High-order unconditionally-stable FC-AD PDE solvers for general domains

Thesis by

Mark Lyon

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

2009
(Defended July 29, 2008)

To

Jen, Matthew, Sarah, and Alyssa

Acknowledgements

I am extremely grateful for the time I have been allowed to spend at the California Institute of Technology. In particular I am deeply indebted to my advisor, Prof. Oscar P. Bruno. I greatly benefited from his mathematical insight and rigor, his guidance in all aspect of my work and career, and, last but not least, for his patience. I would like to thank the members of my committee, namely, Prof. Niles Pierce, Prof. Dale Pullin, and Prof. Peter Schröder, for their insights and suggestions on this work. I am also grateful for the support I have received for my research from the Applied and Computational Mathematics option at Caltech, Prof. Bruno, the National Science Foundation, the Department of Defense, and the ARCS foundation. My wife has stood by me for the last five years through the ups and the downs and I would not have made it to this point without her support. I would like to thank my children as well. They have given stability to my life and brought me joy even on my worst days.

Abstract

A new methodology is introduced for the numerical solution of Partial Differential Equations in general spatial domains. The methodology is based on the use of the well-known Alternating Direction Implicit (ADI) approach of Peaceman and Rachford in conjunction with one-dimensional and high-order accurate Fourier representations of non-periodic data, obtained by way of a certain "continuation method" introduced recently for the resolution of the Gibbs phenomenon. We construct a number of high-order convergent PDE solvers on the basis of this strategy. Unlike previous alternating direction methods for general domains of order higher than one, the new algorithms possess the desirable property of unconditional stability for general spatial domains; the computational time required for these methods to advance one time-step, in turn, grows in an essentially linear manner with the number of spatial discretization points. In particular, the new methodology yields significant advantages over traditional low-order methods for computations involving wave propagation and "large domains," as well as PDEs including diffusive terms. In all, we treat Dirichlet problems for the Heat Equation, the Poisson Equation, and the Wave Equation in two- and three-dimensional spatial domains with smooth boundaries. A stability analysis we present hinges upon the numerical evaluation of certain singular value decompositions. Numerical results arising from the implementations of two- and three-dimensional versions of the method for linear parabolic, hyperbolic and elliptic PDEs, exhibit unconditional stability and high-order convergence, in agreement with our theoretical results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is concerned with the numerical solution of Partial Differential Equations (PDEs) of Parabolic, Hyperbolic and Elliptic type, and considers algorithms related to the well-known Alternating Direction Implicit approach introduced in reference [105] and further developed in [44, 45, 48, 49]. Such algorithms, which address each one of the dimensions of the spatial domains of the PDE sequentially, have been aggressively pursued over the last half century. They have been demonstrated to yield *unconditional stability* at approximately the same cost per time step as explicit (conditionally stable) finite difference formulations. The application of alternating direction algorithms has been hindered by a significant limitation however; previous algorithms could not be directly applied to arbitrary domains without reducing the truncation error near the boundary to first order [97]. The few applications of high-order alternating direction methods to non-rectangular geometries have relied upon prohibitive domain mappings (e.g., [54, 85, 92]) to effectively map the problem into a rectangular geometry. To the author's knowledge, prior to this work, unconditionally stable high-order alternating direction algorithms for general domains, without some form of domain mapping, have not been produced. In this thesis a new, simple, Fourier Continuation based Alternating Direction (FC-AD) algorithms is presented that can produce high-order accurate and unconditionally stable results for general geometries with limited computational cost—on the order of a Fast Fourier Transform over the space-discretization per time-step.

Over the more than 50 years since Peaceman and Rachford introduced their Alternating Direction Implicit method for the Heat Equation and Laplace's Equation, many variations have been introduced (c.f., [123] and references therein), including methods for solving other differential equations (e.g., [7, 12, 87, 96, 101, 142]) and nonlinear equations (e.g., [78, 81, 136, 138]), additionally improving both spatial and temporal accuracy (e.g., [60, 65, 94, 128, 139]). The alternating direction technique has been studied in a wide range of applications in nearly all areas of science and engineering, and frequently appears in textbooks concerning numerical methods. Although other types of operator splittings, including partial splittings, have also been introduced under the name of ADI (e.g., [13, 69]), the algorithms of interest in the present context are those that split differential operators spatially, thus reducing the solution of a PDE to the solution of sequences of ODEs.

The treatment of boundaries and boundary conditions in high-order finite difference schemes for PDEs entails significant difficulties—even for the simple rectangular domains, if the solution is not periodic. Recent advances in these regards include [3, 4, 35]; these methods allow for high-order on the basis of a "simultaneous approximation term" (SAT) approach for enforcement of the boundary conditions. The development of alternating direction schemes using high-order finite difference discretizations have thus far restricted attention to rectangular/parallelepiped domains (c.f., [94]), and even, in some cases, to situations in which the solution is periodic (c.f., [139]): the development of higher-order finite difference alternating direction schemes for rectangular geometries are currently an active research area (e.g., [39, 86, 128, 139]).

