General-domain compressible Navier-Stokes solvers exhibiting quasi-unconditional stability and high-order accuracy in space and time

Thesis by

Max Cubillos-Moraga

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2015

(Defended May 22, 2015)

To my brother Yves

Acknowledgements

It's been a long journey, and of all the things to come out of it, I think the most rewarding and lasting part of my time here will be the people I have met and the bonds of friendship that were forged.

I must first thank my advisor, Professor Oscar Bruno, who goes above and beyond all expectation in his dedication to his students. The many lessons I have learned through his insistence on excellent, thorough work will stay with me wherever I go. One of the unexpected joys of preparing this thesis was the conversations we had over many hours of Skype calls. I will definitely miss that big, full laugh.

I would also like to thank the other members of my thesis committee, Professors Guillaume Blanquart, Tim Colonius, and Houman Owhadi, for their valuable comments and suggestions, and for what turned out to be a very lively and enjoyable thesis defense.

A special thanks goes to Edwin Jimenez: besides all the interesting conversations we have had over many lunches and coffee breaks, his cleanly written, readable code and his unique insights on mathematics were a huge inspiration to me. I am grateful for all the help he selflessly extended to me on the coding side of things, and for the opportunity I had to collaborate with him on the vision we both shared for general and flexible numerical solvers.

I don't think I could have made it this far without the friends I made during my time here. To Joshua Reyna, my brother from another mother, the most unique and amazing person I have ever met: you are a true warrior in both mind, body, and heart, and whether you know it or not, I look up to you more than anyone. To Mary

Fu, a true friend who has never held back her generosity: your genuine spirit was a refreshing breeze in the desert of LA. To Morgan Smith: you have kept me honest and true to myself over the years, reminding me to do the right thing especially when it's most difficult. To Melodie Kao: the most important lessons I have learned from you, more valuable than the sum of my graduate studies—to be a better version of myself. There are many others of course: Gregory Atrian, Jen Bae, Sharath Rajashekar, Lauren Kendrick, Cameron Voloshin—thank you all for being a part of my life and, in one way or another, making my time at Caltech a joy.

Of course, I could not have made it this far without my family. I'd like to thank my sister Giselle for putting up with me longer than just about everyone. And finally, to my parents: Even though these few words will do little justice to the immense gratitude I feel towards you, thank you for everything.

Abstract

This thesis presents a new class of solvers for the subsonic compressible Navier-Stokes equations in general two- and three-dimensional spatial domains. The proposed methodology incorporates: 1) A novel linear-cost implicit solver based on use of higher-order backward differentiation formulae (BDF) and the alternating direction implicit approach (ADI); 2) A fast explicit solver; 3) Dispersionless spectral spatial discretizations; and 4) A domain decomposition strategy that negotiates the interactions between the implicit and explicit domains. In particular, the implicit methodology is *quasi-unconditionally stable* (it does not suffer from CFL constraints for adequately resolved flows), and it can deliver *orders of time accuracy between two and six in the presence of general boundary conditions*. In fact this thesis presents, for the first time in the literature, high-order time-convergence curves for Navier-Stokes solvers based on the ADI strategy—previous ADI solvers for the Navier-Stokes equations have not demonstrated orders of temporal accuracy higher than one. An extended discussion is presented in this thesis which places on a solid theoretical basis the observed quasi-unconditional stability of the s order methods with $2 \leq s \leq 6$. The performance of the proposed solvers is favorable. For example, a two-dimensional rough-surface configuration including boundary layer effects at Reynolds number 10^6 and Mach number $\mathrm{Ma} = 0.85$ (with a well-resolved boundary layer, run up to a sufficiently long time that single vortices travel the entire spatial extent L of the domain, and with spatial mesh sizes near the wall of the order of $10^{-5} \cdot L$) was successfully tackled in a relatively short ($\sim$ thirty-hour) single-core run; for such discretizations an explicit solver would require truly prohibitive computing times. As demonstrated

via a variety of numerical experiments in two- and three-dimensions, further, the proposed multi-domain parallel implicit-explicit implementations exhibit high-order convergence in space and time, useful stability properties, limited dispersion, and high parallel efficiency.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The direct numerical simulation of fluid flow at high Reynolds numbers presents a number of significant challenges—including the presence of structures such as boundary layers, eddies, vortices and turbulence, whose accurate spatial discretization requires use of fine spatial meshes. CFD (computational fluid dynamics) simulation of such flows by means of explicit solvers is highly demanding, even on massively parallel super computers, in view of the severe restrictions on time steps required for stability: the time step must scale like the square of the spatial mesh size. Classical implicit solvers do not suffer from such time step restrictions but they do require solution of large systems of equations at each time step, and they can therefore be extremely expensive as well.

The celebrated Beam and Warming method [6, 8] provides one of the most attractive alternatives to explicit and classical implicit algorithms. Based on the alternating direction implicit method [69] (ADI), the Beam and Warming scheme enables stable solution of the compressible Navier-Stokes equations in times that grow only linearly with the size of the underlying discretization, and without recourse to either nonlinear iterative solvers or solutions of large linear systems at each time step. As discussed in detail in the introductory portion of Chapter 2, however, previous work in the context of the Beam and Warming method has not demonstrated accuracies beyond the first order of temporal accuracy.

Nevertheless, high-order time accuracy may be crucial in the treatment of long-time simulations or highly-inhomogeneous flows—for which the dispersion inherent in low-order approaches would make it necessary to use inordinately small time-steps.

This thesis presents, in particular, extensions of the ADI methodology based on the backward differentiation formulae (BDF) that exhibit *orders of time accuracy between two and six*, and which are *quasi-unconditionally stable*, in a sense that is made clear in Section 2.2—which essentially amounts to true unconditional stability within certain regions in the space of discretization parameters. Further, full unconditional stability of the second order scheme is established in the context of the convection and parabolic linear equations. An extended discussion is presented in this thesis which places on a solid theoretical basis the observed quasi-unconditional stability of the s order methods with $2 \leq s \leq 6$. In fact this thesis presents, for the first time in the literature, high-order time-convergence curves for Navier-Stokes solvers based on the ADI strategy.

The proposed methodology employs the BDF schemes (which are known for their robust stability properties) together with a quasiliner-like formulation with high-order extrapolation for nonlinear components (to produce a linear high-order time-accurate method) and the Douglas-Gunn splitting (an ADI strategy that greatly simplifies boundary condition treatment while retaining the order of time-accuracy of the solver). The performance of the proposed solvers is favorable: for example, a two-dimensional rough-surface configuration including boundary layer effects at Reynolds number 10^6 and Mach number Ma $= 0.8$ (with a well-resolved boundary layer, run up to a sufficiently long time that single vortices travel the entire spatial extent L of the domain, and with spatial mesh sizes near the wall of the order of $10^{-5} \cdot L$) was successfully tackled in a relatively short ($\sim$ thirty-hour) single-core run; under similar circumstances an explicit solver would require truly prohibitive computing times. The highest order solvers can be greatly advantageous for problems involving long evolution times or solutions that oscillate rapidly in time; methods of lower order

may be more advantageous under other circumstances.

While the computational cost of the proposed BDF-ADI schemes mentioned above grows only linearly with the size of the spatial discretization, these schemes are significantly more expensive per time step than their explicit counterparts—such as the explicit Fourier Continuation solver [2] (FC) we use. Thus the strategy proposed in this thesis calls for use of multi-domain implicit-explicit solvers—implicit near boundaries and other regions where fine spatial discretizations are used (which might require extremely small time steps in an explicit solver), and explicit in regions in which the size of the spatial discretization does not entail significant CFL constraints. (The proposed multi-domain implicit-explicit schemes should not be confused with similarly named IMEX methods [4] which, e.g., in an advection-diffusion equation incorporate explicit treatment of the convective term and implicit treatment of the diffusive term.) A brief description of the Fourier continuation methodology and associated explicit solvers is presented below followed by an outline of the proposed multi-domain implicit-explicit strategy; complete descriptions and illustrations of these solvers follow as part of the main body of this thesis.

Most structured-grid solvers for Partial Differential Equations (PDE) are based on the use of finite differences (FD). These methods are intuitively attractive, they can be implemented easily, and they require limited cost per spatial discretization point. As is well known, however, reduction of the dispersion error inherent in FD methods requires either use of large numbers of points per wavelength, or use of higher-order methods which typically entail higher costs and restrictive CFL constraints [2, 3, 35]. Spectral methods are an attractive alternative in dealing with these challenges [10, 19, 47]. Unfortunately, polynomial spectral methods require clustering of points at the boundaries of the domain, resulting in severe time step restrictions for explicit methods. Classical Fourier methods, on the other hand, are only applicable to periodic problems—otherwise they suffer from the Gibbs phenomenon and first order spatial convergence in the interior of the domain (see, e.g., [10, Ch. 2.2]). The recently

introduced Fourier Continuation method (FC) provides spectral-like resolution in non-periodic contexts without recourse to use of fine meshes; we briefly discuss this methodology in what follows.

The FC method produces an interpolating Fourier series representation by relying on a "periodic extension" of a given function, that closely approximates it in the physical domain, but which is periodic on a slightly enlarged domain. In the context of explicit algorithms, following [2, 3, 35] the FC spatial discretizations are used in conjunction with the Adams-Bashforth (AB) method [58, Ch. 3.9] of orders two through four. As shown in Section 3.1.2 as well as in previous references [2, 3, 35, 64], the resulting FC time-domain solvers (whether explicit or implicit) do give rise to significantly improved dispersion properties, low computing costs, high accuracies and favorable spectral asymptotics in CFL constraints—as well as parallelization with perfect scaling. In particular, the explicit solver is significantly more accurate than other explicit methods for similar computing times, and significantly faster than other schemes for a given accuracy; cf. [2] and Section 3.1.2.

Unlike previous general Navier-Stokes solvers, all of the methods presented in this thesis, including the explicit, implicit, and multi-domain solvers mentioned above, enjoy *near spatial dispersionlessness* as well as *higher orders of accuracy in both space and time*. Such desirable characteristics are demonstrated, in particular, by means of implicit solutions in single domains as well as explicit and multi-domain implicit-explicit solutions *with non-trivial boundary conditions*—all of which include no-slip boundary conditions at walls, and, depending on the case under consideration, absorbing boundary conditions and inflow conditions. The proposed BDF-ADI solvers, further, enjoy both the properties of quasi-unconditional stability, dispersionlessness, and high-order accuracy in time. The multi-domain implicit-explicit solver, in turn, is highly effective: results of two-dimensional flow past a cylinder and three-dimensional flow past a sphere were produced with a significant cost savings over purely explicit or implicit solvers. These results also represent the first demonstrations of high-order

time-accuracy for any Navier-Stokes solver with an implicit component (let alone any hybrid solver) in flows of physical interest. In view of a variety of numerical examples presented in this thesis we suggest that the accuracy levels achieved by the proposed solvers for given spatial and temporal discretizations are unprecedented in the literature.

In the remainder of this chapter we provide a brief account of the background leading to the contributions in this thesis. The proposed BDF-ADI solvers are then introduced in Chapter 2, including the concept of quasi-unconditional stability as well as energy stability proofs for the second order schemes and spectral stability proofs for the higher-order BDF methods. The multi-domain implicit-explicit schemes are then presented in Chapter 3. Numerical results follow in Chapter 4, and concluding remarks, finally, are presented in Chapter 5.

1.1 The Navier-Stokes equations for a compressible gas

We consider the Navier-Stokes equations for a continuum fluid. Denoting by $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$ the material derivative, the Navier-Stokes system combines the equations describing conservation of mass,

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{u} = 0, \tag{1.1}$$

conservation of momentum,

$$\rho \frac{D\mathbf{u}}{Dt} + \nabla p = \nabla \cdot \boldsymbol{\sigma}, \tag{1.2}$$

and conservation of energy,

$$\rho \frac{De}{Dt} + p \nabla \cdot \mathbf{u} + \nabla \cdot q = \Phi, \tag{1.3}$$

where d denotes the spatial dimensions ($d = 2, 3$) and where, using integer-valued indices $i, j = 1, \ldots, d$, $\mathbf{u} = (u_i)$ denotes the velocity vector, and ρ, e, p, q, $\boldsymbol{\sigma} = (\sigma_{ij})$, and $\Phi = \sum_{ij} \sigma_{ij} \partial_{x_i} u_j$ denote density, specific internal energy, pressure, heat flux, deviatoric stress tensor, and viscous dissipation function, respectively. We narrow our consideration to the evolution of a subsonic compressible perfect gas satisfying the following assumptions:

1. The fluid is Newtonian, i.e., $\boldsymbol{\sigma} = \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T - \frac{2}{3} (\nabla \cdot \mathbf{u}) \mathbf{I} \right)$, where μ is the viscosity and $\mathbf{I}$ is the identity tensor.

2. The internal energy and temperature T satisfy the thermodynamic relation $e = c_v T$, where c_v is the specific heat at constant volume.

3. The pressure, density, and temperature are related by the equation of state for an ideal gas $p = \rho R T$, where R is the gas constant.

4. Fourier's law of heat conduction $q = -\kappa \nabla T$ holds, where κ is the thermal conductivity.

5. For simplicity, μ and κ are functions of temperature only.

With these assumptions, choosing a characteristic length L_0, velocity u_0, density ρ_0, temperature T_0, viscosity μ_0, and heat conductivity κ_0, and with a slight notational abuse by which the non-dimensional density, velocity, and temperature ρ/ρ_0, $\mathbf{u}/u_0$, and T/T_0 are denoted everywhere below in this thesis by the symbols ρ, $\mathbf{u}$, and T, respectively, the non-dimensional form of the Navier-Stokes equations

$$\rho_t + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{1.4a}$$

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\gamma \mathrm{Ma}^2} \frac{1}{\rho} \nabla (\rho T) = \frac{1}{\mathrm{Re}} \frac{1}{\rho} \nabla \cdot \boldsymbol{\sigma} \tag{1.4b}$$

$$T_t + \mathbf{u} \cdot \nabla T + (\gamma - 1) T \nabla \cdot \mathbf{u} = \frac{\gamma}{\mathrm{RePr}} \frac{1}{\rho} \nabla \cdot (\kappa \nabla T) + \frac{\gamma(\gamma - 1)\mathrm{Ma}^2}{\mathrm{Re}} \frac{1}{\rho} \Phi \tag{1.4c}$$

results, where the non-dimensional constants $\gamma = c_p/c_v$, $\mathrm{Re} = \rho_0 u_0 L_0/\mu_0$, $\mathrm{Ma} = u_0/\sqrt{\gamma R T_0}$ and $\mathrm{Pr} = \mu_0 c_p/\kappa_0$ denote the ratio of specific heats, the Reynolds number, the Mach number and the Prandtl number, respectively.

The system is completed by means of the relevant boundary conditions for a given configuration; see, e.g., [95, Sec. 1-4] and Remark 2.1.

1.2 Implicit solvers

This section provides a brief overview of the history of implicit methods, including considerations of *stability* and *accuracy*. Section 1.3 then discusses one of the highly significant innovations concerning efficiency in implicit methods, namely, the alternating direction implicit strategy.

1.2.1 Stability and convergence

The 1928 landmark paper by Courant, Friedrichs and Lewy [23] established that a consistent numerical method need not converge to the exact solution of the corresponding PDE, even though the numerical approximation of the problem is arbitrarily accurate. Specifically, that paper showed that the centered difference scheme for the wave equation cannot converge for general initial conditions unless the numerical domain of dependence includes the physical domain of dependence. This leads to a linear constraint (the CFL constraint) of the form

$$\Delta t \leq C\, h$$

for the time step Δt and the spatial mesh size h. Note that the result is only concerned with *convergence*—it does not indicate what happens to a non-converging numerical solution of a consistent scheme. It was not until Lax's equivalence theorem [59] in the 1950s that the connection with stability was made clear: any consistent numerical

method for a linear PDE converges if and only if it is stable. Certainly, while not the name itself, the concept of stability does predate this contribution. For example, Crank and Nicolson presented in the 1947 paper [24] the first implicit method for PDEs based on the trapezoidal rule for time integration and showed (using a suggestion by von Neumann communicated to those authors by Hartree) that the method was stable for the heat equation for all grid spacings h and time steps Δt, whereas the leapfrog scheme ("Richardson's method") was not. However, the word "stable" in any form does not appear in the article—what is now known as instability was termed "rapidly increasing oscillatory error" in that early contribution.

In 1956 Dahlquist [26] established a convergence theorem for the numerical solution of ordinary differential equations (ODE) with linear multistep methods, which is similar in spirit to Lax's equivalence theorem (in the later contribution [27] Dahlquist mentions, "When I wrote [that paper], I was not yet familiar with the work of Lax"). Dahlquist's result is as follows: given an ODE of the form

$$y' = f(y,t), \quad y(0) = y_0,$$

where the function $f(y,t)$ satisfies certain Lipschitz conditions, a linear multistep method for the ODE, given by a formula of the form

$$\sum_{j=0}^{s} a_j y^{n+j} = \Delta t \sum_{j=0}^{s} b_j f^{n+j} \tag{1.5}$$

for some coefficients a_j and b_j, converges if and only if the method is stable for the ODE $y' = 0$ (i.e., the method is *zero-stable*).

As a consequence of these results the stability of a scheme takes absolute precedence over considerations of accuracy. The challenges arising from stability constraints in practical applications became painfully clear with the consideration of "stiff" differential equations, a term first used by Curtiss and Hirschfelder in [25]. In stiff problems the stability constraint requires use of time steps that are much

smaller than is otherwise necessary to resolve the time evolution of the problem. The contribution [25] also introduces what would later become known as the backward differentiation formulae (BDF) multistep methods as a remedy for this difficulty, and thereby demonstrates the great value of the unconditional stability property that is sometimes afforded by implicit methods for solutions of stiff differential equations. Unfortunately, soon after this was established, certain severe limitations of implicit methods in terms of temporal accuracy order were soon discovered, as discussed in the following section.

1.2.2 Order barriers

Consider the test problem

$$y'(t) = \lambda y(t) \tag{1.6}$$

with λ in the complex plane $\mathbb{C}$, together with an associated numerical scheme and a given time step Δt; as is known, any linear multistep numerical method for equation (1.6) can be expressed in terms of the quantity $z = \lambda \Delta t$. Letting $R \subseteq \mathbb{C}$ denote the set of $z = \lambda \Delta t$ for which the scheme is stable when applied to the above equation, the question thus arises as to whether the scheme is "optimally" stable in this context, that is, whether it is stable for all Δt and for all λ for which the ODE solution is asymptotically stable as $t \to +\infty$. Or, equivalently, since (1.6) is asymptotically stable for for all complex values λ in the set $\mathbb{C}^-$ of complex numbers with non-positive real part, the question becomes whether the numerical scheme is stable for all $z \in \mathbb{C}^-$. A method satisfying this condition is said to be *A-stable*. For stiff problems, which include spectral components of the form (1.6) with large magnitude values of $\lambda \in \mathbb{C}^-$, the value of A-stable methods is unquestionable.

Unfortunately, however, a fundamental limit to the accuracy order of A-stable methods is imposed by *Dahlquist's second barrier*: There are no A-stable explicit linear multistep methods, and an (implicit) A-stable linear multistep method has ac-

curacy order not higher than two. There have been many attempts to "break" this barrier by considering more general classes of multistep methods; see, e.g., the methods surveyed in [46, Chap. V.3]. There are also higher-order implicit Runge-Kutta methods, which are not covered by Dahlquist's theorem. Nevertheless, all such methods are subject to a more general result—the Daniel-Moore conjecture [29], proved in [93]—which demands that higher-order A-stability comes at the cost of a certain number of implicit solves. Specifically, any A-stable Runge-Kutta or generalized multistep method with a number s of implicit stages can have time accuracy not higher than $2s$.

Given that the use of methods that include s implicit steps can be exceedingly expensive in the PDE context (cf. the discussion in Section 3.5.2 concerning even a single fully-dimensional implicit solve), the alternative is to consider higher-order methods which, while not A-stable, admit useful stability regions. In the language of this thesis, higher-order multistep methods for the Navier-Stokes equations and other PDEs do exist, namely quasi-unconditionally stable methods, which enjoy favorable stability restrictions.

1.2.3 Higher order implicit methods: ODE theory

Following Dahlquist's landmark 1963 contribution [28], a number of attempts were made to identify and study classes of ODE solvers with favorable stability properties. Two important concepts, namely, *A(α)-stability* and *stiff stability*, arose from these efforts. A method is said to be A(α)-stable if the stability region R contains the infinite "α-wedge" with vertex at the origin, given by $\{z \mid \arg(z) \in (\pi-\alpha,\, \pi+\alpha),\ z \neq 0\}$. A method is stiffly stable if the stability region includes the semi-infinite region $\{z \,|\, \mathrm{Re}z < -a\}$ as well as the rectangle $\{z \,|\, \mathrm{Re}z \in (-a, 0),\ \mathrm{Im}z \in (-c, c)\}$ for positive numbers a and c. Both of these concepts are illustrated in Figure 1.1.

Unfortunately, these definitions do not provide the level of detail necessary to adequately discuss the stability of PDEs such as the Navier-Stokes equations amongst

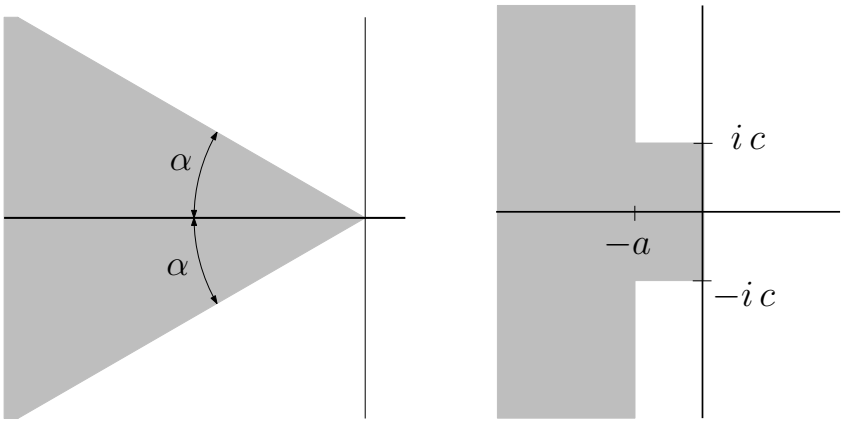


Figure 1.1: A(α)-stability (left) and stiff stability (right) takes place provided the shaded area is contained in the stability region of the ODE scheme.

many others. To demonstrate this difficulty we consider the advection-diffusion equation

$$u_t + \alpha u_x = \beta u_{xx}, \tag{1.7}$$

which is undoubtedly the simplest model problem that could be used to understand basic aspects of the Navier-Stokes equation. As will be shown in Section 2.4, the eigenvalues associated with the multistep schemes for this equation are distributed in curves that are not contained in any α-wedge with $\alpha < \pi/2$; cf. Figure 1.1.

The concept of stiff stability, on the other hand, is not well suited for discussion of the PDEs under consideration, since the stiff-stability regions, which are bounded by vertical and horizontal lines, can only provide relatively crude bounds on the parabolic-bounded eigenvalue distributions for the types of PDEs under consideration. In fact, the BDF stability regions, which approximate more closely the PDE eigenvalue distributions and which provide highly stable algorithms, *are not stiffly stable* in some cases. Additionally, considerations based on stiff stability might suggest that the fifth and sixth order BDF-ADI methods proposed in this work, which are stiffly stable, ought to give rise to better stability properties in practice than the corresponding BDF-ADI methods of orders three and four, which are not stiffly stable. This suggestion is not accurate, however. In practice, and as shown rigorously in

Section 2.4 for the linear advection-diffusion equation, the BDF-ADI methods of orders three and four are stable for a significantly larger set of discretization parameters than those required for stability in the methods of orders five and six.

We thus see that some of the concepts from ODE theory are not well adapted to the context of the PDE under consideration—at least for methods of order higher than two. (In contrast, the concept of quasi-unconditional stability introduced in Section 2.2 does accurately capture the stability character of the BDF-ADI methods proposed in this thesis.) Additionally, as discussed in the following section, implicit methods with orders of temporal accuracy higher than two have received only sparse attention in the literature. Thus the goal of the present thesis: to provide temporally high-order Navier-Stokes solvers with unconditional stability or, failing that, with as close a substitute as possible.

1.2.4 Higher order implicit methods: PDE applications

The state of the art for solvers of compressible flow is second order time accuracy as far as implicit methods are concerned—and, indeed, we believe second or higher order time-accuracy for general domains and boundary conditions has not been demonstrated before this work. The most significant innovations for compressible fluid solvers have concerned implementation techniques that improve efficiency (e.g., operator splittings and multigrid) or *relative* accuracy (e.g., Newton-like subiterations), but such improvements do not increase the *order* of accuracy of the underlying method.

Perhaps the existence of Dahlquist's second barrier may explain the widespread use of implicit methods of orders less than or equal to two (such as backward Euler, the trapezoidal rule and BDF2, all of which are A-stable), and the virtual absence of implicit methods of orders higher than two—despite the near-universality of the fourth order Runge-Kutta and Adams-Bashforth explicit counterparts. Clearly, A-stability is not necessary for all problems—for example, any method whose stability region contains the negative real axis (such as the BDF methods of orders two to six)

generally results in an unconditionally stable solver for the heat equation. A number of important questions thus arise: Do the compressible Navier-Stokes equations inherently require A-stability? Are the stability constraints of all higher-order implicit methods too stringent to be useful in the Navier-Stokes context? How close to unconditionally stable can a Navier-Stokes solver be whose temporal order of accuracy is higher than two?

Unfortunately, clear answers to these questions are not available in the literature. For example, the 2002 reference [9] compares various implicit methods for the Navier-Stokes equations, and it states: "Practical experience indicates that large-scale engineering computations are seldom stable if run with BDF4. The BDF3 scheme, with its smaller regions of instability, is often stable but diverges for certain problems and some spatial operators. Thus, a reasonable practitioner might use the BDF2 scheme exclusively for large-scale computations." However, the paper and references therein do not investigate the stability restrictions for the higher-order BDF methods, either theoretically or experimentally. As abundantly demonstrated in Chapter 4, however, methods of order higher than two can have very significant advantages for certain classes of problems, and thus, it seems useful to make available methods of a variety of temporal orders, each one of which may be best adapted to corresponding classes of subproblems—say, to high-frequency or to low-frequency problems; to problems requiring solutions for small times or to problems requiring solution for long times, etc.

The recent 2015 article [37], in turn, presents applications of the BDF scheme up to third order of time accuracy in a finite element context for the incompressible Navier-Stokes equations with turbulence modelling. This contribution does not discuss stability restrictions for the third order solver, and, in fact, it only presents numerical examples resulting from use of BDF1 and BDF2. The 2010 contribution [53], which considers a three-dimensional advection-diffusion equation, presents various ADI-type schemes, one of which is based on BDF3. The BDF3 stability analysis in

this paper, however, is restricted to the purely diffusive case.

The above examples illustrate the need for theoretical analyses and numerical investigations of higher-order implicit methods for PDEs. A major goal of this thesis is to make progress on both of these fronts, thus laying the groundwork for further work in this area.

1.3 Alternating direction implicit methods

ADI methods are based on a certain operator splitting technique (in fact, the first such technique ever introduced): they tackle PDE problems by "splitting" the relevant underlying operator, giving rise to relatively simpler problems. In the context of a first order system of PDEs $u_t = Lu$, for example, operator splitting techniques use expressions of the operator L as the sum of two or more operators, $L = \sum_j L_j$, which describe different characteristics of the problem. For example, the splitting may be along the lines of slow and fast processes, small and large scales, advective and diffusive terms, linear and nonlinear terms, or, as in the case of the ADI methods, derivatives in each spatial dimension.

The first ADI methods were introduced in the landmark papers by Peaceman and Rachford [69] and Douglas [30], where the schemes were used to solve the heat equation in two dimensions,

$$u_t = u_{xx} + u_{yy}.$$

Using a time step Δt, and centered finite difference approximations δ_{xx} and δ_{yy} for the second order derivatives, the Peaceman-Rachford scheme for the approximate solution u^{n+1} at time $t = (n+1)\Delta t$ (for non-negative integer n) can be written as

$$(I - \Delta t\, \delta_{xx})u^* = (I + \Delta t\, \delta_{yy})u^n$$
$$(I - \Delta t\, \delta_{yy})u^{n+1} = (I + \Delta t\, \delta_{xx})u^*,$$

which is formally second order accurate in space and time. Thus, the ADI splitting turns a large sparse system of equations into two sets of one-dimensional equations which can be solved efficiently with tridiagonal algorithms, greatly reducing the time and memory requirements previously needed by implicit methods for multi-dimensional PDEs.

The original papers [30, 69] generated much interest in the ADI approach, giving rise to a number of early contributions on the subject, such as the works of Douglas and Gunn [31, 32], Fairweather and Mitchell [36], D'Yakonov [33], and Yanenko [96]. Applications to problems of fluid dynamics began with the works of Pearson [70] and Chorin [22] for the incompressible Navier-Stokes equations. Briley and McDonald [11] developed ADI schemes for the compressible Navier-Stokes and Euler equations.

Undoubtedly, the best-known ADI schemes for compressible fluid-dynamics are the methods of Beam and Warming [6, 8] (also known as approximate factorization methods (AF)), which have been successfully used for years in many compressible Navier-Stokes solvers, e.g., [34, 41, 54, 72, 89, 90]. Besides the advantages gained by using the ADI methodology, the Beam and Warming method also enjoys other attractive properties: The time discretization is cleverly chosen in such a way that the stability of the scheme (for certain two dimensional linear problems) follows immediately from the stability for the underlying one-dimensional multistep method [94]. The method does use a linearization strategy based on first-order Taylor expansion of certain nonlinear fluxes, which is consistent with the nominally second-order temporal accuracies inherent in the underlying time-stepping schemes used.

Despite the success of ADI methods in general and the Beam and Warming method in particular, challenges have remained. For example, the linearization strategy based on the Taylor expansion mentioned above cannot be used in a higher-order method (since higher order terms necessarily give rise to nonlinearities). The stability of ADI methods is also difficult to analyze and, indeed, it is known [94] that the Beam and Warming method is unstable for three dimensional linear advection equations,

although it is unconditionally stable in the two dimensional advection case.

Significant follow-up efforts [34, 41, 54, 72, 89, 90] have centered around the ideas first put forth in the celebrated papers [6, 8], focusing, in particular, on enhancing stability and restoring the (nominal) second order of accuracy inherent in the original derivation of the method. The aforementioned follow-up algorithms incorporate various kinds of Newton-like subiterations to reduce the errors arising from the nonlinear terms while maintaining stability. In spite of these additions, however, the follow-up contributions still do not demonstrate second order accuracy in time by means of numerical examples—even though in all such cases nominally second order time stepping schemes are used. In contrast, these contributions do demonstrate the expected *spatial* order of accuracy with a variety of numerical examples.

Perhaps the lack of numerical evidence for second-order accuracy of the Beam and Warming method can be attributed to one of the most persistent challenges for ADI schemes—namely, the prescription of boundary conditions for intermediate unknowns that are stable and do not degrade the order of accuracy of the scheme (see, e.g., the discussions in [10, Ch. 13.3] and Section 2.1.5). Although methods can sometimes be derived for simple Dirichlet conditions (such as the boundary treatment proposed by Beam and Warming in [7] for a scalar parabolic equation), they cannot be applied to more general boundary conditions. Many attempts have been made to overcome this difficulty; for example, the contribution [77] proposes a general finite difference boundary treatment for the intermediate steps of the Beam and Warming method, but the numerical experiments do not show second order convergence of the scheme. Furthermore, the authors note the following:

> "Beam and Warming indicated that the implicit factored method employed in the present study should be unconditionally stable. Nevertheless, instability occurs when the time step size exceeds a certain limit. Numerical experiments performed here showed that for the conditions of the present study, the solution was always stable when the time step size (Δt) satisfied

the expression

$$\Delta t < 60\Delta W/a_0.$$

ΔW is the smallest grid size employed in the study and a_0, is the speed of sound."

It is unclear whether the CFL stability constraint is due to the boundary treatment or the Beam and Warming method itself.

The boundary condition difficulties that exist for many ADI schemes can usually be traced back to a simple fact: the intermediate unknowns that arise from the splitting are not necessarily consistent approximations of the physical solution. A notable exception in this regard is the splitting procedure developed by Douglas and Gunn [32]. Although not mentioned in the original papers, it was later understood [12] that the Douglas-Gunn splitting (with formal order of accuracy $s = 2$) yields equations for the intermediate unknowns that approximate the original PDE to order $s - 1 = 1$. It follows that using the physical boundary conditions at t^{n+1} for the intermediate steps preserves the order of accuracy of the method.

This thesis proposes ADI methods that address and overcome all the extant challenges to ADI-based solvers. The underlying BDF multistep method together with BDF-like extrapolation for the nonlinear terms provides higher-order-accurate methods with quasi-unconditional stability. An extension of the Douglas-Gunn splitting to our context guarantees the correct order of accuracy even in the presence of general boundary conditions.

1.4 Domain decomposition

Without question, a necessary component of any solver for the challenging problems in CFD is a method of domain decomposition. The advantages are twofold: 1) The decomposition provides a covering of the global solution domain with simpler subdomains on which an approximate solution can more easily be computed and 2) a

domain decomposition is the natural basis for dividing the computational workload in a parallel implementation of a numerical solver. In this section we give a brief history of some domain decomposition strategies relevant to the one presented in this thesis.

1.4.1 The Schwarz alternating method

The earliest contribution to domain decompositions for partial differential equations is also the foundation of most modern domain decomposition solution strategies—namely, the Schwarz method [75] which Schwarz developed for the same reason as given in point 1) in the introduction to this section—to solve a problem on a complex domain by using known solution methods on simpler ones. Here we give a brief history of the Schwarz method; see [40] for a more detailed account.

In his Ph.D. thesis, Riemann had taken for granted the existence of solutions to Laplace's equation in general domains when he proved what would later be known as the Riemann mapping theorem. When it came to his attention, he invoked what is now called Dirichlet's principle—that the solution of Laplace's equation in a domain Ω with $u = g$ on the boundary is given by the minimizer of the non-negative functional

$$J(u) = \int_\Omega \frac{1}{2} \nabla u^2$$

among all twice-differentiable u satisfying the boundary conditions. However, Weierstrass showed with a counterexample that a non-negative functional need not attain a minimizer. Of course, the existence of harmonic functions was established for simple domains, like disks and rectangles. Schwarz used this fact to construct solutions in more complex geometries.

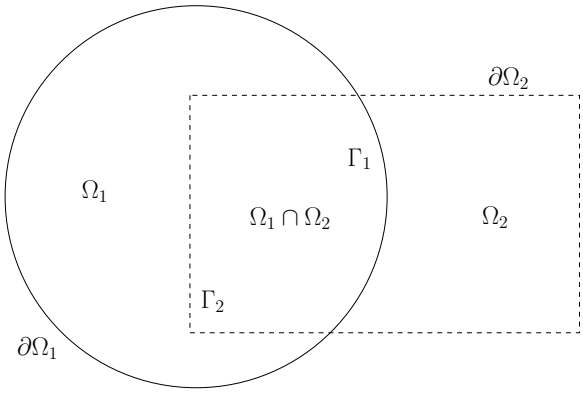


Figure 1.2: A domain Ω given by the union of a disk Ω_1 and a rectangle Ω_2.

For example, consider the problem

$$\begin{aligned} \Delta u &= 0 \quad \text{in } \Omega \\ u &= g \quad \text{on } \partial\Omega \end{aligned}$$

where the domain Ω is given by the union of two sub-domains Ω_1 and Ω_2 such that $\Omega_1 \cap \Omega_2 \neq \{\varnothing\}$, as shown in Figure 1.2. (The disk and rectangle geometry is the example Schwarz himself used in his paper.) Let $\Gamma_1 = \partial\Omega_1 \cap \Omega_2$ and $\Gamma_2 = \partial\Omega_2 \cap \Omega_1$. The original Schwarz method produces the two sequences u_1^k and u_2^k given by the solutions of the sub-problems

$$\begin{cases} \Delta u_1^{k+1} = 0 & \text{in } \Omega_1 \\ u_1^{k+1} = g & \text{on } \partial\Omega_1 \cap \partial\Omega \\ u_1^{k+1} = u_2^k & \text{on } \Gamma_1 \end{cases} \qquad \begin{cases} \Delta u_2^{k+1} = 0 & \text{in } \Omega_2 \\ u_2^{k+1} = g & \text{on } \partial\Omega_2 \cap \partial\Omega \\ u_2^{k+1} = u_1^{k+1} & \text{on } \Gamma_2. \end{cases} \tag{1.8}$$

This is also known as the *alternating* Schwarz method. Notice that the solution of the second problem requires the solution of the first, so that the procedure is *sequential*. Convergence of this method follows, in essence, from the maximum principle for harmonic functions.

In the early 1990's, Lions formally introduced the *parallel* Schwarz method in [63],

which is a modification of the original method (1.8) given by the sub-problems

$$\begin{cases} \Delta u_1^{k+1} = 0 & \text{in } \Omega_1 \\ u_1^{k+1} = g & \text{on } \partial\Omega_1 \backslash \Gamma_1 \\ u_1^{k+1} = u_2^k & \text{on } \Gamma_1 \end{cases} \quad \begin{cases} \Delta u_2^{k+1} = 0 & \text{in } \Omega_2 \\ u_2^{k+1} = g & \text{on } \partial\Omega_2 \backslash \Gamma_2 \\ u_2^{k+1} = u_1^k & \text{on } \Gamma_2. \end{cases} \tag{1.9}$$

Notice that the problems are independent, and thus form the basis for an elliptic PDE solver in a distributed computing environment.

1.4.2 Overset/Chimera/composite grid methods

More than 20 years before the work of Lions, Volkov made the first application of the Schwarz method to fully discrete PDEs in the method of composite meshes [92]—indeed, Section 8 of that paper is entitled "The use of Schwarz's alternating method for solving a system of difference equations." This was also the first instance of a general class of methods which were developed around the same time and under different names—the composite mesh method [79], the Chimera grid method [80], and the overset grid method [13] being among the most common. In this thesis, we will use the latter of these terms.

The overset grid method is ideally suited for solvers relying on spatial discretizations that make use of structured grids. Briefly, it involves decomposing the physical domain into a set overlapping logical rectangles, whereby a sub-problem is solved on each of the constituent grids. Data is communicated between these component grids by means of interpolation.

After the introduction of the overset grid method by Volkov and subsequent development by Starius [79], applications to CFD problems were explored by Steger, Dougherty, and Benek [80]. The method reached a state of maturity in the 1990s with the development of general purpose grid generation software, such as CMPGRD [21], later to evolve into the object oriented suite Overture [13], which also includes basic

capabilities for solving certain PDEs on overset grids. Of course, all the early work on overset methods was done in the context of finite differences.

More recently, the overset grid strategy has been successfully developed with the FC methodology for the solution of the compressible Navier-Stokes equations in two dimensions [2] and the elasticity equations in three dimensions [3]. A key development in those contributions is the extension of the FC method to overlapping "sub-patch" block-decompositions of larger meshes. Although the contributions [2,3] have successfully used the overset method in the context of explicit solvers, the goal of extending the framework to implicit and multi-domain implicit-explicit solvers has not yet been fully realized until now. This thesis presents the first steps toward a general framework for the solution of time-domain problems using multi-domain implicit-explicit FC solvers.

1.5 Outline of this thesis

The general outline of this thesis is as follows:

Chapter 2 introduces the BDF-ADI solver for the Navier-Stokes equations. A detailed derivation is presented, which includes consideration of curvilinear coordinate systems, treatment of nonlinear terms, the Douglas-Gunn splitting technique, and handling of boundary conditions for the intermediate unknowns. The heart of this chapter is the rigorous mathematical framework that is developed in support of the BDF-ADI method. Rigorous energy proofs of unconditional stability for the Fourier-based BDF2-ADI scheme are given for two-dimensional linear advection and parabolic equations. The concept of quasi-unconditional stability is introduced, and we prove that the Fourier-based BDF methods of orders $s = 2, \ldots, 6$ for the linear advection-diffusion equation in one, two, and three dimensions are quasi-unconditionally stable. Finally, numerical investigations compare the stability of BDF schemes with explicit Adams-Bashforth methods, and quasi-unconditional stability is numerically demon-

strated for the BDF-ADI schemes applied to the full Navier-Stokes equations in two dimensions.

