Title

New numerical methods for Computational Fluid Dynamics, Forecast and Control

Author

Cavaglieri, Daniele

Publication Date

2016

UNIVERSITY OF CALIFORNIA, SAN DIEGO

New numerical methods for Computational Fluid Dynamics, Forecast and Control

A Dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Daniele Cavaglieri

Committee in charge:

Professor Thomas Bewley, Chair
Professor Robert Bitmead
Professor Juan Carlos Del Alamo
Professor Philip Gill
Professor Michael Holst

2016

The Dissertation of Daniele Cavaglieri is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

____

____

____

____

____

Chair

University of California, San Diego

2016

DEDICATION

To my family, in the broadest sense possible.

EPIGRAPH

You have to be burning with an idea, or a problem, or a wrong that you want to right.

If you're not passionate enough from the start, you'll never stick it out.

—Steve Jobs

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First, I would like to thank my advisor Prof. Thomas Bewley for the continuous support during my PhD studies and research. His enthusiasm, motivation, and immense knowledge have helped in innumerable ways. His guidance has contributed to my personal growth, as a person and as a professional. I could not have imagined a better advisor and mentor during the development of my research. For him, I have nothing but the utmost respect and gratitude.

I would also like to thank the rest of my thesis committee: Prof. Robert Bitmead, Prof. Juan Carlos Del Alamo, Prof. Philip Gill, and Prof. Michael Holst, for their time and feedback. My sincere thanks also goes to Mirko Previsic, who gave me the opportunity to join his team as an intern. While I was working at his company, I had a pleasant time, and that experience has helped to improve my technical skills.

I thank my friends Andrés Cortés, Eduardo Ramírez, Pedro Franco, Aman Khosa, Gianluca Meneghello, Robert Moroto, Pooriya Beyhaghi, Shahrouz Alimohammadi, Ali Mashayek, Lorenzo Ferrari, David Mateos, Lukas Nonnenmacher, Ashish Cherukuri, Shuxia Tang, and Anantha Karthikeyan, for the academic discussions, BBQs, parties, dinners, hikes, and coffee times. With you guys, I've always felt at home. A special thanks also goes to Marina Maffezzoli, my friend overseas, who constantly reminded me during these years how beautiful our country is. This is something one should never forget.

I would like to thank my family: my parents and my brothers, for supporting me not only during these years, but also in every moment of my life. Without their endless love and constant support, none of this would have been possible. I would also like to express my gratitude to my extended family, who has aided me and encouraged me throughout this endeavor.

A special thanks goes to Minyi Ji, my girlfriend and the love of my life. Her endless support and encouragement have accompanied me in every moment during this long, beautiful journey. She helped me in more ways that I can imagine, and I am grateful for every second I spent with her. For this, and much more, she deserves my endless gratitude.

Finally, I would like to thank running, gym, and yogurt, for helping me lose all the extra weight gained during this time. However, I should probably acknowledge myself, since I am the one who put all the effort, after all. Good job, Daniele.

This thesis contains parts of the following publications with Prof. Thomas Bewley, Dr Ali Mashayek, Dr Anantha Karthikeyan, and Mirko Previsic as coauthors:

- Chapter 2: D. Cavaglieri, T.R. Bewley, “Low-storage implicit/explicit Runge-Kutta schemes for the simulation of stiff high-dimensional ODE systems", *Journal of Computational Physics*, (286) 172-193, 2015

- Chapter 3: D. Cavaglieri, T.R. Bewley, “Low-storage implicit/explicit Runge-Kutta schemes for the simulation of the Navier-Stokes Equations, Part 1: Theory", *Submitted to Journal of Computational Physics*, 2016

- Chapter 4: D. Cavaglieri, A. Mashayek, T.R. Bewley, “Tweed and box relaxation: improved smoothing algorithms for multigrid solution of PDEs on stretched structured grids", *Submitted to Journal of Computational Physics*, 2016

- Chapter 5: D. Cavaglieri, T.R. Bewley, “Extensions of Thomas algorithm for the efficient solution of PDEs on wireframe structures", *Submitted to Journal of Computational Physics*, 2016

- Chapter 6: D. Cavaglieri, T.R. Bewley, M. Previsic, “Short-term ensemble ocean wave forecasting", *Submitted to IEEE Transactions on Sustainable Energy*, 2016

- Chapter 7: D. Cavaglieri, T.R. Bewley, A. Karthikeyan, M. Previsic, “Nonlinear Model Predictive Control of a point absorber wave energy converter", *Submitted to IEEE Transactions on Sustainable Energy*, 2016

VITA

2007	B. S. in Mechanical Engineering *Summa cum laude*, Politecnico di Milano, Italy
2010	M. S. in Mechanical Engineering *Summa cum laude*, Politecnico di Milano, Italy
2016	Ph. D. in Engineering Sciences (Mechanical Engineering), University of California, San Diego

PUBLICATIONS

D. Cavaglieri, T.R. Bewley, “Low-storage implicit/explicit Runge-Kutta schemes for the simulation of the Navier-Stokes Equations, Part 1: Theory", *Submitted to Journal of Computational Physics*, 2016

D. Cavaglieri, T.R. Bewley, A. Karthikeyan, M. Previsic, “Nonlinear Model Predictive Control of a point absorber wave energy converter", *Submitted to IEEE Transactions on Sustainable Energy*, 2015

D. Cavaglieri, T.R. Bewley, M. Previsic, “Short-term ensemble ocean wave forecasting", *Submitted to IEEE Transactions on Sustainable Energy*, 2015

D. Cavaglieri, T.R. Bewley, “Extensions of Thomas algorithm for the efficient solution of PDEs on wireframe structures", *Submitted to Journal of Computational Physics*, 2015

D. Cavaglieri, A. Mashayek, T.R. Bewley, “Tweed and box relaxation: improved smoothing algorithms for multigrid solution of PDEs on stretched structured grids", *Submitted to Journal of Computational Physics*, 2015

P. Beyhaghi, D. Cavaglieri, T.R. Bewley, “Delaunay-based Derivative-free Optimization via Global Surrogates, Part I: Linear constraints", *Journal of Global Optimization*, 1-52, 2015

D. Cavaglieri, T.R. Bewley, M. Previsic, “Model Predictive Control leveraging Ensemble Kalman Forecasting for optimal power take-off in wave energy conversion systems", *Proceedings of the American Control Conference*, 2015

D. Cavaglieri, T.R. Bewley, “Low-storage implicit/explicit Runge-Kutta schemes for the simulation of stiff high-dimensional ODE systems", *Journal of Computational Physics*, (286) 172-193, 2015

P. Beyhaghi, D. Cavaglieri, T.R. Bewley, “Delaunay-based Derivative-free Optimization via Global Surrogate, Part 1: Theory", *21st AIAA Computational Fluid Dynamics Conference*, 2013

D. Cavaglieri, P. Beyhaghi, T.R. Bewley, “Low-storage IMEX Runge-Kutta schemes for the simulation of Navier-Stokes equations", *21st AIAA Computational Fluid Dynamics Conference*, 2013

F. Cheli, G. Audisio, M. Brusarosco, F. Mancosu, D. Cavaglieri, S. Melzi, “Cyber Tyre: A Novel Sensor to Improve Vehicle’s Safety", *SAE World Congress & Exhibition*, 2011

ABSTRACT OF THE DISSERTATION

New numerical methods for Computational Fluid Dynamics, Forecast and Control

by

Daniele Cavaglieri

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California, San Diego, 2016

Professor Thomas Bewley, Chair

The accurate space-time discretization of the partial differential equations (PDEs) governing the dynamic behavior of complex physical phenomena is a core challenge in the field of Computational Fluid Dynamics, and in the simulation of turbulence in particular. However, it appears that over the last 30 years a disproportionate amount of attention has been addressed toward the improvement of spatial discretization techniques, while temporal discretization has relied, in most cases, on old consolidated approaches. Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS) of the incompressible Navier-Stokes Equation (NSE) today often use a mixed implicit/explicit (IMEX) time integration approach developed in the mid 1980s, which combines the second-order implicit Crank-Nicolson (CN) method for the integration of the linear stiff terms and a third-order explicit low-storage Runge-Kutta-Wray (RKW3) method for the nonlinear terms. This hybrid approach, dubbed CN/RKW3, guarantees overall second-order accuracy for the time integration, while allowing an efficient storage implementation.

Our work focuses on the development of new mixed implicit/explicit time integration schemes of the Runge-Kutta type for the simulation of high-dimensional stiff ODEs, with particular attention to the simulation of the NSE. Compared with the venerable CN/RKW3 method, our numerical schemes have better accuracy, improved stability properties, and require the same or slightly increased storage.

We have also developed new relaxation schemes for the iterative solution of linear and, with some modification, nonlinear systems arising from the discretization of PDEs. These schemes prove especially advantageous when applied as the smoothing step in the multigrid solution of elliptic PDEs over stretched grids. A noteworthy application is the iterative solution of the pressure Poisson equation arising when imposing the diverge-free constraint during the simulation of the incompressible NSE using a fractional step method. Compared with the standard approach, our schemes require significantly less computation, while providing comparable converge rates.

We then discuss the implementation of an Ensemble Kalman Filter (EnKF) for the short-term prediction of ocean waves. The approach leverages one of our low-storage IMEX Runge-Kutta schemes for the highly-resolved simulation of the nonlinear equations used to describe wave propagation. We found that, using EnKF for data assimilation of wave measurement data, it is possible to perform accurate wave forecasting up to thirty seconds into the future, provided a sufficient number of ensemble members is employed.

Finally, we introduce a new direct multiple shooting algorithm for Nonlinear Model Predictive Control (NMPC). The new approach allows analytic calculation of the discretized trajectories and associated gradients, which are required when solving the nonlinear programming problem arising within the NMPC formulation. For the discretization of the trajectories, two solutions are proposed: one based on a Runge-Kutta discretization of the continuous-time model, and one leveraging a nonlinear discrete-time model based on Taylor-Lie algebra. This algorithm is then applied to the optimization of the power take-off of a point-absorber wave energy converter (WEC). Results have shown that NMPC improves the WEC power take-off with respect to linear MPC, since the nonlinear viscous forces affecting WEC dynamics are better accounted for. Moreover, the nonlinear formulation also allows the investigation of more complex configurations, such as one-way power flow.

Chapter 1

Introduction

We can define Computational Fluid Dynamics as the art of investigating the behavior of complex phenomena involving fluid flows, such as gas and liquids, using a combination of applied mathematics, physical modeling, and numerical analysis. Most of these phenomena take place in everyday life, such as meteorological events, convective heat transfer, chemical reactions, combustion, fluid-structure interaction, human body processes and so on. While providing scientists and engineers with a deeper understanding of the world that surrounds us, Computational Fluid Dynamics has represented, and still represents, an invaluable driver of progress in the development of new numerical schemes for the analysis and simulation of fluid flows.

The goal of this thesis is to present new numerical methods for the accurate and efficient simulation of fluid phenomena, with a particular emphasis on the simulation of turbulence and free-surface flows. All the schemes presented in this work offer a certain degree of improvement with respect to other existing methods which are widely adopted in the literature and often regarded as the state of the art for the discretization of particular classes of problems. In this framework, improvement is considered in a broad sense: it may signify reduced computational time, reduced storage requirement, or improved numerical properties, such as stability, accuracy, etc. Of course, this is achieved while ensuring that the other properties of the numerical schemes do not deteriorate significantly with respect to the standard approach.

Our work focuses in particular on two main fields of numerical methods: time discretization schemes for the simulation of partial differential equations and iterative methods for the solution of large linear and nonlinear systems. More specifically, Chapter 2 presents new high-order time discretization schemes for the integration of systems of partial differential equations with a separable right-hand side in which one term is stiff but easy to compute, e.g. linear, while the other is nonstiff and usually nonlinear. This is the case of many physical models of complex fluid flows arising in Computational Fluid Dynamics. Compared to the state of the art, our schemes present the same order of accuracy with comparable computational cost, but significantly reduced storage requirement and improved stability properties.

Chapter 3 focuses on the improvement of a particular family of time integration schemes of the Runge-Kutta type which is generally applied to the simulation of high-dimensional systems of partial differential equations. Compared to the other schemes of this kind available in literature, our methods offer improved stability and accuracy with slightly increased computational cost and same or slightly higher storage requirement.

In Chapter 4, we introduce two new relaxation schemes, denoted as *tweed* and *box* relaxation, for the iterative solution of linear and, with some modification, nonlinear systems using multigrid. These schemes show their maximal efficiency when employed for the solution of elliptic systems defined over stretched structured grids. Compared to the standard approach, which instead employs alternating-direction zebra for the relaxation step in the multigrid formulation, our schemes offer comparable convergence rate, with significantly reduced computational time. This is achieved through the implementation of *ad hoc* modifications of the Thomas algorithm for the solution of tridiagonal systems. This allows to efficiently exploit the sparsity in the structure of the linear system arising within the relaxation step. Chapter 5 describes these extensions and the details of their numerical implementation.

In Chapter 6, one of our low-storage high-order Runge-Kutta schemes is leveraged for the time integration of a nonlinear pseudospectral model for ocean wave propagation. Such model is employed in an Ensemble Kalman Filter forecasting algorithm in order to provide accurate short-time wave prediction. The forecasting performance of such framework is then assessed against synthetic data emulating wave radar and monitoring buoys measurements. Results show that accurate wave forecasting is possible up to thirty seconds ahead, provided a large number of ensemble members is employed.

Finally, in Chapter 7, we introduce a new method for the calculation of the discretized trajectories and associated gradients within the Nonlinear Model Predictive Control formulation leveraging direct multiple shooting. In particular, two approaches have been followed: one based on a Runge-Kutta discretization of the continuous-time model, and the other based on a nonlinear discrete-time model derived using Taylor-Lie algebra. These methods guarantee faster convergence rate in the iterative solution of the nonlinear programming problem using a sequential quadratic programming algorithm. Following this formulation, Nonlinear Model Predictive Control is then applied to the optimization of the power take-off of a point absorber wave energy converter. Results show a significant improvement with respect to both passive control and Linear Model Predictive Control, due to a better representation of the viscous forces affecting the dynamics of the device. Furthermore, the nonlinear formulation allows to consider more realistic power flow configurations, such as non-reversible power flow, in the form of nonlinear inequality constraints.

1.1 Organization of the thesis

The content of this thesis can be summarized as follows: Chapter 2 introduces new low-storage high-order time integration schemes of the Runge-Kutta type for the integration of linearly stiff ordinary differential equations. Compared to the state of the art, the new schemes offer comparable or improved stability and accuracy properties with significantly reduced storage requirement.

Chapter 3 introduces new low-storage IMEXRK schemes specifically designed for incremental implementation. These schemes improve accuracy and stability properties of the other two schemes of this kind available in literature: **CN/RKW3** and the scheme in [1]. Remarkably, this is achieved while guaranteeing the same or slightly increased storage requirement. Moreover, the higher computational cost associated to some of these schemes proves to be largely compensated by the improvement in accuracy and stability that such schemes allow.

Chapter 4 describes two new relaxation algorithms for the smoothing step in the multigrid solution of elliptic PDEs over stretched structured grids. Compared to the standard approach, the new schemes guarantee comparable convergence rate, at significantly reduced computational time.

Chapter 5 presents some modifications of the Thomas algorithm typically used for the factorization of tridiagonal matrices. Such algorithms allow the efficient solution of sparse linear systems arising from the discretization of one-dimensional partial differential equations defined over closed connected domains. A remarkable application of these algorithms is given by the relaxation schemes in Chapter 4.

Chapter 6 describes the implementation of Ensemble Kalman Filter for the short-term prediction of ocean wave elevation. A nonlinear pseudospectral model, leveraging one our low-storage time integration schemes, is used to simulate wave propagation. Performance is assessed considering different measurement setups involving either an ocean wave radar or arrays of monitoring buoys.

Finally, Chapter 7 introduces a new approach to the analytic computation of discretized trajectories and associated gradients in Nonlinear Model Predictive Control leveraging a direct multiple shooting formulation. This guarantees a reduced computational cost with respect to other approaches relying on numerical differentiation schemes, such as finite differences, automatic differentiation, and complex step derivative.

Chapter 2

Low-storage implicit/explicit Runge-Kutta schemes for the simulation of stiff high-dimensional ODE systems

2.1 Introduction

Although a wide variety of methods have been used for spatial discretization and subgrid-scale modeling in the Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES) of turbulent flows, time marching schemes for such systems have relied, in most cases, on an implicit scheme for the advancement of the stiff terms and an explicit scheme for the advancement of the nonstiff terms. Among these so-called IMEX schemes, an approach that gained favor due to [2] and [3] coupled the (implicit, second-order) Crank-Nicolson (CN) scheme for the stiff terms with the (explicit) second-order Adams-Bashforth (AB2) scheme for the nonstiff terms. This approach was refined in [4], which used the (implicit) CN scheme for the stiff terms, at each RK substep, together with the (explicit) third-order low-storage Runge-Kutta-Wray (RKW3) scheme [5] for the nonstiff terms. This venerable IMEX algorithm, dubbed **CN/RKW3**, still enjoys extensive use today, and is particularly appealing, as only two registers are required for advancing the ODE in time, though if three registers are used, the number of flops required by the algorithm may be significantly reduced. In high-dimensional discretizations of 3D PDE systems on modern computational hardware, the reduced memory footprint of this time marching algorithm, in its two-register or three-register form, can significantly reduce the execution time of a simulation. However, the **CN/RKW3** scheme has the considerable disadvantage of being only second-order accurate, and its implicit part is only A-stable. In recent years, there have been relatively few attempts to refine the **CN/RKW3** time-marching scheme for turbulence simulations, perhaps due to a mistaken notion that modifying it to achieve higher order might result in either increased storage requirements, significantly more computation per timestep, or the loss of A stability of the implicit part. It turns out that this is untrue; in fact, there is much to be gained by revising this algorithm.

When using an IMEX scheme, such as those described above, to march the incompressible Navier-Stokes equation, one natural choice is to treat the (linear) diffusion terms as the “stiff terms” and the (nonlinear) convective terms as the “nonstiff terms”. Note that a better choice for discretizations with significant grid clustering implemented in one or more spatial directions, as usually present when simulating wall-bounded turbulent flows, is to treat the diffusion and linearized convection terms with derivatives in the direction of most significant grid clustering (e.g., in the direction normal to the nearest wall) as the “stiff” terms, and the remaining terms as the “nonstiff” terms, as suggested by [6]. Note further that so-called fractional step methods are often combined with such IMEX schemes in order to enforce the incompressibility constraint (see, e.g., [4]). This chapter focuses exclusively on the IMEXRK part of such time-advancement algorithms; various creative choices for which terms to take implicitly at different points in the physical domain of interest, and various methods for implementing fractional step techniques for enforcing exactly the divergence-free constraint, may subsequently be addressed in an identical manner as discussed in [4], and [6], and elsewhere in the literature.

Over the last 30 years, there has been significant development of (full-storage) IMEXRK algorithms. A comprehensive review of this literature is given in [7], and a brief summary of this subject is given in §2.1.1 below, including the general structure of full-storage IMEXRK schemes, their general implementation, conditions on their parameters for second-, third-, and fourth-order accuracy, and characterizations of their stability.

Further, in the years since the development of RKW3 in [5], there has been significant development of alternative low-storage explicit RK schemes; a comprehensive review of this literature is given in [8], and a brief summary of this subject is given in §2.1.2 below, including the extension to implicit RK schemes, the introduction of a general 2-register IMEXRK form, efficient 3-register & 2-register implementations of this form, as well as the introduction of a general 3-register IMEXRK form, and efficient 4-register & 3-register implementations of this form.

We then develop eight new low-storage IMEXRK schemes well suited for turbulent flow simulations, and other computational grand challenge applications, using two, three, or four registers of length N (the dimension of the ODE under consideration). With an eye on the computational cost of their implementation, we focus on schemes with the smallest number of stages possible for a given order, stability, and storage requirement. A comprehensive summary of the schemes described in this chapter is given in Table 2.1. In short:

- §2.2 presents two second-order, 2-register IMEXRK schemes: the classic 3-stage, A-stable, **CN/RKW3** scheme; a new, $(2,3)$-stage, that is, a scheme with 2 implicit stages and 3 explicit stages, L-stable, strong-stability-preserving scheme, dubbed **IMEXRKCB2**.

- §2.3 presents five new third-order 2-register IMEXRK schemes: a $(2,3)$-stage strongly A-stable scheme, dubbed **IMEXRKCB3a**; a $(3,4)$-stage, strongly A-stable scheme with ESDIRK implicit part, dubbed **IMEXRKCB3b**; three $(3,4)$-stage, L-stable schemes: one with coefficients selected to maximize stability of the ERK part on the negative real axis while being strong stability preserving, dubbed **IMEXRKCB3c**; one with coefficients selected to be strong stability preserving for the maximum possible timestep, dubbed **IMEXRKCB3d**; one with coefficients selected to maximize accuracy of the ERK part, dubbed **IMEXRKCB3e**.

- §2.4 presents a new third-order, 3-register, 4-stage, L-stable, stage-order-2 scheme dubbed **IMEXRKCB3f**.

- §2.5 presents a new fourth-order, 3-register, 6-stage, L-stable, stage-order-2 scheme dubbed **IMEXRKCB4**.

In §2.6, we provide an analysis of the well-known order reduction phenomenon arising during the integration of very stiff ODEs using these IMEXRK schemes. Finally, §2.7 considers the application of all of these low-storage IMEXRK schemes, and some of their full-storage IMEXRK competitors, to a representative test problem in order to compare their computational efficiency.

Table 2.1: Summary of the properties of the eight IMEXRK schemes presented in this chapter (top) and eight of the leading IMEXRK competitors from literature (bottom), including the leading-order computational cost per timestep for efficient finite-difference (FD) and pseudospectral (PS) implementation of each scheme on the 1D Kuramoto-Sivashinsky (KS) equation.

Scheme	Order	Registers	Stages $(s^{\mathrm{I}}, s^{\mathrm{E}})$	Stability of DIRK part $[\sigma(z^{\mathrm{I}} \to \infty; z^{\mathrm{E}})]$	Stability of ERK part on negative real axis	Truncation error	Other properties	FD cost for 1D KS	PS cost
IMEXRKCB2	second	[2R]	$(2, 3)$	L-stable [0]	$-5.81 \leq z^{\mathrm{E}} \leq 0$	$A^{(3)} = 0.114$	embedded, SSP ($c = 1.0$)	$90N$ flops (3-reg), $101N$ flops (2-reg)	6 FFTs (3-reg)
IMEXRKCB3a	third	[2R]	$(2, 3)$	strongly A-stable $[-0.738]$	$-2.51 \leq z^{\mathrm{E}} \leq 0$	$A^{(4)} = 0.226$		$90N$ flops (3-reg), $101N$ flops (2-reg)	6 FFTs (3-reg)
IMEXRKCB3b			$(3, 4)$	strongly A-stable $[-0.732 - 0.366z^{\mathrm{E}}]$	$-2.21 \leq z^{\mathrm{E}} \leq 0$	$A^{(4)} = 0.186$	ESDIRK	$130N$ flops (3-reg), $139N$ flops (2-reg)	8 FFTs (3-reg)
IMEXRKCB3c				L-stable [0]	$-6.00 \leq z^{\mathrm{E}} \leq 0$	$A^{(4)} = 0.113$	embedded, SSP ($c = 0.70$)	$133N$ flops (3-reg), $157N$ flops (2-reg)	8 FFTs (3-reg)
IMEXRKCB3d					$-2.52 \leq z^{\mathrm{E}} \leq 0$	$A^{(4)} = 0.207$	embedded, SSP ($c = 0.77$)		
IMEXRKCB3e					$-2.79 \leq z^{\mathrm{E}} \leq 0$	$A^{(4)} = 0.0824$			
IMEXRKCB3f		[3R]	$(4, 4)$	L-stable [0]	$-6.00 \leq z^{\mathrm{E}} \leq 0$	$A^{(4)} = 0.107$	embedded, SO2	$162N$ flops (4-reg), $266N$ flops (3-reg)	8 FFTs (4-reg)
IMEXRKCB4	fourth	[3R]	$(6, 6)$	L-stable [0]	$-6.32 \leq z^{\mathrm{E}} \leq 0$	$A^{(5)} = 0.0157$	embedded, SO2	$253N$ flops (4-reg), $458N$ flops (3-reg)	12 FFTs (4-reg)
CN/RKW3	second	[2R]	$(3, 3)$	A-stable $[-1]$	$-2.51 \leq z^{\mathrm{E}} \leq 0$	$A^{(3)} = 0.0387$		$115N$ flops (3-reg), $127N$ flops (2-reg)	6 FFTs (3-reg)
Ascher(2,3,3) (see [9])	third	7	$(2, 3)$	strongly A-stable $[-0.732 - 0.732\, z^{\mathrm{E}}]$	$-2.51 \leq z^{\mathrm{E}} \leq 0$	$A^{(4)} = 0.206$		$92N$ flops	6 FFTs
Ascher(3,4,3) (see [9])		9	$(3, 4)$	L-stable $[0.106\, z^{\mathrm{E}}]$	$-2.78 \leq z^{\mathrm{E}} \leq 0$	$A^{(4)} = 0.103$		$141N$ flops	8 FFTs
Ascher(4,4,3) (see [9])		10	$(4, 4)$	L-stable [0]	$-2.14 \leq z^{\mathrm{E}} \leq 0$	$A^{(4)} = 0.163$		$190N$ flops	8 FFTs
LIRK3 (see [10])		9	$(3, 4)$		$-2.21 \leq z^{\mathrm{E}} \leq 0$	$A^{(4)} = 0.100$		$139N$ flops	8 FFTs
ARK3(2)4L[2]SA (see [7])		10	$(4, 4)$		$-3.66 \leq z^{\mathrm{E}} \leq 0$	$A^{(4)} = 0.0722$	embedded	$159N$ flops	8 FFTs
LIRK4 (see [10])	fourth	13	$(5, 6)$	L-stable [0]	$-3.41 \leq z^{\mathrm{E}} \leq 0$	$A^{(5)} = 0.0404$		$249N$ flops	12 FFTs
ARK4(3)6L[2]SA (see [7])		14	$(6, 6)$		$-4.23 \leq z^{\mathrm{E}} \leq 0$	$A^{(5)} = 0.0122$	embedded	$270N$ flops	12 FFTs

2.1.1 Full-storage IMEXRK schemes and their Butcher tableaux

A comprehensive review of (full-storage) IMEXRK schemes is given by Kennedy, Carpenter, & Lewis [7]. In short, IMEXRK schemes incorporate a coordinated pair of Diagonally Implicit Runge-Kutta (DIRK, with lower-triangular A) and Explicit Runge-Kutta (ERK, with strictly lower-triangular A) schemes, with parameters as summarized in the standard Butcher tableaux

$$
\begin{array}{c|cccc}
c_1^{\mathrm{I}} & a_{1,1}^{\mathrm{I}} & & & \\
c_2^{\mathrm{I}} & a_{2,1}^{\mathrm{I}} & a_{2,2}^{\mathrm{I}} & & \\
\vdots & \vdots & \ddots & \ddots & \\
c_s^{\mathrm{I}} & a_{s,1}^{\mathrm{I}} & \cdots & a_{s,s-1}^{\mathrm{I}} & a_{s,s}^{\mathrm{I}} \\
\hline
 & b_1^{\mathrm{I}} & \cdots & b_{s-1}^{\mathrm{I}} & b_s^{\mathrm{I}} \\
\hline
 & \hat{b}_1^{\mathrm{I}} & \cdots & \hat{b}_{s-1}^{\mathrm{I}} & \hat{b}_s^{\mathrm{I}}
\end{array}
\qquad
\begin{array}{c|cccc}
c_1^{\mathrm{E}} & 0 & & & \\
c_2^{\mathrm{E}} & a_{2,1}^{\mathrm{E}} & 0 & & \\
\vdots & \vdots & \ddots & \ddots & \\
c_s^{\mathrm{E}} & a_{s,1}^{\mathrm{E}} & \cdots & a_{s,s-1}^{\mathrm{E}} & 0 \\
\hline
 & b_1^{\mathrm{E}} & \cdots & b_{s-1}^{\mathrm{E}} & b_s^{\mathrm{E}} \\
\hline
 & \hat{b}_1^{\mathrm{E}} & \cdots & \hat{b}_{s-1}^{\mathrm{E}} & \hat{b}_s^{\mathrm{E}}
\end{array}
\tag{2.1}
$$

for the time advancement of an ODE of the form

$$
\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x},\, t) + \mathbf{g}(\mathbf{x},\, t), \tag{2.2}
$$

where $\mathbf{f}(\mathbf{x},\, t)$ represents the stiff part of the RHS [advanced with the DIRK method at left in (2.1)], and $\mathbf{g}(\mathbf{x},\, t)$ represents the nonstiff part of the RHS [simultaneously advanced with the ERK method at right in (2.1)].

If the stiff part of the ODE is linear [that is, if $\mathbf{f}(\mathbf{x},\, t) = A\mathbf{x}$] then, denoting the efficient solution of $A\mathbf{x} = \mathbf{b}$ as $A^{-1}\mathbf{b}$, a full-storage implementation of the IMEXRK scheme in (2.1) to advance from $\mathbf{x} = \mathbf{x}_n$ to $\mathbf{x} = \mathbf{x}_{n+1}$ proceeds as follows

```
for  $k = 1 : s$                                                                     (2.3a)
  if  $k == 1$,  $\mathbf{y} = \mathbf{x}$,  else                                    (2.3b)
  $\mathbf{y} = \mathbf{x} + \sum_{i=1}^{k-1} a^{\mathrm{I}}_{k,i}\,\Delta t\,\mathbf{f}_i + \sum_{j=1}^{k-1} a^{\mathrm{E}}_{k,j}\,\Delta t\,\mathbf{g}_j$,  end        (2.3c)
  $\mathbf{f}_k = A\,(I - a^{\mathrm{I}}_{k,k}\,\Delta t\,A)^{-1}\mathbf{y}$   [equivalently, $\mathbf{f}_k = (I - a^{\mathrm{I}}_{k,k}\,\Delta t\,A)^{-1}A\,\mathbf{y}$ ]        (2.3d)
  $\mathbf{g}_k = \mathbf{g}(\mathbf{y} + a^{\mathrm{I}}_{k,k}\,\Delta t\,\mathbf{f}_k,\ t_n + c^{\mathrm{E}}_k\,\Delta t)$        (2.3e)
end                                                                                  (2.3f)
$\mathbf{x} \leftarrow \mathbf{x} + \sum_{i=1}^{s} b^{\mathrm{I}}_i\,\Delta t\,\mathbf{f}_i + \sum_{j=1}^{s} b^{\mathrm{E}}_j\,\Delta t\,\mathbf{g}_j$        (2.3g)
$\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \sum_{i=1}^{s} \hat{b}^{\mathrm{I}}_i\,\Delta t\,\mathbf{f}_i + \sum_{j=1}^{s} \hat{b}^{\mathrm{E}}_j\,\Delta t\,\mathbf{g}_j$        (2.3h)
```

Line (2.3d) above is simply $\mathbf{f}_k = \mathbf{f}(\mathbf{z},\ t_n + c^{\mathrm{I}}_k\Delta t)$, where $\mathbf{z}$ is the solution of $\mathbf{e}(\mathbf{z}) = \mathbf{z} - \mathbf{y} - a^{\mathrm{I}}_{k,k}\,\Delta t\,\mathbf{f}(\mathbf{z}, t_n + c^{\mathrm{I}}_k\Delta t) = 0$ [that is, where $\mathbf{z} = \mathbf{y} + a^{\mathrm{I}}_{k,k}\,\Delta t\,\mathbf{f}(\mathbf{z}, t_n + c^{\mathrm{I}}_k\Delta t)$], in the special case that $\mathbf{f}(\mathbf{x},\ t) = A\mathbf{x}$. More generally, if the stiff part $\mathbf{f}(\mathbf{x},\ t)$ is nonlinear, then line (2.3d) is replaced by a Newton-Raphson iteration (see [11]) to find the $\mathbf{z}$ such that $\mathbf{e}(\mathbf{z}) = 0$:

$$\left.\begin{array}{ll}
\text{Initialize:} & \mathbf{z}_0 = \mathbf{y} + a^{\mathrm{I}}_{k,k}\,\Delta t\,\mathbf{f}(\mathbf{y}, t_n + c^{\mathrm{I}}_k\Delta t) \\[2ex]
\text{Iterate:} & \Big(I - a^{\mathrm{I}}_{k,k}\,\Delta t\,\dfrac{\partial\mathbf{f}(\mathbf{x},\ t_n + c^{\mathrm{I}}_k\Delta t)}{\partial\mathbf{x}}\Big|_{\mathbf{x}=\mathbf{z}_m}\Big)(\mathbf{z}_{m+1} - \mathbf{z}_m) = -\mathbf{z}_m + \mathbf{y} + a^{\mathrm{I}}_{k,k}\,\Delta t\,\mathbf{f}(\mathbf{z}_m,\ t_n + c^{\mathrm{I}}_k\Delta t) \\[2ex]
\text{Upon exit:} & \mathbf{f}_k = \mathbf{f}(\mathbf{z}_{\text{converged}},\ t_n + c^{\mathrm{I}}_k\Delta t)
\end{array}\right\} \tag{3c'}$$

The Jacobian used in this iteration may be computed analytically or approximated numerically. The low-storage IMEXRK algorithms developed in this chapter may be applied in the linear or nonlinear setting, mutatis mutandis; §§2.1.2 - 2.1.2 provide low-storage pseudocode implementations in the case in which the stiff part of the ODE is linear.

Finally, note that the $\hat{b}^{\mathrm{I}}_i$ and $\hat{b}^{\mathrm{E}}_i$ coefficients in the Butcher tableaux, if provided, are used to form a so-called embedded scheme to advance the solution at each timestep with an order of accuracy reduced by one with respect to the main scheme. Using this embedded scheme, one may estimate the error of the simulation at each timestep, and adjust the stepsize at the next iteration accordingly.

As is well known (see, e.g., [12]), for the DIRK and ERK components in (2.1), when used in isolation, to be first-order accurate, it is required that

$$\tau_1^{(1)\mathrm{I}} = \sum\nolimits_i b_i^{\mathrm{I}} - 1 = 0 \qquad \tau_1^{(1)\mathrm{E}} = \sum\nolimits_i b_i^{\mathrm{E}} - 1 = 0, \tag{2.4a}$$

for these schemes, when used in isolation, to be second-order accurate, it is additionally required that

$$\tau_1^{(2)\mathrm{I}} = \sum\nolimits_i b_i^{\mathrm{I}} c_i^{\mathrm{I}} - 1/2 = 0 \qquad \tau_1^{(2)\mathrm{E}} = \sum\nolimits_i b_i^{\mathrm{E}} c_i^{\mathrm{E}} - 1/2 = 0, \tag{2.4b}$$

for these schemes, when used in isolation, to be third-order accurate, it is additionally required that

$$\tau_1^{(3)\mathrm{I}} = (1/2) \sum\nolimits_i b_i^{\mathrm{I}} c_i^{\mathrm{I}} c_i^{\mathrm{I}} - 1/6 = 0 \qquad \tau_1^{(3)\mathrm{E}} = (1/2) \sum\nolimits_i b_i^{\mathrm{E}} c_i^{\mathrm{E}} c_i^{\mathrm{E}} - 1/6 = 0 \tag{2.4c}$$

$$\tau_2^{(3)\mathrm{I}} = \sum\nolimits_{i,j} b_i^{\mathrm{I}} a_{i,j}^{\mathrm{I}} c_j^{\mathrm{I}} - 1/6 = 0 \qquad \tau_2^{(3)\mathrm{E}} = \sum\nolimits_{i,j} b_i^{\mathrm{E}} a_{i,j}^{\mathrm{E}} c_j^{\mathrm{E}} - 1/6 = 0, \tag{2.4d}$$

and for these schemes, when used in isolation, to be fourth-order accurate, it is additionally required that

$$\tau_1^{(4)\mathrm{I}} = (1/6)\sum\nolimits_i b_i^{\mathrm{I}} c_i^{\mathrm{I}} c_i^{\mathrm{I}} c_i^{\mathrm{I}} - 1/24 = 0 \qquad \tau_1^{(4)\mathrm{E}} = (1/6)\sum\nolimits_i b_i^{\mathrm{E}} c_i^{\mathrm{E}} c_i^{\mathrm{E}} c_i^{\mathrm{E}} - 1/24 = 0 \qquad (2.4\mathrm{e})$$

$$\tau_2^{(4)\mathrm{I}} = (1/3)\sum\nolimits_{i,j} b_i^{\mathrm{I}} c_i^{\mathrm{I}} a_{i,j}^{\mathrm{I}} c_j^{\mathrm{I}} - 1/24 = 0 \qquad \tau_2^{(4)\mathrm{E}} = (1/3)\sum\nolimits_{i,j} b_i^{\mathrm{E}} c_i^{\mathrm{E}} a_{i,j}^{\mathrm{E}} c_j^{\mathrm{E}} - 1/24 = 0 \qquad (2.4\mathrm{f})$$

$$\tau_3^{(4)\mathrm{I}} = (1/2)\sum\nolimits_{i,j} b_i^{\mathrm{I}} a_{i,j}^{\mathrm{I}} c_j^{\mathrm{I}} c_j^{\mathrm{I}} - 1/24 = 0 \qquad \tau_3^{(4)\mathrm{E}} = (1/2)\sum\nolimits_{i,j} b_i^{\mathrm{E}} a_{i,j}^{\mathrm{E}} c_j^{\mathrm{E}} c_j^{\mathrm{E}} - 1/24 = 0 \qquad (2.4\mathrm{g})$$

$$\tau_4^{(4)\mathrm{I}} = \sum\nolimits_{i,j,k} b_i^{\mathrm{I}} a_{i,j}^{\mathrm{I}} a_{j,k}^{\mathrm{I}} c_k^{\mathrm{I}} - 1/24 = 0 \qquad \tau_4^{(4)\mathrm{E}} = \sum\nolimits_{i,j,k} b_i^{\mathrm{E}} a_{i,j}^{\mathrm{E}} a_{j,k}^{\mathrm{E}} c_k^{\mathrm{E}} - 1/24 = 0. \qquad (2.4\mathrm{h})$$

Recall that, in the scalar case, the exact solution of $x' = f(x) + g(x)$ has the following terms:

$$\begin{aligned} x_{n+1} &= x_n + \Delta t\, x_n' + (\Delta t)^2\, x_n''/2! + (\Delta t)^3\, x_n'''/3! + O((\Delta t)^4) \\ &= x_n + \Delta t\big\{f + g\big\}_{(x_n,t_n)} + \frac{(\Delta t)^2}{2!}\big\{f'f + f'g + g'f + g'g\big\}_{(x_n,t_n)} \\ &\quad + \frac{(\Delta t)^3}{3!}\big\{f''ff + 2f''fg + f''gg + g''ff + 2g''fg + g''gg + f'f'f + f'g'f \\ &\quad + g'f'f + g'g'f + f'f'g + f'g'g + g'f'g + g'g'g\big\}_{(x_n,t_n)} + O((\Delta t)^4); \end{aligned}$$

note in particular that there are 2 terms at second order and 10 terms at third order that involve both f and g. For the DIRK and ERK components in (2.1), when used together in an IMEX fashion, to be second-order accurate, it is thus additionally required that

$$\tau_1^{(2)\mathrm{IE}} = \sum\nolimits_i b_i^{\mathrm{I}} c_i^{\mathrm{E}} - 1/2 = 0 \qquad \tau_2^{(2)\mathrm{IE}} = \sum\nolimits_i b_i^{\mathrm{E}} c_i^{\mathrm{I}} - 1/2 = 0, \qquad (2.4\mathrm{i})$$

for these schemes, when used together in an IMEX fashion, to be third-order accurate, it is additionally required that

$$\tau_1^{(3)\mathrm{IE}} = (1/2) \sum_i b_i^{\mathrm{I}} c_i^{\mathrm{E}} c_i^{\mathrm{E}} - 1/6 = 0 \qquad \tau_2^{(3)\mathrm{IE}} = (1/2) \sum_i b_i^{\mathrm{E}} c_i^{\mathrm{I}} c_i^{\mathrm{I}} - 1/6 = 0 \tag{2.4j}$$

$$\tau_3^{(3)\mathrm{IE}} = (1/2) \sum_i b_i^{\mathrm{I}} c_i^{\mathrm{I}} c_i^{\mathrm{E}} - 1/6 = 0 \qquad \tau_4^{(3)\mathrm{IE}} = (1/2) \sum_i b_i^{\mathrm{E}} c_i^{\mathrm{I}} c_i^{\mathrm{E}} - 1/6 = 0 \tag{2.4k}$$

$$\tau_5^{(3)\mathrm{IE}} = \sum_{i,j} b_i^{\mathrm{I}} a_{i,j}^{\mathrm{E}} c_j^{\mathrm{E}} - 1/6 = 0 \qquad \tau_6^{(3)\mathrm{IE}} = \sum_{i,j} b_i^{\mathrm{E}} a_{i,j}^{\mathrm{I}} c_j^{\mathrm{I}} - 1/6 = 0 \tag{2.4l}$$

$$\tau_7^{(3)\mathrm{IE}} = \sum_{i,j} b_i^{\mathrm{E}} a_{i,j}^{\mathrm{E}} c_j^{\mathrm{I}} - 1/6 = 0 \qquad \tau_8^{(3)\mathrm{IE}} = \sum_{i,j} b_i^{\mathrm{I}} a_{i,j}^{\mathrm{I}} c_j^{\mathrm{E}} - 1/6 = 0 \tag{2.4m}$$

$$\tau_9^{(3)\mathrm{IE}} = \sum_{i,j} b_i^{\mathrm{I}} a_{i,j}^{\mathrm{E}} c_j^{\mathrm{I}} - 1/6 = 0 \qquad \tau_{10}^{(3)\mathrm{IE}} = \sum_{i,j} b_i^{\mathrm{E}} a_{i,j}^{\mathrm{I}} c_j^{\mathrm{E}} - 1/6 = 0, \tag{2.4n}$$

and for these schemes, when used together in an IMEX fashion, to be fourth-order accurate, 44 additional constraints are required (see [7]), which for brevity aren't listed here.

Stability

The stability of an RK scheme may be characterized by considering the model problem $dx/dt = \lambda x$ and defining $z = \lambda \Delta t$, $\sigma(z) = x_{n+1}/x_n$, and $\sigma(\infty) \triangleq \lim_{|z|\to\infty} \sigma(z)$. The stability function of an RK scheme with Butcher tableau parameters A and $\mathbf{b}$ is then given by $\sigma(z) = 1 + z\mathbf{b}^T(\mathbf{I} - z\mathbf{A})^{-1}\mathbf{e}$, where $\mathbf{e}$ denotes a vector of ones; the RK scheme is said to be stable for any z such that $|\sigma(z)| \leq 1$. Further, considering its application to stiff systems, an RK scheme is said to be

- A-stable if $|\sigma(z)| \leq 1$ over the entire LHP of z,
- strongly A-stable if it is A-stable and $|\sigma(\infty)| < 1$, and
- L-stable if it is A-stable and $\sigma(\infty) = 0$.

The stability of an IMEXRK scheme is a bit more difficult to characterize. Of course, one may start by characterizing the stability of the implicit and explicit parts considered in isolation. To evaluate the stability of the implicit and explicit components of an IMEX scheme working together, we consider the model problem $dx/dt = \lambda_f x + \lambda_g x$, where the first term on the RHS is handled implicitly, and the second term on the RHS is handled explicitly. Defining $z^{\mathrm{I}} = \lambda_f \, \Delta t$, $z^{\mathrm{E}} = \lambda_g \, \Delta t$, and $\sigma(z^{\mathrm{I}}, z^{\mathrm{E}}) = x_{n+1}/x_n$, we may write (see [7])

$$\sigma(z^{\mathrm{I}}, z^{\mathrm{E}}) = \frac{\det\left[I - z^{\mathrm{I}}A^{\mathrm{I}} - z^{\mathrm{E}}A^{\mathrm{E}} + z^{\mathrm{I}}\mathbf{e}(\mathbf{b}^{\mathrm{I}})^T + z^{\mathrm{E}}\mathbf{e}(\mathbf{b}^{\mathrm{E}})^T\right]}{\det\left[I - z^{\mathrm{I}}A^{\mathrm{I}}\right]}. \tag{2.5}$$

We may then characterize the stability of the implicit and explicit parts of an IMEXRK scheme working in concert, when the implicit part of the problem is stiff, by looking at $\sigma(z^{\mathrm{I}}, z^{\mathrm{E}})$ as $z^{\mathrm{I}} \to \infty$ for finite z^{E}.

Strong-stability preserving (SSP) schemes

Consider the 1D hyperbolic PDE

$$\partial u/\partial t = -\partial f(u)/\partial x; \tag{2.6}$$

denoting $u_i(t)$ as the discretization of $u(x, t)$ on N spatial grid points x_i, and denoting $\mathbf{u}(t)$ as a vector containing all of the $u_i(t)$, we write the spatial discretization of this PDE as the ODE

$$d\mathbf{u}/dt = L(\mathbf{u}). \tag{2.7}$$

If a TVD spatial discretization is used, such as a Godunov or MUSCL scheme with an appropriate flux limiter incorporated (see [13]), then applying a simple Explicit Euler time discretization to (2.7),

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \, L(\mathbf{u}^n), \tag{2.8}$$

under the appropriate CFL condition on the timestep, $\Delta t \leq \Delta t_{CFL}$, results in a simulation exhibiting a total variation of the discrete solution which does not increase in time, that is,

$$TV(\mathbf{u}^{n+1}) \leq TV(\mathbf{u}^n), \quad \text{where} \quad TV(\mathbf{u}^n) = \sum\nolimits_j \left| u^n_{j+1} - u^n_j \right|. \tag{2.9}$$

Strong-stability preserving (SSP) explicit time-discretization methods (see [14] and [15]) are simply higher-order time discretization methods that conserve this total variation diminishing property with a modified CFL condition on the timestep, $\Delta t \leq c\,\Delta t_{CFL}$.

In [15] (see also [16]), a condition for an explicit Runge-Kutta scheme to be SSP has been developed. This condition states that if an s-stage explicit Runge-Kutta scheme is written in incremental form, that is,

$$\begin{aligned}
\mathbf{u}^{(0)} &= \mathbf{u}^n \\
\mathbf{u}^{(i)} &= \sum\nolimits_{j=0}^{i-1} \left(\alpha_{i,j}\, \mathbf{u}^{(j)} + \Delta t \beta_{i,j}\, \mathbf{L}(\mathbf{u}^{(j)}) \right) \quad \text{for} \quad i = 1, \ldots, s \\
\mathbf{u}^{n+1} &= \mathbf{u}^{(s)},
\end{aligned}$$

where all of the $\alpha_{i,j} \geq 0$, and if the forward Euler method applied to the ODE (2.7) arising from a TVD spatial discretization of the hyperbolic PDE (2.6) is strongly stable under the appropriate CFL restriction, then such an explicit Runge-Kutta method is SSP provided that all of the $\beta_{i,j} \geq 0$ and that the following CFL restriction is fulfilled:

$$\Delta t \leq c\,\Delta t_{CFL}, \qquad c = \min_{i,j} \frac{\alpha_{i,j}}{\beta_{i,j}}. \tag{2.10}$$

In case an explicit scheme is coupled with an implicit scheme, as in an IMEXRK formulation, then, provided the implicit scheme used to integrate the stiff part of the ODE is

L-stable, in the stiff limit the time integration scheme becomes the explicit Runge-Kutta scheme, and the order of accuracy of the limiting scheme is greater than or equal to the order of accuracy of the IMEXRK scheme itself. Hence, as stated in [17], if the explicit part of the IMEXRK scheme is SSP, then the IMEXRK scheme will also be SSP in the stiff limit.

In [17], three full-storage second-order and two full-storage third-order IMEXRK schemes are presented which are SSP in the stiff limit; no other IMEXRK schemes with this SSP property were found in our review of the IMEXRK literature. In the present chapter we introduce three new IMEXRK schemes which are SSP in the stiff limit (one which is second-order and two which are third-order); unlike the schemes in [17], the IMEXRK schemes derived here are of the low-storage variety.

2.1.2 Low-storage IMEXRK schemes

The existing literature on low-storage RK schemes to date appears to focus exclusively on explicit schemes. Note that a cavalier implementation of a full-storage ERK scheme [see the explicit part of (2.3)] requires storage of the state vector [$\mathbf{x}$], the intermediate vector [$\mathbf{y}$], and s values of the RHS vectors [$\mathbf{g}_k$]; that is, $s + 2$ vectors of length N, where $\mathbf{x} = \mathbf{x}_{N\times 1}$. We now summarize the two main classes of low-storage ERK schemes[1], a comprehensive review of which is given in Kennedy, Carpenter, & Lewis [8].

The two-register Williamson class of ERK schemes [18], denoted “[2*N*]” schemes,

[1] Both the Williamson class and the van der Houwen class of ERK schemes extend to ERK variants that require, at minimum, three, four, or more registers for their implementation; with an eye on the computational cost of their implementation, we focus in this chapter on schemes which admit a two- or three-register implementation.

may be written to advance from $\mathbf{x} = \mathbf{x}_n$ to $\mathbf{x} = \mathbf{x}_{n+1}$ as

```
for   $k = 1 : s$
   if   $k == 1$,   $\Delta\mathbf{x} \leftarrow \Delta t\, \mathbf{g}(\mathbf{x},\, t_n + c_k \Delta t)$,   else
      $\Delta\mathbf{x} \leftarrow \alpha_k\, \Delta\mathbf{x} + \Delta t\, \mathbf{g}(\mathbf{x},\, t_n + c_k \Delta t)$
                                                                        (2.11)
   end
   $\mathbf{x} \leftarrow \mathbf{x} + \beta_k\, \Delta\mathbf{x}$
end
```

If handled with care, such schemes can often be implemented efficiently in two registers of length N, $\mathbf{x}$ and $\Delta\mathbf{x}$.

The two-register van der Houwen class of schemes [19], denoted “[2R]” schemes, restrict the parameters $a_{i,j}$ below the first subdiagonal in the Butcher tableau of the ERK scheme to be equal to the parameters b_j of the corresponding column, and may thus be written to advance from $\mathbf{x} = \mathbf{x}_n$ to $\mathbf{x} = \mathbf{x}_{n+1}$ as

```
for   $k = 1 : s$
   if   $k == 1$,   $\mathbf{y} \leftarrow \mathbf{x}$,   else
      $\mathbf{y} \leftarrow \mathbf{x} + (a_{k,k-1} - b_{k-1})\, \Delta t\, \mathbf{g}(\mathbf{y},\, t_n + c_{k-1} \Delta t)$
                                                                        (2.12)
   end
   $\mathbf{x} \leftarrow \mathbf{x} + b_k\, \Delta t\, \mathbf{g}(\mathbf{y},\, t_n + c_k \Delta t)$
end
```

Such schemes can often be implemented efficiently in two registers of length N (namely, $\mathbf{x}$ and $\mathbf{y}$). If implemented with three registers, however, the function $\mathbf{g}(\mathbf{y},\, t_n + c_k \Delta t)$ can be computed just once per timestep (instead of twice). RKW3 [5] is a commonly-used example of a two-register, three-stage, third-order van der Houwen ERK scheme, with a Butcher tableau of

$$\begin{array}{c|ccc} 0 & 0 & & \\ 8/15 & 8/15 & 0 & \\ 2/3 & 1/4 & 5/12 & 0 \\ \hline & 1/4 & 0 & 3/4 \end{array} \tag{2.13}$$

In the three-register van der Houwen class of schemes, denoted "[3R]" schemes, only the parameters $a_{i,j}$ below the *second* subdiagonal of the Butcher tableau of the ERK scheme must equal the parameters b_j of the corresponding column. An effective implementation of such [3R] schemes that uses only three registers of length N (namely, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$) is given by

```
for  $k = 1 : s$
  if  $k == 1$,  $\mathbf{y} \leftarrow \mathbf{x}$,  $\mathbf{z} \leftarrow \mathbf{x}$,  else,
    $\mathbf{z} \leftarrow \mathbf{y} + a_{k,k-1}\,\Delta t\,\mathbf{g}(\mathbf{y},\, t_n + c_{k-1}\Delta t)$
    if  $k < s$,  $\mathbf{y} \leftarrow \mathbf{x} + (a_{k+1,k-1} - b_{k-1})\,\mathbf{g}(\mathbf{y},\, t_n + c_{k-1}\Delta t)$,  end        (2.14)
  end
  $\mathbf{x} \leftarrow \mathbf{x} + b_k\,\Delta t\,\mathbf{g}(\mathbf{y},\, t_n + c_k\Delta t)$
end
```

Again, if implemented with four registers, the function $\mathbf{g}(\mathbf{y},\, t_n + c_k\Delta t)$ can be computed just once per timestep (instead of thrice). In the present chapter, we extend the two- and three-register van der Houwen classes of ERK schemes to the DIRK case, which can be accomplished with precisely the same restrictions on the (lower triangular) DIRK Butcher tableau as in the (strictly lower triangular) ERK case, as specified above. Further, we will develop coordinated pairs of such [2R] and [3R] DIRK and ERK schemes in the IMEX setting described in §2.1.1. In particular, we will develop a [2R] second-order IMEX scheme, [2R] and [3R] third-order IMEX schemes, and a [3R] fourth-order IMEX scheme.

As shown in §2.1.1, six constraints on the parameters of the IMEX Butcher tableaux (2.1) must be satisfied for second-order accuracy, fourteen additional constraints must be satisfied for third-order accuracy, and fifty-two additional constraints must be satisfied for fourth-order accuracy. Before proceeding, we thus introduce some significant simplifying assumptions. Following [7] and [17] and the **CN/RKW3** scheme of [4], we synchronize the stages of DIRK and ERK components by imposing $c_i^{\mathrm{I}} = c_i^{\mathrm{E}} = c_i$ for $i = 1, \ldots, s$. We also coordinate the constituent DIRK and ERK components such that $b_i^{\mathrm{I}} = b_i^{\mathrm{E}} = b_i$ for $i = 1, \ldots, s$, as also done in [7] and [17], but which is not satisfied by **CN/RKW3**. Finally, for each stage, a stage-order of one is also imposed such that

$$\sum_{j=1}^{i} a_{i,j}^{\mathrm{I}} = \sum_{j=1}^{i-1} a_{i,j}^{\mathrm{E}} = c_i \quad \text{for} \ \ i = 1, \ldots, s; \tag{2.15}$$

it follows that $c_1 = a_{1,1}^{\mathrm{I}} = a_{1,1}^{\mathrm{E}} = 0$. As a result of these assumptions, the number of constraints on the IMEX parameters [see (2.4)] for second-order accuracy is reduced to just two, the number of constraints for third-order accuracy is reduced to five, and the number of constraints for fourth-order accuracy is reduced to fourteen.

For several of the IMEXRK schemes developed in this chapter, a lower-order embedded scheme is also developed, relaxing the $\hat{b}_i^{\mathrm{I}} = \hat{b}_i^{\mathrm{E}}$ restriction to provide increased freedom during the design phase. As a general guideline, none of the leading-order truncation terms of an embedded scheme should vanish, so that each of these terms will contribute to the error estimate (subject to this restriction, the remaining free parameters of the embedded scheme are then optimized to maximize the magnitude of the leading-order truncation terms). Unfortunately, this is not always achievable; as a result, not all of the schemes developed in this chapter are listed with embedded schemes. For all of the embedded schemes we do report, the DIRK part of the embedded scheme is at least A-stable, which is a property of the embedded scheme recommended by [20]; note, however, that the embedded scheme is not used for time marching, it is only used to adjust the timestep.

The IMEX Butcher tableaux in (2.1) for coordinated pairs of [2R] DIRK and ERK schemes are thus simplified to

$$\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a^{\mathrm{I}}_{2,1} & a^{\mathrm{I}}_{2,2} & & & & \\
c_3 & b_1 & a^{\mathrm{I}}_{3,2} & a^{\mathrm{I}}_{3,3} & & & \\
c_4 & b_1 & b_2 & a^{\mathrm{I}}_{4,3} & a^{\mathrm{I}}_{4,4} & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & b_{s-2} & a^{\mathrm{I}}_{s,s-1} & a^{\mathrm{I}}_{s,s} \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
\hline
 & \hat{b}^{\mathrm{I}}_1 & \hat{b}^{\mathrm{I}}_2 & \cdots & \hat{b}^{\mathrm{I}}_{s-2} & \hat{b}^{\mathrm{I}}_{s-1} & \hat{b}^{\mathrm{I}}_s
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a^{\mathrm{E}}_{2,1} & 0 & & & & \\
c_3 & b_1 & a^{\mathrm{E}}_{3,2} & 0 & & & \\
c_4 & b_1 & b_2 & a^{\mathrm{E}}_{4,3} & 0 & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & b_{s-2} & a^{\mathrm{E}}_{s,s-1} & 0 \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
\hline
 & \hat{b}^{\mathrm{E}}_1 & \hat{b}^{\mathrm{E}}_2 & \cdots & \hat{b}^{\mathrm{E}}_{s-2} & \hat{b}^{\mathrm{E}}_{s-1} & \hat{b}^{\mathrm{E}}_s
\end{array}
\tag{2.16}$$

and the IMEX Butcher tableaux for coordinated pairs of [3R] DIRK and ERK schemes are simplified to

$$\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a^{\mathrm{I}}_{2,1} & a^{\mathrm{I}}_{2,2} & & & & \\
c_3 & a^{\mathrm{I}}_{3,1} & a^{\mathrm{I}}_{3,2} & a^{\mathrm{I}}_{3,3} & & & \\
c_4 & b_1 & a^{\mathrm{I}}_{4,2} & a^{\mathrm{I}}_{4,3} & a^{\mathrm{I}}_{4,4} & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & a^{\mathrm{I}}_{s,s-2} & a^{\mathrm{I}}_{s,s-1} & a^{\mathrm{I}}_{s,s} \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
\hline
 & \hat{b}^{\mathrm{I}}_1 & \hat{b}^{\mathrm{I}}_2 & \cdots & \hat{b}^{\mathrm{I}}_{s-2} & \hat{b}^{\mathrm{I}}_{s-1} & \hat{b}^{\mathrm{I}}_s
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a^{\mathrm{E}}_{2,1} & 0 & & & & \\
c_3 & a^{\mathrm{E}}_{3,1} & a^{\mathrm{E}}_{3,2} & 0 & & & \\
c_4 & b_1 & a^{\mathrm{E}}_{4,2} & a^{\mathrm{E}}_{4,3} & 0 & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\
c_s & b_1 & b_2 & \cdots & a^{\mathrm{I}}_{s,s-2} & a^{\mathrm{E}}_{s,s-1} & 0 \\
\hline
 & b_1 & b_2 & \cdots & b_{s-2} & b_{s-1} & b_s \\
\hline
 & \hat{b}^{\mathrm{E}}_1 & \hat{b}^{\mathrm{E}}_2 & \cdots & \hat{b}^{\mathrm{E}}_{s-2} & \hat{b}^{\mathrm{E}}_{s-1} & \hat{b}^{\mathrm{E}}_s
\end{array}
\tag{2.17}$$

Note also that, as the DIRK component the IMEXRK form considered above has an explicit first stage, its stability function (2.5) may be written

$$\sigma(z^{\mathrm{I}}, z^{\mathrm{E}}) = \frac{1 + \sum_{i=1}^{s} p_i(z^{\mathrm{E}})\,[z^{\mathrm{I}}]^i}{1 + \sum_{i=1}^{s-1} q_i\,[z^{\mathrm{I}}]^i} \quad \text{where} \quad p_i(z^{\mathrm{E}}) = \sum_{j=0}^{s-i} \hat{p}_{i,j}\,[z^{\mathrm{E}}]^j. \tag{2.18}$$

General three-register implementation of [2R] IMEXRK schemes

Note that, if the stiff part of the ODE is linear [that is, if $\mathbf{f}(\mathbf{x}, t) = A\mathbf{x}$] then, denoting the efficient solution of $A\mathbf{x} = \mathbf{b}$ as $A^{-1}\mathbf{b}$, a straightforward implementation of the low-storage IMEXRK scheme in (2.16) that uses three registers[2] of length N to advance from $\mathbf{x} = \mathbf{x}_n$ to $\mathbf{x} = \mathbf{x}_{n+1}$ proceeds as follows:

```
for  $k = 1 : s$
  if  $k == 1$,  $\mathbf{y} \leftarrow \mathbf{x}$,  else
    $\mathbf{y} \leftarrow \mathbf{x} + (a^{\mathrm{I}}_{k,k-1} - b^{\mathrm{I}}_{k-1})\,\Delta t\,\mathbf{z} + (a^{\mathrm{E}}_{k,k-1} - b^{\mathrm{E}}_{k-1})\,\Delta t\,\mathbf{y}$
  end
  $\mathbf{z} = (I - a^{\mathrm{I}}_{k,k}\,\Delta t\,A)^{-1} A\,\mathbf{y}$                                  (2.19)
  $\mathbf{y} \leftarrow \mathbf{g}(\mathbf{y} + a^{\mathrm{I}}_{k,k}\,\Delta t\,\mathbf{z},\ t_n + c^{\mathrm{E}}_k \Delta t)$
  $\mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{I}}_k\,\Delta t\,\mathbf{z} + b^{\mathrm{E}}_k\,\Delta t\,\mathbf{y}$
  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{I}}_k\,\Delta t\,\mathbf{z} + \hat{b}^{\mathrm{E}}_k\,\Delta t\,\mathbf{y}$
end
```

where $\mathbf{z}$ and $\mathbf{y}$ store the implicit and explicit parts of the RHS at each stage, $\mathbf{x}$ is used to advance the solution of the main scheme[3], and $\hat{\mathbf{x}}$ stores the solution of the embedded

[2] That is, in addition to any extra memory required to solve the linear system, which is problem dependent, plus an additional register of length N for the embedded scheme, if adaptive time stepping is implemented.

[3] Note again that $b^{\mathrm{I}}_i = b^{\mathrm{E}}_i = b_i$ for $i = 1, \ldots, s$ for the schemes developed herein, though this property is not shared by **CN/RKW3** (see §2.2).

scheme if adaptive time stepping is implemented. Note that one linear solve of the form $(I - c\,A)^{-1}\mathbf{b}$, one matrix/vector product $A\mathbf{y}$, and one nonlinear function evaluation $\mathbf{g}(\mathbf{y}, t)$ are computed per stage, in addition to various level-1 BLAS (basic linear algebra subroutine) operations. As discussed in §2.1.1, implementation in the case of a nonlinear stiff part is a straightforward extension.

General two-register implementation of [2R] IMEXRK schemes

By applying the matrix inversion lemma: $(\hat{A} + \hat{B}\hat{C}\hat{D})^{-1} = \hat{A}^{-1} - \hat{A}^{-1}\hat{B}(\hat{C}^{-1} + \hat{D}\hat{A}^{-1}\hat{B})^{-1}\hat{D}\hat{A}^{-1}$ with $\hat{A} = \hat{C} = I$, $\hat{D} = A$, and $B = -a^{\mathrm{I}}_{k,k}\,\Delta t$, the algorithm laid out in §2.1.2 may be rewritten in a form that only requires two registers[2] of length N:

```
for   $k = 1 : s$
   if   $k == 1$,   $\mathbf{y} \leftarrow \mathbf{x}$,   else
      $\mathbf{y} \leftarrow \mathbf{x} + (a^{\mathrm{I}}_{k,k-1} - b^{\mathrm{I}}_{k-1})\,\Delta t\,A\,\mathbf{y} + (a^{\mathrm{E}}_{k,k-1} - b^{\mathrm{E}}_{k-1})\,\Delta t\,\mathbf{g}(\mathbf{y},\ t_n + c^{\mathrm{E}}_{k-1}\Delta t)$
   end
   $\mathbf{y} \leftarrow (I - a^{\mathrm{I}}_{k,k}\,\Delta t\,A)^{-1}\mathbf{y}$
   $\mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{I}}_{k}\,\Delta t\,A\,\mathbf{y} + b^{\mathrm{E}}_{k}\,\Delta t\,\mathbf{g}(\mathbf{y},\ t_n + c^{\mathrm{E}}_{k}\Delta t)$
   $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{I}}_{k}\,\Delta t\,A\,\mathbf{y} + \hat{b}^{\mathrm{E}}_{k}\,\Delta t\,\mathbf{g}(\mathbf{y},\ t_n + c^{\mathrm{E}}_{k}\Delta t)$
end
```

(2.20)

In this case, one linear solve of the form $(I - c\,A)^{-1}\mathbf{b}$ and two operations of the form[4] $\mathbf{x} + c\,A\,\mathbf{y} + d\,\mathbf{g}(\mathbf{y}, t)$ are computed per stage, in addition to various level-1 BLAS operations.

[4] When using finite-difference methods, an operation of this form can, with care, usually be performed *in place* in the computer memory using $O(1)$ temporary storage variables; how this is best accomplished, of course, depends on the precise form of A and $\mathbf{g}(\mathbf{y}, t)$. When using spectral methods, such a two-register implementation is generally not available.

However, the storage requirement is reduced from three registers of length N to only two, which is quite significant. In many cases, some of the coefficients in the above algorithm turn out to be zero, so the increased computational cost associated with the extra nonlinear function evaluations and matrix/vector products in this implementation is not as bad as one might initially anticipate, as quantified in §2.7.

General four-register implementation of [3R] IMEXRK schemes

For the development of the stage-order-two schemes **IMEXRKCB3f** and **IMEXRKCB4** in §2.4 and §2.5, the [3R] IMEXRK structure (2.17) will be used to provide increased freedom during the design phase. Such schemes admit the following four-register implementation:

```
for  $k = 1 : s$
  if  $k == 1$,  $\mathbf{y} \leftarrow \mathbf{x}$,  $\mathbf{z}^{\mathrm{I}} = \mathbf{x}$,  $\mathbf{z}^{\mathrm{E}} \leftarrow \mathbf{x}$,  else
    $\mathbf{z}^{\mathrm{E}} \leftarrow \mathbf{y} + a^{\mathrm{E}}_{k,k-1}\, \Delta t\, \mathbf{z}^{\mathrm{E}}$
    if  $k < s$,  $\mathbf{y} \leftarrow \mathbf{x} + (a^{\mathrm{I}}_{k+1,k-1} - b^{\mathrm{I}}_{k-1})\, \Delta t\, \mathbf{z}^{\mathrm{I}} + (a^{\mathrm{E}}_{k+1,k-1} - b^{\mathrm{E}}_{k-1})\, (\mathbf{z}^{\mathrm{E}} - \mathbf{y})/a^{\mathrm{E}}_{k,k-1}$,  end
    $\mathbf{z}^{\mathrm{E}} \leftarrow \mathbf{z}^{\mathrm{E}} + a^{\mathrm{I}}_{k,k-1}\, \Delta t \mathbf{z}^{\mathrm{I}}$
  end                                                                  (2.21)
  $\mathbf{z}^{\mathrm{I}} = (I - a^{\mathrm{I}}_{k,k}\, \Delta t\, A)^{-1} A\, \mathbf{z}^{\mathrm{E}}$
  $\mathbf{z}^{\mathrm{E}} \leftarrow \mathbf{g}(\mathbf{z}^{\mathrm{E}} + a^{\mathrm{I}}_{k,k}\, \Delta t\, \mathbf{z}^{\mathrm{I}},\ t_n + c^{\mathrm{E}}_k \Delta t)$
  $\mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{I}}_k\, \Delta t\, \mathbf{z}^{\mathrm{I}} + b^{\mathrm{E}}_k\, \Delta t\, \mathbf{z}^{\mathrm{E}}$
  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{I}}_k\, \Delta t\, \mathbf{z}^{\mathrm{I}} + \hat{b}^{\mathrm{E}}_k\, \Delta t\, \mathbf{z}^{\mathrm{E}}$
end
```

where $\mathbf{z}^{\mathrm{I}}$ and $\mathbf{z}^{\mathrm{E}}$ store the implicit and explicit parts of the RHS at each stage, $\mathbf{y}$ is a temporary variable which contributes to advance the solution to the next stage, $\mathbf{x}$ is used to advance the solution of the main scheme, and $\hat{\mathbf{x}}$ stores the solution of the embedded scheme if adaptive time stepping is used. As in the three-register implementation of the [2R] scheme, only one linear solve of the form $(I - c\,A)^{-1}\mathbf{b}$, one matrix/vector product, and one nonlinear function evaluation are computed per stage.