There has been some previous work in which, as in the present approach, a combination of a Fourier basis and ADI operator splitting is used [7, 55, 100, 141]. Like other ADI approaches, these previous techniques have also been restricted to rectangular geometries—in spite of a variety of efforts seeking generalization to general domains (see Section 4.4). Spline collocation methods have also been introduced in this context (c.f., [16, 43] and references therein.) In particular, Cubic-spline interpolation was used with an alternating direction algorithm in an embedding scheme to solve elliptic PDEs for complex geometries (c.f., [39] and references therein). Only conditional stability was achieved by this method and, further, the authors point to certain geometric issues (for instance, inability to solve on the region in between two squares) that remain unresolved.

Chebyshev methods [30, 141] and any other methods that rely on use of unevenly spaced discretizations do not provide a consistent basis for alternating direction splittings on complex geometries—unless domain mappings into suitable rectangular domains are used; clearly, however, such domain mappings are generally prohibitively complex. Spectral element methods and related techniques do simplify the required mappings [33]; yet construction of adequate three-dimensional finite element meshes, existence of restrictive time-step (CFL) conditions (especially for higher-order spectral methods), and the appearance of considerable pollution error for the lower-order spectral methods often considered, remain significant issues under investigation in this area.

A unique characteristic of a Fourier basis, utilized by the present approach, is that it diagonalizes constant coefficient differential operators. Use of Fourier bases for the approximation of non-periodic functions requires resolution of a classical problem in numerical analysis, namely: the Gibbs phenomenon. A variety of methods have been introduced to reduce or eliminate the Gibbs phenomenon. The problem of continuation is discussed in Section 3.5; additional information about these methods can be found in [30, 68, 73] and the references therein. Briefly, the present approach uses a form of the "continuation methods" (e.g., [22, 27, 28]) which evaluates a periodic extension of a smooth function; the overall continuation approach and the particular continuation strategy we have introduced for use in conjunction with the alternating direction algorithm, which differs significantly from previous continuation approaches, is presented in detail in Sections 3.5.1 and 5.1. The bulk of the computational cost for the Fourier continuation used in this thesis lies in the application of the Fast Fourier Transform (FFT) (see [41, 108]): the continuation Fourier series we use can be evaluated very efficiently.

Historically, the primary challenge in extending alternating direction methods to general geometries has been the lack of stability: it has been shown that the classical ADI convergence estimates [49, 105] only hold for simple rectangular geometries (see [18, 97]). Although the analysis of alternating direction methods has continued to develop (e.g., [5, 77, 81, 91, 114]), in particular to tensor product methods (e.g., [54, 92]), there is no complete theory for the ADI on general domains. The lack of unconditionally stable algorithms for complex geometries has heretofore severely limited the use of alternating direction methods in practical applications.

In this thesis a class of unconditionally stable high-order accurate alternating direction methods for general domains is introduced. For a given domain, the algorithm uses a Cartesian grid as shown in Figure 1.1. The PDE is discretized in time and then split into uncoupled ODEs by Alternating Direction techniques. Each one of the resulting separate ODEs is then solved with high-order accuracy over the resulting 1D grids that are evenly spaced for all interior points $x_j, j = 1, \ldots, n$ with two boundary points, x_ℓ and x_r, spaced

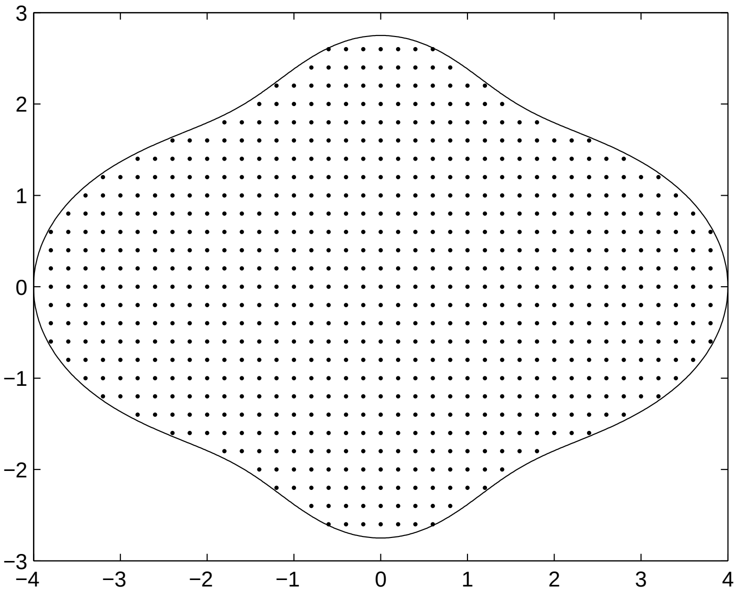


Figure 1.1: Evenly spaced discretization for a non-rectangular domain that provides lines of data in both directions.

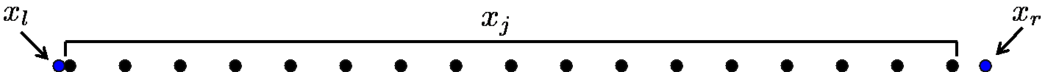


Figure 1.2: Sample discretization mesh for the ODEs resulting from an alternating direction splitting of a PDE.

such that for $h = x_{j+1} - x_j$ and therefore $x_1 - x_\ell \leq h$ and $x_r - x_n \leq h$ (see Figure 1.2). The computational cost required for solution of the resulting ODEs, which is based on our new Fourier Continuation approach, is proportional to that of one dimensional FFTs. The resulting unconditionally stable FC-AD methodology is fast, accurate, and memory efficient.