Chapter 3 presents the remaining elements necessary to complete the full multi-domain implicit-explicit solver. The Fourier continuation methodology is presented, together with examples showing its higher-order accuracy and dispersion relation preserving property. We review the explicit time marching used in the explicit zones of the multi-domain solver as well as the overset method and sub-patch domain decomposition strategies. The implicit-explicit time marching method is presented, and a simple example using the advection-diffusion equation in one dimension shows the convergence rate of the parallel time-marching method. Numerical performance studies of the implicit multi-domain algorithm in a distributed computing environment are also documented.

Chapter 4 showcases the BDF-ADI and multi-domain solvers with a variety of numerical examples. The single domain BDF-ADI results use a Chebyshev collocation spatial discretization, demonstrating the stability of the solvers even in the face of very fine grid spacing. Numerical tests for this single-domain BDF-ADI solver include two dimensional unsteady flow over a bumpy plate at Reynolds number 10^6 as well as three dimensional wall bounded Taylor-Couette flow. Subsequently, results of the implicit-explicit multi-domain solver in fully parallel simulations of two dimensional flow past a cylinder and three dimensional flow past a sphere are presented. In all cases, convergence studies are included that verify the expected temporal order of accuracy of the proposed solvers—a first for implicit Navier-Stokes solvers; limited emphasis is placed on the well understood [2,10,19,47] *spatial* high-order convergence and dispersionlessness of the methods used.

Chapter 2

BDF-ADI time marching method

This chapter introduces ADI solvers of higher orders of time accuracy (orders $s = 2$ to 6) for the compressible Navier-Stokes equations in two- and three-dimensional curvilinear domains. The new ADI algorithms successfully address the difficulties discussed in Section 1.3: (i) They (provably) enjoy high orders of time-accuracy (orders two to six) *even in presence of general (and, in particular, non-periodic) boundary conditions*; and (ii) They possess remarkable stability properties, with *rigorous unconditional-stability proofs for constant coefficient hyperbolic and parabolic equations* for $s = 2$, and demonstrating in practice *quasi-unconditional stability for* $2 \leq s \leq 6$ (Definition 2.1) and *mild CFL-like constraints outside the unconditional-stability window for* $s \geq 3$ (see Section 2.4.1); and (iii) They do not require use of iterative nonlinear solvers for accuracy or stability, and they rely, instead, on a BDF-like extrapolation technique for certain components of the nonlinear terms.

The algorithms presented in this chapter, which are based on a recently developed ADI algorithm for the two-dimensional nonlinear Burgers system [15], are applicable to general *single domain* curvilinear coordinate systems and are restricted in this chapter to spectral spatial discretizations resulting from use of Fourier or polynomial spectral expansions; an accuracy order-preserving spectral filter is used in our scheme to ensure stability. Extensions of these algorithms to the multi-domain overset-grid context [13] as well as to the Fourier Continuation spatial discretization [2], are pre-

sented in subsequent chapters of this thesis. In particular, the present curvilinear domain algorithms form the single-domain implicit component of our general multi-domain implicit-explicit solver.

This chapter is organized as follows: Section 2.1 presents a derivation of the BDF-ADI method in two and three dimensions, starting with a quasilinear-like formulation of the equations and a transformation to general coordinates. The equation is then discretized in time using the BDF scheme and the treatment of nonlinearities by means of temporal extrapolation is presented. The resulting semi-discrete linear equation is factored and split using the Douglas-Gunn procedure, and enforcement of boundary conditions for the intermediate unknowns is discussed. After a brief review of relevant stability ideas and introducing the concept of quasi-unconditional stability in Section 2.2, unconditional stability is proved for the full BDF2-ADI scheme in two dimensions applied to linear constant coefficient advection and parabolic equations in Section 2.3. Next, proofs of quasi-unconditional stability for the (non-ADI) BDF methods applied to the constant coefficient advection-diffusion equation in one, two, and three dimensions are presented in Section 2.4. This section also provides qualitative analysis of the linearized Navier-Stokes equations in one spatial dimension, and numerical experiments of the full Navier-Stokes equations in two dimensions demonstrate the quasi-unconditional stability of the solvers in practice.

2.1 Proposed BDF-ADI methodology

2.1.1 Quasilinear-like Cartesian formulation

Letting $Q = (\mathbf{u}^{\mathrm{T}}, T, \rho)^{\mathrm{T}} \in \mathbb{R}^{d+2}$ denote the full $d+2$-dimensional solution vector, clearly the equations (1.4) can be expressed in the form

$$Q_t = \mathcal{P}(Q,t) \quad , \quad x \in \Omega \quad , \quad t \geq 0, \tag{2.1}$$

where $\mathcal{P}$ is a vector-valued nonlinear differential operator. The operator $\mathcal{P}$ for the Navier-Stokes equations (1.4) is autonomous, of course, but we include a possible t dependence to allow for the presence of time-dependent source terms.

The derivation of the ADI method begins with a quasilinear formulation of the equations, assuming for the moment that μ and κ are constant and neglecting the viscous dissipation function Φ:

$$\begin{aligned} Q_t &+ M^{x,1}(Q)\frac{\partial}{\partial x}Q + M^{y,1}(Q)\frac{\partial}{\partial y} + M^{z,1}(Q)\frac{\partial}{\partial z}Q \\ &+ M^{x,2}(Q)\frac{\partial^2}{\partial x^2}Q + M^{y,2}(Q)\frac{\partial^2}{\partial y^2}Q + M^{z,2}(Q)\frac{\partial^2}{\partial z^2}Q \\ &+ M^{xy}(Q)\frac{\partial^2}{\partial x \partial y}Q + M^{xz}(Q)\frac{\partial^2}{\partial x \partial z}Q + M^{yz}(Q)\frac{\partial^2}{\partial y \partial z}Q + M^{0}(Q)Q \\ &= 0; \end{aligned} \tag{2.2}$$

and the corresponding equations for $d = 2$ are given by

$$\begin{aligned} Q_t &+ M^{x,1}(Q)\frac{\partial}{\partial x}Q + M^{y,1}(Q)\frac{\partial}{\partial y} + M^{x,2}(Q)\frac{\partial^2}{\partial x^2}Q + M^{y,2}(Q)\frac{\partial^2}{\partial y^2}Q \\ &+ M^{xy}(Q)\frac{\partial^2}{\partial x \partial y}Q + M^{0}(Q)Q = 0. \end{aligned} \tag{2.3}$$

Here the various M matrices ($M^{x,1}$, $M^{x,2}$ etc.) are matrix-valued functions of Q. The purpose of using the quasilinear form of the equations is, upon temporal discretization, to treat all spatial derivatives implicitly (if possible) and to approximate the nonlinear coefficients of the derivatives explicitly in time, resulting in a linear system of equations in Q at the current time level together with its derivatives; the details are presented in the following sections.

The actual Navier-Stokes equations (for which μ and κ are generally functions of T and for which Φ is non-zero) are not quasilinear, but can still be expressed in the form (2.2) or (2.3) by allowing the matrices to incorporate some derivative terms. For example, squared terms such as u_x^2 are handled by including one u_x term in the

matrix $M^{x,1}$ and the second u_x term in the vector $\partial_x Q$ in equation (2.3). Similarly, the product $\mu(T)_x u_y$ is expanded using the chain rule and written as

$$\begin{aligned}\mu(T)_x u_y &= \mu'(T) T_x u_y \\ &= \left(\frac{1}{2}\mu'(T) T_x\right) u_y + \left(\frac{1}{2}\mu'(T) u_y\right) T_x.\end{aligned}$$

The two quantities in parentheses are included in the matrices $M^{y,1}$ and $M^{x,1}$ respectively. Thus, the implicit treatment of the product of two spatial derivatives is symmetric. The matrices resulting from this treatment of nonlinear terms can be found in Appendix A. Clearly, there are other ways of treating nonlinear products of derivatives, but we chose the above for symmetry.

Remark 2.1: *For notational simplicity our description of the BDF-ADI algorithms assumes that no-slip boundary conditions of the form*

$$\begin{pmatrix} \mathbf{u} \\ T \end{pmatrix}\Bigg|_{\partial D} = \begin{pmatrix} g^{\mathbf{u}} \\ g^{T} \end{pmatrix} \tag{2.4}$$

are prescribed, where $g^{\mathbf{u}}$ and g^T are given functions defined on ∂D. Certainly, other relevant types of boundary conditions can be incorporated within the proposed framework—Section 4.1 includes an example of unsteady boundary layer flow that incorporates no-slip boundary conditions at a rough boundary as well as inflow and absorbing boundary conditions.

2.1.2 Quasilinear-like curvilinear formulation

Let $\xi(x, y, z)$, $\eta(x, y, z)$, $\zeta(x, y, z)$ define a smooth mapping from the physical (Cartesian) domain $\Omega \subset \mathbb{R}^d$ to the (ξ, η, ζ) computational domain, which we take to be the cube $D = [\ell_1, \ell_2]^d$ for some real numbers ℓ_1 and ℓ_2, $d = 2$, 3. Using the chain rule, the derivatives with respect to x, y, and z are expressed in terms of derivatives with

respect to ξ, η, and ζ; see, e.g., [48]. We can then collect terms to obtain an equation in general coordinates for $Q = Q(\xi, \eta, \zeta, t)$:

$$\begin{aligned} & Q_t + M^{\xi,1}(Q)\frac{\partial}{\partial \xi}Q + M^{\eta,1}(Q)\frac{\partial}{\partial \eta} + M^{\zeta,1}(Q)\frac{\partial}{\partial \zeta}Q \\ & \quad + M^{\xi,2}(Q)\frac{\partial^2}{\partial \xi^2}Q + M^{\eta,2}(Q)\frac{\partial^2}{\partial \eta^2}Q + M^{\zeta,2}(Q)\frac{\partial^2}{\partial \zeta^2}Q \\ & \quad + M^{\xi\eta}(Q)\frac{\partial^2}{\partial \xi \partial \eta}Q + M^{\xi\zeta}(Q)\frac{\partial^2}{\partial \xi \partial \zeta}Q + M^{\eta\zeta}(Q)\frac{\partial^2}{\partial \eta \partial \zeta}Q + M^0(Q)Q \\ & \quad = 0 \end{aligned} \tag{2.5}$$

for $(\xi, \eta, \zeta) \in D$ where the matrix functions $M^{\xi,1}$ of Q, etc. are computed using the Cartesian matrices and metric terms.

For $d = 2$ the computational domain is $D = [\ell_1, \ell_2]^2$ and we have the equation

$$\begin{aligned} & Q_t + M^{\xi,1}(Q)\frac{\partial}{\partial \xi}Q + M^{\eta,1}(Q)\frac{\partial}{\partial \eta} + M^{\xi,2}(Q)\frac{\partial^2}{\partial \xi^2}Q + M^{\eta,2}(Q)\frac{\partial^2}{\partial \eta^2}Q \\ & \quad + M^{\xi\eta}(Q)\frac{\partial^2}{\partial \xi \partial \eta} + M^0(Q)Q = 0. \end{aligned} \tag{2.6}$$

To simplify the presentation of boundary conditions for the ADI scheme, we decompose the boundary ∂D by defining

$$\partial^\xi D = \{\, (\xi, \eta, \zeta) \in D \,|\, \xi = \ell_1 \text{ or } \ell_2 \,\}, \tag{2.7a}$$

$$\partial^\eta D = \{\, (\xi, \eta, \zeta) \in D \,|\, \eta = \ell_1 \text{ or } \ell_2 \,\}, \tag{2.7b}$$

$$\partial^\zeta D = \{\, (\xi, \eta, \zeta) \in D \,|\, \zeta = \ell_1 \text{ or } \ell_2 \,\}, \tag{2.7c}$$

for $d = 3$ and

$$\partial^\xi D = \{\, (\xi, \eta) \in D \,|\, \xi = \ell_1 \text{ or } \ell_2 \,\}, \tag{2.8a}$$

$$\partial^\eta D = \{\, (\xi, \eta) \in D \,|\, \eta = \ell_1 \text{ or } \ell_2 \,\}, \tag{2.8b}$$

for $d = 2$.

2.1.3 BDF semi-discretization; treatment of non-linearities.

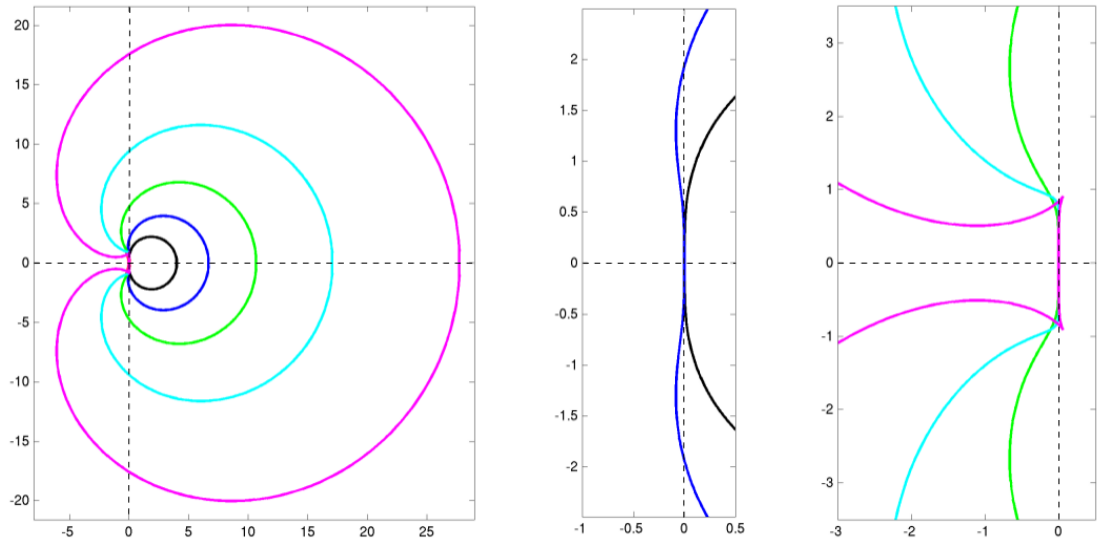


Figure 2.1: The left pane shows a plot of the boundaries of the regions of absolute stability for all BDF methods of orders $s = 2, \ldots, 6$. From innermost to outermost, the curves correspond to the methods of increasing order s. The regions of absolute stability are exterior to the corresponding boundaries. The middle (resp. right) pane shows a close-up near the origin of the boundaries for the methods of order $s = 2, 3$ (resp. $s = 4, 5, 6$).

To produce our BDF-based numerical solver for the system (1.4) we first lay down a semi-discrete approximation of this equation—discrete in time but continuous in space—on the basis of the BDF multistep method of order s [58, Ch. 3.12]. Considering the concise expression (2.1) we thus let Q^j denote the numerical approximation of Q at time $t = t^j$ and we approximate Q_t at $t = t^{n+1}$ by the time derivative of the $(d+2)$-dimensional vector $V = V(t)$ of polynomials of degree s in the variable t that interpolates the vector values $\{t^{n+1-j}, Q^{n+1-j}\}$, $0 \le j \le s$. Using the right-hand side value $\mathcal{P}(Q^{n+1}, t^{n+1})$ this procedure results in the well known implicit $(\Delta t)^{s+1}$-accurate

order-s BDF formula

$$Q^{n+1} = \sum_{k=0}^{s-1} a_k Q^{n-k} + b\Delta t\, \mathcal{P}(Q^{n+1}, t^{n+1}), \tag{2.9}$$

where a_k and b are the s-th order BDF coefficients. Table 2.1 displays the BDF coefficients for $s = 1$ through 6, and Figure 2.1 shows the regions of absolute stability in the complex plane. (BDF methods of orders greater than 6 have stability regions that do not include a neighborhood of the origin in the region $\mathrm{Re}\, z < 0$ and therefore are not convergent as $\Delta t \to 0$.)

s	a_0	a_1	a_2	a_3	a_4	a_5	b
1	1						1
2	$\frac{4}{3}$	$-\frac{1}{3}$					$\frac{2}{3}$
3	$\frac{18}{11}$	$-\frac{9}{11}$	$\frac{2}{11}$				$\frac{6}{11}$
4	$\frac{48}{25}$	$-\frac{36}{25}$	$\frac{16}{25}$	$-\frac{3}{25}$			$\frac{12}{25}$
5	$\frac{300}{137}$	$-\frac{300}{137}$	$\frac{200}{137}$	$-\frac{75}{137}$	$\frac{12}{137}$		$\frac{60}{137}$
6	$\frac{360}{147}$	$-\frac{450}{147}$	$\frac{400}{147}$	$-\frac{225}{147}$	$\frac{72}{147}$	$-\frac{10}{147}$	$\frac{60}{147}$

Table 2.1: Coefficients for BDF methods of orders $s = 1, \ldots, 6$.

In order to express the resulting algorithm in terms of the M-matrices in equations 2.5 and 2.6, for a given $(d+2)$-vector R we define the differential operators

$$\mathcal{A}[R] = \sum_{j=0}^{2} M^{\xi,j}(R)\frac{\partial^j}{\partial \xi^j}, \tag{2.10a}$$

$$\mathcal{B}[R] = \sum_{j=1}^{2} M^{\eta,j}(R)\frac{\partial^j}{\partial \eta^j}, \tag{2.10b}$$

$$\mathcal{C}[R] = \sum_{j=1}^{2} M^{\zeta,j}(R)\frac{\partial^j}{\partial \zeta^j}, \tag{2.10c}$$

$$\mathcal{F}[R] = M^{\xi\eta}(R)\frac{\partial^2}{\partial \xi \partial \eta}, \tag{2.10d}$$

$$\mathcal{G}[R] = M^{\xi\eta}(R)\frac{\partial^2}{\partial \xi \partial \eta} + M^{\xi\zeta}(R)\frac{\partial^2}{\partial \xi \partial \zeta} + M^{\eta\zeta}(R)\frac{\partial^2}{\partial \eta \partial \zeta}, \tag{2.10e}$$

in the variables (ξ, η, ζ); note the definition $M^{\xi,0}(R) \equiv M^0(R)$ which is implicit in these equations. For example, an application of the differential operator $\mathcal{A}[R]$ to a vector function S results in the expression

$$\mathcal{A}[R]S = M^{\xi,0}(R)S + M^{\xi,1}(R)\frac{\partial S}{\partial \xi} + M^{\xi,2}(R)\frac{\partial^2 S}{\partial \xi^2},$$

and similarly for $\mathcal{B}, \mathcal{C}, \mathcal{F}$ and $\mathcal{G}$. (We note that, as pointed out in Section 2.1.1, the M matrices in these expressions may contain derivatives of the vector R. Notice, further, that, as indicated in what follows, both R and S are taken in our method as suitable approximations of the solution vector Q.) Equation (2.9) can then be re-expressed in the $(\Delta t)^{s+1}$-accurate form

$$\left(I + b\Delta t\left(\mathcal{A}\left[Q^{n+1}\right] + \mathcal{B}\left[Q^{n+1}\right]\right)\right)Q^{n+1} = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{F}\left[Q^{n+1}\right]Q^{n+1} \tag{2.11}$$

in two dimensions and

$$\left(I + b\Delta t\left(\mathcal{A}\left[Q^{n+1}\right] + \mathcal{B}\left[Q^{n+1}\right] + \mathcal{C}\left[Q^{n+1}\right]\right)\right)Q^{n+1} = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{G}\left[Q^{n+1}\right]Q^{n+1} \tag{2.12}$$

in three dimensions.

As mentioned in the introduction to this chapter, previous ADI-based Navier-Stokes solvers have relied on either linearization or iterations to adequately account for nonlinear terms. The methods proposed in this thesis, in turn, use the polynomial extrapolations

$$\widetilde{Q}_p^{n+1} \equiv \sum_{k=0}^{p-1} (-1)^k \binom{p}{k+1} Q^{n-k} \quad (p \geq 1) \tag{2.13}$$

to approximate the matrix-valued functions M in the operators (2.10)—which, in particular, gives rise to *high-order-accurate* approximations of the full nonlinear term at time t_{n+1}. The formula (2.13) can be obtained by evaluating at $t = t^{n+1}$ the Lagrange interpolating polynomial

$$\widetilde{Q}_p(t) = \sum_{k=0}^{p-1} \frac{\ell_k(t)}{\ell_k(t^{n-k})} Q^{n-k},$$

where $t^m = m\,\Delta t$ are equispaced points in time and where

$$\ell_k(t) = \prod_{\substack{0 \leq j \leq p-1 \\ j \neq k}} (t - t^{n-j}).$$

It follows that

$$\widetilde{Q}_p^{n+1} = Q^{n+1} + \mathcal{O}((\Delta t)^p).$$

The extrapolated solution is used as follows: defining the variable coefficient differential operators

$$\mathcal{A}_s = \mathcal{A}[\widetilde{Q}_s^{n+1}], \quad \mathcal{B}_s = \mathcal{B}[\widetilde{Q}_s^{n+1}], \quad \mathcal{C}_s = \mathcal{C}[\widetilde{Q}_s^{n+1}], \quad \mathcal{F}_s = \mathcal{F}[\widetilde{Q}_s^{n+1}], \quad \mathcal{G}_s = \mathcal{G}[\widetilde{Q}_s^{n+1}], \tag{2.14}$$

we have

$$\mathcal{A}_s Q^{n+1} = \mathcal{A}[Q^{n+1}]Q^{n+1} + \mathcal{O}((\Delta t)^s),$$

with similar expressions for the other operators in (2.14). We thus obtain the *linear* equations

$$(I + b\Delta t\,\mathcal{A}_s + b\Delta t\,\mathcal{B}_s)\,Q^{n+1} = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{F}_s Q^{n+1} \tag{2.15}$$

and

$$(I + b\Delta t\,\mathcal{A}_s + b\Delta t\,\mathcal{B}_s + b\Delta t\,\mathcal{C}_s)\,Q^{n+1} = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{G}_s Q^{n+1} \tag{2.16}$$

for Q^{n+1} in two and three dimensions, respectively. Clearly, these equations are equivalent to the corresponding $s+1$-th order equations (2.11) and (2.12) respectively up to an error of order $(\Delta t)^{s+1}$, and thus they themselves are accurate to order $s+1$ in time. Clearly, approximations of order higher than s for the operators (2.14) (e.g., approximation of $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{F}$ in equation (2.11) by $\mathcal{A}_m$, $\mathcal{B}_m$, and $\mathcal{F}_m$, respectively, with $m > s$) also preserves the order of the local truncation error, but we have found the resulting algorithms to be unstable.

2.1.4 ADI factorizations and splittings

Sections 2.1.4.1–2.1.4.4 describe our application of the Douglas-Gunn method to the semidiscrete linear (but accurate to high order in time!) schemes (2.15) and (2.16). Adequate treatment of the boundary conditions is a subject of great importance that is taken up in Section 2.1.4.2 for two dimensions and Section 2.1.4.4 for three dimensions.

2.1.4.1 Application of the Douglas-Gunn method in two space dimensions

Adding the cross term $(b\Delta t)^2 \mathcal{A}_s \mathcal{B}_s Q^{n+1}$ to both sides of equation (2.15) and factoring the resulting left-hand side exactly we obtain

$$(I + b\Delta t\, \mathcal{A}_s)\,(I + b\Delta t\, \mathcal{B}_s)\, Q^{n+1} = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\, \mathcal{F}_s Q^{n+1} + (b\Delta t)^2 \mathcal{A}_s \mathcal{B}_s Q^{n+1}. \tag{2.17}$$

To eliminate the dependence on Q^{n+1} on the right-hand side of this equation we resort once again to extrapolation: the arguments Q^{n+1} in the right-hand side terms $b\,\Delta t\, \mathcal{F}_s Q^{n+1}$ and $(b\,\Delta t)^2 \mathcal{A}_s \mathcal{B}_s Q^{n+1}$ are substituted, with errors of order $(\Delta t)^{s+1}$, by the extrapolated values $\widetilde{Q}_s^{n+1}$ and $\widetilde{Q}_{s-1}^{n+1}$, respectively (see equation (2.13)), and we thus obtain the equation

$$(I + b\Delta t\, \mathcal{A}_s)\,(I + b\Delta t\, \mathcal{B}_s)\, Q^{n+1} = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\, \mathcal{F}_s \widetilde{Q}_s^{n+1} + (b\Delta t)^2 \mathcal{A}_s \mathcal{B}_s \widetilde{Q}_{s-1}^{n+1} \tag{2.18}$$

whose solution only requires inversion of the operators $(I + b\Delta t\, \mathcal{A}_s)$ and $(I + b\Delta t\, \mathcal{B}_s)$.

Remark 2.2: *Notice that, although the approximation $\widetilde{Q}_{s-1}^{n+1}$ is accurate to order $s-1$ the overall accuracy order in the quantity $(b\Delta t)^2 \mathcal{A}_s \mathcal{B}_s \widetilde{Q}_{s-1}^{n+1}$ is $(\Delta t)^{s+1}$, as needed—in view of the $(\Delta t)^2$ prefactor in this expression. While the approximation $\widetilde{Q}_s^{n+1}$ could have been used while preserving the accuracy order, we have found that use of the lower order extrapolation $\widetilde{Q}_{s-1}^{n+1}$ is necessary to ensure stability. Similar comments apply to the term $b\Delta t\, \mathcal{F}_s \widetilde{Q}_s^{n+1}$.*

To complete the proposed ADI scheme an appropriate splitting of equation (2.18) (that is, an alternating direction method for evaluation of Q^{n+1}) must be used. The form of the right-hand side of equation (2.18) motivates the consideration of a generic

ADI splitting in terms of $\sum_k Q^{n-k}$, $\widetilde{Q}_s^{n+1}$, and $\widetilde{Q}_{s-1}^{n-1}$:

$$(I + b\Delta t\,\mathcal{A}_s)\,Q^* = A_1 \sum_{k=0}^{s-1} Q^{n-k} + A_2\widetilde{Q}_s^{n+1} + A_3\widetilde{Q}_{s-1}^{n+1} \tag{2.19a}$$

$$(I + b\Delta t\,\mathcal{B}_s)\,Q^{n+1} = B_0 Q^* + B_1 \sum_{k=0}^{s-1} Q^{n-k} + B_2\widetilde{Q}_s^{n+1} + B_3\widetilde{Q}_{s-1}^{n+1} \tag{2.19b}$$

where $Q^* = (\mathbf{u}^{*\mathrm{T}}, T*, \rho*)^{\mathrm{T}} \in \mathbb{R}^{d+2}$ is a new *intermediate unknown*, and where the various A and B terms on the right-hand sides denote operators that are as yet to be determined. Multiplying the second equation on the left by $(I + b\Delta t\,\mathcal{A}_s)$ and assuming B_0 and $\mathcal{A}_s$ commute, we have

$$\begin{aligned}
(I + b\Delta t\,\mathcal{A}_s)\,(I + b\Delta t\,\mathcal{B}_s)\,Q^{n+1} &= B_0(A_1 \sum_{k=0}^{s-1} Q^{n-k} + A_2\widetilde{Q}_s^{n+1} + A_3) \\
&\quad + (I + b\Delta t\,\mathcal{A}_s)\,(B_1 \sum_{k=0}^{s-1} Q^{n-k} + B_2\widetilde{Q}_s^{n+1} + B_3\widetilde{Q}_{s-1}^{n+1}) \\
&= (B_0A_1 + (I + b\Delta t\,\mathcal{A}_s)B_1) \sum_{k=0}^{s-1} Q^{n-k} \\
&\quad + (B_0A_2 + (I + b\Delta t\,\mathcal{A}_s)B_2)\widetilde{Q}_s^{n+1} \\
&\quad + (B_0A_3 + (I + b\Delta t\,\mathcal{A}_s)B_3)\widetilde{Q}_{s-1}^{n+1}.
\end{aligned} \tag{2.20}$$

By equating the right-hand sides of equations (2.18) and (2.20), the operators A_i and B_i must satisfy

$$B_0A_1 + (I + b\Delta t\,\mathcal{A}_s)B_1 = I, \tag{2.21a}$$

$$B_0A_2 + (I + b\Delta t\,\mathcal{A}_s)B_2 = -b\Delta t\,\mathcal{F}_s, \tag{2.21b}$$

$$B_0A_3 + (I + b\Delta t\,\mathcal{A}_s)B_3 = (b\Delta t)^2\,\mathcal{A}_s\mathcal{B}_s. \tag{2.21c}$$

Operators satisfying these equations exist but they are not uniquely determined. For example, the selection $B_0 = I$, $B_1 = B_2 = B_3 = 0$ results in the *classical splitting*

$$(I + b\Delta t\,\mathcal{A}_s)\,Q^* = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{M}_s\,\widetilde{Q}_s^{n+1} + b^2(\Delta t)^2\mathcal{A}_s\mathcal{B}_s\,\widetilde{Q}_{s-1}^{n+1} \tag{2.22a}$$

$$(I + b\Delta t\,\mathcal{B}_s)\,Q^{n+1} = Q^*. \tag{2.22b}$$

However, as discussed in the next section, the classical splitting presents certain challenges: computable boundary conditions for the intermediate quantity Q^* result in reductions in the order of accuracy of the method [62]. We avoid such difficulties by using instead the Douglas-Gunn splitting [31]. To derive it we simplify the equations (2.21) by taking $B_0 = b\Delta t\,\mathcal{A}_s$ and I, which naturally satisfy the assumption that B_0 and $\mathcal{A}_s$ commute. In the case $B_0 = b\Delta t\,\mathcal{A}_s$ we have

$$\begin{aligned} b\Delta t\,\mathcal{A}_s(A_1 + B_1) &= I - B_1 \\ b\Delta t\,\mathcal{A}_s(A_2 + B_2) &= -b\Delta t\,\mathcal{F}_s - B_2 \\ b\Delta t\,\mathcal{A}_s(A_3 + B_3) &= (b\Delta t)^2\,\mathcal{A}_s\mathcal{B}_s - B_3 \end{aligned}$$

so that with the selections $B_1 = I$, $B_2 = -b\Delta t\,\mathcal{F}_s$, $A_1 = -B_1 = -I$, $A_2 = -B_2 = b\Delta t\,\mathcal{F}_s$, and $A_3 = b\Delta t\,\mathcal{A}_s$ equations (2.21) are satisfied. These selections give rise to the Douglas-Gunn splitting

$$(I + b\Delta t\,\mathcal{A}_s)\,Q^* = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{F}_s\,\widetilde{Q}_s^{n+1} - b\Delta t\,\mathcal{B}_s\,\widetilde{Q}_{s-1}^{n+1} \tag{2.23a}$$

$$(I + b\Delta t\,\mathcal{B}_s)\,Q^{n+1} = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{F}_s\,\widetilde{Q}_s^{n+1} - b\Delta t\,\mathcal{A}_s Q^*. \tag{2.23b}$$

The alternative selection $B_0 = I$ in equations (2.21) leads to the system

$$A_1 + (I + b\Delta t\, \mathcal{A}_s)B_1 = I$$
$$A_2 + (I + b\Delta t\, \mathcal{A}_s)B_2 = -b\Delta t\, \mathcal{F}_s$$
$$A_3 + (I + b\Delta t\, \mathcal{A}_s)B_3 = (b\Delta t)^2\, \mathcal{A}_s\mathcal{B}_s$$

which is satisfied provided, e.g., the additional selections $B_1 = B_2 = 0$, $B_3 = b\Delta t\, \mathcal{B}_s$, $A_1 = I$, $A_2 = -b\Delta t\, \mathcal{F}_s$, and $A_3 = -B_3 = -b\Delta t\, \mathcal{B}_s$ are made. The resulting ADI splitting reads

$$(I + b\Delta t\, \mathcal{A}_s)\, Q^* = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\, \mathcal{F}_s \widetilde{Q}_s^{n+1} - b\Delta t\, \mathcal{B}_s\, \widetilde{Q}_{s-1}^{n+1} \tag{2.24a}$$

$$(I + b\Delta t\, \mathcal{B}_s)\, Q^{n+1} = Q^* + b\Delta t\, \mathcal{B}_s\, \widetilde{Q}_{s-1}^{n+1}. \tag{2.24b}$$

This splitting is in fact equivalent to (2.23) (as it can be checked by subtracting equation (2.23a) from (2.23b)). The splitting (2.24) is less expensive than (2.23)—since 1) Equation (2.24) does not require the additional computation of the term $\mathcal{A}_s Q^*$, and 2) The term $b\Delta t\, \mathcal{B}_s\, \widetilde{Q}_{s-1}^{n+1}$ in (2.24) can be computed once for each full time step and used in both ADI sweeps—and is therefore used in the two-dimensional implementation presented in Section 2.5.

2.1.4.2 Order-preserving boundary conditions for the splitting (2.24)

Use of the two-dimensional ADI splitting (2.24) entails evaluation of solutions of systems of ODEs for the intermediate unknown Q^* as well as the physical unknown Q^{n+1}. Both of these solves require use of appropriate boundary conditions. Here we show that equations (2.24) possess the following remarkable property: imposing boundary conditions for Q^* which coincide with the corresponding boundary conditions for Q at time $t = t^{n+1}$ preserves the overall $(\Delta t)^{s+1}$ truncation error otherwise implicit in these equations.

In view of Remark 2.1, in what follows we assume the Navier-Stokes boundary conditions

$$\begin{pmatrix} \mathbf{u}(\xi,\eta,t) \\ T(\xi,\eta,t) \end{pmatrix} = \begin{pmatrix} g^{\mathbf{u}}(\xi,\eta,t) \\ g^{T}(\xi,\eta,t) \end{pmatrix}, \quad (\xi,\eta) \in \partial D, \tag{2.25}$$

for the unknown $Q = (\mathbf{u}^{\mathrm{T}}, T, \rho)^{\mathrm{T}} \in \mathbb{R}^{2+2}$ ($Q = Q(\xi,\eta,t)$) at a solid-fluid interface. Comparison of equations (2.23b) and (2.15) shows that the truncation error in (2.23b) is a quantity of order Δt^{s+1} if and only if Q^* is an $(s-1)$ order-accurate approximation of Q^{n+1} everywhere in the domain D. But the intermediate unknown Q^* is indeed accurate to order $s-1$ throughout D provided the boundary conditions of Q^* are taken to coincide with those for $Q(t^{n+1})$ at the relevant interval endpoints—since, clearly, substitution of Q^* by Q^{n+1} in (2.23a) results in an equation containing an error of order $(\Delta t)^{s+1}$. Thus, use of boundary values of the solution at time $t = t^{n+1}$ for the intermediate-time unknown $Q^* = (\mathbf{u}^{*\mathrm{T}}, T^*, \rho^*)^{\mathrm{T}}$, that is

$$\begin{pmatrix} \mathbf{u}^*(\xi,\eta) \\ T^*(\xi,\eta) \end{pmatrix} = \begin{pmatrix} g^{\mathbf{u}}(\xi,\eta,t^{n+1}) \\ g^{T}(\xi,\eta,t^{n+1}) \end{pmatrix} \quad \text{for } \xi = \ell_1, \ell_2 \text{ and } \eta \in [\ell_1, \ell_2] \tag{2.26a}$$

$$\begin{pmatrix} \mathbf{u}^{n+1}(\xi,\eta) \\ T^{n+1}(\xi,\eta) \end{pmatrix} = \begin{pmatrix} g^{\mathbf{u}}(\xi,\eta,t^{n+1}) \\ g^{T}(\xi,\eta,t^{n+1}) \end{pmatrix} \quad \text{for } \eta = \ell_1, \ell_2 \text{ and } \xi \in [\ell_1, \ell_2], \tag{2.26b}$$

maintains the overall $\mathcal{O}(\Delta t)^{s+1}$ truncation error in the Douglas-Gunn scheme for the complete time step $t^n \to t^{n+1}$. The results in Section 4.1 demonstrate the expected order of accuracy is achieved in the case of general boundary conditions, including cases in which time-dependent boundary conditions are specified.

2.1.4.3 ADI factorization and splitting in three spatial dimensions

To derive the BDF-ADI method in three dimensions, we begin by factoring (2.12) to obtain

$$\begin{aligned}(I + b\Delta t\,\mathcal{A}_s)\,(I + b\Delta t\,\mathcal{B}_s)\,(I + b\Delta t\,\mathcal{C}_s)\,Q^{n+1} &= \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{G}_s Q^{n+1} \\ &\quad +(b\Delta t)^2\,(\mathcal{A}_s\mathcal{B}_s + \mathcal{A}_s\mathcal{C}_s + \mathcal{B}_s\mathcal{C}_s)\,Q^{n+1} \\ &\quad +(b\Delta t)^3\mathcal{A}_s\mathcal{B}_s\mathcal{C}_s Q^{n+1}. \qquad (2.27)\end{aligned}$$

Using polynomial extrapolations to substitute for Q^{n+1} on the right-hand side of the above equation we have

$$\begin{aligned}(I + b\Delta t\,\mathcal{A}_s)\,(I + b\Delta t\,\mathcal{B}_s)\,(I + b\Delta t\,\mathcal{C}_s)\,Q^{n+1} &= \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{M}_s \widetilde{Q}_s^{n+1} \\ &\quad +(b\Delta t)^2\,(\mathcal{A}_s\mathcal{B}_s + \mathcal{A}_s\mathcal{C}_s + \mathcal{B}_s\mathcal{C}_s)\,\widetilde{Q}_{s-1}^{n+1} \\ &\quad +(b\Delta t)^3\mathcal{A}_s\mathcal{B}_s\mathcal{C}_s\,\widetilde{Q}_{s-1}^{n+1}, \qquad (2.28)\end{aligned}$$

which is equivalent to (2.27) up to an error term of order $\mathcal{O}((\Delta t)^{s+1})$. Note that the term of order $(\Delta t)^3$ is extrapolated to order $s-1$ rather than $s-2$. Although our experiments suggest that using a higher order extrapolation than strictly necessary for the terms of order Δt and $(\Delta t)^2$ can give rise to instability, we have found that the extrapolation to order $s-1$ for the $(\Delta t)^3$ term does not affect the stability of the method.

A procedure similar to the one described in the two-dimensional case can be used

to derive the Douglas-Gunn splitting

$$\begin{aligned}
(I + b\Delta t\,\mathcal{A}_s)\,Q^* &= \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{M}_s \widetilde{Q}_s^{n+1} \\
&\quad -b\Delta t\,(\mathcal{B}_s + \mathcal{C}_s)\,\widetilde{Q}_{s-1}^{n+1} && \text{(2.29a)} \\
(I + b\Delta t\,\mathcal{B}_s)\,Q^{**} &= \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{M}_s \widetilde{Q}_s^{n+1} \\
&\quad -b\Delta t\,\mathcal{A}_s Q^* - b\Delta t\,\mathcal{C}_s \widetilde{Q}_{s-1}^{n+1} && \text{(2.29b)} \\
(I + b\Delta t\,\mathcal{C}_s)\,Q^{n+1} &= \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{M}_s \widetilde{Q}_s^{n+1} \\
&\quad -b\Delta t\,\mathcal{A}_s Q^* - b\Delta t\,\mathcal{B}_s Q^{**}. && \text{(2.29c)}
\end{aligned}$$

The equivalent form used in the implementation of the method is

$$\begin{aligned}
(I + b\Delta t\,\mathcal{A}_s)\,Q^* &= \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{M}_s \widetilde{Q}_s^{n+1} \\
&\quad - b\Delta t\,(\mathcal{B}_s + \mathcal{C}_s)\,\widetilde{Q}_{s-1}^{n+1} && \text{(2.30a)} \\
(I + b\Delta t\,\mathcal{B}_s)\,Q^{**} &= Q^* + b\Delta t\,\mathcal{B}_s \widetilde{Q}_{s-1}^{n+1} && \text{(2.30b)} \\
(I + b\Delta t\,\mathcal{C}_s)\,Q^{n+1} &= Q^{**} + b\Delta t\,\mathcal{C}_s \widetilde{Q}_{s-1}^{n+1}. && \text{(2.30c)}
\end{aligned}$$

Multiplying (2.30c) on the left by $(I + b\Delta t\mathcal{A}_s)(I + b\Delta t\mathcal{B}_s)$ and eliminating Q^* and Q^{**}, it follows that (2.30) is equivalent to (2.28) up to terms on the order of the truncation error, $\mathcal{O}((\Delta t)^{s+1})$.

