbine wakes.

In this thesis numerical methods are developed for the simulation of turbulent flows governed by the incompressible Navier-Stokes equations. This is inspired by the need for accurate and efficient computations of the flow of air in wind-turbine wakes. The state-of-the-art in computing such flows is to use Large Eddy Simulation (LES) as a turbulence model. In LES the Navier-Stokes equations are filtered such that only the large, energy-containing scales of motion are simulated - the smaller scales are modeled. However, even with such a model, LES simulations remain expensive (not only in wind energy applications) and are typically 'under-resolved': the mesh is too coarse to resolve all important scales. Thus, an ongoing challenge is to construct numerical methods that are stable and accurate even on coarse meshes, and do not introduce false ('artificial') diffusion that can destroy the delicate features of turbulent flows.

The approach taken is to construct high-order *energy-conserving* discretization methods. Such methods mimic an important property of the continuous incompressible Navier-Stokes equations, namely the conservation of kinetic energy in the limit of vanishing viscosity. The energy equation is, for incompressible flows, derived from the equations for conservation of mass and momentum. An energy-conserving discretization method is nonlinearly stable, independent of mesh, time step, or viscosity, and does not introduce artificial diffusion.

The first part of this thesis addresses *spatially* energy-conserving discretization methods, in particular second and fourth order finite volume methods on staggered cartesian grids. Special attention is paid to the proper treatment of boundary conditions for high order methods. New boundary conditions are derived such that the boundary contributions to the discrete energy equation mimic the boundary contributions of the continuous equations. An important theoretical result is obtained: higher order energy-conserving finite volume discretizations are limited to second order global accuracy in the presence of boundaries. On properly chosen non-uniform grids, designed such that the maximum error is not at the boundary, fourth order accuracy can be recovered.

The second part of this thesis addresses time integration of the incompressible Navier-Stokes equations with Runge-Kutta methods. Runge-Kutta methods are often applied to the spatially discretized incompressible Navier-Stokes equations, but order of accuracy proofs that address both velocity and pressure are missing. By viewing the spatially discretized Navier-Stokes equations as a system of differential-algebraic equations the order conditions for velocity and pressure are derived. Based on these conditions new explicit Runge-Kutta methods are derived, that have high-order accuracy for both velocity and pressure. These explicit meth-

ods are not strictly energy-conserving but can be efficient, depending when the time step is determined by accuracy instead of stability. However, for truly energy-conserving Runge-Kutta methods implicit methods need to be considered. High-order Runge-Kutta methods based on Gauss quadrature are proposed. In particular, the two-stage fourth order Gauss method is investigated and combined with the fourth order spatial discretization, resulting in a *fourth order energy-conserving method in space and time*, which is stable for any mesh and any time step. A disadvantage of the Gauss methods is that they are less suitable for integrating the diffusive terms, since they lack  $L$ -stability. Therefore, new additive Runge-Kutta methods are investigated: the diffusive terms are integrated with an  $L$ -stable Runge-Kutta method, and the convective terms with an energy-conserving Runge-Kutta method, both based on the same quadrature points. Unfortunately, their low stage order does not make them more efficient than the original Gauss methods. In practice, the second order Gauss method (implicit midpoint) is therefore the preferred time integration method.

The third part of this thesis addresses actuator methods. Actuator methods are simplified models to represent the effect of a body (such as a wind turbine) on a flow field, without requiring the actual geometry of the body to be taken into account. Actuator forces introduce discontinuities in flow variables and should therefore be treated carefully. A new immersed interface method in finite volume formulation is proposed, which leads to a sharp, non-diffusive, representation of the actuator. This does not require the choice for a discrete Dirac function and regularization parameter.

The ideas put forth in this thesis have been implemented in a new parallel 3D incompressible Navier-Stokes solver: ECNS (Energy-Conserving Navier-Stokes solver). The resulting method combines stability, no numerical viscosity and high-order accuracy. This makes it a valuable tool for simulating turbulent flow problems governed by the incompressible Navier-Stokes equations and suitable for the development and comparison of LES models. For the particular case of wind-turbine wake aerodynamics a number of simulations have been performed: flow over a wing as a model for a wind turbine blade, and flow through an array of actuator disks representing a wind farm.

### Energiebehoudende discretisatiemethoden voor de incompressibele Navier-Stokes vergelijkingen, toegepast op de simulatie van windturbinezoggen.