1.1 Overview of Chapters

The next three chapters present a brief overview of existing methods for PDE solution as well as basic concepts relating to pseudo-spectral methods and ADI techniques, the main thesis work and conclusions are then presented in Chapters 5 to 11. Chapter 2 provides a brief overview of the main characteristics of finite-difference and finite-element methods with an emphasis on certain and some specific challenges they present which are successfully addressed by means of the FC-AD methodology introduced in this thesis. In Chapter 3 we discuss pseudo-spectral methods and we consider Chebyshev polynomials as well as methods for the resolution of Gibbs phenomenon (see Sections 3.3 and 3.5 respectively). Chapter 4 contains a discussion of the continued development and some limitations of previous Alternating Direction implicit methods. In particular, related work that combines spectral methods with directional splitting will be discussed in detail in Section 4.4. In Chapter 5, our new Fourier-polynomial continuation method, FC(Gram) is introduced, on the basis of the Gram polynomials and the continuation methods of Section 3.5.1. Theoretical and numerical results are presented in this chapter demonstrating the high-order accuracy of the FFT-fast FC(Gram) algorithm. In Chapter 6, the FC-AD methodology is introduced through consideration of an application to the Heat equation. Numerical results are also presented in this chapter, demonstrating both the speed and accuracy of the FC-AD approach. Chapters 7 and 8 extend the FC-AD methodology to the Poisson Equation and Wave Equation respectively. Chapter 9 discusses the particular benefits of the FC-AD approach, including the effective elimination of both the so called "pollution error" and restrictive CFL conditions; this chapter also compares the accuracy, stability, and computational cost of the FC-AD to those arising from other available techniques. A theoretical discussion of the stability of the FC-AD algorithm is presented in Chapter 10. The thesis is concluded in Chapter 11.

Chapter 2

Background: Numerical Solution of PDEs

Most algorithms for the numerical solution of PDEs fall into one of the three categories: either Finite-Difference Methods (FDM), Finite-Element Methods (FEM), or Pseudospectral Methods. In view of their special significance to the present work, Pseudospectral methods will be discussed in greater detail in Chapter 3. Both FEM and FDM can be used to solve a wide range of PDEs in a variety of applications; it is often the case that particular applications are dominated by either FEM or FDM approaches. For example, most of the work in Computational Aeroacoustics, is pursued by means of FDM (see [40]). In contrast, structural mechanics computations are typically based on FEM methods, owing in part to the inherent geometric versatility of finite-element methods. Finite-Difference Methods and Finite-Element Methods are each very broad classes of numerical techniques and, in particular, they both admit explicit and implicit implementations. In what follows, we give a very brief overview of FDM and FEM and discuss two significant challenges posed by methods which are addressed successfully by the algorithms introduced in this work.

2.1 Finite-Difference Methods

Finite-difference methods are obtained as derivatives in a PDE are approximated by difference quotients. The accuracy of finite-difference methods can be estimated by means of Taylor series approximations; we refer to the works [123] and [97] for thorough introductions to FDM and its application to PDEs of elliptic, parabolic, and hyperbolic types.

A significant benefit of the FDM method lies in its simplicity. Implementations of explicit FDM algorithms can be completely straightforward if one uses low-order accurate methods in rectangular domains. Difficulties arise in applying high-order FDM techniques even on rectangular domains for simple, but non-periodic PDEs. Stable methods have been devised to treat this difficulty by maintaining a discrete summation-by-parts formula although such methods result in some degradation of the order of accuracy toward the boundary [34, 102, 103, 120]. Alternate formulations have also been developed which rely on use of unevenly spaced discretization around boundaries to regain stability, allowing for high-order accuracy with minimal modifications to the mesh [79].

Unlike FEM, FDM must be modified significantly in order to account accurately for the presence of curved boundaries. Various methods for handling non-rectangular domains have been developed which primarily rely on domain or coordinate mappings. In particular, we note that curvilinear coordinate systems have been used to produce accuracies of very high-order in a variety of contributions, including [80, 117]. Of course, the construction of such domain mappings is prohibitively complex for wide ranges of engineering and scientific geometries. Combinations of low- and high-order finite-differences have been used as well (see [71] and references therein): in these approaches fine unstructured meshes (resembling those used in connection with FEM) are utilized around complex boundary regions, while high-order structured FD methods are used in interior regions.

While the finite-difference methodology has been advanced greatly through decades of research, some broad challenges have still not been addressed successfully. For wave propagation problems, which arise in both elliptic and hyperbolic PDE forms, the error typically accumulates in a linear manner with the number of waves involved in the solution. Therefore, in order to keep the compound error below a certain tolerance, it is often necessary to greatly increase the number of points in the discretization. For explicit FDM, further, the CFL stability condition limits the largest allowable time-step that the algorithm may take and remain stable [124]. These stability conditions can be highly restrictive, especially for domains containing curved boundaries. Unlike wave propagation problems, in the case of diffusive PDEs the error does not tend to compound; in this case, however, the stability conditions for explicit FDM are severe: they require at minimum that the time-step be proportional to the mesh spacing squared. Again, curved boundaries can significantly amplify this effect. Sections 2.3 and 2.4 provide detailed discussions of both the compounding of error for wave propagation (known as the “pollution error” in the FEM [11]) and time-step

CFL restrictions along with the expenses associated with implicit formulations frequently used to overcome this difficulties.