2.1.4.4 Boundary conditions in three dimensions

As in the two-dimensional case, we show that using the boundary conditions at $t = t^{n+1}$ for the intermediate unknowns Q^* and Q^{**} preserves the order of accuracy of the physical solution Q^{n+1}. In two-dimensional space, this followed from the fact that the ODE for Q^* was an $s-1$ order accurate approximation of the unfactored PDE (2.11) and that the ODE for Q^{n+1} was an s order accurate approximation of (2.11) if and

only if Q^* was $s-1$ order accurate. By comparing the equations (2.29) to (2.12) it follows that the ADI sweeps for Q^*, Q^{**} and Q^{n+1} satisfy similar conditions – namely, Q^* is an $s-1$ order accurate solution, Q^{**} is $s-1$ order accurate if and only if Q^* is $s-1$ order accurate, and Q^{n+1} is s order accurate if and only if Q^* and Q^{**} are both $s-1$ order accurate. It follows that the Douglas-Gunn splitting in three dimensions allows the use of boundary conditions at $t = t^{n+1}$ when solving for the intermediate unknowns while preserving the order of accuracy of the method.

For example, given boundary conditions of the form

$$\begin{pmatrix} \mathbf{u} \\ T \end{pmatrix} = \begin{pmatrix} g^{\mathbf{u}}(\xi,\eta,\zeta,t) \\ g^{T}(\xi,\eta,\zeta,t) \end{pmatrix}, \quad (\xi,\eta,\zeta) \in \partial D, \tag{2.31}$$

the boundary conditions to be used in conjunction with the three dimensional ADI scheme (2.30) are

$$\begin{pmatrix} \mathbf{u}^* \\ T^* \end{pmatrix} = \begin{pmatrix} g^{\mathbf{u}}(\xi,\eta,\zeta,t^{n+1}) \\ g^{T}(\xi,\eta,\zeta,t^{n+1}) \end{pmatrix}, \quad (\xi,\eta,\zeta) \in \partial^{\xi} D, \tag{2.32a}$$

$$\begin{pmatrix} \mathbf{u}^{**} \\ T^{**} \end{pmatrix} = \begin{pmatrix} g^{\mathbf{u}}(\xi,\eta,\zeta,t^{n+1}) \\ g^{T}(\xi,\eta,\zeta,t^{n+1}) \end{pmatrix}, \quad (\xi,\eta,\zeta) \in \partial^{\eta} D, \tag{2.32b}$$

$$\begin{pmatrix} \mathbf{u}^{n+1} \\ T^{n+1} \end{pmatrix} = \begin{pmatrix} g^{\mathbf{u}}(\xi,\eta,\zeta,t^{n+1}) \\ g^{T}(\xi,\eta,\zeta,t^{n+1}) \end{pmatrix}, \quad (\xi,\eta,\zeta) \in \partial^{\zeta} D. \tag{2.32c}$$

2.1.5 Discussion: enforcement of boundary conditions in previous ADI schemes

The discussion presented in this section justifies our use of the boundary values of $Q(t^{n+1})$ in the solution of the intermediate equations and it thus explains the fundamental advantages provided by the Douglas-Gunn splitting in our context.

As discussed in [62], substitution of physical boundary values as boundary conditions for the intermediate (non-physical) variables leads to reductions in the order of

accuracy of the overall solver unless the intermediate time steps in the ADI scheme satisfy certain accuracy conditions. For two spatial dimensions and for a desired overall order s of time-accuracy (i.e., the overall truncation error is $\mathcal{O}((\Delta t)^{s+1})$), for example, the desired full-step order of temporal accuracy can be guaranteed provided the first half time step can be viewed as an $(s-1)$ order accurate discretization (truncation error of order $(\Delta t)^s$) throughout the domain, up to and including the boundary, of a certain "modified" PDE of the form

$$Q_t^* = L^* Q^*; \tag{2.33}$$

see [62]. (The corresponding procedure for three dimensional problems is similar in nature but it involves two intermediate quantities Q^* and Q^{**}—such as those mentioned in Section 2.1.4.3 above. Details concerning boundary condition determination in a particular three-dimensional example are provided in [62, Sec. 6].) The intermediate solution Q^* is then used as initial data in a corresponding modified problem for Q^{n+1}; the accuracy obtained for the intermediate solution Q^* guarantees that the complete time step (comprised of two or three intermediate time steps in two and three dimensions, respectively) yields a solution that coincides with the unsplit discrete solution with an error of order $\mathcal{O}((\Delta t)^{s+1})$.

In fact the necessary time-accurate boundary conditions for Q^* at $t = t^{n+1}$ can be obtained [62] by means of a Taylor expansion of Q^* at $t = t^n$: applying the Taylor series procedure (whereby the PDE is differentiated to obtain subsequent terms in a Taylor series expansion of the solution) to equation (2.33) with initial conditions $Q^* = Q^n$ at $t = t^n$, a truncated power series solution of the appropriate order of accuracy is constructed which is then evaluated at $t = t^{n+1}$ to produce the desired boundary condition for Q^* at that time. This prescription ensures that the errors in boundary values for intermediate variables Q^* are quantities of the appropriate order of time accuracy and, thus, that agreement between the full-step split and

unsplit discrete schemes to order $(\Delta t)^{s+1}$ takes place throughout the domain, up to and including the domain boundary.

The order-s boundary-condition prescriptions provided in [62] are expressed in terms of certain spatial derivatives of the numerical solution Q^n: each subsequent order of time accuracy requires an additional term in the formal power series solution, and, thus, in view of the Taylor series method used, it requires several spatial derivatives of the numerical solution Q^n at the boundary. For the heat equation and the Navier-Stokes equation, for example, two additional spatial derivatives of the numerical solution at the boundary are in principle necessary for each additional order of time accuracy. But, as pointed out in [62], the original PDE can be used to express such derivatives in terms of derivatives with respect to time together with a derivative of the highest order along the boundary (which do not present difficulties as they can be obtained from the boundary conditions) as well as numerical derivatives of the discrete solution Q^n of orders lower than the maximum order of spatial differentiation in the original PDE. In some cases, simplifications can be made such that the resulting expression for Q^* at the boundary is only in terms of the given boundary data for Q at $t = t^n$ and t^{n+1}. For example, in the context of the split (2.22), using equation (2.22b), the boundary condition

$$Q^* = (I + b\,\Delta t\,\mathcal{B})(\,Q(t^{n+1}) - Q^n\,), \quad (\xi, \eta) \in \partial^\xi D \tag{2.34}$$

can be used when Dirichlet data for Q is specified. For the Navier-Stokes equations, however, evaluation of the boundary condition for Q^* requires differentiation of the numerical solution at the boundary of orders as high as the desired order s of temporal accuracy—which could give rise to accuracy losses and, owing to its dependence on solution values at time $t = t^n$, it would give rise to CFL-type constraints in otherwise unconditionally stable implicit solvers.

The Douglas-Gunn scheme is exceptional in that a modified PDE can always be

obtained yielding boundary values at $t = t^{n+1}$ which can be obtained simply and without any recourse to differentiation: the $t = t^{n+1}$ boundary condition implied by this modified PDE for the intermediate variables exactly coincides with the physical boundary conditions at time $t = t^{n+1}$—for any time accuracy order used. In detail, in the Douglas-Gunn scheme the intermediate relation (2.23a) is an order $s-1$ accurate approximation of the full PDE (1.4), with truncation error of order $(\Delta t)^s$. Note that since Q^* is multiplied by Δt in equation (2.23b) this $\mathcal{O}((\Delta t)^s)$ additional error does not change the temporal order of accuracy $(\Delta t)^s$ of the overall scheme carried up to $\mathcal{O}(1)$ time. Thus, the Taylor expansion procedure applied at a boundary point and at time t^n provides solutions at time t^{n+1} which, in view of the smoothness of solutions and prescribed initial boundary data for the original equation (2.1), must satisfy the boundary conditions imposed on the exact solution Q up to an error of the relevant order $(\Delta t)^s$.

Remark 2.3: *It is interesting to note that the boundary conditions for the intermediate variables Q^* in the Douglas-Gunn scheme necessarily coincide with those imposed on the exact solution up to a difference of order $(\Delta t)^s$*—for any type of boundary conditions imposed on Q^{n+1}, *whether of Dirichlet type, Neumann type, Robin type, etc. Indeed, the exact physical solution Q is a solution of the modified PDE for Q^* with an error of the order $(\Delta t)^s$ for $t^n \leq t \leq t^{n+1} = t^n + \Delta t$ and, thus, the full Taylor series in space and time must coincide, up to order $(\Delta t)^s$ and for all orders in the spatial variables not only at the boundary but* throughout the physical domain. *Additionally, since they do not entail boundary or solution values at time $t = t^n$, the Douglas-Gunn boundary conditions do not induce any CFL-type constraints.*

2.2 Stability and quasi-unconditional stability proofs: discussion

Clearly, an implicit solver such as the one presented in this thesis is more expensive per point per time step than an explicit solver. However, using an implicit solver makes sense when the time step restrictions imposed by the mesh spacing h are too severe for the explicit solver—assuming the implicit solver has favorable stability properties, or course. In this section we apply the relevant stability concepts from ODE and PDE theory, and we introduce the new concept of quasi-unconditional stability.

There are many definitions of stability to be found in the literature. Some authors (e.g., [45, 61]) define stability as the uniform boundedness for small enough mesh sizes h and time steps Δt of the solution operators arising from spatial and temporal discretization of the PDE. In this chapter we will instead define stability as the boundedness of solutions in terms of their initial data (see, e.g., [81]): using a norm $|\cdot|$ which quantifies the size of the solution at some fixed point t in time, we say that a scheme is *stable* within some region $\Lambda \subset \{(h, \Delta t) : h > 0,\ \Delta t > 0\}$ of the discretization-parameter space if and only if for any final time $T \in \mathbb{R}$ and all $(h, \Delta t) \in \Lambda$ the estimate

$$|Q^n| \leq C_T \sum_{j=0}^{J} |Q^j| \quad \text{for all } 0 \leq n\Delta t \leq T \tag{2.35}$$

holds for some integer J, for some constant C_T which depends only on T (see [81]). A method is *unconditionally stable* for a given PDE problem if such a constant C_T can be found for all discretization parameters $(h, \Delta t) \in (0, \infty) \times (0, \infty)$.

The *region of absolute stability* R of an ODE scheme is the set of complex numbers $z = \lambda\,\Delta t$ such that the solution of the ODE $y'(t) = \lambda\, y(t)$ is stable with time step Δt. Ideally, one might hope for a method to be stable when the exact solution is stable—i.e., the method is stable for all λ with non-positive real part, regardless of Δt. A

scheme with this property is said to be *A-stable*. In fact, the first- and second order BDF ODE solvers are A-stable, and thus may lead to unconditionally stable methods for linear PDEs. As is well known, however, implicit linear multistep methods of order greater than two are not A-stable (Dahlquist's second barrier [28]). It follows that the BDF schemes of order $s \geq 3$ are not A-stable; nevertheless, we will see that for some problems they may enjoy the property of *quasi-unconditional stability*—a concept that we define below.

Definition 2.1: *Let Ω_h be a family of spatial discretizations of a domain Ω controlled by a mesh-size parameter h and let Δt be a temporal step size. A numerical method for the solution of the PDE $Q_t = \mathcal{P}\,Q$ in Ω is said to be* **quasi-unconditionally stable** *if there exist positive constants M_h and M_t such that the method is stable for all $h < M_h$ and all $\Delta t < M_t$.*

Clearly, quasi-unconditional stability implies that for small enough Δt, the method is stable for arbitrarily fine spatial discretizations. Note that, outside of the region of quasi-unconditional stability, there could be other stability conditions. For example, Figure 2.2 illustrates the concept of quasi-unconditional stability in the parameter space $(h, \Delta t)$ in a case where a CFL type constraint exists outside the window $(0, M_h) \times (0, M_t)$.

In lieu of a full stability analysis for the main problem under consideration (the fully nonlinear compressible Navier-Stokes equations, for which stability analyses are not available for any of the various extant algorithms), in support of the stability behavior observed in our numerical experiments we present rigorous stability results for simpler related problems. In particular, in Section 2.3 we establish the unconditional stability of the Fourier-based BDF2-ADI scheme for linear constant coefficient hyperbolic and parabolic equations in two spatial dimensions. In Section 2.4 we prove quasi-unconditional stability of Fourier-spectral BDF methods ($2 \leq s \leq 6$, without ADI) for the advection-diffusion equation in one- and two-dimensional space using

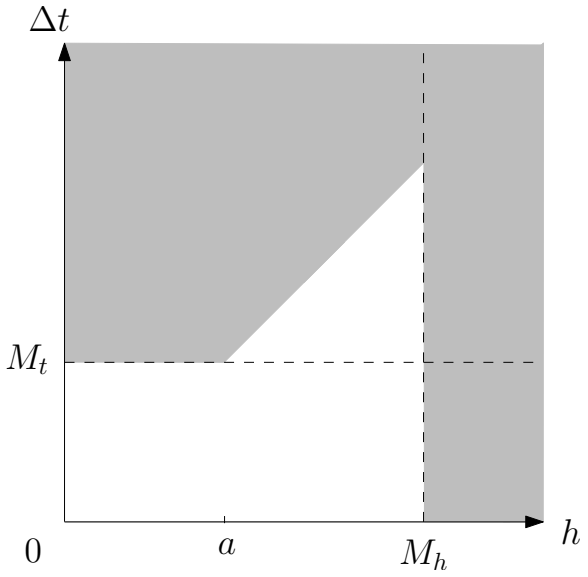


Figure 2.2: The stability region of a hypothetical quasi-unconditionally stable method is shown in white in the parameter space $(h, \Delta t)$. The grey region is the set of h and Δt where the method is unstable. Notice that outside of this window in the region $a < h < M_h$ and $\Delta t > M_t$, the method is stable for time steps satisfying the condition $\Delta t < h$. Quasi-unconditional stability does not exclude the possibility of other stability constraints outside of the rectangular region of stability.

von Neumann stability analysis, together with numerical results that demonstrate quasi-unconditional stability for the full compressible Navier-Stokes equations.

Remark 2.4: *In general, the stability of a PDE solver can be ensured provided relevant discrete operators are power bounded [87]. The von Neumann criterion provides a necessary but not sufficient condition for the power-boundedness of solution operators. If the discrete operators are non-normal, then stability analysis requires application of the Kreiss matrix theorem [45, p. 177]. In particular, it is known [87, 88] that certain discretizations and numerical boundary conditions can give rise to non-normal families of solution operators that are not power-bounded (and unstable) even though the underlying problem is linear with constant coefficients and all eigenvalues are inside the unit disk. (An operator P with adjoint P^* is normal if $P^*P = PP^*$.) In our Fourier-spectral context, however, all operators are normal (which follows from the fact that the first derivative operators are skew-Hermitian, the second derivative*

operators are Hermitian, and all derivative operators commute) and consequently the von Neumann criterion is both necessary and sufficient (cf. [47, p. 189]).

2.3 Stability estimates: linear case, Fourier-BDF2

We use the energy method to prove the unconditional stability of the BDF2-ADI scheme for constant coefficient hyperbolic and parabolic equations with periodic boundary conditions and a Fourier collocation spatial approximation. The proofs in both the parabolic and hyperbolic case rely on the facts that certain boundary terms that arise from integration by parts vanish and that the discrete trapezoidal quadrature based on the underlying spatial discretization is exact for all the integrations used. Therefore, the proof for the parabolic equation is essentially the same if a Legendre collocation spatial approximation is used with homogeneous boundary conditions. However, the same generalization cannot be made for the hyperbolic problem because not all boundary terms from integration by parts cancel (see [43] for a discussion of spectral method stability proofs for hyperbolic problems).

2.3.1 Preliminary definitions

We consider the domain

$$\Omega = [0, 2\pi) \times [0, 2\pi) \tag{2.36}$$

and discretize Ω on the basis of an odd number $N+1$ of discretization points (N even, for definiteness) in both x and y directions ($x_j = 2\pi j/(N+1)$ and $y_k = 2\pi k/(N+1)$, $0 \leq j, k \leq N$), which defines the grid

$$\{(x_j, y_k) : 0 \leq j, k \leq N\}. \tag{2.37}$$

(Our restriction to even values of N is made so as to avoid having to change the limits of the Fourier series (2.40). Similarly, our requirement that equal numbers of points

in both directions simplifies the presentation, but extensions of these constructions that allow for odd values of N as well as unequal numbers of points in the x and y direction are straightforward.)

For (complex valued) grid functions

$$f = \{f_{jk}\} \text{ and } g = \{g_{jk}\}, \quad 0 \le j, k \le N, \tag{2.38}$$

we define the discrete inner product and norm

$$(f, g) = \frac{1}{(N+1)^2} \sum_{j,k} f_{jk} \bar{g}_{jk}, \tag{2.39}$$

$$|f| = \sqrt{(f, f)}.$$

Associated with each grid function f, as in (2.38), is the trigonometric interpolant $f_N(x, y)$ ($f_N(x_j, y_k) = f_{jk}$) which is given by

$$f_N(x, y) = \sum_{|j|,|k| \le \frac{N}{2}} \hat{f}_{jk} e^{i(jx+ky)} \tag{2.40}$$

where

$$\hat{f}_{jk} = \frac{1}{(N+1)^2} \sum_{j,k} f_{jk} e^{-i(jx_j + ky_k)}.$$

Note that the inner product (2.39) coincides with the trapezoidal quadrature rule applied to the grid functions f and g over the underlying domain $[0, 2\pi) \times [0, 2\pi)$. Since the trapezoidal rule (2.39) is exact for all truncated Fourier series containing exponentials of the form $e^{-i(jx+ky)}$ with $-N \le j, k \le N$, it follows that the discrete inner product (2.39) equals the integral inner product of the corresponding trigonometric interpolants—i.e.,

$$(f, g) = \frac{1}{(2\pi)^2} \int_0^{2\pi} \int_0^{2\pi} f_N(x, y) \bar{g}_N(x, y) \, dx \, dy. \tag{2.41}$$

To discretize solutions of time-domain PDEs, we use *time sequences of grid functions* $u = \{u^n : n \geq 0\}$, where, for each n, $u^n = u^n_{jk}$ is a grid function such as those displayed in equation (2.38). For such time series the scalar product (2.39) at fixed n can be used to produce a time series of scalar products: the inner product of two time series of grid functions $u = \{u^n\}$ and $v = \{v^n\}$ is thus a time series of complex numbers:

$$(u, v) = \{(u^n, v^n) : n \geq 0\}.$$

2.3.2 Discrete spatial and temporal operators

The discretization of PDE differential operators proceeds by defining spatial and temporal differentiation operators that act on grid functions and time-series, respectively.

We consider spatial differentiation first: the Fourier x-derivative operator δ_x applied to a grid function f, for example, is defined as the grid function $\delta_x f$ whose jk value equals the value of the derivative of the interpolant f_N at the point (x_j, y_k):

$$(\delta_x f)_{jk} = \frac{\partial}{\partial x} f_N(x_j, y_k). \tag{2.42}$$

The operators δ_{xx}, δ_y, δ_{yy}, $\delta_{xy} = \delta_x \delta_y = \delta_y \delta_x$ etc. are defined similarly.

Using the exactness relation (2.41) and integration by parts together with the periodicity of the domain, it follows that the first derivative operators δ_x and δ_y are skew-Hermitian and the second derivative operators δ_{xx}, δ_{yy} are Hermitian:

$$(\delta_x f, g) = -(f, \delta_x g), \quad (\delta_y f, g) = -(f, \delta_y g), \tag{2.43a}$$

$$(\delta_{xx} f, g) = (f, \delta_{xx} g), \quad (\delta_{yy} f, g) = (f, \delta_{yy} g). \tag{2.43b}$$

Certain temporal differentiation and extrapolation operators we use, in turn, produce new time series from a given numerical time series or time series of grid functions. These operators include the regular first and second order finite difference operators

D and D^2, the three-point backward difference operator $\hat{D}$ that is inherent in the BDF2 algorithm, as well as the second order accurate extrapolation operator "$\sim$":

$$(Du)^n = u^n - u^{n-1} \qquad n \geq 1, \tag{2.44}$$

$$(D^2u)^n = (Du)^n - (Du)^{n-1} = u^n - 2u^{n-1} + u^{n-2} \qquad n \geq 2, \tag{2.45}$$

$$(\hat{D}u)^n = \frac{3}{2}u^n - 2u^{n-1} + \frac{1}{2}u^{n-2} \qquad n \geq 2, \tag{2.46}$$

$$\widetilde{u}^{n+1} = 2u^n - u^{n-1} \qquad n \geq 1. \tag{2.47}$$

Note that the members of the time series $\hat{D}u$ can also be expressed as follows

$$(\hat{D}u)^n = D\left(u^n + \frac{1}{2}(Du)^n\right) \tag{2.48}$$

$$= \frac{1}{2}((Du)^n + (D\widetilde{u})^{n+1}) \tag{2.49}$$

$$= \frac{3}{2}(Du)^n - \frac{1}{2}(Du)^{n-1}. \tag{2.50}$$

In what follows we will make frequent use of the finite difference product rule for two time series u and v:

$$u\,Dv = D(uv) - v\,Du + (Du)\,(Dv). \tag{2.51}$$

One immediate consequence of (2.51), which will also prove useful, concerns the real part of scalar products with a given operator P that is self-adjoint with respect to the discrete inner product (2.39) and which commutes with D:

$$\Re(Du, P\,u) = \frac{1}{2}D(u, P\,u) + \frac{1}{2}(Du, P\,Du). \tag{2.52}$$

This expression follows easily from the relations

$$\begin{aligned}(Du, P\,u) &= D(u, P\,u) - (u, DP\,u) + (Du, DP\,u)\\ &= D(u, P\,u) - (P\,u, Du) + (Du, P\,Du)\\ &= D(u, P\,u) - \overline{(Du, P\,u)} + (Du, P\,Du).\end{aligned}$$

2.3.3 Fourier-based BDF2-ADI stability: hyperbolic equation

This section establishes the unconditional stability of the BDF2-ADI method for the constant-coefficient convection equation

$$U_t + \alpha U_x + \beta U_y = 0 \tag{2.53}$$

in the domain (2.36) with real constants α and β, subject to periodic boundary conditions. The factored form (2.18) of the BDF2-ADI scheme for the Navier-Stokes equations can easily be specialized to the present case. Indeed, using the Fourier collocation approximation described in the previous two sections, setting $\mathcal{A}_s = \alpha\delta_x$ and $\mathcal{B}_s = \beta\delta_y$ (cf. equation (2.42)), letting u denote the discrete approximation of the solution U (that is, letting $Q = u$, $s = 2$ in (2.18)), and noting that the extrapolated term is given by $\widetilde{Q}_1^{n+1} = u^n$, the factored form (2.18) for the BDF2-ADI method applied to the convection equation (2.53) reads

$$(I + b\Delta t\alpha\delta_x)(I + b\Delta t\beta\delta_y)u^{n+1} = a_0u^n + a_1u^{n-1} + \alpha\beta(b\Delta t)^2\delta_x\delta_yu^n. \tag{2.54}$$

Before proceeding to our stability result we derive a more convenient (equivalent) form for equation (2.54): using the numerical values $a_0 = 4/3$, $a_1 = -1/3$, and

$b = 2/3$ of the BDF2 coefficients (Table 2.1), the manipulations

$$\begin{aligned}
0 &= (I + b\Delta t\alpha\delta_x)(I + b\Delta t\beta\delta_y)u^{n+1} - a_0u^n - a_1u^{n-1} - \alpha\beta(b\Delta t)^2\delta_x\delta_y u^n \\
&= u^{n+1} - a_0u^n - a_1u^{n-1} + b\Delta t\alpha\delta_x u^{n+1} + b\Delta t\beta\delta_y u^{n+1} + \alpha\beta(b\Delta t)^2\delta_x\delta_y(u^{n+1} - u^n) \\
&= \frac{1}{b}(u^{n+1} - a_0u^n - a_1u^{n-1}) + \Delta t\alpha\delta_x u^{n+1} + \Delta t\beta\delta_y u^{n+1} + b\alpha\beta(\Delta t)^2\delta_x\delta_y(u^{n+1} - u^n)
\end{aligned}$$

reduce equation (2.54) to the form

$$\hat{D}u + A\,u + B\,u + bAB\,Du = 0, \tag{2.55}$$

where $A = \alpha\Delta t\delta_x$ and $B = \beta\Delta t\delta_y$.

We are now ready to establish an energy stability estimate for the BDF2-ADI equation (2.54).

Theorem 2.1: *The solution u of* (2.54) *with initial conditions u^0 and u^1 satisfies*

$$|u^n|^2 + |\widetilde{u}^{n+1}|^2 + \frac{2}{3}\left(|Au^n|^2 + |Bu^n|^2 + \sum_{m=2}^{n}|(D^2u)^m|^2\right) \le M$$

for all $n \ge 2$, where

$$M = |u^1|^2 + |\widetilde{u}^2|^2 + \frac{2}{3}(|Au^1|^2 + |Bu^1|^2).$$

In particular, the scheme is unconditionally stable in the sense of equation (2.35).

PROOF: Taking the inner product of equation (2.55) with u we obtain

$$\begin{aligned}
0 &= (u, \hat{D}u) + (u, A\,u) + (u, B\,u) + b(u, AB\,Du) \\
&= (I) \qquad + (II) \qquad + (III) \qquad + (IV),
\end{aligned} \tag{2.56}$$

where $(I) = (u, \hat{D}u)$, $(II) = (u, A\,u)$, etc. Our goal is to express the real part of the right-hand side in (2.56) as a sum of non-negative terms and telescoping terms of the

form Df for some non-negative numerical time series f. To that end, we consider the terms (I) through (IV) in turn.

(I): Using the expression (2.49) for $\hat{D}u$ we obtain

$$(I) = \frac{1}{2}(u, Du) + \frac{1}{2}(u, D\widetilde{w}), \tag{2.57}$$

where $\widetilde{w}$ denotes the time series obtained by shifting $\widetilde{u}$ forwards by one time step:

$$\widetilde{w} = \{\widetilde{w}^n = \widetilde{u}^{n+1} : n \geq 1\}. \tag{2.58}$$

To re-express (2.57) we first note that for any two grid functions a and b we have the relation

$$\begin{aligned} |a-b|^2 &= |a|^2 + |b|^2 - 2\Re(a,b) \\ \implies \Re(a,b) &= \frac{1}{2}(|a|^2 + |b|^2 - |a-b|^2). \end{aligned}$$

Therefore, for any time series g we have

$$\begin{aligned} \Re(u, Dg)^n &= \Re(u^n, g^n) - \Re(u^n, g^{n-1}) \\ &= \frac{1}{2}(|u^n|^2 + |g^n|^2 - |u^n - g^n|^2) - \frac{1}{2}(|u^n|^2 + |g^{n-1}|^2 - |u^n - g^{n-1}|^2) \\ &= \frac{1}{2}(D|g^n|^2 - |u^n - g^n|^2 + |u^n - g^{n-1}|^2). \end{aligned} \tag{2.59}$$

Letting $g = u$ and $g = \widetilde{w}$ in (2.59) we obtain

$$\Re(u, Du) = \frac{1}{2}(D|u|^2 + |Du|^2) \tag{2.60}$$

and

$$\Re(u, D\widetilde{w}) = \frac{1}{2}(D|\widetilde{w}|^2 - |Du|^2 + |D^2u|^2). \tag{2.61}$$

Replacing (2.60) and (2.61) in (2.57) we obtain

$$\Re(I) = \frac{1}{4}D(|u|^2 + |\widetilde{w}|^2) + \frac{1}{4}|D^2u|^2. \tag{2.62}$$

Notice that this equation expresses $\Re(I)$ as the sum of a telescoping term and a positive term, as desired.

(II) and (III): The operator A is clearly skew-Hermitian since δ_x is. Therefore

$$\begin{aligned} (II) = (u, Au) = -(Au, u) = -\overline{(u, Au)} \\ \implies \Re(II) = 0. \end{aligned} \tag{2.63}$$

The relation

$$\Re(III) = \Re(u, Bu) = 0 \tag{2.64}$$

follows similarly, of course.

(IV): Lemma 2.1 below tells us that

$$\Re(u, AB\,Du) \geq \frac{1}{4}D\left(|Au|^2 + |Bu|^2\right) - \frac{1}{8}|D^2u|^2. \tag{2.65}$$

Combining equations (2.62), (2.63), (2.64), and (2.65) into equation (2.56) (recalling $b = 2/3$) and taking the real part we obtain

$$0 \geq \frac{1}{4}D\left(|u|^2 + |\widetilde{w}|^2\right) + \frac{1}{6}\left(|Au|^2 + |Bu|^2 + |D^2u|^2\right), \tag{2.66}$$

which is the sum of a telescoping term and a non-negative term. Multiplying by the number four and summing the elements of the above numerical time series from $m = 2$ to n proves the result. ∎

The following lemma concerns the bound (2.65) used in Theorem 2.1.

Lemma 2.1: *Any solution of equation* (2.55) *satisfies* (2.65).

PROOF: Taking the inner product of (2.55) with $A\,Du$ (using the form (2.48) of $\hat{D}u$) we obtain

$$0 = (Du, A\,Du) + \frac{1}{2}(D^2u, A\,Du) + (A\,u, A\,Du) + (B\,u, A\,Du) + b(AB\,Du, A\,Du). \tag{2.67}$$

Since A and B commute and since B is skew-Hermitian (equation (2.43)) we have

$$(B\,u, A\,Du) = -(u, AB\,Du)$$

for the next-to-last term in (2.67). Therefore, equation (2.67) can be rearranged to

$$\begin{aligned}(u, AB\,Du) &= (Du, A\,Du) + \frac{1}{2}(D^2u, A\,Du) + (A\,u, A\,Du) + b(AB\,Du, A\,Du) \\ &= (I) \qquad\quad + (II) \qquad\quad + (III) \qquad + (IV).\end{aligned} \tag{2.68}$$

We consider each term in (2.68) in turn.

(I): Since A is skew-Hermitian it follows that the real part of this term vanishes:

$$\begin{aligned}(I) = (Du, A\,Du) = -(A\,Du, Du) = -\overline{(Du, A\,Du)} = -\overline{(I)} \\ \Longrightarrow \Re(I) = 0.\end{aligned} \tag{2.69}$$

(II): Using Young's inequality

$$ab \le \frac{r}{2}a^2 + \frac{1}{2r}b^2 \tag{2.70}$$

(which, as is easily checked, is valid for all real numbers a and b and for all $r > 0$) together with the Cauchy-Schwarz inequality we obtain

$$\begin{aligned}
\Re(II) &= \frac{1}{2}\Re(D^2u, A\,Du) \\
&\geq -\frac{1}{2}|(D^2u, A\,Du)| \\
&\geq -\frac{1}{2}|D^2u|\,|A\,Du| \\
&\geq -\frac{1}{2}(\frac{1}{4}|D^2u|^2 + |A\,Du|^2) \\
&= -\frac{1}{8}|D^2u|^2 - \frac{1}{2}|A\,Du|^2. \qquad (2.71)
\end{aligned}$$

(III): By the finite-difference product rule (2.51) we obtain

$$\begin{aligned}
(III) \quad &= (Au, D(Au)) \\
&= D(Au, Au) - (DAu, Au) + (DAu, DAu) \\
&= D|Au|^2 - \overline{(III)} + |A\,Du|^2 \\
\Longrightarrow \Re(III) &= \frac{1}{2}D|Au|^2 + \frac{1}{2}|A\,Du|^2. \qquad (2.72)
\end{aligned}$$

(IV): Again using the fact that B is skew-Hermitian and commutes with A it follows that

$$\begin{aligned}
(IV) &= b(BA\,Du, A\,Du) = -b(A\,Du, BA\,Du) = -\overline{(IV)} \\
\Longrightarrow \Re(IV) &= 0. \qquad (2.73)
\end{aligned}$$

Combining the real parts of equations (2.68), (2.69), (2.71), (2.72) and (2.73) we obtain

$$\Re(u, AB\,Du) \geq \frac{1}{2}D|Au|^2 - \frac{1}{8}|D^2u|^2. \qquad (2.74)$$

An analogous result can be obtained by taking the inner product of equation (2.55) with $B\,Du$ instead of $A\,Du$ and following the same steps used to arrive at equa-

tion (2.74). The result is

$$\Re(u, AB\,Du) \geq \frac{1}{2}D|Bu|^2 - \frac{1}{8}|D^2u|^2. \tag{2.75}$$

The lemma now follows by averaging equations (2.74) and (2.75). ∎

2.3.4 Fourier-based BDF2-ADI stability: parabolic equation

This section establishes the unconditional stability of the BDF2-ADI method for the constant-coefficient parabolic equation

$$U_t = \alpha\, U_{xx} + \beta\, U_{yy} + \gamma\, U_{xy}. \tag{2.76}$$

Notice the inclusion of the mixed derivative term, which is treated explicitly using temporal extrapolation in the BDF-ADI algorithm. Theorem 2.2 in this section proves, in particular, that extrapolation of the mixed derivative does not compromise the unconditional stability of the method.

The parabolicity conditions $\alpha > 0$, $\beta > 0$ and

$$\gamma^2 \leq 4\alpha\beta, \tag{2.77}$$

which are assumed throughout this section, ensure that

$$\int_0^{2\pi}\int_0^{2\pi} f\,(\alpha\, f_{xx} + \beta\, f_{yy} + \gamma\, f_{xy})\; dx\, dy \leq 0 \tag{2.78}$$

for any twice continuously differentiable bi-periodic function f defined in the domain (2.36)—as can be established easily by integration by parts and completion of the square in the sum $\alpha(f_x)^2 + \gamma f_x f_y$ together with some simple manipulations. In preparation for the stability proof that is put forth below in this section, in what follows we present a few preliminaries concerning the BDF2-ADI algorithm for equa-

tion (2.76).

We first note that a calculation similar to that leading to equation (2.55) shows that the Fourier-based BDF2-ADI scheme for (2.76) can be expressed in the form

$$\hat{D}u - \Delta t(\alpha\,\delta_{xx} + \beta\,\delta_{yy}u + \gamma\,\delta_x\delta_y)u + \Delta t\,\gamma\,\delta_x\delta_y\,D^2u + b(\Delta t)^2\alpha\beta\,\delta_{xx}\delta_{yy}\,Du = 0. \quad (2.79)$$

Letting

$$\begin{aligned} A &= -\Delta t\,\alpha\,\delta_{xx}, \\ B &= -\Delta t\,\beta\,\delta_{yy}, \\ F &= -\Delta t\,\gamma\,\delta_x\delta_y, \\ L &= A + B + F, \end{aligned}$$

equation (2.79) becomes

$$\hat{D}u + Lu - F\,D^2u + bAB\,Du = 0. \quad (2.80)$$

Note that the operators A and B above do not coincide with the corresponding A and B operators in Section 2.3.3.

In view of the exactness relation (2.41) together with the Fourier differentiation operators (cf. (2.42)), it follows that A, B, AB and L are positive semidefinite operators. Indeed, in view of equation (2.78), for example, we have

$$\begin{aligned} (u, L\,u) &= -\frac{\Delta t}{(2\pi)^2}\int_0^{2\pi}\int_0^{2\pi} u_N\ (\alpha(u_N)_{xx} + \beta(u_N)_{yy} + \gamma(u_N)_{xy})\ dx\,dy \\ &\geq 0; \end{aligned} \quad (2.81)$$

similar relations for A, B and AB follow directly by integration by parts.

Finally we present yet another consequence of the parabolicity condition (2.77)

which will prove useful: for any grid function g we have

$$|Fg|^2 = \gamma^2 (\Delta t)^2 (\delta_x \delta_y g, \delta_x \delta_y g) \leq 4\alpha\beta(\Delta t)^2 (g, \delta_x^2 \delta_y^2 g) = 4(g, ABg). \tag{2.82}$$

Thus, defining the seminorm

$$|u|_P = \sqrt{(u, Pu)} \tag{2.83}$$

for a given positive semidefinite operator P and using $P = AB$ we obtain

$$|Fg|^2 \leq 4|g|_{AB}^2. \tag{2.84}$$

The following theorem can now be established.

Theorem 2.2: *The solution u of the Fourier-based BDF2-ADI scheme* (2.79) *for equation* (2.76) *with initial conditions u^0, u^1 satisfies*

$$\frac{1}{4}|u^n|^2 + \frac{1}{4}|\widetilde{u}^{n+1}|^2 + \frac{1}{3}|(Du)^n|_{AB}^2 + \frac{1}{4}\sum_{m=1}^{n} |D^2 u|^2 + \sum_{m=1}^{n} |u^n|_L^2 \leq M$$

for $n \geq 2$, where

$$\begin{aligned} M =& \frac{1}{4}|u^1|^2 + \frac{1}{4}|\widetilde{u}^2|^2 + \frac{1}{3}|u^1|_{AB}^2 + 3|u^1|_L - \Re(u^1, F\,(Du)^1) \\ &+ 3|(Du)^1|^2 + \frac{3}{2}\left(|(Du)^1|_A^2 + |(Du)^1|_B^2\right) + \frac{1}{3}|(Du)^1|_{AB}^2. \end{aligned}$$

In particular, the scheme is unconditionally stable in the sense of equation (2.35).

PROOF: Taking the inner product of (2.80) with u we obtain

$$\begin{aligned} 0 &= (u, \hat{D}u) + (u, Lu) - (u, F\,D^2 u) + b(u, AB\,Du) \\ &= (I) \qquad + (II) \qquad + (III) \qquad + (IV), \end{aligned} \tag{2.85}$$

where $(I) = (u, \hat{D}u)$, $(II) = (u, Lu)$, etc. As in Theorem 2.1, we re-express the above

equation using telescoping and non-negative terms to obtain the desired energy bound.

The term (I) already occurs in the proof of Theorem 2.1; there we obtained the relation

$$\Re(I) = \frac{1}{4}D(|u|^2 + |\widetilde{w}|^2) + \frac{1}{4}|D^2 u|^2, \tag{2.86}$$

where $\widetilde{w}$ is defined in (2.58). The term $(II) = |u|_L^2$, in turn, is non-negative by equation (2.81) and needs no further treatment. The remaining two terms are considered in what follows.

(III): This term presents the most difficulty, since F is not positive semi-definite. In what follows the term (III) is re-expressed as a a sum of two quantities, the first one of which can be combined with a corresponding term arising from the quantity (IV) to produce a telescoping term, and the second of which will be addressed towards the end of the proof by utilizing Lemma 2.2.

Let v denote the time series obtained by shifting u backwards by one time step:

$$v = \{v^n = u^{n-1} : n \geq 1\}; \tag{2.87}$$

clearly we have

$$Du = u - v \quad \text{and} \quad D^2 u = Du - Dv. \tag{2.88}$$

Thus, using the finite difference product rule (2.51) and the second relation in (2.88) we obtain

$$\begin{aligned}(III) &= -(u, F\, D(Du)) = -(u, D\, F(Du)) \\ &= -D(u, F\, Du) + (Du, F\, Du) - (Du, F\, D^2 u) \\ &= -D(u, F\, Du) + (Du, F\, Dv).\end{aligned}$$

Applying the Cauchy-Schwarz inequality and Young's inequality (2.70) with $r = 6$

together with (2.84) we obtain

$$\begin{aligned}\Re(III) &\geq -D\Re(u, F\,Du) - |Du|\,|F\,Dv| \\ &\geq -D\Re(u, F\,Du) - 3|Du|^2 - \frac{1}{12}|F\,Dv|^2 \\ &\geq -D\Re(u, F\,Du) - 3|Du|^2 - \frac{1}{3}|Dv|^2_{AB}. \end{aligned} \tag{2.89}$$

The last term in the above inequality will be used to form the desired telescoping term with an associated expression in (IV) below.