General three-register implementation of [3R] IMEXRK schemes

Leveraging matrix inversion lemma as done in §2.1.2, we obtain a general three-register implementation of any [3R] IMEXRK scheme:

```
for  $k = 1 : s$
  if  $k == 1$,   $\mathbf{y} \leftarrow \mathbf{x}$,   $\mathbf{z} \leftarrow \mathbf{x}$,   else
    if  $k < s$
      $\mathbf{z} \leftarrow \mathbf{y} + a^{\mathrm{I}}_{k,k-1}\,\Delta t\,A\,\mathbf{z}$
      $\mathbf{y} \leftarrow A^{-1}\,(\mathbf{z} - \mathbf{y})/(a^{\mathrm{I}}_{k,k-1}\,\Delta t)$
      $\mathbf{z} \leftarrow \mathbf{z} + a^{\mathrm{E}}_{k,k-1}\,\Delta t\,\mathbf{g}(\mathbf{y},\ t_n + c^{\mathrm{E}}_{k-1}\Delta t)$
      $\mathbf{y} \leftarrow \mathbf{x} + (a^{\mathrm{I}}_{k+1,k-1} - b^{\mathrm{I}}_{k-1})\,\Delta t\,A\,\mathbf{y} + (a^{\mathrm{E}}_{k+1,k-1} - b^{\mathrm{E}}_{k-1})\,\Delta t\,\mathbf{g}(\mathbf{y},\ t_n + c^{\mathrm{E}}_{k-1}\Delta t)$
    else                                                                  (2.22)
      $\mathbf{z} \leftarrow \mathbf{y} + a^{\mathrm{I}}_{k,k-1}\,\Delta t\,A\,\mathbf{z} + a^{\mathrm{E}}_{k,k-1}\,\Delta t\,\mathbf{g}(\mathbf{y},\ t_n + c^{\mathrm{E}}_{k-1}\Delta t)$
    end
  end
  $\mathbf{z} \leftarrow (I - a^{\mathrm{I}}_{k,k}\,\Delta t\,A)^{-1}\,\mathbf{z}$
  $\mathbf{x} \leftarrow \mathbf{x} + b^{\mathrm{I}}_{k}\,\Delta t\,A\,\mathbf{z} + b^{\mathrm{E}}_{k}\,\Delta t\,\mathbf{g}(\mathbf{z},\ t_n + c^{\mathrm{E}}_{k}\Delta t)$
  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \hat{b}^{\mathrm{I}}_{k}\,\Delta t\,A\,\mathbf{z} + \hat{b}^{\mathrm{E}}_{k}\,\Delta t\,\mathbf{g}(\mathbf{z},\ t_n + c^{\mathrm{E}}_{k}\Delta t)$
end
```

Note that this algorithm requires the invertibility of the matrix A, a condition that is often true when A arises from the discretization of a PDE. In this case, two linear systems, three matrix/vector products, and three nonlinear function evaluations must be performed per stage (except for the last stage), plus an additional matrix/vector product and one nonlinear function evaluation if the embedded scheme is used for adaptive time stepping.

Finally, note that a (hardware-dependent) trade-off between flops and storage must ultimately be conducted to select between the two-register and three-register implementation of any [2R] scheme, or between the three-register and four-register implementation of any [3R] scheme.

2.2 Two second-order, 2-register IMEXRK schemes

The classical second-order, A-stable **CN/RKW3** method may easily be written in the low-storage IMEXRK Butcher tableaux form (2.16) (albeit with the $b_i^{\mathrm{I}} = b_i^{\mathrm{E}} = b_i$ constraint relaxed) with the four-stage IMEX Butcher tableaux

$$
\begin{array}{c|cccc}
0 & 0 & & & \\
8/15 & 4/15 & 4/15 & & \\
2/3 & 4/15 & 1/3 & 1/15 & \\
1 & 4/15 & 1/3 & 7/30 & 1/6 \\
\hline
 & 4/15 & 1/3 & 7/30 & 1/6
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & & & \\
8/15 & 8/15 & 0 & & \\
2/3 & 1/4 & 5/12 & 0 & \\
1 & 1/4 & 0 & 3/4 & 0 \\
\hline
 & 1/4 & 0 & 3/4 & 0
\end{array}
\tag{2.23}
$$

A DIRK scheme with $c_1 = 0$ and $c_s = 1$ [such as that shown at left in (2.23)] is known as a first-same-as-last (FSAL) scheme. In such a scheme, the implicit part of the last stage of one timestep is precisely the implicit part of the first stage of the next timestep, and thus an FSAL scheme, such as the implicit part of the **CN/RKW3** scheme shown above, actually incorporates only $s - 1$ implicit solves per timestep. Note also that, since $b_s^{\mathrm{E}} = 0$

above, $\mathbf{g}_s$ actually never needs to be computed. Thus, though **CN/RKW3** is written above as a four-stage IMEX Butcher tableaux, a careful implementation of **CN/RKW3** actually incorporates only three implicit stages and three explicit stages per timestep.

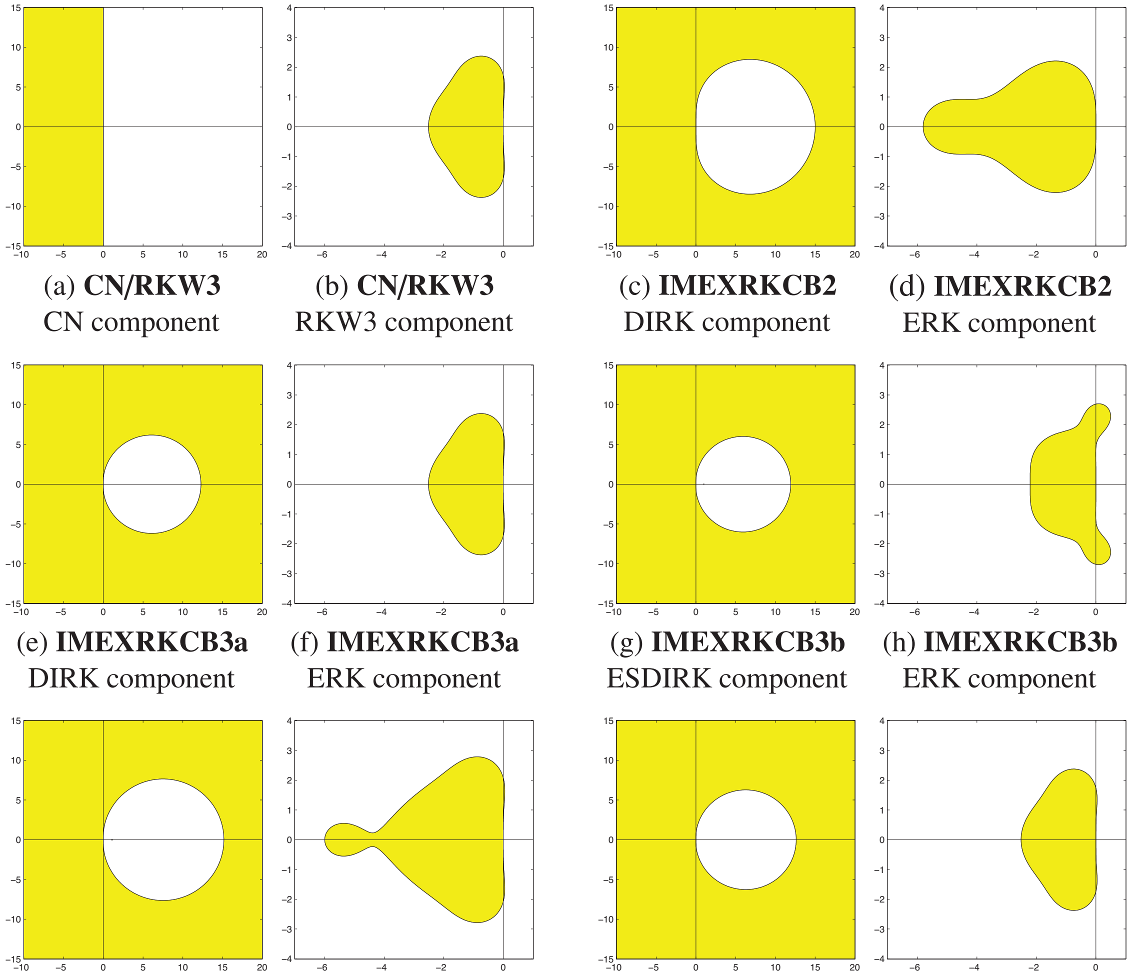
The stability boundaries of the constituent CN and RKW3 schemes of (2.23) are shown in Figures 2.1a-2.1b; the CN scheme, applied over each of three stages, is A stable, and the stability of the RKW3 scheme is that of any third-order, three-stage ERK scheme, with (denoting $z = z^{\mathrm{E}}$) a stability function of

$$\begin{aligned}\sigma^{\mathrm{E}}(z) &= 1 + z\sum_{i=1}^{4} b_i + z^2\sum_{i=1}^{4} b_i\,c_i + z^3\sum_{i,j=1}^{4} b_i\,a^{\mathrm{E}}_{i,j}\,c_j + z^4\sum_{i,j,k=1}^{4} b_i\,a^{\mathrm{E}}_{i,j}\,a^{\mathrm{E}}_{j,k}\,c_k\\ &= 1 + z + z^2/2 + z^3/6,\end{aligned}$$

where, again, $|\sigma^{\mathrm{E}}(z)| \leq 1$ indicates the stability region.

The **CN/RKW3** scheme was initially developed simply by joining together two existing schemes, CN and RKW3, in an IMEXRK fashion. It was, e.g., not designed with the constraints (2.4i)-(2.4n) in mind, and thus leaves significant room for improvement. For example, a remarkably simple second-order [2R] alternative to **CN/RKW3** which

- requires fewer flops per timestep to implement than **CN/RKW3**,
- comes with a first-order embedded scheme, following the guidelines listed in §2.1.2, for adaptive time stepping,
- whose implicit part is L-stable, and
- whose explicit part is both SSP and exhibits much improved stability on the negative real axis as compared to **CN/RKW3**,



(a) **CN/RKW3**
CN component

(b) **CN/RKW3**
RKW3 component

(c) **IMEXRKCB2**
DIRK component

(d) **IMEXRKCB2**
ERK component

(e) **IMEXRKCB3a**
DIRK component

(f) **IMEXRKCB3a**
ERK component

(g) **IMEXRKCB3b**
ESDIRK component

(h) **IMEXRKCB3b**
ERK component

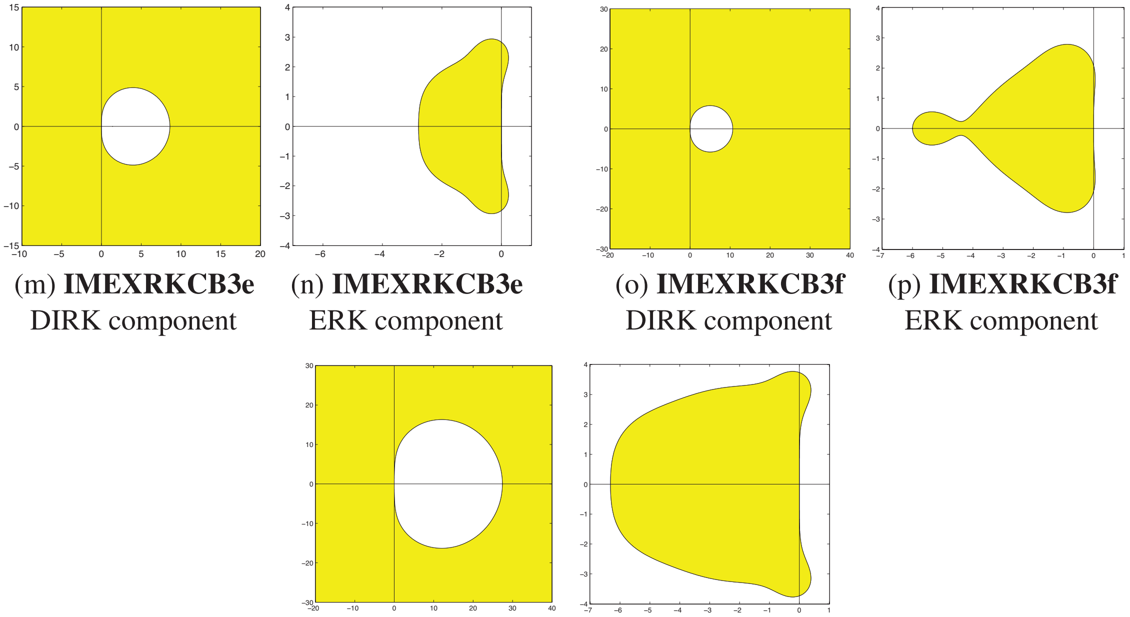
(i) **IMEXRKCB3c**
DIRK component

(j) **IMEXRKCB3c**
ERK component

(k) **IMEXRKCB3d**
DIRK component

(l) **IMEXRKCB3d**
ERK component

Figure 2.1: Stability regions $|\sigma(z)| \leq 1$ for the low-storage IMEXRK schemes considered in this chapter.



(m) **IMEXRKCB3e** DIRK component

(n) **IMEXRKCB3e** ERK component

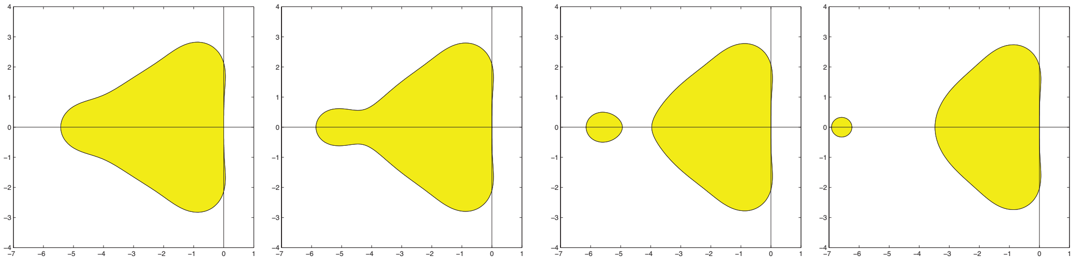
(o) **IMEXRKCB3f** DIRK component

(p) **IMEXRKCB3f** ERK component

(q) **IMEXRKCB4** DIRK component

(r) **IMEXRKCB4** ERK component

Figure 2.1: Stability regions $|\sigma(z)| \leq 1$ for the low-storage IMEXRK schemes considered in this chapter (continued from previous page).



(a) $\delta = 1/50$

(b) $\delta = 1/53$

(c) $\delta = 1/55$

(d) $\delta = 1/60$

Figure 2.2: Stability regions $|\sigma(z)| \leq 1$ for $\sigma(z) = 1 + z + z^2/2 + z^3/6 + \delta z^4$ for various values of δ; note that the case with $\delta = 1/24$ is given in Figure 2.1l, and the case with $\delta = 1/54$ is given in Figure 2.1j.

dubbed **IMEXRKCB2**, is given by[5]

$$\begin{array}{c|ccc} 0 & 0 & & \\ 2/5 & 0 & 2/5 & \\ 1 & 0 & 5/6 & 1/6 \\ \hline & 0 & 5/6 & 1/6 \\ \hline & 0 & 4/5 & 1/5 \end{array} \qquad \begin{array}{c|ccc} 0 & 0 & & \\ 2/5 & 2/5 & 0 & \\ 1 & 0 & 1 & 0 \\ \hline & 0 & 5/6 & 1/6 \\ \hline & 0 & 4/5 & 1/5 \end{array} \tag{2.24}$$

The coefficient for strong stability in (2.10) for this scheme is $c = 1$, which is the maximum possible, as proved in [16]. Note also that the so-called "stiff accuracy" conditions have been imposed on the implicit component of this scheme; that is, we have set $a^{\mathrm{I}}_{s,i} = b_i$ for $i = 1, \ldots, s$. These conditions improve the convergence of such a scheme for the integration of stiff ODEs, as noted in [20] and [21] and described further in §2.6. Moreover, these conditions have the benefit of reducing by one the order of the polynomial in the numerator of the stability function, facilitating the attainment of L-stability [i.e., $\sigma(\infty) = 0$], as we will show in §2.3.3. Applying the stiff accuracy conditions to (2.4a) and (2.15), we obtain $c_s = 1$. Together with the condition $c_1 = 0$, it follows that all IMEX schemes developed herein with DIRK components achieving L-stability via the stiff accuracy conditions, such as (2.24), are FSAL, and thus require only $s - 1$ implicit solves per timestep. This is especially apparent in (2.24), in which the entire first column of the Butcher tableau of the implicit component equals zero. Since this IMEXRK scheme has two implicit stages and three explicit stages per timestep, as a shorthand, we report the scheme as requiring $(2, 3)$ stages per timestep in Table 2.1; the stage requirements of the other schemes developed in this chapter are denoted similarly.

[5] For details on how this scheme was discovered, see §2.3.3, which applies the same techniques used to discover (2.24) to the 3rd-order, 3-stage implicit, 4-stage explicit, L-stable case.

The stability boundaries of the constituent DIRK and ERK components of (2.24) are shown in Figures 2.1c-2.1d.

2.3 Five third-order, 2-register IMEXRK schemes

2.3.1 A $(2,3)$-stage, strongly A-stable scheme

As suggested by (2.24), to streamline the implementation, we can suppress the first stage of the DIRK scheme by imposing $b_1 = a^{\mathrm{I}}_{2,1} = 0$. Following this simplification, the entire first column of the DIRK scheme is zero, thus leading to a scheme with $s-1$ implicit stages and s explicit stages. In the $s = 3$ case, the IMEXRK Butcher tableaux take the general form

$$\begin{array}{c|ccc} 0 & 0 & & \\ c_2 & 0 & a^{\mathrm{I}}_{2,2} & \\ c_3 & 0 & a^{\mathrm{I}}_{3,2} & a^{\mathrm{I}}_{3,3} \\ \hline & 0 & b_2 & b_3 \end{array} \qquad \begin{array}{c|ccc} 0 & 0 & & \\ c_2 & a^{\mathrm{E}}_{2,1} & 0 & \\ c_3 & 0 & a^{\mathrm{E}}_{3,2} & 0 \\ \hline & 0 & b_2 & b_3 \end{array} \tag{2.25a}$$

To achieve third-order accuracy, after imposing stage-order-one conditions on both implicit and explicit part, we arrive at five nonlinear equations in five parameters:

$$b_2 + b_3 - 1 = 0, \quad b_2 c_2 + b_3 c_3 - 1/2 = 0, \quad b_2 c_2^2 + b_3 c_3^2 - 1/3 = 0, \quad b_3 c_2 c_3 - 1/6 = 0,$$
$$b_2 c_2^2 + b_3 a^{\mathrm{I}}_{3,3} c_3 + b_3 (c_3 - a^{\mathrm{I}}_{3,3}) c_2 - 1/6 = 0.$$

This system of nonlinear equations has a single closed-form solution among the real numbers. Defining c_2 as the sole real root of the polynomial $18c_2^3 - 27c_2^2 + 12c_2 - 2 = 0$, closed-form expressions for the parameters of this scheme, dubbed **IMEXRKCB3a**, are:

$$c_2 = a^{\mathrm{I}}_{2,2} = a^{\mathrm{E}}_{2,1} = \left(27 + \sqrt[3]{2187 - 1458\sqrt{2}} + 9\sqrt[3]{3 + 2\sqrt{2}}\right)/54,$$

$$c_3 = a^{\mathrm{E}}_{3,2} = c_2/(6c_2^2 - 3c_2 + 1), \quad b_2 = (3c_2 - 1)/(6c_2^2), \quad b_3 = (6c_2^2 - 3c_2 + 1)/(6c_2^2),$$

$$a^{\mathrm{I}}_{3,3} = (1/6 - b_2c_2^2 - b_3c_2c_3)/[b_3(c_3 - c_2)], \quad a^{\mathrm{I}}_{3,2} = a^{\mathrm{I}}_{3,3} - c_3 \tag{2.25b}$$

The stability boundaries of the constituent DIRK and ERK components of (2.25) are shown in Figures 2.1e-2.1f; note that the stability boundary of the 3-stage, 3rd-order ERK component necessarily coincides with that of RKW3. As compared with (2.24), which has a Butcher tableaux of the same structure, the present scheme sacrifices *L*-stability of its DIRK component in order to achieve third-order accuracy.

It is instructive to note that, even after removing the assumption $b_1 = 0$, it is not possible to achieve *L*-stability of the DIRK component of a third-order IMEXRK scheme of the general form given in (2.16) using only three stages due to a conflict that arises in the $\tau^{\mathrm{IE}(3)} = 0$ constraints (2.4j)-(2.4n), as observed previously by [9]. For this reason, the remainder of this chapter explores four-stage schemes of an analogous form for third-order accuracy.

2.3.2 A $(3, 4)$-stage, strongly *A*-stable scheme with ESDIRK implicit part

Extending the simplifying assumptions used in the previous section to a four-stage two-register scheme, by taking $b_1 = b_2 = 0$, and additionally imposing equal values for the diagonal terms of the implicit scheme (that is, $a^{\mathrm{I}}_{i,i} = \gamma$ for $i = 2, 3, 4$), the Butcher tableaux

(2.16) reduce to:

$$\begin{array}{c|cccc}
0 & 0 & & & \\
c_2 & 0 & \gamma & & \\
c_3 & 0 & a^{\mathrm{I}}_{3,2} & \gamma & \\
c_4 & 0 & 0 & a^{\mathrm{I}}_{4,3} & \gamma \\
\hline
 & 0 & 0 & b_3 & b_4
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & & & \\
c_2 & a^{\mathrm{E}}_{2,1} & 0 & & \\
c_3 & 0 & a^{\mathrm{E}}_{3,2} & 0 & \\
c_4 & 0 & 0 & a^{\mathrm{E}}_{4,3} & 0 \\
\hline
 & 0 & 0 & b_3 & b_4
\end{array}
\tag{2.26a}$$

After imposing stage-order-one conditions, determining all the parameters requires the solution of the following system of five nonlinear equations:

$$\begin{aligned}
&b_3 + b_4 - 1 = 0,\\
&b_3c_3 + b_4c_4 - 1/2 = 0,\\
&b_3c_3^2 + b_4c_4^2 - 1/3 = 0,\\
&b_3c_2c_3 + b_4c_3c_4 - 1/6 = 0,\\
&b_3c_2c_3 + b_3c_2c_3 - b_3c_2^2 + b_4c_2c_4 + b_4c_3c_4 - b_4c_2c_3 - 1/6 = 0.
\end{aligned}$$

This system of equations has two closed-form solutions, one of which does not lead to an A-stable scheme, and the other of which, dubbed **IMEXRKCB3b**, is given by

$$\begin{gathered}
\gamma = c_2 = a^{\mathrm{E}}_{2,1} = 1/2 + \sqrt{3}/6, \quad c_3 = a^{\mathrm{E}}_{3,2} = 1/2 - \sqrt{3}/6, \quad c_4 = a^{\mathrm{E}}_{4,3} = 1/2 + \sqrt{3}/6,\\
a^{\mathrm{I}}_{3,2} = -\sqrt{3}/3, \quad a^{\mathrm{I}}_{4,3} = 0, \quad b_3 = b_4 = 1/2.
\end{gathered}\tag{2.26b}$$

The stability boundaries of the constituent DIRK and ERK components of (2.26) are shown in Figures 2.1g-2.1h. This scheme again achieves strong A-stability of its DIRK component while, as compared with **IMEXRKCB3a**, slightly extending the limit of stability of the ERK component in the imaginary directions, and slightly reducing the limit of stability of the ERK component in the negative real direction.

Imposing the nonzero diagonal terms of the DIRK scheme to be equal [a simplification resulting in what is usually called an Explicit-first-stage Singly Diagonally Implicit Runge Kutta (ESDIRK) method] facilitates use of the LU decomposition of the matrix $(I - \gamma \Delta t A)$ to simplify all of the implicit solves. This can significantly reduce the number of flops needed for the implicit solves, but may increase the number of registers required by the code; whether or not use of the LU decomposition in the implicit solves represents an overall speedup of the simulation depends both on the structure and size of A and the computational hardware being used.

2.3.3 Three $(3, 4)$-stage, L-stable schemes

The simplifying assumptions considered in the previous section again facilitated a closed-form expression of the parameters, but prevented the DIRK component from achieving L-stability. In order to achieve L-stability of the DIRK component, as noted previously, a useful simplifying assumption is the "stiff accuracy" conditions $a_{s,i} = b_i$ for $i = 1, \ldots, s$ [and hence, by (2.4a) and (2.15), $c_s = 1$]. Taking $s = 4$ and defining $a^{\mathrm{I}}_{i,i} = \alpha_i$ for $i = 2, 3$, the Butcher tableaux (2.16) reduce to the following form (with, again, an FSAL implicit part):

$$
\begin{array}{c|cccc}
0 & 0 & & & \\
c_2 & a^{\mathrm{I}}_{2,1} & a^{\mathrm{I}}_{2,2} & & \\
c_3 & b_1 & a^{\mathrm{I}}_{3,2} & a^{\mathrm{I}}_{3,3} & \\
1 & b_1 & b_2 & b_3 & b_4 \\
\hline
 & b_1 & b_2 & b_3 & b_4 \\
\hline
 & \hat{b}^{\mathrm{I}}_1 & \hat{b}^{\mathrm{I}}_2 & \hat{b}^{\mathrm{I}}_3 & \hat{b}^{\mathrm{I}}_4
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & & & \\
c_2 & a^{\mathrm{E}}_{2,1} & 0 & & \\
c_3 & b_1 & a^{\mathrm{E}}_{3,2} & 0 & \\
1 & b_1 & b_2 & a^{\mathrm{E}}_{4,3} & 0 \\
\hline
 & b_1 & b_2 & b_3 & b_4 \\
\hline
 & \hat{b}^{\mathrm{E}}_1 & \hat{b}^{\mathrm{E}}_2 & \hat{b}^{\mathrm{E}}_3 & \hat{b}^{\mathrm{E}}_4
\end{array}
\tag{2.27}
$$

In order to impose third-order accuracy, five order constraints must again be imposed. To achieve L-stability of the DIRK component, a further simplification of (2.27) is motivated. To understand this simplification, we may rewrite the stability function of the scheme as a rational function of z^{I} and z^{E}, as suggested by (2.5) and (2.18), as

$$\sigma(z^{\mathrm{I}},\, z^{\mathrm{E}}) = \frac{1 + \sum_{i=1}^{2} p_i(z^{\mathrm{E}})\,[z^{\mathrm{I}}]^i + \left(\hat{p}_{3,0} + \hat{p}_{3,1} z^{\mathrm{E}}\right)\,[z^{\mathrm{I}}]^3 + \hat{p}_{4,0}\,[z^{\mathrm{I}}]^4}{1 + \sum_{i=1}^{s-1} q_i\,[z^{\mathrm{I}}]^i},$$

where the p_i, $\hat{p}_{i,j}$, and q_i are polynomials in the Butcher tableaux parameters. Due to stiff accuracy, $\hat{p}_{4,0} = 0$; thus, in order to impose L-stability of the DIRK component [i.e., $\lim_{z^{\mathrm{I}}\to\infty} \sigma(z^{\mathrm{I}},\, z^{\mathrm{E}}) = 0$], it is sufficient to impose that $q_3 = a^{\mathrm{I}}_{2,2}\, a^{\mathrm{I}}_{3,3}\, b_4 \neq 0$ and

$$\begin{aligned}
\tau_1^{L\text{-stab}} = \hat{p}_{3,0} = {} & -a^{\mathrm{I}}_{2,2}\, a^{\mathrm{I}}_{3,3}\, b_1 - a^{\mathrm{I}}_{2,2}\, a^{\mathrm{I}}_{3,3}\, b_2 - a^{\mathrm{I}}_{2,2}\, a^{\mathrm{I}}_{3,3}\, b_3 + a^{\mathrm{I}}_{3,3}\, b_2\, c_2 \\
& + a^{\mathrm{I}}_{3,3}\, b_3\, c_2 + b_1\, b_3\, c_2 + a^{\mathrm{I}}_{2,2}\, b_3\, c_3 - b_3\, c_2\, c_3 = 0, \qquad \text{(A)} \\
\tau_2^{L\text{-stab}} = \hat{p}_{3,1} = {} & -a^{\mathrm{I}}_{2,2}\, a^{\mathrm{I}}_{3,3}\, b_4 + a^{\mathrm{I}}_{3,3}\, b_4\, c_2 + b_1\, b_4\, c_2 - a^{\mathrm{I}}_{3,3}\, b_1\, b_4\, c_2 - b_1^2\, b_4\, c_2 - b_1\, b_2\, b_4\, c_2 \\
& + a^{\mathrm{I}}_{2,2}\, b_4\, c_3 - a^{\mathrm{I}}_{2,2}\, b_1\, b_4\, c_3 - a^{\mathrm{I}}_{2,2}\, b_2\, b_4\, c_3 - b_4\, c_2\, c_3 + b_1\, b_4\, c_2\, c_3 + b_2 b_4\, c_2\, c_3 = 0. \\
& \qquad \text{(B)}
\end{aligned}$$

As noted in [7] and [20], suppressing the first column of the DIRK component, by imposing $b_1 = 0 = a^{\mathrm{I}}_{2,1} = 0$ in (2.27), together with stiff-accuracy condition, satisfies both (A) and (B) identically; we thus incorporate these additional simplifications in the two subsections that follow. Notice that in the full-storage setting this strategy is *not* recommended, as it sacrifices $s - 1$ degrees of freedom. For a [2R] scheme, however, only two degrees of freedom are sacrificed to enforce these two equations, and thereby gain L-stability.

Maximizing the extent of stability of the ERK component over the negative real axis

A final (sixth) constraint is obtained by maximizing the stability envelope of the ERK component over the negative real axis. Using Cramer's rule, we may rewrite the

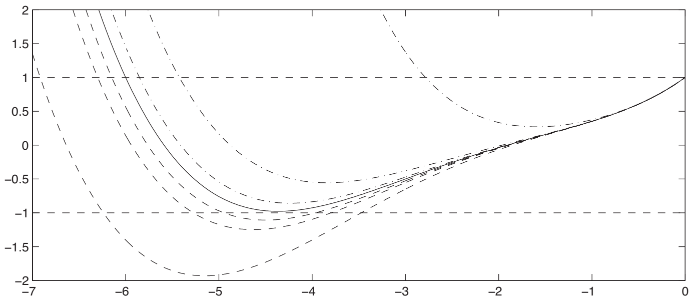


Figure 2.3: The real value of $\sigma(z) = 1 + z + z^2/2 + z^3/6 + \delta\, z^4$ for real, negative values of z and various values of δ: (dashed) $\delta = 1/60,\ 1/56,\ 1/55$; (solid) $\delta = 1/54$; (dot-dashed) $\delta = 1/53,\ 1/50,\ 1/24$. See also Figure 2.2.

stability function of the third-order, four-stage ERK component as

$$\sigma^{\mathrm{E}}(z;\delta) = 1 + z\,\mathbf{b}^T(I - z\,A^{\mathrm{E}})^{-1}\mathbf{e} = 1 + z + z^2/2 + z^3/6 + \delta\, z^4, \quad \text{where} \quad \delta = \sum\nolimits_{i,j,k=1}^{4} b_i\, a^{\mathrm{E}}_{i,j}\, a^{\mathrm{E}}_{j,k}\, c_k.$$

For z on the negative real axis, the stability region $|\sigma^{\mathrm{E}}(z;\delta)| \le 1$ is defined by the two conditions

$$-1 \le 1 + z + z^2/2 + z^3/6 + \delta\, z^4 \le 1.$$

Plots of $\sigma^{\mathrm{E}}(z;\delta)$ for $-7 \le z \le 0$ and various values of δ are given in Figure 2.3. For

$$\delta > \delta_{\mathrm{crit}} = \Big(139 - 5255/\sqrt[3]{-210253 + 60928\sqrt{51}} + \sqrt[3]{-210253 + 60928\sqrt{51}}\Big)/6144 = 0.0184557,$$

the condition $-1 \le \sigma^{\mathrm{E}}(z;\delta)$ is satisfied everywhere in this interval; we thus choose $\delta = 1/54 = 0.0185 > \delta_{\mathrm{crit}}$, which gives $|\sigma^{\mathrm{E}}(z)| \le 1$ for $-6.00 < z < 0$, as larger values of δ reduce the extent of stability (see Figures 2.2 and 2.3).

Parametric variation reveals that the extent of the stability region along the imaginary axis is relatively insensitive to changes in δ. Among the third-order, four-stage IMEXRK scheme available in literature, the one with the widest stability region of the ERK part, which is the (full-storage) **ARK3(2)4L[2R]SA** scheme developed in [7], has a maximum extent along the negative real axis which is $\sim$ 40% *less* than that of that of the present scheme, and a maximum extent along the imaginary axis which is only $\sim$ 5% greater than that of the present scheme; the stability characteristics of the present scheme are thus seen to be quite competitive.

Thus, in order to determine the parameters of the Butcher tableaux, we impose our final (sixth) constraint as

$$\tau^{\delta=1/54} = \sum\nolimits_{i,j,k=1}^{4} b_i\, a^{\mathrm{E}}_{i,j}\, a^{\mathrm{E}}_{j,k}\, c_k - 1/54 = 0. \tag{C}$$

The complete solution of this set of six nonlinear constraint equations has been obtained using Mathematica [22]. The scheme associated to such solution, dubbed **IMEXRKCB3c**, is given by (2.27) with

$$\begin{gathered} a^{\mathrm{I}}_{2,2} = \frac{3375509829940}{4525919076317}, \quad a^{\mathrm{I}}_{3,2} = -\frac{11712383888607531889907}{32694570495602105556248}, \quad a^{\mathrm{I}}_{3,3} = \frac{566138307881}{912153721139}, \\ b_1 = 0, \quad b_2 = \frac{673488652607}{2334033219546}, \quad b_3 = \frac{493801219040}{853653026979}, \quad b_4 = \frac{184814777513}{1389668723319}, \\ c_2 = a^{\mathrm{E}}_{2,1} = \frac{3375509829940}{4525919076317}, \quad c_3 = a^{\mathrm{E}}_{3,2} = \frac{272778623835}{1039454778728}, \quad a^{\mathrm{I}}_{4,3} = \frac{1660544566939}{2334033219546}; \end{gathered} \tag{2.28a}$$

the associated second-order embedded scheme has the following coefficients:

$$\begin{gathered} \hat{b}^{\mathrm{I}}_1 = 0, \quad \hat{b}^{\mathrm{I}}_2 = \frac{366319659506}{1093160237145}, \quad \hat{b}^{\mathrm{I}}_3 = \frac{270096253287}{480244073137}, \quad \hat{b}^{\mathrm{I}}_4 = \frac{104228367309}{1017021570740}, \\ \hat{b}^{\mathrm{E}}_1 = \frac{449556814708}{1155810555193}, \quad \hat{b}^{\mathrm{E}}_2 = 0, \quad \hat{b}^{\mathrm{E}}_3 = \frac{210901428686}{1400818478499}, \quad \hat{b}^{\mathrm{E}}_4 = \frac{480175564215}{1042748212601}. \end{gathered} \tag{2.28b}$$

The stability boundaries of the constituent DIRK and ERK components are shown in Figures 2.1i-2.1j. This scheme is SSP under the condition (2.10) with $c = 0.7027915$. This result can be improved up to $c = 0.7703947$, which is achieved by replacing condition (C)

with

$$\tau^{\delta=0} = \sum\nolimits_{i,j,k=1}^{4} b_i\, a^{\mathrm{E}}_{i,j}\, a^{\mathrm{E}}_{j,k}\, c_k - 0 = 0. \tag{C'}$$

This constraint does not lead to an IMEXRK scheme with an L-stable implicit component; we thus instead choose a small positive δ, thus ensuring L-stability and a nearly optimal value c for strong stability. Choosing $\delta = 1/10000$ results in a scheme, dubbed **IMEXRKCB3d**, given by (2.27) with

$$\begin{gathered}
a^{\mathrm{I}}_{2,2} = \frac{418884414754}{469594081263}, \quad a^{\mathrm{I}}_{3,2} = -\frac{304881946513433262434901}{718520734375438559540570}, \quad a^{\mathrm{I}}_{3,3} = \frac{684872032315}{962089110311}, \\
b_1 = 0, \quad b_2 = \frac{355931813527}{1014712533305}, \quad b_3 = \frac{709215176366}{1093407543385}, \quad b_4 = \frac{755675305}{1258355728177}, \\
c_2 = a^{\mathrm{E}}_{2,1} = \frac{418884414754}{469594081263}, \quad c_3 = a^{\mathrm{E}}_{3,2} = \frac{214744852859}{746833870870}, \quad a^{\mathrm{E}}_{4,3} = \frac{658780719778}{1014712533305};
\end{gathered} \tag{2.29a}$$

the associated second-order embedded scheme has the following coefficients:

$$\begin{gathered}
\hat{b}^{\mathrm{I}}_1 = 0, \quad \hat{b}^{\mathrm{I}}_2 = \frac{226763370689}{646029759300}, \quad \hat{b}^{\mathrm{I}}_3 = \frac{1496839794860}{2307829317197}, \quad \hat{b}^{\mathrm{I}}_4 = \frac{353416193}{889746336234}, \\
\hat{b}^{\mathrm{E}}_1 = \frac{1226988580973}{2455716303853}, \quad \hat{b}^{\mathrm{E}}_2 = 0, \quad \hat{b}^{\mathrm{E}}_3 = \frac{827818615}{1665592077861}, \quad \hat{b}^{\mathrm{E}}_4 = \frac{317137569431}{634456480332}.
\end{gathered} \tag{2.29b}$$

The coefficient for strong stability in this case is $c = 0.7701444$. The stability boundaries of the associated DIRK and ERK components are shown in Figures 2.1k-2.1l. Since δ is chosen close to zero, the stability region of the ERK component closely resembles that of a third-order three-stage explicit Runge-Kutta scheme.

Maximizing accuracy of the ERK component

An alternative third-order four-stage 2-register L-stable strategy, with closed-form parameter values and improved accuracy, is given by replacing the final constraint, (C),

with

$$\tau^{\delta=1/24} = \sum\nolimits_{i,j,k=1}^{4} b_i a^{\mathrm{E}}_{i,j} a^{\mathrm{E}}_{j,k} c_k - 1/24 = 0, \tag{C”}$$

which sets to zero one of the fourth-order truncation terms for the explicit component. This results in a scheme, dubbed **IMEXRKCB3e**, given by

$$\begin{array}{c|cccc} 0 & 0 & & & \\ 1/3 & 0 & 1/3 & & \\ 1 & 0 & 1/2 & 1/2 & \\ 1 & 0 & 3/4 & -1/4 & 1/2 \\ \hline & 0 & 3/4 & -1/4 & 1/2 \end{array} \qquad \begin{array}{c|cccc} 0 & 0 & & & \\ 1/3 & 1/3 & 0 & & \\ 1 & 0 & 1 & 0 & \\ 1 & 0 & 3/4 & 1/4 & 0 \\ \hline & 0 & 3/4 & -1/4 & 1/2 \end{array} \tag{2.30}$$

A second-order embedded scheme having all third-order truncation terms nonzero could not be achieved because of assumption (C”). The stability boundaries of the constituent DIRK and ERK components are shown in Figures 2.1m-2.1n; **IMEXRKCB3e** has improved accuracy but reduced stability on the negative real axis for the ERK component, as compared with **IMEXRKCB3c**. In particular, because of (C”), the stability region for the ERK part coincides with the stability region of a standard 4-stage fourth-order explicit RK scheme.

2.4 A third-order, 3-register, 4-stage, *L*-stable scheme

All of the schemes so-far described have stage-order one for both the implicit and explicit components. It is well-known in the literature (see [21]) that this limits the order of convergence of such methods when applied to stiff ODEs. In particular, if the stiffness is so high that the ODE turns into an index-1 DAE, some variables convert from differential to algebraic and their convergence rate is determined by the stage-order of the method. For this reason, an attempt has been made to improve the stage-order of the implicit scheme, as done in [7]. In this way, when the DIRK scheme is employed alone, a better convergence will be observed during the integration of a stiff ODE, as we will show in §2.6.

Hence, after imposing the same b_i and c_i over the explicit and implicit components and stiff accuracy for the implicit component as done previously, we impose the stage-order-two condition for the implicit component, that is:

$$\sum_{j=1}^{s} a^{\mathrm{I}}_{i,j}\, c_j = c_i^2/2, \quad i = 2,\, 3,\, \ldots,\, s-1. \tag{2.31}$$

With these constraints, $\tau_2^{(3)\mathrm{I}} = 0$ in (2.4d) is automatically satisfied. Hence, we must only impose four constraints for third-order accuracy, two for L-stability, $2(s-2)$ constraints for stage-order two for the implicit component, and $(s-1)$ constraints for stage-order one for the explicit component. We also impose $c_1 = 0$ and $c_4 = 1$ for FSAL structure. Considering a four-stage three-register scheme,

$$\begin{array}{c|cccc}
0 & 0 & & & \\
c_2 & a^{\mathrm{I}}_{2,1} & a^{\mathrm{I}}_{2,2} & & \\
c_3 & a^{\mathrm{I}}_{3,1} & a^{\mathrm{I}}_{3,2} & a^{\mathrm{I}}_{3,3} & \\
1 & b_1 & b_2 & b_3 & b_4 \\
\hline
 & b_1 & b_2 & b_3 & b_4 \\
\hline
 & \hat{b}^{\mathrm{I}}_1 & \hat{b}^{\mathrm{I}}_2 & \hat{b}^{\mathrm{I}}_3 & \hat{b}^{\mathrm{I}}_4
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & & & \\
c_2 & a^{\mathrm{E}}_{2,1} & 0 & & \\
c_3 & a^{\mathrm{E}}_{3,1} & a^{\mathrm{E}}_{3,2} & 0 & \\
1 & b_1 & a^{\mathrm{E}}_{4,2} & a^{\mathrm{E}}_{4,3} & 0 \\
\hline
 & b_1 & b_2 & b_3 & b_4 \\
\hline
 & \hat{b}^{\mathrm{E}}_1 & \hat{b}^{\mathrm{E}}_2 & \hat{b}^{\mathrm{E}}_3 & \hat{b}^{\mathrm{E}}_4
\end{array} \tag{2.32a}$$

after these constraints are imposed, we are left with three degrees of freedom. We choose the constraint (C) to maximize the extent of the stability region of the explicit component on the negative real axis, and perform a parametric variation over the coefficients c_2 and c_3, the remaining two degrees of freedom, between 0 and 1 in order to identify an IMEXRK scheme with coefficients of the Butcher tableaux within the interval $[-5, 5]$, L-stability of the implicit part over the entire LHP, and minimum truncation error, defined, following [7], as

$$A^{(q+1)} = \sqrt{\sum\nolimits_i \left(\tau_i^{(q+1)\mathrm{I}}\right)^2 + \sum\nolimits_i \left(\tau_i^{(q+1)\mathrm{E}}\right)^2 + \sum\nolimits_i \left(\tau_i^{(q+1)\mathrm{IE}}\right)^2}. \tag{2.32b}$$

where q is the order of accuracy of the Runge-Kutta scheme, in this case equal to 3. [The same definition is used in Table 1 to compare the truncation error of the various schemes considered.]

This approach is convenient, as the constraint equations depending on both b_i and c_i become linear in b_i, which allows a significant simplification of the corresponding optimization problem. Note that all of the schemes developed in [7] follow this approach. In the present case, this strategy leads, for each pair (c_2, c_3), to a set of solutions which depend on the roots of a fifth-order polynomial. Among these, only three are real, and only one of these gives an L-stable solution[6]. The resulting scheme, dubbed **IMEXRKCB3f**, is obtained for $c_2 = 49/50$ and $c_3 = 1/25$. The other parameter values are:

$$\begin{gathered}
a^{\mathrm{I}}_{3,1} = -\frac{785157464198}{1093480182337}, \quad a^{\mathrm{I}}_{3,2} = -\frac{30736234873}{978681420651}, \quad a^{\mathrm{I}}_{3,3} = \frac{983779726483}{1246172347126}, \\
a^{\mathrm{E}}_{3,1} = \frac{13244205847}{647648310246}, \quad a^{\mathrm{E}}_{3,2} = \frac{13419997131}{686433909488}, \\
a^{\mathrm{E}}_{4,2} = \frac{231677526244}{1085522130027}, \quad a^{\mathrm{E}}_{4,3} = \frac{3007879347537}{683461566472}, \\
b_1 = -\frac{2179897048956}{603118880443}, \quad b_2 = \frac{99189146040}{891495457793}, \\
b_3 = \frac{6064140186914}{1415701440113}, \quad b_4 = \frac{146791865627}{668377518349},
\end{gathered} \tag{2.32c}$$

and $a^{\mathrm{I}}_{2,1} = a^{\mathrm{I}}_{2,2} = c_2/2$ and $a^{\mathrm{E}}_{2,1} = c_2$ from stage-order conditions. The scheme is not SSP.

[6] The other solutions give a stability region which does not cover the entire LHP; note that this is not in contradiction with the way we have imposed stability on the scheme during the optimization of the coefficients, since we only impose the behavior of the stability function at infinity, then check the boundary of the resulting stability region only after all the parameters of the scheme have been determined.

The associated second-order embedded scheme is given by:

$$\hat{b}_1^{\mathrm{I}} = 0, \quad \hat{b}_2^{\mathrm{I}} = \frac{337712514207}{759004992869}, \quad \hat{b}_3^{\mathrm{I}} = \frac{311412265155}{608745789881}, \quad \hat{b}_4^{\mathrm{I}} = \frac{52826596233}{1214539205236},$$
$$\hat{b}_1^{\mathrm{E}} = 0, \quad \hat{b}_2^{\mathrm{E}} = 0, \quad \hat{b}_3^{\mathrm{E}} = \frac{25}{48}, \quad \hat{b}_4^{\mathrm{E}} = \frac{23}{48}. \tag{2.32d}$$

The stability boundaries of the DIRK and ERK components are shown in Figures 2.1o-2.1p. Notice that the stability region of the explicit component coincides with that of **IMEXRKCB3c**.

2.5 A fourth-order, 3-register, 6-stage, *L*-stable scheme

Solving the nonlinear system of equations arising from the imposition of the fourth-order accuracy constraints is a difficult task. For this reason, stage-order conditions higher than one are usually imposed, as pointed out in [20]. These conditions simplify the search for a solution by significantly reducing the nonconvexity of the corresponding optimization problem. For this reason, after imposing the same b_i and c_i over the explicit and implicit components and stiff accuracy for the implicit component, we require stage-order two for the implicit component[7]. We also again impose $c_1 = 0$ and $c_6 = 1$ for FSAL structure. This reduces the number of nonlinear equations from fourteen, i.e. one for first order, one for second order, three for third order, and nine for fourth order, to only ten, to which we have to add two constraints for *L*-stability, $2(s-2)$ constraints for stage-order two for the implicit component and $(s-1)$ constraints for stage-order one for the explicit component.

[7] Note that, even if it were desired to impose the same stage-order for both implicit and explicit components, in order to improve algebraic variable accuracy, this is not possible, as the low-storage structure used here removes the necessary degrees of freedom to impose such a condition.

Leveraging a six-stage three-register IMEXRK scheme, i.e.

$$
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a^{\mathrm{I}}_{2,1} & a^{\mathrm{I}}_{2,2} & & & & \\
c_3 & a^{\mathrm{I}}_{3,1} & a^{\mathrm{I}}_{3,2} & a^{\mathrm{I}}_{3,3} & & & \\
c_4 & b_1 & a^{\mathrm{I}}_{4,2} & a^{\mathrm{I}}_{4,3} & a^{\mathrm{I}}_{4,4} & & \\
c_5 & b_1 & b_2 & a^{\mathrm{I}}_{5,3} & a^{\mathrm{I}}_{5,4} & a^{\mathrm{I}}_{5,5} & \\
1 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \\
\hline
 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 & \hat{b}_5 & \hat{b}_6
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a^{\mathrm{E}}_{2,1} & 0 & & & & \\
c_3 & a^{\mathrm{E}}_{3,1} & a^{\mathrm{E}}_{3,2} & 0 & & & \\
c_4 & b_1 & a^{\mathrm{E}}_{4,2} & a^{\mathrm{E}}_{4,3} & 0 & & \\
c_5 & b_1 & b_2 & a^{\mathrm{E}}_{5,3} & a^{\mathrm{E}}_{5,4} & 0 & \\
1 & b_1 & b_2 & b_3 & a^{\mathrm{E}}_{6,4} & a^{\mathrm{E}}_{6,5} & 0 \\
\hline
 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & \hat{b}_4 & \hat{b}_5 & \hat{b}_6
\end{array}
\qquad (2.33a)
$$

we have 30 degrees of freedom to satisfy 25 constraints. [For the embedded scheme, the coordination assumption $\hat{b}^{\mathrm{E}} = \hat{b}^{\mathrm{I}} = \hat{b}$ is again imposed, which proves to provide sufficient freedom in the search for a solution.] As in §2.4, we again perform a (tedious) parametric variation over the coefficients c_2, c_3, c_4, and c_5 in the range [0, 1]. The last degree of freedom is taken as one of the diagonal terms of the Butcher tableau of the implicit part (we select $a^{\mathrm{I}}_{5,5}$), which is varied in the range [0, 1/2] in order to minimize the truncation error (2.32b). With this approach, it is possible to numerically solve the nonlinear systems arising during the IMEXRK scheme design phase. In particular, 114 solutions are found for each quintuplet (c_2, c_3, c_4, c_5, $a^{\mathrm{I}}_{5,5}$). Among these, over half have imaginary coefficients, and are therefore discarded immediately. Among of the remaining solutions, only a few satisfy L-stability of the implicit part, and have coefficients in the range [−5, 5]. Among the schemes that survived this initial downselection, we have selected the one offering the smallest truncation error while still exhibiting a large extent of the stability region of the explicit part on the negative real axis. It has been found that the set

$$c_2 = 1/4, \quad c_3 = 3/4, \quad c_4 = 3/8, \quad c_5 = 1/2, \quad a^{\mathrm{I}}_{5,5} = 1/2 \tag{2.33b}$$

gives the best results. The scheme thus obtained, dubbed **IMEXRKCB4** is given by:

$$\begin{gathered}
a^{\mathrm{I}}_{3,1} = \frac{216145252607}{961230882893}, \quad a^{\mathrm{I}}_{3,2} = \frac{257479850128}{1143310606989}, \quad a^{\mathrm{I}}_{3,3} = \frac{30481561667}{101628412017}, \\
a^{\mathrm{I}}_{4,2} = -\frac{381180097479}{1276440792700}, \quad a^{\mathrm{I}}_{4,3} = -\frac{54660926949}{461115766612}, \quad a^{\mathrm{I}}_{4,4} = \frac{344309628413}{552073727558}, \\
a^{\mathrm{I}}_{5,3} = -\frac{100836174740}{861952129159}, \quad a^{\mathrm{I}}_{5,4} = -\frac{250423827953}{1283875864443}, \\
a^{\mathrm{E}}_{3,1} = \frac{153985248130}{1004999853329}, \quad a^{\mathrm{E}}_{3,2} = \frac{902825336800}{1512825644809}, \\
a^{\mathrm{E}}_{4,2} = \frac{99316866929}{820744730663}, \quad a^{\mathrm{E}}_{4,3} = \frac{82888780751}{969573940619}, \\
a^{\mathrm{E}}_{5,3} = \frac{57501241309}{765040883867}, \quad a^{\mathrm{E}}_{5,4} = \frac{76345938311}{676824576433}, \\
a^{\mathrm{E}}_{6,4} = -\frac{4099309936455}{6310162971841}, \quad a^{\mathrm{E}}_{6,5} = \frac{1395992540491}{933264948679}, \\
b_1 = \frac{232049084587}{1377130630063}, \quad b_2 = \frac{322009889509}{2243393849156}, \quad b_3 = -\frac{195109672787}{1233165545817}, \\
b_4 = -\frac{340582416761}{705418832319}, \quad b_5 = \frac{463396075661}{409972144477}, \quad b_6 = \frac{323177943294}{1626646580633},
\end{gathered} \tag{2.33c}$$

and $a^{\mathrm{E}}_{2,1} = c_2$ and $a^{\mathrm{I}}_{2,1} = a^{\mathrm{I}}_{2,2} = c_2/2$ from the stage-order conditions. The scheme is not SSP. The associated third-order embedded scheme is:

$$\begin{gathered}
\hat{b}_1 = \frac{5590918588}{49191225249}, \quad \hat{b}_2 = \frac{92380217342}{122399335103}, \quad \hat{b}_3 = -\frac{29257529014}{55608238079}, \\
\hat{b}_4 = -\frac{126677396901}{66917692409}, \quad \hat{b}_5 = \frac{384446411890}{169364936833}, \quad \hat{b}_6 = \frac{58325237543}{207682037557}
\end{gathered} \tag{2.33d}$$

The stability boundaries of the DIRK and ERK components are shown in Figures 2.1q-2.1r.

2.6 Order reduction

We now consider the *order reduction* present when the schemes developed above are applied to the van der Pol equation. It is well documented in the literature (see, e.g., [20]) that whenever an RK method is used to integrate a singular perturbation problem (that is, an ODE characterized by a stiffness parameter ε whose behavior transitions towards that of an index-1 DAE as the stiffness increases), the observed convergence rate appears to be lower than the nominal order of accuracy of the RK scheme used. In the seminal work of Hairer et al. [21], it is shown that the global error of DIRK schemes applied to singular perturbation problems may be written in the convenient form $E = C_1\,(\Delta t)^{n_1} + C_2\,\varepsilon\,(\Delta t)^{n_2}$. For the differential variables, DIRK methods have $n_1 = n$ and $n_2 = n_{SO} + 1$, where n is the nominal order of accuracy and n_{SO} is the stage order of the scheme. For the algebraic variables, if the DIRK method satisfies the aforementioned “stiff-accuracy" conditions, it turns out that[8] $n_1 = n$ and $n_2 = n_{SO}$; if not, however, $n_1 = n_{SO} + 1$ and $C_2 = 0$, which is generally much worse.

For IMEXRK methods, very little is known about order reduction outside of the empirical work of Kennedy & Carpenter in [7] and [23], where various IMEX schemes are tested on a range of singular perturbation problems. In this work, the greatest order reduction is observed in the case of the van der Pol equation; for this reason, we focus on this model problem in the present chapter in order to characterize the order reduction phenomenon. The van der Pol equation describes the dynamics of a nonlinear oscillator of the form

$$\frac{dy}{dt} = z, \qquad \varepsilon\,\frac{dz}{dt} = (1 - y^2)\,z - y, \tag{2.34}$$

where ε is known as the stiffness parameter. It is seen that, for $\varepsilon \to 0$, this ODE sys-

[8] Indeed, it is precisely for this reason that these “stiff-accuracy” conditions are so named.

tem transitions into an index-1 DAE, where $y(t)$ is a differential variable, and $z(t)$ transitions into an algebraic variable. The initial conditions used are $y(0) = 2$ and $z(0) = -0.6666654321121172$. All of the schemes introduced in this chapter have been tested on this system over the time interval $0 \leq t \leq T$, taking $T = 0.5$, with various values for the (constant) stepsize Δt and stiffness parameter ε. The error at $t = T$ has then been used to estimate the convergence rate (that is, n_1 and n_2) as the stiffness parameter ε is decreased. The procedure used is analogous to that described in [7]: by fixing ε and varying Δt in the $\Delta t \to \varepsilon$ limit, the change of slope in the convergence rate has been detected and used to estimate n_1 and n_2. Results of such simulations are reported in Figure 2.4, and empirical estimates of the convergence rates for each method are reported in Table 2.2. When only the DIRK component of the schemes are used, the results generally show good agreement with the theoretical bounds provided in [21]. If the entire IMEX schemes are used, results do not differ substantially from those reported in [7]. The order-reduction phenomenon tends to be problem dependent; results in practice (see [7]) often indicate behavior significantly better than the corresponding theoretical bounds. Note also that imposing stage-order two on the DIRK component of a scheme does not influence the convergence of the entire IMEX scheme, though it significantly improves the accuracy when the DIRK component only is used.

2.7 Computational cost

To illustrate the relative computational cost of our new low-storage IMEXRK schemes on a representative PDE model problem discretized on $N \gg 1$ gridpoints, we now compare the efficient implementation of each of the methods developed herein to **CN/RKW3** and several full-storage IMEX Runge-Kutta schemes available in literature. We consider

Table 2.2: Estimated convergence rates of the differential and algebraic variables on the van der Pol equation for **CN/RKW3** and the IMEXRK schemes presented in this chapter, and their associated DIRK components only.

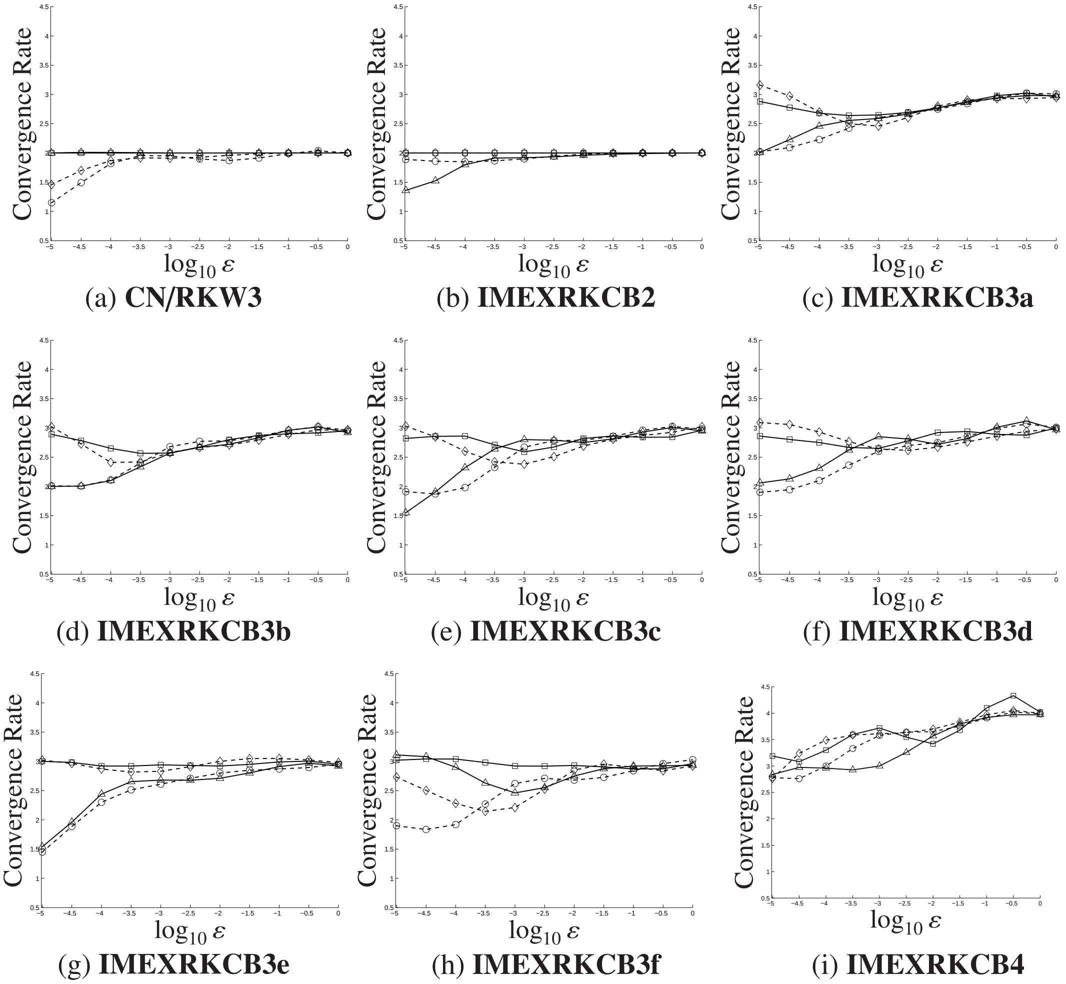
Method	IMEXRK scheme differential part	IMEXRK scheme algebraic part
CN/RKW3	$(\Delta t)^2 + \varepsilon(\Delta t)^1$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
IMEXRKCB2	$(\Delta t)^2 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
IMEXRKCB3a	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
IMEXRKCB3b	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
IMEXRKCB3c	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
IMEXRKCB3d	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
IMEXRKCB3e	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
IMEXRKCB3f	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
IMEXRKCB4	$(\Delta t)^4 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^1$

Method	DIRK scheme only differential part	DIRK scheme only algebraic part
CN/RKW3	$(\Delta t)^2 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^2$
IMEXRKCB2	$(\Delta t)^2 + \varepsilon(\Delta t)^2$	$(\Delta t)^2 + \varepsilon(\Delta t)^1$
IMEXRKCB3a	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^1$
IMEXRKCB3b	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^1$
IMEXRKCB3c	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^1$
IMEXRKCB3d	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^1$
IMEXRKCB3e	$(\Delta t)^3 + \varepsilon(\Delta t)^2$	$(\Delta t)^3 + \varepsilon(\Delta t)^1$
IMEXRKCB3f	$(\Delta t)^3 + \varepsilon(\Delta t)^3$	$(\Delta t)^3 + \varepsilon(\Delta t)^2$
IMEXRKCB4	$(\Delta t)^4 + \varepsilon(\Delta t)^3$	$(\Delta t)^4 + \varepsilon(\Delta t)^2$

as a model PDE problem the one-dimensional Kuramoto-Sivashinsky equation

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4} \tag{2.35}$$

over the domain $x \in [-L/2, L/2]$ with $u = \partial u/\partial x = 0$ at $x = \pm L/2$, where L is the width of the domain. It should be remarked that, unlike the van der Pol case, this example



(a) **CN/RKW3** (b) **IMEXRKCB2** (c) **IMEXRKCB3a**

(d) **IMEXRKCB3b** (e) **IMEXRKCB3c** (f) **IMEXRKCB3d**

(g) **IMEXRKCB3e** (h) **IMEXRKCB3f** (i) **IMEXRKCB4**

Figure 2.4: Convergence rates for the low-storage IMEXRK schemes considered in this chapter when applied to the van der Pol equation, as a function of ε. Solid lines are for simulations using the DIRK component only (squares for the differential variables, triangles for the algebraic variables), whereas dashed lines are for the simulations using the entire IMEXRK scheme (diamonds for the differential variables, circles for the algebraic variables).

represents a rather undemanding application of our IMEXRK schemes. The sole purpose of this analysis is the comparison of the computational cost that our new schemes require with respect to other IMEXRK schemes available in literature; the implementation of a selection of these schemes in a DNS code for the simulation of an incompressible turbulent channel flow is currently underway, and will be reported elsewhere. The RHS of (3.22) consists of a nonlinear convective term, treated explicitly, and two linear terms, treated implicitly.

Following a five-point central finite-difference (FD) approach on a uniform grid, (3.22) can be approximated as

$$\frac{d\mathbf{u}}{dt} = A\,\mathbf{u} + \mathbf{g}(\mathbf{u}),$$

where A is a pentadiagonal Toeplitz matrix obtained by discretizing the last two terms on the RHS of (3.22), and $g_i(\mathbf{u}) = -u_i(u_{i-2} - 8u_{i-1} + 8u_{i+1} - u_{i+2})/(12\Delta x)$. As an example, using the 3-register implementation (2.19) of the **CN/RKW3** method (2.23), $6N$ flops times 3 stages are required for the evaluation of the nonlinear term, $19N$ flops times 3 stages are required for the implicit (pentadiagonal) solves, and $40N$ additional flops are required for basic product/sum operations; thus, $115N$ flops per timestep are required.

Following a pseudospectral (PS) approach, with nonlinear products computed in physical space and spatial derivatives computed in Fourier space, (3.22) can be written in wavenumber space as

$$\frac{d\hat{u}_n}{dt} = -\frac{\imath\, k_{x_n}}{2}\widehat{(u^2)}_n + (k_{x_n}^2 - k_{x_n}^4)\hat{u}_n \tag{2.36}$$

where $\imath = \sqrt{-1}$, $k_{x_n} = 2\pi n/L$ is the wavenumber, and $\widehat{(u^2)}_n$ denotes the n'th wavenumber component of the function computed by transforming u to physical space on $N = 2^p$ equispaced gridpoints, computing u^2 at each gridpoint, and transforming the result back to Fourier space. Since computing FFTs requires $\sim 5N \log N$ real flops while all other operations are linear in N, the number of FFTs performed represents the leading-order computational cost for large N. As an example, the 3-register implementation of **CN/RKW3** requires 2 FFTs per stage for each of three stages.

The computational cost of the other schemes may be counted similarly; results are summarized in the last two columns of Table 2.1. It is seen that, if computational cost is naïvely characterized simply by the number of floating point operations required per timestep, the present low-storage IMEXRK schemes are in fact competitive with both **CN/RKW3** and all of the full-storage IMEXRK schemes available in the literature of the corresponding order. The fact that **CN/RKW3** and all of our low-storage IMEXRK schemes admit two-, three-, or four-register implementations, however, bestows them with a distinct operational advantage for high-dimensional ODE discretizations of PDE systems.

2.8 Conclusions

We have developed eight new IMEX Runge-Kutta schemes with reduced storage requirements, the properties of which are succinctly summarized and compared with competing schemes in Table 2.1. It is seen that:

- **IMEXRKCB2** is second-order accurate, like **CN/RKW3**; **IMEXRKCB3a-3f** are third-order accurate, and **IMEXRKCB4** is fourth-order accurate.

- **IMEXRKCB2, 3a-3e**, like **CN/RKW3**, admit both two-register and three-register implementations, with the three-register implementations requiring slightly fewer flops.

- **IMEXRKCB3f, 4** admit both three-register and four-register implementations, with the four-register implementations requiring significantly fewer flops; the four-register implementations of these two schemes are thus generally recommended, unless the additional storage that the four-register implementations require represents a particularly acute computational disadvantage.

- **IMEXRKCB2, 3a** generally require fewer floating-point operations per timestep than **CN/RKW3**, whereas the other schemes we have developed generally require progressively more; this comparison, however, is somewhat problem dependent.

- **IMEXRKCB2, 3c-3f, 4** are *L*-stable, whereas **IMEXRKCB3a-3b** are strongly *A*-stable (**CN/RKW3** is only *A*-stable), making them well suited for stiff ODEs.

- **IMEXRKCB2, 3c, 3d, 3f, 4** are each provided with a reduced-order embedded scheme following the guidelines listed in §2.1.2, making them well suited for application in adaptive time-stepping applications.

- **IMEXRKCB3b** incorporates an ESDIRK implicit component, and is thus better suited to leverage an LU decomposition during the implicit solves than either **CN/RKW3** or our other schemes.

- **IMEXRKCB2, 3c, 3d** are strong stability preserving (SSP) under the appropriate timestep restriction, and are thus better suited for application to hyperbolic systems than either **CN/RKW3** or our other schemes.

- **IMEXRKCB3f, 4** have stage order two, whereas **CN/RKW3** and our other schemes have stage order one; these two schemes thus show better convergence properties when applied to especially stiff ODE systems.