Recent advances in FDM techniques have led to several notable algorithms that are able to directly handle some simple curved geometries with high-order accuracy. While representing important progress, these methods still suffer from either significant time-step restrictions in the diffusive case or are prone to pollution errors otherwise. In particular, the simultaneous approximating term (SAT) methods of [34] have been applied to some non-rectangular domains. In [1], SAT methods are applied to fourth-order accuracy for the Heat Equation resulting in an explicit FDM approach. Similar to the current work, a Cartesian mesh is overlaid on the geometry; a corresponding implicit formulation has not been produced as yet. Similarly the SAT methods are applied to wave type equations in [2] and electrodynamics in [90]; naturally, these methods give rise to significant pollution errors, particulary in the work in [2] in which only second-order spatial accuracy was obtained.

A novel approach is taken in [89], where, once again, a Cartesian mesh is overlayed on the complex geometry, but in which, for evaluation of spatial derivatives certain interior mesh points that are too close to the boundary are ignored. It is shown for a few simple test cases that, in this technique, the stability is not further restricted by the irregular geometry for incompressible fluid flow. While this is a significant advance, as no implicit formulation is given, the method still requires time-steps proportional to the square of spatial mesh size.

Additionally, the work in [88] avoids the additional time-step restrictions imposed by a curved domain for the Wave Equations by using a certain integral equation formulation to enforce the boundary conditions. While some stable results were obtained, the method did not prove to be stable for all of the test cases considered, in [88].

To the author's knowledge current high-order FDM techniques for complex geometries are restricted to fourth-order spatial accuracy at most. These methods are thus subject to either severe stability restrictions or costly implicit implementations (for diffusion problems), or to pollution errors (for wave propagation/advection problems).

2.2 Finite-Element Methods

Finite-elements were introduced many years before a theoretical understanding of their convergence was developed [62]; the theoretical understanding of these methods has since increased dramatically and convergence has been estabilished for many important cases. We refer the readers to [9, 10, 121] for a discussion of significant contributions to the theoretical developments of FEM. Since its emergence in the later 1950s, FEM has grown at a fast pace resulting in hundreds of books on the topic and billions of dollars spent on software and computer time for FEM codes (see [63])—a trend that will likely continue well into the future.

A primary advantage of FEM over FDM and other methods is that it conforms well to applications in general domains. FEM methods can use unstructured meshes and therefore do not require significant modification for complex domains, though generation of meshes of good quality for a given arbitrary geometry is a significant problem in and of itself. Many of the geometric difficulties mentioned in the previous section for FDM algorithms do not arise in the FEM context. FEM algorithms have been broadly applied to a wide-range of fields including structural mechanics, fluid flow, and semi-conductor modeling [37]. As the methods have continued to develop, significant variations have been produced to improve performance for various applications, including the development of Discontinuous Finite Elements and Characteristic Finite Elements [37].

Another significant advantage of FEM over other traditional techniques is the relative ease with which adaptive mesh refinement may be performed. FEM element methods can also be constructed to high-order or spectral accuracy [33] with specific requirements; note, in particular, that for true high-order accuracy to result, the elements must approximate the domain boundary to high-order as well. This requirement can be best met by utilizing appropriate mappings at the level of the individual elements.

The FDM and FEM are related in some ways: they both suffer from pollution errors (see [11]) and conditional stability which can be "very restrictive, particularly for long time integration" (see [38]). The additional computational cost required by unconditionally stable implicit methods per time-step is significant, making the benefit of implicit FEM methods highly dependent on the details of a specific problem.

2.3 Pollution Error

Often the accuracy of a given PDE solution technique for wave propagation problems is stated in terms of a number of points required per wavelength (PPW) for the approximate

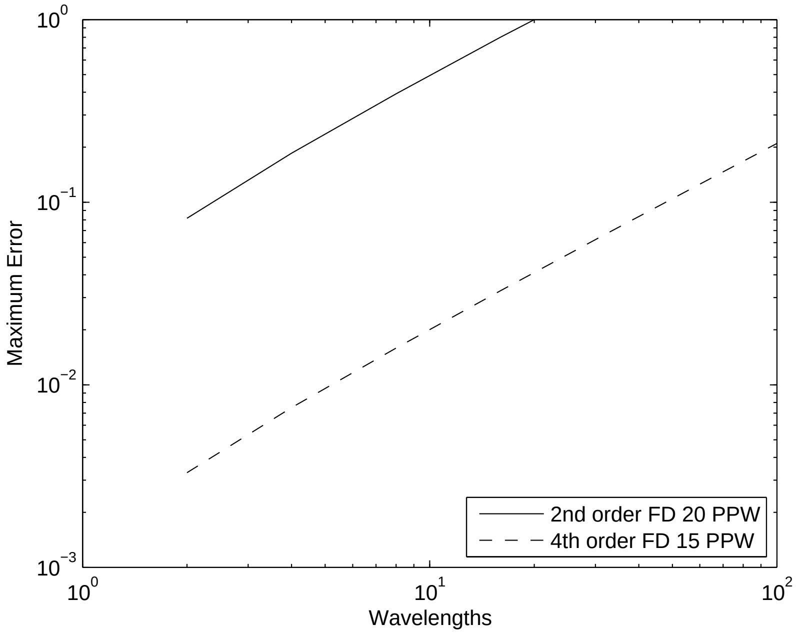


Figure 2.1: Maximum error arising in the solution of the Wave Equation up to a final time T=1, where w is the number of wavelengths in the domain. The PDE has been set up in such a way that the solution is given by $\sin(2\pi w(x-t))$. The error is shown as a function of w for both a second-order finite-difference method and a fourth-order finite-difference method. Sufficiently many time steps were taken to insure the error was due to the spatial discretization.