(IV): Using the finite difference product rule (2.52) together with the fact that AB is a Hermitian positive semi-definite operator we obtain

$$\begin{aligned}\Re(IV) &= \frac{2}{3}\Re(u, AB\,Du) = \frac{2}{3}\Re(Du, ABu) \\ &= \frac{1}{3}D(u, AB\,u) + \frac{1}{3}(Du, AB\,Du) \\ &= \frac{1}{3}D|u|^2_{AB} + \frac{1}{3}|Du|^2_{AB} \end{aligned} \tag{2.90}$$

(see equation (2.83)). Substituting equations (2.86), (2.89) and (2.90) into equation (2.85) and taking real parts, we obtain

$$\begin{aligned}0 \geq &\frac{1}{4}D(|u|^2 + |\widetilde{w}|^2) + \frac{1}{4}|D^2u|^2 + |u|^2_L - D\Re(u, F\,Du) - 3|Du|^2 \\ &+ \frac{1}{3}(|Du|^2_{AB} - |Dv|^2_{AB}) + \frac{1}{3}D|u|^2_{AB} \\ =&D\left(\frac{1}{4}|u|^2 + \frac{1}{4}|\widetilde{w}|^2 + \frac{1}{3}|u|^2_{AB} + \frac{1}{3}|Du|^2_{AB} - \Re(u, F\,Du)\right) \\ &+ |u|^2_L + \frac{1}{4}|D^2u|^2 - 3|Du|^2. \end{aligned} \tag{2.91}$$

Adding the time series (2.91) from $m = 2$ to n and using the identity $\widetilde{w}^n = \widetilde{u}^{n+1}$ we obtain

$$\begin{aligned} M_1 \;\geq\; & \frac{1}{4}|u^n|^2 + \frac{1}{4}|\widetilde{u}^{n+1}|^2 + \frac{1}{3}|u^n|^2_{AB} + \frac{1}{3}|(Du)^n|^2_{AB} + \sum_{m=2}^{n} |u^n|^2_L \\ & + \frac{1}{4}\sum_{m=2}^{n} |(D^2u)^n|^2 - 3\sum_{m=2}^{n} |(Du)^m|^2 - \Re(u^n, F\,(Du)^n) \end{aligned} \tag{2.92}$$

where

$$M_1 = \frac{1}{4}|u^1|^2 + \frac{1}{4}|\widetilde{u}^2|^2 + \frac{1}{3}|u^1|^2_{AB} + \frac{1}{3}|(Du)^1|^2_{AB} - \Re(u^1, F\,(Du)^1).$$

Using Cauchy-Schwarz and Young's inequalities along with the parabolicity relation (2.84) and the fact that F is a Hermitian operator, the last term $-\Re(u^n, F\,(Du)^n)$ in (2.92) is itself estimated as follows:

$$\begin{aligned} -\Re(u^n, F\,(Du)^n) &= -\Re(F\,u^n, (Du)^n) \\ &\geq -|F\,u^n||(Du)^n| \\ &\geq -\frac{1}{12}|F\,u^n|^2 - 3|(Du)^n|^2 \\ &\geq -\frac{1}{3}|u^n|^2_{AB} - 3|(Du)^n|^2. \end{aligned}$$

Equation (2.92) may thus be re-expressed in the form

$$\begin{aligned} \frac{1}{4}|u^n|^2 \;+\; & \frac{1}{4}|\widetilde{u}^{n+1}|^2 + \frac{1}{3}|(Du)^n|^2_{AB} + \sum_{m=2}^{n} |u^n|^2_L + \frac{1}{4}\sum_{m=2}^{n} |D^2u|^2 \\ \leq\; & M_1 + 3|(Du)^n|^2 + 3\sum_{m=2}^{n} |(Du)^m|^2. \end{aligned} \tag{2.93}$$

Finally, applying Lemma 2.2 below to the last two terms on the right-hand side of equation (2.93) we obtain

$$3|(Du)^n|^2 + 3\sum_{m=2}^{n} |(Du)^m|^2 \leq 3M_2,$$

where the constant M_2 is given by equation (2.95). Using this inequality to bound the last two terms in equation (2.93) completes the proof of the theorem. ∎

The following lemma, which provides a bound on sums of squares of the temporal difference Du, is used in the proof of Theorem 2.2 above.

Lemma 2.2: *The solution u of the Fourier-based BDF2-ADI scheme* (2.79) *for equation* (2.76) *with initial conditions u^0, u^1 satisfies*

$$|(Du)^n|^2 + |u^n|_L^2 + \frac{1}{2}\left(|(Du)^n|_A^2 + |(Du)^n|_B^2\right) + \sum_{m=2}^{n} |(Du)^m|^2 \leq M_2 \tag{2.94}$$

for $n \geq 2$, where

$$M_2 = |(Du)^1|^2 + |u^1|_L^2 + \frac{1}{2}\left(|(Du)^1|_A^2 + |(Du)^1|_B^2\right). \tag{2.95}$$

PROOF: We start by taking the inner product of equation (2.80) with Du to obtain

$$\begin{aligned} 0 &= (Du, \hat{D}u) + (Du, Lu) - (Du, F\,D^2u) + b(Du, AB\,Du) \\ &= (I) \qquad + (II) \qquad + (III) \qquad + (IV). \end{aligned} \tag{2.96}$$

We now estimate each of the terms (I) through (IV) in turn; as it will become apparent, the main challenge in this proof is to estimate the term (III).

(I): Using (2.48) and the finite difference product rule (2.52), (I) can be expressed in the form

$$\begin{aligned}\Re(I) &= \Re(Du, Du + \frac{1}{2}D^2u) \\ &= |Du|^2 + \frac{1}{4}D|Du|^2 + \frac{1}{4}|D^2u|^2. \end{aligned} \tag{2.97}$$

(II): Using equation (2.52) we obtain

$$\Re(II) = \Re(Du, Lu) = \frac{1}{2}D(u, Lu) + \frac{1}{2}(Du, L\,Du).$$

Since $L = A + B + F$ we may write

$$\Re(II) = \frac{1}{2}D|u|_L^2 + \frac{1}{2}|Du|_{A+B}^2 + \frac{1}{2}(Du, F\,Du). \tag{2.98}$$

The last term in this equation (which is a real number in view of the Hermitian character of the operator F) will be used below to cancel a corresponding term in our estimate of (III).

(III): Using (2.87) together with the second equation in (2.88), (III) can be expressed in the form

$$\begin{aligned}(III) &= -(Du, F\,D^2u) \\ &= -\frac{1}{2}(Du, F\,Du) + \frac{1}{2}(Du, F\,Dv) - \frac{1}{2}(Du, F\,D^2u). \end{aligned} \tag{2.99}$$

As mentioned in the treatment of (II) above, the first term on the right-hand side of (2.99) will be used to cancel the last term in (2.98). Hence it suffices to obtain bounds for the second and third terms on the right-hand side of equation (2.99).

To estimate the second term in (2.99) we consider the relation

$$\frac{1}{2}(Du, F\,Dv) = \frac{1}{2}\gamma\,\Delta t\,(Du, \delta_x\delta_y Dv) = -\frac{\gamma}{4}\,\Delta t\,(\delta_x\,Du, \delta_y\,Dv) - \frac{\gamma}{4}\,\Delta t\,(\delta_y\,Du, \delta_x\,Dv), \tag{2.100}$$

which follows from the fact that δ_x and δ_y are skew-Hermitian operators. Taking real parts and applying the Cauchy-Schwarz and Young inequalities together with the parabolicity condition (2.77) we obtain

$$\begin{aligned}
\frac{1}{2}\Re(Du, F\,Dv) \geq & -\frac{\sqrt{\alpha\beta}}{2}\,\Delta t\,\left(\frac{1}{2}\sqrt{\frac{\alpha}{\beta}}\,|\delta_x\,Du|^2 + \frac{1}{2}\sqrt{\frac{\beta}{\alpha}}\,|\delta_y\,Dv|^2\right) \\
& -\frac{\sqrt{\alpha\beta}}{2}\,\Delta t\,\left(\frac{1}{2}\sqrt{\frac{\beta}{\alpha}}\,|\delta_y\,Du|^2 + \frac{1}{2}\sqrt{\frac{\alpha}{\beta}}\,|\delta_x\,Dv|^2\right) \\
= & -\frac{1}{4}\Delta t\,(\alpha\,|\delta_x\,Du|^2 + \beta\,|\delta_y\,Du|^2) - \frac{1}{4}\Delta t\,(\alpha\,|\delta_x\,Dv|^2 + \beta\,|\delta_y\,Dv|^2) \\
= & -\frac{1}{4}|Du|^2_{A+B} - \frac{1}{4}|Dv|^2_{A+B}.
\end{aligned} \tag{2.101}$$

To estimate third term in (2.99) we once again use the Cauchy-Schwarz and Young inequalities and we exploit the relation (2.84); we thus obtain

$$\begin{aligned}
-\frac{1}{2}\Re(Du, F\,D^2u) &= -\frac{1}{2}\Re(F\,Du, D^2u) \\
&\geq -\frac{1}{6}|F\,Du|^2 - \frac{3}{8}|D^2u|^2 \\
&\geq -\frac{2}{3}|Du|^2_{AB} - \frac{3}{8}|D^2u|^2.
\end{aligned} \tag{2.102}$$

Taking the real part of (2.99) and using equations (2.101) and (2.102) we obtain the relation

$$\Re(III) \geq -\frac{1}{2}\Re(Du, F\,Du) - \frac{1}{4}|Du|^2_{A+B} - \frac{1}{4}|Dv|^2_{A+B} - \frac{2}{3}|Du|^2_{AB} - \frac{3}{8}|D^2u|^2, \tag{2.103}$$

which, as shown below, can be combined with the estimates for (I), (II), and (IV) to produce an overall estimate that consists solely of non-negative and telescoping

terms—as desired.

(IV): In view of (2.83) we see that (IV) coincides with the P-seminorm of Du with $P = AB$,

$$\Re(IV) = (IV) = \frac{2}{3}|Du|^2_{AB}, \tag{2.104}$$

which, of course, is non-negative, and therefore this term does not require any further treatment.

To complete the proof of the lemma we take real parts in equation (2.96) and we substitute (2.97), (2.98), (2.103) and (2.104); the result is

$$\begin{aligned} 0 \geq & |Du|^2 + \frac{1}{4}D|Du|^2 - \frac{1}{8}|D^2u|^2 + \frac{1}{2}D|u|^2_L + \frac{1}{4}|Du|^2_{A+B} - \frac{1}{4}|Dv|^2_{A+B} \\ = & |Du|^2 - \frac{1}{8}|D^2u|^2 + D\left(\frac{1}{4}|Du|^2 + \frac{1}{2}|u|^2_L + \frac{1}{4}|Du|^2_{A+B}\right). \end{aligned} \tag{2.105}$$

The first two terms on the right-hand-side can be bounded by expanding $|D^2u|^2$ and using Cauchy-Schwarz and Young's inequalities to obtain

$$|Du|^2 - \frac{1}{8}|D^2u|^2 = |Du|^2 - \frac{1}{8}|Du - Dv|^2 \tag{2.106}$$

$$= |Du|^2 - \frac{1}{8}(|Du|^2 + |Dv|^2) + \frac{1}{4}\Re(Du, Dv) \tag{2.107}$$

$$\geq |Du|^2 - \frac{1}{8}(|Du|^2 + |Dv|^2) - \frac{1}{4}|Du||Dv| \tag{2.108}$$

$$\geq |Du|^2 - \frac{1}{4}(|Du|^2 + |Dv|^2) \tag{2.109}$$

$$= \frac{1}{2}|Du|^2 + \frac{1}{4}D|Du|^2. \tag{2.110}$$

Substituting this result into (2.105), we obtain

$$0 \geq \frac{1}{2}|Du|^2 + D\left(\frac{1}{2}|Du|^2 + \frac{1}{2}|u|^2_L + \frac{1}{4}|Du|^2_{A+B}\right), \tag{2.111}$$

which, as needed, is expressed as a sum of non-negative and telescoping terms. Adding

the time-series (2.111) from $m = 2$ to n yields the desired equation (2.94), and the proof is thus complete. ∎

Remark 2.5: *It is interesting to point out that Lemma 2.2 by itself implies a weak stability result that follows from equation* (2.94) *and the Cauchy-Schwarz inequality:*

$$\begin{aligned} |u^n| &= |u^1 + \sum_{m=2}^{n} (Du)^m| \\ &\leq |u^1| + \sum_{m=2}^{n} |(Du)^m| \\ &\leq |u^1| + \left(n \sum_{m=2}^{n} |(Du)^m|^2 \right)^{\frac{1}{2}} \\ &\leq |u^1| + \sqrt{nM_2}, \end{aligned} \tag{2.112}$$

Theorem 2.2 provides a much tighter energy estimate than (2.112)*, of course.*

2.3.4.1 Stability in non-periodic domain with Legendre collocation

The stability result for the parabolic equation can easily be extended to a non-periodic setting using a Legendre polynomial collocation spatial approximation. Here we provide the main necessary elements to produce the extensions of the proofs. Background on polynomial collocation methods may be found, e.g., in [57].

Under Legendre collocation we discretize the domain $\Omega = [-1, 1]\times[-1, 1]$ by means of the $N + 1$ Legendre Gauss-Lobatto quadrature nodes $x_j = y_j$ $(j = 0, \ldots, N)$ in each one of the coordinate directions, which defines the grid $\{(x_j, y_k) : 0 \leq j, k \leq N\}$ (with $x_0 = y_0 = -1$ and $x_N = y_N = 1$). For *real-valued* grid functions $f = (f_{jk})$ and $g = (g_{jk})$ we use the inner product

$$(f, g) = \sum_{j=0}^{N} \sum_{k=0}^{N} w_j w_k f_{jk} g_{jk}, \tag{2.113}$$

where w_ℓ $(0 \leq \ell \leq N)$ are the Legendre Gauss-Lobatto quadrature weights. The

interpolant f_N of a grid function f is a linear combination of the form

$$f_N(x,y) = \sum_{j=0}^{N} \sum_{k=0}^{N} \hat{f}_{jk} P_j(x) P_k(y)$$

of Legendre polynomials P_j.

A certain exactness relation similar to the one we used in the Fourier case exists in the Legendre context as well. Namely, *for grid functions f and g for which the product of the interpolants has polynomial degree $\leq 2N-1$ in the x (resp. y) variable*, the j (resp. k) summation in the inner product (2.113) of the two grid functions is equal to the integral of their corresponding polynomial interpolants with respect to x (resp. y) [47, Sec. 5.2.1]—i.e.,

$$(f,g) = \sum_{k=0}^{N} \int_{-1}^{1} f_N(x,y_k) g_N(x,y_k)\, dx, \tag{2.114a}$$

provided

$$\deg\big(\, f_N(x,y_k) g_N(x,y_k) \,\big) \leq 2N-1 \text{ for all } 0 \leq k \leq N,$$

and

$$(f,g) = \sum_{j=0}^{N} \int_{-1}^{1} f_N(x_j,y) g_N(x_j,y)\, dy, \tag{2.114b}$$

provided

$$\deg\big(\, f_N(x_j,y) g_N(x_j,y) \,\big) \leq 2N-1 \text{ for all } 0 \leq j \leq N.$$

Thus, for example, defining the Legendre x-derivative operator δ_x as the derivative of the Legendre interpolant (cf. (2.42)) (with similar definitions for δ_y, δ_{xx}, δ_{yy} etc.), the exactness relation (2.114a) holds whenever one or both of the grid functions f and g is a Legendre x-derivative of a certain grid function.

A stability proof for the parabolic equation with zero Dirichlet boundary condi-

tions

$$U_t = \alpha\, U_{xx} + \beta\, U_{yy} + \gamma\, U_{xy} \text{ in } \Omega, \quad U = 0 \text{ on } \partial\Omega,$$

can now be obtained by reviewing and modifying slightly the strategy presented for the periodic case in Section 2.3.4. Indeed, the latter proof relies on the following properties of the spatial differentiation operators:

1. The discrete first and second derivative operators are skew-Hermitian and Hermitian, respectively.

2. The operators A, B, L and AB defined in Section 2.3.4 are positive semi-definite.

Both of these results were established using the exactness relation between the discrete and integral inner products together with vanishing boundary terms arising from integration by parts—which also hold in the present case since the exactness relations (2.114) are only ever required to convert inner products involving derivatives, so that the degree of polynomial interpolants will satisfy the requirements of the relations (2.114). Since all other aspects of the proofs in Section 2.3.4 are independent of the particular spatial discretization or boundary conditions used, we have the following theorem:

Theorem 2.3: *The stability estimate given in Theorem 2.2 also holds on the domain* $[-1,1] \times [-1,1]$ *with homogeneous boundary conditions using the Legendre Gauss-Lobatto collocation method, where the inner products and norms are taken to be the Legendre versions instead.*

2.4 Quasi-unconditional stability for higher-order BDF Fourier methods

This thesis does not present stability proofs for the BDF-ADI methods of order higher than two. In order to provide some additional insights into the stability properties

arising from the BDF strategy in the context of time-domain PDE solvers, this section investigates the stability of the BDF schemes of order $s \geq 2$—cf. Remark 2.4 as well as the last paragraph in Section 2.2—under periodic boundary conditions and Fourier discretizations. Because of Dahlquist's second barrier [58, p. 243] the $s \geq 3$ schemes cannot be unconditionally stable for general (even linear) PDEs. However, we will rigorously establish that the BDF methods of order s with $2 \leq s \leq 6$ are *quasi-unconditionally stable* for the advection-diffusion equation—in the sense of Definition 2.1. (As shown in Section 2.3 further, the $s = 2$ algorithms are indeed unconditionally stable, at least for certain linear PDE.)

To introduce the main ideas in our quasi-unconditional stability analysis for BDF-based schemes we consider first a Fourier-BDF scheme for the advection-diffusion equation in one spatial dimension with periodic boundary conditions:

$$\begin{aligned} &U_t + \alpha U_x = \beta U_{xx}, \quad x \in \mathbb{R}, \quad t \geq 0, \\ &U(x,0) = f(x), \quad U(x,t) = U(x+2\pi,t), \end{aligned} \tag{2.115}$$

where $\beta > 0$. Using the N-point Fourier discretization described in Sections 2.3.1 and 2.3.2, the resulting semi-discrete equation is given by

$$\frac{\partial}{\partial t} u = (-\alpha\, \delta_x + \beta\, \delta_x^2)\, u. \tag{2.116}$$

As mentioned in Remark 2.4, the von Neumann criterion provides a necessary and sufficient stability condition for this problem: the scheme is stable if and only if the eigenvalues of the spatial operator in the semi-discrete system (2.116) multiplied by Δt lie within the region R of absolute stability of the BDF method. We will see that these eigenvalues lie on a parabola in the complex plane which does not change as N is varied. To prove quasi-unconditional stability (Definition 2.1) it is therefore sufficient (although not necessary!) to show that a certain family of "complete parabolas" lie in the stability region of the BDF scheme for $\Delta t < M_t$ and $\Delta x < M_h$ for some constants

M_t and M_h. This follows from an application of Lemma 2.3, which establishes that the stability regions of the BDF schemes contain such families of parabolas.

Lemma 2.3: *Let Γ_m be a left-facing parabola passing through the origin in the complex plane with focus $-m/4$ and vertical directrix passing through $+m/4$. Then, for each $2 \leq s \leq 6$, there exists a critical m-value, m_C, such that the parabola Γ_m lies in the stability region R of the BDF method of order s for all $0 \leq m < m_C$.*

PROOF: The A-stability (Section 2.4) of the BDF method of order $s = 2$ implies directly that all left-facing parabolas are contained in R, and thus $m_C = \infty$ in this case. The remaining cases ($3 \leq s \leq 6$) are considered next.

The parabola Γ_m coincides with the set $\Gamma_m = \{\, w \,|\, w = -\frac{1}{m}y^2 - iy,\ y \in \mathbb{R} \,\}$. Let G_m be the set of points to the left of the parabola together with the parabola itself. Clearly, G_m equals the set of points w such that the distance d_1 from w to the focus is less than the distance d_2 from w to the directrix. Equivalently, it is the set of points such that $d_1^2 - d_2^2 \leq 0$, where

$$\begin{aligned} d_1^2 &= |w + m/4|^2 \\ &= (\Re w + m/4)^2 + (\Im w)^2 \end{aligned}$$

and

$$d_2^2 = (\Re w - m/4)^2.$$

It follows that

$$G_m = \{\, w \,|\, (\Im w)^2 + m\,\Re w < 0 \,\}.$$

Let

$$\hat{m}(x) = \sup \left\{ \frac{(\Im w)^2}{-\Re w} \,|\, w \in R,\ \Re w = x \right\}. \tag{2.117}$$

The above definition of $\hat{m}(x)$ is such that the parabola $\Gamma_{\hat{m}}$ intersects the boundary

of the stability region R at a point z_0 with real part x, and the vertical segment $\{\, z | \Re z = x,\ \Im z < \Im z_0 \,\}$ is contained in R. Since $G_{m_1} \subset G_{m_2}$ if and only if $m_1 \leq m_2$, it follows that the the stability region R contains G_m provided $m \leq \hat{m}(x)$ for all negative numbers x. Clearly, the quantity

$$m_C = \inf \{\hat{m}(x) \,|\, x < 0\} \tag{2.118}$$

is the "critical" m value—that is, m_C is the largest such value of m. It follows that $G_m \subset R$ if and only if $m \leq m_C$. We now show that $m_C > 0$ for all BDF methods of orders $3 \leq s \leq 6$.

The BDF methods under consideration are A(0)-stable [58, Ch. 3.12]—that is, using a polar coordinate system with angles measured in the counter-clockwise direction from the positive real axis, there is an angle $\alpha_0 > 0$ such that the wedge $\{r\, e^{i(\pi+\theta)} \,|\, r \geq 0,\ |\theta| \leq \alpha_0\}$ is part of the stability region. Clearly, $w \in R$ provided $\Im w < \Re w \tan \alpha_0$. Therefore, $\hat{m}(x) \geq -x \tan^2 \alpha_0 > 0$ for any $x < 0$ and all that is left is to ensure that $\hat{m}(x)$ is positive in a neighborhood of $x = 0$.

In terms of the stability polynomial [61, p. 153] $p(\zeta, z)$ associated with a given multistep method, R is the set of z such that the roots of p (as a function of ζ) lie in the closed unit disk, with only simple roots on the boundary. For the order-s BDF method, the stability polynomial is given by

$$p(\zeta, z) = (1 - bz)\zeta^s - \sum_{j=0}^{s-1} a_j \zeta^{s-1-j}.$$

Using the boundary locus method [58, Ch. 3.8] we let the boundary of the stability region be given by the implicit relation $p(e^{i\theta}, z) = 0$ for $\theta \in [0, 2\pi]$ (where θ denotes the polar angle in the complex z plane). Solving for z we have

$$z(\theta) = \frac{1}{b}\left(1 - \sum_{j=0}^{s-1} a_j e^{-i(j+1)\theta}\right),$$

which is a complex analytic function of θ in the entire plane. In order to Taylor-expand the function $z = z(\theta)$ we first note that $z(0) = 0$—in view of the temporal consistency of the scheme. The Taylor expansion of the function $z(\theta)$ around $\theta = 0$ is thus given by

$$z(\theta) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{(2n)!} \left(\frac{1}{b} \sum_j a_j (j+1)^{2n} \right) \theta^{2n} + i \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} \left(\frac{1}{b} \sum_j a_j (j+1)^{2n+1} \right) \theta^{2n+1}.$$

Using (2.9) and Table 2.1 a direct computation also shows that

$$\frac{1}{b} \sum_j a_j (j+1) = 1$$
$$\frac{1}{b} \sum_j a_j (j+1)^n = 0, \quad 2 \le n \le s$$

for $2 \le s \le 6$.

s	3	4	5	6
$\Re z(\theta) \sim$	$-\frac{1}{4}\theta^4$	$-\frac{1}{3}\theta^6$	$\frac{1}{6}\theta^6$	$\frac{3}{8}\theta^8$

Table 2.2: Leading order term for the real part of $z(\theta)$, the boundary locus of the BDF method of order s stability region as $\theta \to 0$.

It follows that the leading order term for the imaginary part of $z(\theta)$ is 1. The leading order terms for the real part, in turn, are given in Table 2.2. For $s = 5, 6$, the leading order term is positive, indicating that, near the origin, the boundary locus opens towards the right of the origin. Therefore, the stability region contains a segment of the imaginary axis near the origin. It follows that $\hat{m}(x) \to \infty$ as $x \to 0$,

and, thus, m_C is positive. For $s = 3, 4$, it also follows that

$$\hat{m}(0) = \lim_{\theta \to 0} \frac{(\Im z(\theta))^2}{-\Re z(\theta)}$$
$$= \infty.$$

Therefore the critical constant m_C is positive for BDF schemes of order $3 \leq s \leq 6$ and the proof is complete. ∎

Remark 2.6: *Equations* (2.117) *and* (2.118) *can be used to evaluate numerically the constant m_C for any given s; the results for each BDF method of orders 3 through 6 are summarized in Table 2.3.*

s	3	4	5	6
m_C	14.0	5.12	1.93	0.191

Table 2.3: Numerical estimate of the constant m_C such that for all $m < m_C$ the parabola Γ_m described in Lemma 2.3 is contained in the region of absolute stability of the BDF method of order s. By Theorem 2.4, the order-s BDF method applied to the advection-diffusion equation $u_t + \alpha\, u_x = \beta\, u_{xx}$ with Fourier collocation is stable for all $\Delta t < \frac{\beta}{\alpha^2} m_C$

.

Theorem 2.4: *Let $2 \leq s \leq 6$. The solution of the problem* (2.115) *by the s-th order Fourier-based BDF scheme described in this section is quasi-unconditionally stable, with $M_h = \infty$ and $M_t = \frac{\beta}{\alpha^2} m_C$ ($M_t = \infty$ for $\alpha = 0$), where m_C is a real constant which depends only on s.*

PROOF: Applying the discrete Fourier transform,

$$\hat{u}_k = \frac{1}{N+1} \sum_{j=0}^{N} u_j e^{-i x_j k}, \quad -\frac{N}{2} \leq k \leq \frac{N}{2},$$

to equation (2.116), we have the set of ODEs

$$\frac{\partial}{\partial t}\hat{u}_k = -(i\alpha k + \beta k^2)\,\hat{u}_k \tag{2.119}$$

for the Fourier coefficients $\hat{u}_k$. It is clear from this transformed equation that the eigenvalues of the spatial operator for the semi-discrete system are given by

$$\lambda(k) = -(i\alpha k + \beta k^2). \tag{2.120}$$

Using von Neumann stability analysis, it is sufficient to show that these eigenvalues multiplied by Δt lie in the stability region of the BDF method.

Let $z = \lambda\,\Delta t$ where $\lambda = \lambda(k)$ is an eigenvalue of the semi-discrete system (2.116). If $\alpha = 0$, then z is a non-positive real number. In view of the A(0)-stability of the BDF methods, we immediately see that the methods are unconditionally stable in this case.

Let us now consider the case $\alpha \neq 0$. We must find the values of Δt for which the complex numbers

$$\begin{aligned} z_k &= -\beta\,\Delta t\,k^2 - i\alpha\,\Delta t\,k \\ &= -\frac{\beta}{\alpha^2\Delta t}(\alpha\,\Delta t\,k)^2 - i(\alpha\,\Delta t\,k) \\ &= -\frac{1}{\frac{\alpha^2\Delta t}{\beta}}(\alpha\,\Delta t\,k)^2 - i(\alpha\,\Delta t\,k) \quad \text{with} \quad -\frac{N}{2} \le k \le \frac{N}{2} \end{aligned} \tag{2.121}$$

lie in the stability region R. But from (2.121) it is clear that z lies on the set $\Gamma_m = \{\, w \,|\, w = -\frac{1}{m}y^2 - iy,\ y \in \mathbb{R} \,\}$ with $m = \frac{\alpha^2\Delta t}{\beta}$, which is a left-facing parabola with focus $-m/4$ and vertical directrix passing through the point $+m/4$. By Lemma 2.3, it follows that the parabola lies in the stability region R for all m satisfying

$$m < m_C \quad \Longrightarrow \quad \Delta t < \frac{\beta}{\alpha^2} m_C.$$

Furthermore, the above condition holds for all spatial discretizations, so that $M_h = \infty$ and the proof is complete. ∎

We now establish the quasi-unconditional stability of the BDF methods applied to the two- and three-dimensional advection diffusion equation

$$u_t + \boldsymbol{\alpha} \cdot \nabla u = \beta \Delta u, \quad \text{in } [0, 2\pi]^d, \ d = 2, 3 \tag{2.122}$$

with periodic boundary conditions, where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^{\mathrm{T}}$ and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)^{\mathrm{T}}$ for $d = 2$ and 3 respectively. To define a single mesh-size parameter h and quasi-unconditional stability constant M_h, we fix positive integers r_1 and r_2 and discretize the domain with $N_x + 1$ points in the x direction, $N_y + 1 = r_1 N_x + 1$ points in the y direction, and $N_z + 1 = r_2 N_x + 1$ points in the z direction (N_x even). The mesh size parameter is then $h = 2\pi/(N_x + 1)$.

Theorem 2.5: *The solution of* (2.122) *using the BDF scheme of order s (not ADI!) with $3 \leq s \leq 6$ and the Fourier collocation discretization is quasi-unconditionally stable with constants $M_t = \frac{|\boldsymbol{\alpha}|^2}{\beta} m_C$ and $M_h = \infty$.*

PROOF: We present the proof for the $d = 2$ case; the same procedure yields the $d = 3$ result.

Substituting the Fourier series

$$u(x, y) = \sum_{k=-\frac{N_x}{2}}^{\frac{N_x}{2}} \sum_{\ell=-\frac{N_y}{2}}^{\frac{N_y}{2}} \hat{u}_{k\ell} e^{i(kx+\ell y)}$$

into equation (2.122), we have the system of ODEs for the Fourier coefficients $\hat{u}_{k\ell}$

$$\frac{\partial \hat{u}_{k\ell}}{\partial t} = \left(-i(\alpha_1 k + \alpha_2 \ell) - \beta(k^2 + \ell^2)\right) \hat{u}_{k\ell}.$$

It follows that the eigenvalues of the semi-discrete system are

$$\lambda_{k\ell} = -i(\alpha_1 k + \alpha_2 \ell) - \beta(k^2 + \ell^2),$$

which clearly do not all lie on a parabola as in the one dimensional case. In fact, the eigenvalues are on a family of parabolas: the subset of eigenvalues for fixed ℓ as a function of k lie on a parabola centered at $-i\alpha_2\ell - \beta\ell^2$. Nevertheless, it suffices to show that there exists one parabola that bounds all the eigenvalues to its left.

Let $\xi \geq 0$, $\xi^2 = k^2 + \ell^2$, and relax for the moment the assumption that k and ℓ are integers, allowing them to vary continuously. For each fixed ξ, we wish to find the eigenvalue with the largest imaginary part in magnitude—i.e., letting $f(k, \xi) = \alpha_1 k + \alpha_2\sqrt{\xi^2 - k^2}$, we solve

$$\max_{|k| \leq \xi} f(k, \xi).$$

Assume that $\alpha_1 > 0$ and $\alpha_2 > 0$ (the extension to other values of α_1 and α_2 is straightforward). The critical points of f with respect to k are given by

$$\frac{\partial f}{\partial k} = \alpha_1 k - \frac{\alpha_2 k}{\sqrt{\xi^2 - k^2}} = 0$$

and the solutions are

$$k = \pm \frac{\alpha_1 \xi}{\sqrt{\alpha_1^2 + \alpha_2^2}},$$

which satisfy $|k| \leq \xi$. The function f achieves its maximum value with the positive root of k above, and is given by

$$\max_{|k| \leq \xi} f(k, \xi) = |\boldsymbol{\alpha}|\xi.$$

It follows that the eigenvalues are bounded on the right by the parabola

$$\Gamma = \{\ i|\boldsymbol{\alpha}|\xi - \beta\xi^2 \mid \xi \in \mathbb{R}\ \}$$

regardless of the values of N_x and N_y. The theorem now follows from an application of Lemma 2.3. ∎

2.4.1 Order-s BDF methods outside the region of quasi-unconditional stability

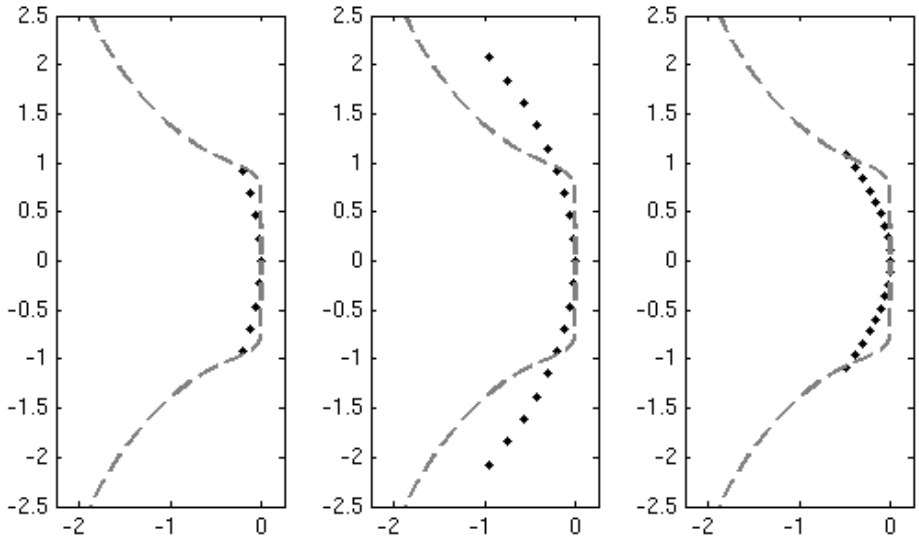
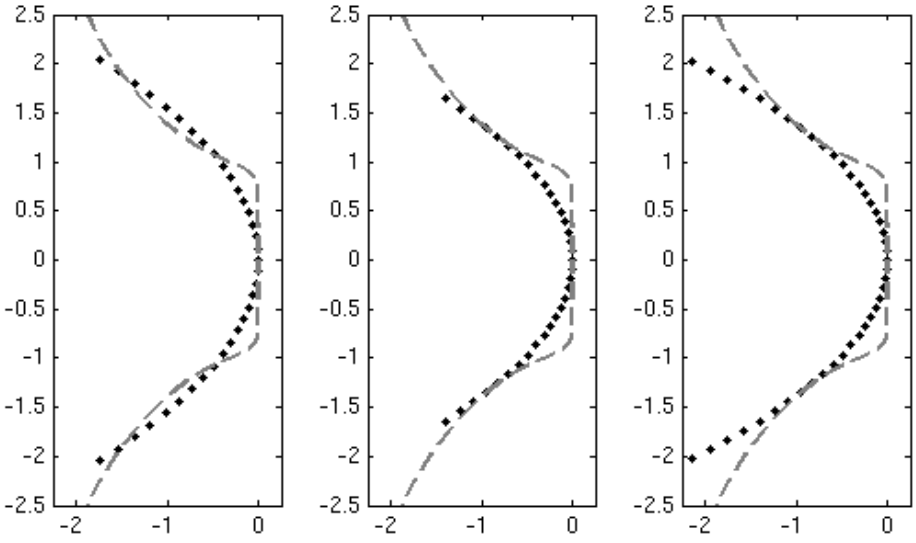


Figure 2.3: Demonstration of a CFL-like stability constraint when Δt is outside the rectangular window of quasi-unconditional stability for the advection-diffusion equation with $\alpha = 1.0$ and $\beta = 0.05$ (parameters selected for clarity of visualization. Theoretical value: $M_t = 0.0965$ for this selection of physical parameters). The eigenvalues multiplied by Δt (black dots) are plotted together with the boundary of the BDF5 stability region (dashed grey curve; cf. Figure 2.1). (a) Using $N + 1 = 9$ grid points and time step $\Delta t = 0.23$ all eigenvalues lie within the stability region. (b) The number of points is increased to $N + 1 = 19$ while the time step is held constant. The ten additional eigenvalues are not in the stability region, which indicates the method is unstable for these parameter values. (c) The number of points is again $N + 1 = 19$, but the time step is reduced to $\Delta t = 0.12$, causing all eigenvalues to be contained in the stability region.

Theorems 2.4 and 2.5 should not be viewed as a suggestion that the s-th order BDF methods are not stable when the constraints in the theorem are not satisfied.



(a) (b) (c)

Figure 2.4: Continuation of Figure 2.3. (a) The time step is $\Delta t = 0.12$ (as in Figure 2.3(c)) and the number of grid points is increased to $N+1 = 35$. Once again, some eigenvalues do not lie in the stability region. (b) The number of grid points is held at $N + 1 = 35$ while the time step is reduced to $\Delta t = M_t = 0.0965$, which is the maximum allowed for the window of quasi-unconditional stability. All eigenvalues now lie in the stability region. (c) With $\Delta t = 0.0965$, additional eigenvalues (arising from further increasing the number of grid points) remain within the stability region, thus demonstrating the quasi-unconditional stability of the BDF scheme of order 5.

Indeed, while, by definition, for $\Delta t > M_t$ the complete parabolic region Γ_m passes through the region where the BDF method is unstable (as demonstrated in Figure 2.3 as well as in the first two images in Figure 2.4), stability can still be ensured for such a value of Δt provided adequate values of the discretization parameter $N+1 = 2\pi/h$ are used. Indeed, taking into account that only a bounded segment in the parabola is actually relevant to the stability of the ODE system that results for each fixed value of N, we see that stability may be ensured provided this particular segment, and not necessarily the complete parabola Γ_m, is contained in the stability region of the s-th order BDF algorithm.

From equation (2.121) we see that increasing values of N lead to corresponding increases in the length of the parabolic segment on which the eigenvalues actually lie,

while decreasing Δt results in reductions of both the length of the relevant parabolic segment as well as the width of the parabola itself. Therefore, for $\Delta t > M_t$, increasing the number of grid points will inevitably cause some eigenvalues to eventually enter the region of instability. But stability can be restored by a corresponding reduction in Δt—see Figure 2.3. This argument suggests that a CFL condition of the form $\Delta t \leq C/N$ exists for $\Delta t > M_t$. Of course, when Δt is reduced to the value M_t or below, then no increases in N (reductions in h) result in instability—as demonstrated in Figure 2.4. We may thus emphasize: within the quasi-unconditional stability window no such CFL-like stability constraints exist.

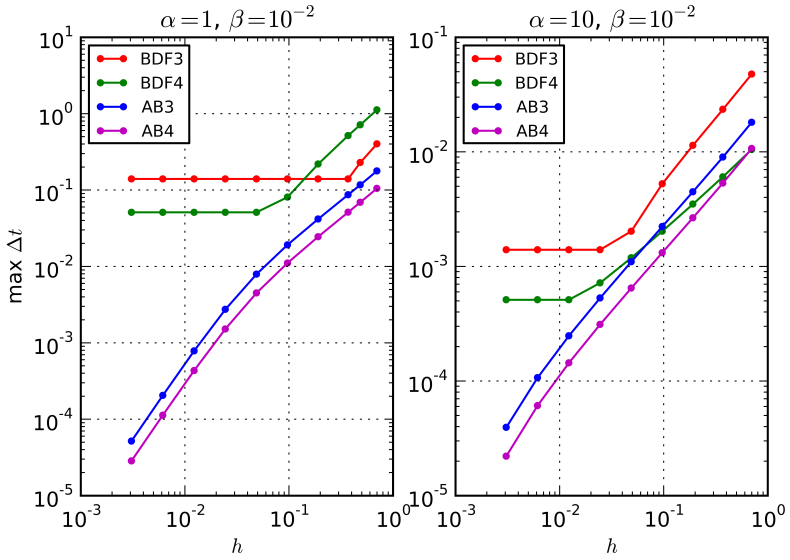


Figure 2.5: Maximum stable Δt versus spatial mesh size h for Fourier-based BDF and AB methods of orders three and four when applied to the advection-diffusion equation (2.115), with $\alpha = 1$, $\beta = 10^{-2}$ on the left and $\alpha = 1$, $\beta = 10^{-2}$ on the right.

To better understand when the BDF methods are preferable to an explicit scheme, we compare their stability to that of the explicit Adams-Bashforth (AB) multistep methods. For a given number of discretization points $N + 1$ and physical parameter values α and β, we can use the equation for the eigenvalues (2.120) of the advection-diffusion equation and the boundary locus $z(\theta)$ of the stability regions to estimate the maximum stable Δt by solving $z(\theta) = \lambda(N/2)\,\Delta t$ for θ and Δt.

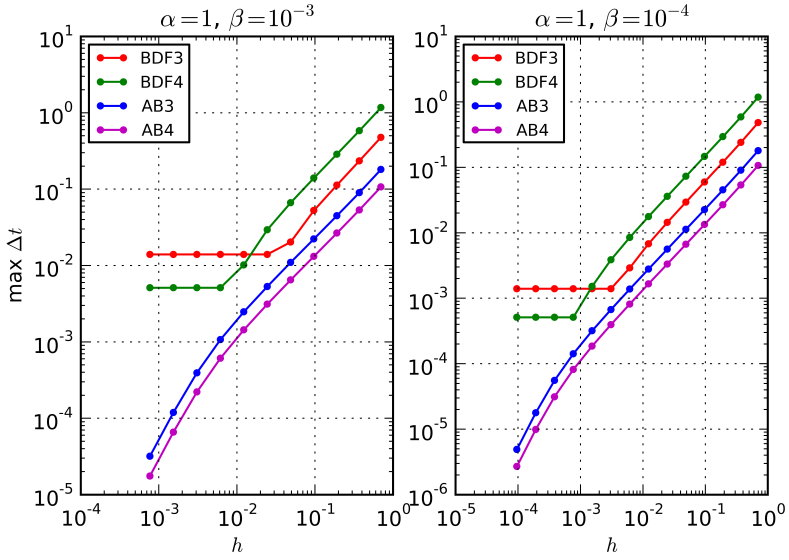


Figure 2.6: Maximum stable Δt versus spatial mesh size h for Fourier-based BDF and AB methods of orders three and four when applied to the advection-diffusion equation (2.115), with $\alpha = 1$, $\beta = 10^{-3}$ on the left and $\alpha = 1$, $\beta = 10^{-4}$ on the right.