Acknowledgements

This chapter contains work previously published in:

- D. Cavaglieri, T.R. Bewley, “Low-storage implicit/explicit Runge-Kutta schemes for the simulation of stiff high-dimensional ODE systems", *Journal of Computational Physics*, (286) 172-193, 2015

Chapter 3

Low-storage implicit/explicit Runge-Kutta schemes for the simulation of the Navier-Stokes Equations

3.1 Introduction

Many physical phenomena in fluid dynamics can be modeled as systems of partial differential equations of the form

$$\frac{d\mathbf{u}}{dt} = \mathcal{L}\,\mathbf{u} + \mathcal{N}(\mathbf{u}), \tag{3.1}$$

where $\mathcal{L}$ is a linear spatial operator, and $\mathcal{N}$ is a nonlinear operator. This is the case, for example, of the incompressible Navier-Stokes Equations, if we assume that no time-dependent error is committed while accounting for the incompressibility requirement. In general, the linear operator of the NSE is associated to the discretization of the diffusive terms, while the nonlinear operator accounts for the convective terms. In the case of the NSE, the nonlinear operator is in fact bilinear. Since the diffusive terms are in generally stiff, while the convective terms are usually nonstiff, time discretization for DNS and LES simulations in the past three decades has relied principally on a mixed implicit/explicit approach, in which the integration of the diffusive term is carried out implicitly, while the convective term is marched explicitly. In this way, numerical stability of the time stepping scheme is limited solely by the Courant-Friedrichs-Lewy (CFL) condition [24].

The first attempt in Moin et al. [2] combined second-order explicit Adams-Bashforth (AB2) for the integration of the bilinear term and second-order implicit Crank-Nicolson (CN) at each substep for the integration of the linear term. However, since AB2 has no stability over the imaginary axis, the integration of the convective term produces a weak instability which generally does not compromise the overall stability of the simulation but it definitely affects its accuracy. For this reason, AB2 was later replaced by explicit third-order low-storage Runge-Kutta-Wray (RKW3) scheme [5]. The resulting hybrid scheme, first presented in [3] and often referred to as **CN/RKW3**, improves the order of accuracy of the time integration of the convective terms. Besides, it guarantees numerical stability of the time stepping scheme under the CFL limit. Furthermore, its numerical implementation, 2which leverages an incremental formulation, allows to keep storage requirements to a minimum. Independently, Spalart et al. [1] proposed an IMEXRK scheme following the same incremental form. This scheme, referred to as **IMEXRKiSMR**, integrates the solution $\mathbf{u}_n$ at time t_n of PDE (3.1) over the time interval $[t_n, t_{n+1}]$, following a three-step formulation:

$$\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}_n + \Delta t \left(\alpha_1^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(1)} + \beta_1^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}_n + \beta_1^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}_n)\right) \\
\mathbf{u}^{(2)} &= \mathbf{u}^{(1)} + \Delta t \left(\alpha_2^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(2)} + \beta_2^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(1)} + \beta_2^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}^{(1)}) + \gamma_2^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}_n)\right) \qquad (3.2) \\
\mathbf{u}_{n+1} &= \mathbf{u}^{(2)} + \Delta t \left(\alpha_3^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}_{n+1} + \beta_3^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(2)} + \beta_3^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}^{(2)}) + \gamma_3^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}^{(1)})\right),
\end{aligned}$$

where $\Delta t = t_{n+1} - t_n$ is the step size and $\mathbf{u}_{n+1}$ is the solution at time t_{n+1}. The coefficients $\alpha_i^{I/E}$, $\beta_i^{I/E}$, and γ_i^E in (3.2) were determined in an attempt to match the Taylor expansion of such scheme with the expansion of the discretized NSE up to third order [1]. Additional constraints were impose in order to obtain an equal size for the integration substeps for both implicit and explicit components. However, it was observed by the authors that it is not possible, within this formulation, to satisfy all the third-order accuracy constraints arising from Taylor expansion analysis. For this reason, the coefficients were chosen in order to satisfy all accuracy constraints but one. This decision lead to a one-parameter family, in which the remaining degree of freedom was chosen in order to find a compromise between minimizing the third-order truncation error and having relatively even substeps. These considerations lead to the coefficients

$$\begin{aligned}
&\alpha_1^{\mathrm{I}} = \frac{37}{160}, \quad \beta_1^{\mathrm{I}} = \frac{29}{96}, \quad \beta_1^{\mathrm{E}} = \frac{8}{15}, \\
&\alpha_2^{\mathrm{I}} = \frac{5}{24}, \quad \beta_2^{\mathrm{I}} = -\frac{3}{40}, \quad \beta_2^{\mathrm{E}} = \frac{5}{12}, \quad \gamma_2^{\mathrm{E}} = -\frac{17}{60} \qquad (3.3) \\
&\alpha_3^{\mathrm{I}} = \frac{1}{6}, \quad \beta_3^{\mathrm{I}} = \frac{1}{6}, \quad \beta_3^{\mathrm{E}} = \frac{3}{4}, \quad \gamma_3^{\mathrm{E}} = -\frac{5}{12}.
\end{aligned}$$

The scheme thus obtained is second-order accurate on the linear term, while it is third-order accurate on the explicit term and the mixed implicit/explicit terms arising from Taylor expansion. Furthermore, the implicit part is strongly A-stable, while the explicit one has a stability limit along the imaginary axis equal to $\sqrt{3}$. In comparison, **CN/RKW3**, rearranged in order to fit the incremental form in (3.2), presents a different choice of coefficients:

$$\begin{aligned} &\alpha_1^{\mathrm{I}} = \beta_1^{\mathrm{I}} = \frac{4}{15}, \quad \beta_1^{\mathrm{E}} = \frac{8}{15}, \\ &\alpha_2^{\mathrm{I}} = \beta_2^{\mathrm{I}} = \frac{1}{15}, \quad \beta_2^{\mathrm{E}} = \frac{5}{12}, \quad \gamma_2^{\mathrm{E}} = -\frac{17}{60} \\ &\alpha_3^{\mathrm{I}} = \beta_3^{\mathrm{I}} = \frac{1}{6}, \quad \beta_3^{\mathrm{E}} = \frac{3}{4}, \quad \gamma_3^{\mathrm{E}} = -\frac{5}{12}. \end{aligned} \tag{3.4}$$

This alternative choice preserves third-order accuracy and imaginary stability of the explicit part (RKW3 scheme is used in both cases), whereas it leads to a slight increase in the third-order truncation error of the implicit component and mixed terms. Furthermore, the implicit part is now only A-stable, since it matches the stability properties of Crank-Nicolson.

Our goal is to develop new low-storage Runge-Kutta schemes with the same incremental formulation in (3.2), but better overall accuracy and improved stability properties for both implicit and explicit components. In particular, schemes up to five steps will be considered. For the four-step scheme only, the possibility of allowing extra storage in order to improve certain stability properties will be investigated.

3.1.1 Low-storage IMEXRK formulation

Although particularly appealing from an implementation point of view, the incremental form slightly complicates the imposition of the constraints needed for a scheme to achieve a prescribed order of accuracy or desirable stability properties. For this reason, the development of these schemes is carried out by first resorting to the IMEXRK formulation leveraging Butcher coefficients [7, 9, 17, 25]. In this framework, each scheme is represented by two coupled Butcher tableaux, one for the implicit integration of the stiff component and the other for the explicit integration of the nonstiff component. In this way, a generic $(s-1)$-step incremental scheme like the one in (3.2) can be represented as

$$
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & \beta_1^{\mathrm{I}} & \alpha_1^{\mathrm{I}} & & & & \\
c_3 & \beta_1^{\mathrm{I}} & \beta_2^{\mathrm{I}}+\alpha_1^{\mathrm{I}} & \alpha_2^{\mathrm{I}} & & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & & \\
c_{s-1} & \beta_1^{\mathrm{I}} & \beta_2^{\mathrm{I}}+\alpha_1^{\mathrm{I}} & \cdots & \beta_{s-2}^{\mathrm{I}}+\alpha_{s-3}^{\mathrm{I}} & \alpha_{s-2}^{\mathrm{I}} & \\
1 & \beta_1^{\mathrm{I}} & \beta_2^{\mathrm{I}}+\alpha_1^{\mathrm{I}} & \cdots & \beta_{s-2}^{\mathrm{I}}+\alpha_{s-3}^{\mathrm{I}} & \beta_{s-1}^{\mathrm{I}}+\alpha_{s-2}^{\mathrm{I}} & \alpha_{s-1}^{\mathrm{I}} \\
\hline
 & \beta_1^{\mathrm{I}} & \beta_2^{\mathrm{I}}+\alpha_1^{\mathrm{I}} & \cdots & \beta_{s-2}^{\mathrm{I}}+\alpha_{s-3}^{\mathrm{I}} & \beta_{s-1}^{\mathrm{I}}+\alpha_{s-2}^{\mathrm{I}} & \alpha_{s-1}^{\mathrm{I}}
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & \beta_1^{\mathrm{E}} & 0 & & & & \\
c_3 & \beta_1^{\mathrm{E}}+\gamma_2^{\mathrm{E}} & \beta_2^{\mathrm{E}} & 0 & & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & & \\
c_{s-1} & \beta_1^{\mathrm{E}}+\gamma_2^{\mathrm{E}} & \beta_2^{\mathrm{E}}+\gamma_3^{\mathrm{E}} & \cdots & \beta_{s-2}^{\mathrm{E}} & 0 & \\
1 & \beta_1^{\mathrm{E}}+\gamma_2^{\mathrm{E}} & \beta_2^{\mathrm{E}}+\gamma_3^{\mathrm{E}} & \cdots & \beta_{s-2}^{\mathrm{E}}+\gamma_{s-1}^{\mathrm{E}} & \beta_{s-1}^{\mathrm{E}} & 0 \\
\hline
 & \beta_1^{\mathrm{E}}+\gamma_2^{\mathrm{E}} & \beta_2^{\mathrm{E}}+\gamma_3^{\mathrm{E}} & \cdots & \beta_{s-2}^{\mathrm{E}}+\gamma_{s-1}^{\mathrm{E}} & \beta_{s-1}^{\mathrm{E}} & 0
\end{array}
\tag{3.5}
$$

A trivial change of variables allows to resort to the standard formulation

$$
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & b_1^{\mathrm{I}} & a_{2,2}^{\mathrm{I}} & & & & \\
c_3 & b_1^{\mathrm{I}} & b_2^{\mathrm{I}} & a_{3,3}^{\mathrm{I}} & & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & & \\
c_{s-1} & b_1^{\mathrm{I}} & b_2^{\mathrm{I}} & \cdots & b_{s-2}^{\mathrm{I}} & a_{s-1,s-1}^{\mathrm{I}} & \\
1 & b_1^{\mathrm{I}} & b_2^{\mathrm{I}} & \cdots & b_{s-2}^{\mathrm{I}} & b_{s-1}^{\mathrm{I}} & b_s^{\mathrm{I}} \\
\hline
 & b_1^{\mathrm{I}} & b_2^{\mathrm{I}} & \cdots & b_{s-2}^{\mathrm{I}} & b_{s-1}^{\mathrm{I}} & b_s^{\mathrm{I}}
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a_{2,1}^{\mathrm{E}} & 0 & & & & \\
c_3 & b_1^{\mathrm{E}} & a_{3,2}^{\mathrm{E}} & 0 & & & \\
\vdots & \vdots & \vdots & \ddots & \ddots & & \\
c_{s-1} & b_1^{\mathrm{E}} & b_2^{\mathrm{E}} & \cdots & a_{s-1,s-2}^{\mathrm{E}} & 0 & \\
1 & b_1^{\mathrm{E}} & b_2^{\mathrm{E}} & \cdots & b_{s-2}^{\mathrm{E}} & b_{s-1}^{\mathrm{E}} & 0 \\
\hline
 & b_1^{\mathrm{E}} & b_2^{\mathrm{E}} & \cdots & b_{s-2}^{\mathrm{E}} & b_{s-1}^{\mathrm{E}} & 0
\end{array}
\tag{3.6}
$$

The structure of these Butcher tableaux offers some interesting insights into the properties of this family of IMEXRK schemes. First of all, all the coefficients below the first subdiagonal for both the explicit and implicit component equal the b_i coefficients. As we recently showed in [26], this condition allows a low-storage implementation of these IMEXRK schemes which requires only two-registers, or three registers in case less computation is desired. Such implementation follows closely that of explicit low-storage [2R] Runge-Kutta algorithms. A comprehensive review of this subject is given in [8]. Remarkably, the structure of the Butcher tableaux in (3.6) appears to be even more restrictive, since also the coefficients along the first subdiagonal of the implicit component equal the b_i coefficients.

A significant difference between these low-storage IMEXRK schemes and those developed in [26] is that the coordination condition, i.e. $b_i^{\mathrm{I}} = b_i^{\mathrm{E}}$, is not imposed. This increases the number of constraints that have to be satisfied in order to meet the desired order of accuracy. However, a substantial reduction in the number of accuracy conditions to impose still remains, since the synchronization condition, i.e. $c_i^{\mathrm{I}} = c_i^{\mathrm{E}}$, is verified.

The last observation regards the first-same-as-last (FSAL) condition, which is achieved for both implicit and explicit part. This results in the scheme implementation requiring only $(s-1)$ stages, even if the associated Butcher tableaux formally figure s stages. This result is somehow to be expected, since the corresponding incremental form has only $(s-1)$ steps. Furthermore, the FSAL condition allows some simplifications in the structure of the stability polynomial, as we will show in Section 3.1.3.

3.1.2 Accuracy conditions

Considering a generic ODE

$$\frac{d\mathbf{u}(t)}{dt} = \mathbf{f}(\mathbf{u},\, t) + \mathbf{g}(\mathbf{u},\, t), \tag{3.7}$$

where **f** represents the stiff part, to be treated implicitly, and **g** the nonstiff part, to be treated explicitly, an s-stage IMEXRK scheme with synchronization condition only has to satisfy 2 constraints for first order accuracy, 4 constraints for second order, 10 for third order, and 28 for fourth order. Additionally, as is customary, stage-order-one (SO1) condition, i.e. $\sum_j a_{i,j}^{\mathrm{I/E}} = c_i$ for all $i = 1, 2, \ldots, s$, is imposed. These conditions, also outlined in [27], are reported below:

- First order conditions

$$\tau_1^{(1)\mathrm{I}} = \sum_{i=1}^{s} b_i^{\mathrm{I}} - 1 \qquad \tau_1^{(1)\mathrm{E}} = \sum_{i=1}^{s} b_i^{\mathrm{E}} - 1 \tag{3.8}$$

- Second order conditions

$$\tau_1^{(2)\mathrm{I}} = \sum_{i=1}^{s} b_i^{\mathrm{I}}\, c_i - \frac{1}{2} \qquad \tau_1^{(2)\mathrm{E}} = \sum_{i=1}^{s} b_i^{\mathrm{E}}\, c_i - \frac{1}{2} \tag{3.9}$$

- Third order conditions

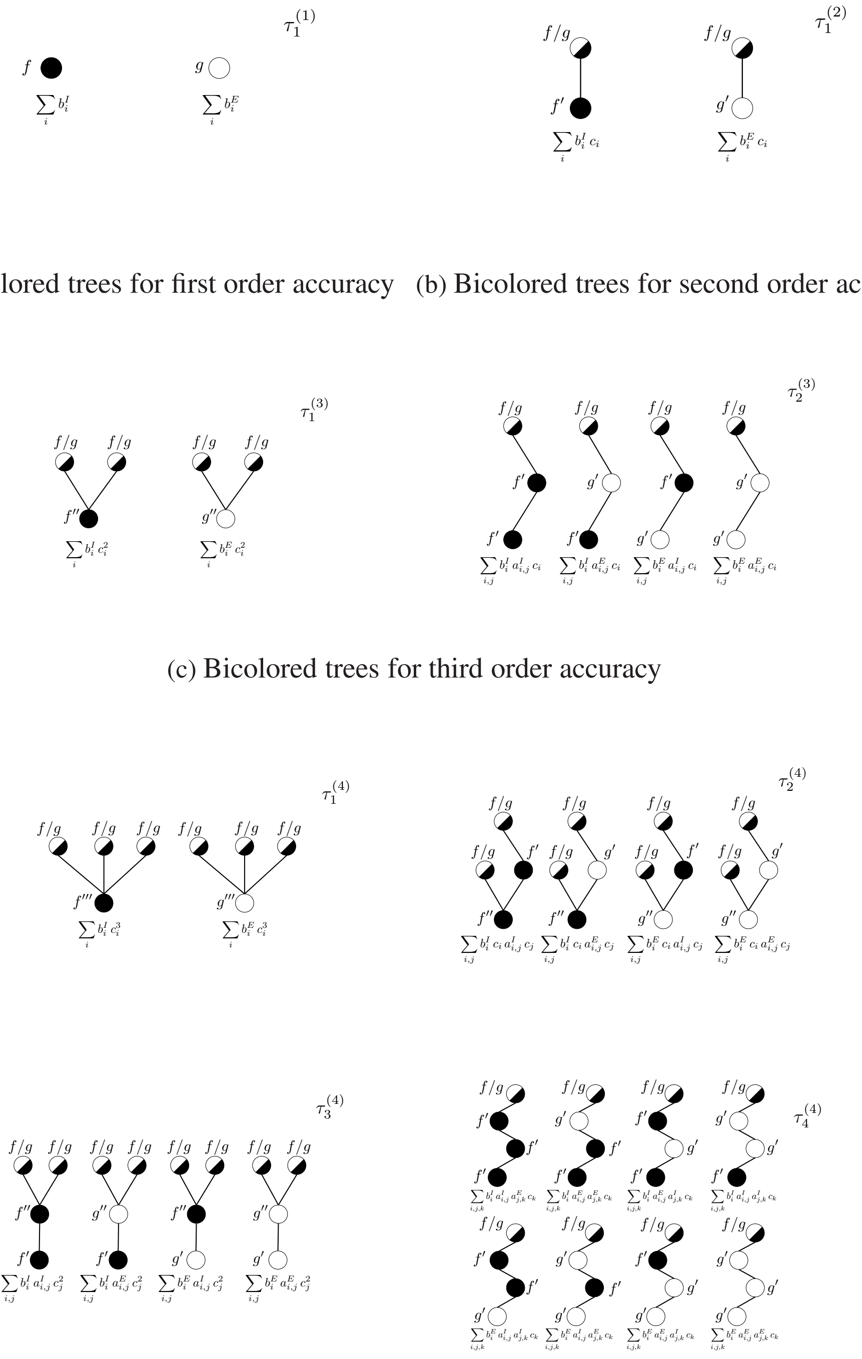
$$\begin{aligned}
\tau_1^{(3)\mathrm{I}} &= \frac{1}{2}\sum_{i=1}^{s} b_i^{\mathrm{I}}\, c_i^2 - \frac{1}{6} & \tau_1^{(3)\mathrm{E}} &= \frac{1}{2}\sum_{i=1}^{s} b_i^{\mathrm{E}}\, c_i^2 - \frac{1}{6} \\
\tau_2^{(3)\mathrm{II}} &= \sum_{i,j=1}^{s} b_i^{\mathrm{I}}\, a_{i,j}^{\mathrm{I}}\, c_i - \frac{1}{6} & \tau_2^{(3)\mathrm{IE}} &= \sum_{i,j=1}^{s} b_i^{\mathrm{I}}\, a_{i,j}^{\mathrm{E}}\, c_j - \frac{1}{6} \\
\tau_2^{(3)\mathrm{EI}} &= \sum_{i,j=1}^{s} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{I}}\, c_i - \frac{1}{6} & \tau_2^{(3)\mathrm{EE}} &= \sum_{i,j=1}^{s} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{E}}\, c_j - \frac{1}{6}
\end{aligned} \tag{3.10}$$

- Fourth order conditions

$$\begin{aligned}
\tau_1^{(4)\mathrm{I}} &= \frac{1}{6}\sum_{i=1}^{s} b_i^{\mathrm{I}}\, c_i^3 - \frac{1}{24} & \tau_1^{(4)\mathrm{E}} &= \frac{1}{6}\sum_{i=1}^{s} b_i^{\mathrm{E}}\, c_i^3 - \frac{1}{24} \\
\tau_2^{(4)\mathrm{II}} &= \sum_{i,j=1}^{s} b_i^{\mathrm{I}}\, c_i\, a_{i,j}^{\mathrm{I}}\, c_j - \frac{3}{24} & \tau_2^{(4)\mathrm{IE}} &= \sum_{i,j=1}^{s} b_i^{\mathrm{I}}\, c_i\, a_{i,j}^{\mathrm{E}}\, c_j - \frac{3}{24} \\
\tau_2^{(4)\mathrm{EI}} &= \sum_{i,j=1}^{s} b_i^{\mathrm{E}}\, c_i\, a_{i,j}^{\mathrm{I}}\, c_j - \frac{3}{24} & \tau_2^{(4)\mathrm{EE}} &= \sum_{i,j=1}^{s} b_i^{\mathrm{E}}\, c_i\, a_{i,j}^{\mathrm{E}}\, c_j - \frac{3}{24} \\
\tau_3^{(4)\mathrm{II}} &= \frac{1}{2}\sum_{i,j=1}^{s} b_i^{\mathrm{I}}\, a_{i,j}^{\mathrm{I}}\, c_j^2 - \frac{1}{24} & \tau_3^{(4)\mathrm{IE}} &= \frac{1}{2}\sum_{i,j=1}^{s} b_i^{\mathrm{I}}\, a_{i,j}^{\mathrm{E}}\, c_j^2 - \frac{1}{24} \\
\tau_3^{(4)\mathrm{EI}} &= \frac{1}{2}\sum_{i,j=1}^{s} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{I}}\, c_j^2 - \frac{1}{24} & \tau_3^{(4)\mathrm{EE}} &= \frac{1}{2}\sum_{i,j=1}^{s} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{E}}\, c_j^2 - \frac{1}{24} \\
\tau_4^{(4)\mathrm{III}} &= \sum_{i,j,k=1}^{s} b_i^{\mathrm{I}}\, a_{i,j}^{\mathrm{I}}\, a_{j,k}^{\mathrm{I}}\, c_k - \frac{1}{24} & \tau_4^{(4)\mathrm{IIE}} &= \sum_{i,j,k=1}^{s} b_i^{\mathrm{I}}\, a_{i,j}^{\mathrm{I}}\, a_{j,k}^{\mathrm{E}}\, c_k - \frac{1}{24} \\
\tau_4^{(4)\mathrm{IEI}} &= \sum_{i,j,k=1}^{s} b_i^{\mathrm{I}}\, a_{i,j}^{\mathrm{E}}\, a_{j,k}^{\mathrm{I}}\, c_k - \frac{1}{24} & \tau_4^{(4)\mathrm{IEE}} &= \sum_{i,j,k=1}^{s} b_i^{\mathrm{I}}\, a_{i,j}^{\mathrm{E}}\, a_{j,k}^{\mathrm{E}}\, c_k - \frac{1}{24} \\
\tau_4^{(4)\mathrm{EII}} &= \sum_{i,j,k=1}^{s} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{I}}\, a_{j,k}^{\mathrm{I}}\, c_k - \frac{1}{24} & \tau_4^{(4)\mathrm{EIE}} &= \sum_{i,j,k=1}^{s} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{I}}\, a_{j,k}^{\mathrm{E}}\, c_k - \frac{1}{24} \\
\tau_4^{(4)\mathrm{EEI}} &= \sum_{i,j,k=1}^{s} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{E}}\, a_{j,k}^{\mathrm{I}}\, c_k - \frac{1}{24} & \tau_4^{(4)\mathrm{EEE}} &= \sum_{i,j,k=1}^{s} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{E}}\, a_{j,k}^{\mathrm{E}}\, c_k - \frac{1}{24}.
\end{aligned} \tag{3.11}$$

However, the particular structure of the **f** and **g** operators in the discretized NSE (3.1) is such that not all the order conditions need to be satisfied in order to time march these equations with a prescribed order of accuracy. More specifically, since the stiff term is linear and autonomous, condition $\tau_1^{(3)\mathrm{I}}$ does not need to be imposed to achieve third-order accuracy, since such term is proportional to the second derivative of **f**. When fourth order accuracy is addressed, further constraints are dropped from consideration, since not only the stiff term is linear, but also the nonstiff component is bilinear and autonomous, and therefore derivatives of **g** higher than second do not appear. Bicolored trees [28], which are an extension of Butcher single-color trees (see [20, 29, 30]) allow to identify which terms contribute to define the order of accuracy of the IMEXRK schemes when applied to the time integration of the NSE. Bicolored trees up to fourth order are shown in Figure 3.1. By inspection, we can conclude that $\tau_1^{(4)\mathrm{I}}$, $\tau_1^{(4)\mathrm{E}}$, $\tau_2^{(4)\mathrm{II}}$, $\tau_2^{(4)\mathrm{IE}}$, $\tau_3^{(4)\mathrm{II}}$, and $\tau_3^{(4)\mathrm{EI}}$ do not appear in the Taylor expansion of the NSE operator. Therefore, only nine constraints, instead of ten, must be satisfied for third order accuracy, plus twelve additional constraints for fourth order accuracy.

After a desired level of accuracy p has been achieved, the remaining degrees of freedom will be used to minimize the error norm $A^{(p+1)}$, a metric first introduced in [31] and adopted for both full-storage [7] and low-storage IMEXRK schemes [26]. This norm is defined as the square root of the sum of the squares of all the error terms $\tau_j^{(p+1)\mathrm{I/E}}$ that contribute to the leading order, i.e. $(p + 1)$, of the truncation error related to the time advancement of the NSE.


$\tau_1^{(1)}$
f
g
$\sum_i b_i^I$
$\sum_i b_i^E$
$\tau_1^{(2)}$
f/g
f'
g'
$\sum_i b_i^I c_i$
$\sum_i b_i^E c_i$
$\tau_1^{(3)}$
f''
g''
$\sum_i b_i^I c_i^2$
$\sum_i b_i^E c_i^2$
$\tau_2^{(3)}$
$\sum_{i,j} b_i^I a_{i,j}^I c_i$
$\sum_{i,j} b_i^I a_{i,j}^E c_i$
$\sum_{i,j} b_i^E a_{i,j}^I c_i$
$\sum_{i,j} b_i^E a_{i,j}^E c_i$
$\tau_1^{(4)}$
f'''
g'''
$\sum_i b_i^I c_i^3$
$\sum_i b_i^E c_i^3$
$\tau_2^{(4)}$
$\sum_{i,j} b_i^I c_i a_{i,j}^I c_j$
$\sum_{i,j} b_i^I c_i a_{i,j}^E c_j$
$\sum_{i,j} b_i^E c_i a_{i,j}^I c_j$
$\sum_{i,j} b_i^E c_i a_{i,j}^E c_j$
$\tau_3^{(4)}$
$\sum_{i,j} b_i^I a_{i,j}^I c_j^2$
$\sum_{i,j} b_i^I a_{i,j}^E c_j^2$
$\sum_{i,j} b_i^E a_{i,j}^I c_j^2$
$\sum_{i,j} b_i^E a_{i,j}^E c_j^2$
$\tau_4^{(4)}$
$\sum_{i,j,k} b_i^I a_{i,j}^I a_{j,k}^E c_k$
$\sum_{i,j,k} b_i^I a_{i,j}^E a_{j,k}^E c_k$
$\sum_{i,j,k} b_i^I a_{i,j}^E a_{j,k}^I c_k$
$\sum_{i,j,k} b_i^I a_{i,j}^I a_{j,k}^I c_k$
$\sum_{i,j,k} b_i^E a_{i,j}^I a_{j,k}^I c_k$
$\sum_{i,j,k} b_i^E a_{i,j}^I a_{j,k}^E c_k$
$\sum_{i,j,k} b_i^E a_{i,j}^E a_{j,k}^I c_k$
$\sum_{i,j,k} b_i^E a_{i,j}^E a_{j,k}^E c_k$

(a) Bicolored trees for first order accuracy (b) Bicolored trees for second order accuracy

(c) Bicolored trees for third order accuracy

(e) Bicolored trees for fourth order accuracy

Figure 3.1: Bicolored trees for accuracy conditions of IMEXRK schemes up to fourth order.

3.1.3 Stability

Considering the linear scalar test problem

$$\frac{d\mathbf{u}}{dt} = \lambda_f \mathbf{u} + \lambda_g \mathbf{u} \tag{3.12}$$

and using the implicit part of a generic s-stage incremental IMEXRK scheme to integrate the term $\lambda_f \mathbf{u}$ and the explicit part to integrate $\lambda_g \mathbf{u}$, linear stability can be analyzed using the stability function [10, 27]

$$\sigma(z^{\mathrm{I}}, z^{\mathrm{E}}) = \frac{\det\left[I - z^{\mathrm{I}}A^{\mathrm{I}} - z^{\mathrm{E}}A^{\mathrm{E}} + z^{\mathrm{I}}\mathbf{e}(\mathbf{b}^{\mathrm{I}})^T + z^{\mathrm{E}}\mathbf{e}(\mathbf{b}^{\mathrm{E}})^T\right]}{\det\left[I - z^{\mathrm{I}}A^{\mathrm{I}}\right]} = \frac{P(z^{\mathrm{I}}, z^{\mathrm{E}})}{Q(z^{\mathrm{I}})} \tag{3.13}$$

where $\mathbf{e}$ is a vector of ones, I is the identity matrix, $z^{\mathrm{I}} = \lambda_f \Delta t$, $z^{\mathrm{E}} = \lambda_g \Delta t$, $A^{\mathrm{I/E}} = a_{i,j}^{\mathrm{I/E}}$, and $\mathbf{b}^{\mathrm{I/E}} = b_i^{\mathrm{I/E}}$. Furthermore,

$$\begin{aligned} P(z^{\mathrm{I}}, z^{\mathrm{E}}) &= \sum_{i=0}^{s} \left(\sum_{j=0}^{s-i} p_{i,j}[z^{\mathrm{E}}]^j \right) [z^{\mathrm{I}}]^i \\ &= \sum_{i=0}^{s-2} \left(\sum_{j=0}^{s-2-i} p_{i,j}[z^{\mathrm{E}}]^j \right) [z^{\mathrm{I}}]^i + \left(p_{s-1,0} + p_{s-1,1}\, z^{\mathrm{E}} \right) [z^{\mathrm{I}}]^{s-1} + p_{s,0}[z^{\mathrm{I}}]^s \end{aligned} \tag{3.14}$$

$$Q(z^{\mathrm{I}}) = \sum_{i=0}^{s-1} q_i[z^{\mathrm{I}}]^i = \sum_{i=1}^{s-2} q_i[z^{\mathrm{I}}]^i + q_{s-1}[z^{\mathrm{I}}]^{s-1}, \tag{3.15}$$

where each coefficient $p_{i,j}$ and q_i is a function of the Butcher coefficients $a_{i,j}^{\mathrm{I/E}}$, $b_i^{\mathrm{I/E}}$, and c_i. In order to guarantee L-stability for a generic s-stage IMEXRK scheme, $p_{s,0}$, $p_{s-1,0}$, and $p_{s-1,1}$ must vanish, provided q_{s-1} at the denominator does not reduce to zero at the same time. Since both the implicit and explicit part of these IMEXRK schemes satisfy the FSAL condition, we have that $p_{s,0}$ and $p_{s-1,1}$ are already zero. Hence, L-stability will be achieved by imposing $p_{s-1,0} = 0$, with $q_{s-1} \neq 0$. Whenever L-stability could not be achieved, at least strong A-stability, i.e. $\sigma_\infty = |p_{s-1,0}/q_{s-1}| < 1$ will be sought.

In order to provide a graphic representation of the extension of the stability region associated to the IMEXRK schemes here derived, we introduce two additional stability functions: one, denoted as σ^{I}, associated to the term treated implicitly and the other, σ^{E} for the term integrated explicitly. These stability functions are obtained from σ in (3.13) as $\sigma^{\mathrm{I}}(z) = \sigma(z^{\mathrm{I}}, 0)$ and $\sigma^{\mathrm{E}}(z) = \sigma(0, z^{\mathrm{E}})$. In this way, linear stability of each IMEXRK scheme can be assessed by verifying that stability condition $|\sigma(z)| \leq 1$ is satisfied for both implicit and explicit components, independently.

3.2 Three-step incremental IMEXRK schemes

As already pointed out in [1] and also mentioned in the introduction, the nine degrees of freedom offered by a three-step incremental IMEXRK scheme do not allow to satisfy the nine nonlinear equations required in order to impose third order accuracy. The reason appears clear after resorting to the Butcher formulation, i.e.

$$
\begin{array}{c|cccc}
0 & 0 & & & \\
c_2 & b_1^{\mathrm{I}} & a_{2,2}^{\mathrm{I}} & & \\
c_3 & b_1^{\mathrm{I}} & b_2^{\mathrm{I}} & a_{3,3}^{\mathrm{I}} & \\
1 & b_1^{\mathrm{I}} & b_2^{\mathrm{I}} & b_3^{\mathrm{I}} & b_4^{\mathrm{I}} \\
\hline
 & b_1^{\mathrm{I}} & b_2^{\mathrm{I}} & b_3^{\mathrm{I}} & b_4^{\mathrm{I}}
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & & & \\
c_2 & a_{2,1}^{\mathrm{E}} & 0 & & \\
c_3 & b_1^{\mathrm{E}} & a_{3,2}^{\mathrm{E}} & 0 & \\
1 & b_1^{\mathrm{E}} & b_2^{\mathrm{E}} & b_3^{\mathrm{E}} & 0 \\
\hline
 & b_1^{\mathrm{E}} & b_2^{\mathrm{E}} & b_3^{\mathrm{E}} & 0
\end{array}
\tag{3.16}
$$

Solving the nonlinear system composed by the eight constraints $\tau_1^{(1)\mathrm{I,E}} = 0$, $\tau_1^{(2)\mathrm{I,E}} = 0$, $\tau_1^{(3)\mathrm{E}} = 0$, $\tau_2^{(3)\mathrm{IE,EI,EE}} = 0$, together with SO1 condition, the last constraint $\tau_2^{(3)\mathrm{II}} = 0$ can be expressed as a nonlinear function in the only parameter c_3. Solving this equation leveraging a symbolic solver like Mathematica [22], gives 13 distinct roots, all of them having a nonzero imaginary component. However, we can choose c_3 in order to minimize the residual of $\tau_2^{(3)\mathrm{II}}$ by solving $d\tau_2^{(3)\mathrm{II}}(c_3)/dc_3 = 0$, under the constraint that all the parameters are real-valued. This is achieved if the radicand $17c_3^2 - 60c_3^3 + 84c_3^4 - 48c_3^5$ is non-negative, which is verified for $c_3 \in [0, (7 - 11/\sqrt[3]{19 + 6\sqrt{47}} + \sqrt[3]{19 + 6\sqrt{47}})/12]$. The optimum within this interval is achieved for $c_3 \approx 0.616466445813626$. Approximating the solution to $c_3 = 5/8$, we obtain $A^{(3)} = |\tau_2^{(3)\mathrm{II}}| = 0.0179$, which is a slight improvement as compared to **IMEXRKiSMR**, for which $A^{(3)} = 37/1920 \approx 0.0193$. Furthermore, we have $\sigma_\infty = 0.34$, which reflects into better damping of the largest eigenvalues of the linear operator $\mathcal{L}$ in (3.1) with respect to **IMEXRKiSMR**, for which $\sigma_\infty = 87/185 \approx 0.47$. Both schemes prove to be superior to **CN/RKW3**, for which $A^{(3)} = 0.0353$, and $\sigma_\infty = 1$, being the scheme only A-stable. The analytic coefficients for the scheme thus derived, dubbed **IMEXRKiCB2(3s)**, are listed in Table 3.1 in both Butcher and incremental form. Figures 3.2e-f show the stability regions for the implicit and explicit part. We have to remark that the three-step formulation, together with the imposition of third order accuracy for the explicit component automatically fixes the shape of the stability region for explicit component, which appears unchanged with respect to **IMEXRKiSMR** and **CN/RKW3** (see Figures 3.2b and 3.2d).

At this point, third order accuracy could be achieved by either increasing the number of steps or by allowing one extra storage for the implicit part, due to the increased number of degrees of freedom. In particular, adding one step would allow to shape the stability region of the explicit component in order to extend it along the imaginary axis. At the same time, better accuracy and stability properties could be achieved. Interestingly, relaxing the storage requirements in the three-step scheme (3.2) for the implicit part does

Table 3.1: Optimal parameters for the second-order, strongly A-stable, three-step incremental IMEXRK scheme **IMEXRKiCB2(3s)**.

Butcher coefficients	
Parameter	Value
b_1^{I}	$(42861 - 752\sqrt{38})/129459$
b_2^{I}	$(6998 - 303\sqrt{38})/43153$
b_3^{I}	$8(1632 + 65\sqrt{38}))/43153$
b_4^{I}	$(26436 + 101\sqrt{38})/129459$
$a_{2,2}^{\mathrm{I}}$	$(2522730 - 164629\sqrt{38})/8803212$
$a_{3,3}^{\mathrm{I}}$	$(12405 + 1208\sqrt{38})/94152$
b_1^{E}	$(113 + \sqrt{38})/439$
b_2^{E}	$-3(1522 + 1245\sqrt{38})/97897$
b_3^{E}	$8(22 + \sqrt{38})/223$
$a_{2,1}^{\mathrm{E}}$	$(126 - 5\sqrt{38})/204$
$a_{3,2}^{\mathrm{E}}$	$(1291 - 8\sqrt{38})/3512$
c_2	$(126 - 5\sqrt{38})/204$
c_3	$5/8$

Incremental-form coefficients	
Parameter	Value
α_1^{I}	$(2522730 - 164629\sqrt{38})/8803212$
β_1^{I}	$(42861 - 752\sqrt{38})/129459$
α_2^{I}	$(12405 + 1208\sqrt{38})/94152$
β_2^{I}	$(-99558 + 9347\sqrt{38})/800292$
α_3^{I}	$(26436 + 101\sqrt{38})/129459$
β_3^{I}	$(176889 - 808\sqrt{38})/1035672$
β_1^{E}	$(126 - 5\sqrt{38})/204$
γ_1^{E}	0
β_2^{E}	$(1291 - 8\sqrt{38})/3512$
γ_2^{E}	$(-32262 + 2399\sqrt{38})/89556$
β_3^{E}	$8(22 + \sqrt{38})/223$
γ_3^{E}	$(-739 - 64\sqrt{38})/1784$

not allow to achieve full third order accuracy, since the associated system of nine accuracy constraints and SO1 condition admits only two families of solutions, one requiring $a^{\mathrm{I}}_{2,2} = 0$, the other $b^{\mathrm{I}}_4 = 0$, which is not acceptable since it would make one of the step explicit, thus compromising the stability properties of the algorithm.

3.3 Four-step incremental IMEXRK schemes

Adding one step in the incremental formulation allows to increase the number of design parameters from nine to twelve. The scheme thus obtained, in both incremental and Butcher form, appears as

$$
\begin{array}{c|ccccc}
0 & 0 & & & & \\
c_2 & b^{\mathrm{I}}_1 & a^{\mathrm{I}}_{2,2} & & & \\
c_3 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & a^{\mathrm{I}}_{3,3} & & \\
c_4 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & b^{\mathrm{I}}_3 & a^{\mathrm{I}}_{4,4} & \\
1 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & b^{\mathrm{I}}_3 & b^{\mathrm{I}}_4 & b^{\mathrm{I}}_5 \\
\hline
 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & b^{\mathrm{I}}_3 & b^{\mathrm{I}}_4 & b^{\mathrm{I}}_5
\end{array}
\qquad
\begin{array}{c|ccccc}
0 & 0 & & & & \\
c_2 & a^{\mathrm{E}}_{2,1} & 0 & & & \\
c_3 & b^{\mathrm{E}}_1 & a^{\mathrm{E}}_{3,2} & 0 & & \\
c_4 & b^{\mathrm{E}}_1 & b^{\mathrm{E}}_2 & a^{\mathrm{E}}_{4,3} & 0 & \\
1 & b^{\mathrm{E}}_1 & b^{\mathrm{E}}_2 & b^{\mathrm{E}}_3 & b^{\mathrm{E}}_4 & 0 \\
\hline
 & b^{\mathrm{E}}_1 & b^{\mathrm{E}}_2 & b^{\mathrm{E}}_3 & b^{\mathrm{E}}_4 & 0
\end{array}
$$

$$
\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}_n + \Delta t \left(\alpha^{\mathrm{I}}_1\, \mathcal{L}\, \mathbf{u}^{(1)} + \beta^{\mathrm{I}}_1\, \mathcal{L}\, \mathbf{u}_n + \beta^{\mathrm{E}}_1\, \mathcal{N}(\mathbf{u}_n)\right) \\
\mathbf{u}^{(2)} &= \mathbf{u}^{(1)} + \Delta t \left(\alpha^{\mathrm{I}}_2\, \mathcal{L}\, \mathbf{u}^{(2)} + \beta^{\mathrm{I}}_2\, \mathcal{L}\, \mathbf{u}^{(1)} + \beta^{\mathrm{E}}_2\, \mathcal{N}(\mathbf{u}^{(1)}) + \gamma^{\mathrm{E}}_2\, \mathcal{N}(\mathbf{u}_n)\right) \\
\mathbf{u}^{(3)} &= \mathbf{u}^{(2)} + \Delta t \left(\alpha^{\mathrm{I}}_3\, \mathcal{L}\, \mathbf{u}^{(3)} + \beta^{\mathrm{I}}_3\, \mathcal{L}\, \mathbf{u}^{(2)} + \beta^{\mathrm{E}}_3\, \mathcal{N}(\mathbf{u}^{(2)}) + \gamma^{\mathrm{E}}_3\, \mathcal{N}(\mathbf{u}^{(1)})\right) \\
\mathbf{u}_{n+1} &= \mathbf{u}^{(3)} + \Delta t \left(\alpha^{\mathrm{I}}_4\, \mathcal{L}\, \mathbf{u}_{n+1} + \beta^{\mathrm{I}}_4\, \mathcal{L}\, \mathbf{u}^{(3)} + \beta^{\mathrm{E}}_4\, \mathcal{N}(\mathbf{u}^{(3)}) + \gamma^{\mathrm{E}}_4\, \mathcal{N}(\mathbf{u}^{(2)})\right).
\end{aligned}
\tag{3.17}
$$

In this way, all third order accuracy constraints can be enforced, with three degrees of freedom left for optimization. In order to simplify the solution of the nonlinear system of nine equations, the parameters c_2, c_3, and c_4 are chosen as free parameters. The design phase proceeds as follows: first, the c_i parameters are chosen within the range [0, 1]. Then, conditions $\tau_1^{(1)\mathrm{E}} = 0$, $\tau_1^{(2)\mathrm{E}} = 0$, and $\tau_1^{(3)\mathrm{E}} = 0$ are imposed, together with SO1 condition for the explicit component, in order to express $b^{\mathrm{E}}_{1,2,3}$ and all the $a^{\mathrm{I}}_{i,i}$ and $a^{\mathrm{E}}_{i,i-1}$ coefficients as a function of the other coefficients. Once the c_i coefficients have been chosen, these equations are linear in the b_i^{E} coefficients and are easily solved. The resulting expressions are replaced into $\tau_2^{(3)\mathrm{EE}} = 0$. This gives a second order equation in b_4^{E}, which gives two solutions. Replacing the values for b_4^{E} back into $\tau_1^{(1,2,3)\mathrm{E}} = 0$ gives two sets of solutions for the b_i^{E} parameters. After all b_i^{E} have been determined, their expressions are replaced into $\tau_2^{(3)\mathrm{EI}} = 0$ and $\tau_2^{(3)\mathrm{IE}} = 0$. These two equations, together with $\tau_1^{(1)\mathrm{I}} = 0$, $\tau_1^{(2)\mathrm{I}} = 0$, and SO1 for the implicit part, are linear in the b_i^{I} parameters and are used to determine $b^{\mathrm{I}}_{1,2,3,4}$. Substitution of the resulting expressions into the last constraint $\tau_2^{(3)\mathrm{II}} = 0$ gives a second order equation in b_5^{I}. Notice that, since we obtained two sets of parameters, one for each choice of b_4^{E}, we actually have two second order equations to be solved. Therefore, we for each choice of c_i, we obtain four sets of parameters. Among the possible choices for the c_i parameters, we retain only those that satisfy the following criteria:

- All the Butcher coefficients are small and real-valued
- All the diagonal terms of the implicit scheme $a^{\mathrm{I}}_{i,i}$ are positive and less than $1/2$
- All the accuracy constraints are linearly independent
- The imaginary stability of the explicit part is close enough to the achievable limit
- The scheme is at least strongly A-stable with $\sigma_\infty < 1/10$
- The fourth-order truncation error $A^{(4)}$ is less than $1/10$

Regarding stability, the function $\sigma^{\mathrm{E}}(z)$, after imposing third order accuracy, appears as

$$\begin{aligned}\sigma^{\mathrm{E}}(z) &= 1 + z \sum_i b_i^{\mathrm{E}} + z^2 \sum_i b_i^{\mathrm{E}}\, c_i + z^3 \sum_{i,j} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{E}}\, c_j + z^4 \sum_{i,j,k} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{E}}\, a_{j,k}^{\mathrm{E}}\, c_k \\ &= 1 + z + z^2/2 + z^3/6 + \delta\, z^4.\end{aligned} \tag{3.18}$$

The stability region $|\sigma^{\mathrm{E}}(z)| \leq 1$ achieves the maximum extension along the imaginary axis for $\delta = 1/24$, with a value of $2\sqrt{2}$. Compared to the three-step incremental schemes, this corresponds to an improvement of more than 60%. Since this choice for δ also satisfies the fourth-order constraint $\tau_4^{(4)\mathrm{EEE}} = 0$, the stability region is the same as the classical fourth-order explicit RK algorithm. It is important to observe that any value of δ greater that $1/24$ leads to a scheme which is unstable for very small eigenvalues along the imaginary axis. For this reason, only those sets of c_i coefficients for which δ is sufficiently close to $1/24$ without exceeding are considered.

Based on these guidelines, extensive search over the three-dimensional parameter space lead to the coefficients of the four-step incremental IMEXRK scheme dubbed **IMEXRKiCB3(4s)**, which are reported in Table 3.2. We have to remark that, although these coefficients are available in analytical form, the extreme length of their expression makes them quite impractical to implement. For this reason, as done also in [26] and [27], a rational expression accurate up to 24 digits is preferred. Figures 3.2g-h show the stability regions of **IMEXRKiCB3(4s)** for both the implicit and explicit part. The scheme obtained is strongly A-stable with $\sigma_\infty = 0.0325$ and the imaginary stability limit for the explicit part is 2.7838, which is only 2.5% less than the achievable maximum. The truncation error is $A^{(4)} = 0.0592$.

We want to remark that another exploratory search was conducted among the two- and one-parameter subfamilies of IMEXRK schemes arising for those choices of c_i caus-

Table 3.2: Optimal parameters for the third-order, strongly A-stable, four-step incremental IMEXRK scheme **IMEXRKiCB3(4s)**.

Butcher coefficients	
Parameter	Value
b^{I}_1	268403570813/1046659493064
b^{I}_2	539124791465/1721977093901
b^{I}_3	−197050443577/700240830834
b^{I}_4	239563607837/443403175235
b^{I}_5	204443804709/1191419405951
$a^{\mathrm{I}}_{2,2}$	147427810807/485660101531
$a^{\mathrm{I}}_{3,3}$	243165146010/1055051926313
$a^{\mathrm{I}}_{4,4}$	514970586192/1250290449433
b^{E}_1	1450061836715/5978969592807
b^{E}_2	106792727210/477274043037
b^{E}_3	7353068969/671689278676
b^{E}_4	253095336536/484142576807
$a^{\mathrm{I}}_{2,1}$	14/25
$a^{\mathrm{I}}_{3,2}$	798923023415/1433115308036
$a^{\mathrm{I}}_{4,3}$	223463754637/956128100809
c_2	14/25
c_3	4/5
c_4	7/10

Incremental-form coefficients	
Parameter	Value
α^{I}_1	147427810807/485660101531
β^{I}_1	268403570813/1046659493064
α^{I}_2	243165146010/1055051926313
β^{I}_2	20920302827/2196806104873
α^{I}_3	514970586192/1250290449433
β^{I}_3	−216678405507/423298589287
α^{I}_4	204443804709/1191419405951
β^{I}_4	74577069499/580804002576
β^{E}_1	14/25
γ^{E}_1	0
β^{E}_2	798923023415/1433115308036
γ^{E}_2	−206225727739/649585186686
β^{E}_3	223463754637/956128100809
γ^{E}_3	−226857275186/679788613965
β^{E}_4	253095336536/484142576807
γ^{E}_4	−190080827984/853259476461

ing some of the accuracy constraints to become linearly dependent, but no improvement was found with respect to **IMEXRKiCB3(4s)**. Similar results were obtained for the other schemes developed in this chapter.

Since it was not possible to achieve L-stability within the four-step framework without compromising $A^{(4)}$ and imaginary stability of the explicit component, we extended the search in two different directions: first by considering extra storage for the implicit component, then by allowing another extra step in the incremental formulation. The first approach assumes the following scheme

$$
\begin{array}{c|ccccc}
0 & 0 & & & & \\
c_2 & a^{\mathrm{I}}_{2,1} & a^{\mathrm{I}}_{2,2} & & & \\
c_3 & b^{\mathrm{I}}_1 & a^{\mathrm{I}}_{3,2} & a^{\mathrm{I}}_{3,3} & & \\
c_4 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & a^{\mathrm{I}}_{4,3} & a^{\mathrm{I}}_{4,4} & \\
1 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & b^{\mathrm{I}}_3 & b^{\mathrm{I}}_4 & b^{\mathrm{I}}_5 \\
\hline
 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & b^{\mathrm{I}}_3 & b^{\mathrm{I}}_4 & b^{\mathrm{I}}_5
\end{array}
\qquad
\begin{array}{c|ccccc}
0 & 0 & & & & \\
c_2 & a^{\mathrm{E}}_{2,1} & 0 & & & \\
c_3 & b^{\mathrm{E}}_1 & a^{\mathrm{E}}_{3,2} & 0 & & \\
c_4 & b^{\mathrm{E}}_1 & b^{\mathrm{E}}_2 & a^{\mathrm{E}}_{4,3} & 0 & \\
1 & b^{\mathrm{E}}_1 & b^{\mathrm{E}}_2 & b^{\mathrm{E}}_3 & b^{\mathrm{E}}_4 & 0 \\
\hline
 & b^{\mathrm{E}}_1 & b^{\mathrm{E}}_2 & b^{\mathrm{E}}_3 & b^{\mathrm{E}}_4 & 0
\end{array}
$$

$$
\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}_n + \Delta t \left(\alpha^{\mathrm{I}}_1 \,\mathcal{L}\,\mathbf{u}^{(1)} + \beta^{\mathrm{I}}_1 \,\mathcal{L}\,\mathbf{u}_n + \beta^{\mathrm{E}}_1 \,\mathcal{N}(\mathbf{u}_n)\right) \\
\mathbf{u}^{(2)} &= \mathbf{u}^{(1)} + \Delta t \left(\alpha^{\mathrm{I}}_2 \,\mathcal{L}\,\mathbf{u}^{(2)} + \beta^{\mathrm{I}}_2 \,\mathcal{L}\,\mathbf{u}^{(1)} + \gamma^{\mathrm{I}}_2 \,\mathcal{L}\,\mathbf{u}_n + \beta^{\mathrm{E}}_2 \,\mathcal{N}(\mathbf{u}^{(1)}) + \gamma^{\mathrm{E}}_2 \,\mathcal{N}(\mathbf{u}_n)\right) \\
\mathbf{u}^{(3)} &= \mathbf{u}^{(2)} + \Delta t \left(\alpha^{\mathrm{I}}_3 \,\mathcal{L}\,\mathbf{u}^{(3)} + \beta^{\mathrm{I}}_3 \,\mathcal{L}\,\mathbf{u}^{(2)} + \gamma^{\mathrm{I}}_3 \,\mathcal{L}\,\mathbf{u}^{(1)} + \beta^{\mathrm{E}}_3 \,\mathcal{N}(\mathbf{u}^{(2)}) + \gamma^{\mathrm{E}}_3 \,\mathcal{N}(\mathbf{u}^{(1)})\right) \\
\mathbf{u}_{n+1} &= \mathbf{u}^{(3)} + \Delta t \left(\alpha^{\mathrm{I}}_4 \,\mathcal{L}\,\mathbf{u}_{n+1} + \beta^{\mathrm{I}}_4 \,\mathcal{L}\,\mathbf{u}^{(3)} + \gamma^{\mathrm{I}}_4 \,\mathcal{L}\,\mathbf{u}^{(2)} + \beta^{\mathrm{E}}_4 \,\mathcal{N}(\mathbf{u}^{(3)}) + \gamma^{\mathrm{E}}_4 \,\mathcal{N}(\mathbf{u}^{(2)})\right),
\end{aligned}
\tag{3.19}
$$

in which the Butcher tableau for the implicit part has the same low-storage structure of the [2*R*] IMEXRK schemes in [26], except for the coordination constraint. This formulation provides 15 design parameters to satisfy nine nonlinear equations for third order accuracy, plus the L-stability condition $\sigma_\infty = 0$. As already shown in [26], L-stability is easily achieved by suppressing the first column of the Butcher tableau associated to the implicit part, i.e. by setting $b_1^{\mathrm{I}} = a_{2,1}^{\mathrm{I}} = 0$. Among the remaining 13 parameters, c_2, c_3, c_4, and $a_{4,4}^{\mathrm{I}}$ are chosen as optimization variables, with each c_i in the range [0, 1], while $a_{4,4}^{\mathrm{I}} \in (0, 1/2]$. The design procedure is analogous to the one followed for **IMEXRKiCB3(4s)**. First, $\tau_1^{(1)\mathrm{E}} = 0$, $\tau_1^{(2)\mathrm{E}} = 0$, and $\tau_1^{(3)\mathrm{E}} = 0$, together with SO1 condition, are used to determine $b_{1,2,3}^{\mathrm{E}}$, $a_{2,2}^{\mathrm{I}}$, $a_{3,2}^{\mathrm{I}}$, $a_{4,3}^{\mathrm{I}}$, and all the $a_{i,i-1}^{\mathrm{E}}$ coefficients. After substitution, the quadratic equation $\tau_2^{(3)\mathrm{EE}} = 0$ is solved for b_4^{E}. The two families of solutions are replaced into $\tau_2^{(3)\mathrm{EI}} = 0$ and $\tau_2^{(3)\mathrm{IE}} = 0$. These two equations, together with $\tau_1^{(1)\mathrm{I}} = 0$, $\tau_1^{(2)\mathrm{I}} = 0$ and SO1 condition, are then used to calculate the coefficients $b_{2,3,4}^{\mathrm{I}}$ and $a_{3,3}^{\mathrm{I}}$. After substituting the previous expressions into the quadratic equation $\tau_2^{(3)\mathrm{II}} = 0$, we can finally solve for b_5^{I}.

Among the possible choices of the four design parameters, we chose that which guarantees real-valued coefficients and small diagonal terms for the implicit component, while offering the best compromise between imaginary stability of the explicit part and fourth-order truncation error $A^{(4)}$. The coefficients of the resulting scheme, dubbed **IMEXRKiCB3(4s+)**, are shown in Table 3.3. Figures 3.2i-j show the stability regions of such scheme for both implicit and explicit part. The imaginary stability limit for the explicit part is 2.8217, which is only 0.25% less than the theoretical limit, while the error is $A^{(4)} = 0.0698$, which is nearly 20% higher than **IMEXRKiCB3(4s)**. However, this result should not sound discouraging, since in the context of turbulent simulations, the extension of the imaginary stability for the explicit term, which is related to the CFL condition, and L-stability, which guarantees good damping of the largest eigenvalues of the terms treated implicitly, represent far more appealing features for a time stepping scheme.

Finally, we want to remark that the increased freedom given by allowing extra storage could be exploited to develop an incremental IMEXRK scheme with ESDIRK con-

Table 3.3: Optimal parameters for the third-order, L-stable, four-step incremental IMEXRK scheme **IMEXRKiCB3(4s+)**.

Butcher coefficients	
Parameter	Value
b^{I}_1	0
b^{I}_2	$4078465402807/5118992086463$
b^{I}_3	$-1068609889687/1488061735778$
b^{I}_4	$94533336407/126055292720$
b^{I}_5	$112416685574/655665149019$
$a^{\mathrm{I}}_{2,1}$	0
$a^{\mathrm{I}}_{2,2}$	$9/25$
$a^{\mathrm{I}}_{3,2}$	$379490756215/588608184103$
$a^{\mathrm{I}}_{3,3}$	$81921593785/419520366036$
$a^{\mathrm{I}}_{4,3}$	$-39458195936308/94684797045601$
$a^{\mathrm{I}}_{4,4}$	$12/25$
b^{E}_1	$67447694372/739814670703$
b^{E}_2	$640712099409/1099358471078$
b^{E}_3	$-299809194319/611386646053$
b^{E}_4	$878905218902/1076559421011$
$a^{\mathrm{I}}_{2,1}$	$9/25$
$a^{\mathrm{I}}_{3,2}$	$869434674241/1161054947863$
$a^{\mathrm{I}}_{4,3}$	$359201878931/1930920984086$
c_2	$9/25$
c_3	$21/25$
c_4	$43/50$

Table 3.3: Optimal parameters for the third-order, L-stable, four-step incremental IMEXRK scheme **IMEXRKiCB3(4s+)** (continued from previous page).

Incremental-form coefficients	
Parameter	Value
α_1^{I}	9/25
β_1^{I}	0
γ_1^{I}	0
α_2^{I}	81921593785/419520366036
β_2^{I}	218263380385/766574524329
γ_2^{I}	0
α_3^{I}	12/25
β_3^{I}	−454484525049/742613847476
γ_3^{I}	149986191080/986708857737
α_4^{I}	112416685574/655665149019
β_4^{I}	170133979507/630276463600
γ_4^{I}	−267746892839/888373818197
β_1^{E}	9/25
γ_1^{E}	0
β_2^{E}	869434674241/1161054947863
γ_2^{E}	−436940426403/1625331138472
β_3^{E}	359201878931/1930920984086
γ_3^{E}	−210795378052/1269651340659
β_4^{E}	878905218902/1076559421011
γ_4^{E}	−180800545132/267297489417

dition, i.e. first-stage explicit, singly-diagonally implicit. This is a particularly appealing condition since at every step of the incremental algorithm a linear system with the same matrix on the LHS has to be solved, provided a constant time step is employed. An ESDIRK scheme would therefore allow to reuse the LU decomposition of such matrix after the first step, thus reducing the overall computational cost. Interestingly, it was found that imposing all $a^{\mathrm{I}}_{i,i}$ to be equal, together with third order accuracy constraints and SO1 condition, automatically leads to the unacceptable condition $a^{\mathrm{I}}_{i,i} = 0$, making the scheme explicit. However, we want to point out that the high number of degrees of freedom required in the simulation of turbulence generally makes the storage of the LU decomposition of the LHS quite impractical. In addition, most simulations are performed with a non-constant time step Δt, which is instead recomputed every few time steps based on the CFL limit condition.

3.4 Five-step incremental IMEXRK schemes

An incremental IMEXRK scheme with five steps has the following Butcher and incremental form

$$
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & b^{\mathrm{I}}_1 & a^{\mathrm{I}}_{2,2} & & & & \\
c_3 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & a^{\mathrm{I}}_{3,3} & & & \\
c_4 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & b^{\mathrm{I}}_3 & a^{\mathrm{I}}_{4,4} & & \\
c_5 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & b^{\mathrm{I}}_3 & b^{\mathrm{I}}_4 & a^{\mathrm{I}}_{5,5} & \\
1 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & b^{\mathrm{I}}_3 & b^{\mathrm{I}}_4 & b^{\mathrm{I}}_5 & b^{\mathrm{I}}_6 \\
\hline
 & b^{\mathrm{I}}_1 & b^{\mathrm{I}}_2 & b^{\mathrm{I}}_3 & b^{\mathrm{I}}_4 & b^{\mathrm{I}}_5 & b^{\mathrm{I}}_6
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a^{\mathrm{E}}_{2,1} & 0 & & & & \\
c_3 & b^{\mathrm{E}}_1 & a^{\mathrm{E}}_{3,2} & 0 & & & \\
c_4 & b^{\mathrm{E}}_1 & b^{\mathrm{E}}_2 & a^{\mathrm{E}}_{4,3} & 0 & & \\
c_5 & b^{\mathrm{E}}_1 & b^{\mathrm{E}}_2 & b^{\mathrm{E}}_3 & a^{\mathrm{E}}_{5,4} & 0 & \\
1 & b^{\mathrm{E}}_1 & b^{\mathrm{E}}_2 & b^{\mathrm{E}}_3 & b^{\mathrm{E}}_4 & b^{\mathrm{E}}_5 & 0 \\
\hline
 & b^{\mathrm{E}}_1 & b^{\mathrm{E}}_2 & b^{\mathrm{E}}_3 & b^{\mathrm{E}}_4 & b^{\mathrm{E}}_5 & 0
\end{array}
$$

$$
\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}_n + \Delta t \left(\alpha_1^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(1)} + \beta_1^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}_n + \beta_1^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}_n) \right) \\
\mathbf{u}^{(2)} &= \mathbf{u}^{(1)} + \Delta t \left(\alpha_2^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(2)} + \beta_2^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(1)} + \beta_2^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}^{(1)}) + \gamma_2^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}_n) \right) \\
\mathbf{u}^{(3)} &= \mathbf{u}^{(2)} + \Delta t \left(\alpha_3^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(3)} + \beta_3^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(2)} + \beta_3^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}^{(2)}) + \gamma_3^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}^{(1)}) \right) \\
\mathbf{u}^{(4)} &= \mathbf{u}^{(3)} + \Delta t \left(\alpha_4^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(4)} + \beta_4^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(3)} + \beta_4^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}^{(3)}) + \gamma_4^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}^{(2)}) \right) \\
\mathbf{u}_{n+1} &= \mathbf{u}^{(4)} + \Delta t \left(\alpha_5^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}_{n+1} + \beta_5^{\mathrm{I}}\, \mathcal{L}\, \mathbf{u}^{(4)} + \beta_5^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}^{(4)}) + \gamma_5^{\mathrm{E}}\, \mathcal{N}(\mathbf{u}^{(3)}) \right).
\end{aligned}
\tag{3.20}
$$

This gives 15 design parameters, which allow to impose third order accuracy, and L-stability for the implicit part. As done for **IMEXRKiCB3(4s)**, the c_i coefficients are chosen as free parameters to optimize the truncation error $A^{(4)}$ and extend the stability region of the explicit scheme over the imaginary axis. To this matter, the addition of one stage in the Butcher formulation has the direct effect of increasing by one the order of the polynomial associated to the stability of the explicit scheme, which, after imposing third order accuracy, appears as

$$
\begin{aligned}
\sigma^{\mathrm{E}}(z) &= 1 + z \sum_i b_i^{\mathrm{E}} + z^2 \sum_i b_i^{\mathrm{E}}\, c_i + z^3 \sum_{i,j} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{E}}\, c_j \\
&\quad + z^4 \sum_{i,j,k} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{E}}\, a_{j,k}^{\mathrm{E}}\, c_k + z^5 \sum_{i,j,k,l} b_i^{\mathrm{E}}\, a_{i,j}^{\mathrm{E}}\, a_{j,k}^{\mathrm{E}}\, a_{k,l}^{\mathrm{E}}\, c_l \\
&= 1 + z + z^2/2 + z^3/6 + \delta\, z^4 + \varepsilon\, z^5.
\end{aligned}
\tag{3.21}
$$

As done for **IMEXRKiCB3(4s)**, we choose $\delta = 1/24$, since this choice is equal to imposing $\tau_4^{(4)\mathrm{EEE}} = 0$. With this choice of δ, it was found that $\varepsilon = 1/144$ produces a stability region with the largest extension along the imaginary axis, i.e. $2\sqrt{3}$. Interestingly, any value of ε higher than $1/144$ results in a stability region which does not include the entire portion of the imaginary axis between the origin and the furthest intersection of said region with the imaginary axis. For this reason, during the process of designing the scheme coefficients, only those schemes with ε close enough but not exceeding $1/144$ are considered.

Coefficients derivation is carried out as follows: first, the four c_i coefficients are chosen in the range [0, 1]. Then, equations $\tau_1^{(1)\mathrm{E}} = 0$, $\tau_1^{(2)\mathrm{E}} = 0$, and $\tau_1^{(3)\mathrm{E}} = 0$ are solved, together with SO1 condition, for the coefficients $b^{\mathrm{E}}_{1,2,3}$, and all the $a^{\mathrm{I}}_{i,i}$ and $a^{\mathrm{E}}_{i,i-1}$. The resulting expressions are replaced into $\tau_2^{(3)\mathrm{EE}} = 0$, and $\tau_4^{(4)\mathrm{EEE}} = 0$. Solving this system of nonlinear equations gives six solutions for b^{E}_4 and b^{E}_5. Differently from **IMEXRKiCB3(4s)** and **IMEXRKiCB3(4s+)**, this step requires the solution of a sixth-order polynomial, hence an analytic expression for the corresponding Butcher coefficients is not available. The remaining coefficients are determined by first setting $b^{\mathrm{I}}_1 = 0$. This allows to suppress the first column of the Butcher tableau associated to the implicit part in (3.20) and, together with FSAL condition, it guarantees L-stability. Then, equations $\tau_2^{(3)\mathrm{IE}} = 0$ and $\tau_2^{(3)\mathrm{EI}} = 0$, after replacing the numerical values of b^{E}_i, are solved, together with $\tau_1^{(1)\mathrm{I}} = 0$, $\tau_1^{(2)\mathrm{I}} = 0$ and SO1 condition, in order to determine $b^{\mathrm{I}}_{2,3,4,5}$. Finally, the quadratic equation $\tau_2^{(3)\mathrm{II}} = 0$ is solved for b^{I}_6. Overall, twelve solutions are obtained for each choice of c_i.