solution to be correct to within a given accuracy (see [40]). Due to the accumulation of error over multiple waves, finite-difference and finite-element methods typically require significant increases in the number of PPW, for large-scale problems. This issue has been discussed for FDM in [84] and in [11] for the closely related "pollution error" effect exhibited by finite-element methods.

For example, it is shown in [84] that, for a simple one-way Wave Equation and for a fixed number of PPW, the finite-difference error increases linearly with the total number of wavelengths across the domain. This is demonstrated in Figure 2.1 where the Wave equation was solved using explicit second- and fourth-order finite-difference methods. From Figure 2.1, we see that for a fourth-order Finite Difference scheme with 15 PPW, if the domain has 10 wavelengths then the maximum error at any time step is about 2%. If the number of wavelengths increases to 100, then the error is 20%. These results are consistent with those shown in [84].

In order to mitigate this pollution effect as much as possible, high-order methods should be used, as they reduce the number of additional PPW required to reach a given accuracy. Indeed, Pseudospectral methods exhibit essentially no pollution error due to the spectral accuracy of the approximation, but unfortunately are prohibitively complicated for complex domains, as will be discussed in the next chapter.

2.4 Time-Step Stability Restrictions

In general for explicit methods, the approximation at a future time step, at a given location, is dependent only upon values at neighboring grid locations. This locality allows for fast computations per time-step, but the maximum size of the time-step is severly limited by stability conditions. In practice the stability restriction is often most severe for parabolic equations like the Heat Equation where the maximum time-step is limited by the smallest mesh spacing squared. For complex geometries, either significant work must be done to optimize the mesh for the particular geometry, or the time-step restriction will be exceedingly severe (consider Figure 1.1 where the closest boundary point to and interior point may be arbitrarily small).

Implicit methods are used to reduce, and in many cases eliminate altogether, the stability restrictions on the time-step, but they require the solution of a linear system at each time step and therefore can give rise to significant computational costs. Implicit methods have been improved with the introduction of Conjugate Gradient (CG) and other Krylov subspace algorithms which can sometimes dramatically reduce the time required to solve the linear system (see [112]). In particular as the solution approaches a steady state, these iterative solvers may require very few iterations and be very effective indeed. The convergence can be prohibitively slow in other cases, and therefore considerable effort has also been devoted to the development of adequately preconditioned iterative methods; see [112]. In general the convergence of the CG algorithm depends on the conditioning of the system—which, as it happens, deteriorates rapidly as the size of the problem increases (see [36]). Indeed, our own numerical results demonstrate that for a sample problem switching from a first-order accurate in time explicit FDM for the Heat Equation to second-order accurate in time, implicit and unconditionally stable, FDM yielded an efficiency improvement of about a factor of three to achieve a 10% maximum error but actually decreased the efficiency when attempting to reach a 1% maximum error. The efficiency increase in the 10% case was obtained by modifying the parameters of the CG algorithm and the time-step to maximize efficiency for the given problem. More details of the calculation are shown in Section 9.2.

It should be noted that while the time-step stability restrictions are the most severe for parabolic problems, they can be significant for hyperbolic problems as well if either the individual mesh points are not optimized for the geometry or other significant modification to the basic algorithms are made. Assuming the geometry has been appropriately meshed, hyperbolic and elliptic problems are still prone to the pollution error, discussed in the previous section; we note that for parabolic problems, pollution errors do not arise. By considering sufficiently simple situations, a comparison of the FC-AD algorithms presented in this thesis to other techniques will be presented in Chapter 9. In particular, in that set of comparisons the main topics of the present and previous sections, stability restrictions and pollution error, are examined in detail.

Chapter 3

Background: Pseudospectral Methods and the Gibbs Phenomenon

Pseudospectral methods are a class of algorithms which use discrete data to construct approximate solutions for which the error decays spectrally fast as the discretizations are refined. Figure 3.1 displays typical convergence for a one-dimensional spectral approximation (upon which pseudospectral methods are based). A mere doubling of the number of discretization points yields a decrease in error of the approximation by six orders of magnitude. The present text will only attempt to give a very broad overview of the features of pseudospectral methods relevant to this thesis. Detailed information on these methods can be found in the references [21, 30, 64, 72].

3.1 Connection to Finite Difference Techniques

It is interesting to note that, according to [64] pseudospectral methods can be viewed as the limit of finite difference methods, as the stencil and the order of the approximation are maximized. This analogy is most easily appreciated when dealing with periodic solutions where each discrete point may be considered to have infinitely many neighboring points. The approximate value or approximate derivatives of a function could then be calculated using a stencil of effectively unlimited size and therefore an unlimited order of accuracy.