Figures 2.5 and 2.6 show the maximum stable Δt allowed by the Fourier-based BDF and AB methods of orders three and four for the advection-diffusion equation, with various discretizations and values of the parameters α and β. We observe from the $\alpha = 1$ plots that both the BDF and AB methods have a CFL-type constraint of the form $\Delta t < Ch$ for large values of h. When h is decreased to the order of β the CFL condition for the explicit method becomes more severe ($\Delta t < Ch^2$). By this point, the BDF methods have already entered the of window quasi-unconditional stability. At $h = \beta$, the stable Δt for the BDF methods are about one hundred times larger than their AB counterparts. Clearly, the BDF methods are preferable in regimes where the AB methods suffer from the severe $\Delta t < Ch^2$ CFL condition. However, as we will see in Tables 2.4, 2.5 and 2.6 in the next section, much higher stable Δt can be achieved in practice by the BDF-ADI methods for the full Navier-Stokes equations in two dimensions than suggested by the linear stability analysis.

2.4.2 Quasi-unconditional stability: linearized and full Navier-Stokes equations

To further extend the analysis provided in the previous sections for the advection-diffusion equation, here we consider the Navier-Stokes equations linearized about a temperature and density equal to 1 and a non-dimensional constant velocity U in the x direction, so that $T = 1 + \widetilde{T}$, $\rho = 1 + \widetilde{\rho}$, $u = U + \widetilde{u}$, $v = 0$, and $w = 0$, where the perturbations are denoted with a tilde. Inserting these expressions into (1.4) and neglecting terms that are quadratic in the perturbations, the linearized equations

$$\widetilde{\rho}_t + U\,\widetilde{\rho}_x + \widetilde{u}_x = 0 \tag{2.123a}$$

$$\widetilde{u}_t + U\,\widetilde{u}_x + \frac{1}{\gamma\,\mathrm{Ma}^2}(\widetilde{T}_x + \widetilde{\rho}_x) = \frac{4}{3}\frac{1}{\mathrm{Re}}\widetilde{u}_{xx} \tag{2.123b}$$

$$\widetilde{T}_t + U\,\widetilde{T}_x + (\gamma - 1)\widetilde{u}_x = \frac{\gamma}{\mathrm{Re}\,\mathrm{Pr}}\widetilde{T}_{xx} \tag{2.123c}$$

result; in what follows the tildes are dropped for simplicity. The above equations can thus be re-expressed in the matrix form

$$Q_t = L\,Q,$$

where $Q = (\rho,\, u,\, T)^{\mathrm{T}}$,

$$L = -\left(M_1\,\frac{\partial}{\partial X} + \frac{1}{\mathrm{Re}}\,M_2\,\frac{\partial^2}{\partial X^2}\right),$$

and the matrices M_1 and M_2 are given by

$$M_1 = \begin{pmatrix} U & 1 & 0 \\ \frac{1}{\gamma\,\mathrm{Ma}^2} & U & \frac{1}{\gamma\,\mathrm{Ma}^2} \\ 0 & (\gamma-1) & U \end{pmatrix} \quad \text{and} \quad M_2 = \mathrm{diag}\left(0, \frac{4}{3}, \frac{\gamma}{\mathrm{Pr}}\right).$$

Take the domain to be $x \in [0, 2\pi]$ with periodic boundary conditions and consider solutions of the form $\hat{Q} = (\hat{\rho}, \hat{u}, \hat{T})^{\mathrm{T}}$, with $\hat{\rho} = \hat{u} = \hat{T} = \exp(\, ikx\,)$ for some integer k. These solutions satisfy

$$\hat{Q}_t = \left(-i\, M_1\, k - \delta\, M_2\, k^2\right) \hat{Q}., \tag{2.124}$$

where $\delta = \mathrm{Re}^{-1}$.

To carry the analysis further, we use perturbation theory in the limit of small δ (large Reynolds number), which is consistent with the continuum fluid approximation and with non-negligible compressibility effects. The continuum limit is characterized by small Knudsen number ($\mathrm{Kn} \ll 1$), which is the ratio of the mean free path of the molecules in the gas to the macroscopic length scale of the problem. By the von Karman relation [78, p. 60], the Knudsen number is proportional to Ma/Re. Accordingly, for the present analysis we assume $\mathrm{Ma}\,\mathrm{Re}^{-1} \ll 1$ and we linearize the equations accordingly. Further, compressibility effects become negligible as $\mathrm{Ma} \to 0$. Therefore, we also take the Mach number to be of order one or greater, which together with the von Karman relation implies $\delta = \mathrm{Re}^{-1} \ll 1$.

The eigenvalues of the matrix in parenthesis on the right-hand side of equation (2.124) can now be approximated by means of a perturbation series in δ. Let $\lambda = \sum_{j=0}^{\infty} \delta^j \bar{\lambda}_j$ be such an eigenvalue, i.e., it is a root of the characteristic equation

$$\det\left(\lambda I - i\, M_1\, k + \delta\, M_2\, k^2\right) = 0. \tag{2.125}$$

Dropping terms of order δ and higher, we recover the eigenvalues of the linearized Euler equations:

$$\bar{\lambda}_0 \in \left\{ iUk,\, i\left(U + \frac{1}{\mathrm{Ma}}\right) k,\, i\left(U - \frac{1}{\mathrm{Ma}}\right) k \right\}.$$

Expanding equation(2.125), keeping terms up to first order in δ and solving for $\bar{\lambda}_1$ we

have

$$\bar{\lambda}_1 = \frac{\bar{\lambda}_0 \mathrm{Ma}^2 \left(\gamma + \frac{4}{3}\mathrm{Pr}\right)\left(2iUk - \bar{\lambda}_0\right) k^2 + \mathrm{Ma}^2 U^2 \left(\left(\gamma + \frac{4}{3}\mathrm{Pr}\right) - 1\right) k^4}{\mathrm{Pr}\left(3\mathrm{Ma}^2 U^2 - 1\right) k^2 + \bar{\lambda}_0 \mathrm{Ma}^2 \mathrm{Pr}\left(2iUk - \bar{\lambda}_0\right)}.$$

The $\bar{\lambda}_1$ are determined by substituting the three different values of $\bar{\lambda}_0$. The eigenvalues of L corresponding to the eigenvector $\hat{Q}$ can then be approximated as

$$\lambda_1(k) = iUk + \delta \frac{1}{\mathrm{Pr}} k^2 + \mathcal{O}(\delta^2), \tag{2.126a}$$

$$\lambda_2(k) = i\left(U + \frac{1}{\mathrm{Ma}}\right) k + \delta \left(\frac{2}{3} + \frac{\gamma - 1}{2\,\mathrm{Pr}}\right) k^2 + \mathcal{O}(\delta^2), \tag{2.126b}$$

$$\lambda_3(k) = i\left(U - \frac{1}{\mathrm{Ma}}\right) k + \delta \left(\frac{2}{3} + \frac{\gamma - 1}{2\,\mathrm{Pr}}\right) k^2 + \mathcal{O}(\delta^2). \tag{2.126c}$$

For each λ, the first term is the leading order term of the imaginary part of the eigenvalue, and the second term is the leading order term of the real part. Comparing (2.126) to (2.120), we see that the asymptotic behavior for large Reynolds number is like that of the advection-diffusion equation. The first eigenvalue corresponds to convective waves and the second two eigenvalues to acoustic waves. Extending the analysis further, we can estimate the constant M_t of quasi-unconditional stability for the linearized Navier-Stokes equations:

$$M_t \sim \frac{4\mathrm{Pr} + 3(\gamma - 1)}{6(|\mathbf{u}|_{\max} + \mathrm{Ma}^{-1})^2 \mathrm{Pr}\,\mathrm{Re}} m_C. \tag{2.127}$$

Tables 2.4, 2.5 and 2.6 show the numerically estimated maximum stable Δt for the full Navier-Stokes equations in two dimensional space, using the Chebyshev collocation method described in section 2.5 in the unit square $(x, y) \in [0, 1] \times [0, 1]$ at Mach number 0.9 and various Reynolds numbers and discretizations. The initial condition is $\mathbf{u} = 0$, $\rho = T = 1$ and a source term of the form

$$f(x, y, t) = A\,\sin(2\pi t)\,\exp\left(-\frac{1}{2\sigma^2}\left((x - x_0)^2 + (y - y_0)^2\right)\right)$$

	$s = \ldots$				
N_y	2	3	4	5	6
12	6.4e-1	3.4e-1	5.9e-2	3.4e-2	1.5e-2
16	6.3e-1	2.7e-1	5.0e-2	2.4e-2	9.9e-3
24	6.3e-1	1.1e-1	4.5e-2	1.9e-2	6.1e-3
32	6.3e-1	9.2e-2	3.7e-2	1.7e-2	5.2e-3
48	6.3e-1	7.2e-2	3.2e-2	1.7e-2	5.1e-3
M_t	∞	3.0e-2	1.1e-2	4.1e-3	4.1e-4

Table 2.4: Maximum stable Δt for BDF-ADI methods of orders $s = 2, \ldots, 6$ at Reynolds number $\mathrm{Re} = 10^2$ and Mach number 0.9 in 2D with various numbers N_y of discretization points in the y variable. The number of discretization points in the x direction is fixed at $N_x = 12$. The constant M_t of quasi-unconditional stability predicted by the linear theory (equation (2.127)) is given in the last row.

	$s = \ldots$				
N_y	2	3	4	5	6
12	4.5e-1	8.5e-2	2.9e-2	2.5e-2	1.2e-2
16	4.5e-1	7.4e-2	2.2e-2	1.5e-2	7.5e-3
24	4.5e-1	6.2e-2	1.4e-2	8.2e-3	3.6e-3
32	4.5e-1	6.0e-2	1.1e-2	5.7e-3	2.2e-3
48	4.5e-1	6.0e-2	7.8e-3	4.0e-3	1.0e-3
M_t	∞	3.0e-3	1.1e-3	4.1e-4	4.1e-5

Table 2.5: Same as Table 2.4 but with Reynolds number $\mathrm{Re} = 10^3$.

	$s = \ldots$				
N_y	2	3	4	5	6
12	4.8e-1	2.5e-2	2.5e-2	2.3e-2	1.1e-2
16	4.7e-1	1.8e-2	2.0e-2	1.3e-2	4.7e-3
24	4.5e-1	1.2e-2	1.0e-2	7.2e-3	Q
32	4.5e-1	1.0e-2	8.7e-3	4.5e-3	Q
48	4.4e-1	5.9e-3	5.5e-3	Q	Q
M_t	∞	3.0e-4	1.1e-4	4.1e-5	4.1e-6

Table 2.6: Same as Table 2.4 but with Reynolds number $\mathrm{Re} = 10^4$. A "Q" in the table means there was no stable Δt found for the given discretization. However, using 16 points in the x direction, all entries in the table can be filled, which is an indication of the quasi-unconditional stability of the method.

is added to the u equation, with parameters $A = 6.0$, $\sigma^2 = 0.05$, $x_0 = y_0 = 0.5$. The boundary conditions at $y = 0, 1$ are no-slip isothermal ($\mathbf{u} = 0$, $T = 1$) and there is a sponge layer (see equation (4.3)) of thickness 0.1 and amplitude 2.0 at $x = 0, 1$. The method was determined to be stable for a given Δt if the solution did not blow up for 20000 time steps or for the number of time steps required to exceed $t = 100$, whichever was greater.

Tables 2.4, 2.5 and 2.6 suggest that quasi-unconditional stability indeed exists even for the full Navier-Stokes equations. Furthermore, we observe that the BDF-ADI methods seem to enjoy stability for Δt far greater than those predicted by the linear stability analysis. In particular, the tables suggest that the BDF-ADI methods may be particularly advantageous when a very fine mesh is only required in one of the spatial dimensions.

2.5 Numerical implementation

In this section we present a spatial discretization for the semi-discrete BDF-ADI schemes considered in this chapter and in Section 4.1. Spatial discretizations of various kinds can be used in our context, including finite-difference, polynomial-spectral, and Fourier-continuation discretizations (as in the following chapter). For the sake of definiteness we restrict our treatment to the Chebyshev-collocation spatial approximation, which is the discretization we use for the numerical examples of the BDF-ADI method in single domains (see Section 4.1). Details concerning Chebyshev collocation can be found, e.g., in [10,57]; the text in the following section includes a brief overview as well as a few specifics associated with our two-dimensional implementation. Extensions to the three-dimensional case do not present any additional difficulties. We also note that many of the same elements described in the framework below apply to FC and other spatial discretizations as well.

2.5.1 Spectral collocation

The Fourier-spectral collocation method was described in Sections 2.3.1 and 2.3.2; here we briefly outline the corresponding Chebyshev discretization that also is incorporated in our test solvers. For simplicity the description in this section is restricted to two spatial dimensions and it assumes the domain is discretized by means of $N+1$ Gauss-Lobatto Chebyshev nodes in both the ξ and η directions:

$$\begin{aligned} \xi_i &= -\cos(\pi i/N) \quad , \quad i = 0, \ldots, N \\ \eta_j &= -\cos(\pi j/N) \quad , \quad j = 0, \ldots, N. \end{aligned}$$

The actual solvers we use are not so restricted: they are applicable to two- and three-dimensional problems, and they can use different numbers of discretization points in the various coordinate directions.

By analogy with Section 2.3.1, grid functions $Q_{ij} \sim Q(\xi_i, \eta_j)$ for the solution Q are used together with the associated Lagrange interpolating polynomials $Q_N(\xi, \eta)$. The discrete ξ and η spatial differentiation operators, in turn, are given by the Chebyshev derivative operator which, as is well known, can be evaluated efficiently by means of the fast cosine transform.

Using the Chebyshev (resp. Fourier-spectral) discretizations mentioned above, the one-dimensional boundary value problems given by the ODEs (2.24) and the boundary conditions (2.26) (resp. 2π-periodic boundary conditions) become systems of linear equations. In order to fully take advantage of the fast cosine transform and fast Fourier transform we solve these systems by means of the GMRES iterative solver with second order finite difference preconditioner (cf. [10, p.] and [17]).

For both Chebyshev and Fourier-spectral discretizations, an exponential filter [44], which does not degrade the s-th order accuracy of the method (cf. [2, Sec. 4.3]), is employed to eliminate high-frequency aliasing errors and thus ensure stability. In the Chebyshev case, for example, the filtered coefficients $\hat{c}_n$ for a given function

$f = \sum_n c_n T_n(x)$ are given by

$$\hat{c}_n = \exp\left(-\alpha\left(\frac{n}{N}\right)^{2p}\right) c_n.$$

For all of the results presented in Section 4.1 we have set $\alpha = 16\log 10$ and $p = 8$. The filter is applied at the end of the time step to each line of discretization points in each dimension, requiring one fast cosine transform per line to obtain the coefficients c_n, and one transform to obtain the filtered physical function values.

The transformation of the equations to general coordinates requires the metric terms ξ_x, ξ_y, etc; see Section 2.1.2. The solvers presented in this thesis use the so-called "invariant form" of these metric terms [86], but other (accurate) alternatives could be equally advantageous. The derivatives of the physical coordinates (x_ξ, x_η, etc.) needed in the actual expressions for the metric terms are produced by means of the discrete derivative operators implicit in the Chebyshev or Fourier spatial approximation used in each case.

2.5.2 Overall algorithmic description and treatment of boundary values

Any successful algorithm for the Navier-Stokes equations must address the salient difficulty posed by the lack of a physical boundary condition for the density ρ. This section thus provides an overall description of the proposed BDF-ADI schemes, with emphasis on topics concerning boundary conditions. Some comments are also presented with regards to the impact of boundary corners and edges on stability and accuracy.

Given the elements described in previous parts of this chapter, the BDF-ADI algorithms of s-th order of temporal accuracy can now be described in rather simple terms—except perhaps for some details concerning boundary values of the fluid density, which require a few additional considerations. The absence of a density bound-

ary condition has previously been successfully addressed by means of discretization strategies based on use of staggered grids see, e.g., [18, Ch. 4.6] and the references therein. In some such strategies the velocity and the temperature are collocated on a Gauss-Lobatto grid while the density is collocated on a Gauss grid—so that the density mesh contains no boundary points, and therefore no density boundary conditions are needed. In the context of ADI-based methods such as the ones considered in this chapter, however, it is not clear that a natural staggered-grid ADI method could be designed—since the ADI approach requires solution of one-dimensional boundary value problems which couple all field components. An alternative approach is proposed in this thesis. This method uses the same Gauss-Lobatto grid for all unknowns, including the density, and therefore it requires determination of the boundary values of the density as part of the overall solution.

We describe our implementation for spatially two-dimensional problems; extension to three dimensions is analogous. The main observation is as follows: throughout the domain (including the boundary) the density is entirely determined by the equations (2.24). In particular, the density boundary values at $t = t^{n+1}$ can be obtained by interpreting equation (2.24b) for each j line ($0 \le j \le N$) as a linear system satisfied by $(d+2)(N-1)+2$ unknowns, namely, the discrete values Q_{jk}^{n+1} of the vector $Q = (\mathbf{u}^{\mathrm{T}}, T, \rho)^{\mathrm{T}}$ that correspond to discretization points in the interior of the PDE domain ($1 \le k \le N-1$) *together with* the density boundary values (ρ_{jk}^{n+1} for $k = 0$ and N). Clearly, collocation of (2.24b) at all interior points along a relevant coordinate segment furnishes $(d+2)(N-1)$ equations for these unknowns. The necessary two additional equations are obtained by enforcing the portion of (2.24b) that arises from the mass conservation equation at each one of the two boundary points $k = 0$ and $k = N$. Note that in order to formulate this system of $(d+2)(N-1)+2$ equations, the values of $Q = (\mathbf{u}^{\mathrm{T}}, T, \rho)^{\mathrm{T}}$ at all interior discretization points and boundary points must be available for all time steps t^{ℓ} with $n-s+1 \le \ell \le n$, as well as the values of $\mathbf{u}$ and T at the boundary points at time t^{n+1} (using the boundary conditions (2.26))

and the values of $Q^* = ((\mathbf{u}^*)^{\mathrm{T}}, T^*, \rho^*)^{\mathrm{T}}$ at interior and boundary points, which are provided by the previous half-step in the ADI process. Using such data the algorithm produces the corresponding interior and boundary values of Q_{jk}^{n+1}.

In order to obtain the necessary values of Q^* at all discretization points, we note that equation (2.24a) provides, for each k line ($0 \leq k \leq N$), a similar linear system of $(d+2)(N-1)+2$ unknowns—the values of Q_{jk}^* in the interior of the domain ($1 \leq j \leq N-1$) together with the intermediate density ρ^* at the boundary points ($j = 0,\ N$). These equations are obtained by collocating the equations (2.24a) at the interior points, as well as the mass conservation component of (2.24a) at the boundary points. The right-hand side of this linear system, in turn, requires the values of Q at interior discretization points and boundary points for a few time steps t^ℓ with $\ell \leq n$, as well as the values of $\mathbf{u}$ and T at the boundary points at time t^{n+1} (using (2.26a)), all of which are either data given as part of the problem, or were otherwise produced by the algorithm over a few previous time steps. Thus, the system for the intermediate unknowns Q^* is closed: the solution Q^{n+1} can be obtained for all n by means of the procedure described in this paragraph and the previous one provided the solution is known for the first s time steps.

Although the solvers enjoy quasi-unconditional stability for the implementation described thus far, a reduction in the temporal order of accuracy may occur due to the solutions u^{n+1} and T^{n+1} of the ODEs in the second step of (2.24) for lines along the boundaries not satisfying the physical boundary conditions prescribed there. Re-imposing the boundary conditions for u and T along these boundaries eliminates this problem without affecting stability.

We emphasize here that no special boundary conditions are required for either the intermediate density ρ^* or the final density ρ^{n+1}. The density is determined entirely by the equations (2.24) throughout the domain, up to and including the boundary. Furthermore, the presence of corners does not impact the stability of the solver: no special boundary treatment for the corners of the domain are necessary.

Remark 2.7: *Although corners do not affect the stability of our solvers, we note here that the spatial accuracy may deteriorate (as is well known for spectral discretizations [10, Ch. 6.12]) which leads to a corresponding loss of time accuracy order. This problem is alleviated by using a multi-domain decomposition that smooths out all corners, which is considered in Chapter 3. Nevertheless, the numerical examples in Section 4.1 show the correct order of time accuracy of the solvers in a domain with no corners (an annulus) and in Cartesian domains with a manufactured solution.*

A wide variety of effective strategies are available for the evaluation of the solution for the initial s time steps. The simplest one of them is the one by which the solution is ramped-up from a constant field state (usually zero for all velocities and one for the density and temperature). But in some situations genuine initial value problems must be solved; see, e.g., the example provided in Section 4.1 involving flow in an annulus, where the density has a non-constant initial condition. In such cases use of explicit solvers is sometimes recommended, but such explicit solvers generally require use of significantly smaller time steps than those used by the implicit solver—in view of their inherent properties of conditional stability. Furthermore, a high-order multistep explicit solver would also require previous time levels, and a Runge-Kutta method requires special treatment of boundary conditions for the intermediate stages. In order to avoid such difficulties, we utilize a strategy based on the first order BDF-ADI method followed by Richardson extrapolation of a sufficiently high order so as to match the overall order of time accuracy of the method.

Chapter 3

Multi-domain implicit-explicit Navier-Stokes solver

In the previous chapter, we introduced the implicit component of the proposed implicit-explicit Navier-Stokes solver. This chapter completes the presentation of the full solver with the remaining necessary elements—namely, a spectral-like single domain spatial approximation, an explicit time marching method for explicit zones, a method of domain decomposition, and a Schwarz subiteration strategy for the solution of the governing equations on the decomposed domain.

3.1 Fourier continuation spatial approximation

The nature of the BDF-ADI solver presented in this thesis requires the use of structured grids. Most PDE solvers for structured grids are based on the use of finite differences (FD), in view of their intuitiveness and ease of implementation. However, achieving high-order spatial accuracy in practice is not without challenges. For example, high-order biased stencils near and at the boundaries can lead to instability. Although compact schemes [5, 60] and summation-by-parts (SBP) operators [67, 68, 82, 83] are effective in restoring stability to FD methods, they do so at the cost of reduced accuracy orders near the boundary.

Moreover, it is well known that FD methods suffer from high numerical dispersion.

Reducing the dispersion error requires an increasing number of points per wavelength proportional to the size of the problem for a fixed level of accuracy. Certain techniques have been proposed which offer some improvements in this regard, such as dispersion-relation-preserving schemes [84,85] and modified Padé operators [60]. Although these methods can be tuned to provide perfect dispersion for a few select frequencies, there is a corresponding reduction in the order of accuracy of the FD scheme, and these methods also do not address the broadband dispersion inherent in nonlinear problems.

Spectral methods are an attractive alternative in dealing with both of the challenges mentioned above [10, 19, 47]. They are spectrally accurate throughout the domain, including the boundary; they generally require fewer discretization points for a given accuracy tolerance compared to finite differences; and they reproduce the dispersion characteristics of the PDE remarkably well (perfectly in the case of Fourier methods). Unfortunately, polynomial spectral methods require clustering of points at the boundaries of the domain, resulting in severe time step restrictions for explicit methods. Classical Fourier methods, on the other hand, are only applicable to periodic problems—otherwise they will suffer from the Gibbs phenomenon and first order spatial convergence in the interior of the domain (see, e.g., [10, Ch. 2.2]).

The goal of extending the advantages of Fourier methods (dispersionless-ness and high-order accuracy, in particular) to general non-periodic domains has lead to the development of the FC methods that are the foundation of the spatial approximation used in the multi-domain solver. The Fourier continuation (FC) method produces an interpolating Fourier series representation by relying on a "periodic extension" of a given function, that closely approximates it in the physical domain, but which is periodic on a slightly enlarged domain. In other words, given a function f defined, without loss of generality, on the unit interval as

$$f(x) : [0, 1] \to \mathbb{R},$$

the FC method produces a periodic function f^c defined on an extended interval,

$$f^c(x) : [0, b] \to \mathbb{R}$$

with $b > 1$, which closely approximates f on the original interval $[0, 1]$.

Fully discrete Fourier continuation algorithms generally proceed as follows: letting N be the number of equispaced discretization points over the unit interval ($x_i = ih$, $i = 0, \ldots, N-1$, $h = 1/(N-1)$) together with the function values $f(x_i)$, the FC method produces a b-periodic trigonometric polynomial f^c,

$$f^c(x) = \sum_{k=-M}^{M} a_k e^{\frac{2\pi ik}{b}x}, \tag{3.1}$$

that matches the given discrete values of f, i.e., $f^c(x_i) = f(x_i)$, $i = 0, \ldots, N-1$. Derivatives of the function can then be easily computed term-by-term as with any Fourier series; e.g.,

$$\frac{\partial f}{\partial x} \approx \frac{\partial f^c}{\partial x} = \sum_{k=-M}^{M} a_k \frac{2\pi ik}{b} e^{\frac{2\pi ik}{b}x}.$$

In the simplest treatment [14] (also known as the FC(SVD) algorithm) the coefficients a_k of the series (3.1) are the solution of the least-squares minimization

$$\{a_k\} = \arg\min_{a_k} \sum_{i=0}^{N-1} |f_i - f^c(x_i)|^2,$$

which is found in practice by means of the singular value decomposition (SVD). For time dependent problems, unfortunately, this version of the FC method is far too expensive. To circumvent this difficulty, an accelerated method was proposed in [16] which allows for Fourier continuation of functions on the basis of a small number $d = d_\ell, d_r$ of points at the left and right ends of the interval and a projection onto a Gram polynomial basis whose FC extensions are precomputed via a high-precision SVD. In effect, this procedure produces a "basis" of continuation functions that can be

utilized by a PDE solver. The following section presents a high-level description of the "FC(Gram)" method; see [2, 16] for more detailed discussions of the implementation.

3.1.1 Accelerated Fourier continuation: FC(Gram)

Let $\mathbf{f} = (f_0, \ldots, f_{N-1})^T$ be the column vector containing N values of a given function f in the domain $[0, 1]$ at the equispaced points $x_i = ih,\ 0 \leq i \leq N-1$, and let $\mathbf{f}^c = (f_0^c, \ldots, f_{N+C-1}^c)^T$ be the vector of $N + C$ continuation values ($C > 0$) on the extended domain $[0, b]$ at the points $x_i = ih,\ 0 \leq i \leq N + C - 1$, the first N of which coincide with the points of the original interval. For notational simplicity, we also use the periodic continuation of this discrete function of x_i to the whole line by defining

$$x_{i+m(N+C)} = (i + m(N + C))h \quad \text{and} \quad f^c_{i+m(N+C)} \equiv f_i^c \quad \forall m \in \mathbb{Z}. \tag{3.2}$$

A general continuation such that f^c interpolates f in the original interval ($f_i^c = f_i$ for $0 \leq i \leq N-1$) can be written as

$$\mathbf{f}^c = \begin{pmatrix} I \\ A \end{pmatrix} \mathbf{f} \tag{3.3}$$

where I is the $N \times N$ identity matrix and A is a $C \times N$ matrix. The FC(Gram) algorithm produces the C continuation function values using only a small number of points d_ℓ and d_r at the left and right ends of the interval respectively. That is, the matrix A in equation (3.3) is taken to be of the form

$$A\mathbf{f} = A_\ell \begin{pmatrix} f_0 \\ \vdots \\ f_{d_\ell - 1} \end{pmatrix} + A_r \begin{pmatrix} f_{N-d_r} \\ \vdots \\ f_{N-1} \end{pmatrix},$$

where A_ℓ and A_r are respectively $C \times d_\ell$ and $C \times d_r$ matrices defining a smooth "blend to zero" operation—i.e., A_ℓ takes the function values $f_0, \ldots, f_{d_\ell - 1}$ at $x_0, \ldots, x_{d_\ell - 1}$ and

provides a smooth continuation to the left terminating with the function values 0 at the points $x_{-C-d_r}, \ldots, x_{-C-1}$. Similarly, A_r provides a continuation to the right with values $f_{N-d_r}, \ldots, f_{N-1}$ at $x_{N-d_r}, \ldots, x_{N-1}$ and 0 at the points $x_{N+C}, \ldots, x_{N+C+d_\ell-1}$.

More precisely, the blend-to-zero operations proceed as follows: Without loss of generality let $d_\ell = d_r = d$ (all FC computations in this thesis satisfy this assumption) and define $\delta = 1 - x_{N-d}$ to be the width of the left and right fringe regions. The rightward extension is obtained by applying the FC(SVD) algorithm described in the previous section to a certain polynomial $p(x)$ defined on $[1-\delta, b+\delta]$ and extended periodically to the interval $[1-\delta, b+\delta+(b-1)]$. The polynomial $p(x)$ is the interpolant of the data $f_{N-d}, \ldots, f_{N-1}$ at the points $x_{N-d}, \ldots, x_{N-1}$ and zero at the points $x_{N+C}, \ldots, x_{N+C+d}$. The resulting Fourier series is then sampled at the continuation points $x_N, \ldots, x_{N+C-1}$ to obtain the function values $p^c_N, \ldots, p^c_{N+C-1}$. Similarly, the leftward extension is obtained by applying the same procedure to the polynomial $q(x)$ interpolating zero at the points $x_{N-d}, \ldots, x_{N-1}$ and the data $\{f_0, \ldots, f_{d-1}\}$ at the points $x_{N+C}, \ldots, x_{N+C+d}$. Once the left and right continuations are computed, the values of the function f^c are simply the sum of the left and right extension values: $f^c_i = p^c_i + q^c_i,\ N \le i \le N+C-1$. The procedure is illustrated in Figure 3.1.

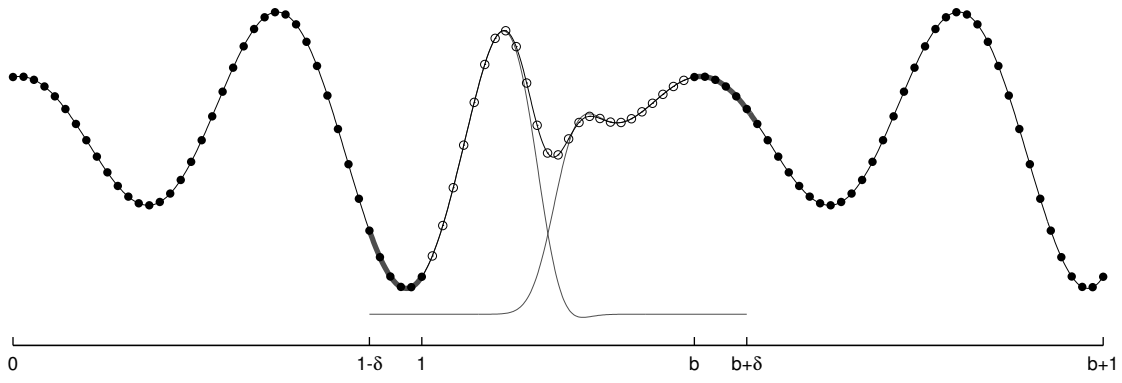


Figure 3.1: Illustration of the FC(Gram) method, showing the original function values on the b-periodic domain (solid circles) together with the continuation values (open circles) which are obtained by summing the left and right blend-to-zero extensions (thin gray lines). The thick black curves indicate the polynomial approximations in the fringe regions which are used to produce the blend-to-zero extensions.

Computing the SVDs necessary to complete the continuation procedure described above can be done once for all time on appropriate bases of Gram polynomials in a precomputation stage. The continuation operation (3.3) is then divided into two steps, which amounts to decomposing the matrices A_ℓ and A_r into the products

$$A_\ell = B_\ell Q_\ell, \quad A_r = B_r Q_r.$$

The $d \times d$ matrices Q_ℓ and Q_r are projections onto orthogonal bases of Gram polynomials which are zero at the right and left fringe points, respectively; and the $C \times d$ matrices B_ℓ and B_r are, respectively, the precomputed left and right blend-to-zero extensions on those bases, as described in the previous paragraph.

Remark 3.1: *For all numerical examples presented in this thesis using the FC spatial approximation, the number of left and right fringe points is* $d = 5$ *and the number of continuation points is* $C = 25$. *For simplicity, the biased order extensions introduced in [2] are not used.*

3.1.2 One-dimensional advection example

In this section, we demonstrate the advantages of the FC methodology with a simple example in one spatial dimension. We consider the advection equation

$$\begin{aligned} u_t + u_x = 0, &\quad (x,t) \in [x_l, x_r] \times [0, t_f], \\ u(x,0) = u_0(x), &\quad x \in [x_l, x_r], \\ u(x_l,t) = g(t), &\quad t \in [0, t_f]. \end{aligned} \tag{3.4}$$

For this example, the extents of the domain are $x_l = 0$, $x_r = 20$, the final time is $t_f = 15$, and the initial and boundary functions u_0 and g are given by the prescribed

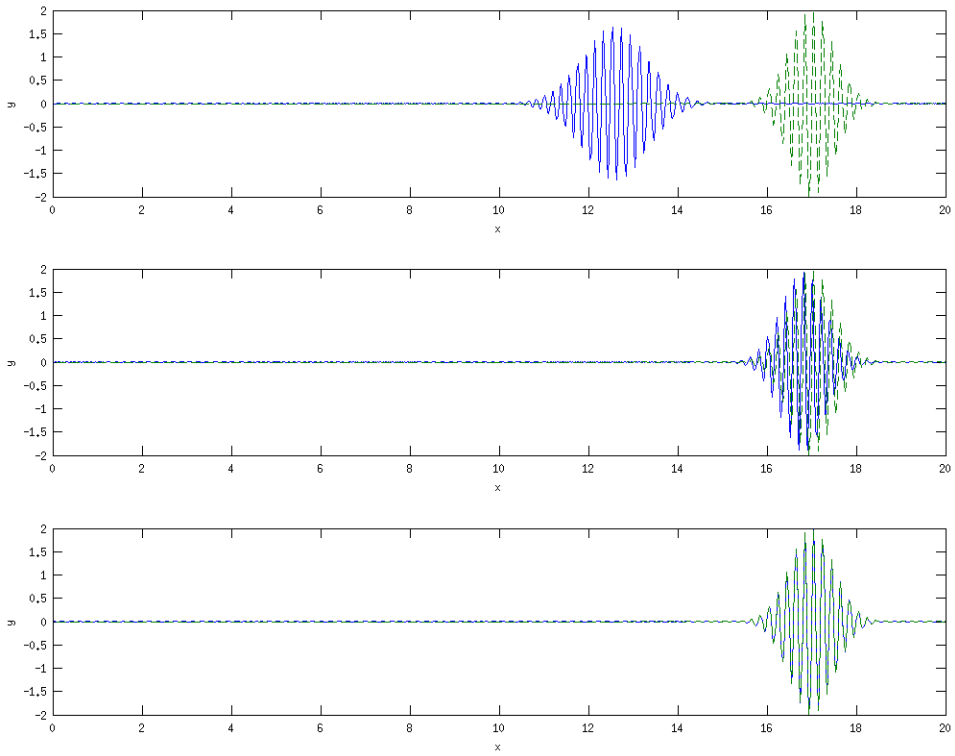


Figure 3.2: Numerical solution of the advection equation (3.4) at time $t = 15$ using second order finite differences (top), fourth order compact schemes (middle), and Fourier continuation (bottom). The numerical dispersion in the finite difference solutions is clearly visible at this solution time.

exact solution

$$u(x,t) = A \exp\left(-\frac{(x-2-t)^2}{2\sigma^2}\right) \sin(2\pi k(x-2-t)), \tag{3.5}$$

where $A = 2$, $k = 5$, and $\sigma^2 = 0.5$.

We compare the FC methodology with second order finite differences (FD2) and the fourth order compact scheme (CP4) by discretizing the domain with a total of $N = 800$ equispaced points. For the FC solver, the domain is divided into four overlapping sub-domains, with each sub-domain sharing six points with neighboring

sub-domains. The solution is marched forward to the final time using the Adams-Bashforth method of order four with $\Delta t = 0.1(x_r - x_l)/(N-1)$ (see Sections 3.3 and 3.4 for description of the domain decomposition and time marching strategies). The FC solver uses fourth order FC(Gram) continuation.

Figure 3.2 shows the solutions produced by each of the three solvers at the final time $t_f = 15$. Both finite difference solutions suffer from enough dispersion that the error is essentially equal to the amplitude A of the exact solution. The FC solver, on the other hand, has an error of 1.49×10^{-2}, using approximately 2.5 points per wavelength. We also report the overall computation times (in seconds) for each of the solvers (which were run on a single core): 0.464 (FC), 0.454 (CP4), 0.346 (FD2). This example provides a powerful testimony to the strengths of the FC methodology for wave-propagation problems, highlighting the near dispersionless-ness and efficiency of the method.

3.1.3 Variable coefficient FC-ODE system solver

The implementation of the implicit component of the implicit-explicit solver presented in this thesis requires solutions of one dimensional boundary-value ODE systems discretized by the FC spatial approximation. To that end, we develop in this section the FC-ODE system solver for general boundary-value problems that we use in the Navier-Stokes solver.

Consider the ODE

$$A\,q + B\,q_x + C\,q_{xx} = f \tag{3.6}$$

on the interval $x \in [0, 1]$, with general Robin boundary conditions

$$a\,q + b\,q_x = g, \quad x = 0 \tag{3.7a}$$

$$c\,q + d\,q_x = h, \quad x = 1 \tag{3.7b}$$

where q and f are m-dimensional vector-valued functions of x; A, B, and C are $m \times m$

matrix-valued functions of x; a, b, c, and d are $m \times m$ matrices; and g and h are m-vectors. The matrices C, a, b, c, and d could have rows equal to zero—however, we assume that each row of C is zero at $x = 0$ (respectively $x = 1$) if the corresponding rows of a and b (c and d) are both zero (i.e., a boundary condition is supplied for each row in equation 3.6 with second derivative terms).

Discretization of the interval into N equispaced points yields the vector- or matrix-valued grid functions $q_i = q(x_i)$, $A_i = A(x_i)$, etc. Let δ_x and δ_{xx} be the discrete FC first and second derivative operators respectively. The system (3.6) is block-decomposed, with the i-th interior block ($i = 2, \ldots, N-1$) given by

$$A_i q_i + B_i \sum_{j=1}^{N} (\delta_x)_{ij} q_j + C_i \sum_{j=1}^{N} (\delta_{xx})_{ij} q_j = f_i. \tag{3.8}$$

The non-zero rows of the boundary conditions (3.7) are used to replace the rows of the discretized ODE for $i = 1, N$. In detail, let a superscript k denote the k-th row of a vector or matrix. We introduce the modified matrices and vectors $\hat{A}_i$, $\hat{B}_i$ and f_i ($i = 1, N$) given by

$$\hat{A}_1^k = \begin{cases} A_1^k & \text{if } a^k = b^k = 0 \\ a^k & \text{otherwise} \end{cases} \qquad \hat{A}_N^k = \begin{cases} A_N^k & \text{if } c^k = d^k = 0 \\ c^k & \text{otherwise} \end{cases}$$

$$\hat{B}_1^k = \begin{cases} B_1^k & \text{if } a^k = b^k = 0 \\ b^k & \text{otherwise} \end{cases} \qquad \hat{B}_N^k = \begin{cases} B_N^k & \text{if } c^k = d^k = 0 \\ d^k & \text{otherwise} \end{cases}$$

$$\hat{f}_1^k = \begin{cases} f_1^k & \text{if } a^k = b^k = 0 \\ g^k & \text{otherwise} \end{cases} \qquad \hat{f}_N^k = \begin{cases} f_N^k & \text{if } c^k = d^k = 0 \\ h^k & \text{otherwise} \end{cases}.$$

The first and N-th blocks of the discretized ODE system are then defined to be

$$\hat{A}_i q_i + \hat{B}_i \sum_{j=1}^{N} (\delta_x)_{ij} q_j = \hat{f}_i, \quad i = 1, N. \tag{3.9}$$

In general the linear system defined by equations (3.8) and (3.9) is full and non-symmetric, so we use the GMRES iterative solver to invert the system [73]. A second order finite difference solver is used to left precondition the system, which greatly reduces the number of iterations needed for convergence [17]; cf. [10, Ch. 15.3]. The finite difference system is given by equations (3.8) and (3.9) with the discrete FC derivative operators δ_x and δ_{xx} replaced by corresponding centered difference schemes in the interior and a two-point one sided scheme at the boundary. The result is a block-tridiagonal system which we invert efficiently using a block-LU decomposition. The factorization need only be performed once per call of the GMRES solver.