In dit proefschrift worden numerieke methoden ontwikkeld voor de simulatie van turbulente stromingen, beschreven door de incompressibele Navier-Stokes vergelijkingen. Zulke methoden kunnen bijvoorbeeld gebruikt worden om de stroming van lucht in windturbineparken uit te rekenen. De 'state-of-the-art' voor het berekenen van zulke turbulente stromingen is het gebruik van 'Large Eddy Simulation' (LES) als turbulentiemodel. In de LES aanpak worden de Navier-Stokes vergelijkingen gefilterd, zodat alleen de grote, energiebehoudende schalen worden gesimuleerd - de kleinere schalen worden gemodelleerd. Echter, zelfs met een LES model blijven simulaties duur in termen van rekentijd. Veel LES simulaties worden daarom uitgevoerd met een rooster dat te grof is om alle belangrijke schalen op te lossen. Een uitdaging is om numerieke methoden te construeren die stabiel en nauwkeurig zijn op zulke grove roosters, en geen kunstmatige diffusie introduceren. Kunstmatige diffusie kan de delicate eigenschappen van turbulente stromingen verstoren.

De aanpak in dit proefschrift is het construeren van hogere-orde *energiebehoudende* discretisatiemethoden. Dergelijke methoden bezitten discreet een belangrijke eigenschap van de continue incompressibele Navier-Stokes vergelijkingen: behoud van kinetische energie in het niet-visceuze geval. Een energiebehoudende discretisatie heeft belangrijke eigenschappen: simulaties zijn stabiel, onafhankelijk van gekozen rooster, tijdstap of viscositeit, en er wordt geen kunstmatige diffusie geïntroduceerd.

Het eerste deel van dit proefschrift behandelt *ruimtelijke* energiebehoudende discretisatiemethoden, in het bijzonder tweede- en vierde-orde nauwkeurige eindigevolume methoden op 'staggered' cartesische roosters. Nieuwe randvoorwaarden voor hogere-orde methoden worden afgeleid, die zodanig zijn dat de bijdragen van de rand aan de discrete energievergelijking de bijdragen van de continue energievergelijking nabootsen. Een belangrijk theoretisch resultaat is dat hogere-orde energiebehoudende eindigevolume discretisaties beperkt zijn tot tweede-orde nauwkeurigheid door de aanwezigheid van randen. Als een niet-uniform rooster echter zo wordt gekozen dat de maximale fout niet bij de rand ligt, is het mogelijk om toch vierde-orde nauwkeurigheid te behouden.

Het tweede deel van dit proefschrift behandelt de tijdsintegratie van de incompressibele Navier-Stokes vergelijkingen met Runge-Kutta methoden. Runge-Kutta methoden worden vaak gebruikt voor tijdsintegratie van de ruimtelijk gediscrètiseerde incompressibele Navier-Stokes vergelijkingen. De ordecondities voor zowel de snelheid als de druk worden afgeleid door de ruimtelijke gediscrètiseerde vergelijkingen te zien als een stelsel van differentiaal-algebraïsche vergelijkingen. Nieuwe expliciete Runge-Kutta methoden worden voorgesteld die hogere-orde nauwkeurigheid hebben voor zowel snelheid als druk. Zulke expliciete methoden zijn niet

energiebehoudend, maar kunnen efficiënt zijn als de tijdstap bepaald wordt door nauwkeurigheid in plaats van stabiliteit. Om werkelijk energiebehoudende Runge-Kutta methoden te construeren moeten impliciete methoden worden beschouwd: methoden gebaseerd op Gauss kwadratuur. De twee-staps vierde-orde Gauss methode is onderzocht en gecombineerd met de vierde-orde ruimtelijke discretisatie, wat resulteert in een *vierde-orde energiebehoudende methode in ruimte en tijd*. De Gauss methoden zijn echter minder geschikt voor tijdsintegratie van de diffusieve termen door het ontbreken van  $L$ -stabiliteit. Als alternatief zijn daarom nieuwe additieve Runge-Kutta methoden voorgesteld: de diffusieve termen worden hierbij geïntegreerd met een  $L$ -stabiele methode, en de convectieve termen met een energiebehoudende methode, gebaseerd op dezelfde kwadratuurpunten. Helaas zorgt hun lage 'stage orde' ervoor dat deze methoden niet meer efficiënt zijn dan de originele Gauss methoden. In praktijk is de tweede-orde Gauss methode daarom de te verkiezen tijdsintegratiemethode.