A more typical finite difference calculation would normally calculate the value of the function or a derivative using the values of the function at a small number of nearby points. This allows for efficient algorithms since the cost of calculating derivatives is on the order of

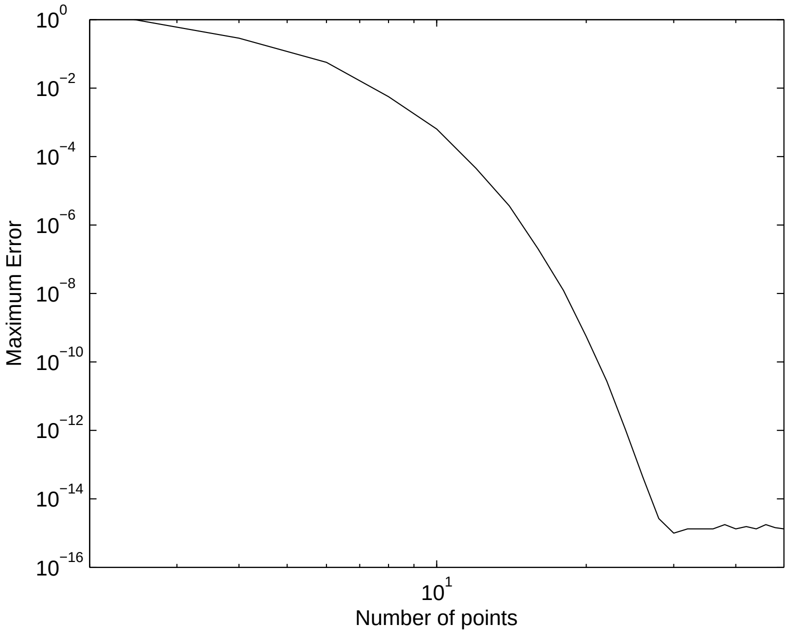


Figure 3.1: Maximum interpolation error resulting from a discrete Fourier transform for the function $e^{\sin(x)}$ as a function of the number of discretization points in one dimension.

the number of unknowns, resulting in a sparse and banded matrix, which can be significantly easier to invert than a full matrix—either by direct or iterative methods. For instance, in 1-D and over n discrete data points, a derivative can be calculated with finite differences at each of the n points in $O(mn)$ operations where m is the size of the stencil.

The corresponding cost for a pseudo-spectral method would nominally be $O(n^2)$ since the value of the function at each point depends on the value at every other point. This would be the cost of applying an unaccelerated discrete Fourier transform to the data. The FFT uses a recursion algorithm to reduce the computational cost to $O(n \log(n))$ ([41]). Fourier, Cosine, Sine, and Chebyshev transformations all benefit from the efficiencies of the FFT and therefore the computational costs for these techniques are nearly the same as lower-order finite-differences for an equivalent number of points. When applicable, pseudo-spectral methods have significantly higher accuracy, and thus require significantly smaller discretizations than finite difference or finite element methods to reach a given level of accuracy: spectral convergence gives rise to significantly more efficient algorithms.

3.2 Fourier Methods

If a function, $f(x)$, periodic on the unit interval, is known at the n evenly spaced points

$$x_j = \frac{2\pi j}{n}, \quad j = 0, \dots, n-1, \tag{3.1}$$

then one can approximate the function as Fourier series of the form

$$f(x) \approx \sum_{k \in t(n)} a_k e^{2\pi i k x}, \tag{3.2}$$

where $t(n) = \{k \in \mathbb{N} : -n/2 + 1 \leq k \leq n/2\}$ for n even and $t(n) = \{k \in \mathbb{N} : -(n-1)/2 \leq k \leq (n-1)/2\}$ for n odd. This particular form is chosen because of the discrete orthogonality property

$$\sum_{j=0}^{n-1} e^{2\pi i k x_j} e^{-2\pi i \tilde{k} x_j} = n \text{ if } k = \tilde{k}, \ 0 \text{ otherwise}, \tag{3.3}$$

which allows the coefficients a_k to be calculated efficiently by

$$a_k = \sum_{j=0}^{n-1} f(x_j) e^{-2\pi i k x_j}, \quad \forall k \in t(n). \tag{3.4}$$

It is precisely the calculations in Equations (3.2) and (3.4) that are accelerated by the FFT.

The exact error bounds for approximating a function by a discrete Fourier series depends on the smoothness of the function; these approximation properties are considered in detail in many texts on spectral methods (c.f., [30]); a standard result is

$$\|f(x) - \sum_{k \in t(n)} a_k e^{2\pi i k x}\|_{L^\infty(0,1)} \leq C \log(n) n^{-m} \|f^m(x)\|_{L^\infty(0,1)} \tag{3.5}$$

for a function $f(x)$ with bounded m^{th} derivative, $f^m(x)$, and with $f^q(x)$ continuous and periodic for $q < m$. In particular this result accounts for the error made by truncating Fourier series to n terms and the interpolation error arising from discrete sampling the function at n points (see [31]). If the function is not periodic and the discrete Fourier series is taken, the "Gibbs phenomenon" results: near discontinuities, oscillations develop and slow convergence is achieved as n grows. Figure 3.2 demonstrates this effect on the interpolation of $f(x) = x$. Alternatively, if the function has significant number of bounded