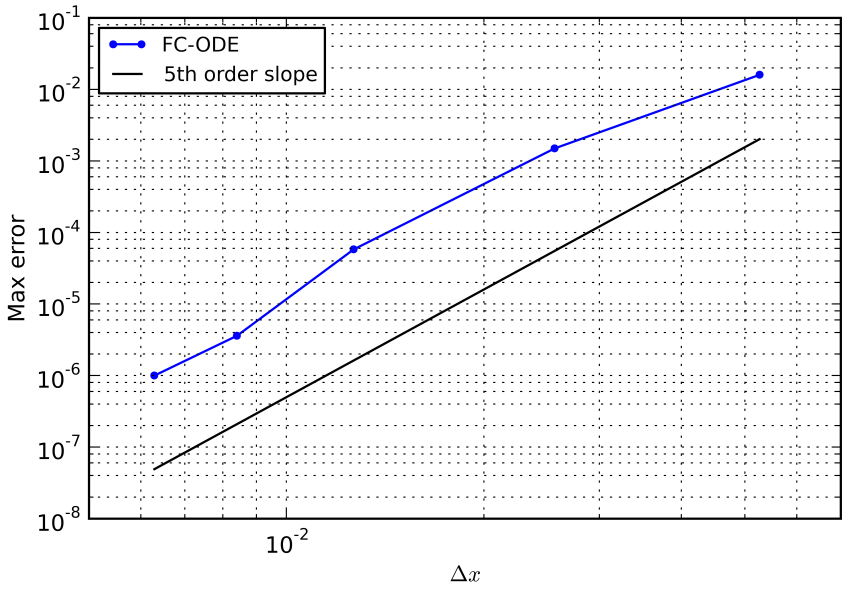


Figure 3.3: One dimensional spatial convergence test of the variable coefficient FC-ODE solver for the system (3.10) with exact solution (3.11).

Figure 3.3 demonstrates the convergence of the solver for the model ODE system

$$u + \Delta t \left(\widetilde{u}\, u_x + T_x + \frac{\widetilde{T}}{\widetilde{\rho}} \rho_x - \frac{\nu}{\widetilde{\rho}} u_{xx} \right) = f^1 \tag{3.10a}$$

$$T + \Delta t \left(\widetilde{u}\, T_x + \widetilde{T}\, u_x - \frac{\nu}{\widetilde{\rho}} T_{xx} \right) = f^2 \tag{3.10b}$$

$$\rho + \Delta t \left(\widetilde{u}\, \rho_x + \widetilde{\rho}\, u_x \right) = f^3 \tag{3.10c}$$

in the domain $x \in [0, 1]$, where $\Delta t = 0.01$ and $\nu = 0.01$ are constant, the functions $\widetilde{u}$, $\widetilde{T}$, and $\widetilde{\rho}$ are given by

$$\widetilde{u} = 1 - 6e^{\sin(5x)}, \quad \widetilde{T} = \log(3 + \sin(6x)), \quad \widetilde{\rho} = 1 + \frac{1}{2}\cos(4x^2 + 2), \tag{3.11}$$

and the right-hand side functions are chosen so that the exact solution is $u = \widetilde{u}$, $T = \widetilde{T}$, $\rho = \widetilde{\rho}$. Letting $q = (u, T, \rho)^T$, the boundary conditions are given by (3.7) with pseudorandom matrices

$$a = \begin{pmatrix} 0.71 & 0.39 & 0.19 \\ 0.51 & 0.71 & 0.06 \\ 0 & 0 & 0 \end{pmatrix} \quad b = \begin{pmatrix} 0.27 & 0.60 & 0.29 \\ 0.73 & 0.58 & 0.04 \\ 0 & 0 & 0 \end{pmatrix}$$

$$c = \begin{pmatrix} 0.87 & 0.57 & 0.20 \\ 0.19 & 0.41 & 0.51 \\ 0 & 0 & 0 \end{pmatrix} \quad d = \begin{pmatrix} 0.07 & 0.72 & 0.99 \\ 0.84 & 0.85 & 0.25 \\ 0 & 0 & 0 \end{pmatrix}$$

with the right-hand sides chosen to enforce the exact solution. Notice that no boundary conditions are prescribed for ρ.

Remark 3.2: *The paper [17] presents an alternative variable coefficient scalar FC-ODE solver, which differs from the one presented here mainly in two respects: 1) It solves for the particular and homogeneous solutions separately, using the homogeneous solutions to correct the particular solution. This approach could be useful when solving*

many ODEs with the same coefficients and right-hand side but different boundary conditions. However, due to the nonlinearity of the Navier-Stokes equations, the ODEs we must solve have coefficients that change at each time step, and the approach in [17] provides no significant advantage. 2) It uses high-order asymptotic matching for the narrow boundary layers that occur in highly singularly perturbed ODEs, which were necessary to ensure stability of the PDE solver for very small Δt. These corrections were found to be unnecessary for the stability of the Navier-Stokes solver for all time discretizations we considered.

3.1.4 Filtering

Most high-order PDE solvers use some form of filtering or artificial viscosity to ensure stability of the numerical solution, particularly for nonlinear problems. Previous FC-based solvers have employed a Fourier-space exponential filter to great advantage [2, 3, 35]. In detail, if the Fourier coefficients of the continued function are $\hat{f}_k^c$, $-N_c/2 \leq k \leq N_c/2$, then the filtered coefficients are given by

$$(\hat{f}_k^c)_{\text{filter}} = e^{-\alpha \left|\frac{2k}{N_c}\right|^{2p}} \hat{f}_k^c,$$

where p is the order of the filter and α is a parameter such that the highest mode is filtered to the level $e^{-\alpha}$.

Unfortunately, it is possible for the aforementioned filter to introduce relatively large errors in the numerical solution. Qualitatively, the reason is that the continuation portion of a function can have large peaks (sometimes several orders of magnitude greater than the L^∞ norm of the function in the physical domain) as well as the largest spatial gradients. The exponential filter (or any Fourier-space filter for that matter) will "smear" these peaks and gradients into the physical domain, leading to especially large errors near the boundary.

The development of a suitable alternative filter is currently a topic of research.

In the meantime, we use a Padé-type finite difference filter [39], which has been shown to be effective in solving the Navier-Stokes equations in a finite-difference context [38,90,91]. Given function values f_i, the filtered function values $\widetilde{f}_i$ sufficiently far from the boundary satisfy

$$\alpha\widetilde{f}_{i-1} + \widetilde{f}_i + \alpha\widetilde{f}_{i+1} = \sum_{j=0}^{n} \frac{a_n}{2}(f_{i+j} + f_{i-j}),$$

where $-0.5 < \alpha \leq 0.5$ is a parameter determining the strength of the filter (with no filtering at $\alpha = 0.5$) and the $n+1$ coefficients a_i can be chosen so that the filtering order is $2n$. Reference [39] provides formulas for the coefficients, as well as similar one-sided formulas for points near the boundaries.

In all our numerical results employing the FC methodology, we use a sixth order filter ($n = 3$) with $\alpha = 0.3$. Application of the filter is limited only to the physical function values (ignoring the continuation region). The filter is applied to each line of the computational domain in each dimension at the end of the time step.

3.2 Explicit time marching

Following [2], the Adams-Bashforth (AB) method [58, Ch. 3.9] is employed in all domains requiring explicit time marching. Although the fourth order Runge-Kutta (RK4) scheme is a popular choice for Navier-Stokes solvers, it is more expensive than AB methods (requiring four evaluations of the right-hand side for every time step) and proper treatment of boundary conditions at intermediate stages can be problematic [1, 20].

In this thesis, AB methods of orders 2 to 4 are used, depending on the desired order of time accuracy. Given the form of the PDE

$$Q_t = \mathcal{P}(Q, t),$$

the AB scheme of order s for the approximate solution Q^{n+1} at time $t = t^{n+1}$ is

$$Q^{n+1} = Q^n + \Delta t \sum_{j=0}^{s-1} b_j \mathcal{P}(Q^{n-j}, t^{n-j}), \tag{3.12}$$

where the b_j are the AB coefficients summarized in Table 3.1.

s	b_0	b_1	b_2	b_3
1	1			
2	$\frac{3}{2}$	$-\frac{1}{2}$		
3	$\frac{23}{12}$	$-\frac{4}{3}$	$\frac{5}{12}$	
4	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{3}{8}$

Table 3.1: Coefficients for AB methods of orders $s = 1, \ldots, 4$.

Explicit domains in all the numerical examples in this thesis use Dirichlet type boundary conditions. Enforcement is accomplished by injecting the boundary values at all boundary points at the end of each time step, as described in [2].

3.3 Domain decomposition

The decomposition of a general domain Ω proceeds by first dividing the space into a set of larger overlapping logical rectangles Ω_j such that $\Omega = \cup_j \Omega_j$. These larger patches are then divided into a set of sub-patches which share a layer of points with neighboring sub-patches. In effect, each overset grid has one of two types of overlapping boundaries: 1) the overlap region is imperfect, in that one or more points (generally all) do not correspond to any grid point in any neighboring patch, and 2) all points in the boundary region are also grid points in a neighboring patch, resulting in a perfect overlap. Boundaries of the first type are referred to as *interpolation boundaries*. They consist of sets of *interpolation points*, and they enable information transfer from one grid to another by means of suitably high-order interpolation

methods. The other kind of boundary, which we call an *exchange boundary*, does not require interpolation.

Let $p = (x^*, y^*, z^*)$ be an interpolation point in some domain Ω_i. By definition, p also lies in one or more other patches. Suppose Ω_j is one such patch and let $q = (\xi^*, \eta^*, \zeta^*)$ be the point in the computational domain of Ω_j corresponding to p—that is, $x(\xi^*, \eta^*, \zeta^*) = x^*$ and similarly for the other coordinates. For Ω_j to be an acceptable donor patch, an $m \times m \times m$ stencil of points (m is the stencil width, resulting in an interpolation order of m) must be found such that the stencil includes q in its interior, in such a way that no points in the interpolation stencil used are themselves receivers of interpolation data from other patches. (Exchange points are not excluded from being part of a donor stencil to an interpolation point). Of the stencils satisfying these requirements, the one with q closest to its center is chosen. A function is then interpolated in the computational domain using the tensor product Lagrange interpolation formula.

Exchange points can be thought of as interpolation points where the point q is itself one of the points in the donor stencil. Indeed, in such a case, the computed interpolation weights will be zero for all but the coinciding point in the stencil, which will be one. Thus, without loss of generality, in the remainder of this thesis "interpolation" boundaries (points) will refer to both interpolation and exchange boundaries (points). Differences in implementation will be noted where necessary.

In all our examples, a stencil width of $m = 7$ is employed, and each interpolation boundary consists of a layer of two points deep, so that the overlap at an exchange boundary is four points wide. Figure 3.4 illustrates the exchange data process in a one-dimensional case.

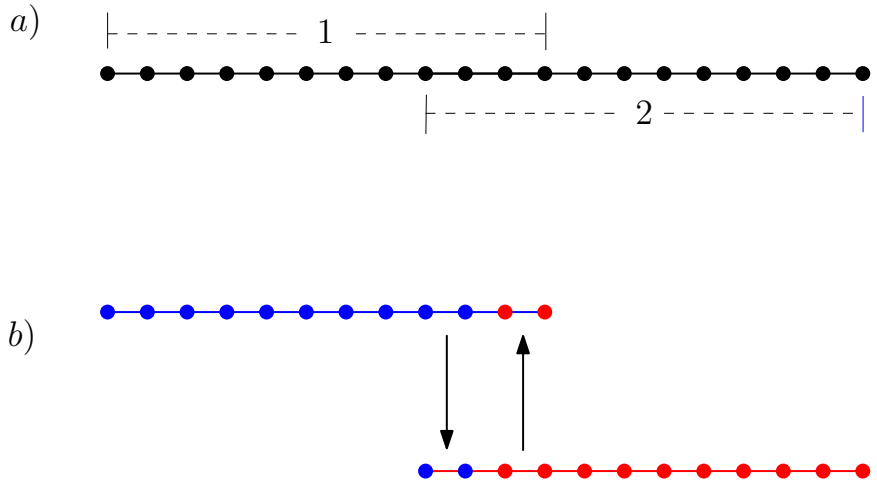


Figure 3.4: One dimensional illustration of exchange boundaries. *a*) Domains 1 and 2 overlap perfectly in a region four points wide. *b*) At the data-passing step of the algorithm, the solution values of the last two points in each domain are substituted by the corresponding values in the neighboring subpatch.

3.4 Multi-domain implicit-explicit subiteration strategy

The time marching method of the multi-domain implicit-explicit solver uses a fixed time step Δt for all meshes, chosen according to the strictest stability and accuracy requirements found in any of the component meshes. After the explicit patches have evolved the solution from time t^n to time t^{n+1} the parallel Schwarz method is used to advance all implicit subpatches forward one time step, using physical boundary conditions, where available, and boundary conditions given by neighboring explicit solvers otherwise. (In fact, a slight modification of this strategy is actually implemented in our solvers to improve parallel efficiency; see Remark 3.3.) This section describes briefly the overall implicit-explicit strategy.

An implicit patch requires boundary values at the beginning of the time step, which are not available yet at interpolation boundaries shared with neighboring implicit zones. Therefore, the solver uses subiterations to successively correct the solution. Explicit solvers, on the other hand, impose the boundary conditions at the end

of the time step, so no subiteration is required. The multi-domain implicit-explicit subiteration algorithm ideally proceeds as follows:

1. All explicit zones are marched forward in time one time step.

2. Interpolation data from explicit zones is sent to neighboring zones.

3. All implicit zones are marched forward in time to obtain the subiteration solution.

4. After completion of each subiteration, interpolation data for boundary points is exchanged between all implicit zones. (After the final subiteration, interpolation data is sent to all explicit zones as well.)

5. The interpolated data received is used to set the boundary conditions at relevant boundary points in all implicit zones, and steps 3 through 5 are repeated for the specified number of subiterations.

Remark 3.3: *Steps 1 and 3 could be performed in parallel provided approximate (e.g., extrapolated) boundary conditions are used. Of course this cannot be carried for all subiterations without leading to instability. But under certain circumstances this strategy is recommended. For example, a processor assigned to an implicit zone can proceed with the first subiteration in parallel with the explicit solves. In this case, an initial approximation for the boundary conditions is obtained by means of the temporal extrapolation formula* (2.13)*. In practice this approach has provided an effective parallelization methodology.*

3.4.1 Convergence rate of the subiterations

This section presents analysis that explains the convergence properties of the implicit-explicit iterations and, indeed, the particularly fast convergence that arises from actual implementations of this approach in practice. This analysis shows that the

error in subiteration solutions decreases exponentially fast with the size of the overlap between sub-domains. Practical application of these results can be used to ensure that a small number of subiterations suffice to meet a given accuracy requirement.

To effectively pursue this agenda we again restrict attention to the advection-diffusion equation

$$u_t + \alpha u_x = \beta u_{xx}$$

on the real line $x \in \mathbb{R}$ with u vanishing as $x \to \pm\infty$. Overlapping patches are simulated in this context by dividing the domain into two overlapping sub-domains, $\Omega_1 = (-\infty, r)$ and $\Omega_2 = (\ell, \infty)$ with $\ell < r$. Using for this example a semi-discrete formalism (discrete in time and continuous in space), discretizing in time according to the order-s BDF scheme and letting u^n denote the approximate solution at time $t = t^n$, we have the ODE boundary-value problem for u^{n+1},

$$u^{n+1} + b\,\Delta t L u^{n+1} = f^n \equiv \sum_{j=0}^{s-1} a_j u^{n-j}, \tag{3.13}$$

where $L = \alpha\frac{\partial}{\partial x} - \beta\frac{\partial^2}{\partial x^2}$. The above equation is solved independently in each one of the two subdomains according to the parallel Schwarz method. To that end, let v_1^k and v_2^k be the approximate solutions of equation (3.13) in Ω_1 and Ω_2, respectively. The subiteration procedure involves repeatedly solving

$$\begin{cases} v_1^{k+1} + b\,\Delta t L v_1^{k+1} = f^n \ \text{ in } \Omega_1 \\ v_1^{k+1} \to 0 \text{ as } x \to -\infty \\ v_1^{k+1}(r) = v_2^k(r) \\ v_1^0(r) = v_r \end{cases} \qquad \begin{cases} v_2^{k+1} + b\,\Delta t L v_2^{k+1} = f^n \ \text{ in } \Omega_2 \\ v_2^{k+1} \to 0 \text{ as } x \to \infty \\ v_2^{k+1}(\ell) = v_1^k(\ell) \\ v_2^0(\ell) = v_\ell, \end{cases} \tag{3.14}$$

where v_ℓ and v_r are initial guesses for the value of u^{n+1} at the points $x = \ell$ and r, respectively. Letting $e_1^k = u^{n+1} - v_1^k$ and $e_2^k = u^{n+1} - v_2^k$ denote the error of the k-th

subiteration in each subdomain, it is clear that the errors satisfy the homogeneous versions of the same ODEs:

$$\begin{cases} e_1^{k+1} + b\,\Delta t L e_1^{k+1} = 0 \ \text{ in } \Omega_1 \\ e_1^{k+1} \to 0 \text{ as } x \to -\infty \\ e_1^{k+1}(r) = e_2^k(r) \\ e_1^0(r) = e_r \end{cases} \qquad \begin{cases} e_2^{k+1} + b\,\Delta t L e_2^{k+1} = 0 \ \text{ in } \Omega_2 \\ e_2^{k+1} \to 0 \text{ as } x \to \infty \\ e_2^{k+1}(\ell) = e_1^k(\ell) \\ e_2^0(\ell) = e_\ell. \end{cases} \tag{3.15}$$

where $e_\ell = u^{n+1}(\ell) - v_\ell$ and $e_r = u^{n+1}(r) - v_r$.

The two homogeneous solutions of the ODE are the exponentials

$$\exp(\lambda_1 x), \quad \exp(-\lambda_2 x),$$

where

$$\lambda_1 = \sqrt{\frac{\alpha^2}{4\beta^2} + \frac{1}{\beta b\,\Delta t}} + \frac{\alpha}{2\beta}, \quad \lambda_2 = \sqrt{\frac{\alpha^2}{4\beta^2} + \frac{1}{\beta b\,\Delta t}} - \frac{\alpha}{2\beta}.$$

The positive exponential satisfies the boundary condition at $-\infty$ in the left domain Ω_1 and the negative one satisfies the boundary condition at $+\infty$ in Ω_2. It follows that the $(k+1)$-th errors are given by

$$e_1^{k+1} = \exp(-\lambda_1(r-x))\, e_2^k(r), \quad e_2^{k+1} = \exp(-\lambda_2(x-\ell))\, e_1^k(\ell). \tag{3.16}$$

Evaluating each solution at the boundary point of the neighboring subdomain, we obtain

$$e_1^{k+1}(\ell) = \exp(-\lambda_1\delta)\, e_2^k(r), \quad e_2^{k+1}(r) = \exp(-\lambda_2\delta)\, e_1^k(\ell), \tag{3.17}$$

where $\delta = r - \ell$. Using this recursive relation in equation (3.16), we obtain expressions

for the error of the $(k+1)$-th subiteration at the boundary of left domain

$$\begin{aligned} e_1^{k+1}(r) &= \exp(-\lambda_2\delta)\, e_1^{k-1}(\ell) \\ &= \exp(-(\lambda_1+\lambda_2)\delta)\, e_2^{k-2}(r). \end{aligned}$$

Letting

$$\overline{\lambda} = \frac{1}{2}(\lambda_1+\lambda_2) = \sqrt{\frac{\alpha^2}{4\beta^2} + \frac{1}{\beta b\,\Delta t}}, \tag{3.18}$$

it follows from an inductive argument that the error after k subiterations is given by

$$e_1^k(r) = \begin{cases} \exp(-k\overline{\lambda}\delta)\, e_r, & k \text{ even} \\ \exp(-((k-1)\overline{\lambda}+\lambda_2)\delta)\, e_\ell, & k \text{ odd} \end{cases} \tag{3.19}$$

The expression for the error at the boundary of the right domain, $e_2^k(\ell)$, can be derived similarly.

Using the even k expression in (3.19) (for simplicity), we can determine the approximate number of subiterations K that are necessary to reduce the error to the level of $\mathcal{O}((\Delta t)^{s+1})$ in Ω_1:

$$\begin{aligned} &\exp(-K\overline{\lambda}\delta)\,|e_r| \sim (\Delta t)^{s+1} \\ \Longrightarrow\ & K \approx \frac{-1}{\overline{\lambda}\delta}\ln\left(\frac{(\Delta t)^{s+1}}{|e_r|}\right). \end{aligned}$$

The corresponding estimate in Ω_2 uses e_ℓ instead of e_r. With temporal extrapolation of order s, the initial boundary conditions can be chosen so that the initial errors will be $e_\ell = \mathcal{O}((\Delta t)^s)$ and $e_r = \mathcal{O}((\Delta t)^s)$. Therefore the approximate number of subiterations necessary for an error tolerance of order $(\Delta t)^{s+1}$ is

$$K \approx \frac{-\ln(\Delta t)}{\overline{\lambda}\delta}. \tag{3.20}$$

Consideration of the two terms under the radical in equation (3.18),

$$\frac{\alpha^2}{4\beta^2} \text{ and } \frac{1}{\beta b\,\Delta t},$$

under certain physical regimes allows us to provide even simpler specialized expressions. Indeed, in the "convection dominated" regime the second term is small compared to the first and we thus have

$$\frac{\beta}{\alpha^2\,\Delta t} \ll 1 \quad \Longrightarrow \quad \overline{\lambda} \sim \frac{\alpha}{2\beta}.$$

This is applicable to the case of high Reynolds number flow with a relatively large time step. The "diffusion dominated" case is given by the other extreme:

$$\frac{\beta}{\alpha^2\,\Delta t} \gg 1 \quad \Longrightarrow \quad \overline{\lambda} \sim \frac{1}{\sqrt{\beta b\,\Delta t}}.$$

The revised estimates for the necessary number of subiterations in each of these cases is

$$K \approx \begin{cases} \dfrac{2\beta}{\alpha\,\delta}|\ln\,\Delta t| & \text{convection dominated} \\ \dfrac{\sqrt{\beta b\,\Delta t}}{\delta}|\ln\,\Delta t| & \text{diffusion dominated.} \end{cases} \tag{3.21}$$

The results of the analysis for the linearized one-dimensional Navier-Stokes equations presented in Section 2.4.2 together with the estimate above are used to provide specific suggestions for the number of iterations needed under subsonic flow of a compressible gas. In this context, the viscosity β scales like Re^{-1} and the advection velocity α scales like either the fluid velocity $|\mathbf{u}|$ for convective-type waves (i.e., vorticity and entropy waves) or $|\mathbf{u}| + \mathrm{Ma}^{-1}$ for acoustic waves. Thus, the number of iterations necessary to eliminate errors associated with convection, acoustics and

diffusion satisfy

$$K \approx \begin{cases} \dfrac{2|\ln \Delta t|}{|\mathbf{u}| \operatorname{Re} \delta} & \text{(vorticity and entropy waves)} \\ \dfrac{2|\ln \Delta t|}{(|\mathbf{u}| + \operatorname{Ma}^{-1}) \operatorname{Re} \delta} & \text{(acoustic waves)} \\ \dfrac{1}{\delta}\sqrt{\dfrac{b\,\Delta t}{\operatorname{Re}}}|\ln \Delta t| & \text{(diffusion).} \end{cases} \tag{3.22}$$

For the overset method employed in this thesis, the amount of overlap will generally be $\delta \approx mh$ for some number $m = \mathcal{O}(1)$ and spatial mesh size h. If h is chosen to be on the order of Re^{-1}, for example, equation (3.22) indicates that the number of subiterations necessary to resolve wave phenomena will be $\mathcal{O}(|\ln \Delta t|)$. Using the same discretization, diffusive phenomena will require $\mathcal{O}(\sqrt{\Delta t \operatorname{Re}}|\ln \Delta t|)$ subiterations to reach the level of the truncation error.

3.5 Parallelization

The overset mesh framework together with the parallel Schwarz subiteration strategy naturally lead to a parallel implementation in a distributed computing environment. An efficient implementation should divide the workload among all processors as equally as possible. Details in this regard for explicit FC solvers is well documented in [2,3]. As the solver presented in this thesis is the first implicit and implicit-explicit multi-domain FC solver, additional details need to be presented. To that end, this section provides heuristics for decomposing a domain to maximize efficiency, as well as an investigation of the computational cost of the implicit multi-domain solver.

3.5.1 Implicit multi-domain load balancing

The present load-balancing algorithm is based on the ones presented in [2,3]. Given a set of implicit patches $\{\Omega_j^I\}$, $j = 1, \ldots, M_I$ and a number of target zones p_{total}, the

target number of sub-zones assigned to an implicit patch Ω_j^I is

$$p_j^I = \frac{N_\xi^j N_\eta^j N_\zeta^j}{\sum_{l=1}^{M_I} N_\xi^l N_\eta^l N_\zeta^l} p_{\text{total}}$$

(rounded to the nearest integer), where N_ξ^j, N_η^j, N_ζ^j are the numbers of discretization points in each dimension. The partitioning of a patch into subpatches is performed in such a way that, for each subpatch, the number of discretization points in each direction is approximately equal. If p is the number of target zones assigned to a patch, then the number of partitions in the ξ direction is given by

$$s_\xi = N_\xi \left(\frac{p}{N_\xi N_\eta N_\zeta} \right)^{\frac{1}{3}},$$

rounded to the nearest integer greater than or equal to one. Similar formulas hold for the number of partitions in the η and ζ directions.

Note that the method of sub-zone distribution and partitioning described above could result in a number of subpatches different from the target number of zones. In practice the difference is negligible and the method produces satisfactory results for our purposes.

3.5.2 Implicit multi-domain performance

In this section, we present the results of various computational experiments of the parallel implicit multi-domain solver in a distributed computing environment. All results were obtained on a Poweredge cluster with Inifiniband networking, consisting of 32 compute nodes, each of which has two eight-core Intel Xeon E5-2665 processors (for up to 32 threads per node with Intel Hyper-threading) and 64 GB of memory.

In [2,3], the authors quantified the cost of the explicit FC solvers using the number S of seconds required per processor to advance one million unknowns forward one time

step, which, for our Navier-Stokes solver, is given by the formula

$$S = \frac{(\#\text{ of processors}) \times (\text{total compute time per step}) \times 10^6}{(d+2) \times (\#\text{ of discretization points})}, \tag{3.23}$$

where the factor of $d+2$ in the denominator is the size of the vector of unknowns Q in $d = 2$ and 3 spatial dimensions. The present explicit FC-based Navier-Stokes solver enjoys approximately the same efficiency as the solvers in those contributions—namely,

$$S \approx 1.6 \text{ sec.} \quad \text{(for explicit solvers)}. \tag{3.24}$$

To test the efficiency of the parallel implicit algorithm, we use a smaller version of the mesh described in Section 4.2.2 of the following chapter, which is used for three dimensional tests of flow past a sphere. In this example, the limits of the domain are $[-3, 3]$ in each dimension, with the extent of the spherical-like curvilinear patches the same as in Section 4.2.2. The mesh spacing in the Cartesian patches is a constant value h, while the radial spacing in the curvilinear patches is $h/10$ near the surface of the sphere and h at the outer edge of the patch. The initial conditions and source terms are the same as in the tests of flow past a sphere.

$h_{\max}$	0.06	0.048	0.04		
# grid points	2,045,990	3,817,125	6,372,454		
# sub-domains	104	194	104	194	294
S (1 sub-iter.)	36.8	33.4	33.9	30.7	32.6
S (2 sub-iter.)	54.3	58.9	53.4	51.6	55.3
S (3 sub-iter.)	86.6	78.8	74.4	74.9	77.0

Table 3.2: Number of seconds S needed per processor for the parallel implicit algorithm to advance one million unknowns forward one time step, with various numbers of discretization points, sub-domains, and subiterations.

Table 3.2 reports the value of S for the BDF2-ADI solver on the composite mesh for various values of h, sub-domain partitionings, and number of sub-iterations. In these tests, the Mach number was 0.8, the Reynolds number was 10^3, and the GMRES

residual tolerance was set to 10^{-6}. The solver was run for a total of 20 time steps with $\Delta t = 10^{-3}$. The results show that S is approximately constant for each fixed number of sub-iterations, which corresponds to approximately linear parallel scaling of the implicit algorithm.

We can also use the data in Table 3.2 to compare the computational costs of the explicit and implicit algorithms. For one sub-iteration, the implicit solver is about twenty times more expensive than the explicit solver—which is reasonable if, for example, each ADI sweep in each of the three spatial dimensions takes about six or seven GMRES iterations to converge. Two sub-iterations of the implicit solver (which we use in our numerical examples) is about thirty-five times more expensive than the explicit solver. This suggests that the implicit solver is most advantageous when the time step Δt required by the explicit solver for stability is about 35 times smaller than that required by the implicit solver.

Remark 3.4: *In the context of the implicit solver presented in this thesis, many factors can affect the convergence rate of GMRES—such as the size of the time step Δt, the physical parameters (particularly Reynolds and Mach numbers), how well the solution is spatially resolved, the accuracy order of the temporal discretization, boundary conditions, etc.—but we have found the most influential factors to be the time step Δt and the Reyolds number. As $\Delta t \to 0$, the operator to be inverted is asymptotic to the identity, leading to faster convergence. On the other hand, large Reynolds number makes the operator more singularly perturbed (i.e., the coefficients of the second derivatives become smaller), and the spectrum of the centered finite difference preconditioner does not approximate the spectrum of the actual operator well (see [10, Ch. 15.3]). Nevertheless, we have observed that the performance results given in Table 3.2 are fairly generic.*

Chapter 4

Numerical results

This chapter presents a variety of numerical examples for the BDF-ADI solver in single domains, as well as the full multi-domain solver. The main purpose of the simulations is to showcase the stability and accuracy of the proposed methodologies. Once again we emphasize that these results represent the first numerical demonstrations of high-order time-accuracy for any ADI solver of the compressible Navier-Stokes equations.

Unless otherwise indicated, all simulations use the parameter values $\mathrm{Pr} = 0.71$ and $\gamma = 1.4$, and the (non-dimensional) viscosity and thermal conductivity are given by Sutherland's law [95, pp. 28–30]

$$\kappa(T) = \frac{1 + S_\kappa}{T + S_\kappa} T^{3/2} \quad \text{and} \quad \mu(T) = \frac{1 + S_\mu}{T + S_\mu} T^{3/2},$$

where S_κ and S_μ are the non-dimensionalized Sutherland constants, which, for definiteness, are set to $S_\kappa = S_\mu = 0.3$.

4.1 The BDF-ADI solver in single domains

The focus of this section is single domain, single core runs of the BDF-ADI solvers introduced in this thesis for two- and three-dimensional spatial domains and for orders s with $2 \leq s \leq 6$. In particular, some of these results demonstrate that the proposed solvers do enjoy the claimed spatial and temporal orders of accuracy, while other

examples demonstrate the methods' stability properties and general applicability. All of the numerical examples were obtained from runs on either a single core of an Intel Core i5-2520M processor with 4 GB of memory, or a single core of an Intel Xeon X5650 processor with 24 GB of memory.

We emphasize that in this section only, Chebyshev and Fourier spectral discretizations in space are used (as opposed to the FC method used in the full multi-domain solver). The reasons are twofold: First, the BDF-ADI algorithm was developed independently of any spatial approximation, and a goal of this thesis is to demonstrate wider applicability of the scheme. Second, the Chebyshev approximation is known for the $\mathcal{O}(1/N^2)$ clustering of points near the boundaries of the domain, and we demonstrate that this does not pose a challenge to the quasi-unconditional stability of the solvers.

4.1.1 Convergence in Cartesian domains

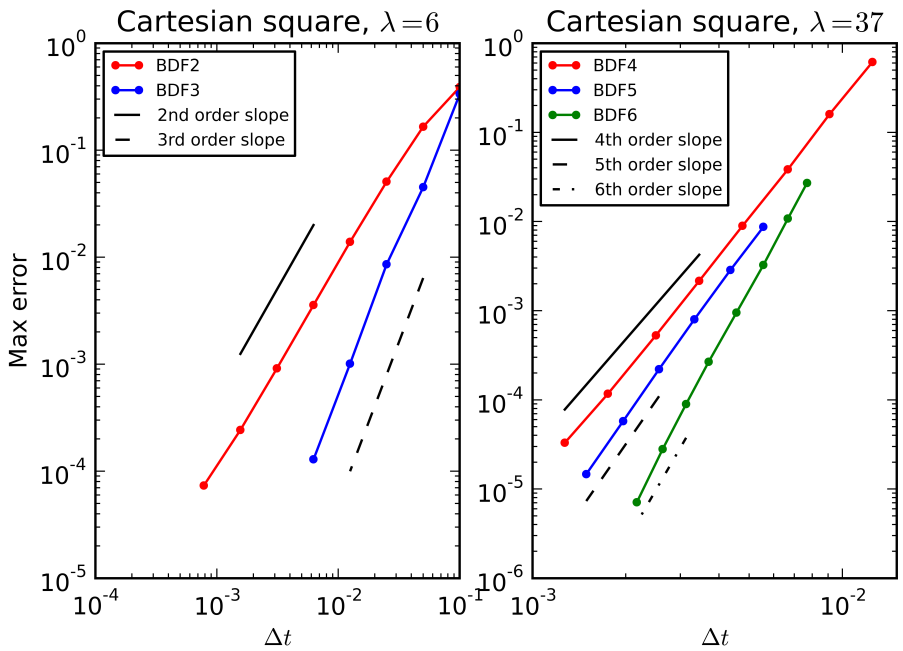


Figure 4.1: Convergence of the two-dimensional BDF-ADI solvers of orders $s = 2, \ldots, 6$ in a Cartesian square.

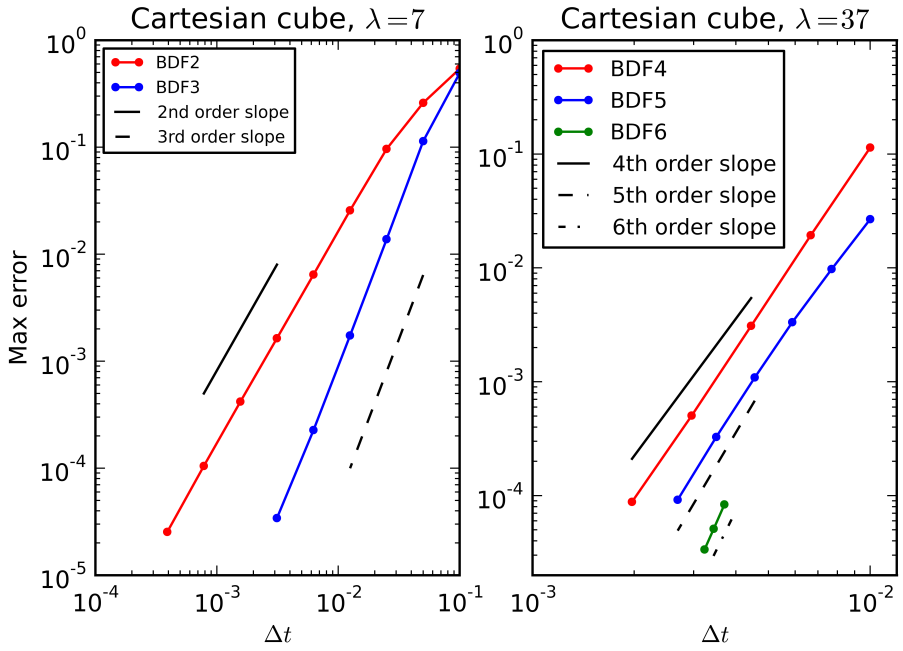


Figure 4.2: Convergence of the three-dimensional BDF-ADI solvers of orders $s = 2, \ldots, 6$ in a Cartesian cube.

Using the method of manufactured solutions (MMS), our first set of examples demonstrates that the proposed solvers achieve the expected temporal order of convergence in two- and three-dimensional problems. According to the MMS strategy, an arbitrary solution Q is prescribed, and a source term is added to the right-hand side of equations (2.5) or (2.6), as appropriate, in such a way that the proposed solution actually satisfies the equation. For this set of examples we use the manufactured solution

$$Q_j(x,t) = \alpha_j + \beta_j \sin(\lambda_j t + \phi_{j,t}) \prod_{i=1}^{d} \sin(2\pi\, x_i + \phi_{j,x_i}) \tag{4.1}$$

($d = 2, 3$), where Q_j is the jth component of the solution vector and where α_j, β_j, λ_j, ϕ_{j,x_i} are constants. The parameter values we use for the two- and three-dimensional solutions are given in Tables 4.1 and 4.2, respectively. The test geometry in this context is a Cartesian box of side-length equal to one. Only the velocity components and temperature T are prescribed at the boundary according to the manufactured

solution; the boundary values of the density are obtained as part of the solution process, as described in Section 2.5.2. The Reynolds number and Mach number in these examples are taken to equal $\mathrm{Re} = 10^3$ and $\mathrm{Ma} = 0.85$. The second- through sixth-order convergence of the BDF-ADI methods is demonstrated in Figures 4.1 and 4.2. All tests were run to a final time of $t = 1.0$, with the L^∞ error measured against the manufactured solution at that time. In the tests for the methods of orders four, five and six, the temporal frequency of the manufactured solution was increased, so that the errors associated with the temporal discretization are large compared to the spatial discretization errors. This allows the convergence slopes to be observed at moderate Δt without requiring very fine spatial resolution.

Q_j	α_j	β_j	λ_j	$\phi_{j,t}$	$\phi_{j,x}$	$\phi_{j,y}$
u	0	1	6 (37)	1	0	0
v	0	1	6 (37)	2	0	0
ρ	1	0.2	6 (37)	3	0.5	1.5
T	1	0.2	6 (37)	4	1	2

Table 4.1: Parameters for two-dimensional manufactured solution. The temporal frequencies λ_j not in parentheses are the ones used in the convergence tests for the methods of orders $s = 2, 3$, while the ones in parentheses are those used in the order $s = 4, 5, 6$ tests.

Q_j	α_j	β_j	λ_j	$\phi_{j,t}$	$\phi_{j,x}$	$\phi_{j,y}$	$\phi_{j,z}$
u	0	1	7 (37)	-1	0	0	0
v	0	1	7 (37)	-2	0	0	0
w	0	1	7 (37)	-3	0	0	0
ρ	1	0.2	7 (37)	-4	4	7	14
T	1	0.2	7 (37)	-5	5	6	15

Table 4.2: Parameters for three-dimensional manufactured solution. The temporal frequencies λ_j not in parentheses are the ones used in the convergence tests for the methods of orders $s = 2, 3$, while the ones in parentheses are those used in the order $s = 4, 5, 6$ tests.

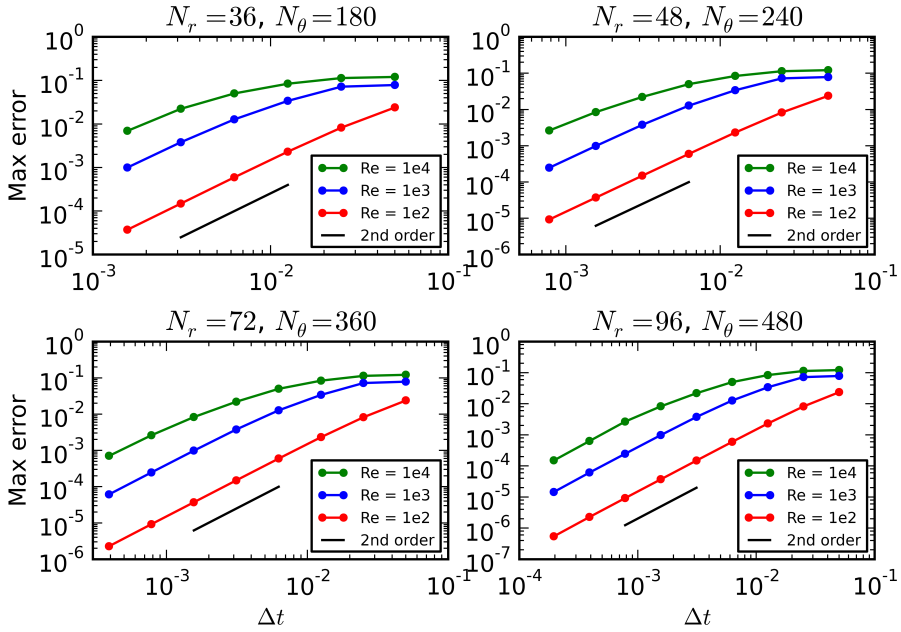


Figure 4.3: Convergence of the BDF2 ADI solver in an annulus with various mesh discretizations and Reynolds numbers.