Het derde deel van dit proefschrift behandelt actuator methoden. Actuator methoden representeren het effect van een lichaam (zoals een windturbineblad) op de stroming door middel van voorgeschreven krachten, zonder dat de daadwerkelijke geometrie hoeft te worden gemodelleerd. Het eenvoudigste windturbinemodel is gegeven door een trekkende schijf ('actuator disk'). De voorgeschreven krachten leiden tot discontinuïteiten in de stromingsvariabelen en moeten daarom zorgvuldig behandeld worden. Een nieuwe 'immersed interface' methode in eindige-volume formulering is voorgesteld, die leidt tot een scherpe, niet-diffusieve representatie van de interface. In tegenstelling tot bestaande methoden is hierbij geen discrete Dirac functie of regularisatieparameter vereist.

De ideeën voorgesteld in dit proefschrift zijn geïmplementeerd in een nieuwe parallelle 3D incompressibele Navier-Stokes code: ECNS (Energy-Conserving Navier-Stokes solver). Deze code combineert stabiliteit, vrijwaring van numerieke diffusie, en hogere-orde nauwkeurigheid in ruimte en tijd. De code is een waardevol gereedschap voor het simuleren van turbulente stromingsproblemen beschreven door de incompressibele Navier-Stokes vergelijkingen, en is in het bijzonder geschikt voor het ontwikkelen en vergelijken van LES modellen. Een aantal simulaties relevant voor windturbinezoggen zijn uitgevoerd: de stroming over een vleugel als een model voor een windturbineblad, en de stroming door een park met turbines gemodelleerd als trekkende schijven.



## LIST OF PUBLICATIONS

---

### Journal articles:

- B. Sanderse. Immersed interface methods for finite volume discretizations. *Submitted to J. Comput. Phys.*, 2013
- B. Sanderse, R.W.C.P. Verstappen, and B. Koren. Boundary treatment for symmetry-preserving discretizations of the incompressible Navier-Stokes equations. *Submitted to J. Comput. Phys.*, 2012
- B. Sanderse. Energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 233:100–131, 2013
- B. Sanderse and B. Koren. Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 231:3041–3063, 2012
- B. Sanderse, S.P. van der Pijl, and B. Koren. Review of computational fluid dynamics for wind turbine wake aerodynamics. *Wind Energ.*, 14:799–819, 2011

### Conference proceedings:

- B. Sanderse and B. Koren. Immersed actuator methods for wind-turbine wake calculations. In *The science of making torque from wind. Journal of Physics: Conference series*, 2012
- B. Sanderse, M. Nool, S.W. van Haren, and B. Koren. ECNS: a new dedicated Navier-Stokes solver for wind-turbine wake simulations. In *The science of making torque from wind. Journal of Physics: Conference series*, 2012
- B. Sanderse and B. Koren. Linearly implicit energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS), Vienna, Austria*, September 2012
- B. Sanderse and B. Koren. New explicit Runge-Kutta methods for the incompressible Navier-Stokes equations. In *Seventh International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii*, number ICCFD7-1403, July 2012
- B. Sanderse and B. Koren. Analysis of fourth-order accurate symmetry-preserving boundary conditions for the incompressible Navier-Stokes equations. In J.C.F. Pereira, A. Sequeira, and J.M.C. Pereira, editors, *Proceedings of the Fifth European Conference on Computational Fluid Dynamics (ECCOMAS CFD)*, 2010

## Technical reports:

- B. Sanderse. ECNS: Energy-Conserving Navier-Stokes Solver. Verification of steady laminar flows. Technical Report ECN-E-11-042, Energy research Centre of the Netherlands, 2011
- B. Sanderse. Aerodynamics of wind turbine wakes. Technical Report ECN-E-09-016, Energy research Centre of the Netherlands, 2009

## CURRICULUM VITAE

---

Benjamin Sanderse was born in The Hague on October 15, 1984. He attended the Christelijke Scholengemeenschap Walcheren (Middelburg) from 1996 until 2002, where he completed a gymnasium education. From 2002 he studied Aerospace Engineering at Delft University of Technology, with a Master in Aerodynamics. He obtained both his Bachelor and Master of Science with honors, the latter in 2008 on a thesis titled 'Cartesian grid methods for preliminary aircraft design', under supervision of prof. dr. ir. B. Koren. In addition he completed an honours track program.

From 2008 to 2012 he performed PhD research under supervision of prof. Koren on the numerical simulation of wind-turbine wake aerodynamics, a combined position at the Energy research Centre of the Netherlands (ECN, Petten) and the Centrum Wiskunde & Informatica (CWI, Amsterdam). During his PhD research he stayed two months at the National Renewable Energy Laboratory (NREL) in Boulder, Colorado.