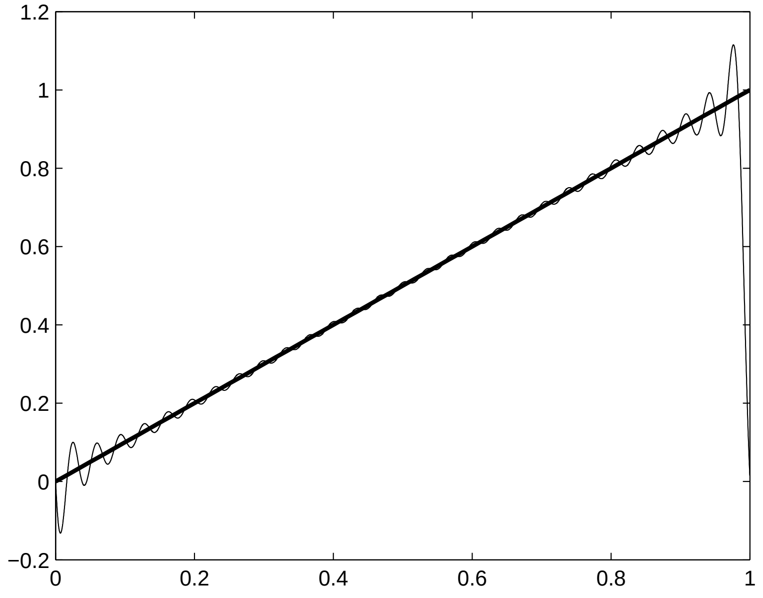


Figure 3.2: Demonstration of the Gibbs' phenomenon on the Fourier interpolation of $f(x) = x$ with 60 Fourier modes.

derivatives, the Chebyshev polynomials can be used to obtain an accurate approximation. Methods designed to alleviate the Gibbs phenomenon (i.e., represent a non-periodic function accurately using trigonometric polynomials) have also been researched for the last 30 plus years; a discussion of efforts in these regards is presented in Section 3.5. In its original form, the FFT is limited to the case of equally spaced data. But in the past two decades significant advances have been made in extending Fourier methods to unevenly spaced data (e.g., [15, 20, 51–53, 61]).

It has been noted that for high Reynolds number turbulent flow problem and for cases in which the solution is smooth and periodic, the numerical solution by means of a fourth-order finite-difference scheme would require "typically a factor of 10 longer in time and a factor of 20 larger in storage," relative to a Fourier spectral method and that for a second-order finite difference scheme, the difference would be three orders of magnitude (see [32]).

3.3 Chebyshev methods

The usefulness of the Chebyshev methods can be easily appreciated by considering its relationship to the Fourier methods. Consider a smooth but non-periodic function $f(x) \in C^k[-1,1]$ for some integer k. By making a change of variables $x = \cos(\theta)$, the function $f(\cos(\theta))$ is now a 2π even periodic function of the variable θ which can be expanded in a Cosine series. The Cosine series converges rapidly for k large enough since $f(\cos(\theta)) \in C^k[0, 2\pi]$. This transform then can be accomplished fast by a FFT or even slightly faster by specialized Discrete Cosine transform algorithms.

It can easily be shown that the resulting approximating functions $T_k(x) = \cos(k\theta)$ are in fact polynomials in the space of the variable x through the recurrence relation

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \tag{3.6}$$

which in itself is a very useful property of the Chebyshev polynomials. A related recurrence relation can be formulated for the derivatives of the polynomials, allowing for the fast computation of derivatives.

The most serious restriction on Chebyshev methods is that the point values at which the function $f(x)$ must be known are given by $x_j = \cos(\theta_j)$, where typically the θ_j are evenly spaced. While unevenly spaced FFT methods could be applied, a significant deviation from the required point spacing might severely decrease the accuracy of the approximation. In particular if the point values of x_j are evenly spaced as in Equation (3.1) then the approximation would suffer from the "Runge phenomenon" (see [109]): polynomial approximations to a smooth function over evenly spaced nodes may diverge as the number of interpolation points tends to infinity. It should be noted that the restriction to a particular point spacing is often not a problem for 1-D or even higher-dimensional problems on rectangular domains, but a significant limitation for general domains.

3.4 Difficulties for General Domains

Chebyshev and Fourier methods can be easily extended to higher dimensional problems, as long as the PDE domain is rectangular. Figure 3.3 displays a two-dimensional Chebyshev grid: given values of a smooth function f on such a grid, a fast Cosine transform can easily