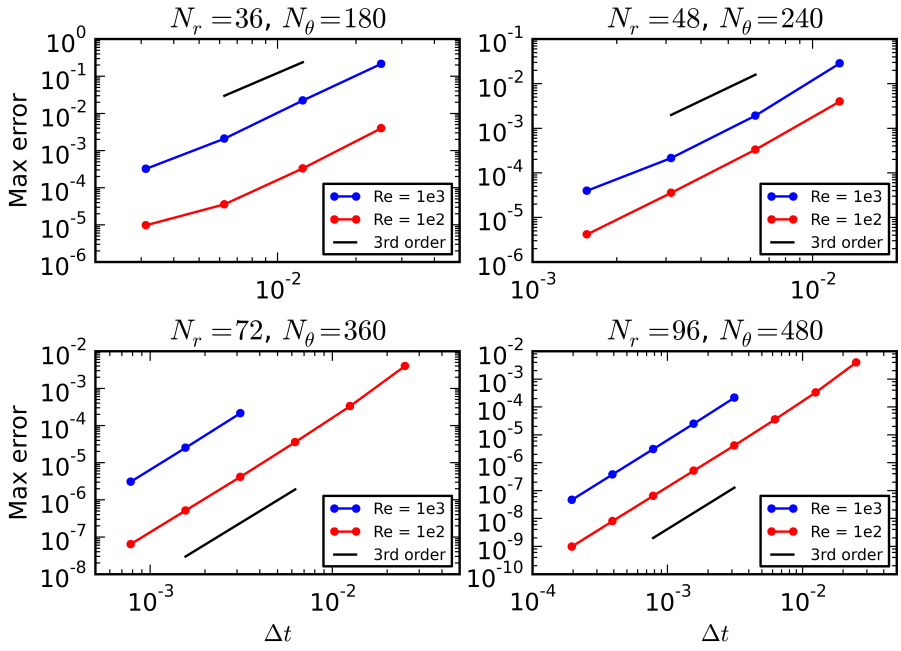


Figure 4.4: Convergence of the BDF3 ADI solver in an annulus with various mesh discretizations and Reynolds numbers.

4.1.2 Convergence in an annulus

Next, we demonstrate the convergence of the solver in two-dimensions with a physical flow example at Ma = 0.8 in an annulus with inner radius 0.1 and outer radius 0.5 using Chebyshev collocation in the radial direction and Fourier collocation in the azimuthal direction. The flow starts with an initial condition of zero in all fields except temperature, which is 1.0, and the density, which equals the sum of the scalar 1.0 plus the sum of two Gaussian functions of the form

$$a \exp\left(-\frac{(x-x_0)^2+(y-y0)^2}{2\sigma^2}\right); \tag{4.2}$$

with parameters $a = 0.3$, $\sigma = 0.1$, $x_0 = -0.2$, $y_0 = 0.2$ and $a = -0.2$, $\sigma = 0.07$, $x_0 = 0.2$, $y_0 = 0$, respectively. For time between $t = 0$ and $t = 0.5$, the rotation of the inner cylinder is ramped up smoothly until it reaches a tangential velocity of 1.0. There is also a temperature source term added ($\sin(2\pi t)$ times a Gaussian in space given by equation (4.2) with $a = 2.5$, $\sigma = 0.05$, $x_0 = -0.2$, $y_0 = -0.2$). The convergence of the solver is estimated by measuring the L^∞ error against the solution on the finest mesh size h and time step Δt ($N_r = 108$, $N_\theta = 540$, $\Delta t = 0.1 \times 2^{-10}$) at time $t = 1.0$. Figures 4.3 and 4.4 verify the expected rates of convergence at various Reynolds numbers.

4.1.3 Flow over a bumpy plate at $\mathrm{Re} = 10^6$

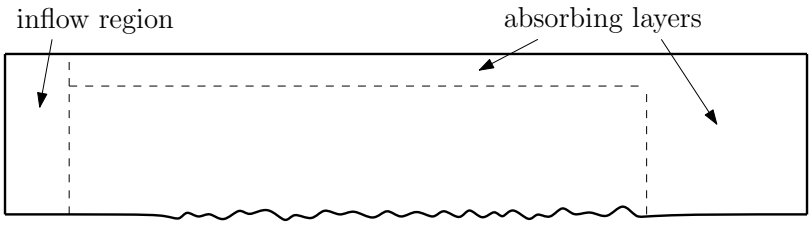


Figure 4.5: Schematic set-up of unsteady flow over a bumpy plate (not to scale).

Next, a demonstration of boundary layer flow over a "bumpy" plate in two-

dimensions at high Reynolds number is given. Here the domain is such that the left edge is $x_l = 1$, the right edge is $x_r = 4$, the top edge is $y_t = 0.2$, and the bottom edge is

$$y_b(x) = \exp\left(-\left(\frac{x - 2.5}{1.2}\right)^{12}\right) \sum_{m=1}^{4} a_m \sin(c_m x),$$

where the coefficients are given by $a_m = 5 \times 10^{-4}$, 8×10^{-4}, 6×10^{-4}, 4×10^{-4} and $c_m = 300$, 207, 161, 124. The interior points of the domain are generated by means of transfinite interpolation [42]. A total of 1536 (resp. 96) Chebyshev collocation points were used in the horizontal direction (resp. in the vertical direction). A schematic illustration of the set-up is provided in Figure 4.5.

To initialize the flow and impose boundary conditions, we first obtain the solution of the compressible boundary layer equations. Here we provide a brief overview of their derivation; a more detailed discussion can be found in, e.g., [95, Ch. 7]. For simplicity, we assume the viscosity and thermal conductivity are linear functions of temperature ($\mu(T) = \kappa(T) = T$) and the Prandtl number is $\mathrm{Pr} = 1$. The x and y directions are tangent and normal to the boundary respectively, and we take the free-stream values as $y \to \infty$ to be $u_\infty = 1$, $v_\infty = 0$, $T_\infty = 1$, $\rho_\infty = 1$.

The boundary layer equations are obtained by transforming the steady ($Q_t = 0$) two-dimensional Navier-Stokes equations (2.3) with the change of variable $y = \delta Y$, where $\delta = \mathrm{Re}^{1/2}$ is the characteristic length scale of the boundary layer. Furthermore, the solution components are assumed to be perturbations of the free-stream values of the form $u = u_\infty + \delta u_1$, $v = \delta v_1$, $T = T_\infty + \delta T_1$, $\rho = \rho_\infty + \delta \rho_1$, which leads to a set of equations for the inner solutions (terms with subscript 1). Using the similarity variable $\bar{\eta} = \bar{\eta}(x^{-1/2} Y)$ together with the Howarth transformation [49], we obtain the

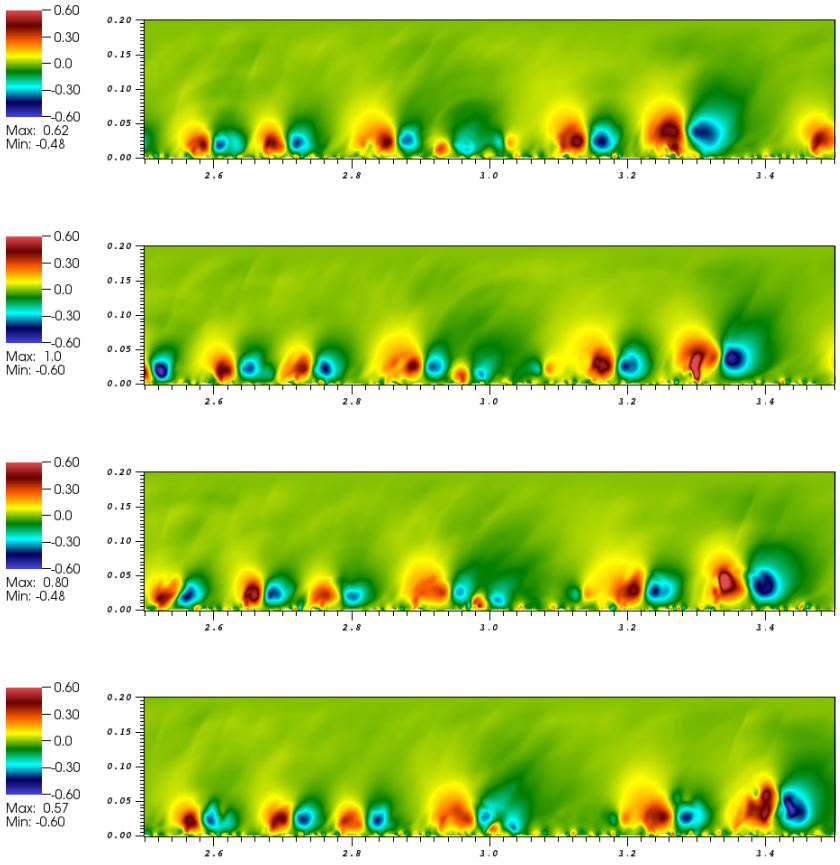


Figure 4.6: Vertical velocity in two-dimensional boundary layer flow over a bumpy plate, showing the presence of vortices and acoustic waves. From top to bottom, the solution times for the figures are $t = 9.76,\ 9.82,\ 9.88,\ 9.94$.

following simplified set of equations for $\bar{\eta}$, u_1, v_1, ρ_1, and T_1 as functions of x and Y:

$$\frac{\partial}{\partial Y}\bar{\eta} = \rho_1,$$

$$\rho_1\, u_1 = f'(\bar{\eta}),$$

$$\rho_1\, v_1 = \frac{1}{2}x^{-1/2}\left(\bar{\eta}\, f'(\bar{\eta}) - f(\bar{\eta})\right),$$

$$T_1 = u_1 + T_{\text{wall}}(1-u_1) + \frac{1}{2}(\gamma-1)\text{Ma}^2(u_1 - u_1^2),$$

$$\rho_1 T_1 = 1,$$

where T_{wall} is the temperature at the wall and f is the solution of the Blasius equation

$$f''' + \frac{1}{2} f\, f' = 0,$$
$$f(0) = f'(0) = 0,$$
$$f'(\eta) \to 1 \text{ as } \eta \to \infty.$$

The similarity variable $\bar{\eta}$ is obtained by eliminating the other unknowns and using a Newton-Kantorovich iterative solver [10, App. C] with initial guess computed by standard fourth order Runge-Kutta. The rest of the unknowns can then be obtained explicitly from the above relations. The solution Q_{ref} of the boundary layer equations is used to provide the initial condition and boundary conditions at inflow and in the absorbing layers of the computational domain.

The boundary conditions for this example are as shown in Figure 4.5—no-slip conditions are imposed on the bottom boundary ($\mathbf{u}_{\text{wall}} = 0$ and $T_{\text{wall}} = 1$), an absorbing layer of thickness 0.05 is used at the top of the domain, another absorbing layer of thickness 0.5 is used at right, and inflow conditions are imposed in a region of thickness 0.1 at the left. For each absorbing layer, a term of the form $\sigma(\xi, \eta) Q_{\text{ref}}$ is added to the right-hand side of the PDE (2.6) and σI is added to the matrix M^0, where I is the identity. The variable coefficient σ is given by

$$\sigma(\xi, \eta) = A \left(1 - \psi \left(\frac{d(\xi, \eta)}{L} \right) \right) \tag{4.3}$$

where A is the absorption factor, L is the width of the layer, $d(\xi, \eta)$ is the distance to the boundary in question, and the function ψ is given by

$$\psi(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x \geq 1 \\ \left[1 + \exp\left(\frac{1}{x} - \frac{1}{1-x} \right) \right]^{-1}, & 0 < x < 1. \end{cases} \tag{4.4}$$

For the top boundary we use $A = 50$, $L = 0.05$, and for the right boundary we use $A = 20$, $L = 0.5$. The boundary layer solution Q_{ref} is imposed in the inflow region. Figure 4.6 shows the vertical velocity for various times, with $\text{Re} = 10^6$, $\text{Ma} = 0.85$, and $\Delta t = 10^{-3}$. With this discretization ($\Delta x_{\min} = 3.1 \times 10^{-6}$ and $\Delta y_{\min} = 5.4 \times 10^{-5}$), an explicit solver would require a prohibitively small time step ($\lesssim 10^{-6}$) for stability.

4.1.4 Wall bounded Taylor-Couette flow

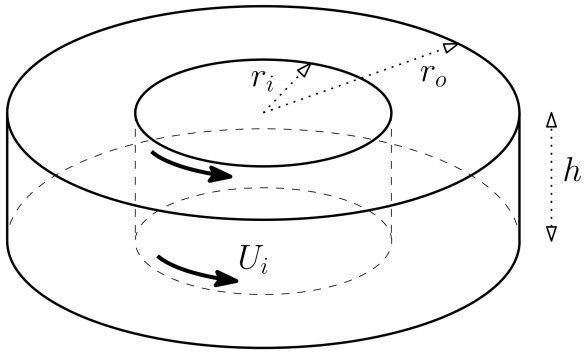


Figure 4.7: Geometry of Taylor-Couette flow. The fluid is confined to the region between two cylinders of radii r_i and r_o, and two planes separated by a length h. The inner cylinder rotates with speed U_i, while all other boundaries remain stationary.

We also verify the three-dimensional solver by simulation of Taylor-Couette flow, which is flow of a fluid between two concentric rotating cylinders. Most studies of Taylor-Couette flow deal with incompressible fluids, but the dynamics for subsonic compressible gases are similar, as shown in [52, 65]. The geometry (illustrated in Figure 4.7) is defined by the inner radius r_i, outer radius r_o, and height h. In what follows, we consider only the case where the inner cylinder rotates and the outer cylinder together with the top and bottom walls are stationary. In this case, the Reynolds number Re is defined with respect to the velocity of the inner cylinder.

The small aspect ratio regime ($\Gamma = \frac{h}{r_o - r_i} \approx 1$ or less) has been extensively studied both numerically and experimentally (in the incompressible case)—in part because of the property that two or more stable flows can exist for the same values of the system

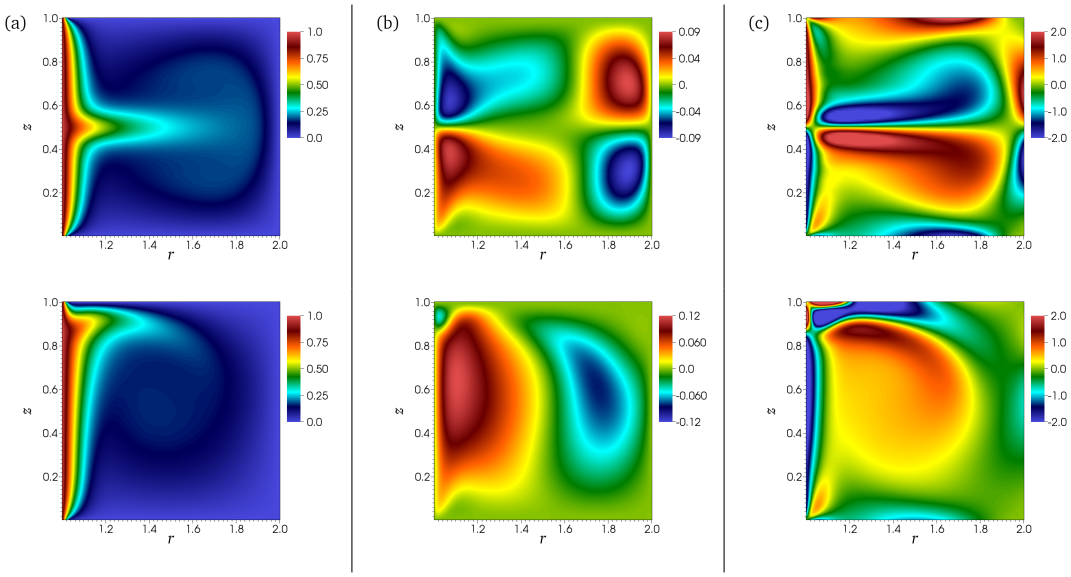


Figure 4.8: Profiles of the (a) azimuthal velocity, (b) vertical velocity, and (c) azimuthal component of vorticity in small-aspect-ratio Taylor-Couette flow at Ma = 0.2 and Re = 700. The top (bottom) row has the profiles of the two-cell (one-cell) stable mode.

parameters Γ, Re, r_o, r_i [66, 71]. For $\Gamma = 1$ and radius ratio $r_i/r_o = 0.5$, there is only one stable flow at small Re, which is characterized by two axisymmetric toroidal vortices, one on top of the other, as pictured in the upper frame of Figure 4.8 (c). At about Re ≈ 133, this mode becomes unstable. The stable mode is then characterized by a single large axisymmetric toroidal vortex in the center and a smaller one in the inner upper corner, as shown in the lower frame of Figure 4.8 (c). Both modes are stable in the range $603 \lesssim \text{Re} \lesssim 786$.

Starting from the same initial condition (zero velocity, density and temperature equal to 1.0) and ending at the same final inner cylinder rotation at Re = 700 and Ma = 0.2, we arrive at both stable modes. For the two-cell mode, the inner cylinder velocity as a function of time is given by

$$U_i(t) = \psi\left(\frac{t}{160}\right),$$

where ψ is defined in (4.4). To demonstrate the one-cell mode, the cylinder velocity

is given by

$$U_i(t) = 0.4\psi\left(\frac{t}{10}\right) + 0.6\psi\left(\frac{t-150}{150}\right).$$

The domain is discretized using 48 Chebyshev collocation points in the radial and z directions and 64 Fourier collocation points in the azimuthal direction. No-slip isothermal boundary conditions are used on all walls, with the axial velocity on the top and bottom boundaries given by

$$u_\theta(r,t) = \exp\left(-\left(2\sqrt{\mathrm{Re}}(r-r_i)\right)^2\right) U_i(t).$$

The time discretization for both simulations was set at $\Delta t = 0.02$ and simulations were stopped at $t = 400$. At $\mathrm{Ma} = 0.2$, there is less than 0.5% deviation in the density from the initial condition $\rho = T = 1$ throughout the simulations. The presence of corners in the geometry undoubtedly reduces the accuracy of the solutions; nevertheless, Figure 4.8 shows both modes at $t = 300$, which compares well to the experimental and numerical results in the literature for the incompressible case [66,71]—as it should given the low value of the Mach number considered.

4.2 Multi-domain implicit-explicit examples

This section showcases the full multi-domain implicit-explicit algorithm with one example each in two and three spatial dimensions. These results were obtained on the cluster described in Section 3.5.2.

4.2.1 Unsteady flow past a cylinder

This section presents numerical results for the problem of flow past a cylinder in two spatial dimensions with $\mathrm{Re} = 200$. At higher values of the Reynolds number three-dimensional effects become important thus reducing the relevance of two-dimensional simulations.

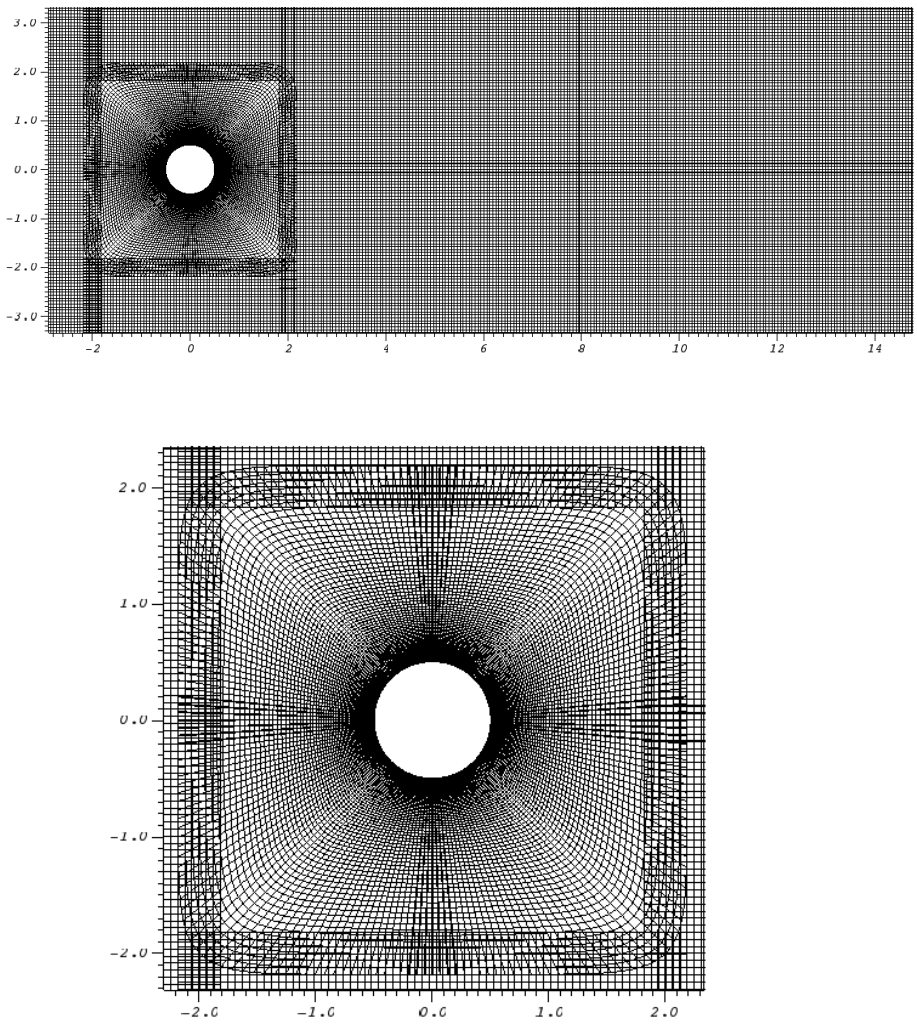


Figure 4.9: Two close-ups of the mesh used in the numerical experiments of flow past a cylinder. The bottom figure shows the clustering of points near the cylinder surface to spatially resolve the boundary layer.

The two-dimensional cross-section is the region inside the rectangle $[-8, 24] \times [-8, 8]$ and outside a circle of radius $r_c = 0.5$ centered at the origin. This region is divided into 13 overlapping patches: four curvilinear annular-like sections surrounding the cylinder and nine rectangular patches away from the cylinder. Our approach to generation of the curvilinear meshes is discussed in Appendix B. Figure 4.9 shows a

portion of the domain including one of the discretizations we have used. Given upper and lower bounds $h_{\max}$ and $h_{\min}$ on the acceptable spatial grid size, the meshes are generated in such a way that the radial spacing is guaranteed to be $\leq h_{\min}$ near the surface of the cylinder and $\leq h_{\max}$ at the outer edge of the annular sections. The mesh spacing in rectangular Cartesian domains is approximately $h_{\max}$. To ensure this, the number of discretization points in the x direction of a Cartesian domain, for example, is taken to equal $1+\frac{L}{h_{\max}}$ rounded to the nearest integer, where L is the side of the rectangle in the x direction. In this example the four curvilinear patches use the BDF-ADI algorithm to march the solution forward in time, while the rectangular patches use explicit Adams-Bashforth time-marching.

For this example we have assumed no-slip and isothermal ($T = 1$) boundary conditions at the boundary of the cylinder. At the outer edges of the domain, an absorbing layer is used. In detail, let $\partial^e\Omega$ denote the outer boundary of the domain Ω (i.e., not including the boundary located at the surface of the cylinder). For an absorbing layer of width w_a, the matrix M^0 in the quasilinear-like formulation (2.6) is given by $M^0 = \sigma(x, y)I$, where I is the identity operator and σ is the function

$$\sigma(x,y) = \begin{cases} A_a \left(1 - \frac{d(\,(x,y),\partial^e\Omega)}{w_a}\right)^{p_a} & d(\,(x,y),\partial^e\Omega) < w_a \\ 0 & \text{otherwise,} \end{cases} \tag{4.5}$$

where $d(\,(x,y),\partial^e\Omega)$ is the distance from the point (x,y) to the boundary $\partial^e\Omega$, and where, for definiteness, we have selected the constants $A_a = 5.0$ and $p_a = 4$. The source term σQ_{ref}, in turn, is added to the right-hand side of equation (2.6), where Q_{ref} is the vector of unknowns corresponding to the initial conditions (4.6).

For our purposes, the initial condition and right-hand side source terms for the equations satisfy three requirements: 1) They must start the simulation in such manner that the time derivative of all the fields at $t = 0$ is 0 (so that the initial condition can be used to initialize all the previous time steps of the multi-step time marching

schemes, without reducing the temporal order of accuracy), 2) they reduce the amount of time it takes for the solution to settle into the periodic vortex shedding mode by breaking the symmetry of the flow early on and 3) they satisfy the no-slip, isothermal boundary condition at the surface of the cylinder and the free stream conditions in the absorbing layer for all times t.

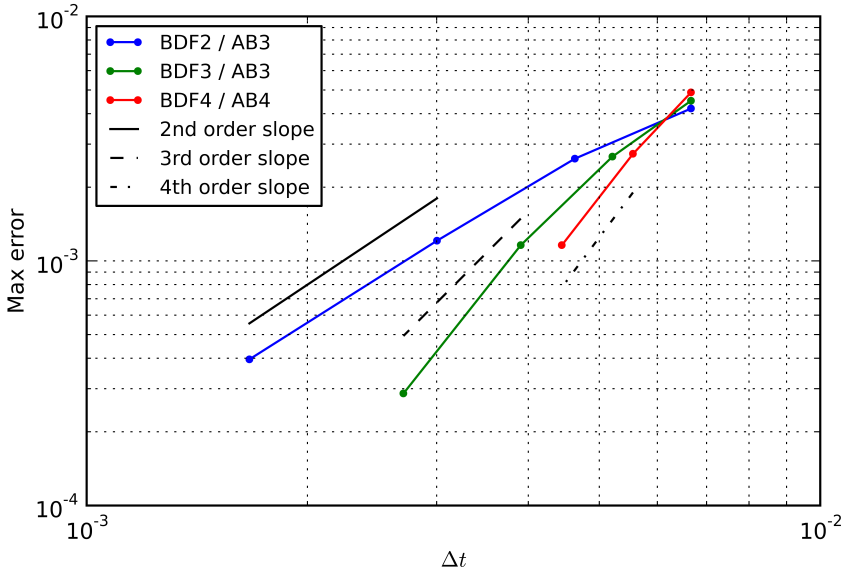


Figure 4.10: Temporal convergence of the solver for flow past a cylinder at time $t = 1.0$, with $\mathrm{Re} = 200$ and $\mathrm{Ma} = 0.8$.

In view of these goals the flow is initialized with the radially symmetric fields

$$\rho = T = 1, \quad v = 0, \quad u = u_0 \psi_s(r - r_c), \tag{4.6}$$

where u_0 is the freestream velocity and ψ_s is the smooth step function

$$\psi_s(x) = \begin{cases} 0 & x \leq 0 \\ \left(1 + \exp\left(\frac{1}{x} - \frac{1}{1-x}\right)\right)^{-1} & 0 < x < 1 \\ 1 & x \geq 1. \end{cases} \tag{4.7}$$

This initial condition satisfies the no-slip and isothermal conditions at the surface of

the cylinder as well as the uniform free-stream conditions in the absorbing layer. A source term is added to the right-hand side of the equations that enables the initial flow to transition smoothly from the initial condition given above on to the truly physical regime for which the right-hand-side resulting from use of a manufactured solution actually vanishes. The transition is effected by smoothly taking the right-hand side source term to zero over the time interval $0 \leq t \leq t_0$ by multiplying it by $\psi_s(1 - t/t_0)$ for some fixed time constant t_0, allowing the physics of the configuration to evolve the solution thereafter. This ensures the solution is smooth in time at $t = 0$. To break the symmetry of the flow, we sometimes add an additional source term to the right-hand side of the v equation:

$$\psi_w(t; 0, 0.2, 0.8, 1) f_G(x, y, t),$$

where ψ_w is the smooth windowing function

$$\psi_w(x; a, b, c, d) = \psi_s\left(\frac{x-a}{b-a}\right)\psi_s\left(\frac{d-x}{d-c}\right), \quad a < b \leq c < d \tag{4.8}$$

and f_G is the Gaussian pulse

$$f_G(\mathbf{x}, t) = A_0 e^{\frac{|\mathbf{x}-\mathbf{x_0}|^2}{2\sigma_0^2}} \sin(\omega_0 t), \tag{4.9}$$

with parameters $\mathbf{x}_0 = (1.5, 1)$, $A_0 = -0.4$, $\sigma_0^2 = 0.25$, and $\omega_0 = 2\pi$.

Having presented the set-up details associated with the problem of flow past a cylinder we now put forth a variety of numerical results that illustrate the properties of the proposed methods. We first consider the convergence of the solver as h and Δt are simultaneously refined. For this test, we use $\mathrm{Re} = 200$ and $\mathrm{Ma} = 0.8$ and the time over which the manufactured initial condition is phased out is $t_0 = 0.5$. The mesh size values $h = h_{\max} = 0.06$, 0.05, 0.04, 0.03, and 0.02 are used for the Cartesian domains and $h_{\min} = 0.1h$ is used at the cylinder surface. In view of accuracy considerations,

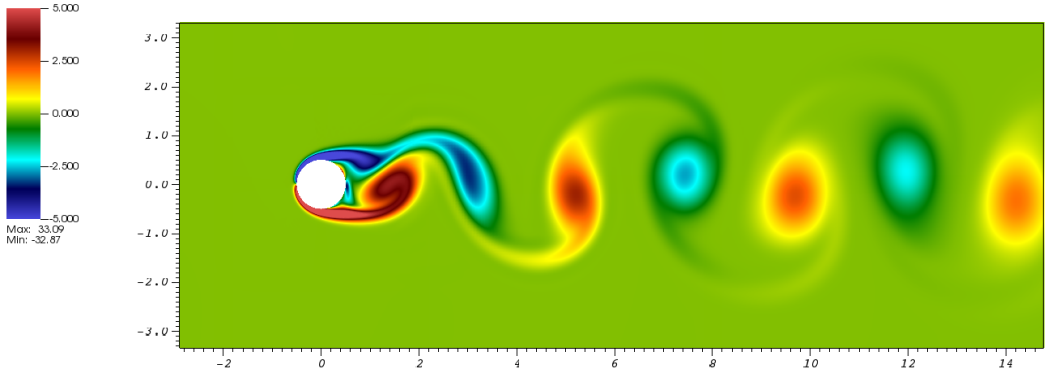


Figure 4.11: Snapshot of the vorticity in a simulation of flow past a cylinder with $\mathrm{Re} = 200$, $\mathrm{Ma} = 0.2$ at time $t = 82.8$.

the time step Δt is chosen to be of the form $\Delta t = Ch^p$ for some power p depending on the order of the method: $p = 2,\ 1.3,\ 1$ for $s = 2,\ 3,\ 4$ respectively. The constant C is given by $C = 0.06^{-p}/150$, so that $\Delta t = 1/150$ when $h = 0.06$. Figure 4.10 plots the L^∞ error at time $t = 1.0$ versus Δt confirming the expected temporal order of accuracy of the solvers. The error is estimated by comparing to the solution on the finest grid ($h = 0.02$) with $\Delta t = 1/800$ using fourth-order BDF-ADI and AB4 time marching.

Next, we show the results of a full simulation of unsteady flow past a cylinder with $\mathrm{Re} = 200$ and $\mathrm{Ma} = 0.2$. For this test, the mesh we use has $h_{\max} = 0.06$ and $h_{\min} = 0.006$. The manufactured initial condition and v velocity source term are phased out at $t_0 = 5.0$. Third order time-marching is used in all zones—BDF3-ADI in the implicit zones and AB3 in the explicit zones. The time step is chosen to be $\Delta t = 3.33e-3$ and the simulation is run for 30,000 times steps. Figure 4.11 shows the characteristic von Karman vortex street at time $t = 82.8$. The Strouhal number St (the non-dimenional frequency of vortex shedding) is found by tracking the vertical velocity at the point $(0, 1)$, and is estimated to be $\mathrm{St} = 0.202$, which is consistent with the results reported in [2] and experimental references therein for the same Reynolds number and a slightly different Mach number. In Figure 4.12, snapshots of

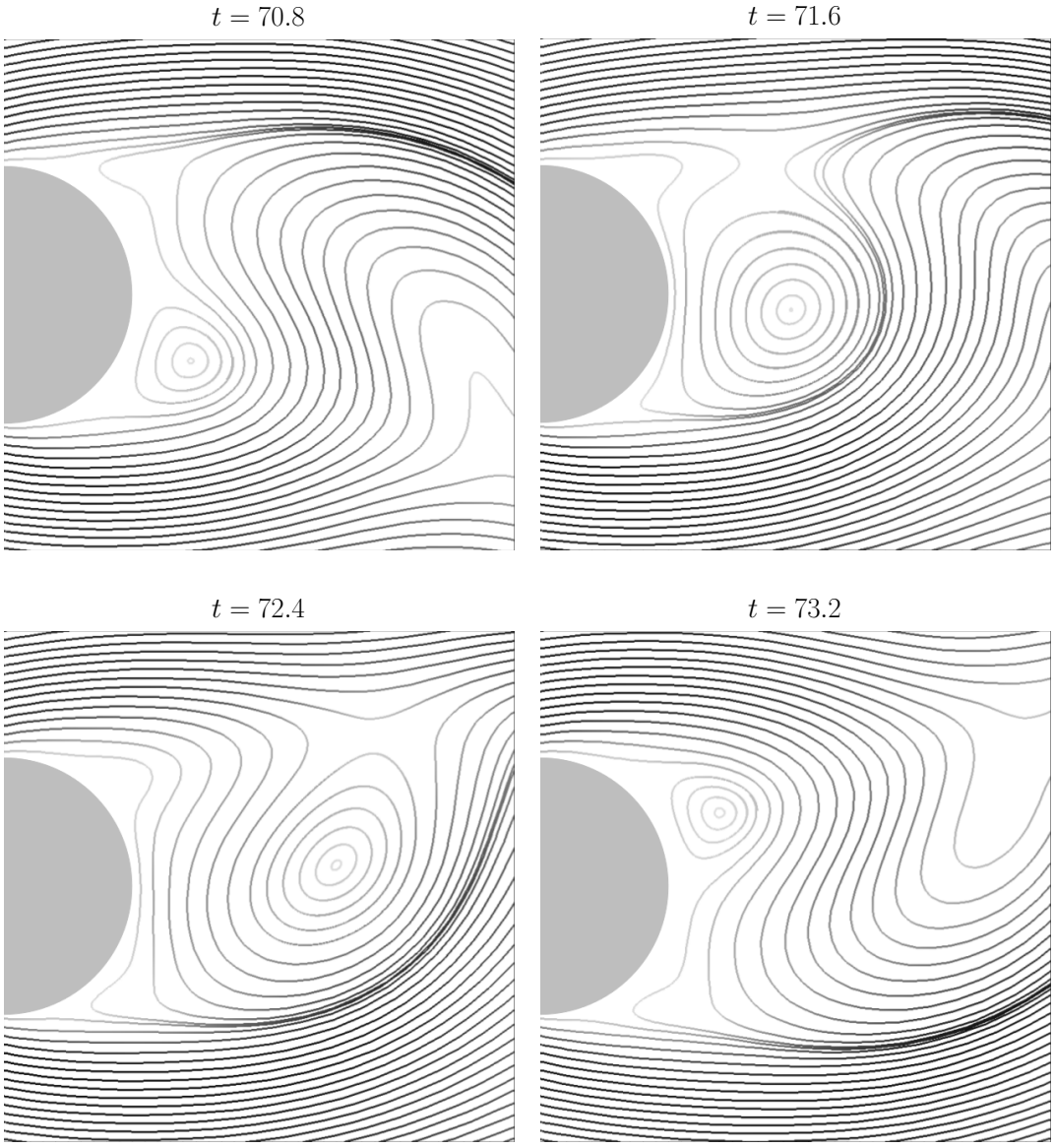


Figure 4.12: Time evolution of streamlines in flow past a cylinder at Re = 200 and Ma = 0.2. Darker shading of the streamline corresponds to a higher magnitude of the velocity at that point.

the streamlines near the cylinder show the time evolution of vortex shedding.

4.2.2 Unsteady flow past a sphere

In this section, we present the results of three-dimensional flow past a sphere. The domain is the region inside the box $(x, y, z) \in [-4, 14] \times [-4, 4] \times [-4, 4]$ and outside the sphere of radius $r_s = 0.5$ centered at the origin. This region is divided into 12 larger overlapping patches: two curvilinear patches forming a modified "Yin-Yang" mesh [51] surrounding the sphere and ten Cartesian patches away from the sphere. Generation of the curvilinear meshes is discussed in Appendix B. The BDF-ADI method is used in the curvilinear patches and explicit AB time marching is used in all Cartesian patches. Given values h_s, h_c, h_f, and h_u, to be selected in what follows, meshes are generated in such a way that the radial spacing is h_s near the surface of the sphere, h_c at the edge of the Yin-Yang meshes and in the Cartesian patches directly downstream from the sphere, h_f in the far-field downstream Cartesian patches, and h_u in the Cartesian patches that are not downstream from the sphere.

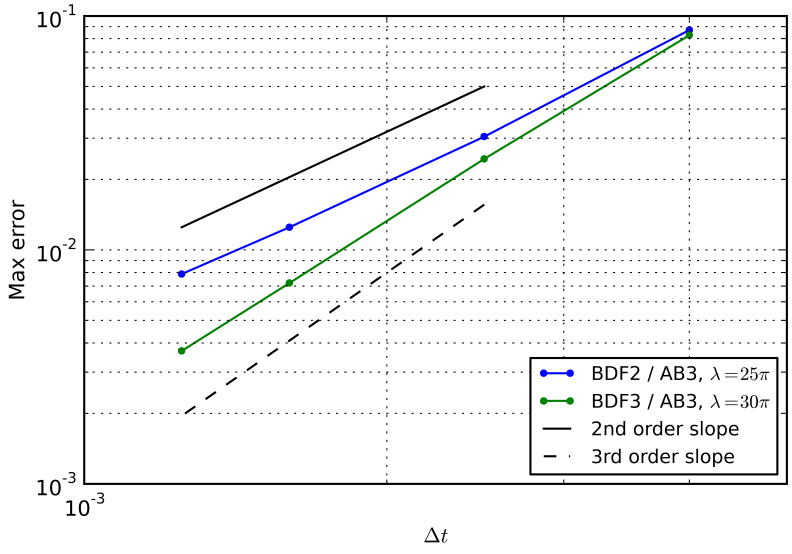


Figure 4.13: Temporal convergence of the three-dimensional multi-domain solver using the method of manufactured solutions at time $t = 1.0$, with $\mathrm{Re} = 500$ and $\mathrm{Ma} = 0.8$.

We use the method of manufactured solutions to demonstrate the temporal convergence of the second and third order solvers at $\mathrm{Re} = 500$ and $\mathrm{Ma} = 0.8$. For this

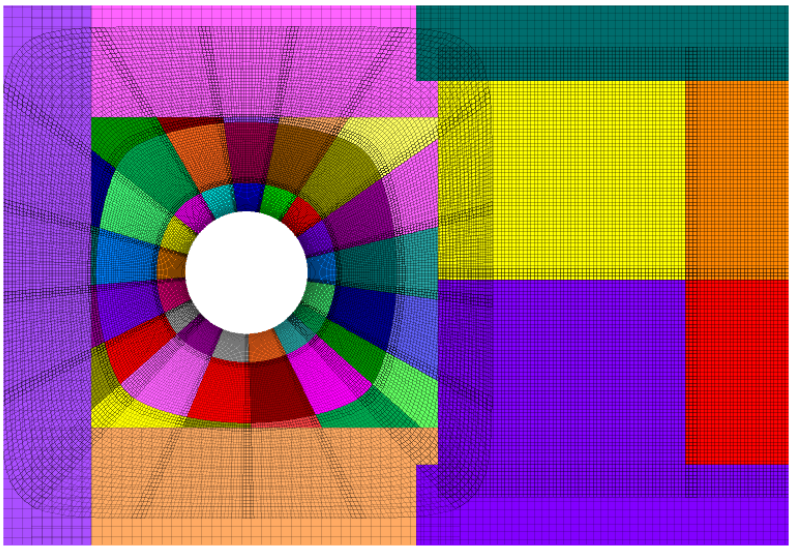


Figure 4.14: Two-dimensional slice of the mesh for flow past a sphere. The coloring shows the sub-patch decomposition.