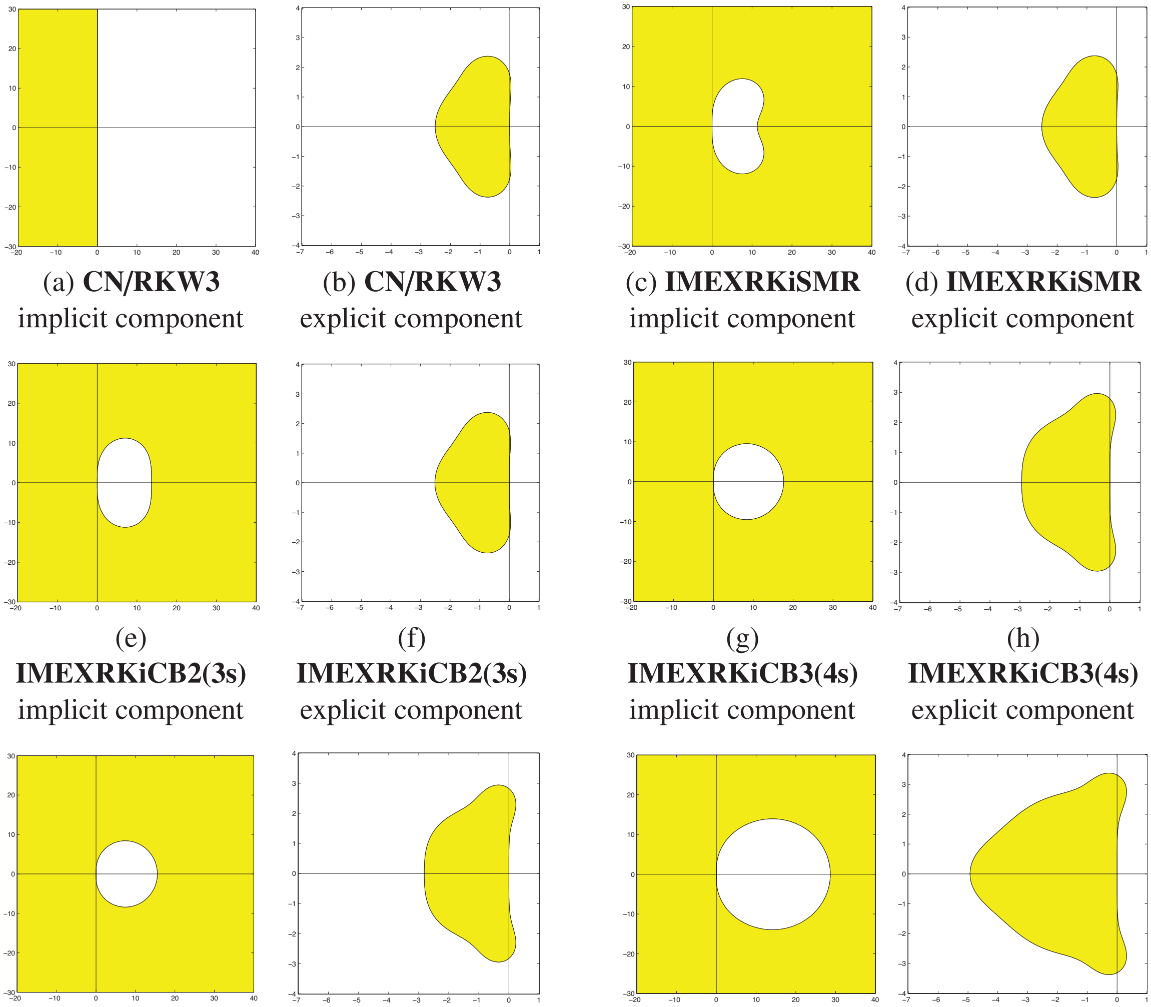
Search over the 4-dimensional parameter space gives the five-step scheme **IMEXRKiCB3(5s)** reported in Table 3.4. The truncation error norm $A^{(4)}$ equals 0.0121, which is five times smaller than **IMEXRKiCB3(4s)**. The stability regions for the implicit and explicit part are reported in Figures 3.2k-l. The imaginary stability limit for the explicit component is 3.3129, which is less than 5% away from the achievable limit. Remarkably, the imaginary stability of **IMEXRKiCB3(5s)** for the explicit part is nearly twice as big as that achievable within the three-step formulation.

Table 3.4: Optimal parameters for the third-order, L-stable, five-step incremental IMEXRK scheme **IMEXRKiCB3(5s)**.

Butcher coefficients	
Parameter	Value
b_1^{I}	0
b_2^{I}	637517882999/1527388578735
b_3^{I}	−227762052797/1215799173540
b_4^{I}	523301946593/1108095937880
b_5^{I}	393321793971/1670714051336
b_6^{I}	30593761609/491309463172
$a_{2,2}^{\mathrm{I}}$	6/25
$a_{3,3}^{\mathrm{I}}$	541585733727/2432898737681
$a_{4,4}^{\mathrm{I}}$	315106973550/1086783771481
$a_{5,5}^{\mathrm{I}}$	116591638520/589766421481
b_1^{E}	581140573286/7425488636757
b_2^{E}	299446732045/1101443065238
b_3^{E}	−258748134110/1807994379809
b_4^{E}	451687329886/916200602845
b_5^{E}	294496188261/981711902785
$a_{2,1}^{\mathrm{I}}$	6/25
$a_{3,2}^{\mathrm{I}}$	154015187090/274176653309
$a_{4,3}^{\mathrm{I}}$	102238376128/601864533117
$a_{5,4}^{\mathrm{I}}$	529485677295/764067597889
c_2	6/25
c_3	16/25
c_4	13/25
c_5	9/10

Table 3.4: Optimal parameters for the third-order, L-stable, five-step incremental IMEXRK scheme **IMEXRKiCB3(5s)** (continued from previous page).

Incremental-form coefficients	
Parameter	Value
α_1^{I}	6/25
β_1^{I}	0
α_2^{I}	541585733727/2432898737681
β_2^{I}	87814798181/495035914552
α_3^{I}	315106973550/1086783771481
β_3^{I}	−888759388641/2167999316938
α_4^{I}	116591638520/589766421481
β_4^{I}	219266163916/1202718563581
α_5^{I}	30593761609/491309463172
β_5^{I}	21089212573/558948398641
β_1^{E}	6/25
γ_1^{E}	0
β_2^{E}	154015187090/274176653309
γ_2^{E}	−190760799409/1179450149947
β_3^{E}	102238376128/601864533117
γ_3^{E}	−310203039833/1070147534785
β_4^{E}	529485677295/764067597889
γ_4^{E}	−178427905715/570088596477
β_5^{E}	294496188261/981711902785
γ_5^{E}	−78529999193/392684761114



(a) **CN/RKW3** implicit component

(b) **CN/RKW3** explicit component

(c) **IMEXRKiSMR** implicit component

(d) **IMEXRKiSMR** explicit component

(e) **IMEXRKiCB2(3s)** implicit component

(f) **IMEXRKiCB2(3s)** explicit component

(g) **IMEXRKiCB3(4s)** implicit component

(h) **IMEXRKiCB3(4s)** explicit component

(i) **IMEXRKiCB3(4s+)** implicit component

(j) **IMEXRKiCB3(4s+)** explicit component

(k) **IMEXRKiCB3(5s)** implicit component

(l) **IMEXRKiCB3(5s)** explicit component

Figure 3.2: Stability regions for the incremental IMEXRK schemes considered in this chapter.

3.5 Numerical experiments

In order to verify the order of accuracy of the schemes here derived and assess their performance in the representation of some statistics of interest during the simulation of turbulence, the one-dimensional Kuramoto-Sivashinsky equation (KSE) is used

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^4 u}{\partial x^4}\right), \tag{3.22}$$

where $u(x, t)$ is the velocity field and ν is the viscosity. This equation is of particular interest since, like the NSE, it possesses a nonlinear convective term. Besides, it has a destabilizing second derivative and a stabilizing fourth derivative, which allow to generate and maintain a turbulent behavior. A periodic domain $x \in [0, 400]$ is considered and 1024 equally-spaced grid points are used for discretization. This allows to adopt a pseudo-spectral approach for the discretization of all spatial derivatives, with the convective term evaluated in conservative form, i.e. $u\,\partial u/\partial x = \partial(u^2/2)/\partial x$, with 2/3 rule for de-aliasing.

The four schemes developed in this chapter, together with **CN/RKW3** and **IMEX-RKiSMR** are used to time march the KSE in (3.22) over a prescribed time horizon, assuming $\nu = 0.5$ and the initial condition

$$u(x, 0) = \sin(\pi/2\, x) + \sin(3\pi/4\, x), \tag{3.23}$$

with a small Gaussian perturbation added in order to trigger chaotic behavior. The second- and fourth-derivatives are treated implicitly, while the convective term is advanced explicitly.

The first batch of simulations is performed in order to verify the theoretical or-

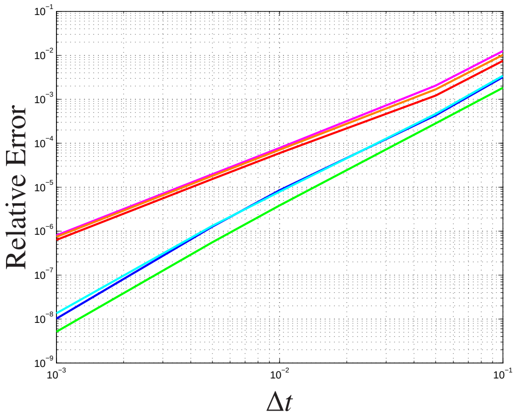


Figure 3.3: Relative error as a function of the step size for the schemes considered in this chapter when applied to the integration of KSE: **CN/RKW3** (red), **IMEXRKiSMR** (magenta), **IMEXRKiCB2(3s)** (orange), **IMEXRKiCB3(4s)** (blue), **IMEXRKiCB3(4s+)** (cyan), and **IMEXRKiCB3(5s)** (green).

der of accuracy. This is achieved by integrating the KSE over a time horizon of 10 time units with a constant time step ranging from 10^{-3} to 10^{-1}. Results are compared to a reference solution obtained by integrating the KSE using the fourth-order IMEXRK scheme **ARK4(3)6L[2]SA** [7] with a constant time step $\Delta t = 10^{-5}$. Results (see Figure 3.3) confirm the theoretical prediction, with all our schemes except **IMEXRKiCB2(3s)** achieving full third-order accuracy while **CN/RKW3** and **IMEXRKiSMR** being only second order overall. In this figure, relative error is defined as the square root of the sum of the squared difference between the computed solutions and the reference at each grid point, divided by the norm of the reference solution.

A second analysis we performed aims at assessing how different time stepping schemes affect the correct representation of average statistics. To this matter, we integrated the KSE over an horizon of 1000 time units, starting from initial conditions (3.23), and we used the time-averaged energy spectrum for comparison. The velocity field $u(x, t)$ obtained is shown in Figure 3.4.

Three different settings have been considered: the first one assumes a constant step

Table 3.5: Summary of the properties of the six incremental IMEXRK schemes considered in the chapter.

Scheme	Steps	Extra storage	Accuracy order	σ_∞	$\mathrm{CFL}_{\mathrm{lim}}$	$\mathrm{CFL}_{\mathrm{lim}}$ / Steps	Truncation error
CN/RKW3	3	-	2	1.00	1.7321	0.577	$A^{(3)} = 0.0353$
IMEXRKiSMR	3	-	2	0.47	1.7321	0.577	$A^{(3)} = 0.0193$
IMEXRKiCB2(3s)	3	No	2	0.34	1.7321	0.577	$A^{(3)} = 0.0179$
IMEXRKiCB3(4s)	4	No	3	0.0325	2.7838	0.696	$A^{(4)} = 0.0592$
IMEXRKiCB3(4s+)	4	Yes	3	0.0	2.8217	0.705	$A^{(4)} = 0.0698$
IMEXRKiCB3(5s)	5	No	3	0.0	3.3129	0.663	$A^{(4)} = 0.0121$

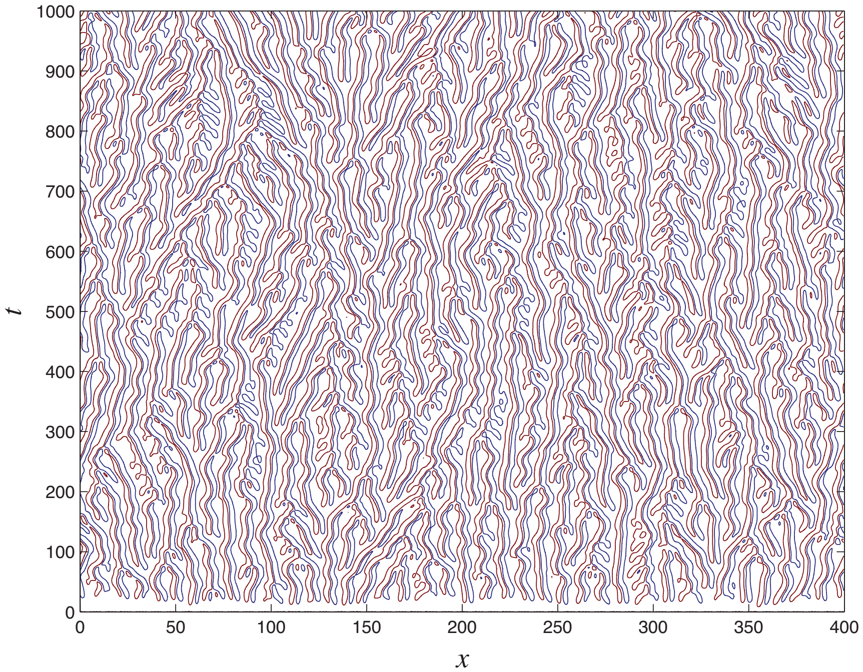


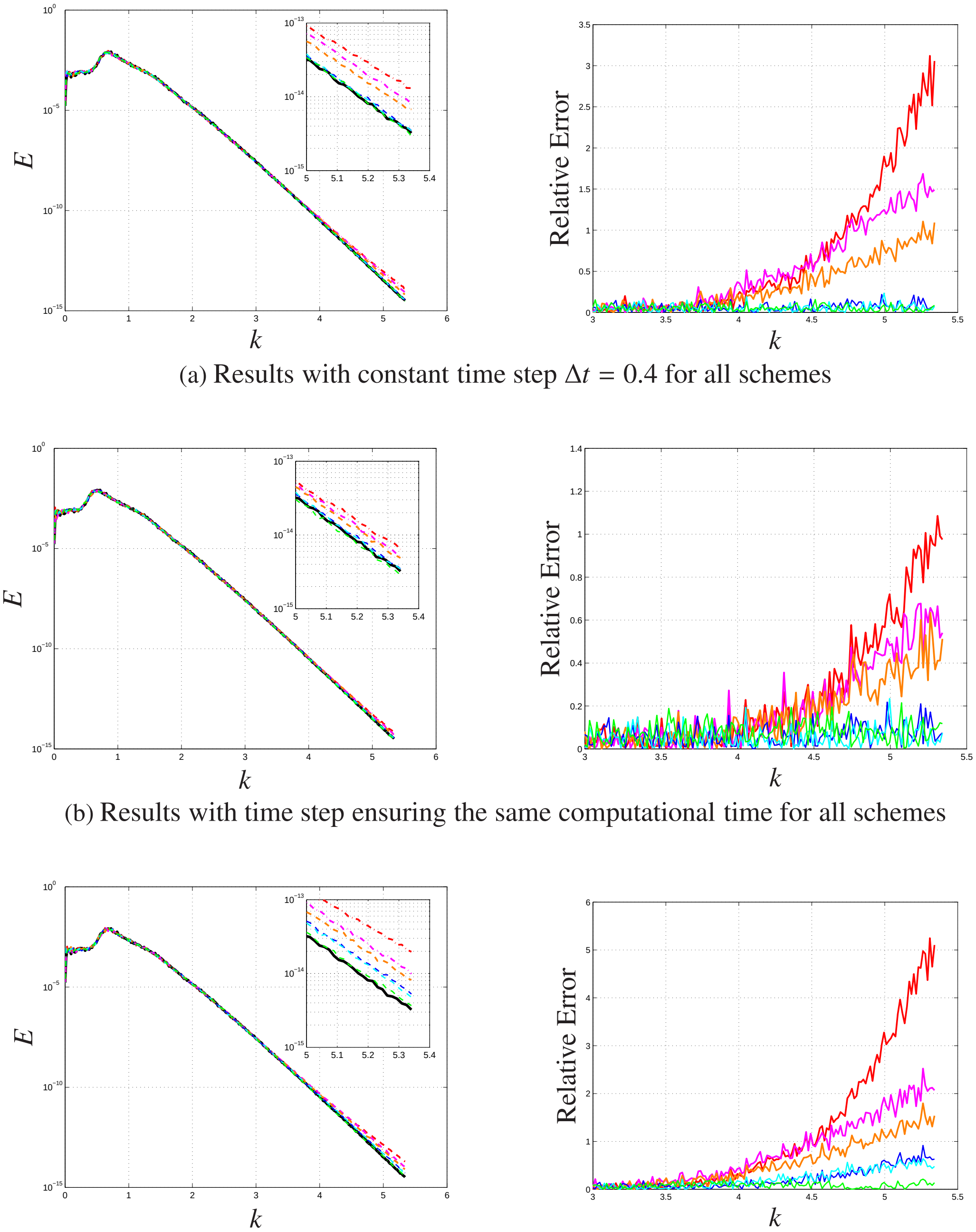
Figure 3.4: Isocontours −0.5 and +0.5 of the velocity field $u(x,\ t)$ propagated through the KSE over an horizon of 1000 time units.

size $\Delta t = 0.4$, which proves to be stable for all the schemes. Then, simulations are performed using the maximum time step allowed at each iteration by the CFL limit condition, i.e.

$$\Delta t = \mathrm{CFL}_{\mathrm{lim}}\ \min \Delta x/|u_i| \tag{3.24}$$

where $\mathrm{CFL}_{\mathrm{lim}}$ corresponds to the imaginary stability limit of the explicit component for each incremental scheme. The last set of simulations is performed considering a constant Δt for each scheme such that the computational time for all simulations is the same, i.e. $\Delta t = 0.3$ for **CN/RKW3**, **IMEXRKiSMR**, and **IMEXRKiCB2(3s)**, $\Delta t = 0.4$ for **IMEXRKiCB3(4s)** and **IMEXRKiCB3(4s+)**, and $\Delta t = 0.5$ for **IMEXRKiCB3(5s)**. The average energy spectrum used as reference is generated after marching the KSE with **ARK4(3)6L[2]SA** using a constant time step $\Delta t = 10^{-3}$.

Results show that not only our schemes outperform the schemes in literature when



(a) Results with constant time step $\Delta t = 0.4$ for all schemes

(b) Results with time step ensuring the same computational time for all schemes

(c) Results with step size recalculated at each iteration using the CFL limit of each scheme

Figure 3.5: Average energy spectrum (left) of the KSE solution propagated using different time stepping schemes and relative error (right) with respect to the reference solution: **CN/RKW3** (red), **IMEXRKiSMR** (magenta), **IMEXRKiCB2(3s)** (orange), **IMEXRKiCB3(4s)** (blue), **IMEXRKiCB3(4s+)** (cyan), and **IMEXRKiCB3(5s)** (green). Reference solution is shown in the left figures with a solid black line.

the same step size is considered (Figure 3.5a) due to the improved order of accuracy, but they reveal superior even when the same computational time for all schemes is considered (Figure 3.5b), or when the CFL limit is used to calculate the time step at each iteration (Figure 3.5c). In particular, **IMEXRKiCB2(3s)** slightly outperforms the schemes from literature in all three tests, while a more substantial difference can be appreciated for the three third order schemes, which show comparable results when the same time step is used. Similar results are obtained for the simulations performed considering the same computational time, with **IMEXRKiCB3(5s)** showing slightly worse performance with respect to the other third order schemes. Interestingly, **IMEXRKiCB3(5s)** shows the best performance for those tests in which the time step is calculated according to the CFL limit, with a significant difference with respect to **IMEXRKiCB3(4s)** and **IMEXRKiCB3(4s+)**. This is somehow unexpected since the increased CFL limit of **IMEXRKiCB3(5s)** comports bigger time steps. However, the reduction in the truncation error with respect to the four-step schemes is such that no degradation of performance is observed. Nonetheless, the increased computational cost that a five-step scheme requires with respect to performing just three or four steps must be accounted for. To this matter, we introduced a measure of efficiency for these family of schemes, defined as the ratio between the CFL limit and the number of steps required. As shown in Table 3.5, the efficiency for the four-step schemes is over 20% higher than that of the three-step schemes, while the five-step scheme **IMEXRKiCB3(5s)**, despite offering the highest CFL limit, has an efficiency which is slightly lower. This means that for those simulations in which only the steady-state value of some average statistics is the main concern, **IMEXRKiCB3(4s)** or **IMEXRKiCB3(4s+)** should be preferred, while **IMEXRKiCB3(5s)** represents the optimal choice when the primary interest is in the accuracy of the simulation.

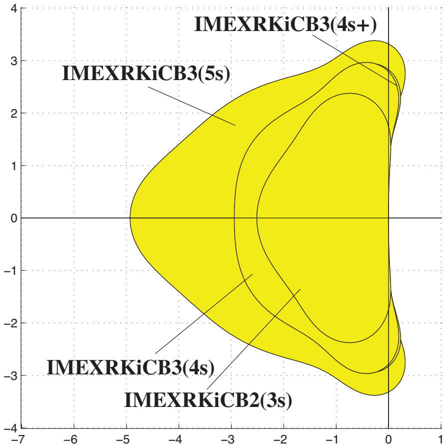


Figure 3.6: Comparison of stability regions of the explicit component for the four schemes derived in this chapter.

3.6 Conclusions

We have developed four new incremental IMEXRK schemes for the time integration of the NSE. Their features and properties are summarized in Table 3.5. All such schemes prove to outperform the only two other incremental IMEXRK schemes available in literature, i.e. **CN/RKW3** and **IMEXRKiSMR**. In particular, they offer better overall accuracy, increased CFL limit (see Figure 3.6 for a comparison), and better asymptotic stability for the implicit part. Remarkably, this is achieved while retaining the same storage requirements, except for the case of **IMEXRKiCB3(4s+)**, which requires one additional register. Simulations leveraging the KSE show that not only these schemes improve the overall time accuracy of the solution but they also provide a more precise representation of the average energy spectrum at the highest wavenumbers, as compared to the schemes from literature. In particular, these schemes prove superior not only considering the same time step, but also accounting for the increased computational time their implementation requires. Even simulations with a time step based on the CFL limit of each scheme provide better results. However, it was also observed that the increased CFL limit justifies the increased computational cost when switching from a three-step (such as **IMEXRKiCB2(3s)**) to a four-step scheme (such as **IMEXRKiCB3(4s)** or **IMEXRKiCB3(4s+)**), while the additional gain obtained leveraging a five-step scheme (i.e. **IMEXRKiCB3(5s)**), is overshadowed by the extra computational cost its implementation requires.

Finally, one could argue that better performances could be achieved with a six-step scheme, since three extra degrees of freedom could be leveraged to improve the overall accuracy and the CFL limit. However, the 18 free parameters are not enough to satisfy all 21 constraints needed to achieve fourth order accuracy. Nonetheless, a reduction in the truncation error could be expected. To this matter, we want to point out that **IMEXRKiCB3(5s)** is already ten times smaller than the error of the low-storage IMEXRK schemes presented in [26] and six time smaller than the state-of-the-art full-storage third-order IMEXRK scheme in Kennedy et al. [7]. Besides, the gain in the CFL limit would not justify the additional step, as already observed when comparing four- and five-step schemes.

Future work involves the implementation of the schemes here developed into a DNS code for the simulation of a fully-developed turbulent channel flow. Results will be compared to the schemes currently employed in literature, i.e. **CN/RKW3** and **IMEXRK-iSMR**. However, we want to remark that while the implementation of such schemes appear straightforward in case a velocity-vorticity formulation is adopted when discretizing the NSE, more effort is instead needed in case a fractional step implementation is sought. In particular, the splitting of the pressure term, which introduces a second order error and which does not appear in the velocity-vorticity formulation, is acceptable whenever an overall second-order time stepping scheme, such as **CN/RKW3**, is used, while it turns out to dominate the time discretization error when higher order schemes are employed, as we intend to do. For this reason, a third order correction term accounting for this splitting error needs to be introduced, like the one proposed in [32], in order to recover the actual order of accuracy.

Acknowledgements

This chapter contains work previously published in:

- D. Cavaglieri, T.R. Bewley, “Low-storage implicit/explicit Runge-Kutta schemes for the simulation of the Navier-Stokes Equations, Part 1: Theory", *Submitted to Journal of Computational Physics*, 2016

Chapter 4

Tweed and box relaxation: improved smoothing algorithms for multigrid solution of elliptic PDEs on stretched structured grids

4.1 Introduction

Geometric multigrid methods are among the fastest techniques available for the numerical solution of the large linear systems arising from the high-resolution discretization of elliptic PDEs on structured grids [33], and can generally be implemented efficiently on parallel architectures [34]. Multigrid methods have also been extended to handle certain non-elliptic PDEs (see [35] and the references in it). Appropriate relaxation schemes for the smoothing step are essential to accelerate the convergence of multigrid methods. This chapter examines two new relaxation schemes that are well suited for multigrid methods on stretched grids.

A typical model problem used to evaluate multigrid performance is the solution of the elliptic 2D Poisson equation with Dirichlet boundary conditions, written here in its general (heterogenous, anisotropic) form:

$$\frac{\partial}{\partial x_i}\Big(\sigma_{ij}\frac{\partial u}{\partial x_j}\Big) = f(x,y) \quad \text{in } \Omega = [0,L_x]\times[0,L_y], \tag{4.1a}$$

$$u = g(x,y) \quad \text{on } \partial\Omega, \tag{4.1b}$$

where x and y are the spatial coordinates, Ω is the domain of interest, $\partial\Omega$ is the boundary of Ω, and the matrix with components $\sigma_{ij}(x,y)$ is symmetric positive definite. In isotropic media, $\sigma_{ij}(x,y) = c(x,y)\,\delta_{ij}$. In homogeneous media, $\sigma_{ij}(x,y)$ is constant in x and y. A system of this form may be isotropic, homogeneous, both, or neither.

To demonstrate our new method, we will consider the discretization of (4.1) in the homogeneous isotropic case, with $\sigma_{ij}(x,y) = \delta_{ij}$, on a stretched rectilinear grid in which all cells are rectangles (in 2D) or rectangular cuboids (in 3D); the extension of this method to anisotropic, inhomogeneous systems, curvilinear grids, other elliptic PDEs, and other boundary conditions follows using standard methods. Discretization of this problem on an $(n_x+1)\times(n_y+1)$ stretched grid using a second-order central finite difference method, with x_i and y_j denoting the grid coordinates in the x and y directions, respectively, and $u_{i,j}$ denoting the discretized value of u at the $\{i, j\}$ gridpoint, leads to a five-point discretization of the Laplacian:

$$W_i\,u_{i-1,j} + E_i\,u_{i+1,j} + S_j\,u_{j-1,j} + N_j\,u_{i,j+1} + C_{i,j}\,u_{i,j} = f_{i,j}, \quad i = 2,\ldots,n_x, \quad j = 2,\ldots,n_y, \tag{4.2a}$$

$$u_{1,*},\ u_{n_x+1,*},\ u_{*,1},\ u_{*,n_y+1} \text{ specified,} \tag{4.2b}$$

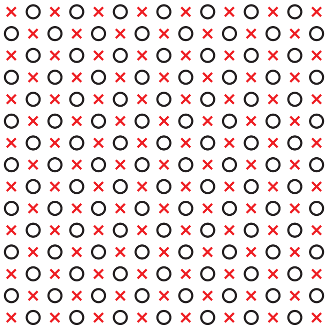


Figure 4.1: Gridpoint arrangement for checkerboard smoothing.

where $\Delta x_{i-1/2} = x_i - x_{i-1}$, $\Delta y_{j-1/2} = y_j - y_{j-1}$, $\Delta x_i = (\Delta x_{i-1/2} + \Delta x_{i+1/2})/2$, $\Delta y_j = (\Delta y_{j-1/2} + \Delta y_{j+1/2})/2$, and

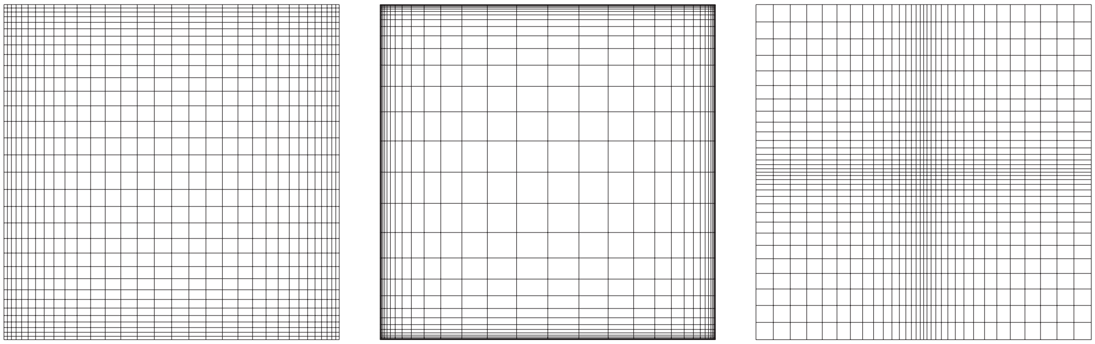
$$W_i = \frac{1}{\Delta x_i \Delta x_{i-1/2}}, \quad E_i = \frac{1}{\Delta x_i \Delta x_{i+1/2}}, \quad S_j = \frac{1}{\Delta y_j \Delta y_{j-1/2}}, \quad N_j = \frac{1}{\Delta y_j \Delta y_{j+1/2}},$$

and $C_{i,j} = -(W_i + E_i + S_j + N_j)$. It is thus seen that, even if the PDE is homogeneous and isotropic, grid stretching causes the discretized Poisson equation (4.2) to be inhomogeneous (with coefficients varying as a function of position) and anisotropic (with coefficients varying as a function of direction).

In the case of an unstretched grid (with Δx and Δy constant in both x and y), we have $W_i = E_i = 1/(\Delta x)^2$ and $S_j = N_j = 1/(\Delta y)^2$; that is, the discretization of the Laplacian becomes homogeneous. Further, if $\Delta x = \Delta y$, the discretization of the Laplacian also becomes isotropic, with $W_i = E_i = S_j = N_j$. As is well known (see [35]), for the case with no grid stretching, standard smoothing approaches such as *checkerboard* (also called red-black point Gauss-Seidel) relaxation (see Figure 4.1) performs exceptionally well when applied within the multigrid framework (see, e.g., [36]). As discussed in §4.4, performance of checkerboard relaxation starts to decay significantly when grid stretching is introduced.

When grid stretching is performed, gridpoints are clustered (that is, denser) in some directions more than others in certain regions of the computational domain, and thus two-delta waves in the error of the solution (that is, gridpoint-to-gridpoint oscillations, which checkerboard smoothing proves most effective at eliminating) are at mismatched spatial wavenumbers in different spatial directions in these regions. When this happens, common wisdom [35] is that *zebra relaxation* (that is, line relaxation on lines of alternating “color”) should be applied in the direction orthogonal to the local direction of densest grid clustering. For example, in a square computational domain with grid clustering near the upper and lower boundaries, zebra relaxation along the lines which are orthogonal to these two boundaries proves to be quite effective, whereas zebra relaxation along lines in the opposite direction proves to be much less effective. In a 2D or 3D computational domain with grid clustering in *multiple* directions (see, e.g., Figure 4.2), zebra relaxation in one direction alone is ineffective. In such case, *alternating-direction zebra relaxation*, in which zebra relaxation is performed in each coordinate direction in succession (see Figure 4.3), is commonly the method used, and with it the rapid convergence of the multigrid approach may be recovered. Note, however, that with the alternating-direction zebra relaxation approach, in any given region, the lines upon which relaxation is performed are orthogonal to the local direction of densest grid clustering during only half of the sweeps in the 2D case, and during only a third of the sweeps in the 3D case.

Thus, rather than requiring two sweeps of zebra relaxation in 2D, or three sweeps of zebra relaxation in 3D, simply to get the relaxation lines used to be locally oriented to the local direction of densest grid clustering during a fraction of the sweeps, we instead suggest a more effective motif for the line smoother. The algorithm described in this chapter represents, we believe, the first attempt at developing a line smoother for multigrid relaxation which relaxes efficiently on branched lines that are locally-orthogonal to the local direc-



(a) Hyperbolic tangent stretching (4.7), $c = 1.5$

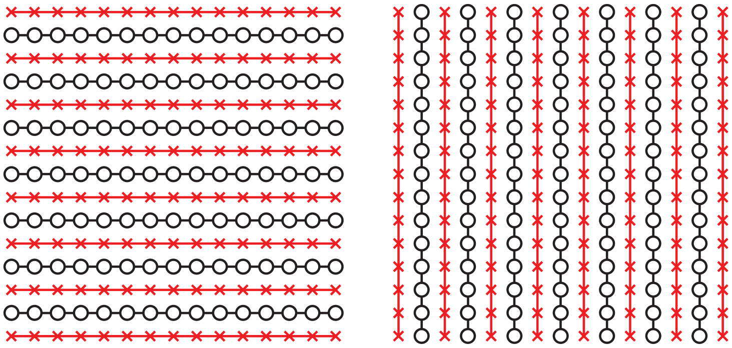
(b) Hyperbolic tangent stretching (4.7), $c = 3.0$

(c) Hyperbolic tangent stretching (4.8), $c = 1.5$

Figure 4.2: 2D stretched grids with clustering near the walls (a) and (b), and clustering near the center (c).

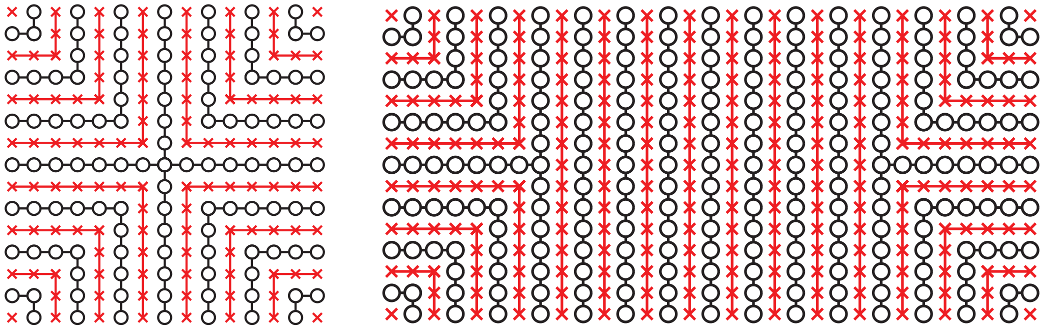
tions of densest grid clustering *everywhere* in the computational domain, even when grid clustering is applied in multiple directions, as indicated in Figure 4.2 for the 2D case.

Towards this end, two different smoothers have been developed, which we denote *tweed* and *box* relaxation. Tweed relaxation efficiently addresses the problem of near-wall grid clustering (see Figure 4.2a-b), whereas box relaxation addresses grid clustering near the center (see Figure 4.2c). The key idea behind the tweed motif (see Figure 4.4) is to perform relaxation in blocks of connecting lines arranged in such a way as to make such lines *everywhere perpendicular to the closest domain boundary*. Red-black alternation, applied in a zebra-like fashion, makes the relaxation of each block independent from the other blocks of the same color. The key idea behind the box motif (see Figure 4.5), in contrast, is to perform relaxation in blocks of connecting lines arranged in such a way as to make such lines *everywhere perpendicular to the closest coordinate plane through the center of the domain* (and which are, thus, everywhere *parallel* to the closest domain boundary). Again, red-black alternation, applied in a zebra-like fashion, makes the relaxation of each block independent from the other blocks of the same color. Note that the tweed and box motifs extend naturally to 3D, as illustrated in Figures 4.6 and 4.7, respectively.



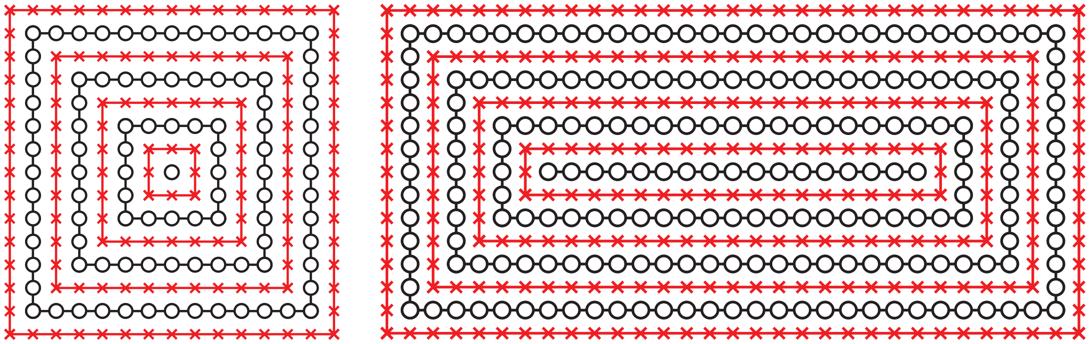
(a) x-direction smoothing (b) y-direction smoothing

Figure 4.3: Relaxation motifs for 2D alternating-direction zebra smoothing.



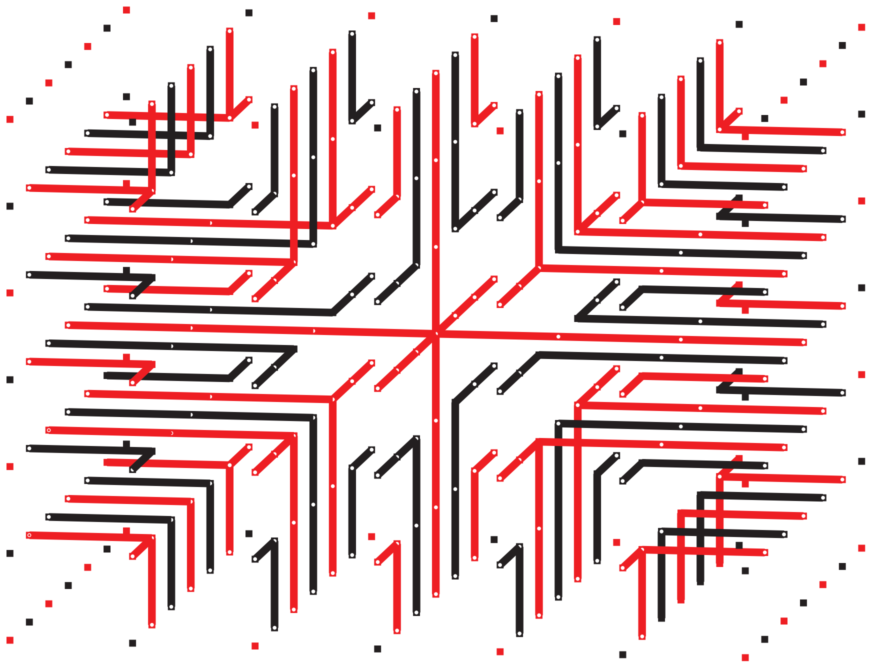
(a) Square grid ($n_x = n_y$) (b) Rectangular grid ($n_x > n_y$)

Figure 4.4: Relaxation motif for 2D tweed smoothing.

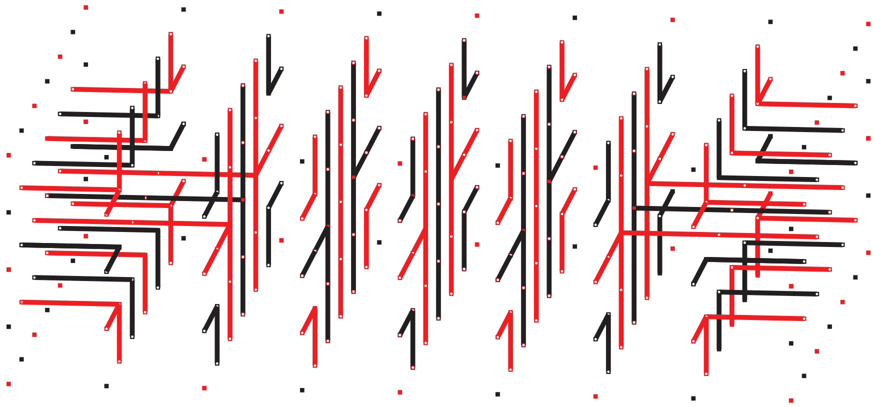


(a) Square grid ($n_x = n_y$) (b) Rectangular grid ($n_x > n_y$)

Figure 4.5: Relaxation motif for 2D box smoothing.

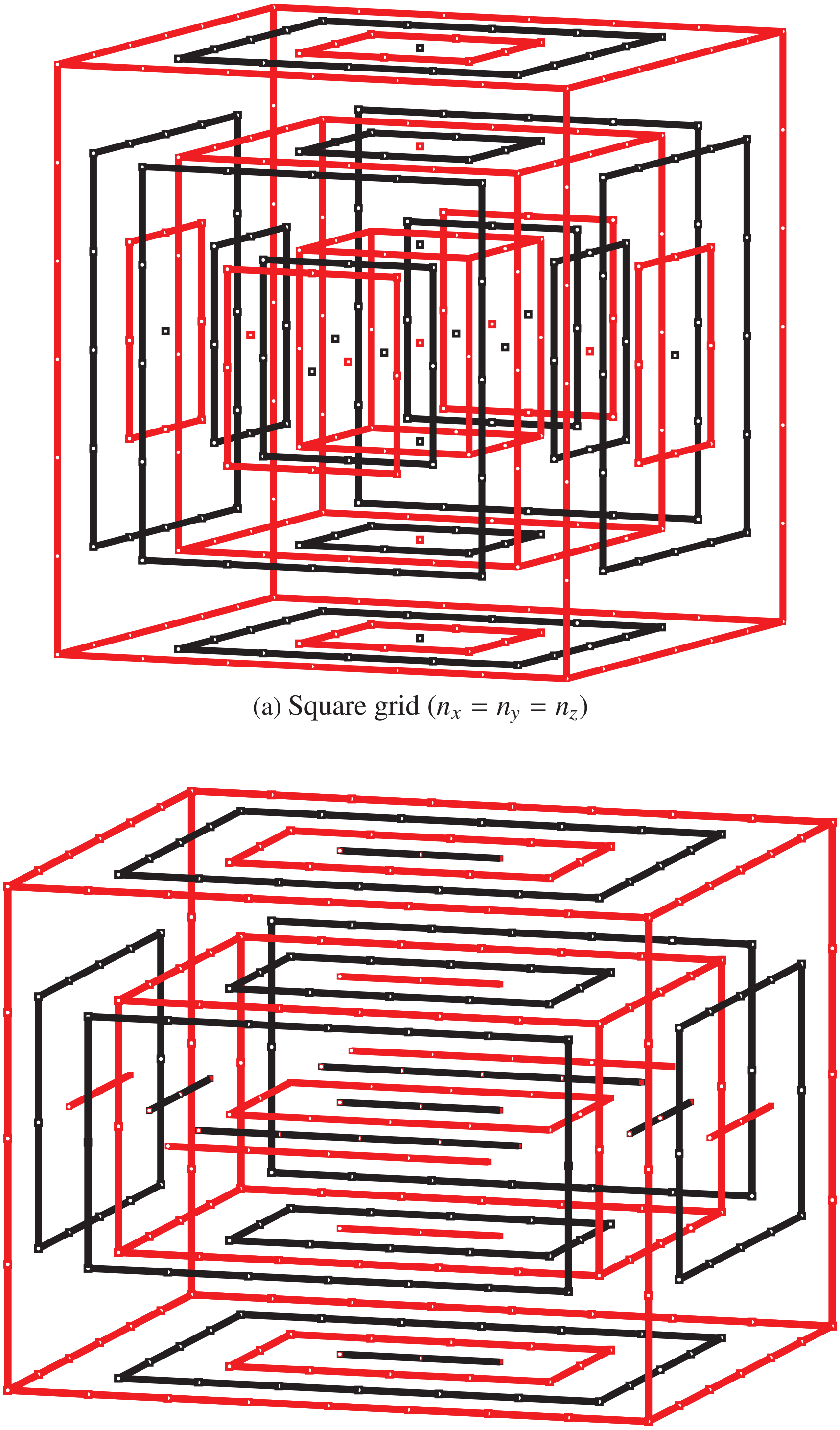


(a) Square grid ($n_x = n_y = n_z$)



(b) Rectangular grid ($n_x > n_y > n_z$)

Figure 4.6: Relaxation motif for 3D tweed smoothing.



(a) Square grid ($n_x = n_y = n_z$)

(b) Rectangular grid ($n_x > n_y > n_z$)

Figure 4.7: Relaxation motif for 3D box smoothing.

The outline of this chapter is as follows. Section 4.2 introduces the technical details regarding the implementation of tweed and box relaxation for the solution of 2D and 3D elliptic PDEs. Section 4.3 shows how to implement tweed and box relaxation in the smoothing step of the multigrid algorithm, and derives convergence factors for different combinations of smoothers, restriction schemes, and number of smoothing steps. Section 4.4 presents convergence results of multigrid with tweed and box relaxation applied to the solution of (4.2), and the results are compared with other available smoothing approaches. Conclusions and future work are discussed in Section 7.5.

4.2 Tweed and box relaxation

Tweed relaxation iteratively solves the linear system of equations arising from the second-order central discretization of second-order 2D or 3D elliptic PDEs. A prototypical example is given in (4.2); for convenience, we take n_x and n_y (and, in the 3D case, n_z) as even. As depicted in Figure 4.4, starting from each corner (labelled as "red"), points are labeled in blocks of alternating colors by forming horizontal and vertical lines of points drawn perpendicular to the domain boundaries, and extended until such lines connect inside the domain. Due to the loose visual analogy between such gridpoint arrangements and certain cloth textures, this smoothing scheme has been dubbed "tweed" relaxation. In the 2D case with $n_x = n_y$, as illustrated in Figure 4.4a, four legs of the same color converge at the center of the domain. In the 2D case with $n_x \neq n_y$, as illustrated in Figure 4.4b, two three-legged blocks arise, and the remaining gridpoints in the central part of the domain are connected by lines of alternating color perpendicular to the closest boundaries.

Following an approach analogous to that used in zebra relaxation, in tweed relaxation, (4.2) is first solved exactly at each red point while holding the values of the unknowns

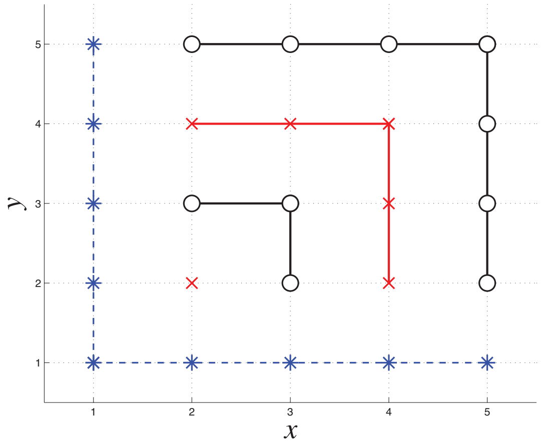


Figure 4.8: Tweed motif near a corner of a 2D domain. Blue dashed lines indicate the domain boundaries where the value of the unknown is specified.

at the black points fixed, then (4.2) is solved exactly at each black point while holding the values of the unknowns at the red points fixed. In the case of zebra relaxation, such an approach leads, for each (linear) block, to a single tridiagonal system of equations that may be solved efficiently using Thomas algorithm. In the case of tweed relaxation, this approach leads, for each block other than the corners[1], to m tridiagonal systems that are interconnected at a common branch point, where m = 2, 3, or 4; an efficient technique to solve this subproblem[2], which we refer to as the m-legged Thomas algorithm, is discussed in [37] and chapter 5. In short, the m-legged Thomas algorithm performs a forward sweep from the tips of each leg in towards the branch point, then performs a solve relating the branch point to the points nearest to the branch point along each leg, then performs a back substitution going back out to the tips of each leg.

To illustrate, consider the iterative solution of (4.2) over a uniform grid; the block of red points that connect at the (4, 4) gridpoint in Figure 4.8 in this case are governed by

[1] At the corners, simple pointwise relaxation is performed.

[2] Note that the case with m = 2 can, of course, be solved directly with the Thomas algorithm itself.

the following equations:

$$\begin{bmatrix} 1 & & & \\ 1/\Delta x^2 & -2/\Delta x^2 - 2/\Delta y^2 & 1/\Delta x^2 & \\ & 1/\Delta x^2 & -2/\Delta x^2 - 2/\Delta y^2 & 1/\Delta x^2 \end{bmatrix} \begin{pmatrix} u_{1,4} \\ u_{2,4} \\ u_{3,4} \end{pmatrix} = \begin{pmatrix} g_{1,4} \\ f_{2,4} - (u_{2,3} + u_{2,5})/\Delta y^2 \\ f_{3,4} - (u_{3,3} + u_{3,5})/\Delta y^2 \end{pmatrix},$$

$$\begin{bmatrix} 1 & & & \\ 1/\Delta y^2 & -2/\Delta x^2 - 2/\Delta y^2 & 1/\Delta y^2 & \\ & 1/\Delta y^2 & -2/\Delta x^2 - 2/\Delta y^2 & 1/\Delta y^2 \end{bmatrix} \begin{pmatrix} u_{4,1} \\ u_{4,2} \\ u_{4,3} \end{pmatrix} = \begin{pmatrix} g_{4,1} \\ f_{4,2} - (u_{3,2} + u_{5,2})/\Delta x^2 \\ f_{4,3} - (u_{3,3} + u_{5,3})/\Delta x^2 \end{pmatrix},$$

$$\begin{bmatrix} 1/\Delta x^2 & 1/\Delta y^2 & -2/\Delta x^2 - 2/\Delta y^2 \end{bmatrix} \begin{pmatrix} u_{3,4} \\ u_{4,3} \\ u_{4,4} \end{pmatrix} = f_{4,4} - u_{5,4}/\Delta x^2 + u_{4,5}/\Delta y^2.$$

More generally, each tweed relaxation involving m legs of length $p + 1$ (including the branch point) can be formulated as the solution of a system of linear equations of the following form:

$$\begin{bmatrix} b_1^{(i)} & c_1^{(i)} & & & & \\ a_2^{(i)} & b_2^{(i)} & c_2^{(i)} & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_{p-1}^{(i)} & b_{p-1}^{(i)} & c_{p-1}^{(i)} & \\ & & & a_p^{(i)} & b_p^{(i)} & c_p^{(i)} \end{bmatrix} \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_{p-1}^{(i)} \\ x_p^{(i)} \\ x_{\text{center}} \end{pmatrix} = \begin{pmatrix} r_1^{(i)} \\ r_2^{(i)} \\ \vdots \\ r_{p-1}^{(i)} \\ r_p^{(i)} \end{pmatrix} \quad i = 1,\ 2,\ \ldots, m, \tag{4.3}$$

$$\begin{bmatrix} d_1 & d_2 & \ldots & d_m & d_{m+1} \end{bmatrix} \begin{pmatrix} x_p^{(1)} \\ x_p^{(2)} \\ \vdots \\ x_p^{(m)} \\ x_{\text{center}} \end{pmatrix} = r_{\text{center}}; \tag{4.4}$$

note that, on a stretched grid, the computations of the RHS terms r_j^i along the legs typically require 4 flops. This system of $mp + 1$ equations in $mp + 1$ unknowns can be solved efficiently by applying an *m-legged* extension of the Thomas algorithm, as described in detail in [37]. A system of equations like the one in (4.4) requires $8mp + m + 1$ operations.

Tweed relaxation applied to the iterative solution of (4.2) over an $n \times n$ square grid requires, at each iteration, 4 pointwise relaxations (one for each corner), $2(n-3)$ two-legged Thomas solves with legs of increasing size starting from the corners of the domain towards the center, and one four-legged Thomas relaxation for the center cross. Hence, applying a full round of tweed relaxation requires:

- Corners: $4 \times 9 = 36$ *(Pointwise relaxation)*
- Corner legs: $4 \times \sum_{i=1}^{(n-3)/2} \left[(8\,i + 4)\,\textit{(RHS computation)} + \text{mLT}\textsc{homas}(p = i,\ m = 2)\right] = 12\,n^2 - 34\,n - 6$
- Center cross: $4 \times 2(n-1)\,\textit{(RHS computation)} + 1 \times \text{mLT}\textsc{homas}(p = (n-1)/2,\ m = 4) = 24\,n - 19$

Overall, $12\,n^2 - 10\,n + 11$ flops are needed (to leading order, taking $N = n^2$, $\sim 8N$ for the forward sweeps and back substitutions, as in the regular Thomas algorithm, and $\sim 4N$ for the RHS computations).

The extension of tweed relaxation to 3D is straightforward, as illustrated in Figure 4.6. As with the 2D scheme, the eight corner points are first relaxed using a pointwise smoother. Then, starting from the corners, points are relaxed in blocks of alternate colors, each composed of $m = 2$ or 3 legs. If $n_x > n_y > n_z$, as illustrated in Figure 4.6b, or $n_x = n_y > n_z$ (not pictured), four $m = 4$ blocks arise, with the $y - z$ and $x - z$ planes in the center of the 3D grid covered with the 2D motif illustrated in Figure 4.4b. If $n_x > n_y = n_z$ (not pictured), two blocks with $m = 5$ arise, with the $y-z$ planes in the center of the 3D grid covered with the 2D motif illustrated in Figure 4.4a. If $n_x = n_y = n_z$, as illustrated in Figure 4.6a, a single $m = 6$ block arises in the center. In all cases, it is seen that each gridpoint is a member of exactly one relaxation block. Further, for large grids, the number of points on the legs dominates the number of branch points. Thus, to leading order, the computation cost of 3D tweed for large grids is the same as that of a *single* set of sweeps (that is, in a single direction) of 1D zebra relaxations. However, on a grid that is stretched in three directions, the 3D alternating-direction zebra scheme requires *three* successive sweeps of 1D zebra relaxations, one in each direction. Hence, the leading-order computational cost of 3D tweed relaxation is one third that of the three sweeps of the 3D alternating-direction zebra scheme for such problems.

The 2D box relaxation strategy starts by performing a block relaxation on all points adjacent to the boundaries, and proceeds towards the center while alternating the color of the blocks. This creates a pattern of concentric box-shaped blocks of alternate color, as depicted in Figure 4.5. The relaxation scheme terminates with one pointwise relaxation if $n_x = n_y$, as shown in Figure 4.5a, or with one line relaxation in the x direction if $n_x > n_y$, as shown in Figure 4.5b. To illustrate, consider the iterative solution of (4.2) over a uniform grid, with a red box with corners (2, 2), (2, 4), (4, 2), (4, 4), as depicted in Figure 4.9. The tridiagonal circulant system of equations associated with such a relaxation is:

$$\begin{bmatrix} -2/\Delta x^2 - 2/\Delta y^2 & 1/\Delta x^2 & & & & & & 1/\Delta y^2 \\ 1/\Delta x^2 & -2/\Delta x^2 - 2/\Delta y^2 & 1/\Delta x^2 & & & & & \\ & 1/\Delta x^2 & -2/\Delta x^2 - 2/\Delta y^2 & 1/\Delta y^2 & & & & \\ & & 1/\Delta y^2 & -2/\Delta x^2 - 2/\Delta y^2 & 1/\Delta y^2 & & & \\ & & & 1/\Delta y^2 & -2/\Delta x^2 - 2/\Delta y^2 & 1/\Delta x^2 & & \\ & & & & 1/\Delta x^2 & -2/\Delta x^2 - 2/\Delta y^2 & 1/\Delta x^2 & \\ & & & & & 1/\Delta x^2 & -2/\Delta x^2 - 2/\Delta y^2 & 1/\Delta y^2 \\ 1/\Delta y^2 & & & & & & 1/\Delta y^2 & -2/\Delta x^2 - 2/\Delta y^2 \end{bmatrix} \begin{pmatrix} u_{2,2} \\ u_{3,2} \\ u_{4,2} \\ u_{4,3} \\ u_{4,4} \\ u_{3,4} \\ u_{2,4} \\ u_{2,3} \end{pmatrix} = \begin{pmatrix} f_{2,2} - u_{1,2}/\Delta x^2 - u_{2,1}/\Delta y^2 \\ f_{3,2} - u_{3,1}/\Delta y^2 - u_{3,3}/\Delta y^2 \\ f_{4,2} - u_{4,1}/\Delta y^2 - u_{5,2}/\Delta x^2 \\ f_{4,3} - u_{3,3}/\Delta x^2 - u_{5,3}/\Delta x^2 \\ f_{4,4} - u_{5,4}/\Delta x^2 - u_{4,5}/\Delta y^2 \\ f_{3,4} - u_{3,5}/\Delta y^2 - u_{3,3}/\Delta y^2 \\ f_{2,4} - u_{2,5}/\Delta y^2 - u_{1,4}/\Delta x^2 \\ f_{2,3} - u_{1,3}/\Delta x^2 - u_{3,3}/\Delta x^2 \end{pmatrix} \tag{4.5}$$

More generally, each box relaxation involving a block of n points connected together re-

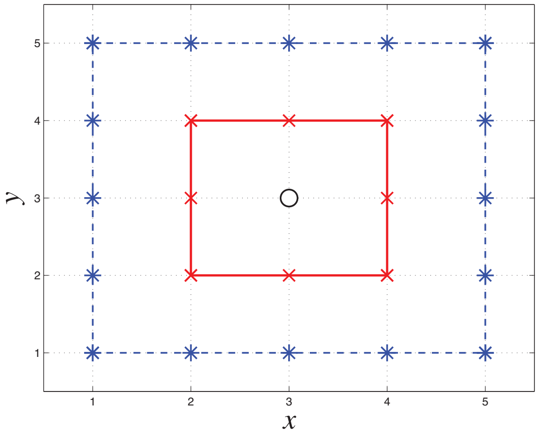


Figure 4.9: Box relaxation scheme for a 5×5 uniform grid on a Cartesian domain. Blue dashed lines indicate the domain boundaries, red lines connect the gridpoints involved in the same block relaxation.

quires the solution of the following linear system:

$$\begin{bmatrix} b_1 & c_1 & & & a_1 \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{m-1} & b_{m-1} & c_{m-1} \\ c_m & & & a_m & b_m \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \\ x_m \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{m-1} \\ r_m \end{pmatrix} \tag{4.6}$$

The circulant tridiagonal system in (4.6) can be solved using a periodic (a.k.a. circulant) Thomas solver such as that introduced in [38]. A minimal storage implementation of such algorithm is presented in [37]. For an $m \times m$ system like that in (4.6), the circulant Thomas solver requires $14m - 16$ flops, which is about $14/8 = 1.75$ times the computational cost of the Thomas algorithm for tridiagonal systems.

Box relaxation applied to the iterative solution of (4.2) over an $n \times n$ square grid requires, at each iteration, 1 pointwise relaxation at the center point and $(n-1)/2$ concentric box relaxations. Hence, applying a full round of box relaxation requires:

- Center point: 9 *(Pointwise relaxation)*
- Elsewhere: $\sum_{i=1}^{(n-1)/2} \left[32\, i\, \textit{(RHS computation)} + \textsc{CircThomas}(m = 8\, i)\right] = 18\, n^2 - 8\, n - 10$

Overall, $18\, n^2 - 8\, n - 1$ flops are needed (to leading order, taking $N = n^2$, $\sim 14N$ for the circulant Thomas solves, and $\sim 4N$ for the RHS computations), which makes box relaxation about 50% more expensive than tweed relaxation.

The extension of box relaxation to 3D is illustrated in Figure 4.7. Starting from the faces of the 3D domain, gridpoints are relaxed in blocks covering the faces of concentric 3D rectangular cuboids filling the domain. Along all the edges of each such rectangular cuboid, a 3D box extension of the Thomas algorithm used, as discussed in [37]. Within each of the six faces of each such rectangular cuboid, a sequence of concentric 2D box relaxations is performed, using the circulant Thomas algorithm, as illustrated in Figure 4.5. If $n_x = n_y = n_z$, as illustrated in Figure 4.7a, this sequence of relaxations includes a point relaxation at the center of each face of each box. If $n_x > n_y > n_z$, as illustrated in Figure 4.7b, this sequence of relaxations includes a line relaxation at the center of each face of each box.

Another approach worth mentioning is the sequential application of tweed and box relaxation, akin to alternating-direction zebra relaxation. In the 2D case, this alternating tweed/box relaxation approach requires $\sim 30\, n^2$ flops, which makes it 20% more expensive than alternating-direction zebra relaxation. This approach might prove useful whenever grid clustering does not happen only in localized regions either near the center or close to the boundaries, but stretching is present in several different areas of the domain.

Considering other 2D relaxation schemes, $\sim 9\, n^2$ flops are required for checkerboard relaxation of (4.2) on an $n \times n$ grid, as 9 flops are required at each gridpoint, and $\sim 12\, n^2$ flops are required for one-direction zebra relaxation of (4.2) on an $n \times n$ grid, as the

Thomas algorithm requires $\sim 8\,n$ flops and the computation of the RHS requires $4\,n$ flops on each of n lines; it follows that $\sim 24\,n^2$ flops are required for alternating-direction zebra relaxation. In other words, tweed relaxation has a computational cost which is comparable to one-direction zebra relaxation, and is about half the computational cost of alternating-direction zebra relaxation. The computational cost of box relaxation is about 25% less than alternating-direction zebra relaxation.

In the following two sections, a multigrid algorithm is applied to (4.2) leveraging the tweed and box relaxation schemes discussed above, and performance is compared (both analytically and numerically) with multigrid leveraging the traditional checkerboard and zebra relaxation schemes.

4.3 Multigrid algorithm and convergence analysis

Consider the problem of solving (4.2), of the form $\mathcal{L}u = f$, on a stretched $(n_x + 1) \times (n_y + 1)$ grid Ω_p (including boundary points) via a multigrid algorithm, with $n_x = 2^p\,a$ and $n_y = 2^p\,b$ where a and b are small positive integers, at least one of which is odd. The multigrid algorithm leverages a sequence of grids $\Omega_p, \Omega_{p-1}, \dots, \Omega_0$, where $\Omega_{\ell-1}$ is obtained by coarsening Ω_ℓ by a factor of two in each direction (that is, removing every other interior grid line in each direction), and thus Ω_0 is $(a{+}1){\times}(b{+}1)$. We indicate with $\mathcal{L}_\ell$ the discretized Laplacian on Ω_ℓ, as defined in (4.2), and will iterate on a sequence of discretized Poisson problems of the form $\mathcal{L}_\ell u^\ell = f^\ell$ on Ω_ℓ for $\ell \in [0, p]$. A skeleton V-cycle linear multigrid algorithm is composed of the following steps:

(1) Initialize $u^p = 0$ and $\ell = p$.

(2) Apply ν_1 *pre-smoothing* relaxations to the problem $\mathcal{L}_\ell u^\ell = f^\ell$ on the grid Ω_ℓ.

(3) Compute the remaining *defect* $d^\ell = f^\ell - \mathcal{L}_\ell u^\ell$ of the solution u^ℓ, and *restrict* this defect d^ℓ from Ω_ℓ to the next coarser grid $\Omega_{\ell-1}$, calling the result $f^{\ell-1}$. Set $\ell \leftarrow \ell - 1$.

(4) If $\ell > 0$, initialize $u^\ell = 0$ and repeat from (2); otherwise, solve $\mathcal{L}_0 u^0 = f^0$ for the correction u^0 directly.

(5) *Prolongate* the correction u^ℓ on Ω_ℓ to the next finer grid $\Omega_{\ell+1}$, calling the result $v^{\ell+1}$, and *update* the solution $u^{\ell+1}$ defined on $\Omega_{\ell+1}$ with the correction $v^{\ell+1}$; that is, take $u^{\ell+1} \leftarrow u^{\ell+1} + v^{\ell+1}$. Set $\ell \leftarrow \ell + 1$.

(6) Apply ν_2 *post-smoothing* relaxations to the problem $\mathcal{L}_\ell u^\ell = f^\ell$ on the grid Ω_ℓ. If $\ell < p$, repeat from (5); otherwise, repeat from (2) until the norm of the defect on Ω_p, $||f^p - \mathcal{L}_p u^p||$, is sufficiently small.

The reason that the multigrid algorithm works so well is that most effective relaxation schemes, such as checkerboard relaxation, *smooth* the error in the solution very quickly (that is, they significantly refine the solution vector u^ℓ on the smallest scales representable on the grid Ω_ℓ being used), but are inefficient at reducing the defect on the larger length scales representable on the grid; thus, following the multigrid approach, the larger length scales of the defect are addressed by applying smoothing to successively coarser representations of the problem at hand, as described above. There are various ways to accelerate the multigrid algorithm. First, to reduce storage, the computation of the defect and its restriction to the coarser grid, in step (3), can be combined into a single step, thereby eliminating the need for the intermediate storage of d^ℓ. Similarly, the computation of the prolongation of the correction, $v^{\ell+1}$, and its use in updating the solution $u^{\ell+1}$, in step (5), can also be combined into a single step, thereby eliminating the need for storage of $v^{\ell+1}$. Other cycling strategies, performing more iterations at the coarser length scales, are sometimes used.

Six different relaxation schemes are considered below for the smoothing applied at steps (2) and (6), namely: checkerboard, one-direction zebra, alternating-direction zebra, tweed, box, and alternating tweed/box. For the restriction step in (3), half weighting and full weighting are considered: denoting with d^ℓ the defect on the grid Ω_ℓ, and with $f^{\ell-1}$ this defect restricted onto the next coarser grid $\Omega_{\ell-1}$, the half-weighting restriction operation is

$$f^{\ell-1}_{i,j} = \frac{1}{2}d^\ell_{2i,2j} + \frac{1}{8}(d^\ell_{2i-1,2j} + d^\ell_{2i,2j-1} + d^\ell_{2i+1,2j} + d^\ell_{2i,2j+1}),$$

whereas the full-weighting restriction operation is

$$\begin{aligned} f^{\ell-1}_{i,j} = \frac{1}{4}d^\ell_{2i,2j} &+ \frac{1}{8}(d^\ell_{2i-1,2j} + d^\ell_{2i,2j-1} + d^\ell_{2i+1,2j} + d^\ell_{2i,2j+1}) + \\ &+ \frac{1}{16}(d^\ell_{2i-1,2j-1} + d^\ell_{2i+1,2j-1} + d^\ell_{2i-1,2j+1} + d^\ell_{2i+1,2j+1}). \end{aligned}$$

For the prolongation step in (5), bilinear interpolation is used, which is the dual of the full-weighting restriction operation: denoting with u^ℓ the correction on the grid Ω_ℓ, and with $v^{\ell+1}$ this correction prolongated onto the next finer grid $\Omega_{\ell+1}$, the bilinear interpolation operation is

$$v^{\ell+1}_{i,j} = \begin{cases} u^\ell_{i/2,\,j/2} & i = \text{even}, \quad j = \text{even} \\ \frac{1}{2}(u^\ell_{i/2,\,(j-1)/2} + u^\ell_{i/2,\,(j+1)/2}) & i = \text{even}, \quad j = \text{odd} \\ \frac{1}{2}(u^\ell_{(i-1)/2,\,j/2} + u^\ell_{(i+1)/2,\,j/2}) & i = \text{odd}, \quad j = \text{even} \\ \frac{1}{4}(u^\ell_{(i-1)/2,\,(j-1)/2} + u^\ell_{(i+1)/2,\,(j-1)/2} + u^\ell_{(i-1)/2,\,(j+1)/2} + u^\ell_{(i+1)/2,\,(j+1)/2}) & i = \text{odd}, \quad j = \text{odd} \end{cases}$$

Calculation of two-grid convergence factors of the associated multigrid operator provides a useful indication of the effectiveness of the combined application of different restriction and prolongation schemes, smoothers, and the number of pre-smoothing and post-smoothing relaxations applied. To proceed, consider an $(n_x + 1) \times (n_y + 1)$ grid Ω_ℓ and a coarsened $(n_x/2 + 1) \times (n_y/2 + 1)$ grid $\Omega_{\ell-1}$. Four different cases of grid stretching are also considered: a uniform grid with $\Delta x = \Delta y$, and three stretched grids, two exhibiting differing amounts of near-wall clustering, and one exhibiting near-center clustering. Near-wall clustering of Ω_ℓ over the domain $[0, L_x] \times [0, L_y]$ is achieved with the hyperbolic tangent stretching function

$$\begin{aligned} x_i &= (L_x/2)\{1 + \tanh[c\,(2\,i/n_x - 1)]/\tanh c\}, \quad i = 0, \ldots, n_x, \\ y_j &= (L_y/2)\{1 + \tanh[c\,(2\,j/n_y - 1)]/\tanh c\}, \quad j = 0, \ldots, n_y, \end{aligned} \tag{4.7}$$

where c is a tuning parameter that determines the amount of stretching: $c = 1.5$ creates mild stretching (see Figure 4.2a), and $c = 3.0$ creates more significant stretching (see Figure 4.2b). Near-center clustering is achieved with a simple shifted version of the stretching function used in (4.7) such that

$$\begin{aligned} x_i &= \begin{cases} (L_x/2)\tanh[2\,c\,i/n_x]/\tanh c, & i = 0, \ldots, n_x/2, \\ (L_x/2)\{2 - \tanh[c\,(2 - 2\,i/n_x)]/\tanh c\}, & i = n_x/2, \ldots, n_x; \end{cases} \\ y_j &= \begin{cases} (L_y/2)\tanh[2\,c\,j/n_y]/\tanh c, & j = 0, \ldots, n_y/2, \\ (L_y/2)\{2 - \tanh[c\,(2 - 2\,j/n_y)]/\tanh c\}, & j = n_y/2, \ldots, n_y. \end{cases} \end{aligned} \tag{4.8}$$

An example of a stretched grid generated using (4.8), with $c = 1.5$, is shown in Figure 4.2c.

We now denote the restriction operator from the fine to the coarse grid as $I_\ell^{\ell-1}$, and the prolongation operator from the coarse grid to the fine grid as $I_{\ell-1}^{\ell}$. Pre- and post-smoothing operators on Ω_ℓ are indicated as $S_\ell^{\nu_1}$ and $S_\ell^{\nu_2}$, where ν_1 and ν_2 indicate the

Table 4.1: Spectral radius of the two-grid multigrid operator $M_\ell^{\ell-1}$ for the homogeneous problem (i.e., a uniform grid) with respect to the number of smoothing steps ν for different smoothers and restriction operators.

Half-weighting restriction						
Smoother	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$	$\nu = 6$
Checkerboard	0.4986	0.1238	0.0340	0.0240	0.0189	0.0155
One-direction zebra	0.5735	0.5014	0.4935	0.4905	0.4881	0.4858
Alternating-direction zebra	0.5836	0.5912	0.5868	0.5823	0.5778	0.5734
Tweed	0.5661	0.5032	0.4951	0.4919	0.4893	0.4869
Box	0.5494	0.5007	0.4929	0.4899	0.4874	0.4850
Alternating tweed/box	0.5833	0.5909	0.5865	0.5820	0.5775	0.5731

Full-weighting restriction						
Smoother	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$	$\nu = 6$
Checkerboard	0.2494	0.0739	0.0526	0.0408	0.0333	0.0283
One-direction zebra	0.2494	0.0622	0.0167	0.0116	0.0092	0.0077
Alternating-direction zebra	0.0839	0.0391	0.0265	0.0201	0.0162	0.0135
Tweed	0.2488	0.0621	0.0163	0.0109	0.0084	0.0069
Box	0.2465	0.0612	0.0161	0.0108	0.0084	0.0069
Alternating tweed/box	0.0830	0.0382	0.0257	0.0193	0.0154	0.0128

number of times the smoother is applied at each step. Following [35], the complete two-grid multigrid operator, denoted $M_\ell^{\ell-1}$, is given by

$$M_\ell^{\ell-1} = S_\ell^{\nu_2}\,(I_\ell - I_{\ell-1}^{\ell}\mathcal{L}_{\ell-1}^{-1}I_\ell^{\ell-1}\mathcal{L}_\ell)\,S_\ell^{\nu_1}, \tag{4.9}$$

where I_ℓ is the identity matrix on the fine grid Ω_ℓ, and $\mathcal{L}_\ell$ and $\mathcal{L}_{\ell-1}$ denote the discrete Laplace operators on the fine and coarse grids, respectively.

Tables 4.1 through 4.4 show the computation of the spectral radius of the two-grid multigrid operator $M_\ell^{\ell-1}$ applied to the solution of (4.2) over a grid with $n_x = n_y = 128$ and $L_x = L_y = 1$ with half-weighting and full-weighting used for the restriction operation. Since only the sum of pre- and post-smoothing steps affects the convergence of the two-grid cycle, the spectral radius is reported as a function of the sum $\nu = \nu_1 + \nu_2$.

Table 4.2: Spectral radius of the two-grid multigrid operator $M_\ell^{\ell-1}$ for the inhomogeneous problem with near-wall clustering (4.7), taking $c = 1.5$, with respect to ν for different smoothers and restriction operators.

Half-weighting restriction						
Smoother	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$	$\nu = 6$
Checkerboard	0.7900	0.6244	0.4936	0.3904	0.3090	0.2446
One-direction zebra	0.7921	0.6302	0.5113	0.4825	0.4774	0.4730
Alternating-direction zebra	0.6879	0.6385	0.6096	0.5903	0.5755	0.5629
Tweed	0.5478	0.4969	0.4867	0.4803	0.4744	0.4687
Box	0.7921	0.6302	0.5113	0.4840	0.4793	0.4754
Alternating tweed/box	0.6891	0.6490	0.6285	0.6138	0.6013	0.5899

Full-weighting restriction						
Smoother	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$	$\nu = 6$
Checkerboard	0.7855	0.6179	0.4867	0.3840	0.3035	0.2403
One-direction zebra	0.7863	0.6190	0.4879	0.3851	0.3043	0.2409
Alternating-direction zebra	0.0816	0.0344	0.0218	0.0162	0.0129	0.0107
Tweed	0.2108	0.0538	0.0257	0.0184	0.0143	0.0117
Box	0.7863	0.6190	0.4879	0.3851	0.3043	0.2409
Alternating tweed/box	0.0696	0.0286	0.0179	0.0126	0.0095	0.0073

Table 4.3: Spectral radius of the two-grid multigrid operator $M_\ell^{\ell-1}$ for the inhomogeneous problem with near-wall clustering (4.7), taking $c = 3.0$, with respect to ν for different smoothers and restriction operators.

Half-weighting restriction						
Smoother	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$	$\nu = 6$
Checkerboard	0.9541	0.9103	0.8686	0.8287	0.7907	0.7545
One-direction zebra	0.9541	0.9104	0.8687	0.8290	0.7911	0.7550
Alternating-direction zebra	0.6356	0.5765	0.5395	0.5092	0.4821	0.4572
Tweed	0.5462	0.4799	0.4566	0.4386	0.4220	0.4062
Box	0.9541	0.9104	0.8687	0.8290	0.7911	0.7550
Alternating tweed/box	0.6538	0.6087	0.5743	0.5440	0.5161	0.4900

Full-weighting restriction						
Smoother	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$	$\nu = 6$
Checkerboard	0.9534	0.9090	0.8666	0.8263	0.7879	0.7512
One-direction zebra	0.9534	0.9091	0.8668	0.8265	0.7880	0.7514
Alternating-direction zebra	0.1002	0.0372	0.0219	0.0148	0.0110	0.0088
Tweed	0.1866	0.0537	0.0282	0.0204	0.0158	0.0135
Box	0.9534	0.9091	0.8668	0.8265	0.7880	0.7514
Alternating tweed/box	0.0987	0.0362	0.0223	0.0161	0.0125	0.0103

Table 4.4: Spectral radius of the two-grid multigrid operator $M_{\ell}^{\ell-1}$ for the inhomogeneous problem with near-center clustering (4.8), taking $c = 1.5$, with respect to ν for different smoothers and restriction operators.

Half-weighting restriction						
Smoother	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$	$\nu = 6$
Checkerboard	0.8865	0.7859	0.6968	0.6178	0.5478	0.4859
One-direction zebra	0.8876	0.7892	0.7047	0.6356	0.5822	0.5431
Alternating-direction zebra	0.7918	0.7463	0.7149	0.6911	0.6724	0.6574
Tweed	0.8876	0.7892	0.7047	0.6356	0.5823	0.5434
Box	0.5496	0.4974	0.4913	0.4888	0.4867	0.4846
Alternating tweed/box	0.5312	0.5433	0.5389	0.5353	0.5320	0.5289

Full-weighting restriction						
Smoother	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$	$\nu = 6$
Checkerboard	0.8826	0.7793	0.6884	0.6084	0.5380	0.4759
One-direction zebra	0.8829	0.7798	0.6891	0.6092	0.5388	0.4767
Alternating-direction zebra	0.0805	0.0375	0.0249	0.0185	0.0146	0.0119
Tweed	0.8829	0.7798	0.6891	0.6092	0.5388	0.4767
Box	0.1887	0.0408	0.0257	0.0195	0.0160	0.0136
Alternating tweed/box	0.0805	0.0363	0.0248	0.0188	0.0150	0.0125

We observe that appreciably reduced spectral radius (and, thus, a significantly improved convergence rate) is obtained in the homogeneous case (Table 4.1) for any choice of smoother with full-weighting restriction, and convergence improves with the number of smoothing steps ν. However, all block relaxations, such as zebra, tweed, box, and alternating tweed/box, show poor convergence when half-weighting restriction is used. Full-weighting restriction is also required for rapid convergence in all of the stretched grid cases considered in Tables 4.2 through 4.4, as discussed below.

For the cases with near-wall clustering (Tables 4.2 and 4.3), checkerboard and one-direction zebra show a significant degradation in convergence for both choices of the restriction operator. Convergence is greatly improved when alternating-direction zebra, tweed, or alternating tweed/box are used for the smoothing and full-weighting restriction is implemented. In particular, the convergence rate of alternating-direction zebra is slightly better than that of tweed for small ν, whereas the convergence rate is comparable for higher ν; note again, however, that the computational cost of 2D tweed is roughly half the computational cost of 2D alternating-direction zebra, thereby rendering tweed with full-weighting restriction the clearly superior choice. Though box relaxation alone is poorly suited for this (near-wall clustering) case, alternating tweed/box slightly outperforms both alternating-direction zebra and tweed for any choice of ν, albeit at substantially increased computational cost. Further, comparing Tables 4.2 and 4.3, it is seen that the convergence of tweed relaxation is affected only slightly by the degree of grid stretching applied.

For the case with near-center clustering (Table 4.4), again, checkerboard and one-direction zebra prove to be inadequate. In this case, convergence is greatly improved when alternating-direction zebra, box, or alternating tweed/box are used for the smoothing and full-weighting restriction is implemented. In particular, the convergence rate of alternating-direction zebra is slightly better than that of box for all values of ν; note again, however, that the computational cost of 2D box is about 25% less than the computational cost of 2D alternating-direction zebra, thereby rendering box with full-weighting restriction a competitive choice. Tweed relaxation alone is poorly suited for this (near-center clustering) case, and alternating tweed/box provides similar convergence as both alternating-direction zebra and tweed, albeit at substantially increased computational cost.

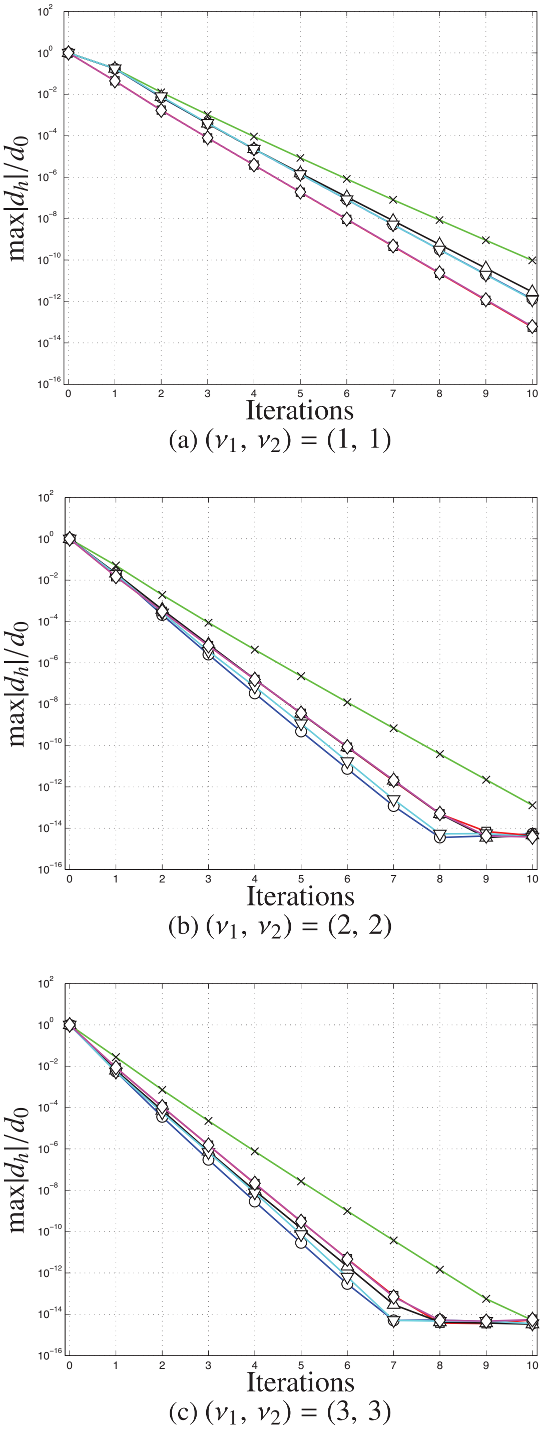
Further insight on checkerboard and zebra relaxation may be achieved by rigorous or local Fourier analysis (see [35]). However, the complicated arrangement of gridpoints in tweed and box relaxation prevents the extension of these analysis tools to the new smoothers proposed here.

4.4 Tests

To assess the performance of the tweed and box relaxation schemes, we applied the multigrid algorithm described in Section 4.3 to the solution of (4.2) over uniform and stretched grids with different smoothing schemes implemented. The RHS vector f^{ℓ} used in these tests is defined using uniformly-generated random numbers, and full-weighting restriction is used in every simulation reported. Also, we take $L_x = L_y = 1$, $n_x = n_y = 128$ (that is, $p = 7$), and $\nu_1 = \nu_2$ (that is, the same number of pre-smoothing and post-smoothing relaxations are used) in every simulation reported.

Convergence of the multigrid algorithm in the uniform-grid case is reported in Figure 4.10, where the maximum defect, normalized by the initial maximum defect d_0, is reported at each multigrid iteration. It can be observed that checkerboard smoothing provides rapid convergence, and that some gains are obtained by introducing block relaxation schemes, albeit at increased computational cost.

Convergence of the multigrid algorithm in the near-wall clustering case, with the grid generated using (4.7) for $c = 1.5$ and $c = 3.0$, is reported in Figure 4.11. Generally speaking, simulations show good agreement with the theoretical results presented in Section 4.3. In particular, we observe that the convergence of checkerboard, one-direction zebra, and box relaxation are significantly degraded, to the point of diverging in certain cases when large grid stretching is applied (see Figure 4.12b). The convergence rates obtained in the tests performed over a uniform grid (reported Figure 4.10) are retrieved in the near-wall clustering case when alternating-direction zebra is applied, a result that is well-known in the multigrid literature [35]. Remarkably, tweed relaxation achieves similar performance as alternating-direction zebra with roughly half the computational cost. Alternating-direction

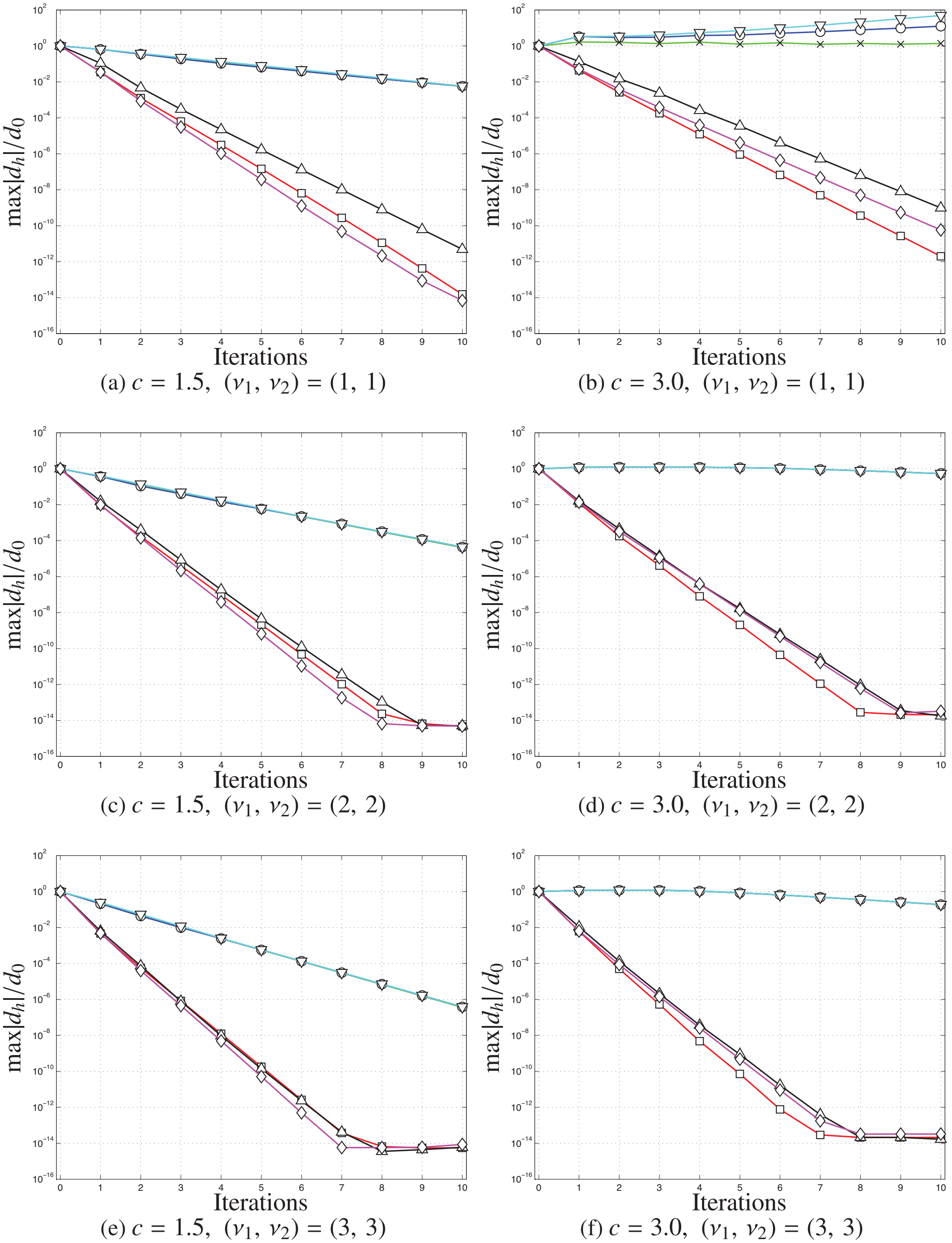


(a) $(\nu_1,\ \nu_2) = (1,\ 1)$

(b) $(\nu_1,\ \nu_2) = (2,\ 2)$

(c) $(\nu_1,\ \nu_2) = (3,\ 3)$

Figure 4.10: Multigrid convergence on (4.2) over a 129 × 129 uniform grid with different smoothers: checkerboard (green crosses), one-direction zebra (blue circles), alternating-direction zebra (red squares), tweed (black upward-pointing triangles), box (light blue downward-pointing triangles), alternating tweed/box (magenta diamonds).



(a) $c = 1.5,\ (\nu_1,\ \nu_2) = (1,\ 1)$

(b) $c = 3.0,\ (\nu_1,\ \nu_2) = (1,\ 1)$

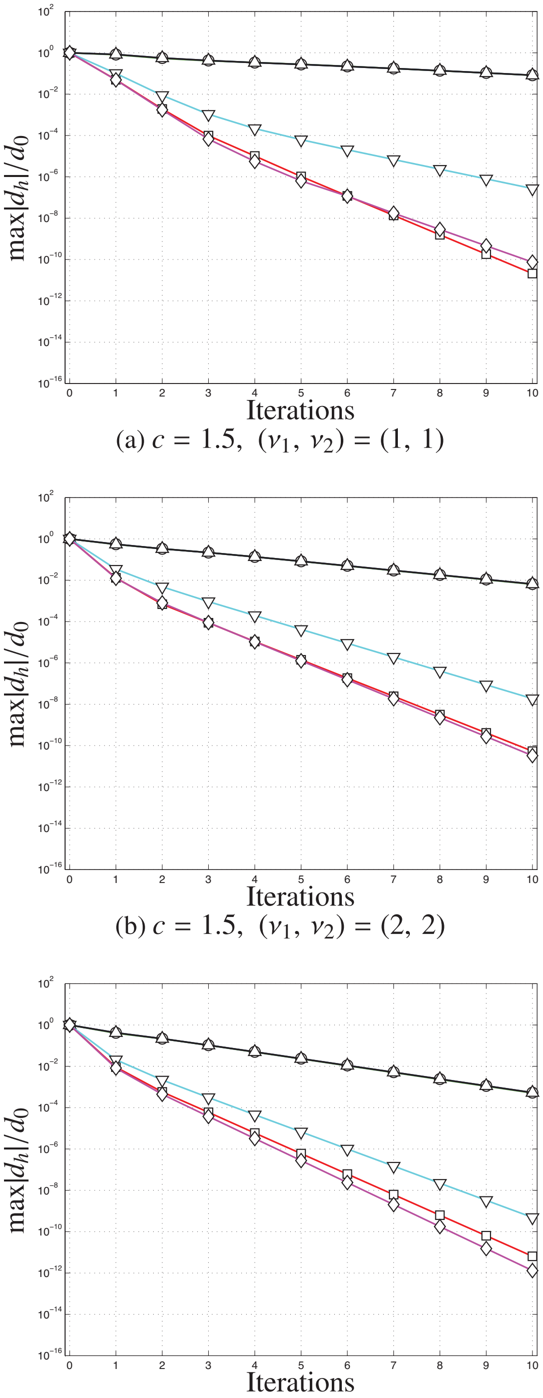
(c) $c = 1.5,\ (\nu_1,\ \nu_2) = (2,\ 2)$

(d) $c = 3.0,\ (\nu_1,\ \nu_2) = (2,\ 2)$

(e) $c = 1.5,\ (\nu_1,\ \nu_2) = (3,\ 3)$

(f) $c = 3.0,\ (\nu_1,\ \nu_2) = (3,\ 3)$

Figure 4.11: Multigrid convergence on (4.2) over a 129 × 129 stretched grid with near-wall clustering and different smoothers: checkerboard (green crosses), one-direction zebra (blue circles), alternating-direction zebra (red squares), tweed (black upward-pointing triangles), box (light blue downward-pointing triangles), alternating tweed/box (magenta diamonds).

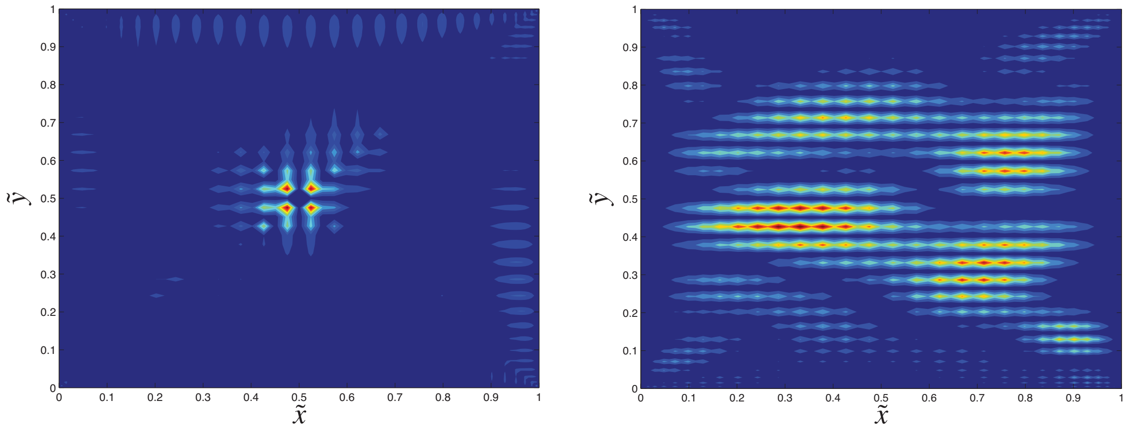


(a) $c = 1.5,\ (\nu_1,\ \nu_2) = (1,\ 1)$

(b) $c = 1.5,\ (\nu_1,\ \nu_2) = (2,\ 2)$

(c) $c = 1.5,\ (\nu_1,\ \nu_2) = (3,\ 3)$

Figure 4.12: Multigrid convergence on (4.2) over a 129 × 129 stretched grid with near-center clustering and different smoothers: checkerboard (green crosses), one-direction zebra (blue circles), alternating-direction zebra (red squares), tweed (black upward-pointing triangles), box (light blue downward-pointing triangles), alternating tweed/box (magenta diamonds).



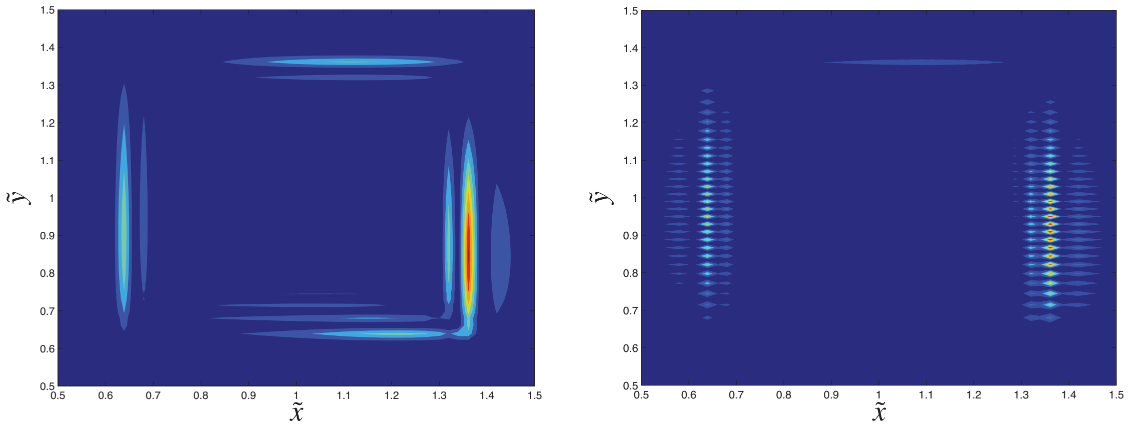
(a) Tweed

(b) Alternating-direction zebra

Figure 4.13: Defect after three cycles of multigrid over a 129×129 stretched grid defined through (4.7) with $(\nu_1, \nu_2) = (2, 2)$ and different smoothing schemes.

zebra slightly outperforms tweed for small $\nu_1 + \nu_2$, while the convergence of tweed is improved for larger $\nu_1 + \nu_2$. Alternating tweed/box gives even better convergence in certain cases, albeit at significantly increased computational cost. Interestingly, as shown in Figure 4.13 after three multigrid cycles, the distribution of the absolute value of the defect d^p on Ω^p is focused near the center of the domain (i.e., in the region where the grid is coarsest) in the case of tweed, and is distributed more uniformly throughout the domain in the case of alternating-direction zebra.

Convergence of the multigrid algorithm in the near-center clustering case, with the grid generated using (4.8) for $c = 1.5$, is reported in Figure 4.12. Again, simulations show close agreement with the theoretical results presented in Section 4.3. In particular, the convergence of checkerboard, one-direction zebra, and tweed relaxation are significantly degraded, whereas the convergence rates are improved when alternating-direction zebra is applied, albeit not recovering the convergence rate in the uniform grid case. Box relaxation achieves somewhat slower convergence rates as alternating-direction zebra for all cases reported, albeit with 25% reduced computational cost. Alternating tweed/box gives slightly



(a) Box

(b) Alternating-direction zebra

Figure 4.14: Defect after three cycles of multigrid over a 129 × 129 stretched grid defined through (4.8) with $(\nu_1, \nu_2) = (2, 2)$ and different smoothing schemes.

better convergence than alternating-direction zebra in certain cases, albeit at significantly increased computational cost. As shown in Figure 4.14 after three multigrid cycles, the distribution of the absolute value of the defect d^p on Ω^p is focused away from the center of the domain (where the grid is densest) in both the box and alternating-direction zebra cases. It is also noted that the performance of alternating-direction zebra, box, and alternating tweed/box degrade as c is increased, failing to achieve convergence in the near-center clustering case when extreme stretching is applied (that is, for $c \gtrsim 2.3$).

4.5 Conclusions

Two new relaxation schemes appropriate for the smoothing step in multigrid algorithms applied on 2D and 3D stretched grids have been introduced. The implementation of multigrid leveraging such smoothers facilitates the efficient solution of large linear (and, ultimately, nonlinear) systems arising from the discretization of elliptic PDEs on grids that are stretched in multiple spatial directions.

Tests on the 2D Poisson equation computed on a stretched grid with near-wall grid clustering conclusively demonstrate that multigrid leveraging tweed relaxation recovers the remarkable convergence speed of multigrid leveraging alternating-direction zebra relaxation, at roughly half of the computational cost in 2D (and, at roughly one third the computational cost in 3D). Further, the amount of stretching applied appears to have a rather minor effect on the convergence rate obtained.

Tests on the 2D Poisson equation computed on a stretched grid with near-center grid clustering demonstrate that multigrid leveraging box relaxation, though effective, unfortunately does not recover the convergence rate of that achieved using alternating-direction zebra relaxation in this case. Application of alternating tweed/box relaxation to this case is competitive with alternating-direction zebra relaxation, but comes at significantly increased computational cost.

Future work involves the implementation of 2D and 3D tweed relaxation for the solution of the Poisson equation for the pressure update in the numerical solution of turbulent duct and cavity flows with grid clustering near the walls.

Acknowledgements

This chapter contains work previously published in:

- D. Cavaglieri, A. Mashayek, T.R. Bewley, “Tweed and box relaxation: improved smoothing algorithms for multigrid solution of PDEs on stretched structured grids", *Submitted to Journal of Computational Physics*, 2016

Chapter 5

Extensions of the Thomas algorithm for the efficient direct solution of discretized PDEs on wireframe structures

5.1 Introduction

This chapter considers the solution of the sparse linear systems arising from the low-order finite-difference modeling of PDEs defined over 1D connected domains using 3-point-stencil operators. Noteworthy applications of these new schemes include heat diffusion over wireframe structures [39] and the deformation of loaded tensegrity structures [40]. Another application is given by the new relaxation schemes recently proposed by our group for the multigrid solution of elliptic PDEs discretized on structured 2D or 3D grids which are stretched in multiple directions [41].

As a prototype problem, consider the steady advection-diffusion-reaction equation given by

$$0 = D\frac{\partial^2 u}{\partial x^2} - C\frac{\partial u}{\partial x} + R(x) \tag{5.1}$$

with assigned boundary conditions, where D is the diffusivity, C is the advection speed, and R is a reaction term. Using finite differences defined on a 3-point-stencil, the discretized version of (5.1), defined along a simple 1D line segment using $n + 1$ points with equal spacing Δx, is

$$-\left(\frac{D}{\Delta x^2} + \frac{C}{2\Delta x}\right)u_{i-1} + \frac{D}{\Delta x^2}u_i - \left(\frac{D}{\Delta x^2} - \frac{C}{2\Delta x}\right)u_{i+1} = R_i, \qquad i = 2, \ldots, n, \tag{5.2}$$

with u_0 and u_{n+1} specified. This system is tridiagonal, and its solution may be obtained via the well-known Thomas[1] algorithm (see, for example, [43]), which provides the solution in $\sim 8n$ flops if solved from scratch, or in $\sim 5n$ flops if the LU decomposition of the LHS matrix is precomputed.

If one considers a *connection* of individual 1D segments, the tridiagonal structure of the problem is lost, and direct solution via the Thomas algorithm in its classical form is no longer possible. The simplest example of this is a periodic connection of a single 1D segment into a ring, which results in a matrix of circulant structure, and which is solvable via the circulant (a.k.a. periodic) Thomas algorithm (see, e.g., [38]) in $\sim 14n$ flops if

[1] As a historical side note, Llewellyn Hilleth Thomas (1903-1992) was a prodigious British physicist and mathematician. His early work in atomic physics includes “Thomas precession", a correction to the spin-orbit interaction of an elementary particle in relativistic kinematics which represented the foundation for spin polarization studies in the years to come [42]. Another key result was the development of the “Thomas-Fermi" model, a statistical representation of the atom which proved useful in describing average properties of heavy atoms in response to external stimuli, and is viewed nowadays as a precursor to modern density functional theory. In the 1930s, while working at Ohio State, he invented an isochronous cyclotron, which is considered a predecessor of modern high-field isochronous cyclotrons currently used in nuclear physics. During World War II, he worked on ballistics and magnetohydrodynamics at the U.S. Army’s Ballistic Research Lab. It was in 1946 that Thomas shifted focus and joined the Watson Scientific Computing Lab at Columbia. There, Thomas made major contributions to the fields of numerical methods and the design of electronic hardware for computers. It is in this period that he developed the Gaussian elimination algorithm for the efficient factorization of tridiagonal matrices which now bears his name.

solved from scratch, or $\sim 8n$ flops if the LU decomposition of the LHS matrix is precomputed. In more complicated structures, we may identify *nodal points* connected to three or more 1D segments. In such cases, careful reordering of the gridpoints can be implemented to minimize the matrix bandwidth, as performed by the Cuthill-McKee algorithm [44], and the reverse Cuthill-McKee algorithm (see [45, 46]), to generate a sparse banded linear system, for which band LU approaches [47] represent the solution method of choice. Though Cuthill-McKee type approaches indeed recover $O(n)$ scaling for the solution of such problems, the computational cost is significantly higher than that following the approach discussed herein, which couples structure-dependent modifications of the Thomas algorithm itself together with appropriate grid ordering. Note that in the following we assume that all the matrices arising from the discretization are diagonally dominant, and thus no pivoting is required during the factorization process.

The outline of this chapter is as follows: Section 5.2 introduces the *m*-Legged Thomas algorithm, which enables the fast direct solution of linear systems arising when gridpoints are arranged along segments which are connected at one nodal point only. Section 5.3 introduces the Box Thomas algorithm, which extends the circulant Thomas algorithm to the solution of linear systems arising when gridpoints are arranged along 3D box-shaped wireframe structures. Section 5.4 shows how to derive fast factorization algorithms for the solution of linear systems arising when gridpoints are arranged along more complicated wireframe structures with nontrivial nodal connections.

5.2 The *m*-legged Thomas algorithm

Consider the solution of the PDE (5.1) defined over a 1D domain comprised of m legs, each discretized on p points, meeting at a common nodal point. The discretized version of (5.1) on this connected domain decouples into m almost-independent linear systems, sharing information only through the central nodal point, and may be written

$$\begin{bmatrix} b_1^{(i)} & c_1^{(i)} & & & & & \\ a_2^{(i)} & b_2^{(i)} & c_2^{(i)} & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_{p-1}^{(i)} & b_{p-1}^{(i)} & c_{p-1}^{(i)} & & \\ & & & a_p^{(i)} & b_p^{(i)} & c_p^{(i)} \end{bmatrix} \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_p^{(i)} \\ x_{\text{center}} \end{pmatrix} = \begin{pmatrix} g_1^{(i)} \\ g_2^{(i)} \\ \vdots \\ g_p^{(i)} \end{pmatrix} \qquad i = 1, \ldots, m, \tag{5.3a}$$

$$\begin{bmatrix} d_1 & d_2 & \ldots & d_m & d_{m+1} \end{bmatrix} \begin{pmatrix} x_p^{(1)} \\ x_p^{(2)} \\ \vdots \\ x_p^{(m)} \\ x_{\text{center}} \end{pmatrix} = g_{\text{center}}. \tag{5.3b}$$

This system of $mp + 1$ equations in $mp + 1$ unknowns can be efficiently solved by first performing m forward sweeps of Gauss elimination to reduce each tridiagonal matrix on the LHS of (5.3a), for $i = 1, \ldots, m$, to upper bidiagonal form, i.e.

$$\begin{bmatrix} b_1^{(i)} & c_1^{(i)} & & & & & \\ 0 & b_2^{(i)} & c_2^{(i)} & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & 0 & b_{p-1}^{(i)} & c_{p-1}^{(i)} & & \\ & & & 0 & b_p^{(i)} & c_p^{(i)} \end{bmatrix} \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_p^{(i)} \\ x_{\text{center}} \end{pmatrix} = \begin{pmatrix} g_1^{(i)} \\ g_2^{(i)} \\ \vdots \\ g_p^{(i)} \end{pmatrix} \qquad i = 1, \ldots, m, \tag{5.4}$$

where, as usual, the b and g elements are changed during this forward sweep. This step of the algorithm is embarrassingly parallel, and may easily be performed in m independent threads. The last row of each resulting linear system in (5.4) is then extracted and assembled together with (5.3b) to compose the following system of $m+1$ equations in $m+1$ unknowns:

$$\begin{bmatrix} b_p^{(1)} & & & & c_p^{(1)} \\ & b_p^{(2)} & & & c_p^{(2)} \\ & & \ddots & & \vdots \\ & & & b_p^{(m)} & c_p^{(m)} \\ d_1 & d_2 & \dots & d_m & d_{m+1} \end{bmatrix} \begin{pmatrix} x_p^{(1)} \\ x_p^{(2)} \\ \vdots \\ x_p^{(m)} \\ x_{\text{center}} \end{pmatrix} = \begin{pmatrix} g_p^{(1)} \\ g_p^{(2)} \\ \vdots \\ g_p^{(m)} \\ g_{\text{center}} \end{pmatrix}. \tag{5.5}$$

The linear system in (5.5), due to the arrowhead structure of the matrix, can be solved in $\sim 8m$ operations. By expressing all of the $x_p^{(i)}$ unknowns with respect to x_{center}, and then replacing them in the last row, it is possible to directly calculate $x_p^{(i)}$ and x_{center} as follows:

$$x_p^{(i)} = \left(g_p^{(i)} - c_p^{(i)}\, x_{\text{center}}\right) / b_p^{(i)} \quad \text{for } i = 1, \dots, m; \tag{5.6a}$$

$$x_{\text{center}} = \left(e - \sum_{i=1}^{m} d_i\, g_p^{(i)} / b_p^{(i)}\right) \Big/ \left(d_{m+1} - \sum_{i=1}^{m} d_i\, c_p^{(i)} / b_p^{(i)}\right); \tag{5.6b}$$

that is, once x_{center} is determined from (5.6b), the $x_p^{(i)}$ unknowns may be determined from (5.6a). At this point, the remaining unknowns ($x_{p-1}^{(i)}$ through $x_1^{(i)}$ for $i = 1, \dots, m$) are then calculated by performing back substitution in each of the upper bidiagonal systems in (5.4), starting from the $(p-1)$th row and working up. Again, this step is embarrassingly parallel across m independent threads. A pseudo-code illustrating this m-legged version of the Thomas algorithm is given in Algorithm 5.1, where the vector of unknowns $x^{(i)}$ and x_{center} are stored in g and $e = g_{\text{center}}$, respectively, for optimized storage. The computational cost of this algorithm is as follows: $5m(p-1)$ flops for the first double loop (lines 2 to 8), $9m+1$ flops for the two single loops (lines 9 to 16), and $3m(p-1)$ flops for the last double loop (lines 17 to 21). Thus, overall, a system of equations like that in (5.3) requires $8mp+m+1$ flops to solve.

Algorithm 5.1 m-legged Thomas

```
function mlThomas($a, b, c, d, e, g, p, m$)
    for i = 1 : m do
        for j = 2 : p do
            $a_j^{(i)} \leftarrow -a_j^{(i)}/b_{j-1}^{(i)}$
            $b_j^{(i)} \leftarrow b_j^{(i)} + a_j^{(i)}\, c_{j-1}^{(i)}$
            $g_j^{(i)} \leftarrow g_j^{(i)} + a_j^{(i)}\, g_{j-1}^{(i)}$
    for i = 1 : m do
        $d_i \leftarrow -d_i/b_p^{(i)}$
        $e \leftarrow e + d_i\, g_p^{(i)}$
        $d_{m+1} \leftarrow d_{m+1} + d_i\, c_p^{(i)}$
    $e \leftarrow e/d_{m+1}$
    for i = 1 : m do
        $g_p^{(i)} \leftarrow \left(g_p^{(i)} - c_p^{(i)}\, e\right)/b_p^{(i)}$
    for i = 1 : m do
        for j = p-1 : -1 : 1 do
            $g_j^{(i)} \leftarrow \left(g_j^{(i)} - c_j^{(i)}\, g_{j+1}^{(i)}\right)/b_j^{(i)}$
```

Algorithm 5.2 Circulant Thomas - Minimum Storage

```
function CircThomas_MS($a, b, c, g, m$)
    for i = 2 : m-1 do
        $a_i \leftarrow -a_i/b_{i-1}$
        $b_i \leftarrow b_i + a_i\, c_{i-1}$
        $a_i \leftarrow a_i\, a_{i-1}$
        $g_i \leftarrow g_i + a_i\, g_{i-1}$
    $a_{m-1} \leftarrow (a_{m-1} + c_{m-1})/b_{m-1}$
    $g_{m-1} \leftarrow g_{m-1}/b_{m-1}$
    for i = m-2 : -1 : 1 do
        $a_i \leftarrow (a_i - c_i\, a_{i+1})/b_i$
        $g_i \leftarrow (g_i - c_i\, g_{i+1})/b_i$
    $g_m \leftarrow (g_m - c_m\, g_1 - a_m\, g_{m-1})/(b_m - c_m\, a_1 - a_m\, a_{m-1})$
    for i = 1 : m-1 do
        $g_i \leftarrow g_i - a_i\, g_m$
```

Algorithm 5.3 Circulant Thomas

```
 1: function CircThomas($a$, $b$, $c$, $g$, $m$)
 2:     $d_1 = a_1$
 3:     for i = 2 : m-1 do
 4:         $a_i \leftarrow -a_i/b_{i-1}$
 5:         $b_i \leftarrow b_i + a_i\, c_{i-1}$
 6:         $d_i = a_i\, d_{i-1}$
 7:         $g_i \leftarrow g_i + a_i\, g_{i-1}$
 8:     $d_{m-1} \leftarrow (d_{m-1} + c_{m-1})/b_{m-1}$
 9:     $g_{m-1} \leftarrow g_{m-1}/b_{m-1}$
10:     for i = m-2 : -1 : 1 do
11:         $d_i \leftarrow (d_i - c_i\, d_{i+1})/b_i$
12:         $g_i \leftarrow (g_i - c_i\, g_{i+1})/b_i$
13:     $g_m \leftarrow (g_m - c_m\, g_1 - a_m\, g_{m-1})/(b_m - c_m\, d_1 - a_m\, d_{m-1})$
14:     for i = 1 : m-1 do
15:         $g_i \leftarrow g_i - d_i\, g_m$
```

5.3 The Box Thomas algorithm

We now consider closed wireframe structures, beginning with a simple loop, for which the discretization of (5.1) takes the form

$$\begin{bmatrix} b_1 & c_1 & & & a_1 \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{m-1} & b_{m-1} & c_{m-1} \\ c_m & & & a_m & b_m \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \\ x_m \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_{m-1} \\ g_m \end{pmatrix}. \tag{5.7}$$

The circulant tridiagonal system (5.7) can be solved using a circulant Thomas solver, such as that presented in [38]. A minimal-storage implementation of this algorithm is given in Algorithm 5.2, where the vector of unknowns x is returned in g. This algorithm requires $6(m-2)$ flops for the first loop (lines 2 to 7), 3 flops for lines 8 and 9, $6(m-2)$ flops for the second loop (lines 10 to 13), 9 flops for line 14, and $2(m-1)$ flops for the third loop (lines 15 to 17). Overall, for a $m\times m$ system like that in (5.7), the circulant Thomas solver requires $14m-16$ flops, which is $\sim 75\%$ more expensive than the standard Thomas algorithm. If the

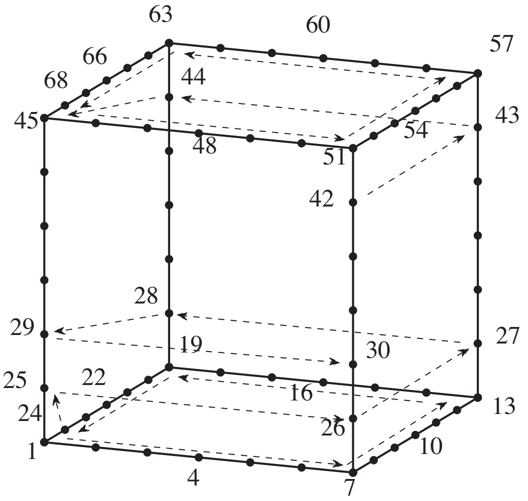


Figure 5.1: Point arrangement for the construction of the linear system associated with the box-shaped wireframe topology.

LU decomposition of A needs to be computed, an extra vector d_i for $i = 1, \dots, m-1$ must be used, as illustrated in Algorithm 5.3; leveraging a precomputed LU decomposition, the computational cost of circulant Thomas reduces to $\sim 7m$.

For 3D box-shaped wireframe structures, some care is required with the point arrangement in order to reduce the resulting discretized PDE problem to a form that can be solved in $O(n)$ operations. The ordering we propose is shown in Figure 5.1 for the case with $p = 5$ gridpoints on each edge (excluding vertices). Overall, the discretization includes $n = 12p + 8$ points. We begin by defining one of the faces of the 3D box as the *lower face*, and the opposite face as the *upper face*. The four edges defining each base are composed of $m = 4p+4$ points. Starting from one of its vertices, all of the gridpoints on the lower face are first enumerated counter-clockwise. The next point (point 25 in Figure 5.1) is that point connected to point 1 which does not belong to the lower face. The following points are the remaining points connected to the vertices of the lower face, enumerated in counter-clockwise order. Enumeration proceeds in an upward-spiraling manner until the upper face is reached. Here, enumeration follows using the same scheme as adopted for the lower face: the first point (point 45 in Figure 5.1) shares an edge with point 1 in the lower face, and point enumeration proceeds along the edges of the upper face in counter-clockwise order (ending with point 68 in Figure 5.1).

The sparsity structure of the resulting block tridiagonal matrix in this problem, denoted A, is illustrated in Figure 5.2a. Such a matrix is composed of a circulant tridiagonal block T_l on the main block diagonal, involving the $m = 4p+4$ gridpoints on the lower face. Two $4{\times}m$ rectangular blocks R_l and S_l appear on the lower and upper block diagonals of the matrix. Each of these blocks has only 4 nonzero elements. Along the main block diagonal, the block T_l is followed by a sequence of p diagonal blocks D_i of size 4×4. Other 4×4 diagonal blocks appear on the lower and upper block diagonals, E_i and F_i, respectively. These blocks account for the points along the edges connecting the lower and upper faces. The remaining m gridpoints on the upper face form another circulant tridiagonal block T_u on the main block diagonal. Again, two $4 \times m$ rectangular blocks R_u and S_u appear on the first lower and upper block diagonals, respectively. As observed for the lower face, these matrices have only 4 nonzero elements. Significantly, the point arrangement proposed here does *not* minimize the bandwidth of the present linear system, and thus would not be identified by simple application of the Cuthill-McKee algorithm to the linear system arising in this problem. However, this special ordering allows us to identify a solution algorithm that minimizes the number of flops required to solve, as described below.

The solution of the linear system $A\,x = g$ in this problem begins by performing a forward sweep inspired by block Gauss elimination, designed to eliminate the blocks E_i with $i = 2, \ldots, p$ along the block lower diagonal, as well as the block R_u^T. This creates certain fill-ins, specifically below R_l. We then perform a backward sweep inspired by the second step of block Gauss elimination, which zeros the blocks F_i with $i = p-1, \ldots, 1$

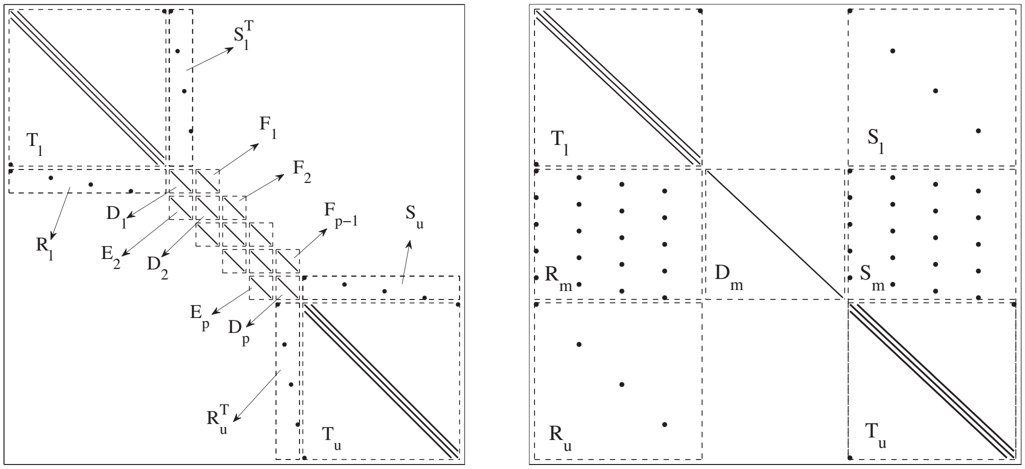


Figure 5.2: Sparsity pattern of the matrix *A* associated with the Box Thomas algorithm (left) before, and (right) after the forward and backward sweeps.

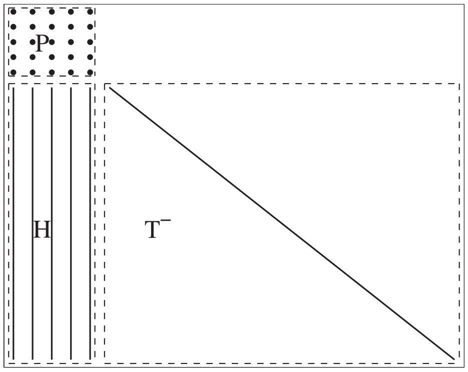


Figure 5.3: Sparsity pattern of the matrix on the LHS of (5.11) after rearranging in the Box Thomas algorithm.

along the block upper diagonal, as well as the block S_l^T. This also creates additional fill-ins, specifically above the block S_u. The resulting sparsity structure of A after these two sweeps is illustrated in Figure 5.2b, and is equivalent to the following linear system of equations:

$$T_l\, x_l \qquad\qquad + S_l\, x_u = g_l, \tag{5.8a}$$

$$R_m\, x_l + D_m\, x_m + S_m\, x_u = g_m, \tag{5.8b}$$

$$R_u\, x_l \qquad\qquad + T_u\, x_u = g_u, \tag{5.8c}$$

where x and g are partitioned such that $x = (x_l, x_m, x_u)$ and $g = (g_l, g_m, g_u)$ and, of course, the T_l and T_u matrices have been modified from their original form. Considering (5.8a), it is possible to determine x_l as a function of x_u^V, where superscript V indicates a partition of the solution at the points x_u involving only the four vertices of the upper face. Denoting as S_l^V a partition of the columns of S_l defined likewise, it is possible to define x_l as

$$x_l = x_l^{(0)} + \sum_{i=1}^{4} x_l^{(i)}\, x_u^{V_i} \tag{5.9}$$

where each $x_l^{(i)}$ is obtained from the solution of a circulant tridiagonal system, i.e.

$$T_l\, x_l^{(0)} = g_l \tag{5.10a}$$

$$T_l\, x_l^{(i)} = -s_l^{V_i} \qquad i = 1, \ldots, 4 \tag{5.10b}$$

where each $s_l^{V_i}$ represents the ith column of the partition S_l^V. Substituting (5.9) into (5.8c) gives

$$\left(T_u + R_u\, X_l^{(1:4)}\, V_u\right) x_u = g_u - R_u\, x_l^{(0)} = r_u \tag{5.11}$$

where $X_l^{(1:4)}$ is a $m \times 4$ matrix grouping the four solutions $x_l^{(i)}$, and V_u is the extraction matrix defined as $x_u^V = V_u\, x_u$. Due to the extreme sparsity of R_u, computation of both the RHS and the LHS of (5.11) can be performed in $O(1)$ operations. The matrix on the LHS of (5.11) is a superposition of a tridiagonal circulant matrix and a matrix which is empty except in four columns, corresponding to the positions of the vertices of the upper face. After performing a forward sweep in order to eliminate the elements on the lower diagonal and, likewise, a backward sweep in order to eliminate the elements on the upper diagonal, and rearranging to put the solution at the vertices of the upper face first, it is possible to partition the resulting matrix as illustrated in Figure 5.3, where P is a 5×5 full matrix, H is a $(4p-1) \times 5$ full matrix, and T^- is a $(4p-1) \times (4p-1)$ diagonal matrix. [Note, of course, that such matrix rearrangement is done here for illustration purposes only; the subsequent computations may actually be performed in place in the numerical implementation.] The solution of the associated linear system can be rewritten as

$$P\, x_u^W \qquad\qquad = r_u^W, \tag{5.12a}$$

$$H\, x_u^W + T^-\, x_u^R = r_u^R, \tag{5.12b}$$

where the x_u is partitioned such that $x_u = \{x_u^W, x_u^R\}$, where $x_u^W = \{x_u^V, x_u^{\text{last}}\}$ contains the values of the x at the four vertices of the upper face together with the last component of x_u (that is, point 68 in Figure 5.1), and the RHS $r_u = \{r_u^W, r_u^R\}$ is partitioned analogously. By (5.12a), x_u^W may be determined from the solution of a (full) 5×5 system. Then, x_u^R is determined by solution of the subsequent diagonal system in (5.12b). Finally, x_l is given by (5.9), and x_m is given by solution of the diagonal system in (5.8b).

We now calculate the computational cost of the algorithm described above. Elimination of the E_i matrices requires $\sim 24p$ flops, and elimination of the F_i requires $\sim 24p$ more flops. Five solutions of an $m \times m$ circulant tridiagonal system are required for the determination of the x_l^i for $i = 0, \ldots, 4$: since only the RHS changes every time, it is possible to calculate the LU decomposition in $\sim 28p$ flops, and then the calculation of the five solutions requires an extra $\sim 140p$ flops. The determination of x_u then requires $\sim 152p$ flops. Then, calculation of x_l requires $\sim 32p$ flops, plus $\sim 20p$ for x_m. Overall, $\sim 420p$ flops are needed; since $n \sim 12p$, this means that $\sim 35n$ flops are required.

A minimal storage implementation of Box Thomas algorithm is given in Algorithm 5.4. The algorithm calculates the solution of the system and stores it into the RHS vector, which has been split into three vectors g^l, g^m, and g^u, which contain the solution at the gridpoints on the lower face, intermediate edges, and upper face, respectively. The three diagonals containing the nonzero elements of T_l are stored in a^l, b^l, and c^l; likewise, the nonzero elements of T_u are stored in a^u, b^u, and c^u. Auxiliary vectors d^l and d^u are also introduced, which are needed while solving the two circulant systems associated with T_l and T_u (d^l does not appear directly in BoxThomas, but is defined within CircThomas). In addition, matrices d^m, e^m, and f^m of size $4p \times 4$ are defined: d^m contains the main diagonals of the 4×4 matrices D_i for $i = 1 : p$; e^m contains the main diagonals of E_i for $i = 2 : p$, and the four nonzero elements of R_u^T in Figure 5.2 in the last row; f^m contains the four nonzero elements of S_l^T in the first row and the main diagonals of F_i for $i = 1 : p - 1$. Two additional $p \times 4$ matrices r^m and s^m are introduced: the first row of r^m contains the four nonzero elements of R_l, while the last row of s_m contains the nonzero elements of S_u. Two initially empty $m \times 4$ matrices s^l and r^u are defined, which are used while solving the circulant linear systems associated to the points on the lower and upper faces. A temporary scalar variable t is used to compute the product $R_u X_l^{(1:4)}$ in (5.11). To simplify the notation, a vector of indices V_j, for $j = 1, \ldots, 4$, is defined which contains the positions of the gridpoints on the vertices of the lower or upper face, i.e. $V_j = (p + 1)(j - 1) + 1$. Remarkably, though an LU decomposition is not computed, much of the computation performed does not need to be repeated for a different RHS g, since only the matrix r^u needs to be recomputed, while all the other matrices and vectors may be reused. In this case, the algorithm requires only $\sim 220p$ flops (that is, $\sim 18.\overline{3}n$ flops), which is roughly half the computational cost of solving the original linear system from scratch.

Application of the reverse Cuthill-McKee algorithm [45] to matrix A in Figure 5.2

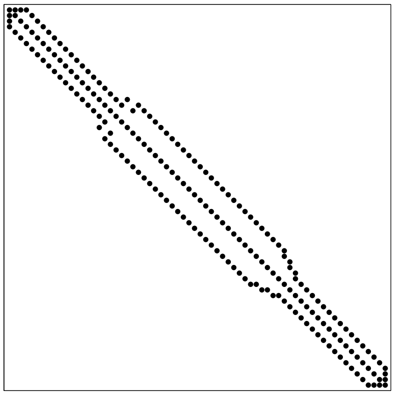


Figure 5.4: Sparsity structure of the matrix A in the Box Thomas problem after reordering for bandwidth minimization by the reverse Cuthill-McKee algorithm.

produces a matrix with lower and upper bandwidth b equal to six (Figure 5.4). This result is independent of the number of points along the edges p, and depends solely on the topology of the 3D box. As discussed in [47], application of a linear solver for band matrices of size $n \times n$ and given bandwidth b, for $n \gg b$, requires $\sim (2b^2+5b+1)n$ flops. Thus, application of such a solver to the solution of the 3D box problem, with a point arrangement minimizing matrix bandwidth, has an overall computational cost of $\sim 103n$, which is nearly *three times* more expensive than the approach described in this chapter.

The extension of the algorithm described in this section to cases with a different number of points on the edges in each dimension (namely p_x, p_y, and p_z), is trivial. Assuming the lower and upper faces of the box are made of $(2p_x+2p_y+4)$ points, the solution of the associated matrix requires $\sim (176p_x + 176p_y + 68p_z)$ flops. Thus, for efficiency, the box should be arranged such that the lower and upper faces contain the smallest number of gridpoints possible.

Similarly, the extension from a cube, interpreted as a prism with a square base, to a prism with an N-sided base (e.g., a triangular, pentagonal, or hexagonal prism) is also trivial, simply by changing 4 to N in Algorithm 5.4 and the accompanying discussion.

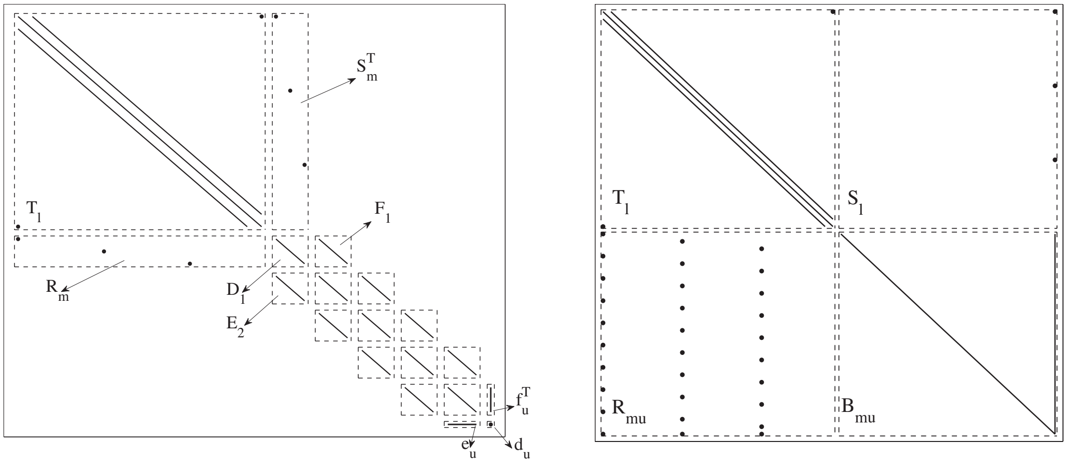


Figure 5.5: Sparsity pattern of the matrix A associated with the Tetrahedral Thomas algorithm (left) before, and (right) after the forward and backward sweeps.

5.4 Extensions of the Box Thomas algorithm to other wireframe topologies

The Box Thomas algorithm developed in §5.3 may be extended easily to handle the sparse linear systems arising from the discretization of 1D PDEs on a variety of different 1D wireframe structures. We now consider the efficient solution of the discretization of 1D PDEs on the edges of the four other platonic solids: the tetrahedron, octahedron, dodecahedron, and icosahedron.

5.4.1 Tetrahedron

In the tetrahedral case, following the approach of §5.3, denote one of the triangular faces as the *base* and, starting from one of the vertices of the base, enumerate all points along the edges of the base counter-clockwise. Then, enumerate the points along the three edges departing the base, one edge at a time, in an upward-spiraling manner until the *apex* is reached. Assuming an equal number of points on each edge of the tetrahedron, p (excluding the vertices), there are $(3p+3)$ points on the base, $3p$ points along the three edges departing from it, and 1 point at the apex; overall, the system is composed of $n = 6p + 4$ points.

The sparsity structure of the resulting block tridiagonal matrix in this problem is illustrated in Figure 5.5a. As with the Box Thomas algorithm discussed previously, a forward sweep is first applied to eliminate the E_i blocks, for $i = 2, \dots, p$, and the vector e_u; this process creates fill-ins below the matrix R_m. Then, a backward sweep is applied to eliminate the F_i blocks, for $i = p - 1, \dots, 1$, and the block S_m^T; this process creates fill-ins above the vector f_u^T. The resulting matrix is shown in Figure 5.5b. The solution of the associated linear system may be written

$$T_l\, x_l + S_l\, x_u = g_l \tag{5.13a}$$

$$R_u\, x_l + B_u\, x_u = g_u \tag{5.13b}$$

where x_l contains all the gridpoints along the base, and x_u the remaining ones. Noting that the matrix S_l is composed of one nonzero column only and following the approach used in §5.3, we may express x_l as

$$x_l = x_l^{(0)} + x_l^{(1)}\, x_u^V, \tag{5.14}$$

where the superscript V indicates a partition of vector x_u containing the apex only, and vectors $x_l^{(0)}$ and $x_l^{(1)}$ are the solutions of

$$T_l\, x_l^{(0)} = g_l, \tag{5.15a}$$

$$T_l\, x_l^{(1)} = -s_l^V, \tag{5.15b}$$

where s_l^V represents the column of S_l associated to the apex. Substituting (5.14) into (5.13)

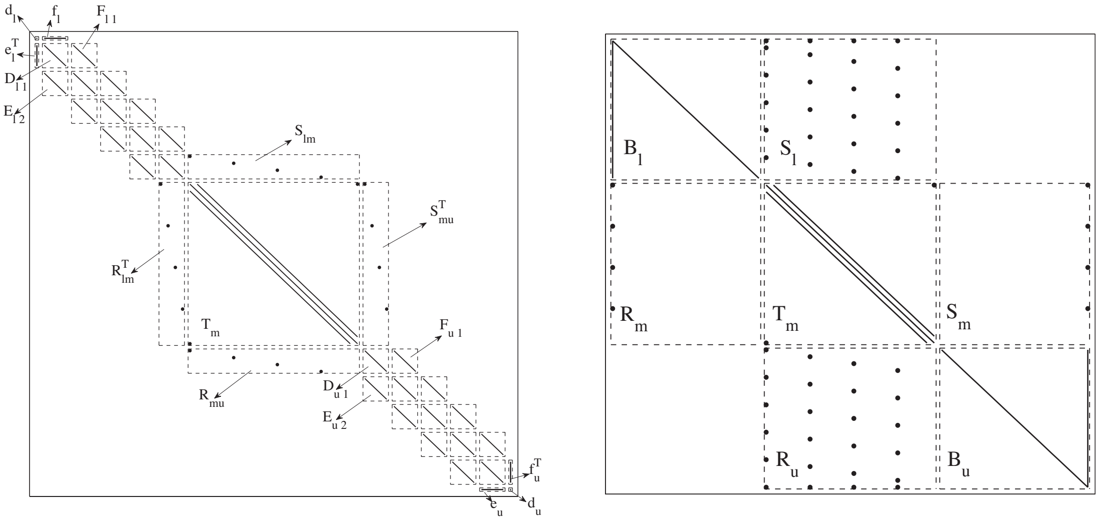


Figure 5.6: Sparsity pattern of the matrix A associated with the Octahedral Thomas algorithm (left) before, and (right) after the forward and backward sweeps.

gives

$$\left(B_u + R_u\, x_l^{(1)}\, V_u\right) x_u = g_u - R_u\, x_l^{(0)} \tag{5.16}$$

where V_u is the matrix extracting the apex from the vector x_u. Note that the matrix on the LHS is diagonal plus a full last column; thus, this linear system may be solved in $O(n)$ operations via a single sweep of back substitution. In the rest of the chapter we will refer to matrices with this shape as V-shaped matrices. Substituting the resulting solution for x_u into (5.14) gives the rest of the solution.

The extension from a tetrahedron, interpreted as a pyramid with a triangular base, to a pyramid with an N-sided base (e.g., a square, pentagonal, or hexagonal pyramid) is entirely straightforward.

5.4.2 Octahedron

The point enumeration in the octahedral case involves defining one vertex as the *lower vertex*, the opposite vertex as the *upper vertex*, and the square connecting the other four vertices as the *midplane*. Points are enumerated starting from the lower vertex and proceeding one point from each edge, in a counter-clockwise upward-spiraling manner, until the midplane is reached. The points on the midplane are then enumerated counter-clockwise, and the enumeration proceeds on the four edges connected to the upper vertex, one point per edge, until the upper vertex is reached. Overall, denoting by p the number of points on each edge (excluding vertices), we have 1 point for the lower vertex, $4p$ points on the four edges connecting the lower vertex to the midplane, $(4p+4)$ points on the midplane, $4p$ points on the four edges connecting the midplane to the upper vertex, and 1 point for the upper vertex; overall, there are $n = 12p + 6$ points.

The sparsity structure of the resulting block tridiagonal matrix in this problem is illustrated in Figure 5.6a. The first step involves a forward sweep to eliminate the matrices $E_{l\,i}$ for $i = 2, \ldots, p$; this creates fill-ins below the vector e_l^T. This forward sweep is then continued to eliminate the matrices $E_{u\,i}$ for $i = 2, \ldots, p$, as well the vector e_u; this creates additional fill-ins below the matrix R_{mu}. A backward sweep is then applied to eliminate the matrices $F_{u\,i}$ for $i = p - 1, \ldots, 1$, and is continued to eliminate the matrices $F_{l\,i}$ for $i = p - 1, \ldots, 1$, as well as the vector f_l. This creates fill-ins above f_u^T, and above S_{lm}. The resulting matrix is shown in Figure 5.6b. The solution of the associated linear system may be written

$$B_l\, x_l \;\; + S_l\, x_m \qquad\qquad = g_l, \tag{5.17a}$$

$$R_m\, x_l + T_m\, x_m + S_m\, x_u = g_m \tag{5.17b}$$

$$R_u\, x_m + B_u\, x_u = g_u. \tag{5.17c}$$

Considering (5.17a), we may express x_l as

$$x_l = x_l^{(0)} + \sum_{i=1}^{4} x_l^{(i)}\, x_m^{V_i}, \tag{5.18}$$

where each $x_l^{(i)}$ requires the solution of another V-shaped system, i.e.

$$B_l\, x_l^{(0)} = g_l, \tag{5.19a}$$

$$B_l\, x_l^{(i)} = -s_l^{V_i} \qquad i = 1, \ldots, 4, \tag{5.19b}$$

where $s_l^{V_i}$ represents each of the 4 nonzero columns of S_l, associated with each of the 4 vertices in the midplane of the octahedron. Likewise, x_u may be expressed as a function of x_m^V:

$$x_u = x_u^{(0)} + \sum_{i=1}^{4} x_u^{(i)}\, x_m^{V_i} \tag{5.20}$$

with

$$B_u\, x_u^{(0)} = g_u, \tag{5.21a}$$

$$B_u\, x_u^{(i)} = -r_u^{V_i} \qquad i = 1, \ldots, 4. \tag{5.21b}$$

Substituting (5.18) and (5.20) into (5.17b) gives

$$\left(T_m + R_m X_l^{(1:4)}\, V_m + S_m X_u^{(1:4)}\, V_m\right) x_m = g_m - R_m\, x_l^{(0)} - S_m\, x_u^{(0)} \tag{5.22}$$

The matrix on the LHS of (5.22) is tridiagonal except for five columns in which additional nonzero terms appear. To proceed, a forward sweep is first applied to eliminate the first subdiagonal, and a backward sweep is applied to eliminate the first superdiagonal; this

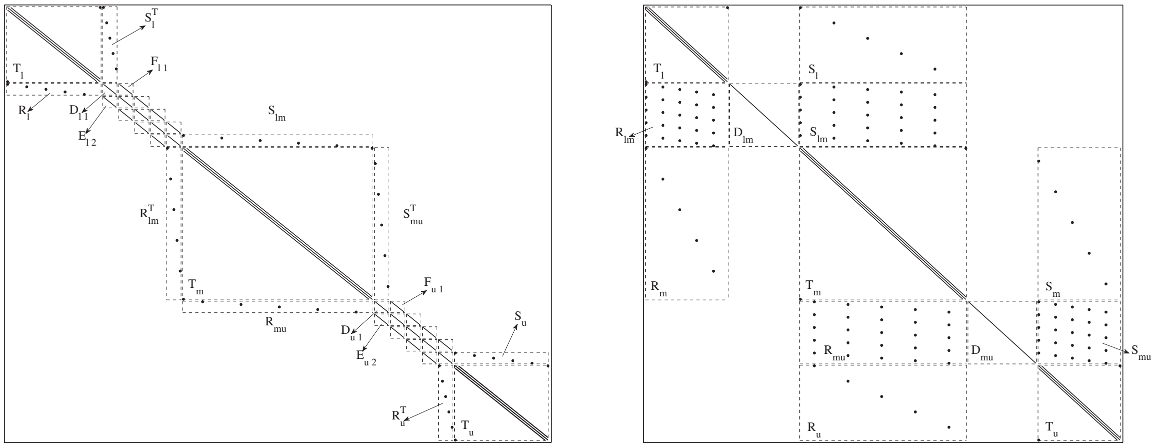


Figure 5.7: Sparsity pattern of the matrix *A* associated with the Dodecahedral Thomas algorithm (left) before, and (right) after the forward and backward sweeps.