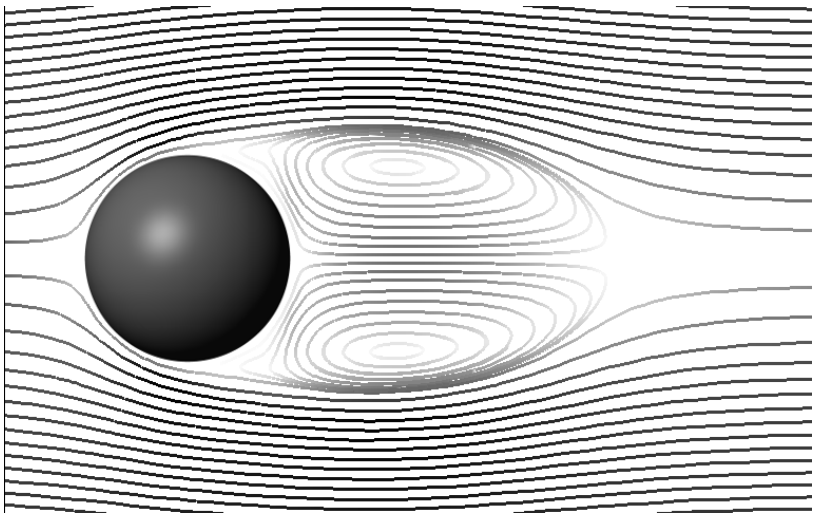


Figure 4.15: Two-dimensional x-z slice of the streamlines in a simulation of flow past a sphere with Re = 500, Ma = 0.5 at time $t = 12$. Darker shades in the streamlines indicate higher velocity magnitude.

test, we use a smaller domain $[-4, 4]^3$ and set the spatial mesh-size parameters to be $h_s = 0.004$, $h_c = h_f = h_u = 0.04$. The manufactured solution is given by equation (4.1) with parameters as in Table 4.2, except for λ_j which is set to $\lambda_j = 25\pi$ and $\lambda_j = 30\pi$ for the second and third order solvers respectively. Figure 4.13 shows

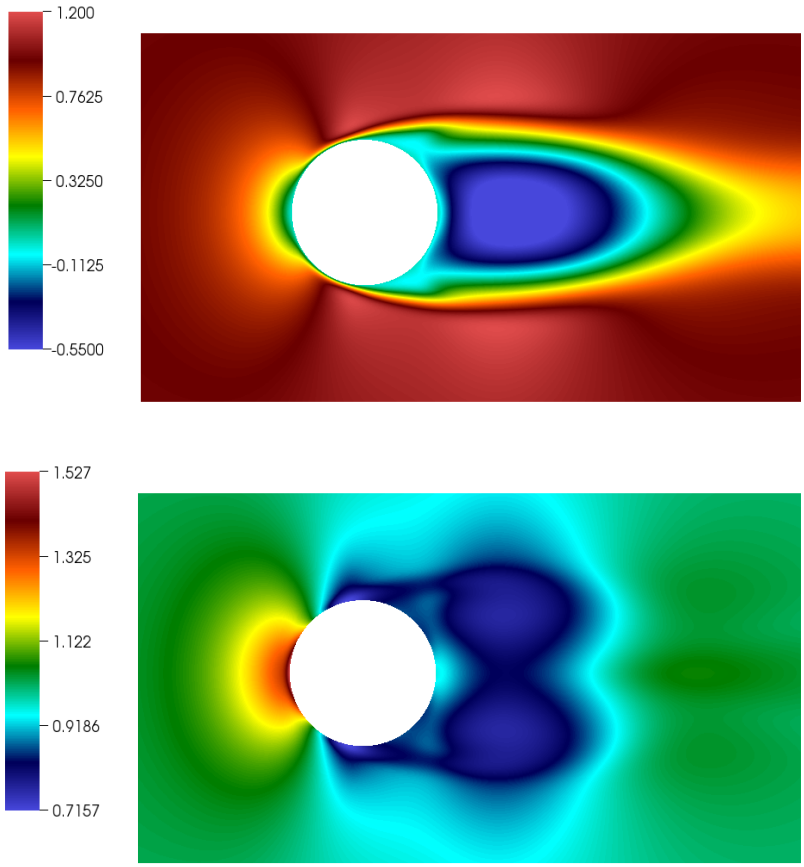


Figure 4.16: Two-dimensional x-z slice of the x-velocity (top) and density (bottom) in flow past a sphere at Re = 500 and Ma = 0.5.

nearly perfect second and third order convergence slopes.

Next, we provide a physical example of flow past a sphere at Re = 500 and Ma = 0.5. The mesh size parameters in this example are $h_s = 0.005$, $h_c = 0.3$, $h_f = 0.6$, and $h_u = 0.08$. The 12 larger patches are divided into 365 subpatches and one processor is assigned to each subpatch. The number of points in each subpatch is such that the costs of one subiteration in an implicit subpatch and one explicit solve in an explicit subpatch are approximately equal. Figure 4.14 shows a two-dimensional slice of a portion of the domain with this discretization.

The boundary conditions at the surface of the sphere are no-slip and isothermal ($T = 1$), and an absorbing layer is used at the outer edges of the domain (as described

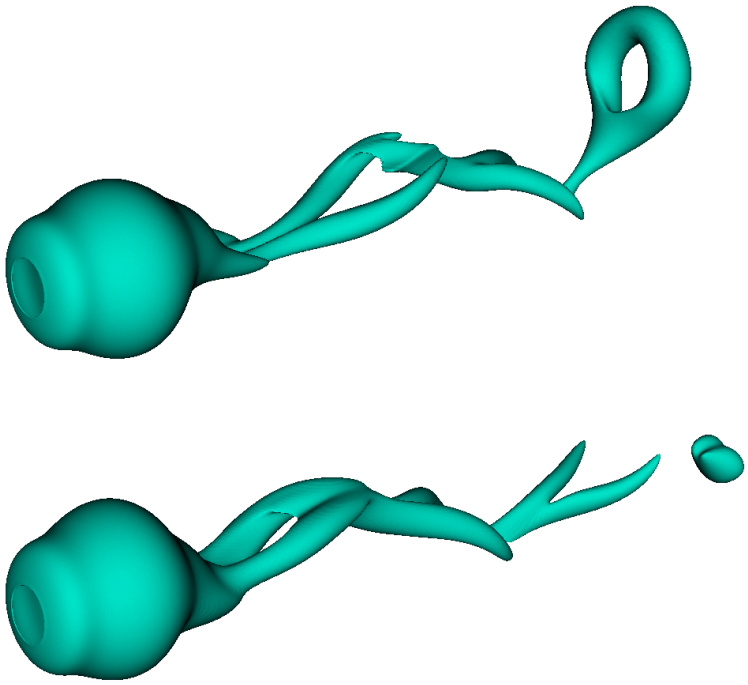


Figure 4.17: Isosurfaces of the density ($\rho = 0.95$) in three-dimensional flow past a sphere at times $t = 64.5$ (top) and $t = 77.5$ (bottom), showing the apperance of hairpin vortices in the flow field.

in the previous section) using the parameters $A - a = 5.0$, $p_a = 4$, and $w_a = 1.0$ in equation (4.5). As in Section 4.2.1, we use the method of manufactured initial conditions: the fields are initialized with the functions

$$\rho = T = 1, \quad v = w = 0, \quad u = u_0 \psi_s(r - r_s), \tag{4.10}$$

where u_0 is the freestream velocity and ψ_s is the smooth step function (4.7). A right-hand side source term is added to the equations that enforce the initial condition, which is made to vanish smoothly by multiplying it with $\psi_s(1 - t/t_0)$.

Figure 4.15 shows a slice of the streamlines in the x-z plane at $t = 12$. At this early time in the simulation, the developing flow is still laminar and axially symmetric (about the x-axis), exhibiting the characteristic axially symmetric vortex behind the sphere. Figure 4.16 shows a slice of the x-velocity and the density at the same time,

which shows that vortices may be characterized by areas of low density relative to the freestream density. In Figure 4.17, snapshots of density isosurfaces indicate the shedding of "hairpin" shaped vortices, the character of which are qualitatively consistent with the results of similar experiments and numerical simulations for incompressible flow [50, 55, 56, 74].

Chapter 5

Conclusions and future work

This thesis introduced a multi-domain implicit-explicit solution strategy for the compressible Navier-Stokes equations. A solid theoretical foundation, including proofs of unconditional stability and quasi-unconditional stability for related linear problems was presented for the implicit component (the BDF-ADI solver) of the proposed multi-domain implicit-explicit solver. The explicit overset FC strategy was extended to the implicit-explicit context. The resulting overall capability provides a basic framework for solvers applicable to significantly more complex geometries, physical problems, and PDEs. The numerical examples presented in this text demonstrate the meaningful advantages inherent in the proposed solvers, which include higher-order accuracy in both space and time, dispersionlessness, quasi-unconditional stability for the BDF-ADI solvers, and efficient parallel implementation. The methodologies introduced in this thesis thus provide general-geometry solvers that can resolve boundary layers and other flow features without recourse to crippling time steps, they can evaluate flow fields with high accuracy, and, in particular, they provide quasi-unconditional stability in boundary regions with *theoretically and numerically demonstrated* orders of temporal accuracy as high as $s = 6$.

Much remains to be done. The theoretical analyses presented in this thesis should be extended to problems more closely resembling the compressible Navier-Stokes equations—including for example variable coefficient equations, higher dimen-

sions, systems of equations, and nonlinearities. Future analyses should also include a wider array of spatial discretizations of interest, such as finite differences, Chebyshev spectral methods, and FC. Importantly, further, it should be useful to study the possible applicability of the ideas introduced in this thesis to other types of splitting schemes where intermediate variables arise which require use of potentially non-physical boundary conditions (e.g., schemes that split velocity and pressure components in the incompressible Navier-Stokes equations; see for example the concise discussion [76, pp. 74-76]).

The software itself should be improved:

1. The multi-domain solver should use a sub-patch decomposition strategy that assigns processors to implicit and explicit sub-patches in such a way as to minimize idle times;

2. A scheduling algorithm should be developed to assign an implicit or explicit solver to a particular sub-patch on the fly, taking into account global and local CFL constraints to produce the most efficient computing time with the maximum possible time step;

3. An accelerated overset subiteration strategy should be developed: since, as pointed out in Section 3.4.1, the error between subiteration solutions and the exact solution is localized at the boundary and drops off exponentially from there, instead of performing a subiteration on the entire domain, it should be possible to restrict inter-iteration updates to a vicinity of the boundary only—which would essentially reduce the cost of each subiteration from a d to a $d-1$ dimensional problem, resulting in massive savings.

Finally, the software should be exercised to the fullest extent of its capacities, which in itself amounts to a significantly challenging endeavor.

Appendix A

Matrices for quasilinear-like Navier-Stokes formulation

Let $\mathbf{u} = (u, v, w)^{\mathrm{T}}$ be the velocity vector in Cartesian coordinates. For simplicity of presentation, let $a = \dfrac{1}{\mathrm{Re}}\dfrac{\mu'(T)}{\rho}$, $b = \dfrac{\gamma(\gamma-1)\mathrm{Ma}^2}{\mathrm{Re}}\dfrac{\mu(T)}{\rho}$, $c = \dfrac{\gamma}{\mathrm{RePr}}\dfrac{\kappa'(T)}{\rho}$, $d = \dfrac{1}{\gamma\mathrm{Ma}^2}$, and $e = \gamma - 1$. The coefficient matrices for the Navier-Stokes equations in quasilinear-like Cartesian form (2.2) are

$$M^x = \begin{pmatrix} u-\frac{2}{3}aT_x & -\frac{1}{2}aT_y & -\frac{1}{2}aT_z & d-a\left(u_x-\frac{1}{3}\nabla\cdot\mathbf{u}\right) & d\frac{T}{\rho} \\ \frac{1}{3}aT_y & u-\frac{1}{2}aT_x & 0 & -\frac{1}{2}a(v_x+u_y) & 0 \\ \frac{1}{3}aT_z & 0 & u-\frac{1}{2}aT_x & -\frac{1}{2}a(w_x+u_z) & 0 \\ eT-b\left(2u_x-\frac{2}{3}\nabla\cdot\mathbf{u}\right) & -b(v_x+u_y) & -b(w_x+u_z) & u-cT_x & 0 \\ \rho & 0 & 0 & 0 & u \end{pmatrix},$$

$$M^y = \begin{pmatrix} v-\frac{1}{2}aT_y & \frac{1}{3}aT_x & 0 & -\frac{1}{2}a(v_x+u_y) & 0 \\ -\frac{1}{2}aT_x & v-\frac{2}{3}aT_y & -\frac{1}{2}aT_z & d-a\left(v_y-\frac{1}{3}\nabla\cdot\mathbf{u}\right) & d\frac{T}{\rho} \\ 0 & \frac{1}{3}aT_z & v-\frac{1}{2}aT_y & -\frac{1}{2}a(w_y+v_z) & 0 \\ -b(v_x+u_y) & eT-b\left(2v_y-\frac{2}{3}\nabla\cdot\mathbf{u}\right) & -b(w_y+v_z) & v-cT_y & 0 \\ 0 & \rho & 0 & 0 & v \end{pmatrix},$$

$$M^z = \begin{pmatrix} w-\frac{1}{2}aT_z & 0 & \frac{1}{3}aT_x & -\frac{1}{2}a(w_x+u_z) & 0 \\ 0 & w-\frac{1}{2}aT_z & \frac{1}{3}aT_y & -\frac{1}{2}a(w_y+v_z) & 0 \\ -\frac{1}{2}aT_x & -\frac{1}{2}aT_y & w-\frac{2}{3}aT_z & d-a\left(w_z-\frac{1}{3}\nabla\cdot\mathbf{u}\right) & d\frac{T}{\rho} \\ -b(w_x+u_z) & -b(w_y+v_z) & eT-b\left(2w_z-\frac{2}{3}\nabla\cdot\mathbf{u}\right) & w-cT_z & 0 \\ 0 & 0 & \rho & 0 & w \end{pmatrix},$$

$$M^{xx} = -\frac{1}{\mathrm{Re}\,\rho}\,\mathrm{diag}\left(\frac{4}{3}\mu,\ \mu,\ \mu,\ \frac{\gamma}{\mathrm{Pr}}\kappa,\ 0\right),$$

$$M^{yy} = -\frac{1}{\mathrm{Re}\,\rho}\,\mathrm{diag}\left(\mu,\ \frac{4}{3}\mu,\ \mu,\ \frac{\gamma}{\mathrm{Pr}}\kappa,\ 0\right),$$

$$M^{zz} = -\frac{1}{\mathrm{Re}\,\rho}\,\mathrm{diag}\left(\mu,\ \mu,\ \frac{4}{3}\mu,\ \frac{\gamma}{\mathrm{Pr}}\kappa,\ 0\right).$$

The matrices M^{xy}, M^{xz}, and M^{yz} are zero except for two elements each, which are

$$M^{xy}_{u,v} = M^{xy}_{v,u} = M^{xz}_{u,w} = M^{xz}_{w,u} = M^{yz}_{v,w} = M^{yz}_{w,v} = -\frac{1}{3}\frac{1}{\mathrm{Re}}\frac{\mu}{\rho}.$$

Using the above definitions and the metric terms ξ_x, ξ_y, etc. the coefficient matrices in general coordinates for use in (2.5) as computed by the chain rule are given by

$$\begin{aligned} M^{\xi,1} =& \xi_x M^x + \xi_y M^y + \xi_z M^z + \xi_{xx} M^{xx} + \xi_{yy} M^{yy} + \xi_{zz} M^{zz} \\ &+ \xi_{xy} M^{xy} + \xi_{xz} M^{xz} + \xi_{yz} M^{yz}, \end{aligned}$$

$$M^{\xi,2} = \xi_x^2 M^{xx} + \xi_y^2 M^{yy} + \xi_z^2 M^{zz} + \xi_x \xi_y M^{xy} + \xi_x \xi_z M^{xz} + \xi_y \xi_z M^{yz},$$

and $M^{\eta,1}$, $M^{\eta,2}$ (resp. $M^{\zeta,1}$, $M^{\zeta,2}$) are obtained by replacing ξ with η (resp. ζ) everywhere in the above two equations. The mixed-derivative matrices can be obtained

similarly and are given by

$$
\begin{aligned}
M^{\xi\eta} &= \xi_x\eta_x M^{xx} + \xi_y\eta_y M^{yy} + \xi_z\eta_z M^{zz} \\
&\quad + (\xi_x\eta_y + \eta_x\xi_y)M^{xy} + (\xi_x\eta_z + \eta_x\xi_z)M^{xz} + (\xi_y\eta_z + \eta_y\xi_z)M^{yz}, \\
M^{\xi\zeta} &= \xi_x\zeta_x M^{xx} + \xi_y\zeta_y M^{yy} + \xi_z\zeta_z M^{zz} \\
&\quad + (\xi_x\zeta_y + \zeta_x\xi_y)M^{xy} + (\xi_x\zeta_z + \zeta_x\xi_z)M^{xz} + (\xi_y\zeta_z + \zeta_y\xi_z)M^{yz}, \\
M^{\eta\zeta} &= \eta_x\zeta_x M^{xx} + \eta_y\zeta_y M^{yy} + \eta_z\zeta_z M^{zz} \\
&\quad + (\eta_x\zeta_y + \zeta_x\eta_y)M^{xy} + (\eta_x\zeta_z + \zeta_x\eta_z)M^{xz} + (\eta_y\zeta_z + \zeta_y\eta_z)M^{yz}.
\end{aligned}
$$

Appendix B

Notes on mesh generation

This appendix provides details regarding the algebraic mesh generation of the curvilinear patches used in the examples of flow past a cylinder and flow past a sphere (Sections 4.2.1 and 4.2.2 respectively).

The annular sections pictured in Figure 4.9 were generated by means of the formulae

$$x(\xi,\eta) = r(\xi,\eta)\cos\theta(\xi), \tag{B.1}$$

$$y(\xi,\eta) = r(\xi,\eta)\sin\theta(\xi), \tag{B.2}$$

$$r(\xi,\eta) = \frac{(1-\eta^3)r_{\min} + \eta^3 r_{\max} + \eta(1-\eta^2)r'_{\text{in}}}{1-\eta+\eta[\cos^p\theta(\xi)+\sin^p\theta(\xi)]^{1/p}}, \tag{B.3}$$

$$\theta(\xi) = (1-\xi)\theta_{\min} + \xi\theta_{\max}, \tag{B.4}$$

with $\xi,\eta \in [0,1]$ and domain parameters $r_{\min}$, $r_{\max}$, r'_{in}, $\theta_{\min}$, $\theta_{\max}$ and p. The denominator of r produces a "rounded square" outer boundary for $p > 2$: larger values of p result in sharper corners ($p = 2$ produces a circular annular section). The annular sections in Figure 4.9 use $p = 8$. The numerator of r controls the radial extents of the domain as well as ensuring that $\partial_\eta r|_{\eta=0} = r'_{\text{in}}$. The parameter r'_{in} and number of discretization points in the η direction can then be chosen to produce the desired radial spacing near the surface of the cylinder and at the outer edge of the annular sections.

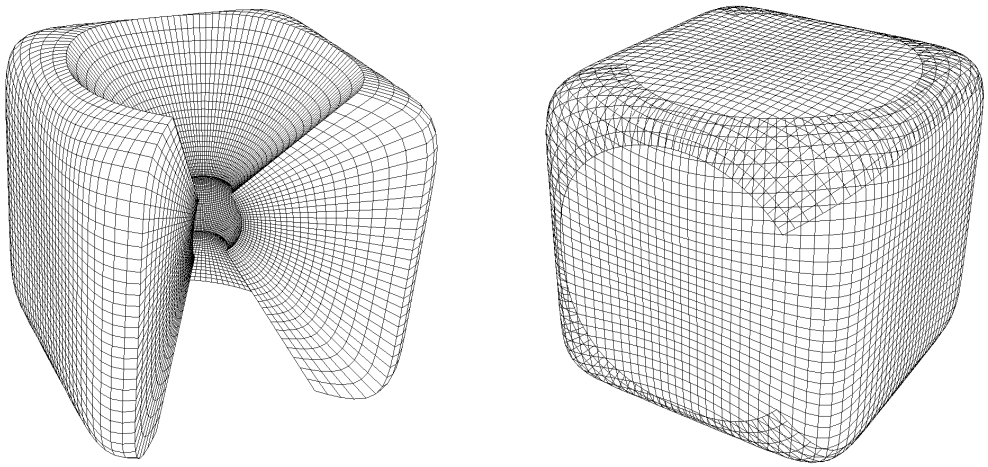


Figure B.1: Left: The "Yin" mesh with coarser grid spacing than used in the numerical examples of flow past a sphere. Right: The composite "Yin-Yang" mesh.

The curvilinear patches in the example of flow past a sphere are based on the so-called "Yin-Yang" mesh [51], which provides a (volumetric) covering of the sphere with two logically rectangular patches. As in the annular mesh above, the original formulae are modified to produce a "rounded cube" appearance and allow for variable radial mesh spacing. In detail, the formula for the Yin mesh is given by

$$x(\xi,\eta,\zeta) = r(\xi,\eta,\zeta)\cos\theta(\xi)\sin\phi(\eta), \tag{B.5}$$

$$y(\xi,\eta,\zeta) = r(\xi,\eta,\zeta)\sin\theta(\xi)\sin\phi(\eta), \tag{B.6}$$

$$z(\xi,\eta,\zeta) = r(\xi,\eta,\zeta)\cos\phi(\eta), \tag{B.7}$$

$$r(\xi,\eta,\zeta) = \frac{(r_{\max}-r_{\min})f(\zeta)}{r_{\max}-f(\zeta)+(f(\zeta)-r_{\min})g(\xi,\eta)}, \tag{B.8}$$

$$g(\xi,\eta) = \left[(\cos^p\theta(\xi)+\sin^p\theta(\xi))\sin^p\phi(\eta)+\cos^p\phi(\eta)\right]^{1/p}, \tag{B.9}$$

$$f(\zeta) = (1-\zeta^2)r_{\min}+\zeta^2 r_{\max}+\zeta(1-\zeta)r'_{\text{in}}, \tag{B.10}$$

$$\theta(\xi) = (1-\xi)\theta_{\min}+\xi\theta_{\max}, \tag{B.11}$$

$$\phi(\eta) = (1-\eta)\phi_{\min}+\eta\phi_{\max}, \tag{B.12}$$

with $\xi,\eta \in [0,1]$ and domain parameters $r_{\min}$, $r_{\max}$, r'_{in}, $\theta_{\min}$, $\theta_{\max}$, $\phi_{\min}$, $\phi_{\max}$, and p

(which is set to $p = 8$). Again, r'_{in} and the number of ζ discretization points are chosen to produce the desired radial grid spacing at the surface of the sphere and at the outer edge of the mesh. The resulting mesh is shown in the left pane of Figure B.1, with much coarser grid spacing than used in the numerical examples (for clarity). The x, y, and z equations for the Yang mesh are given by

$$x(\xi, \eta, \zeta) = r(\xi, \eta, \zeta) \cos \phi(\eta), \tag{B.13}$$

$$y(\xi, \eta, \zeta) = r(\xi, \eta, \zeta) \sin \theta(\xi) \sin \phi(\eta), \tag{B.14}$$

$$z(\xi, \eta, \zeta) = r(\xi, \eta, \zeta) \cos \theta(\xi) \sin \phi(\eta). \tag{B.15}$$

The remaining equations are the same as for the Yin mesh. The right pane of Figure B.1 shows the composite overlapping mesh.

Bibliography

[1] Abarbanel, S., Gottlieb, D., and Carpenter, M. H. On the removal of boundary errors caused by Runge-Kutta integration of nonlinear partial differential equations. *SIAM Journal on Scientific Computing 17*, 3 (1996), 777–782.

[2] Albin, N., and Bruno, O. P. A spectral FC solver for the compressible Navier–Stokes equations in general domains I: Explicit time-stepping. *Journal of Computational Physics 230*, 16 (July 2011), 6248–6270.

[3] Amlani, F. *A new high-order Fourier continuation-based elasticity solver for complex three-dimensional geometries*. PhD, California Institute of Technology, 2014.

[4] Ascher, U. M., Ruuth, S. J., and Wetton, B. T. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis 32*, 3 (1995), 797–823.

[5] Ashcroft, G., and Zhang, X. Optimized prefactored compact schemes. *Journal of Computational Physics 190*, 2 (2003), 459–477.

[6] Beam, R. M., and Warming, R. An implicit factored scheme for the compressible Navier-Stokes equations. *AIAA journal 16*, 4 (1978), 393–402.

[7] Beam, R. M., and Warming, R. Alternating direction implicit methods for parabolic equations with a mixed derivative. *SIAM Journal on Scientific and Statistical Computing 1*, 1 (1980), 131–159.

[8] Beam, R. M., and Warming, R. F. An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *Journal of Computational Physics 22*, 1 (Sept. 1976), 87–110.

[9] Bijl, H., Carpenter, M. H., Vatsa, V. N., and Kennedy, C. A. Implicit time integration schemes for the unsteady compressible Navier–Stokes equations: laminar flow. *Journal of Computational Physics 179*, 1 (June 2002), 313–329.

[10] Boyd, J. P. *Chebyshev and Fourier spectral methods*. Courier Dover Publications, 2001.

[11] Briley, W. R., and McDonald, H. Solution of the three-dimensional compressible Navier-Stokes equations by an implicit technique. In *Proceedings of the Fourth International Conference on Numerical Methods in Fluid Dynamics* (1975), Springer, pp. 105–110.

[12] Briley, W. R., and McDonald, H. On the structure and use of linearized block implicit schemes. *Journal of Computational Physics 34*, 1 (Jan. 1980), 54–73.

[13] Brown, D. L., Henshaw, W. D., and Quinlan, D. J. Overture: Object-oriented tools for overset grid applications. *AIAA paper No. 99 9130* (1999).

[14] Bruno, O. P., Han, Y., and Pohlman, M. M. Accurate, high-order representation of complex three-dimensional surfaces via Fourier continuation analysis. *Journal of Computational Physics 227*, 2 (2007), 1094–1125.

[15] Bruno, O. P., and Jimenez, E. Higher-order linear-time unconditionally stable alternating direction implicit methods for nonlinear convection-diffusion partial differential equation systems. *Journal of Fluids Engineering 136*, 6 (Apr. 2014), 060904–060904.

[16] Bruno, O. P., and Lyon, M. High-order unconditionally stable FC-AD solvers for general smooth domains I. Basic elements. *Journal of Computational Physics 229*, 6 (Mar. 2010), 2009–2033.

[17] Bruno, O. P., and Prieto, A. Spatially dispersionless, unconditionally stable FC–AD solvers for variable-coefficient PDEs. *Journal of Scientific Computing 58*, 2 (2014), 331–366.

[18] Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. *Spectral methods: evolution to complex geometries and applications to fluid dynamics*. Springer Science & Business Media, June 2007.

[19] Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. *Spectral methods: fundamentals in single domains*. Springer Science & Business Media, Sept. 2007.

[20] Carpenter, M. H., Gottlieb, D., Abarbanel, S., and Don, W.-S. The theoretical accuracy of Runge-Kutta time discretizations for the initial boundary value problem: a study of the boundary error. *SIAM Journal on Scientific Computing 16*, 6 (1995), 1241–1252.

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

[22] Chorin, A. J. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation 22*, 104 (1968), 745–762.

[23] Courant, R., Friedrichs, K., and Lewy, H. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen 100*, 1 (1928), 32–74.

[24] Crank, J., and Nicolson, P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Mathematical Proceedings of the Cambridge Philosophical Society 43*, 01 (Jan. 1947), 50–67.

[25] Curtiss, C. F., and Hirschfelder, J. Integration of stiff equations. *Proceedings of the National Academy of Sciences of the United States of America 38*, 3 (1952), 235.

[26] Dahlquist, G. Convergence and stability in the numerical integration of ordinary differential equations. *Math. Scand 4*, 1 (1956), 33–53.

[27] Dahlquist, G. 33 years of numerical instability, Part I. *BIT Numerical Mathematics 25*, 1 (Mar. 1985), 188–204.

[28] Dahlquist, G. G. A special stability problem for linear multistep methods. *BIT Numerical Mathematics 3*, 1 (Mar. 1963), 27–43.

[29] Daniel, J. W., and Moore, R. E. *Computation and theory in ordinary differential equations*. Freeman, Dec. 1970.

[30] Douglas, Jr., J. On the Numerical Integration of u_xx + u_yy = u_t by Implicit Methods. *Journal of the Society for Industrial and Applied Mathematics 3*, 1 (Mar. 1955), 42–65.

[31] Douglas, Jr., J., and Gunn, J. E. Alternating direction methods for parabolic systems in M space variables. *J. ACM 9*, 4 (Oct. 1962), 450–456.

[32] Douglas Jr., J., and Gunn, J. E. A general formulation of alternating direction methods. *Numerische Mathematik 6*, 1 (Dec. 1964), 428–453.

[33] D'yakonov, E. G. Difference schemes with splitting operator for higherdimensional non-stationary problems. *USSR. Comput. Math. Math. Phys 2*, 4 (1962), 549–568.

[34] Ekaterinaris, J. A. Implicit, high-resolution, compact schemes for gas dynamics and aeroacoustics. *Journal of Computational Physics 156*, 2 (Dec. 1999), 272–299.

[35] Elling, T. J. *GPU-accelerated Fourier-continuation solvers and physically exact computational boundary conditions for wave scattering problems*. PhD, California Institute of Technology, 2013.

[36] Fairweather, G., and Mitchell, A. R. A new computational procedure for A.D.I. methods. *SIAM Journal on Numerical Analysis 4*, 2 (June 1967), 163–170.

[37] Forti, D., and Dedè, L. Semi–implicit BDF time discretization of the Navier–Stokes equations with VMS–LES modeling in a high performance computing framework. *Computers & Fluids* (2015).

[38] Gaitonde, D. V., and Visbal, M. R. High-order schemes for Navier-Stokes equations: algorithm and implementation into FDL3di. Tech. rep., DTIC Document, 1998.

[39] Gaitonde, D. V., and Visbal, M. R. Pade-type higher-order boundary filters for the Navier-Stokes equations. *AIAA Journal 38*, 11 (2000), 2103–2112.

[40] Gander, M. J., and others. Schwarz methods over the course of time. *Electronic Transactions on Numerical Analysis 31*, 228-255 (2008), 5.

[41] Gordnier, R. E. High fidelity computational simulation of a membrane wing airfoil. *Journal of Fluids and Structures 25*, 5 (July 2009), 897–917.

[42] Gordon, W. J., and Hall, C. A. Construction of curvilinear co-ordinate systems and applications to mesh generation. *International Journal for Numerical Methods in Engineering 7*, 4 (1973), 461–477.

[43] Gottlieb, D., and Hesthaven, J. S. Spectral methods for hyperbolic problems. *Journal of Computational and Applied Mathematics 128*, 1 (2001), 83–131.

[44] Gottlieb, D., and Shu, C.-W. On the Gibbs phenomenon and its resolution. *SIAM Review 39*, 4 (1997), 644–668.

[45] Gustafsson, B., Kreiss, H., and Oliger, J. *Time-dependent problems and difference methods*. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley, 2013.

[46] Hairer, E., and Wanner, G. *Solving ordinary differential equations II. Stiff and differential-algebraic problems*. Springer Series in Computational Mathematics, 2005.

[47] Hesthaven, J. S., Gottlieb, S., and Gottlieb, D. *Spectral methods for time-dependent problems*, vol. 21. Cambridge University Press, 2007.

[48] Hoffmann, K. A., and Chiang, S. T. Computational fluid dynamics, vol. 2. *Engineering Education System, Wichita, Kansas* (2000), 21–46.

[49] Howarth, L. Concerning the effect of compressibility on laminar boundary layers and their separation. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 194*, 1036 (July 1948), 16–42.

[50] Johnson, T. A., and Patel, V. C. Flow past a sphere up to a Reynolds number of 300. *Journal of Fluid Mechanics 378* (Jan. 1999), 19–70.

[51] Kageyama, A., and Sato, T. "Yin-Yang grid": An overset grid in spherical geometry. *Geochemistry, Geophysics, Geosystems 5*, 9 (Sept. 2004), Q09005.

[52] Kao, K.-H., and Chow, C.-Y. Linear stability of compressible Taylor–Couette flow. *Physics of Fluids A: Fluid Dynamics (1989-1993) 4*, 5 (May 1992), 984–996.

[53] Karaa, S. A hybrid Padé ADI scheme of higher-order for convection–diffusion problems. *International Journal for Numerical Methods in Fluids 64*, 5 (2010), 532–548.

[54] Kawai, S., and Lele, S. K. Large-eddy simulation of jet mixing in supersonic crossflows. *AIAA Journal 48*, 9 (2010), 2063–2083.

[55] Kim, D., and Choi, H. Laminar flow past a sphere rotating in the streamwise direction. *Journal of Fluid Mechanics 461* (June 2002), 365–386.

[56] Kiya, M., Ishikawa, H., and Sakamoto, H. Near-wake instabilities and vortex structures of three-dimensional bluff bodies: a review. *Journal of Wind Engineering and Industrial Aerodynamics 89*, 14 (2001), 1219–1232.

[57] Kopriva, D. A. *Implementing spectral methods for partial differential equations: Algorithms for scientists and engineers*. Springer Science & Business Media, 2009.

[58] Lambert, J. D. *Numerical methods for ordinary differential systems: the initial value problem*. John Wiley & Sons, Inc., 1991.

[59] Lax, P. D., and Richtmyer, R. D. Survey of the stability of linear finite difference equations. *Communications on Pure and Applied Mathematics 9*, 2 (1956), 267–293.

[60] Lele, S. K. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics 103*, 1 (1992), 16–42.

[61] LeVeque, R. *Finite difference methods for ordinary and partial differential equations*. Society for Industrial and Applied Mathematics, Jan. 2007.

[62] Leveque, R. J. Intermediate boundary conditions for LOD, ADI and approximate factorization methods. Technical Report NASA-CR-172591, ICASE-85-21, NASA, Mar. 1985.

[63] Lions, P.-L. On the Schwarz alternating method. I. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations* (1988), Paris, France, pp. 1–42.

[64] Lyon, M., and Bruno, O. P. High-order unconditionally stable FC-AD solvers for general smooth domains II. Elliptic, parabolic and hyperbolic PDEs; theoretical considerations. *Journal of Computational Physics 229*, 9 (May 2010), 3358–3381.

[65] Manela, A., and Frankel, I. On the compressible Taylor–Couette problem. *Journal of Fluid Mechanics 588* (2007), 59–74.

[66] Marques, F., and Lopez, J. M. Onset of three-dimensional unsteady states in small-aspect-ratio Taylor–Couette flow. *Journal of Fluid Mechanics 561* (2006), 255–277.

[67] Mattsson, K., Svärd, M., and Shoeybi, M. Stable and accurate schemes for the compressible Navier–Stokes equations. *Journal of Computational Physics 227*, 4 (Feb. 2008), 2293–2316.

[68] Nordström, J., Gong, J., Van der Weide, E., and Svärd, M. A stable and conservative high order multi-block method for the compressible Navier–Stokes equations. *Journal of Computational Physics 228*, 24 (2009), 9020–9035.

[69] Peaceman, D. W., and Rachford, Jr., H. H. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics 3*, 1 (Mar. 1955), 28–41.

[70] Pearson, C. E. A computational method for viscous flow problems. *Journal of Fluid Mechanics 21*, 04 (Apr. 1965), 611–622.

[71] Pfister, G., Schmidt, H., Cliffe, K. A., and Mullin, T. Bifurcation phenomena in Taylor-Couette flow in a very short annulus. *Journal of Fluid Mechanics 191* (1988), 1–18.

[72] Rizzetta, D. P., Visbal, M. R., and Morgan, P. E. A high-order compact finite-difference scheme for large-eddy simulation of active flow control. *Progress in Aerospace Sciences 44*, 6 (Aug. 2008), 397–426.

[73] Saad, Y., and Schultz, M. H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing 7*, 3 (1986), 856–869.

[74] Sakamoto, H., and Haniu, H. A study on vortex shedding from spheres in a uniform flow. *Journal of Fluids Engineering 112*, 4 (1990), 386–392.

[75] Schwarz, H. A. Ueber einige Abbildungsaufgaben. *Journal für die reine und angewandte Mathematik 70* (1869), 105–120.

[76] Segal, I. A. Finite element methods for the incompressible Navier-Stokes equations. *Delft University of Technology* (2015).

[77] Shih, T. I., Smith, G. E., Springer, G. S., and Rimon, Y. Boundary conditions for the solution of compressible Navier-Stokes equations by an implicit factored method. *Journal of Computational Physics 52*, 1 (1983), 54–79.

[78] Sone, Y. *Kinetic theory and fluid dynamics.* Modeling and Simulation in Science, Engineering and Technology. Birkhäuser Boston, 2012.

[79] Starius, G. Composite mesh difference methods for elliptic boundary value problems. *Numerische Mathematik 28*, 2 (June 1977), 243–258.

[80] Steger, J. L., Dougherty, F. C., and Benek, J. A. A chimera grid scheme. *Advances in Grid Generation* (June 1983).

[81] Strikwerda, J. C. *Finite difference schemes and partial differential equations*. SIAM, 2004.

[82] Svärd, M., Carpenter, M. H., and Nordström, J. A stable high-order finite difference scheme for the compressible Navier–Stokes equations, far-field boundary conditions. *Journal of Computational Physics 225*, 1 (2007), 1020–1038.

[83] Svärd, M., and Nordström, J. A stable high-order finite difference scheme for the compressible Navier–Stokes equations: No-slip wall boundary conditions. *Journal of Computational Physics 227*, 10 (May 2008), 4805–4824.

[84] Tam, C. K. W., and Kurbatskii, K. A. Multi-size-mesh multi-time-step dispersion-relation-preserving scheme for multiple-scales aeroacoustics problems. *International Journal of Computational Fluid Dynamics 17*, 2 (2003), 119–132.

[85] Tam, C. K. W., and Webb, J. C. Dispersion-relation-preserving finite difference schemes for computational acoustics. *Journal of Computational Physics 107*, 2 (Aug. 1993), 262–281.

[86] Thomas, P. D., and Neier, K. L. Navier-Stokes simulation of three-dimensional hypersonic equilibrium flows with ablation. *Journal of Spacecraft and Rockets 27*, 2 (1990), 143–149.

[87] Trefethen, L. N. Pseudospectra of linear operators. *SIAM review 39*, 3 (1997), 383–406.

[88] Trefethen, L. N., and Embree, M. *Spectra and pseudospectra: the behavior of nonnormal matrices and operators*. Princeton University Press, 2005.

[89] Uzun, A., and Hussaini, M. Y. Simulation of noise generation in the near-nozzle region of a chevron nozzle jet. *AIAA Journal 47*, 8 (2009), 1793–1810.

[90] Visbal, M. R., and Gaitonde, D. V. High-order-accurate methods for complex unsteady subsonic flows. *AIAA Journal 37*, 10 (1999), 1231–1239.

[91] Visbal, M. R., and Gaitonde, D. V. On the use of higher-order finite-difference schemes on curvilinear and deforming meshes. *Journal of Computational Physics 181*, 1 (Sept. 2002), 155–185.

[92] Volkov, E. A. The method of composite meshes for finite and infinite regions with piecewise smooth boundary. *Trudy Matematicheskogo Instituta im. VA Steklova 96* (1968), 117–148.

[93] Wanner, G., Hairer, E., and Nørsett, S. P. Order stars and stability theorems. *BIT Numerical Mathematics 18*, 4 (Dec. 1978), 475–489.

[94] Warming, R. F., and Beam, R. M. An extension of A-stability to alternating direction implicit methods. *BIT Numerical Mathematics 19*, 3 (Sept. 1979), 395–417.

[95] White, F. M., and Corfield, I. *Viscous fluid flow*, vol. 3. McGraw-Hill New York, 2006.

[96] Yanenko, N. N. *The method of fractional steps*. Springer, 1971.