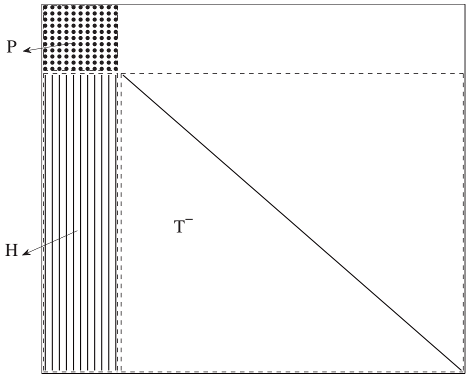


Figure 5.8: Sparsity pattern of the matrix on the LHS of (5.28) after rearranging in the Dodecahedral Thomas algorithm.

creates additional fill-ins within the columns with the additional nonzero terms mentioned previously. Rearranging the matrix in order to have these five columns appear first gives a matrix exactly like that illustrated in Figure 5.2, for which the solution method follows exactly as given in (5.12). Once x_m has been determined, substitution into (5.18) and (5.20) gives the rest of the solution.

5.4.3 Dodecahedron

The point enumeration in the dodecahedral case involves defining one pentagonal face as the *lower face*, the opposite face as the *upper face*, and the ten central vertices together with the edges connecting them as the *middle crown*. The point enumeration then follows in a manner analogous to the octahedral case. The first points to be enumerated are those around the edges of the lower face, in a counter-clockwise fashion (starting from a vertex). The points along the five edges connecting the lower face to the middle crown are then enumerated, in an upward-spiraling fashion as before. Once the first vertex of the middle crown is reached, the remaining points of the middle crown are enumerated counter-clockwise, similar to what was done for the midplane of the octahedron. The enumeration continues in an upward-spiraling fashion over the points connecting the middle crown to the upper face. Once the first vertex of the upper face is reached, enumeration proceeds counter-clockwise around the edges of the upper face.

The sparsity structure of the resulting block tridiagonal matrix in this problem is illustrated in Figure 5.7a. Again, we start with a forward sweep to eliminate the $E_{l\,i}$, R^T_{lm}, $E_{u\,i}$, and R^T_u matrices on the lower block diagonal, and a bacward sweep to eliminate the $F_{u\,i}$, S^T_{mu}, $F_{l\,i}$, and S^T_l matrices on the upper block diagonal, which creates fill-ins below R_l and R_{mu}, and above S_u and S_{lm}, as shown in Figure 5.7b. The resulting system can be written as

$$T_l\,x_l \qquad + S_l\,x_m \qquad = g_l \tag{5.23a}$$

$$R_{lm}\,x_l + D_{lm}\,x_{lm} + S_{lm}\,x_m \qquad = g_{lm} \tag{5.23b}$$

$$R_m\,x_l \qquad + T_m\,x_m \qquad + S_m\,x_u = g_m \tag{5.23c}$$

$$R_{mu}\,x_m + D_{mu}\,x_{mu} + S_{mu}\,x_u = g_{mu} \tag{5.23d}$$

$$R_u\,x_m \qquad + T_u\,x_u = g_u \tag{5.23e}$$

Noting that the R and S matrices all have only 5 nonzero columns, leveraging (5.23a), it is possible to express x_l and x_u as functions of x_m, i.e.

$$x_l = x_l^{(0)} + \sum_{i=1}^{5} x_l^{(i)}\, x_m^{V_i^l} \tag{5.24}$$

where

$$T_l\, x_l^{(0)} = g_l, \tag{5.25a}$$

$$T_l\, x_l^{(i)} = -s_l^{V_i^l} \qquad i = 1, \ldots, 5, \tag{5.25b}$$

and

$$x_u = x_u^{(0)} + \sum_{i=1}^{5} x_u^{(i)}\, x_m^{V_i^u}, \tag{5.26}$$

where

$$T_u\, x_l^{(0)} = g_u, \tag{5.27a}$$

$$T_u\, x_l^{(i)} = -r_u^{V_i^u} \qquad i = 1, \ldots, 5, \tag{5.27b}$$

where $s_l^{V_i^l}$ and $r_u^{V_i^u}$ represent the ith nonzero column of the matrices S_l and R_u, respectively. Distinct from the octahedral case, the nonzero columns of S_l and R_u are not aligned, as the vertices on the lower and upper face connect to the middle crown through different points. Substituting (5.24) and (5.26) into (5.23c) gives

$$\left(T_m + R_m X_l^{(1:5)} V_{m1} + S_m X_u^{(1:5)} V_{m2}\right) x_m = g_m - R_m\, x_l^{(0)} - S_m\, x_u^{(0)} \tag{5.28}$$

The matrix on the LHS is tridiagonal with additional nonzero terms in 11 columns. As done previously, rearranging to put the 11 full columns first, and performing forward and backward sweeps to eliminate the lower and upper subdiagonals of the lower-right block, results in the matrix illustrated in Figure 5.8, where P is an 11×11 full matrix, H is a $(10p - 1) \times 11$ full matrix, and T^- is a $(10p - 1) \times (10p - 1)$ diagonal matrix. Again, the solution follows in a manner analogous to (5.12). After x_m has been determined, x_l is determined through (5.24), and x_u through (5.26). Finally, x_{lm} and x_{mu} are obtained via (5.23b) and (5.23d).

5.4.4 Icosahedron

The point enumeration in the icosahedral case involves defining one vertex as the *lower vertex*, the opposite vertex as the *upper vertex*, the five vertices adjacent to the lower vertex (and the edges connecting them) as the *lower plane*, and the five vertices adjacent to the upper vertex (and the edges connecting them) as the *upper plane*. The point enumeration then follows as before. Starting from the lower vertex, proceed in a counter-clockwise, upward spiraling fashion along the five edges connecting to the lower plane. Once the first vertex of the lower plane is reached, the remaining points of the lower plane are enumerated counter-clockwise. Then, proceed in a counter-clockwise, upward spiraling fashion along the ten edges connecting to the upper plane. Once the first vertex of the upper plane is reached, the remaining points of the upper plane are enumerated counter-clockwise. Then, proceed in a counter-clockwise, upward spiraling fashion along the five edges connecting to the upper vertex.

The sparsity structure of the resulting block tridiagonal matrix in this problem is illustrated in Figure 5.9a. As before, a forward sweep is used to eliminate $E_{l\,i}$, R_{lm}^T, $E_{m\,i}$, $R_{m_2}^T$, $E_{u\,i}$ and e_u, and a backward sweep is used to eliminate $F_{u\,i}$, S_{mu}^T, $F_{m\,i}$, $S_{m_1}^T$, $F_{l\,i}$ and

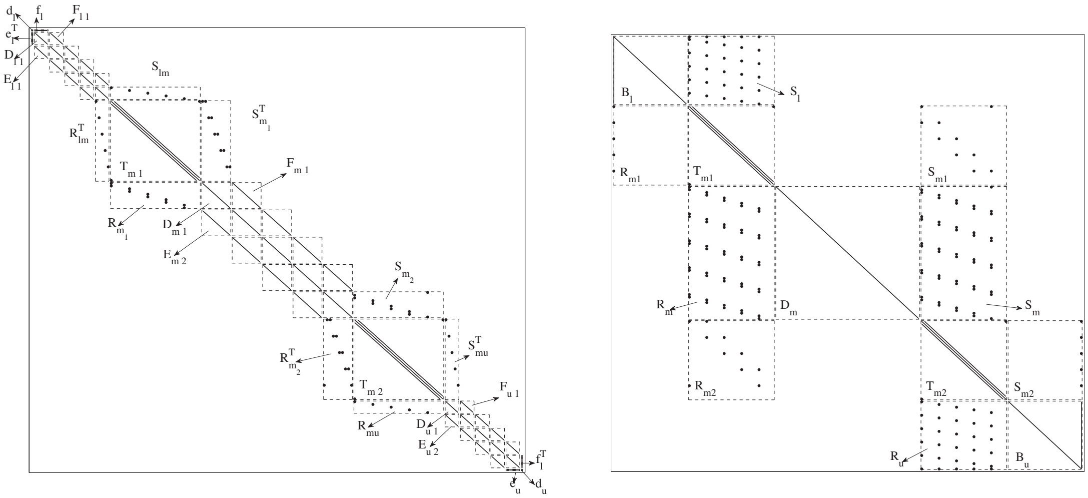


Figure 5.9: Sparsity pattern of the matrix *A* associated with the Icosahedral Thomas algorithm (left) before, and (right) after the forward and backward sweeps.

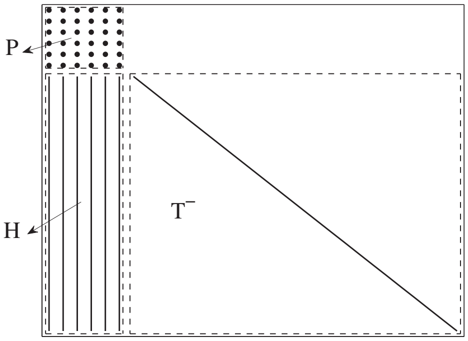


Figure 5.10: Sparsity pattern of the matrix on the LHS of (5.34) after rearranging in the Icosahedral Thomas algorithm.

f_l, which creates fill-ins below e_l^T, R_{m_1}, and R_{mu}, and above f_u^T, S_{m_2}, and S_{lm}, as shown in Figure 5.9b. The resulting system can be written as

$$B_l\, x_l + S_l\, x_{m1} = g_l \tag{5.29a}$$

$$R_{m1}\, x_l + T_{m1}\, x_{m1} + S_{m1}\, x_{m2} = g_{m1} \tag{5.29b}$$

$$R_m\, x_{m1} + D_m\, x_m + S_m\, x_{m2} = g_m \tag{5.29c}$$

$$R_{m2}\, x_{m1} + T_{m2}\, x_{m2} + S_{m2}\, x_u = g_{m2} \tag{5.29d}$$

$$R_u\, x_{m2} + B_u\, x_u = g_u \tag{5.29e}$$

Leveraging (5.29a) and (5.29e), we may express x_l and x_u as functions of x_{m1} and x_{m2}, respectively, i.e.

$$x_l = x_l^{(0)} + \sum_{i=1}^{5} x_{m1}^{(i)}\, x_{m1}^{V_i}, \tag{5.30}$$

where

$$B_l\, x_l^{(0)} = g_l, \tag{5.31a}$$

$$B_l\, x_l^{(i)} = -s_l^{V_i} \qquad i = 1, \ldots, 5, \tag{5.31b}$$

and

$$x_u = x_u^{(0)} + \sum_{i=1}^{5} x_u^{(i)}\, x_{m2}^{V_i}, \tag{5.32}$$

where

$$B_u\, x_l^{(0)} = g_u \tag{5.33a}$$

$$B_u\, x_l^{(i)} = -r_u^{V_i} \qquad i = 1, \ldots, 5, \tag{5.33b}$$

where B_l and B_u are V-shaped matrices, and $s_l^{V_i}$ and $r_u^{V_i}$ represent the ith nonzero columns of matrices S_l and R_u, respectively. Distinct from the dodecahedral case, the topology of the icosahedron allows a point arrangement for which such nonzero columns are aligned. Replacing (5.30) into (5.29b) and rearranging gives

$$(T_{m1} + R_l\, X_l^{(1:5)}\, V_m)x_{m1} = g_{m1} - R_{m1}\, x_l^{(0)} - S_{m1}\, x_{m2}. \tag{5.34}$$

The matrix on the LHS is tridiagonal with additional nonzero terms in 6 columns. Rearranging to put the 6 full columns first, and performing forward and backward sweeps to eliminate the lower and upper subdiagonals of the lower-right block, results the matrix illustrated in Figure 5.10, where P is a full 6×6 matrix, H is full and $(5p-1) \times 6$, and T^{-} is a $(5p-1) \times (5p-1)$ diagonal matrix. The solution of (5.34) can be expressed as a function of x_{m2}, i.e.

$$x_{m1} = x_{m1}^{(0)} + \sum_{i=1}^{5} x_{m1}^{(i)}\, x_{m2}^{V_i} \tag{5.35}$$

where

$$(T_{m1} + R_l\, X_l^{(1:5)}\, V_m)\, x_{m1}^{(0)} = g_{m1} - R_{m1}\, x_l^{(0)} \tag{5.36a}$$

$$(T_{m1} + R_l\, X_l^{(1:5)}\, V_m)\, x_{m1}^{(i)} = -s_{m1}^{V_i} \qquad i = 1, \ldots, 5 \tag{5.36b}$$

Replacing (5.35) and (5.32) into (5.29d) gives

$$\left(T_{m2} + R_{m2}\, X_{m1}^{(1:5)}\, V_m + S_{m2}\, X_u^{(1:5)}\, V_m\right) x_{m2} = g_{m2} - R_{m2}\, x_{m1}^{(0)} - S_{m2}\, x_u^{(0)} \tag{5.37}$$

The matrix on the LHS of (5.37) exhibits the same sparsity pattern as that in (5.34), and thus the same approach may be followed for its solution. After x_{m2} is determined, substitution into (5.32) and (5.35) gives x_u and x_{m1}. Finally, x_m and x_l are obtained via (5.29c) and (5.30).

5.5 Conclusions

This chapter introduces several efficient extensions of the Thomas algorithm for the efficient solution of PDEs discretized with 3-point stencil operators over 1D connected domains. The m-Legged Thomas algorithm facilitates the efficient solution of PDEs over 1D domains which are connected at a single point, while retaining the same leading-order computational cost as the original Thomas algorithm. For more complicated closed 1D geometries, such as the edges of a 3D box, natural extensions of the circulant Thomas algorithm for the solution of the resulting system of equations have been identified. In each case, a careful enumeration of the gridpoints defined leads to a block tridiagonal matrix with exploitable structure. Incomplete forward and backward sweeps are used to significantly simplify this matrix, resulting in some modest fill-in. The resulting sparse systems can ultimately be solved directly in a straightforward fashion. In each case considered, the computational complexity of the resulting algorithm is $O(n)$; further, the prefactor is significantly smaller than that obtained by following a computational approach which simply rearranges the gridpoints to minimize the matrix bandwidth, as done when following a Cuthill-McKee type approach. For example, in the important case of the Box Thomas algorithm, a cost of $\sim 35n$ flops is required by our scheme, as compared with the $\sim 103n$ flops required by a Cuthill-McKee type approach; further, solution in only $\sim 18.\overline{3}n$ flops is possible when leveraging the modified system derived by the Box Thomas algorithm, to solve the system a second time with a new RHS vector.

Acknowledgements

This chapter contains work previously published in:

- D. Cavaglieri, T.R. Bewley, “Extensions of Thomas algorithm for the efficient solution of PDEs on wireframe structures", *Submitted to Journal of Computational Physics*, 2016

Algorithm 5.4 Box Thomas

```
function BoxThomas($a^l, b^l, c^l, s^l, r^m, d^m, e^m, f^m, s^m, r^u, a^u, b^u, c^u, g^l, g^m, g^u, m, p$)
    for i = 2 : p do
        for j = 1 : 4 do
            $e^m_{i-1,j} \leftarrow -e^m_{i-1,j}/d^m_{i-1,j}$
            $d^m_{i,j} \leftarrow d^m_{i,j} + e^m_{i-1,j}\, f^m_{i-1,j}$
            $r^m_{i,j} = e^m_{i-1,j}\, r^m_{i-1,j}$
            $g^m_{4(i-1)+j} \leftarrow g^m_{4(i-1)+j} + e^m_{i-1,j}\, g^m_{4(i-2)+j}$
    for j = 1 : 4 do
        $e^m_{p,j} \leftarrow -e^m_{p,j}/d^m_{p,j}$
        $b^u_{V_j} \leftarrow b^u_{V j} + e^m_{p,j}\, s^m_{p,j}$
        $r^u_{V_j,j} = e^m_{p,j}\, r^m_{p,j}$
        $g^u_{V_j} \leftarrow g^u_{V_j} + e^m_{p,j}\, g^m_{4(p-1)+j}$
    for i = p-1 : -1 : 1 do
        for j = 4 : -1 : 1 do
            $f^m_{i+1,j} \leftarrow -f^m_{i+1,j}/d^m_{i+1,j}$
            $r^m_{i,j} \leftarrow r^m_{i,j} + f^m_{i+1,j}\, r^m_{i+1,j}$
            $s^m_{i,j} = f^m_{i+1,j}\, s^m_{i+1,j}$
            $g^m_{4(i-1)+j} \leftarrow g^m_{4(i-1)+j} + f^m_{i+1,j}\, g^m_{4i+j}$
    for j = 4 : -1 : 1 do
        $f^m_{1,j} \leftarrow -f^m_{1,j}/d^m_{1,j}$
        $b^l_{V_j} \leftarrow b^l_{V j} + f^m_{1,j}\, r^m_{1,j}$
        $s^l_{V_j,j} = f^m_{1,j}\, r^m_{1,j}$
        $g^l_{V_j} \leftarrow g^l_{V_j} + f^m_{1,j}\, g^m_j$
    CircThomas($a^l, b^l, c^l, [g^l, s^l], m$)
    for j = 1 : 4 do
        $g^u_{V_j} \leftarrow g^u_{V_j} - r^u_{V_j,j}\, g^l_{V_j}$
        $t = -r^u_{V_j,j}$
        for k = 1 : 4 do
            $r^u_{V_j,k} = t\, s^l_{V_j,k}$
    $d^u_1 = a^u_1$
    $r^u_m = c^u_m$
    for i = 2 : m do
        $a^u_i \leftarrow -a^u_i/b^u_{i-1}$
        $b^u_i \leftarrow b^u_i + a^u_i\, c^u_{i-1}$
        $d^u_i = a^u_i\, d^u_{i-1}$
        for j = 1 : 4 do
            $r^u_{i,j} \leftarrow r^u_{i,j} + a^u_i\, r^u_{i-1,j}$
        $g^u_i \leftarrow g^u_i + a^u_i\, g^u_{i-1}$
    for i = m-1 : -1 : 1 do
        $c^u_i \leftarrow -c^u_i/b^u_{i+1}$
        $d^u_i \leftarrow d^u_i + c^u_i\, d^u_{i+1}$
        for j = 1 : 4 do
            $r^u_{i,j} \leftarrow r^u_{i,j} + c^u_i\, r^u_{i+1,j}$
        $g^u_i \leftarrow g^u_i + c^u_i\, g^u_{i+1}$
    Gauss$\left(\left[r^u_{[V_{1:4},m],1:4}, d^u_{[V_{1:4},m]}\right] + \mathrm{diag}\left\{b^u_{[V_{1:4},m]}\right\},\ g^u_{[V_{1:4},m]}\right)$
    for i = 1 : m-1, $i \neq V_j, \forall j = 1, \ldots, 4$ do
        $g^u_i \leftarrow g^u_i - r^u_{i,1:4}\, g^u_{V_{1:4}} - d^u_i\, g^u_m)/b^u_i$
    for i = 1 : m do
        $g^l_i \leftarrow g^l_i - s^l_{i,1:4}\, g^u_{V_{1:4}}$
    for i = 1 : p do
        for j = 1 : 4 do
            $g^m_{4(i-1)+j} \leftarrow (g^m_{4(i-1)+j} - r^m_{i,j}\, g^l_{V_j} - s^m_{i,j}\, g^u_{V_j})/d^m_{i,j}$
```

Chapter 6

Short-term ensemble ocean wave forecasting

6.1 Introduction

The complex nature of ocean wave propagation, given by the superposition and nonlinear interaction of numerous waves of different wavelength, frequency, amplitude, and direction, and the suddenness with which large and dangerous ocean waves such as tsunamis and rogue waves sometimes appear, has motivated researchers to develop accurate analytic and numerical methods to better model and predict ocean wave dynamics.

The recent development of wave energy converters (WECs), which attempt to harness a small fraction the massive amount of energy present in ocean waves, has generated renewed interest in accurate short-term ocean wave forecasting. Existing WEC devices work by oscillating at a resonance frequency, tuned to match the peak frequency of the wave spectrum at the location of the WEC device [48]. The relatively broad bandwidth of the wave spectrum generally observed in real sea states renders this passive approach relatively inefficient. The introduction of a control strategy which optimizes the WEC device power take-off parameters on a wave-by-wave basis to maximize the extracted power would greatly improve WEC device performance, thus increasing the competitiveness of marine energy with respect to other more mature fields of renewable energy, such as wind and solar power systems. The noncausality of 2D models of the relationship between the wave elevation at the device location and the dynamic behavior of the device (see, e.g., [49] and [48]) makes the derivation of an optimal control law dependent on the future wavefield. At a minimum, knowledge of the incoming wavefield is needed over a time window of the order of 10-20 s into the future (see [50]). An even longer forecasting horizon is desired when the control inputs are optimized via a receding-horizon Model Predictive Control (MPC) approach, which requires knowledge of the incoming wavefield over the entire optimization horizon considered. It has been observed (see [51]) that optimization horizons of at least 2-3 dominant wave periods are needed to provide an accurate approximation of the optimal control strategy.

A few attempts have emerged in the recent literature to develop a reliable wave forecasting framework. Among the most noteworthy, in [52], deterministic sea-wave prediction (DSWP) has been used for short-term wave forecasting. In this setting, measurements in the proximity of the point of interest are leveraged to develop filters relating the future wavefield at the location of interest to the acquired measurements. The deterministic nature of such an approach does not incorporate stochastic information, and thus this approach is adversely affected by unmodeled events taking place in the region of interest, such as multiple swells, wave diffraction, and radiation. In [53], a variational approach was developed to perform wave forecasting of a one-dimensional JONSWAP spectrum through the assimilation of synthetic radar data. Preliminary results appeared to be promising, although no extension to the more challenging two-dimensional case has been formulated. Finally, in [54] auto-regressive (AR) models, neural networks (NN), and linear and extended Kalman filter were tested against actual experimental data; results showed that AR models outperformed the other approaches tested.

This chapter describes the implementation of an Ensemble Kalman Filter (EnKF) for ocean wave forecasting. The EnKF has been adopted broadly in the numerical weather prediction community over the last 20 years. The accurate low-rank approximation of the state covariance in the EnKF method, and the independent propagation each ensemble member, leveraging nonlinear dynamic equations, are two of the significant advantages that the EnKF approach has over other forecasting approaches for nonlinear multiscale phenomenon.

This chapter is organized as follows: Section 6.2 introduces the ocean wave model for the simulation of ocean wave dynamics and the propagation of the ensemble members in the EnKF formulation. Section 6.3 describes the measurement devices employed to collect ocean data: Doppler radar and wave monitoring buoys. Section 6.4 introduces the Ensemble Kalman Filter, and describes its numerical implementation for short-term ocean wave forecasting. Section 6.5 analyzes the performance of our EnKF implementation when different measurement devices are employed in a realistic sea states.

6.2 Ocean wave model

To model the propagation of ocean waves, we consider a rectangular 3D computational domain with a deformed top, where x and y are the directions parallel to the ocean surface and z is up, where $z = 0$ corresponds to the surface at rest. The bathymetry is considered constant at $z = -h$, and the wave elevation is denoted $\eta(x, y, t)$. We assume that the flow is incompressible and inviscid; denoting $\mathbf{u}(x, y, z, t)$ the velocity at position (x, y, z) and time t, it is possible to introduce a potential function $\phi(x, y, z, t)$ such that $\mathbf{u} = \nabla\phi$. The continuity equation for the incompressible, inviscid flow may thus be written

$$\nabla_T^2 \phi = 0, \tag{6.1}$$

where $\nabla_T = [\partial/\partial x \quad \partial/\partial y \quad \partial/\partial z]$ is the three-dimensional gradient. Continuity holds for the whole volume of fluid, given by $-h \leq z \leq \eta$ for $x \in [0, L_x]$ and $y \in [0, L_y]$, where L_x and L_y represent the spatial extent of the domain in the x and y directions. Periodic boundary conditions on the computational domain are ultimately assumed in the x and y directions[1], while at the interface between fluid and air (i.e. $z = \eta$), dynamic and kinematic boundary conditions are imposed, and a no-flux boundary condition is imposed at the sea bottom (i.e. $z = -h$). The entire model is synthesized as

$$\begin{cases} \nabla_T^2 \phi = 0, & -h \leq z \leq \eta \\ \dfrac{\partial \eta}{\partial t} + \nabla\phi \cdot \nabla\eta - \dfrac{\partial \phi}{\partial z} = 0, & z = \eta \\ \dfrac{\partial \phi}{\partial t} + \dfrac{1}{2}\nabla_T\phi \cdot \nabla_T\phi + g\eta = 0, & z = \eta \\ \dfrac{\partial \phi}{\partial z} = 0, & z = -h \\ \phi(0, y, z, t) = \phi(L_x, y, z, t), & \\ \eta(0, y, t) = \eta(L_x, y, t), & \\ \phi(x, 0, z, t) = \phi(x, L_y, z, t), & \\ \eta(x, 0, t) = \eta(x, L_y, t) & \end{cases} \tag{6.2}$$

[1] As is typical in large-scale pseudospectral simulations (see, e.g., [55]), the physically-relevant region of the simulation is considered as embedded within a non-physical "fringe" region, which allows period boundary conditions to be used.

where g is the gravitational constant and $\nabla = [\partial/\partial x \quad \partial/\partial y]$ is the gradient in the horizontal directions x and y only. Following [56], the dimensionality of the problem in (6.2) can be reduced to surface variables alone. To accomplish this, define the surface potential Φ and surface vertical velocity W as

$$\begin{aligned} \Phi &= \phi(x, y, z, t)|_{z=\eta}\,, \\ W &= \left.\frac{\partial \phi}{\partial z}\right|_{z=\eta}; \end{aligned} \tag{6.3}$$

using the chain rule for differentiation, (6.2) may be written

$$\begin{cases} \nabla^2\Phi + \dfrac{\partial W}{\partial z} = 0, & z = \eta, \\ \nabla_T^2\phi = 0, & -h \le z < \eta, \\ \dfrac{\partial \eta}{\partial t} + \nabla\Phi \cdot \nabla\eta - W(1 + \nabla\eta \cdot \nabla\eta) = 0, & z = \eta, \\ \dfrac{\partial \Phi}{\partial t} + \dfrac{1}{2}\nabla\Phi \cdot \nabla\Phi - \dfrac{1}{2}W^2(1 + \nabla\eta \cdot \nabla\eta) + g\eta = 0, & z = \eta, \\ \dfrac{\partial \phi}{\partial z} = 0, & z = -h, \\ \Phi(0, y, t) = \Phi(L_x, y, t), & \\ \eta(0, y, t) = \eta(L_x, y, t), & \\ \Phi(x, 0, t) = \Phi(x, L_y, t), & \\ \eta(x, 0, t) = \eta(x, L_y, t). & \end{cases} \tag{6.4}$$

To integrate (6.4) numerically, several approaches have been proposed. Among these, the most promising are those presented in [57] and [58]; we prefer the latter due to its improved consistency and numerical stability. Such methods propagate the surface equations only, and account for the other equations indirectly [effectively, by analytic solution of the Laplace equation (6.1)]. The closure problem arising from the introduction of the vertical velocity W is addressed by expanding Φ and W into Taylor series about $z = 0$:

$$\Phi = \phi + \eta w - \frac{\eta^2}{2}\nabla^2\phi - \frac{\eta^3}{6}\nabla^2 w + \frac{\eta^4}{24}\nabla^4\phi + O(\epsilon^5), \tag{6.5}$$

$$W = w - \eta\nabla^2\phi - \frac{\eta^2}{2}\nabla^2 w + \frac{\eta^3}{6}\nabla^4\phi + \frac{\eta^4}{24}\nabla^4 w + O(\epsilon^5), \tag{6.6}$$

where ϵ is the wave steepness, defined as $\epsilon = k\eta$ where k is the wavenumber, and $w = \partial\phi/\partial z|_{z=0}$. The Laplace equation arising from the continuity equation is used to replace $\partial^2\phi/\partial z^2$ with $-\nabla^2\phi$. The linear part (w) of the Taylor expansion of W in (6.6) can then be related to the linear part (ϕ) of the Taylor expansion of Φ in (6.5) via the analytic solution of the linear wave equations, i.e.

$$\begin{cases} \nabla_T^2\phi = 0, & -h \leq z \leq 0 \\ \dfrac{\partial\eta}{\partial t} - \dfrac{\partial\phi}{\partial z} = 0, & z = 0 \\ \dfrac{\partial\phi}{\partial t} + g\eta = 0, & z = 0 \\ \dfrac{\partial\phi}{\partial z} = 0, & z = -h \end{cases} \tag{6.7}$$

As shown in [59], the following expression is obtained:

$$w = \left.\frac{\partial\phi}{\partial z}\right|_{z=0} = \mathcal{F}^{-1}[k\tanh(kh)\mathcal{F}[\phi]] \triangleq -\mathcal{L}[\phi], \tag{6.8}$$

where $\mathcal{F}[\cdot]$ denotes the Fourier transform, and $\mathcal{F}^{-1}[\cdot]$ denotes its inverse.

After substituting (6.8) into (6.5) and (6.6) wherever w arises, inverting the resulting expression in (6.5) to express ϕ as a function of Φ, and finally substituting into (6.6) to eliminate ϕ, it is possible to rewrite W as a function of the surface potential Φ:

$$W = -\mathcal{L}[\Phi] - \eta\nabla^2\Phi - \mathcal{L}[\eta\mathcal{L}[\Phi]] + \frac{1}{2}\eta^2\nabla^2\mathcal{L}[\Phi] - \eta\nabla^2(\eta\mathcal{L}[\Phi]) - \mathcal{L}\left[\frac{1}{2}\eta^2\nabla^2\Phi + \eta\mathcal{L}[\eta\mathcal{L}[\Phi]]\right] + O(\epsilon^3) \tag{6.9}$$

By substituting (6.9) into the dynamic and kinematic boundary conditions in (6.4), and retaining the terms until third order, we get:

$$\frac{\partial\eta}{\partial t} + \mathcal{L}[\Phi] + \nabla\cdot(\eta\nabla\Phi) + \mathcal{L}[\eta\mathcal{L}[\Phi]] + \nabla^2\left(\frac{1}{2}\eta^2\mathcal{L}[\Phi]\right) + \mathcal{L}\left[\eta\mathcal{L}[\eta\mathcal{L}[\Phi]] + \frac{1}{2}\eta^2\nabla^2\Phi\right] = 0 \tag{6.10a}$$

$$\frac{\partial\Phi}{\partial t} + g\eta + \frac{1}{2}\nabla\Phi\cdot\nabla\Phi - \frac{1}{2}\mathcal{L}[\Phi]\mathcal{L}[\Phi] - \mathcal{L}[\Phi]\left(\eta\nabla^2\Phi + \mathcal{L}[\eta\mathcal{L}[\Phi]]\right) = 0. \tag{6.10b}$$

In this way, continuity of the flowfield itself, as well as the no-flux boundary condition at the sea bottom, are accounted for implicitly in the representation. Note also that periodic boundary conditions are incorporated via a pseudo-spectral discretization of the horizontal derivatives, which simplifies significantly the computation of the linear operator $\mathcal{L}[\cdot]$ in (6.8).

6.3 Wave measurement devices

Two common measurement devices used for wave monitoring are considered for data assimilation. The first is a Doppler radar, which measures the radial component of the wave velocity u_r with respect to the radar center, within the radar range $R_{\max}$. The wave radar operates in low-grazing-angle mode. The radar antenna spans 360 degrees every 2 seconds, and provides a surface elevation image with an azimuthal resolution of 1 degree and radial resolution of $5\,m$. With this device, the surface potential Φ at a distance r from the center of the radar can be obtain through integration:

$$\Phi_R(r,\theta,t) = \int_0^r u_r(\rho,\theta,t)d\rho \quad \text{for} \quad r \in [0, R_{\max}]. \tag{6.11}$$

We thus assume that the surface potential Φ is measurable within the radar range. The degradation of the radar signal with the distance from the source/receiver is modeled by adding a distance-dependent white noise to the measurements. In order to mimic the dependence of the signal-to-noise ratio of a radar signal with the fourth power of the distance r from the source/receiver (see [60]), the artificial noise is modeled as Gaussian random noise with zero mean and a distance-dependent covariance of

$$\sigma_R^2(r) = \alpha_R + \beta_R \left(\frac{r}{R_{\max}}\right)^4, \tag{6.12}$$

where α_R represents the sum of the background noise and the rms of the error obtained in the evaluation of the integral in (6.11), while β_R accounts for the signal degradation as the distance from the source increases.

Another common type of measurement device is wave monitoring buoys, which may be organized into arrays. These devices have already been employed in practice to measure the local wave elevation [54]. In this case, the wave elevation η is measured at each buoy location every 2 seconds. Measurement uncertainty is modeled by adding Gaussian random noise with constant covariance σ_B^2 to each measurement.

6.4 The Ensemble Kalman Filter

The Ensemble Kalman Filter (EnKF) is a powerful data assimilation method which has been adopted broadly by the weather forecasting community in the years since it was introduced by Evensen [61]. In [62], the EnKF has been applied to ocean wave forecasting assuming linear wave propagation; this chapter extends this analysis to nonlinear wave propagation. The implementation proceeds as follows. Initially, a specified number of ensemble members N is generated by randomly sampling a sea spectrum considered to be representative of the actual sea state. Then, a physical model of the wave process is employed to advance each member independently over time. Whenever new measurements of the wavefield become available, the mean and second-order statistics of the ensemble distribution are calculated, and a Kalman-like data assimilation step is performed. The updated ensemble members are then propagated in time until new measurements become available, and the process repeated.

The dynamic model used to propagate the "truth" and ensemble wavefields is given by (6.10). The wave elevation $\eta_0(x, y)$ and flow potential $\Phi_0(x, y)$ are initialized following a JONSWAP distribution, as this semi-empirical model has been shown to provide a reasonably accurate approximation of the frequency spectrum of wind-generated waves in deep water [63]. Following [64], the JONSWAP spectrum is given by

$$\begin{gathered}
S(\omega) = 155\,\frac{H_{1/3}^2}{T_p^4\omega^5}\,e^{\frac{-944}{T_p^4\,\omega^4}}\,(3.3)^Y, \quad \text{with} \\
Y = e^{\frac{-(0.191\,\omega T_p - 1)^2}{2\sigma^2}}, \quad \text{and} \quad \sigma = \begin{cases} 0.07, & \omega \leq 5.24/T_p, \\ 0.09, & \omega > 5.24/T_p, \end{cases}
\end{gathered} \tag{6.13}$$

where $H_{1/3}$ is the significant wave height, and T_p the dominant wave period. To account for angular spreading of the waves, an artificial directionality function $f(\theta)$ is defined such that

$$f(\theta) = \begin{cases} \frac{2}{\pi}\cos^2(\theta - \theta_0), & \text{for } \theta \in [\theta_0 - \frac{\pi}{2},\, \theta_0 + \frac{\pi}{2}] \\ \\ 0, & \text{elsewhere,} \end{cases} \tag{6.14}$$

where θ_0 is the primary direction of wave propagation. The initial wavefield is then obtained by randomly selecting $N_w = N_\omega N_\theta$ components, where N_θ is the number of individual directions θ_j modeled, and N_ω is the number of frequency components ω_i modeled per direction, for the directional JONSWAP distribution defined as

$$S(\omega,\, \theta) = S(\omega)\, f(\theta). \tag{6.15}$$

The initial wave elevation $\eta_0(x,\, y)$ is then defined as

$$\eta_0(x,\, y) = \sum_{i=1}^{N_\omega} \sum_{j=1}^{N_\theta} \sqrt{2S(\omega_i)\Delta\omega\Delta\theta}\, \cos(k_i \cos\theta_j\, x + k_i \sin\theta_j\, y + \varepsilon_{ij}), \tag{6.16}$$

where $\Delta\omega$ is the frequency resolution of the spectrum, $\Delta\theta = \pi/(N_\theta - 1)$ is the directional angular resolution, $\varepsilon_{ij} \sim \mathcal{U}(0,\, 2\pi)$ is a uniformly random phase shift, and k_i is the wavenumber associated with each selected frequency component ω_i through the finite-depth dispersion relationship

$$\omega_i = \sqrt{g k_i \tanh(k_i h)}. \tag{6.17}$$

The initial flow potential $\Phi_0(x,\, y)$ is then obtained via solution of the linear wave propagation problem for the initial wave elevation given by (6.16):

$$\Phi_0(x,\, y) = \sum_{i=1}^{N_\omega} \sum_{j=1}^{N_\theta} \frac{g}{\omega_i} \sqrt{2S(\omega_i)\Delta\omega\Delta\theta}\, \sin(k_i \cos\theta_j\, x + k_i \sin\theta_j\, y + \varepsilon_{ij}). \tag{6.18}$$

Note that, in certain sea conditions, this semi-empirical model has been proven to be inaccurate in capturing the actual sea spectrum. In such cases, the use of a more sophisticated spectral propagation model, like WAVEWATCH-III [65], is recommended. Ensemble members can thus be generated according to the spectral distribution provided by this model at a given time, and regenerated accordingly if/when the wave spectrum changes. It is noted that the adoption of the JONSWAP model in the present analysis generates a relatively narrow bandwidth of the wave spectrum. A significant degradation of the forecasting performance of our method is expected for wavefields characterized by a broader spectrum, as observed in [54].

Denoting with X the matrix containing the entire ensemble of state variables, with one ensemble representation in each of its N columns, data assimilation is performed at each measurement time via the EnKF update equations, which may be written

$$V^- = \mu \left[X^- - E\left(X^- \right) \right], \tag{6.19a}$$

$$P^- = V^-(V^-)^T/(N-1), \tag{6.19b}$$

$$X^+ = X^- + P^- C^T (C P^- C^T + R)^{-1} (Y - C X^-), \tag{6.19c}$$

where:

- superscript $-$ denotes the *prior* representation,
- superscript $+$ denotes the *posterior* representation,
- C is the matrix relating the measurements y to the state vector $x = [\eta^T \quad \Phi^T]^T$,
- R is the measurement noise covariance matrix (defined according to device specifics–see §6.3),

- Y is the matrix obtained by perturbing (N times, in each of its N columns) the vector of measurements y with random noise, constructed consistent with R,
- $E\left(X^{-}\right)$ is the mean of the (prior) ensemble set, and
- P^{-} is the (low-rank) ensemble approximation of the (prior) covariance matrix.

Note specifically that, to implement (6.19) in a numerically tractable fashion when the state dimension n is large, the $n \times n$ matrix P^{-} is never explicitly computed. Rather, (6.19b) is kept in its factored form and substituted into (6.19c), and the $n \times N$ factor V^{-}, which itself defines P^{-}, is everywhere multiplied by C, as required by (6.19c), before being used further.

Distinct from the standard EnKF formulation, a fading-memory parameter $\mu \geq 1$ has been introduced, as suggested by [66, 67] (the standard EnKF formulation is retrieved for $\mu = 1$). This fading-memory formulation increases the response of the EnKF to new measurements. A resampling strategy is also implemented, for which a small percentage of the ensemble members are regenerated after a given time interval, as suggested by [68]. This resampling (a.k.a. covariance inflation) strategy also increases the responsiveness of the EnKF to new measurements.

The outer region of each ensemble member (away from the measurements, which are generally centered around or slightly upstream of the WEC device itself) is considered as a nonphysical “fringe” or “sponge” region in the ensemble representation of the actual flowfield. An artificial forcing function, akin to that adopted in [55], is applied specifically to randomly “scramble”, somewhat gently, the relative phase of the waves in the various ensemble members in the ensemble representation of the wavefield in this region. This *Phase Scrambling* of the waves in the ensemble representation in this fringe zone has the effect of *increasing* the modelled variance of the ensemble representation (that is, increasing the corresponding components of P^{-}) in this spatial region, specifically for those wave components that are present in the particular sea state under consideration. Larger (compared to R) variance in this region and in these components tends, by (6.19c), to lead to larger measurement updates as the waves convect again into regions where measurements become available (that is, in the vicinity of the WEC device). Note specifically that the artificial forcing function used in the fringe region is *not* designed simply to diminish the waves in the ensemble members towards zero in the fringe region, which would have the undesired effect of *decreasing* the modelled variance of the ensemble representation in this region for the wave components in question, thereby causing the undesired effect of *decreasing* the measurement updates as the waves convect back into the regions where measurements become available.

6.5 Simulations

Some initial numerical results are now presented to assess the performance of the EnKF wave forecasting strategy proposed above under different measurement configurations and sea states.

In our numerical tests, the computational domain used for simulation of the "truth" wavefield is a Cartesian rectangular domain $[0, L_x] \times [0, L_y]$, with $L_x = 8000\,\mathrm{m}$ and $L_y = 4000\,\mathrm{m}$, and $n_x = 512$ and $n_y = 256$ gridpoints are used in the x and y directions, respectively. A constant depth of $h = 100\,\mathrm{m}$ is assumed. For the simulation of the ensemble members, a rectangular domain of size $[0, L_x^{KF}] \times [0, L_y^{KF}]$ is used, with $L_x^{KF} = 6000\,\mathrm{m}$ and $L_y^{KF} = 4000\,\mathrm{m}$, and $n_x^{KF} = 384$ and $n_y^{KF} = 256$ gridpoints are used in the x and y directions, respectively. For convenience, the two grids described above coincide in the y direction, and coincide over 384 gridpoints in the x direction. The FFTW package [69] is used to implement optimized FFT routines for the wavefield simulations, assuming periodic boundary conditions in x and y on the domains considered, as described further below.

The main direction of propagation of the waves is taken as parallel to the x direction; that is, $\theta_0 = 0$ in (6.14). Significantly, the domain size of the "truth" simulation, and the domain size of the EnKF-based reconstruction of this wavefield, are taken as large and different in the x direction. The results given later in this section show that the EnKF-based reconstruction in the region of interest (near the WEC device) is still quite effective. That is, the wavefield of the truth simulation is reconstructed well by the EnKF representation in the region of interest, even though the domain size in the x direction in the truth and EnKF models do not match. This indicates that the (artificial) periodic boundary conditions used in both sets of simulations are not key factors in the accuracy of the reconstruction, and that a spatially-periodic EnKF-based reconstruction of a nonperiodic wavefield is expected to show similar behavior in terms of the accuracy of the reconstructed wavefield in the region of interest.

Noting (6.16), the initial wavefield $\eta_0(x,\ y)$ for the sea state is obtained by randomly sampling $N_\omega \cdot N_\theta = 50$ wave components from the JONSWAP spectrum in (7.36) with $H_{1/3} = 3\,m$, and $T_p = 10\,s$ (see Figure 6.1). This produces a wavefield with a dominant wavelength of 150 m and an associated phase speed of $16\,m/s$.

Simulations have been performed in which the number of ensemble wavefields, N, is equal to 125 or 250. Each wavefield of the ensemble set is generated by randomly sampling a perturbation of the JONSWAP spectrum used to generate the truth model. This is achieved by considering the significant wave height $H_{1/3}$ as a uniformly random variable within the interval $[2.5\,m,\ 3.5\,m]$, the dominant wave period within the range $[8\,s,\ 12\,s]$, and the main direction of wave propagation within $[-\pi/20,\ \pi/20]$. Furthermore, every 40

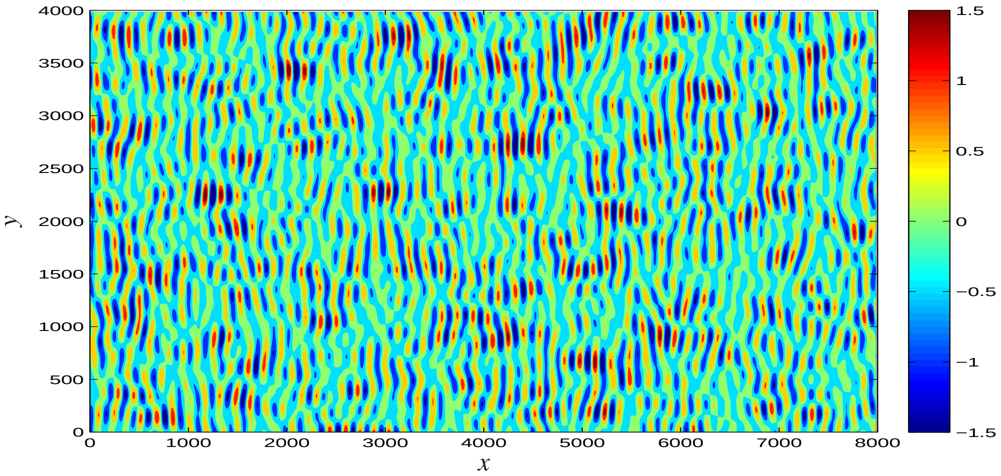


Figure 6.1: Snapshot of the initial wavefield $\eta_0(x, y)$.

seconds, approximately 10% of the ensemble members are regenerated from a perturbed JONSWAP spectrum. The fading-memory parameter μ in (6.19) is taken as 1.1. The simulation is run for $T = 100\,s$, with a constant timestep $\Delta t = 0.5\,s$. The third-order low-storage mixed implicit/explicit Runge-Kutta algorithm **IMEXRKCB3e** in [26] and Chapter 2 is used for the time integration of (6.10), with the linear terms treated implicitly, and the nonlinear treated explicitly.

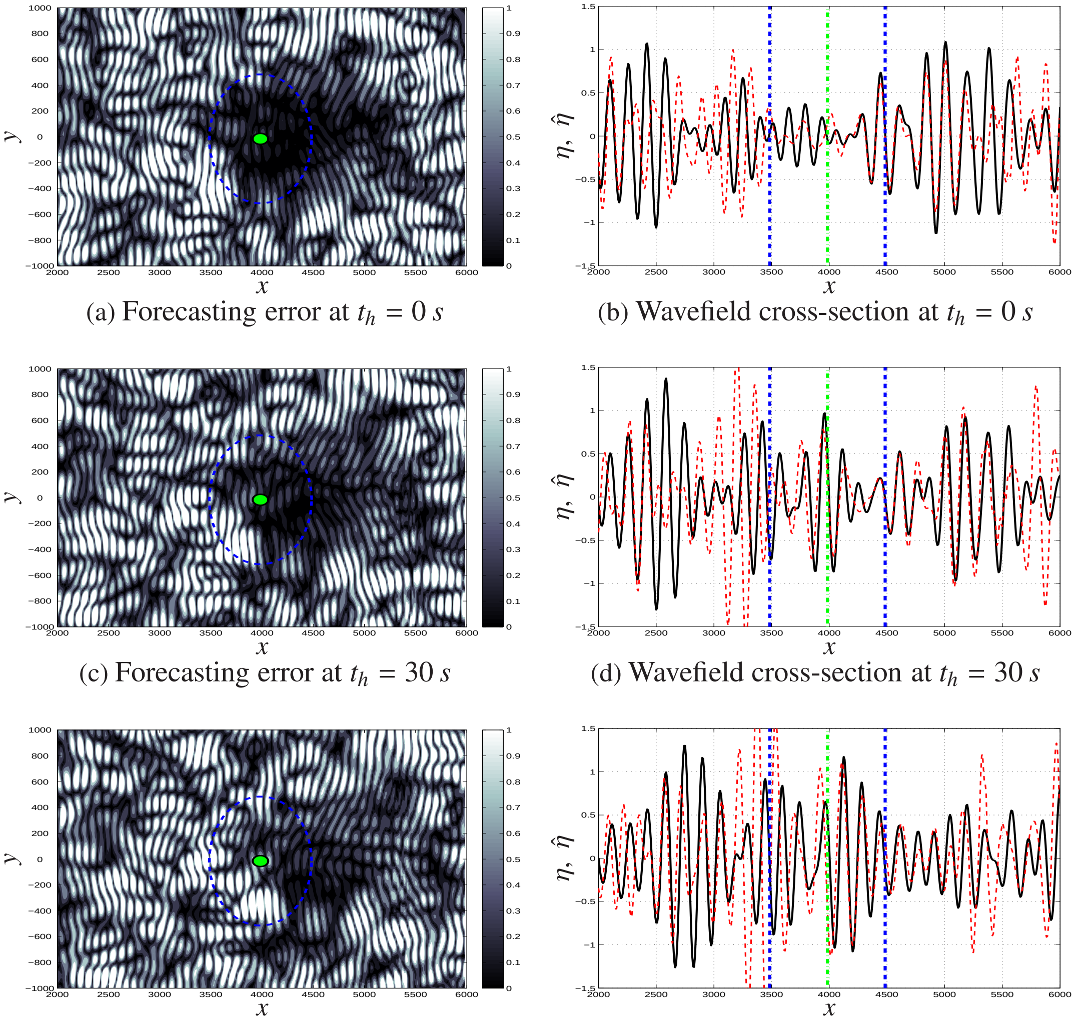
As far as the measurement strategy is concerned, four different scenarios have been considered, two involving a Doppler radar and two involving arrays of measurement buoys, with different configurations, as described below. In all cases, a sampling interval of 2 s is considered. After the simulation has reached time T, the ensemble wavefield is advanced in time over a forecasting horizon T_h of one minute, and the associated ensemble forecast is compared to the actual propagated wavefield. Particular attention is placed on wave estimation at a single point, referred to as *point of interest*, at the center of the computational domain, i.e., at $(x, y) = (L_x/2, L_y/2)$. In the WEC application, this would represent the location of one or more WEC devices, as motivated in §7.1.

6.5.1 Doppler radar

The first case considers the estimator performance in a configuration in which a Doppler radar is colocated with the point of interest, such as a ship using a wave radar to predict oncoming rogue waves. The second case considers a non-colocated Doppler radar configuration, with the radar placed 1000 m upstream of the point of interest, such as for WEC device tuning, or for a monitoring station outside of a harbor or other fixed cargo transfer location. In both configurations, a radar range of 500 m is assumed, and the parameters used for the definition of the measurement noise covariance in (6.12) are $\alpha_R = 10^{-3}$ and $\beta_R = 10^{-3}$.

Results are shown in Figures 6.2-6.5. The case with colocated radar and $N = 125$ shows that wavefield reconstruction (Figures 6.2a-f) is performed with poor accuracy within the radar range. Results improve considerably when $N = 250$ ensemble members are employed. In this case, the wave reconstruction (see Figures 6.3a-b) creates a region of low error which extends outside the radar range approximately 1000 m downstream. This refined estimation of the incoming wavefield significantly improves the thirty-second-ahead forecast (Figures 6.3c-d), as well as the one-minute-ahead forecast (Figures 6.3e-f), although this appears to be the maximum forecasting horizon possible in this configuration, as a high-error region appears upstream close to the point of interest.

The non-colocated case is of particular interest, since in this framework the downstream region of low-error coincides with the point of interest. When 125 ensemble members are employed, wavefield reconstruction is again rather poor (Figures 6.4a-b), with the low-error region barely covering the point of interest. The thirty-second-ahead forecast (Figures 6.4c-d) shows a rapid contraction of the low-error region, as observed in the colocated case, due to the side regions of high error spreading toward the center of the domain.



(a) Forecasting error at $t_h = 0\ s$

(b) Wavefield cross-section at $t_h = 0\ s$

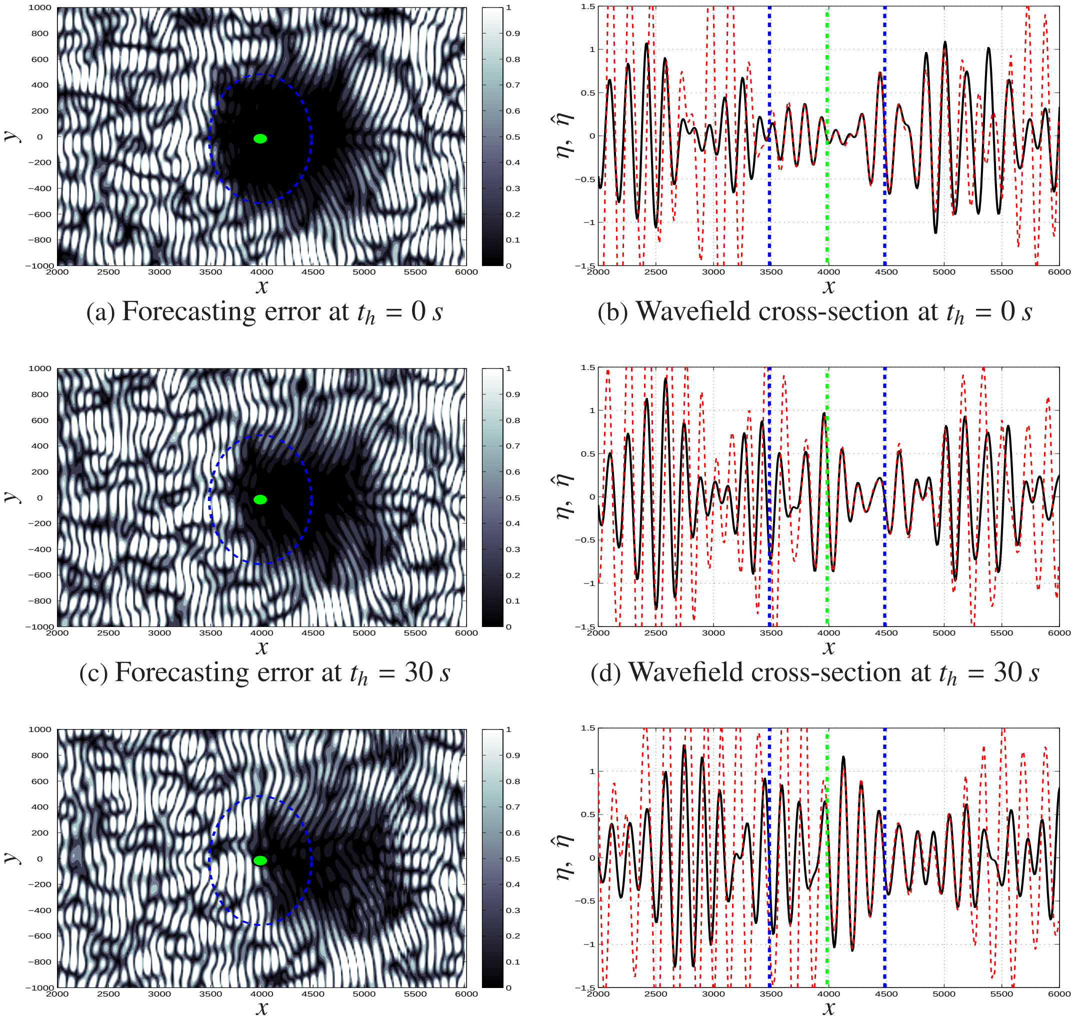
(c) Forecasting error at $t_h = 30\ s$

(d) Wavefield cross-section at $t_h = 30\ s$

(e) Forecasting error at $t_h = 60\ s$

(f) Wavefield cross-section at $t_h = 60\ s$

Figure 6.2: Current estimate, 30-second prediction, and 1-minute prediction using 125 ensemble members and a colocated radar. In the left figures, the green point represents the point of interest, and the blue dashed circle represents the radar range. In the right figures, the black solid line represents the actual wave height, the red dashed line represents the reconstructed wave height, the two blue vertical lines indicate the radar range, and the green vertical line indicates the position of the point of interest.



(a) Forecasting error at $t_h = 0\ s$

(b) Wavefield cross-section at $t_h = 0\ s$

(c) Forecasting error at $t_h = 30\ s$

(d) Wavefield cross-section at $t_h = 30\ s$

(e) Forecasting error at $t_h = 60\ s$

(f) Wavefield cross-section at $t_h = 60\ s$

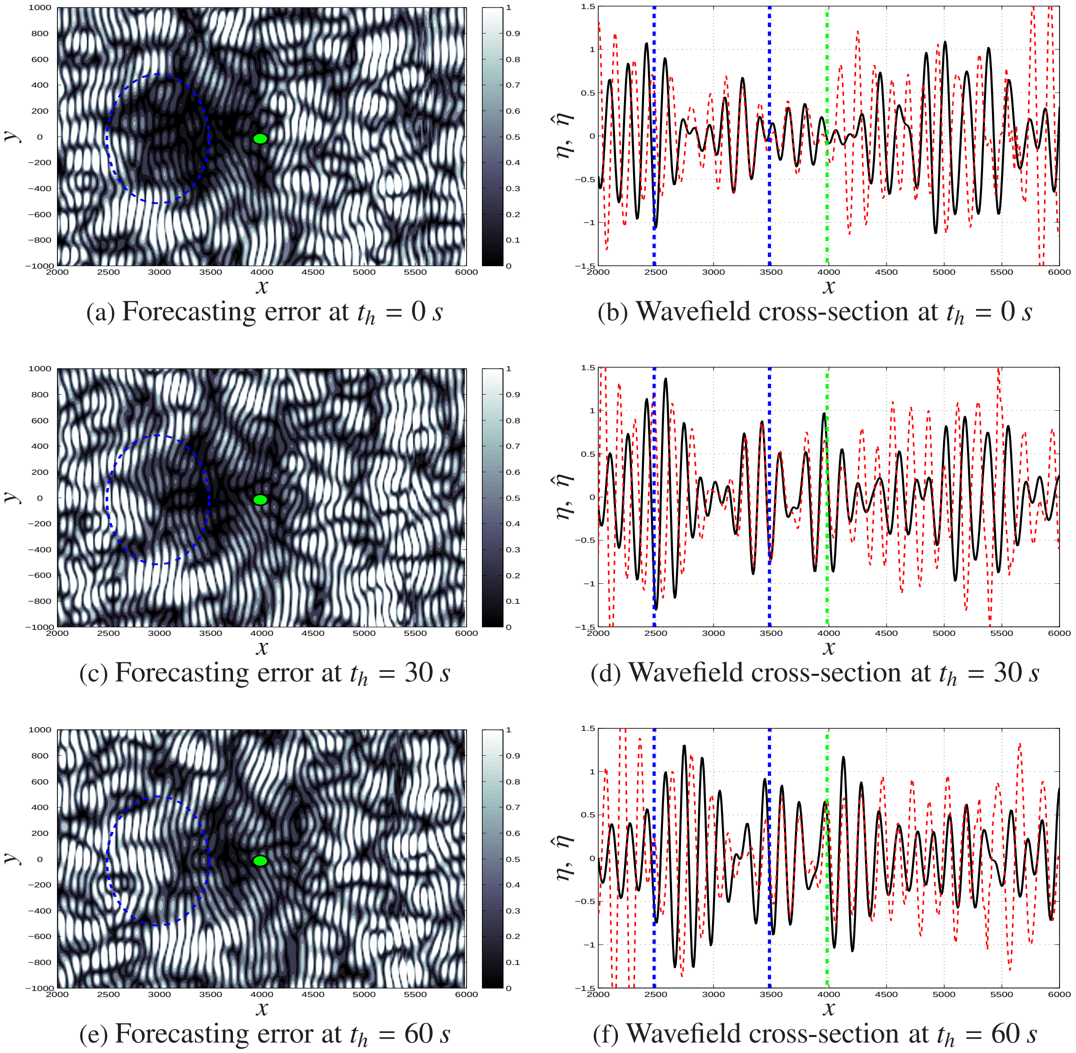
Figure 6.3: Current estimate, 30-second prediction, and 1-minute prediction using 250 ensemble members and a colocated radar. Symbols marked as in Figure 6.2.

This phenomenon is even more evident in the one-minute ahead forecast (Figures 6.4e-f), in which the low-error region essentially disappears altogether.

Again, results improve significantly when 250 ensemble members are employed. In this case, estimation results (Figures 6.5a-b) show that a low-error region is created which extends outside the radar range 1000 m downstream, thus including the point of interest. As time advances, the low-error region convects toward the point of interest, while moderately shrinking; the error of the wave forecast after thirty seconds and one minute (Figures 6.5c-d and 6.5e-f, respectively) at the point of interest are comparatively quite low.

Comparative analysis of the four configurations considered above (Figure 6.6) reveals interesting trends. The case with colocated radar and 125 ensemble members (Figure 6.6a) fails to correctly reconstruct the actual wavefield accurately at the point of interest. Also, the accuracy of the estimate at the point of interest degrades rapidly, due mainly to the low-error region convecting downstream. Results improve with a higher number of ensemble members (Figure 6.6b). In this case, the estimation error at the point of interest remains below 2% for the first 30 seconds, and increases to 8% by 60 seconds. This improvement is mainly explained by the increased extension of the low-error region upstream of the radar range.

Even better results are achieved in the non-colocated configuration: with 125 ensemble members (Figure 6.6c), the relative error is around 12% (on average) for the entire forecast horizon, with substantial magnitude error but minimal phase error for the first 30 seconds. Leveraging 250 ensemble members (Figure 6.6d), the relative error is kept below 5% for the first 25 seconds with virtually zero phase lag. After that time, some magnitude errors are evident, but the phase error is minimal. This result suggests that this configuration (that is, 250 ensemble members and the wave radar situated upstream of the point of



(a) Forecasting error at $t_h = 0\ s$

(b) Wavefield cross-section at $t_h = 0\ s$

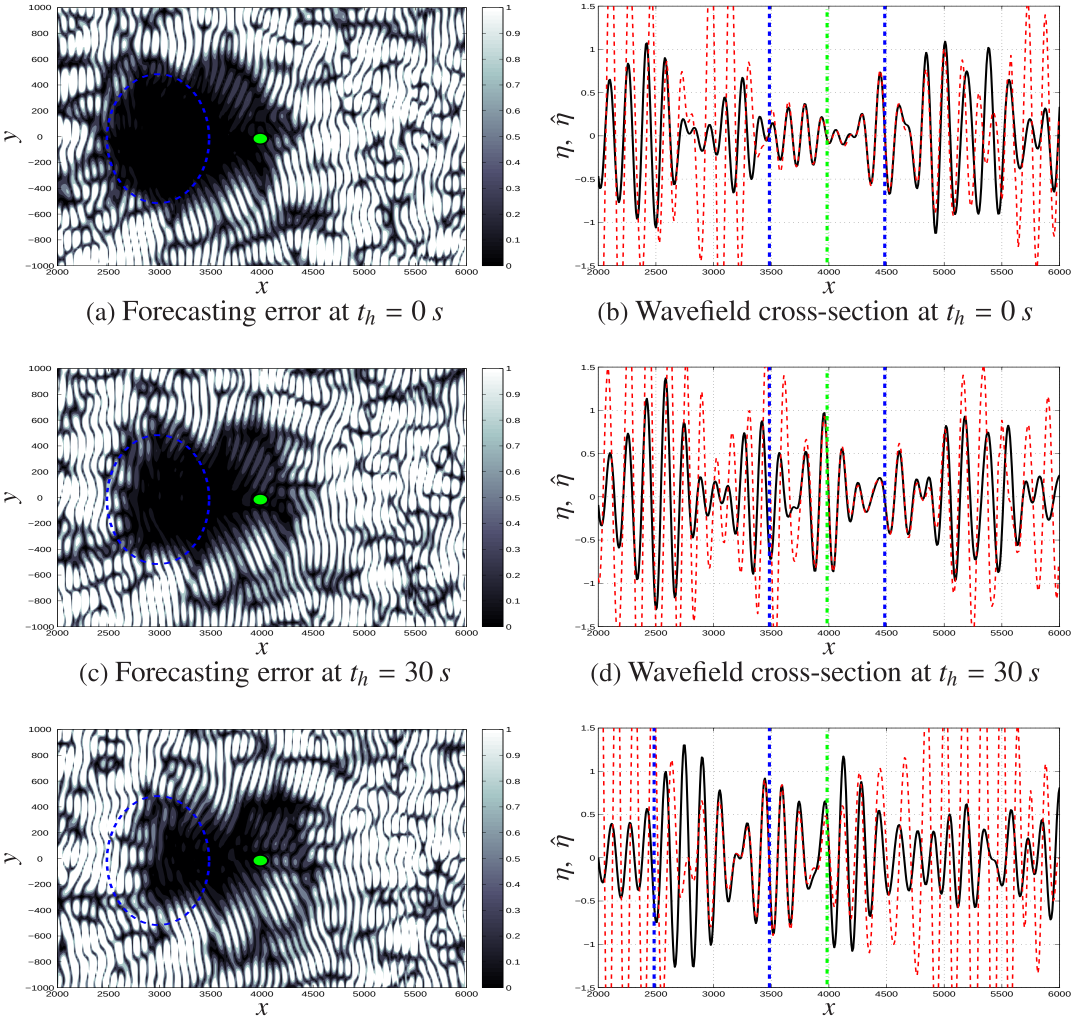
(c) Forecasting error at $t_h = 30\ s$

(d) Wavefield cross-section at $t_h = 30\ s$

(e) Forecasting error at $t_h = 60\ s$

(f) Wavefield cross-section at $t_h = 60\ s$

Figure 6.4: Current estimate, 30-second prediction, and 1-minute prediction using 125 ensemble members and a non-colocated radar. Symbols marked as in Figure 6.2.



(a) Forecasting error at $t_h = 0\ s$

(b) Wavefield cross-section at $t_h = 0\ s$

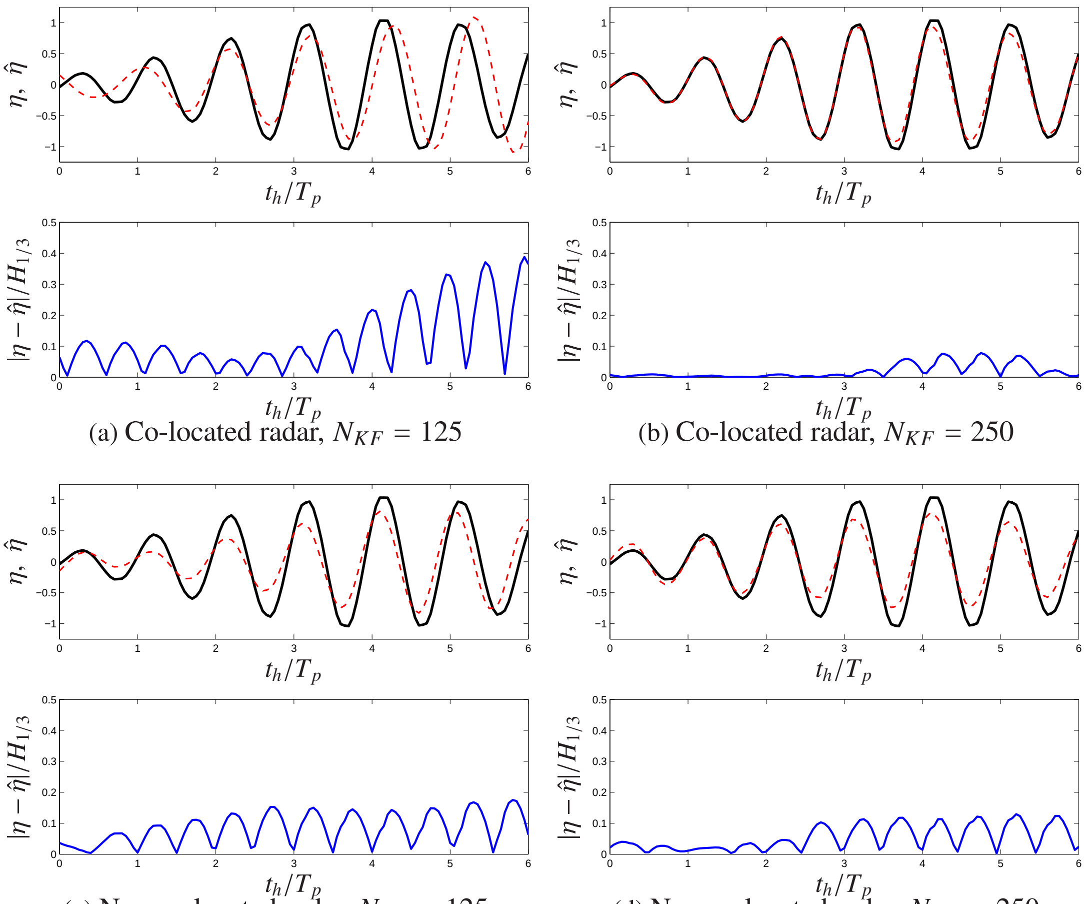
(c) Forecasting error at $t_h = 30\ s$

(d) Wavefield cross-section at $t_h = 30\ s$

(e) Forecasting error at $t_h = 60\ s$

(f) Wavefield cross-section at $t_h = 60\ s$

Figure 6.5: Current estimate, 30-second prediction, and 1-minute prediction using 250 ensemble members and a non-colocated radar. Symbols marked as in Figure 6.2.



(a) Co-located radar, $N_{KF} = 125$

(b) Co-located radar, $N_{KF} = 250$

(c) Non co-located radar, $N_{KF} = 125$

(d) Non co-located radar, $N_{KF} = 250$

Figure 6.6: Zero- to 60-second-ahead wave height (black), predicted wave height (red) and prediction error magnitude (blue, normalized by the significant wave height $H_{1/3}$), at the point of interest, as a function of time (normalized by the dominant wave period $T_p = 60$ sec), estimated using wave radar configured as indicated.

interest) is preferred for longer forecasting horizons, such as for WEC device tuning.

6.5.2 Measurement buoys

The first case considers a single array of three buoys placed 250 m directly upstream of the point of interest, with a lateral separation of 100 m, while the second case considers two arrays of three equally-spaced buoys each, one 250 m and the other 1000 m directly upstream of the point of interest, again with a lateral spacing of 100 m. The measurement noise has been modeled as a Gaussian random variable with zero mean and constant covariance $\sigma_B^2 = 10^{-3}$.

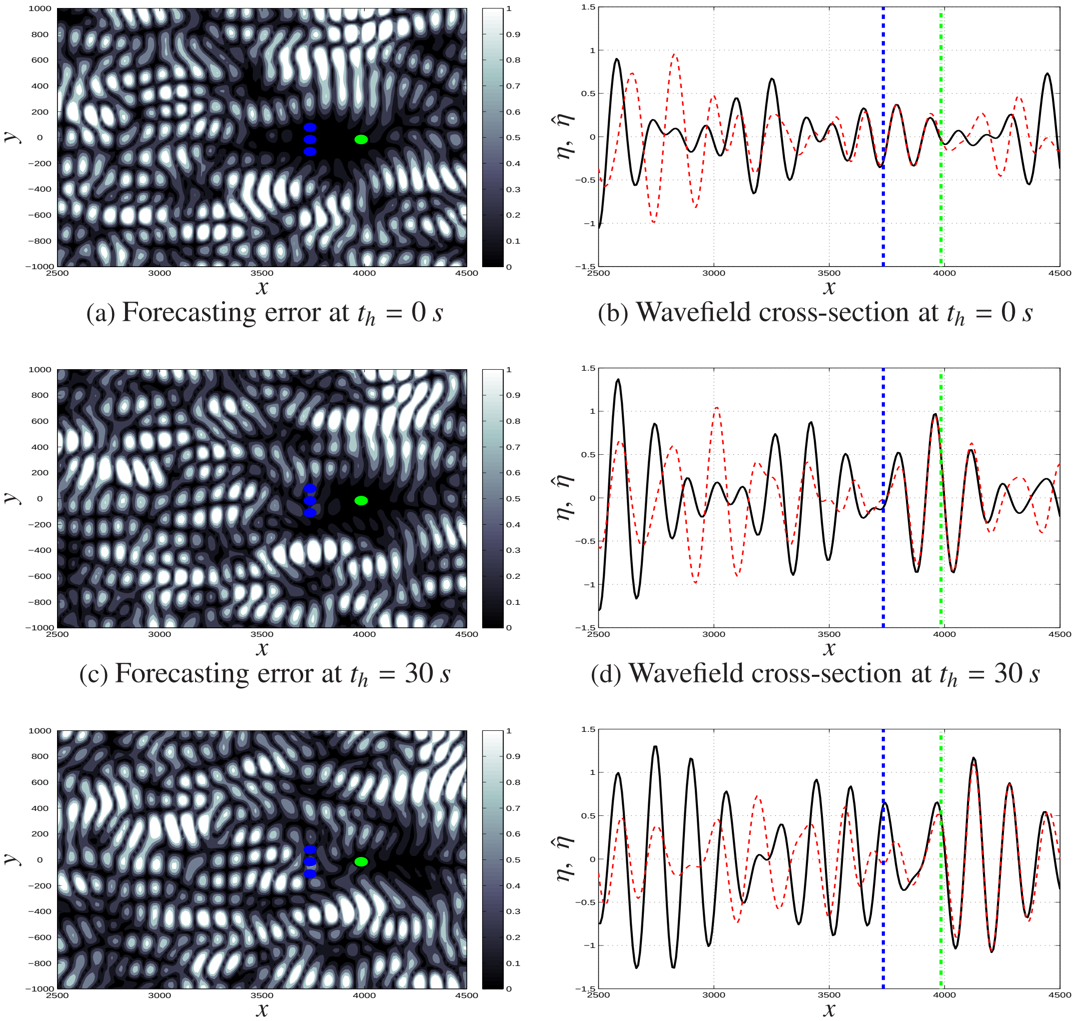
Results are shown in Figures 6.7-6.10. The case with a single array of buoys and 125 ensemble members shows that a region of low estimation error develops in the neighborhood of the buoy array, and extends nearly 500 m upstream and downstream, allowing good estimation error at the point of interest (Figures 6.7a-b). Thirty-second- and one-minute-ahead prediction (Figures 6.7c-d and Figures 6.7e-f) still show fairly good overall results, with the low-error region shrinking while being convected downstream.

Distinct from what observed in the wave radar setting, increasing the number of ensemble members for the single buoy array does not seem to improve performance significantly (see Figures 6.8a-f). It is also observed that the wavefield reconstruction error with 250 ensemble members appears slightly lower at the sides of the buoy array than the case with 125 ensemble members.

The inclusion of a second row of measurement buoys significantly extends the region of low estimation error, as shown in Figures 6.9a-b, where 125 ensemble members have been used. The thirty-second-ahead forecast (Figures 6.9c-d) shows the low-error region shrinking significantly, while maintaining good amplitude and phase estimation at the point of interest. The low-error region effectively vanishes by the sixty-second-ahead forecast (Figures 6.9e-f).

Incorporating 250 ensemble members only slightly increases the initial region of low estimation error, especially at the sides of the two buoy arrays, as shown in Figures 6.10a-b. After thirty seconds, this region shrinks slightly (Figures 6.10c-d), while after one minute it is significantly reduced, as observed in Figures 6.10e-f.

A comparative analysis of the four cases described above is reported in Figure 6.11. The case with a single array and 125 ensemble members (Figure 6.11a) shows a relative



(a) Forecasting error at $t_h = 0\ s$

(b) Wavefield cross-section at $t_h = 0\ s$

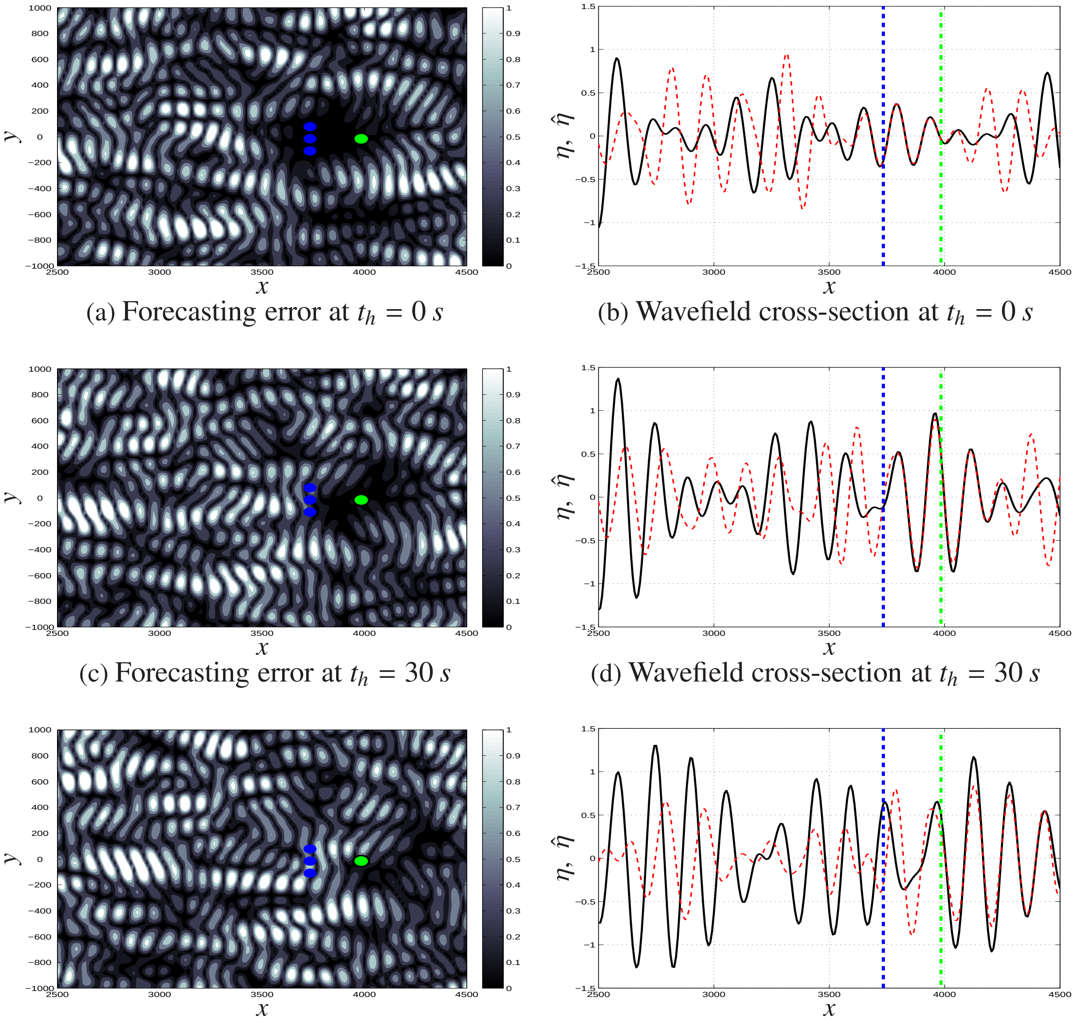
(c) Forecasting error at $t_h = 30\ s$

(d) Wavefield cross-section at $t_h = 30\ s$

(e) Forecasting error at $t_h = 60\ s$

(f) Wavefield cross-section at $t_h = 60\ s$

Figure 6.7: Current estimate, 30-second prediction, and 1-minute prediction using 125 ensemble members and a single array of three buoys placed 250 m upstream of the point of interest. In the left figures, the green point represents the point of interest, and the blue circles represent the measurement buoys. In the right figures, the black solid line represents the actual wave height, the red dashed line represents the reconstructed wave height, the blue vertical line indicates the buoy location, and the green vertical line indicates the position of the point of interest.



(a) Forecasting error at $t_h = 0\ s$

(b) Wavefield cross-section at $t_h = 0\ s$

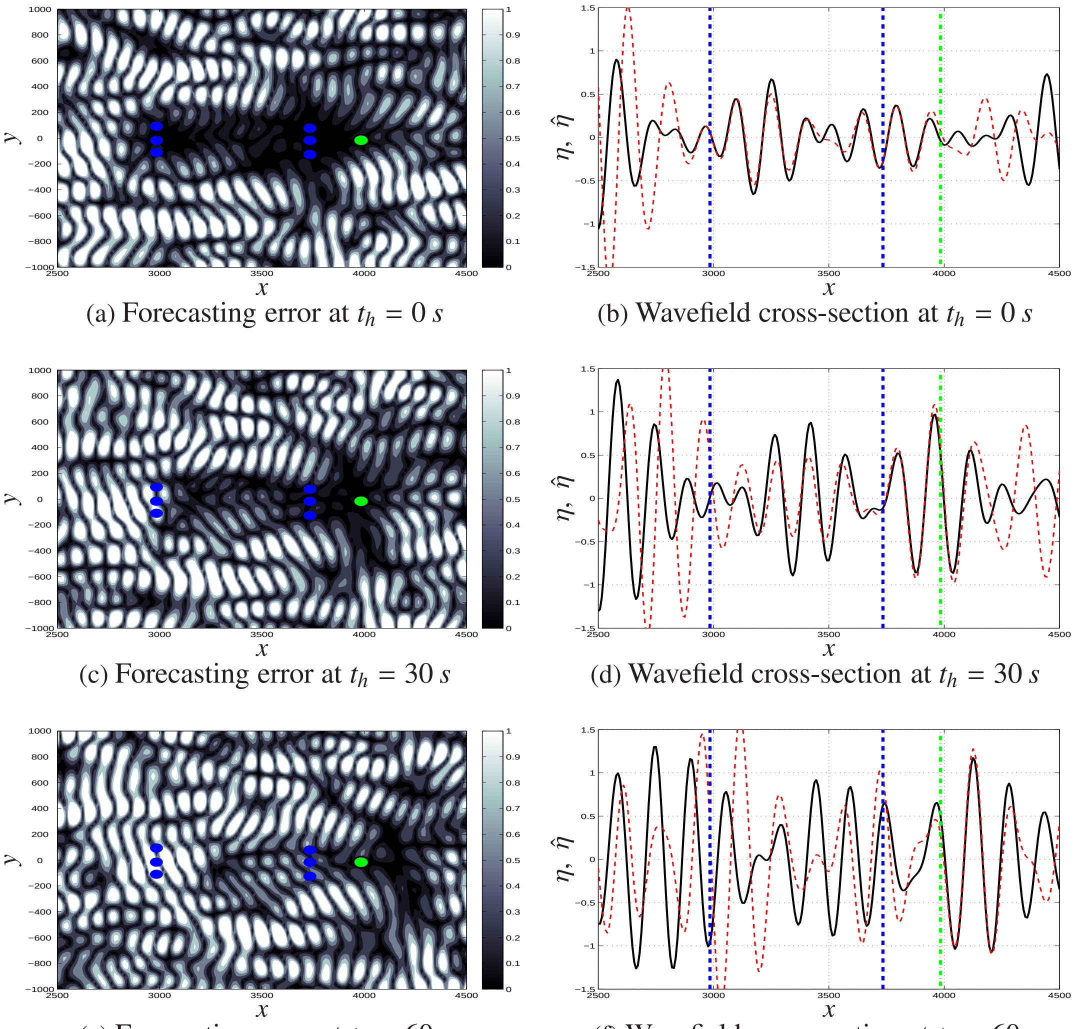
(c) Forecasting error at $t_h = 30\ s$

(d) Wavefield cross-section at $t_h = 30\ s$

(e) Forecasting error at $t_h = 60\ s$

(f) Wavefield cross-section at $t_h = 60\ s$

Figure 6.8: Current estimate, 30-second prediction, and 1-minute prediction using 250 ensemble members and a single row of three buoys placed 250 m upstream of the point of interest. Symbols marked as in Figure 6.7.



(a) Forecasting error at $t_h = 0\ s$

(b) Wavefield cross-section at $t_h = 0\ s$

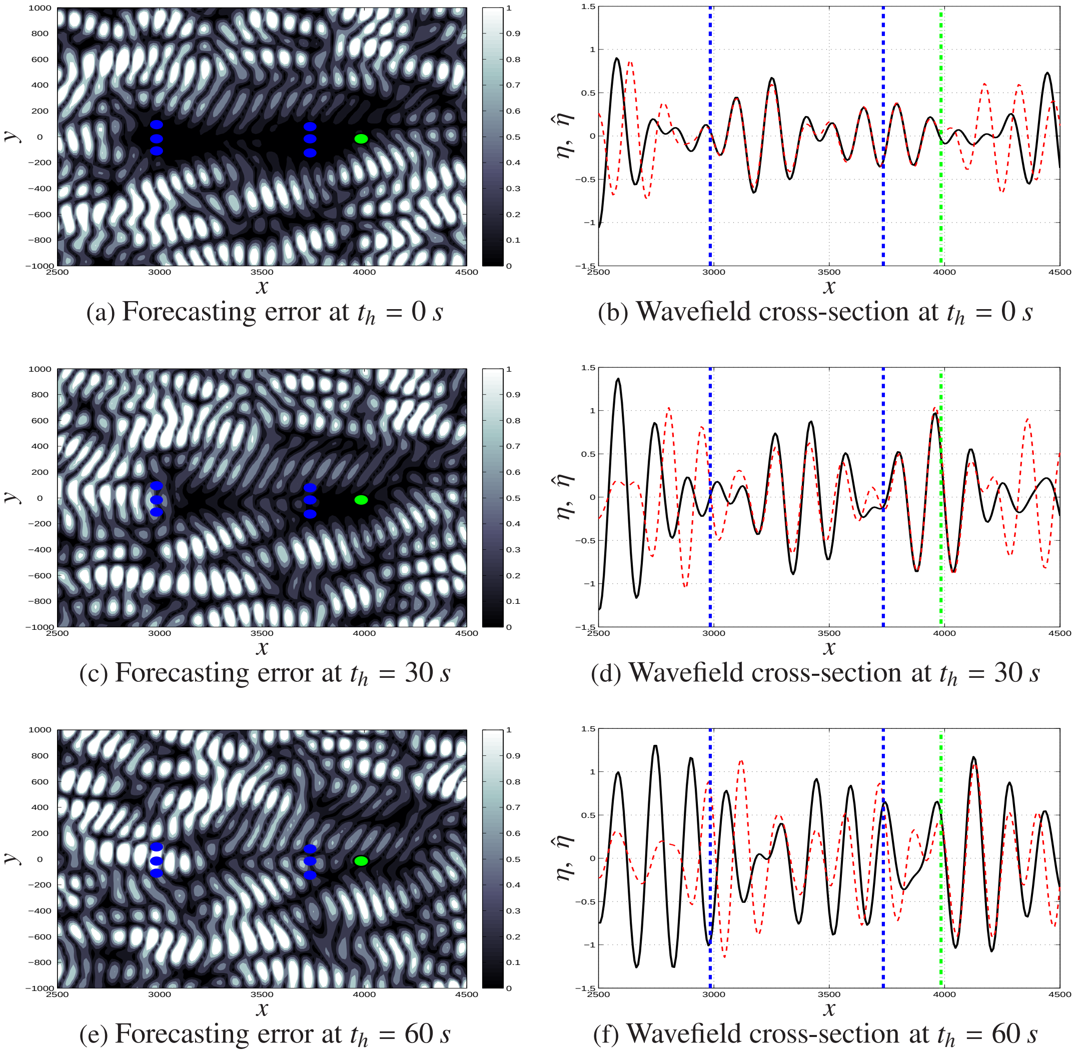
(c) Forecasting error at $t_h = 30\ s$

(d) Wavefield cross-section at $t_h = 30\ s$

(e) Forecasting error at $t_h = 60\ s$

(f) Wavefield cross-section at $t_h = 60\ s$

Figure 6.9: Current estimate, 30-second prediction, and 1-minute prediction using 125 ensemble members and two rows of three buoys each, placed 250 m and 1000 m upstream of the point of interest. Symbols marked as in Figure 6.7.



(a) Forecasting error at $t_h = 0\ s$

(b) Wavefield cross-section at $t_h = 0\ s$

(c) Forecasting error at $t_h = 30\ s$

(d) Wavefield cross-section at $t_h = 30\ s$

(e) Forecasting error at $t_h = 60\ s$

(f) Wavefield cross-section at $t_h = 60\ s$

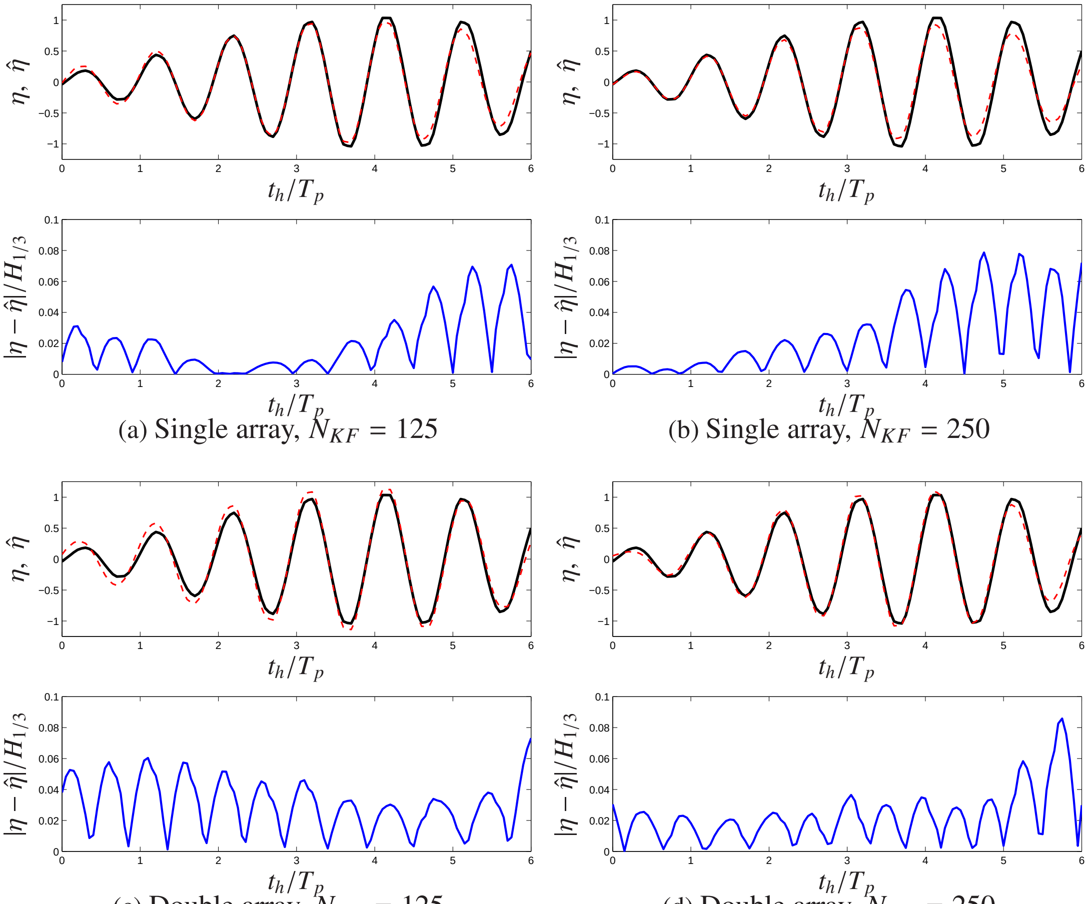
Figure 6.10: Current estimate, 30-second prediction, and 1-minute prediction using 250 ensemble members and two rows of three buoys each, placed 250 m and 1000 m upstream of the point of interest. Symbols marked as in Figure 6.7.

error which is initially decreasing in the first 30 seconds. This happens because the point of interest lies on the margin of the low-error region, while after thirty seconds it appears to be in the middle of it, as this region convects downstream. For the single array case with 250 ensemble members (Figure 6.11b), the initial estimation is improved with respect to the previous case, and increases after 30 seconds. The case with a double array of measurement buoys and 125 ensemble members (Figure 6.11c) initially shows somewhat worse performance in the first half of the forecasting horizon, though results improve slightly in the second half, as compared with the single array case. A similar trend is observed with 250 ensemble members. This indicates that the double array configuration offered only slight improvement in extending the length of the forecasting horizon while retaining adequate accuracy.

Comparing the configurations implementing wave radar alone with those implementing wave monitoring buoys alone shows that the estimation/forecasting error at the point of interest is generally somewhat improved in the wave radar case. Moreover, even those portions of the domain which are farther from the measurement location show higher errors, thus suggesting that reconstructing wave elevation by measuring wave velocity represents a much more challenging task, due to the extremely complicated nature of wave interaction.

6.6 Conclusions

We have developed a novel framework for ensemble wave forecasting based on the assimilation of measured data. Four different representative configurations of sensors have been considered: colocated or non-colocated wave radar, or a single or double array of wave monitoring buoys. Sensitivity with respect to the number of ensemble members has



(a) Single array, $N_{KF} = 125$

(b) Single array, $N_{KF} = 250$

(c) Double array, $N_{KF} = 125$

(d) Double array, $N_{KF} = 250$

Figure 6.11: Zero- to 60-second-ahead wave height (black), predicted wave height (red) and prediction error magnitude (blue, normalized by the significant wave height $H_{1/3}$), as a function of time (normalized by the dominant wave period $T_p = 60$ sec), estimated using measurement buoys configured as indicated.

also been investigated. When a sufficient number of ensemble members are used, all four configurations are shown to estimate and forecast the wave field at the point of interest with reasonable accuracy. Results illustrate that measuring the wave elevation directly (leveraging monitoring buoys), rather than measuring wave velocity (leveraging wave radar) generally offers somewhat improved forecast accuracy.

The number of ensemble members needed to provide accurate estimates and forecasts by the EnKF in this project ranged from 125 to 250. This is somewhat larger than the number of ensemble members typically used in weather forecasting, where 50-70 ensemble members are typically employed. This might be partially explained by the ambitious goal of the present setting, which is to provide an accurate pointwise forecast of the wave elevation at a point of interest. In contrast, in weather forecasting applications, the attention is usually focused on the prediction of atmospheric quantities (like humidity, precipitation, temperature, winds, and pressure) averaged over a broad region.

Simulations have shown that it is possible to perform wave forecasting with reasonable accuracy up to one minute into the future, under the assumption of a relatively narrow-band sea spectrum. The success of the forecasting framework developed here depends upon an accurate knowledge of the average wave spectrum. Note that a wider-band sea spectrum makes forecast much more difficult, as discussed in [54]. Uncertainty of the wave directionality is another factor that could quickly deteriorate the accuracy of a forecast. In this work, we have assumed that actual wavefield is made of a superposition of waves which are all travel close to a main direction of propagation, which is assumed to be accurately known. A degradation of performance is expected in the case of multidirectional wavefields. The accurate forecast results reported in this work should thus be considered in light of these limitations; significant further development and testing of this framework, and validation against actual wave data, is thus motivated.

Finally, we remark that the pseudo-spectral model for wave propagation adopted in this work does not account for near-shore wave shoaling, wave breaking, bottom friction and viscous effects. To account for such phenomena, a different wave model needs to be used, though the EnKF framework developed and implemented in the present work is still quite applicable. Wave forecasting algorithms which account for such significant physical phenomena should be developed and tested in future work.

Acknowledgements

This chapter contains work previously published in:

- D. Cavaglieri, T.R. Bewley, M. Previsic, “Short-term ensemble ocean wave forecasting", *Submitted to IEEE Transactions on Sustainable Energy*, 2016

Chapter 7

Nonlinear Model Predictive Control of a one- and two-body point absorber wave energy converter

7.1 Introduction

With the cost of fossil fuels consistently increasing, renewable energies have been receiving growing interest in recent years. Akin to other more mature fields of renewable energy, such as solar and wind energy, ocean wave energy conversion has recently raised significant attention. This has lead to the development of a variety of topologies for WEC devices, which have been undergoing extensive numerical simulations (see [70] and [71], for example). In some remarkable cases, the design has reached the prototype testing phase, which has been carried out in tanks [72] or actual ocean locations [73]. So far, the design optimization of such devices have relied on a rather passive approach, in which the structural parameters are defined in order to maximize the power take-off when the device is oscillating at the peak frequency of the sea spectrum at the location of installment.

This, however, has led to rather suboptimal performances, preventing wave energy conversion to achieve further competitiveness with respect to other renewable energy alternatives. In order to improve the efficiency of WECs, active control strategies have been developed and thoroughly investigated. Among the different topologies currently available, the point-absorber wave energy converter has emerged as the device of choice for benchmarking [48]. In [74], the performances of a broad selection of promising active control policies applied to the point absorber have been assessed. As a result, linear model predictive control (LMPC) has proved to outperform any other control logic developed so far. Furthermore, MPC allows to handle in a straightforward fashion the presence of structural constraints, such as actuator saturation and device motion constraints, which may prevent the device from experiencing mechanical failure in harsh operating conditions occurring in offshore applications. However, the classical LMPC formulation, although extremely appealing, presents fundamental limitations, since it can handle only linear and quadratic cost functions, and linear equality and inequality constraints. Thus, nonlinear effects affecting WEC dynamics cannot be properly accounted for and a linearization of those is required in the LMPC formulation, often leading to suboptimal results. Recently, nonlinear model predictive control has been applied to the optimization of a point absorber WEC subject to nonlinearities (such as mooring forces [75]) and time-varying parameters (such as adaptive PTO damping [76]).

The goal is to extend the application of NMPC to the optimization of the power take-off of a point-absorber subject to other nonlinear effects, such as drag forces. Two topologies will be considered for the point absorber: the one-body model, in which a floating buoy moored to the sea bed oscillates in heave, and a more realistic two-body model, in which the buoy oscillates with respect to a reaction plate immersed in water and moored to the sea floor. In both cases, two configurations of the PTO unit will be analyzed: one in which the PTO unit is able to absorb and produce power (two-way power flow), and one in which the actuator works in generator-mode only (one-way power flow). This last requirement is of particular interest, since it leads to the implementation of a PTO unit with a much simpler design. Differently from [76], no assumption is made on the actuator dynamics, and one-way power flow constraint is imposed as a nonlinear constraint in the NMPC formulation. In this way, the nonlinear dynamic model in the nonlinear optimization has constant coefficients and the cost function is still quadratic, which simplifies the solution of the NMPC optimization problem. Besides, the adoption of a direct multiple shooting strategy for the discretization of the state trajectories in the NMPC formulation, together with an analytic computation of the associated gradients and function values, contributes to further accelerate convergence of the nonlinear optimization problem, since it rules out any need for numerical differentiation tools such as automatic differentiation, finite differences, complex step differentiation, etc.). We want to remark that complete knowledge of the interacting wavefield is assumed. The problem associated to wave forecasting is not discussed here and we therefore refer to other works, such as [54] and [77], while for the problem of prediction-based MPC some preliminary results are reported in [51] and [62].

This chapter is organized as follows. In Section 7.2, a nonlinear state-space dynamic model is presented for the one-body and two-body wave energy converter. Section 7.3 introduces the direct multiple shooting NMPC formulation and the associated nonlinear programming problem, with particular attention to its numerical implementation. Performances of NMPC applied to a one- and two-body wave energy converter, under different operating conditions, are assessed in Section 7.4.

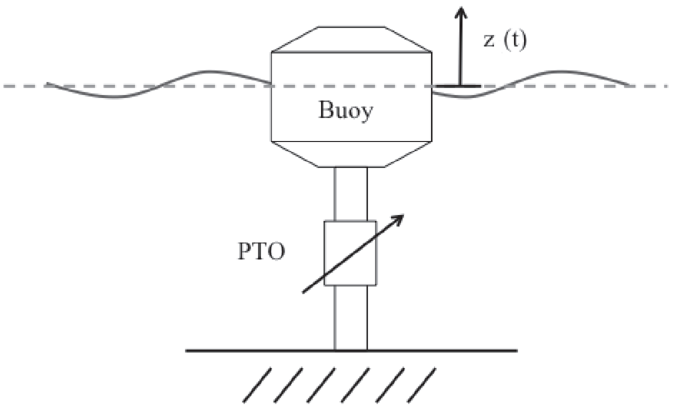


Figure 7.1: Model of a one-body point absorber WEC device.

7.2 One-body and two-body WEC model

7.2.1 One-body WEC

The one-body model for the point absorber wave energy converter (herein also dubbed WEC1) in heave is shown in Figure 7.1, in which the wave elevation at the device location at time t is indicated with $\eta(t)$, and the degree of freedom associated to heave motion is indicated with z. The dynamic model is obtained through a balance of forces affecting the device:

$$m\,\ddot{z}(t) + r\,\dot{z}(t) + k\,z(t) = f_D(t) + f_R(t) + f_e(t) + u(t) \tag{7.1}$$

where m is the WEC mass, r the viscous damping, $k = \rho g S_w$ the hydrostatic stiffness, where ρ is the water density, g is the gravitational constant, and S_w is the waterplane area. On the RHS, f_D is the nonlinear drag force, f_R the radiation force, f_e the excitation force, and u is the control force. The drag force is determined through Morrison equation:

$$f_D(t) = -\frac{1}{2}\rho S_w C_D \dot{z}(t)^2 \operatorname{sgn} \dot{z}(t) \tag{7.2}$$

where C_D is the drag coefficient. The radiation force $f_R(t)$ is defined, according to [78], as

$$\begin{aligned} f_R(t) &= -A_{11}\,\ddot{z}(t) - f_r(t) \\ &= -A_{11}\,\ddot{z}(t) - \int_{-\infty}^{t} h_r(t-\tau)\,\dot{z}(\tau)\,d\tau \end{aligned} \tag{7.3}$$

where A_{11} is the added mass and $h_r(t)$ is the radiation impulse response function. The excitation force f_e is defined as

$$f_e(t) = \int_{-\infty}^{+\infty} h_e(t-\tau)\,\eta(\tau)\,d\tau \tag{7.4}$$

where $h_e(t)$ is the excitation impulse response function. Differently from the radiation force, this impulse response function is noncausal, as thoroughly discussed in [49]. In order to obtain a nonlinear state-space form, the reduced radiation force f_r in (7.3) is approximated by a state-space representation, by introducing a dummy variable X_r, as done in [79]:

$$\begin{aligned} \dot{X}_r(t) &= A_r\,X_r(t) + B_r\,\dot{z}(t) \\ f_r(t) &= C_r\,X_r(t) + D_r\,\dot{z}(t) \end{aligned} \tag{7.5}$$

Combining (7.1) and (7.5) and introducing the state space vector $x = [X_r^T \quad z \quad \dot{z}]^T$ gives the state-space model:

$$\begin{aligned} \dot{x}(t) &= A\,x(t) + E\,f_D(x(t)) + E\,f_e(t) + B\,u(t) \\ &= \mathbf{f_1}(x(t);\ u(t),\ f_e(t)) \end{aligned} \tag{7.6}$$

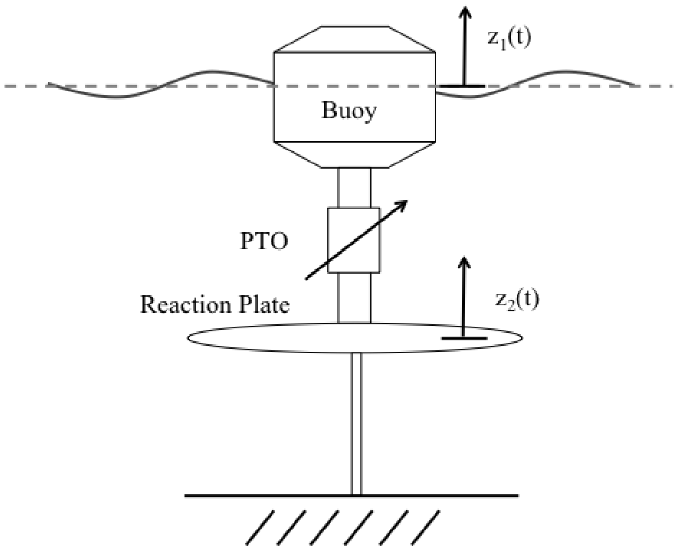


Figure 7.2: Model of a two-body point absorber WEC device.

with

$$A = \begin{bmatrix} A_r & 0 & B_r \\ 0 & 0 & I \\ -\frac{C_r}{m+A_{11}} & -\frac{k}{m+A_{11}} & -\frac{r+D_r}{m+A_{11}} \end{bmatrix} \qquad E = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m+A_{11}} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m+A_{11}} \end{bmatrix}$$

7.2.2 Two-body WEC

The derivation of the state-space model for the two-body WEC (dubbed WEC2) follows the formulation presented in [80]. An illustrative scheme of the device is shown in Figure 7.2, where the buoy displacement is indicated with z_1, while the reaction plate motion is indicated with z_2. The actuator acts between the buoy and the reaction plate. The associated equations of motion for this system are

$$\begin{cases} m_1\ddot{z}_1(t) + r_1\dot{z}_1(t) + k_1 z_1(t) = f_{D_1}(t) + f_{R_{11}}(t) + f_{R_{12}}(t) + f_{e_1}(t) + u(t) \\ m_2\ddot{z}_2(t) + r_2\dot{z}_2(t) + k_2 z_2(t) = f_{D_2}(t) + f_{R_{22}}(t) + f_{R_{21}}(t) + f_{e_2}(t) - u(t) \end{cases} \tag{7.7}$$

where subscript 1 indicated quantities referred to the buoy, while 2 refers to the reaction plate. In particular, k_2 accounts for both the hydrostatic stiffness of the reaction plate and the stiffness of the mooring system. As far as the radiation forces are concerned, each body experiences two contributions, one due to the displacement of the body and the other due to the interaction between the two connected bodies. Each radiation force $f_{R_{ij}}$ can be expressed as

$$\begin{aligned} f_{R_{ij}}(t) &= -A_{ij}\ddot{z}_j(t) - f_{r_{ij}}(t), \qquad i,\ j = 1,\ 2 \\ &= -A_{ij}\ddot{z}_j(t) - \int_{-\infty}^{t} h_{r_{ij}}(t-\tau)\,\dot{z}_j(\tau)\,d\tau \end{aligned} \tag{7.8}$$

The integral representing each reduced radiation force $f_{r_{ij}}$ can be discretized and represented as a state-space subsystem:

$$\begin{aligned} \dot{X}_{r_{ij}}(t) &= A_{r_{ij}}\,X_{r_{ij}}(t) + B_{r_{ij}}\dot{z}_j(t) \\ f_{r_{ij}}(t) &= C_{r_{ij}}\,X_{r_{ij}}(t) + D_{r_{ij}}\,\dot{z}_j(t) \end{aligned} \qquad i,\ j = 1,\ 2 \tag{7.9}$$

Each drag term is then defined as:

$$f_{D_i}(t) = -\frac{1}{2}\rho S_{w_i} C_{D_i}\dot{z}_i(t)^2\ \mathrm{sgn}\,\dot{z}_i(t), \qquad i = 1,\ 2 \tag{7.10}$$

Combining Equations (7.7) and (7.9), and introducing vector

$$x = [X_{r_{11}}^T\ X_{r_{12}}^T\ z_1\ \dot{z}_1\ X_{r_{22}}^T\ X_{r_{21}}^T\ z_2\ \dot{z}_2]^T$$

gives the state-space model

$$\begin{aligned} \dot{x}(t) &= A\,x(t) + E_1\,f_{D_1}(x(t)) + E_2\,f_{D_2}(x(t)) + E_1\,f_{e_1}(t) + E_2\,f_{e_2}(t) + B\,u(t) \\ &= \mathbf{f_2}(x(t);\ u(t),\ f_{e_1}(t),\ f_{e_2}(t)) \end{aligned} \tag{7.11}$$

with

$$A = \begin{bmatrix} A_{r_{11}} & 0 & 0 & B_{r_{11}} & 0 & 0 & 0 & 0 \\ 0 & A_{r_{12}} & 0 & 0 & 0 & 0 & 0 & B_{r_{12}} \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ -\frac{C_{r_{11}}}{m_{T_1}} & -\frac{C_{r_{12}}}{m_{T_1}} & -\frac{k_1}{m_{T_1}} & -\frac{r_1+D_{r_{11}}}{m_{T_1}(m_2+A_{22})} & \frac{A_{12}C_{r_{22}}}{m_{T_1}(m_2+A_{22})} & \frac{A_{12}C_{r_{21}}}{m_{T_1}(m_2+A_{22})} & \frac{A_{12}k_2}{m_{T_1}(m_2+A_{22})} & \frac{A_{12}(r_2+D_{r_{22}})}{m_{T_1}(m_2+A_{22})} \\ 0 & 0 & 0 & 0 & A_{r_{22}} & 0 & 0 & B_{r_{22}} \\ 0 & 0 & 0 & B_{r_{21}} & 0 & A_{r_{21}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \\ \frac{A_{21}C_{r_{11}}}{m_{T_2}(m_1+A_{11})} & \frac{A_{21}C_{r_{12}}}{m_{T_2}(m_1+A_{11})} & \frac{A_{21}k_1}{m_{T_2}(m_1+A_{11})} & \frac{A_{21}(r_1+D_{r_{11}})}{m_{T_2}(m_1+A_{11})} & -\frac{C_{r_{22}}}{m_{T_2}} & -\frac{C_{r_{21}}}{m_{T_2}} & -\frac{k_2}{m_{T_2}} & -\frac{r_2+D_{r_{22}}}{m_{T_2}} \end{bmatrix}$$

$$E_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m_{T_1}} \\ 0 \\ 0 \\ 0 \\ -\frac{A_{21}}{m_{T_2}(m_1+A_{11})} \end{bmatrix} \qquad E_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{A_{12}}{m_{T_1}(m_2+A_{22})} \\ 0 \\ 0 \\ 0 \\ \frac{1}{m_{T_2}} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m_{T_1}} + \frac{A_{12}}{m_{T_1}(m_2+A_{22})} \\ 0 \\ 0 \\ 0 \\ -\frac{1}{m_{T_2}} - \frac{A_{21}}{m_{T_2}(m_1+A_{11})} \end{bmatrix}$$

where $m_{T_1} = m_1 + A_{11} - A_{12}A_{21}/(m_2 + A_{22})$ and $m_{T_2} = m_2 + A_{22} - A_{12}A_{21}/(m_1 + A_{11})$. In the following, the two state-space models here described (7.6) and (7.11), will be leveraged in the NMPC formulation for the optimization of the power take-off.

7.3 Nonlinear Model Predictive Control formulation

The nonlinear Model Predictive Control formulation here described follows directly the work in [81] and [82]. At each instant t_0, a cost function $J(x, u)$ is minimized over a control horizon T_h subject to equality and inequality constraints involving the state-space vector x and the control input u, in order to determine the optimal control value u^* to impose at the next time instant. The associated nonlinear optimization problem appears as follows:

$$
\begin{aligned}
& \min_{x,\,u} J(x,\ u) = \min_{x,\,u} \int_{t_0}^{t_0+T_h} L(x(t),\ u(t))\,dt + E(x(t_0 + T)) \\
& \quad \text{subject to} \\
& \qquad \bar{x}_0 - x(t_0) = 0 && (7.12) \\
& \qquad \mathbf{f}(x(t),\ u(t),\ e(t)) - \dot{x} = 0, && t \in [t_0,\ t_0 + T_h] \\
& \qquad d(x(t),\ u(t)) \geq 0, && t \in [t_0,\ t_0 + T_h]
\end{aligned}
$$

where $\bar{x}_0$ is the value of the state vector at the beginning of the control interval, $e(t)$ represents exogenous inputs affecting system dynamics, and $d(x,\ u)$ represents the inequality constraints. The objective function is of Bolza type and it is composed by a Lagrange term $L(x,\ u)$ and a Meyer term $E(x(t_0 + T_h))$. The continuous-time optimization problem in (7.12) is first discretized following a direct multiple shooting approach: the control interval $[t_0,\ t_0 + T_h]$ is first divided into N smaller intervals $[t_k,\ t_{k+1}]$ of constant size Δt. In each interval, the control input $u(t)$ is discretized under a zero-order hold assumption, which means that a constant value u_k is assumed in each kth interval. The same assumption is made for the external disturbances $e(t)$, such as the excitation forces in the present application. Furthermore, in order to impose the dynamics constraints, a matching condition is imposed at the end of each interval $[t_k,\ t_{k+1}]$, i.e.

$$
\mathcal{X}_k(x_{k+1};\ x_k,\ u_k) - x_{k+1} = 0, \qquad t = t_{k+1}
$$

where $\mathcal{X}_k$ represents the state trajectory discretized over the interval. For brevity of notation the exogenous input $e(t)$ does not appear in the discretized trajectory formulation, even if it is necessary for its computation. After discretization, the nonlinear problem in (7.12) appears as

$$\begin{aligned}
&\min_{x_k,\, u_k} \sum_{k=0}^{N-1} L_k(x_k,\, u_k) + E(x_N) \\
&\text{subject to} \\
&\quad \bar{x}_0 - x_0 = 0 \\
&\quad c(x_k,\, x_{k+1},\, u_k) = \mathcal{X}_k(x_{k+1};\, x_k,\, u_k) - x_{k+1} = 0, \qquad k = 0,\, 1,\, \ldots,\, N-1 \\
&\quad d(x_k,\, u_k) \geq 0, \qquad k = 0,\, 1,\, \ldots,\, N-1
\end{aligned} \tag{7.13}$$

The nonlinear programming problem (NLP) in (7.13) is then solved leveraging a sequential quadratic programming (SQP) algorithm (see [83], for an extensive description). Introducing the vector of optimization variables $w = [\mathcal{U}^T\ \mathcal{X}^T]^T$, where

$$\begin{aligned}
\mathcal{U} &= [u_0^T\ u_1^T\ \ldots\ u_{N-1}^T]^T \\
\mathcal{X} &= [x_0^T\ x_1^T\ \ldots\ x_N^T]^T
\end{aligned}$$

starting from an initial guess $(w_0,\, \lambda_0)$, where λ is the Lagrange multiplier associated to the constraints, at each iteration i it is required to solve the following quadratic programming

(QP) problem:

$$\begin{aligned} &\min_{\Delta w} \frac{1}{2}\Delta w^T B^{(i)} \Delta w + b^{(i)\,T} \Delta w \\ &\text{subject to} \\ &\quad C^{(i)} \Delta w + c^{(i)} = 0 \\ &\quad D^{(i)} \Delta w + d^{(i)} \geq 0 \end{aligned} \tag{7.14}$$

where $B^{(i)}$ is the Hessian of the Lagrangian $\mathcal{L}(w,\ \lambda)$ associated to the NLP problem (7.13), and $b^{(i)}$ is the cost function gradient, evaluated at the current iteration $w^{(i)}$. Besides

$$\begin{aligned} C^{(i)} &= \nabla_w c(w)|_{w^{(i)}}\,, \qquad & c^{(i)} &= c(w^{(i)}) \\ D^{(i)} &= \nabla_w d(w)|_{w^{(i)}}\,, \qquad & d^{(i)} &= d(w^{(i)}) \end{aligned}$$

The solution is then updated through

$$w^{(i+1)} = w^{(i)} + \alpha \Delta w^* \tag{7.15}$$

where Δw^* is the optimal solution of the QP problem in (7.14) and $\alpha \in [0,\ 1]$ is a parameter to be determined using a line search algorithm (see [83] for further details). Convergence is achieved when the norm $||w^{(i+1)} - w^{(i)}||$ is less than a prescribed tolerance.

In the current implementation, the goal is to optimize the WEC average power take-off over the desired control horizon T_h, subject to motion constraints limiting the device maximum velocity and oscillation, actuator constraints defining the maximum control input $u_{\max}$, and the device nonlinear dynamics. This continuous-time nonlinear optimization problem for the one-body model in (7.6) can be formulated as

$$
\begin{aligned}
&\min \frac{1}{T_h} \int_{t_0}^{t_0+T_h} P_a(t)\, dt = \min \frac{1}{T_h} \int_{t_0}^{t_0+T_h} \dot{z}(t) u(t)\, dt \\
&\text{subject to} \\
&\qquad \bar{x}_0 - x(t_0) = 0 \\
&\qquad \mathbf{f_1}(x(t),\, u(t),\, f_e(t)) - \dot{x} = 0, && t \in [t_0,\, t_0 + T_h] \\
&\qquad |z(t)| \leq p_{\max}, && t \in [t_0,\, t_0 + T_h] \\
&\qquad |\dot{z}(t)| \leq v_{\max}, && t \in [t_0,\, t_0 + T_h] \\
&\qquad |u(t)| \leq u_{\max}, && t \in [t_0,\, t_0 + T_h]
\end{aligned}
\tag{7.16}
$$

Discretization of (7.16) leads to

$$
\begin{aligned}
&\min \frac{1}{2N} \sum_{k=0}^{N-1} x_{k+1}^T S_v^T u_k + u_k^T S_v x_{k+1} \\
&\text{subject to} \\
&\qquad \bar{x}_0 - x_0 = 0 \\
&\qquad c_k(x_k,\, x_{k+1},\, u_k) : \mathcal{X}_k(x_{k+1};\, x_k,\, u_k) - x_{k+1} = 0, && k = 0,\, 1,\, \ldots,\, N-1 \\
&\qquad |S_p\, x_k| \leq p_{\max}, && k = 0,\, 1,\, \ldots,\, N-1 \\
&\qquad |S_v\, x_k| \leq v_{\max}, && k = 0,\, 1,\, \ldots,\, N-1 \\
&\qquad |u_k| \leq u_{\max}, && k = 0,\, 1,\, \ldots,\, N-1
\end{aligned}
\tag{7.17a}
$$

where S_p and S_v are vectors extracting WEC position and velocity from the state vector x. For the case in which the PTO power flow is not reversible, i.e. the PTO is able to only absorb energy, the NLP problem in (7.17a) has to be modified by imposing the extra inequality $P_a(t) \leq 0$ over the entire control horizon T_h, i.e.:

$$\begin{aligned}
&\min \frac{1}{2N} \sum_{k=0}^{N-1} x_{k+1}^T S_v^T u_k + u_k^T S_v x_{k+1} \\
&\text{subject to} \\
&\quad \bar{x}_0 - x_0 = 0 \\
&\quad c_k(x_k,\, x_{k+1},\, u_k) : \mathcal{X}_k(x_{k+1};\, x_k,\, u_k) - x_{k+1} = 0, && k = 0, 1, \ldots, N-1 \\
&\quad |S_p\, x_k| \leq p_{\max}, && k = 0, 1, \ldots, N-1 \\
&\quad |S_v\, x_k| \leq v_{\max}, && k = 0, 1, \ldots, N-1 \\
&\quad |u_k| \leq u_{\max}, && k = 0, 1, \ldots, N-1 \\
&\quad \frac{1}{2}(x_{k+1}^T S_v^T u_k + u_k^T S_v x_{k+1}) \leq 0, && k = 0, 1, \ldots, N-1
\end{aligned} \tag{7.17b}$$

This constraint is quadratic, therefore it cannot be enforced using LMPC. We want to remark that, while solving the NLP problem in (7.17b), we have sometimes experienced poor convergence performances, mainly related to the choice of the initial guess in the SQP optimization. This problem is overcome by solving the “relaxed" NLP problem (7.17a) first and use the solution, although possibly unfeasible, as a first guess for problem (7.17b).

For the two-body model, the absorbed power depends instead on the relative velocity between the buoy and the reaction plate, and motion constraints involve relative displacement and velocity between the two bodies. The optimization problem appears as

$$\min \frac{1}{T_h} \int_{t_0}^{t_0+T_h} P_a(t)\, dt = \min \frac{1}{T_h} \int_{t_0}^{t_0+T_h} (\dot{z}_1(t) - \dot{z}_2(t))u(t)\, dt$$

$$\begin{aligned}
&\text{subject to} \\
&\bar{x}_0 - x(t_0) = 0 \\
&\mathbf{f_2}(x(t),\ u(t),\ f_{e\,1,2}(t)) - \dot{x} = 0, && t \in [t_0,\ t_0 + T_h] \\
&|z_1(t) - z_2(t)| \leq \Delta p_{\max}, && t \in [t_0,\ t_0 + T_h] \\
&|\dot{z}_1(t) - \dot{z}_2(t)| \leq \Delta v_{\max}, && t \in [t_0,\ t_0 + T_h] \\
&|u(t)| \leq u_{\max}, && t \in [t_0,\ t_0 + T_h]
\end{aligned} \tag{7.18}$$

Discretization of (7.18) leads to

$$\min \frac{1}{2N} \sum_{k=0}^{N-1} x_{k+1}^T S_{\Delta v}^T u_k + u_k^T S_{\Delta v} x_{k+1}$$

$$\begin{aligned}
&\text{subject to} \\
&\bar{x}_0 - x_0 = 0 \\
&c_k(x_k,\ x_{k+1},\ u_k) : \mathcal{X}_k(x_{k+1};\ x_k,\ u_k) - x_{k+1} = 0, && k = 0,\ 1,\ \ldots,\ N-1 \\
&|S_{\Delta p}\, x_k| \leq \Delta p_{\max}, && k = 0,\ 1,\ \ldots,\ N-1 \\
&|S_{\Delta v}\, x_k| \leq \Delta v_{\max}, && k = 0,\ 1,\ \ldots,\ N-1 \\
&|u_k| \leq u_{\max}, && k = 0,\ 1,\ \ldots,\ N-1
\end{aligned} \tag{7.19a}$$

where $S_{\Delta p}$ and $S_{\Delta v}$ are vectors extracting relative position and velocity between the two bodies of the WEC from the state vector x. As for the one-body WEC optimization case, the one-way power flow design requirement is imposed by adding the extra inequality:

$$\begin{aligned}
&\min \frac{1}{2N} \sum_{k=0}^{N-1} x_{k+1}^T S_{\Delta v}^T u_k + u_k^T S_{\Delta v} x_{k+1} \\
&\text{subject to} \\
&\quad \bar{x}_0 - x_0 = 0 \\
&\quad c_k(x_k,\, x_{k+1},\, u_k) : \mathcal{X}_k(x_k,\, u_k) - x_{k+1} = 0, && k = 0,\, 1,\, \ldots,\, N-1 \\
&\quad |S_{\Delta p}\, x_k| \leq \Delta p_{\max}, && k = 0,\, 1,\, \ldots,\, N-1 \\
&\quad |S_{\Delta v}\, x_k| \leq \Delta v_{\max}, && k = 0,\, 1,\, \ldots,\, N-1 \\
&\quad |u_k| \leq u_{\max}, && k = 0,\, 1,\, \ldots,\, N-1 \\
&\quad \frac{1}{2}(x_{k+1}^T S_{\Delta v}^T u_k + u_k^T S_{\Delta v} x_{k+1}) \leq 0, && k = 0,\, 1,\, \ldots,\, N-1
\end{aligned} \tag{7.19b}$$

Given a number of states n and control input m, the SQP algorithm for the NMPC optimization requires at each iteration i the solution of a QP problem in $mN + n(N+1)$ unknowns. For longer forecasting horizons, i.e. high N, this approach becomes quickly impractical. For this reason, a condensing approach [84] is adopted in order to leverage the sparsity pattern in the gradient of the equality constraints arising from the discretization of the system dynamics. At each iteration of the SQP algorithm, the gradient of the equality constraint, for both approaches described above, appears as:

$$C^{(i)} = \left[\begin{array}{ccc|cccc}
 & & & -I & & & \\
\partial c_0/\partial u_0|_{w^{(i)}} & & & \partial c_0/\partial x_0|_{w^{(i)}} & -I & & \\
 & \ddots & & & \ddots & \ddots & \\
 & & \partial c_{N-1}/\partial u_{N-1}|_{w^{(i)}} & & & \partial c_{N-1}/\partial x_{N-1}|_{w^{(i)}} & -I
\end{array}\right] \tag{7.20}$$

Considering the more compact notation $C_k^x = \partial c_k/\partial x_k|_{w^{(i)}}$, a block Gaussian elimination matrix $G^{(i)}$ can be constructed as

$$G^{(i)} = \begin{bmatrix} I & & & & \\ C_0^x & I & & & \\ C_1^x C_0^x & C_1^x & I & & \\ \vdots & \vdots & \ddots & \ddots & \\ \prod_{j=0}^{N-1} C_j^x & \prod_{j=1}^{N-1} C_j^x & \dots & C_{N-1}^x & I \end{bmatrix}$$

Multiplying the equality constraint equation in (7.14) by $G^{(i)}$ and rearranging allows to obtain an explicit expression of the state increment vector $\Delta\mathcal{X}$ as a linear function of the control input increment vector $\Delta\mathcal{U}$, i.e.

$$\Delta\mathcal{X} = C_u'^{(i)} \Delta\mathcal{U} + c'^{(i)} \tag{7.21}$$

where $c'^{(i)} = G^{(i)} c^{(i)}$, while $C_u'^{(i)}$ is the partition of the product $C'^{(i)} = G^{(i)} C^{(i)}$ inherent to the vector of unknowns $\Delta\mathcal{U}$. Replacing (7.21) into (7.14), allows to obtain a QP problem in the only variable $\Delta\mathcal{U}$. Suppressing superscript (i) for clarity of notation, we have

$$\begin{aligned} &\min_{\Delta\mathcal{U}} \frac{1}{2} \Delta\mathcal{U}^T B' \Delta\mathcal{U} + b'^T \Delta\mathcal{U} \\ &\text{subject to} \\ &\quad D' \Delta\mathcal{U} + d' \geq 0 \end{aligned} \tag{7.22}$$

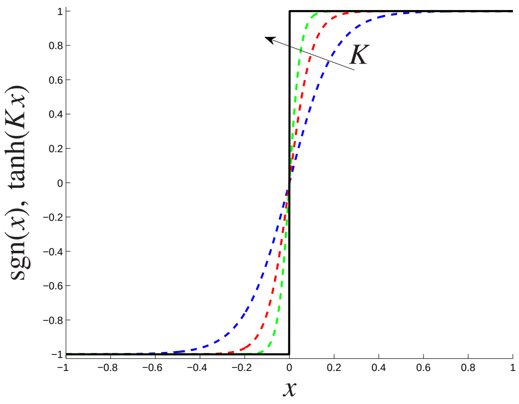


Figure 7.3: Comparison of sgn function (black solid line) against a smooth hyperbolic tangent approximation for different value of K: $K = 5$ (blue dashed), $K = 10$ (red dashed), and $K = 20$ (green dashed).

where

$$
\begin{aligned}
B' &= B_{uu} + B_{ux}C'_u + C'^T_u B_{xu} + C'^T_u B_{xx} C'_u \\
b' &= b_u + C'^T_u b_x + C'^T_u B_{xx} + (B_{ux} + B^T_{xu})\, c'/2 \\
D' &= D_u + D_x C'_u \\
d' &= d + D_x c'
\end{aligned}
$$

The number of unknowns to be determined at each iteration i of the SQP algorithm is now reduced to mN, which represents a significant improvement, since m is generally small in MPC applications ($m = 1$ in the present implementation).

Different choices are available to determine the discretized state trajectories χ_k. In particular, two approaches have been tested: in the first case we adopted a discrete time-stepping scheme to integrate the continuous-time nonlinear dynamics equations over the control interval; in the second we converted the nonlinear state-space model into discrete time following the Taylor-Lie approach derived in [85]. We then leveraged such model to propagate the state vector over the control interval. As it will be shown, both approaches allows a straightforward computation of the analytic gradients for the SQP solver.

The necessity of providing smooth analytic gradients, in order to facilitate convergence of the nonlinear solver, has justified the introduction of a smooth approximation for the drag force. In this fashion, the drag forces in (7.6) and (7.11). are replaced by the approximation:

$$f_{D_i}(t) = -\frac{1}{2}\rho S_{w_i} C_{D_i} \dot{z}_i(t)^2 \tanh\left[K\, \dot{z}_i(t)\right] \tag{7.23}$$

where K is a tuning parameter affecting the zero-crossing slope. As shown in Figure 7.3, a value of K equal to 20 guarantees an excellent approximation.

7.3.1 Continuous-time approach

Two cases will be considered: the first one arises when the time-stepping scheme allows a stepsize equal to the size of the shooting interval Δt. In this case, the state trajectory over each interval $[t_k,\, t_{k+1}]$ can be calculated through a single step of time integration. Amongst the innumerable choices of time integration algorithms, explicit Runge-Kutta (ERK) schemes (see [30] for an extensive discussion) represents the way to go for the present application. As a matter of fact, Runge-Kutta schemes have the advantage over linear multistep methods (LMM) of avoiding the introduction of spurious solutions and being self-starting. The adoption of an explicit scheme over an implicit one is due to the fact that it leads to an easier computation of the gradients. Besides, implicit scheme are generally preferred for the integration of stiff systems [20], which is not the case of our application. Considering the following ODE:

$$\dot{x}(t) = \mathbf{f}(x(t),\, u(t),\, e(t)) \tag{7.24}$$

and an s-stage ERK scheme with the associated Butcher tableau:

$$\begin{array}{c|cccccc} 0 & 0 & & & & \\ c_2 & a_{2,1} & 0 & & & \\ c_3 & a_{3,1} & a_{3,2} & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

discretization of the state constraint through a single step of ERK integration over the time interval $[t_k,\ t_{k+1}]$ appears as

$$\begin{aligned} &c_k(x_k,\ x_{k+1},\ u_k) : x_k + \Delta t \sum_{i=1}^{s} b_i \mathbf{K}_i(x_k,\ u_k) - x_{k+1} = 0, \quad k = 0,\ 1,\ \ldots,\ N-1 \\ &\quad \mathbf{K}_1 = \mathbf{f}\,(x_k,\ u_k,\ e_k) \\ &\quad \mathbf{K}_i = \mathbf{f}\left(x_k + \Delta t \sum_{j=1}^{i-1} a_{i,j}\mathbf{K}_j,\ u_k,\ e_k \right), \quad i = 2,\ 3,\ \ldots,\ s \end{aligned} \tag{7.25}$$

Once vectors $\mathbf{K}_i$ are calculated and stored, gradient computation cal be easily performed through the following recursion:

$$\begin{aligned} &\frac{\partial c_k}{\partial u_k} = \Delta t \sum_{i=1}^{s} b_i \frac{\partial \mathbf{K}_i}{\partial u_k}, \quad k = 0,\ 1,\ \ldots,\ N-1 \\ &\quad \frac{\partial \mathbf{K}_1}{\partial u_k} = \frac{\partial \mathbf{f}}{\partial u}\,(x_k,\ u_k,\ e_k) \\ &\quad \frac{\partial \mathbf{K}_i}{\partial u_k} = \frac{\partial \mathbf{f}}{\partial u}\left(x_k + \Delta t \sum_{j=1}^{i-1} a_{i,j}\mathbf{K}_j,\ u_k,\ e_k \right) + \frac{\partial \mathbf{f}}{\partial x}\left(x_k + \Delta t \sum_{j=1}^{i-1} a_{i,j}\mathbf{K}_j,\ u_k,\ e_k \right)\left(\Delta t \sum_{j=1}^{i-1} a_{i,j} \frac{\partial \mathbf{K}_j}{\partial u_k} \right), \quad i = 2,\ 3,\ \ldots,\ s \\ &\frac{\partial c_k}{\partial x_k} = I + \Delta t \sum_{i=1}^{s} b_i \frac{\partial \mathbf{K}_i}{\partial x_k}, \quad k = 0,\ 1,\ \ldots,\ N-1 \\ &\quad \frac{\partial \mathbf{K}_1}{\partial x_k} = \frac{\partial \mathbf{f}}{\partial x}\,(x_k,\ u_k,\ e_k) \\ &\quad \frac{\partial \mathbf{K}_i}{\partial x_k} = \frac{\partial \mathbf{f}}{\partial x}\left(x_k + \Delta t \sum_{j=1}^{i-1} a_{i,j}\mathbf{K}_j,\ u_k,\ e_k \right)\left(I + \Delta t \sum_{j=1}^{i-1} a_{i,j} \frac{\partial \mathbf{K}_j}{\partial x_k} \right), \quad i = 2,\ 3,\ \ldots,\ s \\ &\frac{\partial c_k}{\partial x_{k+1}} = -I, \quad k = 0,\ 1,\ \ldots,\ N-1 \end{aligned} \tag{7.26}$$

However, as is often the case, in order to guarantee the stability of the time stepping scheme, a smaller stepsize Δt_L must be defined such that $\Delta t_L = \Delta t/L$, where L is a positive integer value. In this case, the equality constraints c_k are determined through the recursion:

$$
\begin{aligned}
& c_k(x_k,\, x_{k+1},\, u_k) : \mathcal{X}_k^{(L)}(x_k,\, u_k) - x_{k+1} = x_k + \sum_{l=0}^{L-1} \Delta\mathcal{X}_k^{(l)}(x_k,\, u_k) - x_{k+1} = 0, \quad k = 0,\, 1,\, \ldots,\, N-1 \\
& \quad \mathcal{X}_k^{(0)} = x_k \\
& \quad \mathcal{X}_k^{(l+1)} = \mathcal{X}_k^{(l)} + \Delta\mathcal{X}_k^{(l)} = \mathcal{X}_k^{(l)} + \Delta t_L \sum_{i=1}^{s} b_i \mathbf{K}_i^{(l)}(\mathcal{X}_k,\, u_k), \quad l = 0,\, 1,\, \ldots,\, L-1 \\
& \quad \mathbf{K}_1^{(l)} = \mathbf{f}\left(\mathcal{X}_k^{(l)},\, u_k,\, e_k\right) \\
& \quad \mathbf{K}_i^{(l)} = \mathbf{f}\left(\mathcal{X}_k^{(l)} + \Delta t_L \sum_{j=1}^{i-1} a_{i,j}\mathbf{K}_j^{(l)},\, u_k,\, e_k\right), \quad i = 2,\, 3,\, \ldots,\, s
\end{aligned}
\tag{7.27}
$$

While the function evaluations $\mathbf{K}_i^{(l)}$ are being calculated in each interval k, the gradients can be recursively computed as well:

$$
\begin{aligned}
& \frac{\partial c_k}{\partial u_k} = \frac{\partial \mathcal{X}_k^{(L)}}{\partial u_k} = \sum_{l=0}^{L-1}\left(\frac{\partial \Delta\mathcal{X}_k^{(l)}}{\partial u_k} + \frac{\partial \Delta\mathcal{X}_k^{(l)}}{\partial \mathcal{X}_k^{(l)}}\frac{\partial \mathcal{X}_k^{(l)}}{\partial u_k}\right) = \ldots, \qquad k = 0,\, 1,\, \ldots,\, N-1 \\
& \quad \frac{\partial \mathcal{X}^{(0)}}{\partial u_k} = 0 \\
& \quad \frac{\partial \Delta\mathcal{X}_k^{(l)}}{\partial u_k} = \Delta t_L \sum_{i=1}^{s} b_i \frac{\partial \mathbf{K}_i^{(l)}}{\partial u_k}, \quad l = 0,\, 1,\, \ldots,\, L-1 \\
& \quad \frac{\partial \mathbf{K}_1^{(l)}}{\partial u_k} = \frac{\partial \mathbf{f}}{\partial u}\left(\mathcal{X}_k^{(l)},\, u_k,\, e_k\right) \\
& \quad \frac{\partial \mathbf{K}_i^{(l)}}{\partial u_k} = \frac{\partial \mathbf{f}}{\partial x}\left(\mathcal{X}_k^{(l)} + \Delta t_L \sum_{j=1}^{i-1} a_{i,j}\mathbf{K}_j^{(l)},\, u_k,\, e_k\right)\left(\Delta t_L \sum_{j=1}^{i-1} a_{i,j}\frac{\partial \mathbf{K}_j^{(l)}}{\partial u_k}\right), \quad i = 2,\, 3,\, \ldots,\, s \\
& \frac{\partial c_k}{\partial x_k} = \frac{\partial \mathcal{X}_k^{(L)}}{\partial x_k} = I + \sum_{l=0}^{L-1} \frac{\partial \Delta\mathcal{X}_k^{(l)}}{\partial \mathcal{X}_k^{(l)}}\frac{\partial \mathcal{X}_k^{(l)}}{\partial x_k} = \ldots, \qquad k = 0,\, 1,\, \ldots,\, N-1 \\
& \quad \frac{\partial \mathcal{X}^{(0)}}{\partial x_k} = I \\
& \quad \frac{\partial \Delta\mathcal{X}_k^{(l)}}{\partial \mathcal{X}_k^{(l)}} = \Delta t_L \sum_{i=1}^{s} b_i \frac{\partial \mathbf{K}_i^{(l)}}{\partial \mathcal{X}_k^{(l)}}, \quad l = 0,\, 1,\, \ldots,\, L-1 \\
& \quad \frac{\partial \mathbf{K}_1^{(l)}}{\partial \mathcal{X}_k^{(l)}} = \frac{\partial \mathbf{f}}{\partial x}\left(\mathcal{X}_k^{(l)},\, u_k,\, e_k\right) \\
& \quad \frac{\partial \mathbf{K}_i^{(l)}}{\partial \mathcal{X}_k^{(l)}} = \frac{\partial \mathbf{f}}{\partial x}\left(\mathcal{X}_k^{(l)} + \Delta t_L \sum_{j=1}^{i-1} a_{i,j}\mathbf{K}_j^{(l)},\, u_k,\, e_k\right)\left(I + \Delta t_L \sum_{j=1}^{i-1} a_{i,j}\frac{\partial \mathbf{K}_j^{(l)}}{\partial \mathcal{X}_k^{(l)}}\right), \quad i = 2,\, 3,\, \ldots,\, s \\
& \frac{\partial c_k}{\partial x_{k+1}} = -I, \quad k = 0,\, 1,\, \ldots,\, N-1
\end{aligned}
\tag{7.28}
$$

This approach allows the computation of analytic gradients in a simple and rapid way.

7.3.2 Discrete-time approach

Consider a nonlinear continuous-time system like

$$\dot{x}(t) = f(x(t)) + g_u(x(t))\,u(t) + g_e(x(t))\,e(t) \tag{7.29}$$

a discrete-time approximation, up to a specified order M, can be obtained leveraging a Taylor-Lie series expansion [85] as

$$x_{k+1} = \Phi_h^M(x_k,\, u_k,\, e_k) = x_k + \sum_{i=1}^{M} \frac{h^i}{i!}\, \frac{d^i x}{dt^i}\bigg|_{t_k} = x_k + \sum_{i=1}^{M} \frac{h^i}{i!} A^{[i]}(x_k,\, u_k,\, e_k) \tag{7.30}$$

where h is the sampling. Each $A^{[i]}$ is then determined through the following recursive formulas:

$$\begin{aligned} A^{[1]}(x,\, u,\, e) &= f(x) + g_u(x)\,u + g_e(x)\,e \\ A^{[i+1]}(x,\, u,\, e) &= \frac{\partial A^{[i]}(x,\, u,\, e)}{\partial x}(f(x) + g_u(x)\,u + g_e(x)\,e) \end{aligned}$$

Significant simplification is obtained in the present application, since the function $g_u(x)$ in (7.29) is constant and equal to B, while $g_e(x) = E$ for WEC1 and $[E_1\, E_2]$ for WEC2. A model order $M = 3$ has been considered, since for $M = 2$ the discrete-time model is unstable for $\Delta t > 0.18$. Introducing for simplicity of notation matrix $\mathbf{B}$ composed of the control matrix B and exogenous inputs E_i and, likewise, vector $\mathbf{u}_k = [u_k\, f_{e_i}]$, the third-order nonlinear discrete-time model for the WEC dynamics is

$$\begin{aligned} x_{k+1} &= x_k + hf(x_k) + \frac{h^2}{2}f'(x_k)f(x_k) + \frac{h^3}{6}f''(x_k)f(x_k)f(x_k) + \frac{h^3}{6}f'(x_k)f'(x_k)f(x_k) + \\ &+ h(\mathbf{Bu}_k) + \frac{h^2}{2}f'(x_k)(\mathbf{Bu}_k) + \frac{h^3}{6}f''(x_k)(\mathbf{Bu}_k)f(x_k) + \frac{h^3}{6}f''(x_k)f(x_k)(\mathbf{Bu}_k) + \\ &+ \frac{h^3}{6}f'(x_k)f'(x_k)(\mathbf{Bu}_k) + \frac{h^3}{6}f''(x_k)(\mathbf{Bu}_k)(\mathbf{Bu}_k) \end{aligned} \tag{7.31}$$

Leveraging this expression, the dynamics constraint c_k over the interval $[t_k,\ t_{k+1}]$ can be written as a single application of (7.31), by considering $h = \Delta t$, i.e.

$$\begin{aligned} c_k(x_k,\ x_{k+1},\ u_k) : &\ x_k + \Delta t f(x_k) + \frac{\Delta t^2}{2}f'(x_k)f(x_k) + \frac{\Delta t^3}{6}f''(x_k)f(x_k)f(x_k) + \frac{\Delta t^3}{6}f'(x_k)f'(x_k)f(x_k) + \\ &+ \Delta t(\mathbf{Bu}_k) + \frac{\Delta t^2}{2}f'(x_k)(\mathbf{Bu}_k) + \frac{\Delta t^3}{6}f''(x_k)(\mathbf{Bu}_k)f(x_k) + \frac{\Delta t^3}{6}f''(x_k)f(x_k)(\mathbf{Bu}_k) + \\ &+ \frac{\Delta t^3}{6}f'(x_k)f'(x_k)(\mathbf{Bu}_k) + \frac{\Delta t^3}{6}f''(x_k)(\mathbf{Bu}_k)(\mathbf{Bu}_k) - x_{k+1} = 0 \end{aligned} \tag{7.32}$$

Gradients with respect to the variables u_k, x_k, and x_{k+1} are easily determined as

$$\begin{aligned} \frac{\partial c_k}{\partial u_k} &= \Delta t B + \frac{\Delta t^2}{2}f'(x_k)B + \frac{\Delta t^3}{6}f''(x_k)Bf(x_k) + \frac{\Delta t^3}{6}f''(x_k)f(x_k)B + \frac{\Delta t^3}{6}f'(x_k)f'(x_k)B + \frac{\Delta t^3}{3}f''(x_k)(Bu_k)B \\ \frac{\partial c_k}{\partial x_k} &= I + \Delta t f'(x_k) + \frac{\Delta t^2}{2}f'(x_k)f(x_k) + \frac{\Delta t^2}{2}f''(x_k)f(x_k) + \frac{\Delta t^2}{2}f'(x_k)f'(x_k) + \\ &+ \frac{\Delta t^3}{6}f'''(x_k)f(x_k)f(x_k) + \frac{\Delta t^3}{6}f''(x_k)f'(x_k)f(x_k) + \frac{\Delta t^3}{6}f''(x_k)f(x_k)f'(x_k) + \\ &+ \frac{\Delta t^3}{6}f''(x_k)f'(x_k)f(x_k) + \frac{\Delta t^3}{6}f'(x_k)f''(x_k)f(x_k) + \frac{\Delta t^3}{6}f'(x_k)f'(x_k)f'(x_k) + \\ &+ \frac{\Delta t^2}{2}f''(x_k)(\mathbf{Bu}_k) + \frac{\Delta t^3}{6}f'''(x_k)(\mathbf{Bu}_k)f(x_k) + \frac{\Delta t^3}{6}f''(x_k)(\mathbf{Bu}_k)f'(x_k) + \frac{\Delta t^3}{6}f'''(x_k)f(x_k)(\mathbf{Bu}_k) + \\ &+ \frac{\Delta t^3}{6}f''(x_k)f'(x_k)(\mathbf{Bu}_k) + \frac{\Delta t^3}{6}f''(x_k)f'(x_k)(\mathbf{Bu}_k) + \frac{\Delta t^3}{6}f'(x_k)f''(x_k)(\mathbf{Bu}_k) + \frac{\Delta t^3}{6}f'''(x_k)(\mathbf{Bu}_k)(\mathbf{Bu}_k) \\ \frac{\partial c_k}{\partial x_{k+1}} &= -I \end{aligned} \tag{7.33}$$

Notice that in the present application, the computation of the tensor $f'''(x_k)$ is not needed, since the only nonlinearity in the system is represented by the drag term, which is quadratic with respect to the state, hence $f'''(x) \equiv 0$.

As for the continuous-time case, a timestep which is independent of the size of the shooting interval is generally preferred, since stepsize directly affects the integration error.

In this case, given a desired timestep Δt_D, the discrete state trajectory is determined through $D = \Delta t/\Delta t_D$ recursive applications of (7.31) with $h = \Delta t_D$, i.e.

$$\begin{aligned} c_k(x_k, x_{k+1}, u_k) : \mathcal{X}_k^{(D)}(x_k, u_k) - x_{k+1} = 0, \quad k = 0, 1, \ldots, N-1 \\ \mathcal{X}_k^{(0)} = x_k \\ \mathcal{X}_k^{(d+1)} = \Phi^M_{\Delta t_D}(\mathcal{X}_k^{(d)}, \mathbf{u}_k), \quad d = 0, 1, \ldots, D-1 \end{aligned} \tag{7.34}$$

Gradients computation is performed through the application of the following recursive formulas:

$$\begin{aligned} &\frac{\partial c_k}{\partial u_k} = \frac{\partial \mathcal{X}_k^{(D)}}{\partial u_k} = \frac{\partial \Phi^M_{\Delta t_D}}{\partial u_k}(\mathcal{X}_k^{(D-1)}, \mathbf{u}_k) + \frac{\partial \Phi^M_{\Delta t_D}}{\partial x_k}(\mathcal{X}_k^{(D-1)}, \mathbf{u}_k)\frac{\partial \mathcal{X}_k^{(D-1)}}{\partial u_k} = \ldots, \quad k = 0, 1, \ldots, N-1 \\ &\quad \frac{\partial \mathcal{X}_k^{(0)}}{\partial u_k} = 0 \\ &\frac{\partial c_k}{\partial x_k} = \frac{\partial \mathcal{X}_k^{(D)}}{\partial x_k} = \frac{\partial \Phi^M_{\Delta t_D}}{\partial x_k}(\mathcal{X}_k^{(D-1)}, \mathbf{u}_k)\frac{\partial \mathcal{X}_k^{(D-1)}}{\partial x_k} = \ldots, \quad k = 0, 1, \ldots, N-1 \\ &\quad \frac{\partial \mathcal{X}_k^{(0)}}{\partial x_k} = I \\ &\frac{\partial c_k}{\partial x_{k+1}} = -I \end{aligned} \tag{7.35}$$

Again, recursion allows a minimal storage implementation.

7.4 Simulations

The performance of NMPC applied to the optimization of the power take-off of a one- and two-body WEC model have been evaluated when the wave energy converter operates under ideal conditions of pure sinusoidal wave excitation and irregular waves sampled from a realistic sea spectrum. More specifically, a JONSWAP distribution [63] has been considered for the generation of irregular waves. Given a specified significant wave height $H_{1/3}$ and dominant wave period T_p, the sea spectrum is defined as a function of the frequency ω as

$$
\begin{aligned}
S(\omega) &= 155\,\frac{H_{1/3}^2}{T_p^4\omega^5}\,e^{\frac{-944}{T_p^4\omega^4}}\,(3.3)^Y, \\
\text{with}\quad Y &= e^{\frac{-(0.191\,\omega\,T_p-1)^2}{2\,\sigma^2}}, \\
\text{and}\quad \sigma &= \begin{cases} 0.07, & \omega \leq 5.24/T_p \\ 0.09, & \omega > 5.24/T_p \end{cases}
\end{aligned}
\tag{7.36}
$$

The wave elevation time series at the device location $\eta(t)$ is determined as a superposition of $N_w = 40$ waves, i.e.

$$
\eta(t) = \sum_{j=1}^{N_w} \sqrt{2S(\omega_j)\Delta\omega}\,\cos(\omega_j t + \varepsilon_j) \tag{7.37}
$$

where $\varepsilon_j \in \mathcal{U}(0,\,2\pi)$ is a uniformly random phase shifting. The physical parameters defining the one-body and two-body WEC devices here considered are reported in Table 7.1.

The optimization constraints in Section 7.3 are chosen as $p_{\max} = \Delta p_{\max} = 5\,m$, $v_{\max} = \Delta v_{\max} = 5\,m/s$, and $u_{\max} = 10\,MN$. A control horizon of two wave periods for the case of pure sinusoidal wave and two dominant wave periods for the case of irregular waves is considered. Such horizon is then divided into N shooting intervals of size 0.4 s. In each interval, the equality constraints arising from the nonlinear dynamics are computed using either the continuous-time approach or the discrete-time approach presented in Section 7.3. For the continuous-time case, **ERK4** is used for time stepping, while for the discrete-time case, a third-order nonlinear model is employed. In both cases, an integration time step of 0.1 s is considered. We have to remark that in all the simulations in this chapter, the employment of either approaches has led to comparable results. In particular, convergence is achieved in 15-20 iterations, independently of the discretization scheme adopted.

Table 7.1: System parameters for WEC1 and WEC2.

WEC1	
Parameter	Value
m	$6.44e5\,kg$
A_{11}	$1.44e6\,kg$
r	$5.04e3\,Ns/m$
k	$3.01e6\,N/m$
$\rho S_w C_D$	$3.62e5\,kg/m$

WEC2	
Parameter	Value
m_1	$6.44e5\,kg$
A_{11}	$1.44e6\,kg$
r_1	$5.04e3\,Ns/m$
k_1	$3.01e6\,N/m$
$\rho S_{w1} C_{D1}$	$3.62e5\,kg/m$
m_2	$3.62e5\,kg$
A_{22}	$9.12e6\,kg$
A_{12}	$2.64e5\,kg$
A_{21}	$2.64e5\,kg$
r_2	$7.12e3\,Ns/m$
k_2	$5.28e5\,N/m$
$\rho S_{w2} C_{D2}$	$1.09e6\,kg/m$

Therefore, as a general guideline, the continuous-time approach leveraging **ERK4** is to be preferred, since it benefits from higher discretization accuracy and lower computational time required to compute the gradients, as compared to the discrete-time approach based on the third-order Taylor-Lie approximation. However, in the present application, the computation of the tensor $f'''(x_k)$ is avoided since it is identically zero. This makes the computational cost of the discrete-time approach comparable to the continuous-time formulation. Furthermore, the second-order discrete model leads to a computationally faster approach, with respect to the **ERK4** approach, hence it is to be preferred whenever it provides a sufficiently accurate approximation of the continuous-time model. Once the optimal solution $u^*(t)$ is determined over the forecasting horizon, only the first control value is imposed at the next time iteration, then the NMPC optimization is repeated with the updated initial state condition $\bar{x}_0$.

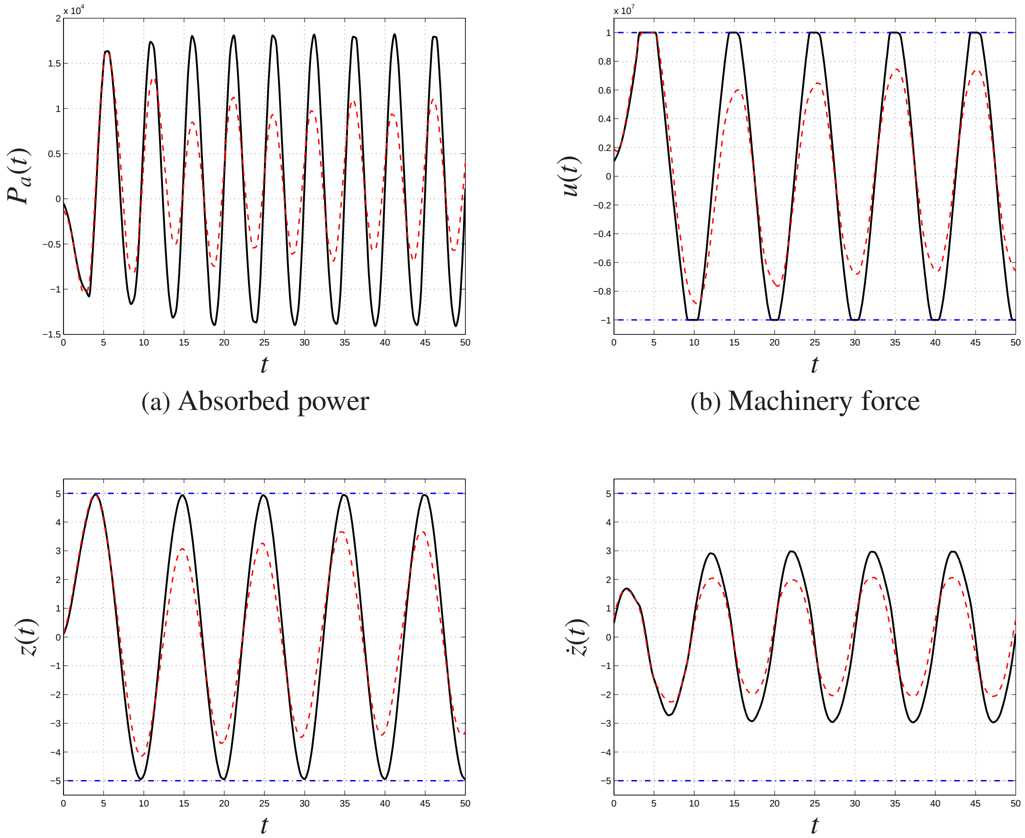
We first compared the improvement that nonlinear MPC guarantees over linear MPC for the one-body WEC. The linear formulation is obtained by linearizing the drag force $f_D(t)$ around a reference velocity $\bar{v}$. The linearized viscous force $\bar{f}_D(t)$ is then defined as

$$\bar{f}_D(t) = -\rho S_w C_D |\bar{v}| \dot{z}(t) \tag{7.38}$$

For the present simulations, a reference velocity $\bar{v} = 2\, m/s$ is assumed. This value has been obtained by minimizing the difference in position and velocity between linear and nonlinear model for WEC1 when operating in a seastate characterized by a JONSWAP spectral distribution with $H_{1/3} = 3\, m$ and $T_p = 10\, s$.

Since the state-space model, as well as other constraints, are linear and the cost function is quadratic, the optimization problem is solved by a single quadratic programming iteration. Results are reported in Figures 7.4-7.6. The case of pure sinusoidal forcing with a peak-to-trough amplitude of $3\,m$ and a period of $10\,s$ shows substantial discrepancy between the performance of linear MPC and nonlinear MPC (Figure 7.4). This is due to an erroneous approximation of the drag term, which leads to an optimal control law with higher magnitude with respect to the one derived through NMPC. As a results, the absorbed power with LMPC has higher oscillations than with NMPC, but the latter gives an average power take-off which is 78% higher. This gain is reduced to 23% for irregular waves with a significant wave height of $3\,m$, and a dominant wave period of $10\,s$. This is due to the constraints on the machinery force, preventing the LMPC solution from overshooting with respect to the nonlinear solution. This phenomenon is also noticed in Figure 7.6, where the device is tested against a pure sinusoid of amplitude $5\,m$ and period $10\,s$. In this case, the performance gain is only 4%, since both control laws oscillates with an amplitude close to the constraints. In general, the gain of NMPC over LMPC is higher in those sea conditions which force the device to work outside the region of validity of the linear drag force approximation. This is the case for irregular waves with small amplitude, for example.

Figures 7.7 and 7.8 show the performance of NMPC applied to the one-body WEC described in Section 7.2 with two-way and one-way power flow configuration. The case with pure sinusoidal forcing (Figure 7.7) shows that the one-way configuration experiences a decrease of performance of nearly 26% with respect to the two-way PTO. This loss increases to 50% for the case of irregular seastate (Figure 7.8). This is mainly due to the short-period waves of the sea spectrum, which cause frequent changes of sign of the device velocity. When this happens, the machinery force has to work to keep the WEC device still (zero velocity) against the wavefield, in order to prevent the power flow from inverting direction. This effect is mitigated in case of irregular waves with higher dominant wave period T_p, since the occurrence of short-period waves is greatly reduced. On the opposite



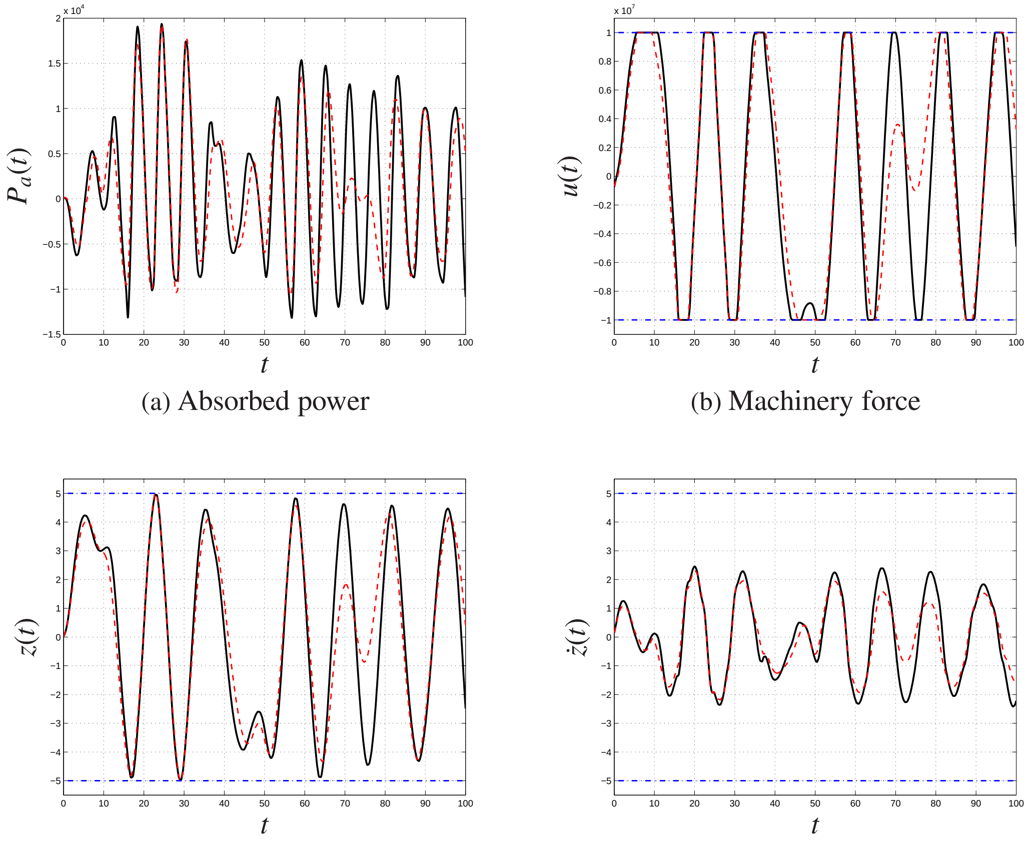
(a) Absorbed power

(b) Machinery force

(c) WEC1 buoy position

(d) WEC1 buoy velocity

Figure 7.4: Comparison between the power take-off of a one-body WEC device leveraging linear (black solid line) and nonlinear (red dashed line) MPC. A sinusoidal wave with a peak-to-trough amplitude of 3 m and period of 10 s is considered. The average absorbed power is $9.75e2\,kW$ with LMPC and $1.74e3\,kW$ with NMPC.



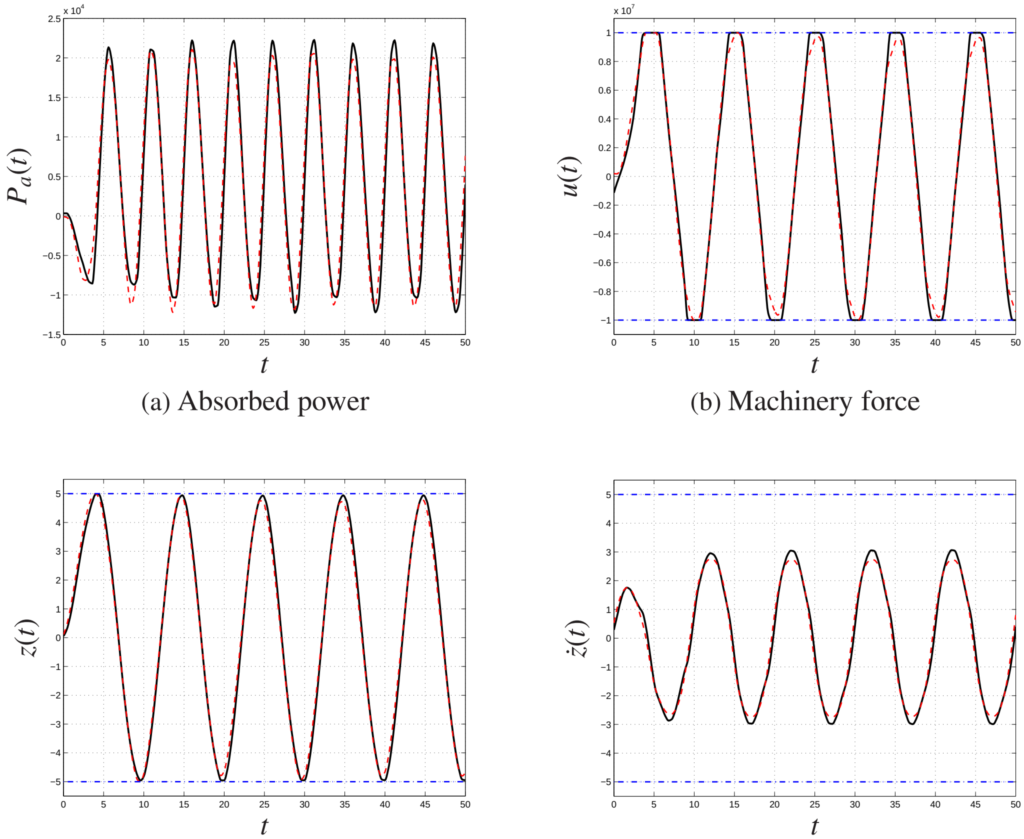
(a) Absorbed power

(b) Machinery force

(c) WEC1 buoy position

(d) WEC1 buoy velocity

Figure 7.5: Comparison between the power take-off of a one-body WEC device leveraging linear (black solid line) and nonlinear (red dashed line) MPC. A realistic seastate with $H_{1/3} = 3\,m$ and $T_p = 10\,s$ is considered. The average absorbed power is $1.09e3\,kW$ with LMPC and $1.34e3\,kW$ with NMPC.



(a) Absorbed power

(b) Machinery force

(c) WEC1 buoy position

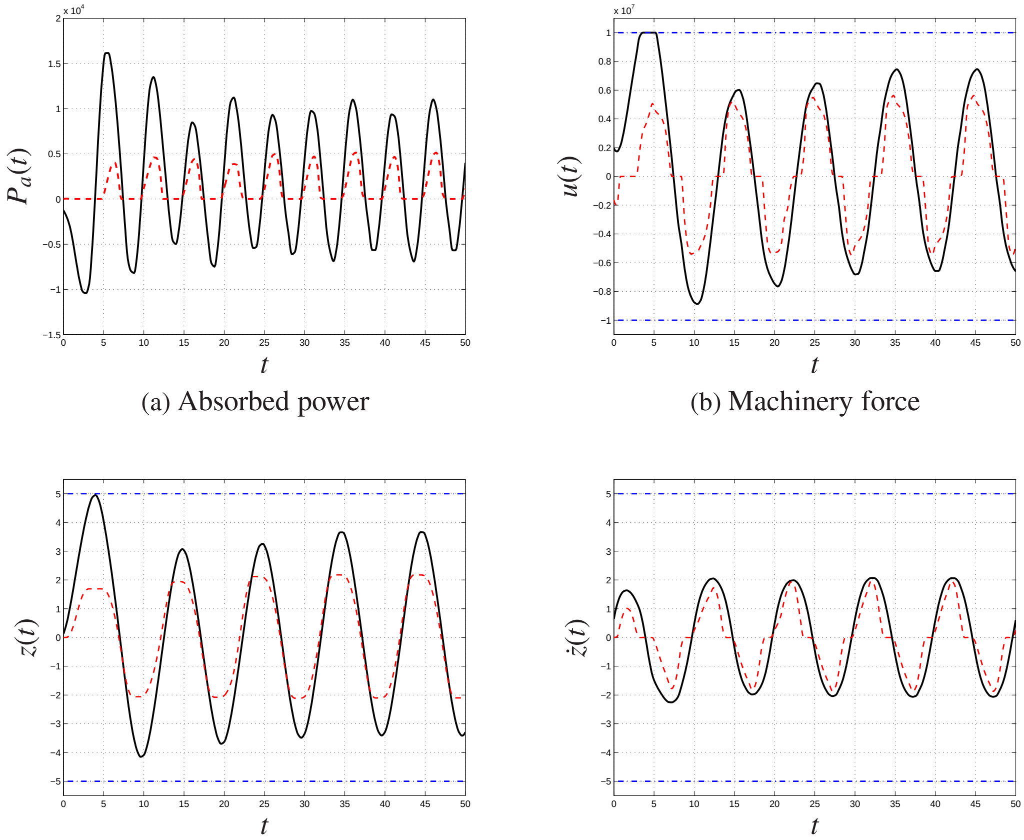
(d) WEC1 buoy velocity

Figure 7.6: Comparison between the power take-off of a one-body WEC device leveraging linear (black solid line) and nonlinear (red dashed line) MPC. A sinusoidal wave with a peak-to-trough amplitude of $5\,m$ and period of $10\,s$ is considered. The average absorbed power is $3.76e3\,kW$ with LMPC and $3.90e3\,kW$ with NMPC.

side, a lower period T_p would result in an even higher performance gap between two-way and one-way power flow configuration.

A comparison of the power take-off associated to the one-body and two-body WEC topologies presented in Section 7.2 is shown in Figures 7.9 and 7.10. In particular, in case of pure sinusoidal forcing (Figure 7.9), the average power take-off of the two-body configuration is 38% less than the average power take-off achieved with the one-body configuration. This is mainly due to the parasite motion of the immersed reaction plate, also characterized by a higher drag coefficient with respect to the WEC buoy. As a result, the machinery force in the two-body WEC is kept lower than in the one-body case, despite the relative velocity between buoy and reaction plate being higher than the buoy velocity of the one-body WEC. Similar considerations can be made for the case of irregular waves (Figure 7.10), even if in this case the performance gap is only 16%. This is motivated by the fact that the larger oscillations of the one-body device are limited by machinery constraints, thus allowing the two-body topology to achieve comparable performances. This result suggests that the the two-body topology should be preferred over the one-body configuration for those sea condition in which the wave energy converter is forced to work near the actuator saturation limits, since the two-body configuration is structurally easier to realize, with respect to the more ideal one-body model.

Finally, Figures 7.11 and 7.12 shows the power take-off results for the two-body WEC with one-way and two-way power flow. As compared to the one-body topology, the one-way power flow constraint leads to a slightly higher decrease in performance. In particular, a loss of average absorbed power of approximately 36% is experienced in the pure sinusoidal case (Figure 7.11), while for irregular waves this loss is increased up to nearly 55%. This is justified by the combined detrimental effects of short-period waves



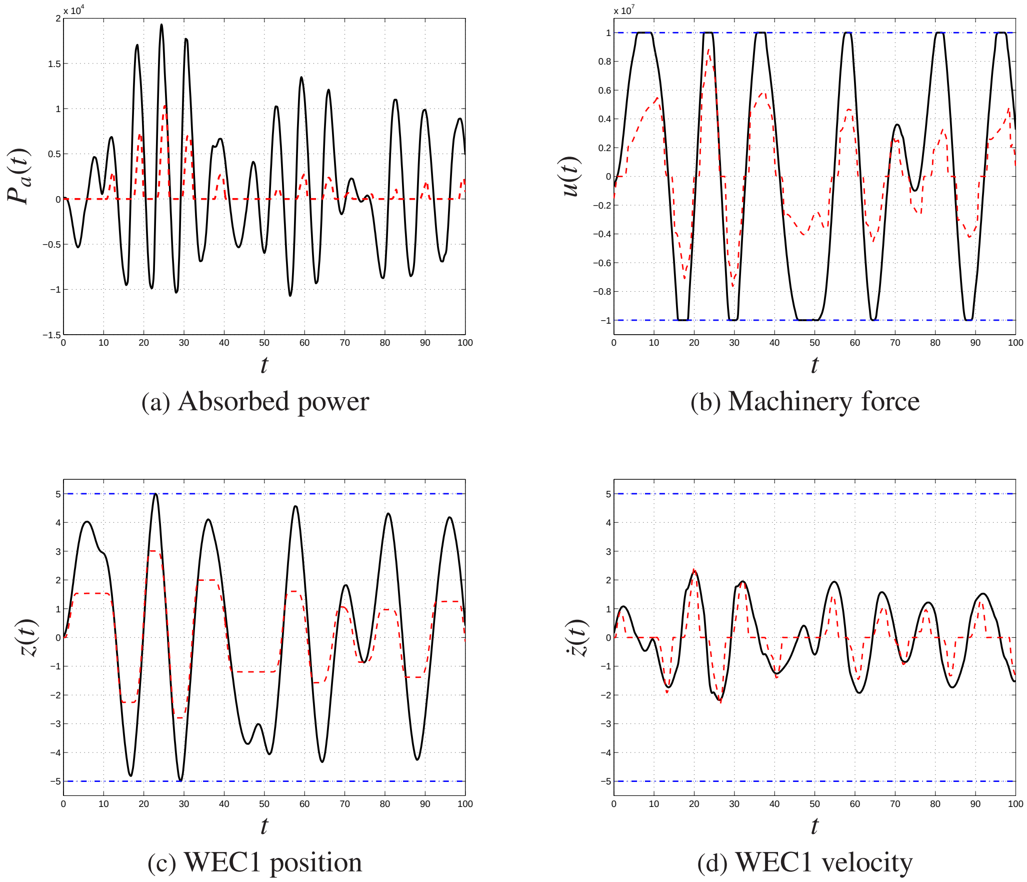
(a) Absorbed power

(b) Machinery force

(c) WEC1 position

(d) WEC1 velocity

Figure 7.7: Comparison between the power take-off of a one-body WEC device leveraging NMPC with two-way (black solid line) and one-way (red dashed line) power flow. A sinusoidal wave with a peak-to-trough amplitude of $3\,m$ and period of $10\,s$ is considered. The average absorbed power is $1.74e3\,kW$ with two-way power flow and $1.28e3\,kW$ with one-way power flow.



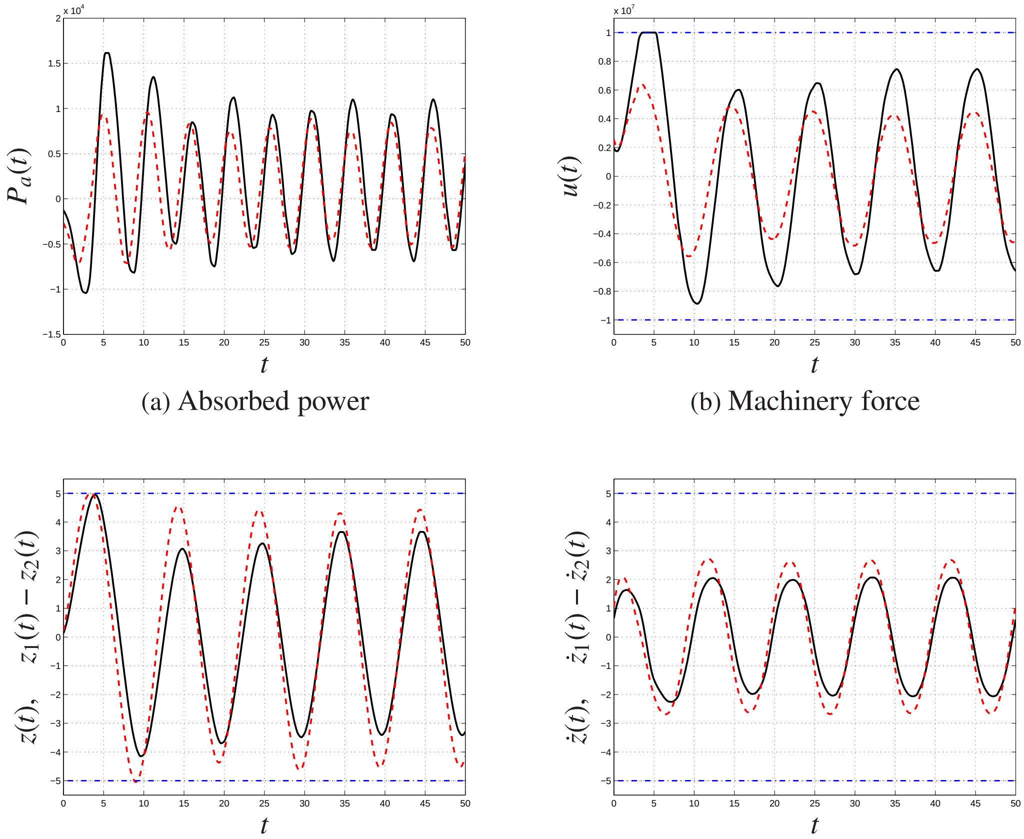
(a) Absorbed power

(b) Machinery force

(c) WEC1 position

(d) WEC1 velocity

Figure 7.8: Comparison between the power take-off of a one-body WEC device leveraging NMPC with two-way (black solid line) and one-way (red dashed line) power flow. A realistic seastate with $H_{1/3} = 3\,m$ and $T_p = 10\,s$ is considered. The average absorbed power is $1.34e3\,kW$ with two-way power flow and $6.66e2\,kW$ with one-way power flow.



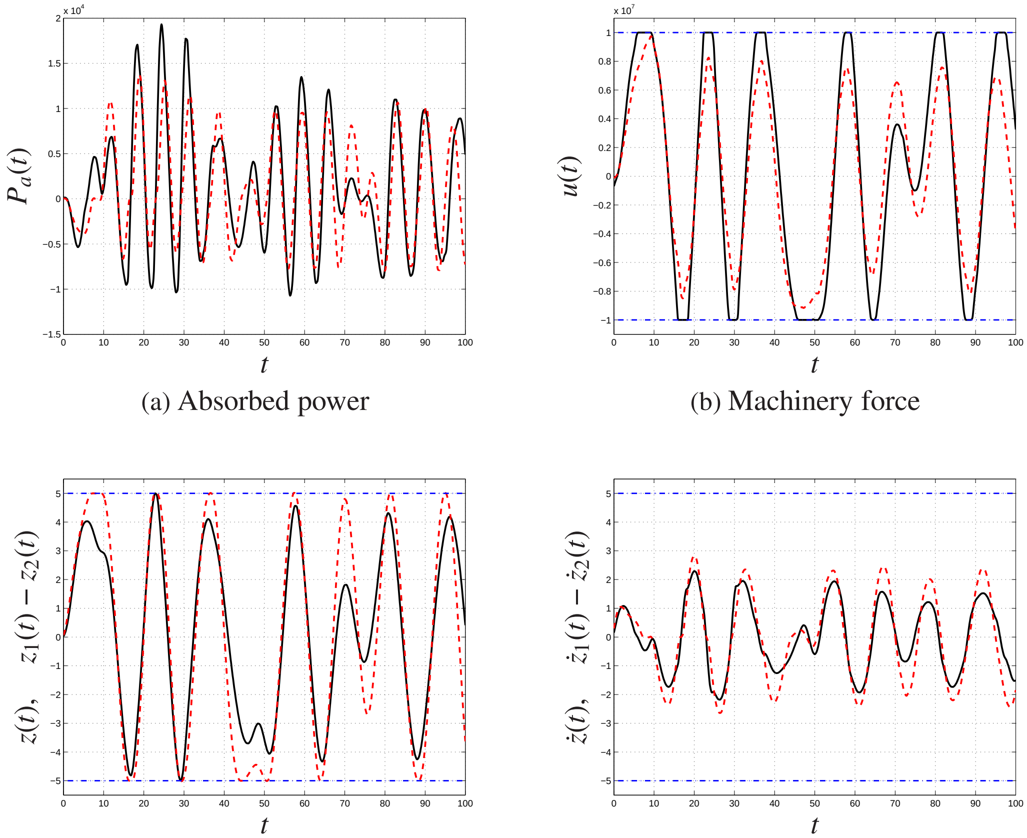
(a) Absorbed power

(b) Machinery force

(c) WEC1 and WEC2 buoy position

(d) WEC1 and WEC2 buoy velocity

Figure 7.9: Comparison between the power take-off of a one-body (black solid line) and two-body (red dashed line) WEC device leveraging NMPC. A sinusoidal wave with a peak-to-trough amplitude of $3\,m$ and period of $10\,s$ is considered. The average absorbed power is $1.74e3\,kW$ for the one-body WEC and $1.08e3\,kW$ for the two-body WEC.



(a) Absorbed power

(b) Machinery force

(c) WEC1 and WEC2 buoy position

(d) WEC1 and WEC2 buoy velocity

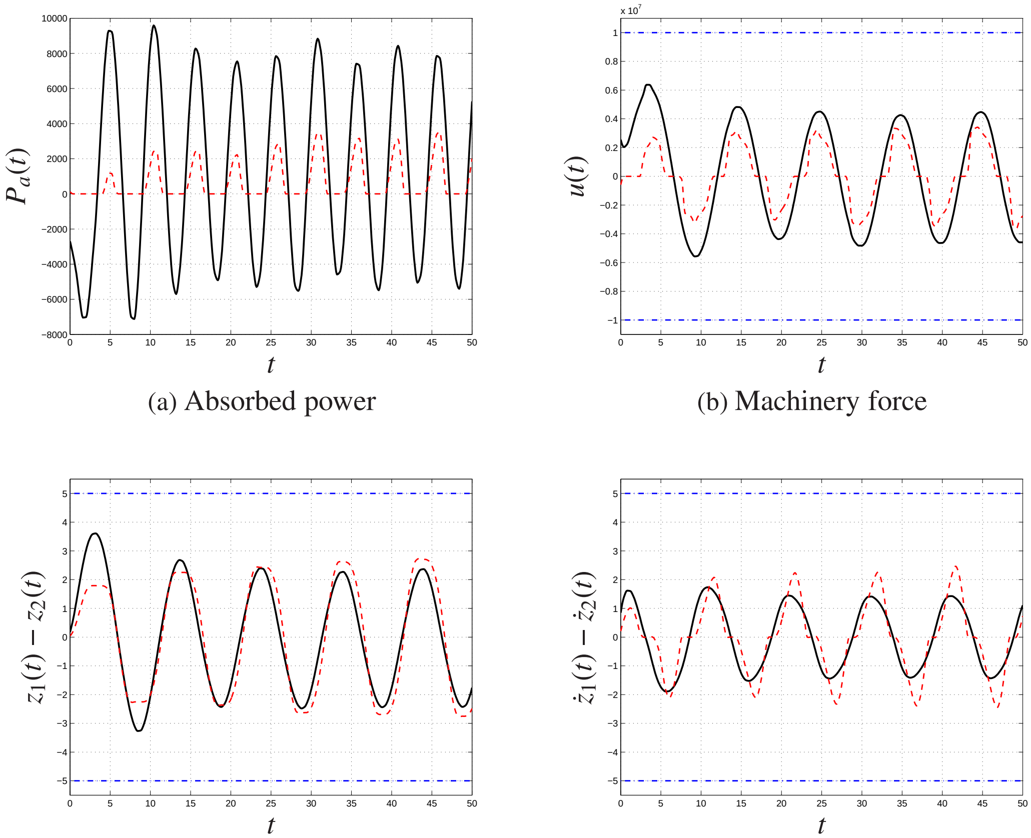
Figure 7.10: Comparison between the power take-off of a one-body (black solid line) and two-body (red dashed line) WEC device leveraging NMPC. A realistic seastate with $H_{1/3} = 3\,m$ and $T_p = 10\,s$ is considered. The average absorbed power is $1.34e3\,kW$ for the one-body WEC and $1.12e3\,kW$ for the two-body WEC.

and parasite motion of the reaction plate, which force the WEC buoy to have zero relative velocity with respect to the reaction plate during most of the operating time, thus limiting the period of actual power generation.

7.5 Conclusions

We have investigated the performance of NMPC applied to the optimization of the power take-off of a point-absorber wave energy converter. Two topologies have been considered: a one-body configuration, in which the device is constituted by a spar oscillating in heave, and a two-body model, in which the spar oscillates with respect to a reaction plate moored to the sea bed. Our NMPC implementation imposes the dynamics constraints leveraging a multiple shooting approach in which the state trajectories are calculated over each shooting interval. Two ways have been proposed in order to discretize such trajectories: a continuous-time and a discrete-time approach. The former leverages an explicit Runge-Kutta scheme for the integration of the continuous-time model, while the former uses a Taylor-Lie expansion to obtain the nonlinear discrete-time dynamics model, given a specified order of accuracy. Both approaches allows fast computation of the gradients needed in the SQP formulation. In particular, such computation can be carried out analytically through the application of recursive formulas, provided an expression for the gradients of the dynamics model is available. Another significant advantage associated to the employment of recursive formulas is that the storage required for the evaluation of the dynamics constraint function and gradient is minimal, so that the memory requirements for performing a single-step and a multi-step time integration of the state trajectory over each shooting interval are the same.

Compared to linear MPC, nonlinear MPC has proved to always lead to better perfor-



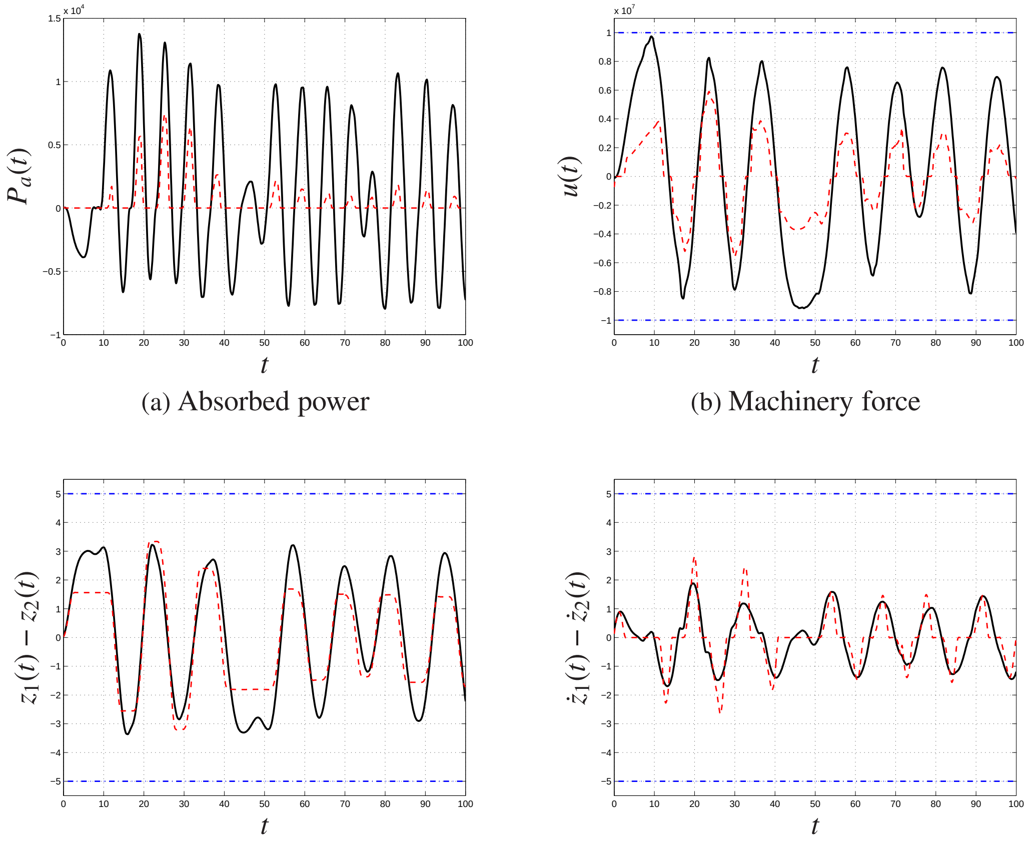
(a) Absorbed power

(b) Machinery force

(c) WEC2 buoy relative position

(d) WEC2 buoy relative velocity

Figure 7.11: Comparison between the power take-off of a two-body WEC device leveraging NMPC with two-way (black solid line) and one-way (red dashed line) power flow. A sinusoidal wave with a peak-to-trough amplitude of $3\,m$ and period of $10\,s$ is considered. The average absorbed power is $1.08e3\,kW$ with two-way power flow and $6.86e2\,kW$ with one-way power flow.



(a) Absorbed power

(b) Machinery force

(c) WEC2 buoy relative position

(d) WEC2 buoy relative velocity

Figure 7.12: Comparison between the power take-off of a two-body WEC device leveraging NMPC with two-way (black solid line) and one-way (red dashed line) power flow. A realistic seastate with $H_{1/3} = 3\,m$ and $T_p = 10\,s$ is considered. The average absorbed power is $1.12e3\,kW$ with two-way power flow and $5.07e2\,kW$ with one-way power flow.

mance, since the nonlinear effects affecting the WEC dynamics are accurately captured over the forecasting horizon. Besides, the nonlinear MPC formulation allows to directly impose the constraint of PTO working in generator-mode only, thus unable to return power, which represents a more simplified solution from a design point of view. Results have shown that the one-way power flow constraint leads to substantial performance decrease in both the one-body and two-body WEC configuration. In particular, the effect is more noticeable in the case of irregular waves, due to the more frequent changes of sign in the device velocity, owing to the presence of short-period waves in the sea spectrum. Moreover, the two-body topology has shown to provide lower levels of average energy absorption with respect to the one-body configuration. However, the presence of the immersed reaction plate has the beneficial effect of reducing the spar oscillation, which reflects into a significant reduction of the performance gap between the two topologies when the device operates close to machinery constraints.

The NMPC formulation here described is extremely flexible and it allows the optimization of nonquadratic cost functions, nonlinear dynamics, and other nonlinear constraints. However, the current approach does not handle the case of integer-valued control inputs, which would require the implementation of a nonlinear mixed-integer programming solver. Finally, the simulations presented in this chapter have been conducted under the ideal assumption of complete knowledge of the future wavefield over the defined control horizon. The application of NMPC for the optimization of a WEC device leveraging forecast data is not discussed here and is object of future work.

Acknowledgements

This chapter contains work previously published in:

- D. Cavaglieri, T.R. Bewley, A. Karthikeyan, M. Previsic, “Nonlinear Model Predictive Control of a point absorber wave energy converter", *Submitted to IEEE Transactions on Sustainable Energy*, 2016

Chapter 8

Conclusions and future work

In this thesis we present new numerical schemes for Computational Fluid Dynamics, forecast and control. In particular, our work includes new time stepping schemes for the efficient integration of high-dimensional systems, new algorithms for the relaxation step in the multigrid solution of large elliptic systems, an Ensemble Kalman Filter forecasting algorithm for short-term ocean wave prediction, and a new analytic approach to the discretization of state trajectories in direct multiple shooting Nonlinear Model Predictive Control, with application to power optimization of wave energy converters.

First, we introduce new IMEXRK schemes for time discretization of high-dimensional PDEs. Such schemes work best when applied to the time integration of discretized PDE systems with a RHS which can be divided into a linear stiff component and a nonlinear nonstiff component. This is the case of the Navier-Stokes Equations, for example, although several other fluid dynamics models belong to this category. Compared to other IMEXRK schemes available in the literature, ours offer comparable or better accuracy and stability properties with significantly reduced memory storage requirement. According to their numerical implementation, these new schemes can be divided into two categories:

the first one can be seen as an extension of low-storage explicit Runge-Kutta scheme to IMEXRK schemes, while the second is an improvement upon the low-storage incremental IMEXRK schemes CN/RKW3 and the scheme in [1], the only two other schemes of this kind developed so far. In comparison, our new schemes have improved accuracy and stability properties, with same or slightly increased storage requirement.

Afterward, we describe two new smoothing schemes, i.e. *tweed* and *box* relaxation, for the multigrid solution of large linear systems arising from the spatial discretization of elliptic PDEs. In particular, these schemes best perform when applied to the iterative solution of elliptic PDE problems defined over stretched structured grids and discretized using 3-point-stencil discrete derivative operators. Compared to the state-of-the-art approach, which involves using alternating-direction zebra relaxation for the smoothing step, together with full weighting for restriction and bilinear interpolation for prolongation, our schemes guarantee comparable convergence results with significantly reduced computational cost. This is achieved through the development of *ad hoc* modifications of the Thomas algorithm employed for the factorization of tridiagonal systems. Beside their application within the multigrid framework, these schemes can also be implemented for the efficient solution of the linear system arising from the spatial discretization of one-dimensional PDEs defined over wireframe structures, since the factorization time scales linearly with the dimension of the system matrix.

Then, we present a new forecasting scheme for the short-term prediction of ocean wave elevation. This scheme uses Ensemble Kalman Filter in order to assimilate synthetic measurement data provided by wave radar and arrays of wave monitoring buoys. Within this formulation, the initial ensemble wavefields are generated from a known spectral distribution. Time propagation is then carried out by leveraging a nonlinear pseudospectral model, featuring one of our new low-storage third-order IMEXRK schemes for time discretization. Results have shown that accurate wave forecasting is possible up to thirty seconds into the future, provided a relatively large number of ensemble members is used.

Finally, we introduce a new approach for the analytic computation of discretized state trajectories and associated state and control input gradients for the solution of the nonlinear optimization problem arising within the Nonlinear Model Predictive Control formulation leveraging a direct multiple shooting approach. The resulting algorithm is then applied to the power take-off optimization of a point absorber wave energy converter, subject to nonlinear constraints.

As future work, a selection of our new third-order incremental IMEXRK schemes is to be tested in the DNS simulation of a turbulent channel flow at high Reynolds numbers. Results will allow to ultimately assess the performance of the new proposed schemes against the most popular approach, which instead relies on **CN/RKW3** for time discretization. Furthermore, a multigrid algorithm leveraging the new smoothers can be implemented to speed up the solution of the pressure Poisson equation arising in the simulation of duct and cavity flows defined over stretched structured grids.

Bibliography

[1] P. Spalart, R. Moser, M. Rogers, Spectral methods for the Navier-Stokes equations with one infinite and two periodic directions, J. Comput. Phys. 96.2 (1991) 297–324.

[2] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-stokes Equations, J. Comput. Phys. 59 (1985) 308–323.

[3] J. Kim, P. Moin, B. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, J. Fluid Mech. 177 (1987) 133–166.

[4] H. Le, P. Moin, An improvement of fractional step methods for the incompressible Navier-Stokes equations, J. Comput. Phys. 92 (1991) 369–379.

[5] A. Wray, Minimal-storage time advancement schemes for spectral methods.

[6] K. Akselvoll, M. P., Large eddy simulation of turbulent confined coannular jets and turbulent flow over a backward facing step, Tech. rep., Rep. TF-63. Thermosciences Division, Dept. of Mech. Eng., Stanford University (1995).

[7] C. Kennedy, M. Carpenter, R. Lewis, Additive Runge-Kutta schemes for convection-diffusion-reaction equations, Appl. Num. Math. 44 (2003) 139–181.

[8] C. Kennedy, M. Carpenter, R. Lewis, Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations, Appl. Num. Math. 35 (2000) 177–219.

[9] U. Ascher, S. Ruuth, R. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Appl. Num. Math. 25 (2) (1997) 151–167.

[10] M. Calvo, J. de Frutos, J. Novo, Linearly implicit Runge-Kutta methods for advection-reaction-diffusion equations, Appl. Num. Math. 37 (4) (2001) 535–549.

[11] L. Shampine, Implementation of implicit formulas for the solution of ODEs, SIAM J. Sci. Comput. 1 (1) (1980) 103–118.

[12] J. Butcher, Numerical methods for ordinary differential equations, Wiley, 2008.

[13] R. LeVeque, Finite volume methods for hyperbolic problems, Cambridge University Press, 2002.

[14] C. Shu, Total-variation-diminishing time discretizations, SIAM J. Sci. Statist. Comput. 9 (1988) 1073–1084.

[15] C. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.

[16] S. Gottlieb, C. Shu, E. Tadmor, Strong-stability-preserving high order time discretization methods, SIAM Review 43 (2001) 89–112.

[17] L. Pareschi, G. Russo, Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation, Journal of Scientific Computing 25 (2003) 129–155.

[18] J. Williamson, Low-storage Runge-Kutta schemes, J. Comput. Phys. 35 (1980) 48–56.

[19] P. van der Houwen, Explicit Runge-Kutta formulas with increased stability boundaries, Numerische Mathematik 20 (1972) 149–164.

[20] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic problems, Springer-Verlag, Berlin, 1996.

[21] E. Hairer, C. Lubich, M. andRoche, Error of Runge-Kutta methods for stiff problems studied via differential algebraic equations, BIT 28 (3) (1988) 678–700.

[22] S. Wolfram, The Mathematica Book, Fifth Edition, Cambridge University Press, Cambridge, 2003.

[23] M. Carpenter, C. Kennedy, H. Bijl, V. S.A., V. Vatsa, Fourth-order Runge-Kutta schemes for fluid mechanics applications, Journal of Scientific Computing 25.

[24] R. Courant, K. Friedrichs, H. Lewy, On the partial difference equations of mathematical physics, IBM J. of Res. and Dev. 11.2 (1967) 215–234.

[25] E. Hofer, A partially implicit method for large stiff systems of ODEs with only few equations introducing small time-constants, SIAM J. Num. An. 13.5 (1976) 645–663.

[26] D. Cavaglieri, T. Bewley, Low-storage implicit/explicit Runge-Kutta schemes for the simulation of stiff high-dimensional ODE systems, J. Comput. Phys. 286 (2015) 172–193.

[27] C. Kennedy, M. Carpenter, Additive Runge-Kutta schemes for convection-diffusion-reaction equations, Tech. rep., NASA Tech. Rep. (2001).

[28] A. Araújo, A. Murua, J. Sanz-Serna, Symplectic methods based on decompositions, SIAM J. Num. An. 34.5 (1997) 1926–1947.

[29] J. Butcher, The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods, Wiley-Interscience, 1987.

[30] E. Hairer, S. Norsett, G. Wanner, Solving Ordinary Differential equations I, Nonstiff problems, Springer-Verlag, Berlin, 1993.

[31] J. Dormand, P. Prince, A family of embedded Runge-Kutta formulae, J. Comput. & Appl. Math. 6.1 (1980) 19–26.

[32] J. Perot, An analysis of the fractional step method, Journal of Computational Physics 108 (1) (1993) 51–58.

[33] A. Brandt, O. Livne, Multigrid techniques: 1984 guide with applications to fluid dynamics, Vol. 67, SIAM, 2011.

[34] A. Brandt, Multigrid solvers on parallel computers.

[35] U. Trottenberg, C. Oosterlee, A. Schuller, Multigrid, Academic press, 2000.

[36] P. Luchini, A. D'Alascio, Multigrid pressure correction techniques for the computation of quasi-incompressible internal flows, International journal for numerical methods in fluids 18 (5) (1994) 489–507.

[37] D. Cavaglieri, T. Bewley, Extensions of the Thomas algorithm for the efficient solution of elliptic PDEs on wireframe structures, Submitted to Journal of Computational Physics.

[38] J. H. Ahlberg, E. Nilson, J. Walsh, The theory of splines and their applications, Mathematics in Science and Engineering, New York: Academic Press, 1967 1.

[39] A. Wazwaz, Partial Differential Equations, CRC Press, 2002.

[40] R. Skelton, M. de Oliveira, Tensegrity systems, Vol. 1, Springer, 2009.

[41] D. Cavaglieri, A. Mashayek, T. Bewley, Tweed and box relaxation: improved smoothing algorithms for multigrid solution of PDEs on stretched structured grids, Submitted to Journal of Computational Physics.

[42] J. Jackson, Llewellyn Hilleth Thomas 1903-1992. Biographical Memoirs, National Academy of Sciences, Washington, D.C., 2009.

[43] L. Thomas, Elliptic problems in linear differential equations over a network, Tech. rep., Watson Sci. Comput. Lab Report, Columbia University, New York (1949).

[44] E. Cuthill, J. McKee, Reducing the bandwidth of sparse symmetric matrices, in: Proceedings of the 1969 24th national conference, ACM, 1969, pp. 157–172.

[45] A. George, J. Liu, Computer solution of large sparse positive definite systems, Prentice Hall Professional Technical Reference, 1981.

[46] W. Liu, A. Sherman, Comparative analysis of the Cuthill-McKee and the reverse Cuthill-McKee ordering algorithms for sparse matrices, SIAM Journal on Numerical Analysis 13 (2) (1976) 198–213.

[47] G. Golub, C. Van Loan, Matrix computations, Vol. 3, JHU Press, 2012.

[48] J. Falnes, Ocean waves and oscillating systems: linear interactions including wave-energy extraction, Cambridge university press, 2002.

[49] J. Falnes, On non-causal impulse response functions related to propagating water waves, Applied Ocean Research 17 (6) (1995) 379–389.

[50] F. Fusco, J. Ringwood, A study of the prediction requirements in real-time control of wave energy converters, Sustainable Energy, IEEE Transactions on 3 (1) (2012) 176–184.

[51] J. Hals, J. Falnes, T. Moan, Constrained optimal control of a heaving buoy wave-energy converter, Journal of Offshore Mechanics and Arctic Engineering 133 (1) (2011) 011401.

[52] M. Belmont, J. Horwood, R. Thurley, J. Baker, Filters for linear sea-wave prediction, Ocean Engineering 33 (17) (2006) 2332–2351.

[53] S. Aragh, O. Nwogu, Variation assimilating of synthetic radar data into a pseudo-spectral wave model, Journal of Coastal Research (2008) 235–244.

[54] F. Fusco, J. Ringwood, Short-term wave forecasting for real-time control of wave energy converters, Sustainable Energy, IEEE Transactions on 1 (2) (2010) 99–106.

[55] L. Brandt, P. Schlatter, D. Henningson, Transition in boundary layers subject to free-stream turbulence, Journal of Fluid Mechanics 517 (2004) 167–198.

[56] V. Zakharov, Stability of periodic waves of finite amplitude on the surface of a deep fluid, Journal of Applied Mechanics and Technical Physics 9 (2) (1968) 190–194.

[57] D. Dommermuth, D. Yue, A high-order spectral method for the study of nonlinear gravity waves, Journal of Fluid Mechanics 184 (1987) 267–288.

[58] B. West, K. Brueckner, R. Janda, D. Milder, R. Milton, A new numerical method for surface hydrodynamics, Journal of Geophysical Research: Oceans (1978–2012) 92 (C11) (1987) 11803–11824.

[59] Y. Matsuno, Nonlinear evolutions of surface gravity waves on fluid of finite depth, Physical review letters 69 (4) (1992) 609.

[60] M. Kanevsky, Radar imaging of the ocean waves, Elsevier, 2008.

[61] G. Evensen, The ensemble Kalman filter: Theoretical formulation and practical implementation, Ocean dynamics 53 (4) (2003) 343–367.

[62] D. Cavaglieri, T. Bewley, M. Previsic, Model Predictive Control leveraging Ensemble Kalman Forecasting for optimal power take-off in wave energy conversion systems, ACC 2015 Proceedings.

[63] K. Hasselmann, T. Barnett, E. Bouws, H. Carlson, D. Cartwright, K. Enke, J. Ewing, H. Gienapp, D. Hasselmann, P. Kruseman, Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP), Tech. rep., Deutches Hydrographisches Institut (1973).

[64] O. Faltinsen, Sea loads on ships and offshore structures, Vol. 1, Cambridge university press, 1993.

[65] H. Tolman, User manual and system documentation of WAVEWATCH iii version 3.14, Technical note, MMAB Contribution 276.

[66] R. Bitmead, A. Tsoi, P. Parker, A Kalman filtering approach to short-time Fourier analysis, IEEE Transactions on Acoustics, Speech and Signal Processing 34 (6) (1986) 1493–1501.

[67] D. Simon, Optimal state estimation: Kalman, H infinity, and nonlinear approaches, John Wiley & Sons, 2006.

[68] J. Anderson, An adaptive covariance inflation error correction algorithm for ensemble filters, Tellus 59 (2) (2007) 210–224.

[69] M. Frigo, S. Johnson, FFTW: An adaptive software architecture for the FFT, Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on 3 (1998) 1381–1384.

[70] F. Antonio, Wave energy utilization: a review of the technologies, Renewable and sustainable energy reviews 14 (3) (2010) 899–918.

[71] A. Babarit, J. Hals, M. Muliawan, A. Kurniawan, T. Moan, J. Krokstad, Numerical benchmarking study of a selection of wave energy converters, Renewable Energy 41 (2012) 44–63.

[72] K. Rhinefrank, A. Schacher, J. Prudell, C. Stillinger, D. Naviaux, T. Brekken, A. von Jouanne, D. Newborn, S. Yim, D. Cox, High resolution wave tank testing of scaled wave energy devices, in: ASME 2010 29th International Conference on Ocean, Offshore and Arctic Engineering, American Society of Mechanical Engineers, 2010, pp. 505–509.

[73] R. Henderson, Design, simulation, and testing of a novel hydraulic power take-off system for the Pelamis wave energy converter, Renewable energy 31 (2) (2006) 271–283.

[74] J. Hals, J. Falnes, T. Moan, A comparison of selected strategies for adaptive control of wave energy converters, Journal of Offshore Mechanics and Arctic Engineering 133 (3) (2011) 031101.

[75] M. Richter, M. Magaña, O. Sawodny, T. Brekken, Nonlinear model predictive control of a point absorber wave energy converter, Sustainable Energy, IEEE Transactions on 4 (1) (2013) 118–126.

[76] N. Tom, R. Yeung, Nonlinear Model Predictive Control applied to a generic ocean-wave energy extractor, Journal of Offshore Mechanics and Arctic Engineering 136 (4) (2014) 041901.

[77] D. Cavaglieri, T. Bewley, M. Previsic, Short-term ensemble ocean wave forecasting, Submitted.

[78] W. Cummins, The impulse response function and ship motions, Tech. rep., DTIC Document (1962).

[79] Z. Yu, J. Falnes, State-space modelling of a vertical cylinder in heave, Applied Ocean Research 17 (5) (1995) 265–275.

[80] K. Ruehl, T. Brekken, B. Bosma, R. Paasch, Large-scale ocean wave energy plant modeling, in: Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES), 2010 IEEE Conference on, IEEE, 2010, pp. 379–386.

[81] D. Leineweber, I. Bauer, H. Bock, J. Schlöder, An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part 1: theoretical aspects, Computers & Chemical Engineering 27 (2) (2003) 157–166.

[82] D. Leineweber, I. Bauer, H. Bock, J. Schlöder, An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization: Part ii: Software aspects and applications, Computers & chemical engineering 27 (2) (2003) 167–174.

[83] J. Nocedal, S. Wright, Numerical optimization, Springer Science & Business Media, 2006.

[84] C. Kirches, L. Wirsching, H. Bock, J. Schlöder, Efficient direct multiple shooting for nonlinear model predictive control on long horizons, Journal of Process Control 22 (3) (2012) 540–550.

[85] N. Kazantzis, K. Chong, J. Park, A. Parlos, Control-relevant discretization of nonlinear systems with time-delay using Taylor-Lie series, in: American Control Conference, 2003. Proceedings of the 2003, Vol. 1, IEEE, 2003, pp. 149–154.