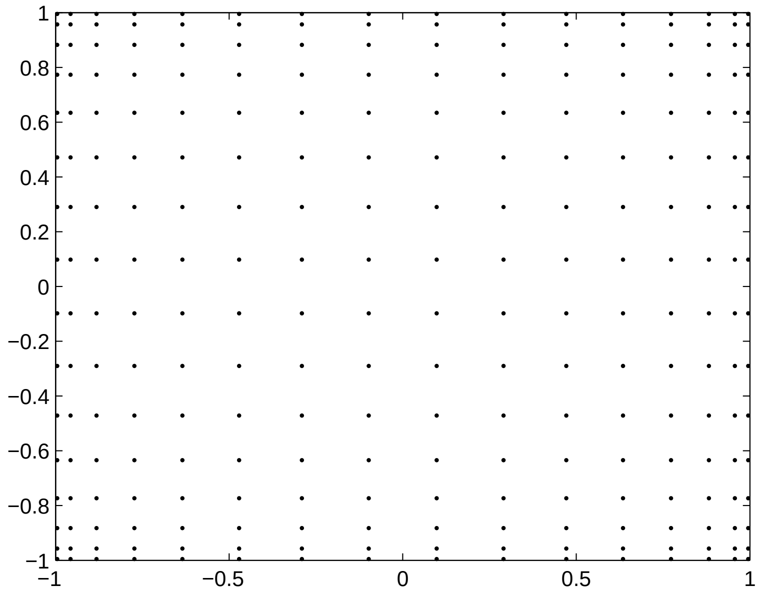


Figure 3.3: Discretization mesh associated with Chebyshev polynomials in a rectangular domain.

be applied in each one of the two dimensions, sequentially, to obtain, rapidly, spectrally accurate numerical approximations of the true Chebyshev coefficients of f. Clearly, for a general domain, including domains as simple as Figure 1.1, it is not possible to represent a function, with spectral accuracy, by means of iterated Chebyshev transforms.

While it is possible to generate a spectral expansion basis functions specifically designed for each geometries, the difficulties make it impractical for all but a few simple domains (e.g., circle, sphere, or annulus) [104]. An approach that is often mentioned, but used only rarely in practice, is based on the fact that complicated geometries can be mapped into simpler domains for which a spectral basis of interpolating functions is available. Even in the cases where such a mapping can be efficiently obtained, it can significantly complicate an otherwise simple PDE.

A significant area of current research interest is the “spectral-element methods” which combine pseudospectral methods with finite-element techniques. Such approaches are based on mapping individual elements to a simple geometry over which a high-order approximation may be made. While simplifying the mapping process, some of the challenges of finite-element methods are also introduced, including those discussed in Chapter 2. The interested reader is referred to [33] for more information about the accomplishments and challenges of these techniques.

3.5 Towards the Resolution of the Gibb's Phenomenon

A glance to Figure 1.1 suggests new ways to apply pseudospectral methods that become available if the Gibbs phenomenon is resolved, the rapid oscillations eliminated, and convergence of Fourier methods to non-periodic functions is adequately accelerated. In this section recent advances in this area are described.

The earlier contributions in this area include filtering of high-order Fourier coefficients [93], use of a specialized integration rule [95], and smoothing [76], all of which increase the accuracy away from the discontinuity but do not provide accuracy over the entire domain of the function. The first real advance that provided spectral accuracy across the domain was the development of the Gegenbauer polynomial methods [75] for which convergence can be proven under certain conditions. These methods reproject the oscillating Fourier functions into an alternate orthogonal basis which minimize the rapid Gibbs oscillations and overall slow convergence. The convergence of Gegenbauer polynomial method is slower than one would hope for and has numerous conditioning and other difficulties (e.g. [23]), some but not all of which have been subsequently addressed in (see [68, 73, 74, 127]); in particular, this approach requires very fine discretizations to produce even minimal accuracy and to reach the convergence regime—upon which, with subsequent mesh refinements fast convergence is obtained.

Other developments include several algorithms that require knowledge of the jumps in the values of the function and its derivatives at discontinuity points [55, 66, 67, 82]. These methods have been demonstrated to give rise to significantly improved convergence but the determination of jumps in derivatives remains a significantly challenging problem. As mentioned in the introduction, the methods [55, 82] have been applied to the solution of differential equations. In particular the contributions [6–8, 24–26, 130–132] present various methods built upon the technique [82] to solve a range of differential equations. For example, [131] extends earlier efforts to complex geometries by *domain mapping* and relies on conjugate gradient iterations to deal with the more complicated PDEs arising from the mapping. Building off of the techniques for singularity detection and subtraction in [56] and [55], the work [100] uses spectral methods with resolution of the Gibbs phenomenon on the basis of a singularity subtraction technique. The work was limited to rectangular domains and, in order to obtain high-order accuracy, a number of additional grid points were added near the domain boundaries—essentially for evaluation of jumps of the function and its derivatives.

3.5.1 Continuation Methods

One of the more recent efforts towards the resolution of Gibbs' phenomenon is considered by [22, 27, 28] and is related to earlier work on embedding partial differential equations, [57, 58], where a related procedure was proposed. These methods rely on the construction of a function which is periodic in a domain larger than the domain of interest. Unlike the traditional Fourier series, no orthogonality condition exists in this case—a fact that has profound consequences for both computation and analysis. Following [28], the continued function $f^c(x)$ is given by the series

$$f^c(x) = \sum_{k \in t(m)} a_k e^{\frac{2\pi i}{b} kx}, \tag{3.7}$$

where again $t(m) = \{k \in \mathbb{N} : -m/2 + 1 \leq k \leq m/2\}$ for m even and $t(m) = \{k \in \mathbb{N} : -(m-1)/2 \leq k \leq (m-1)/2\}$ for m odd and let x_j be a set of n discrete points in the range from 0 to 1. The solution can then be obtained (as in [22, 27, 28]) by solving the least squares system of equations

$$\min_{a_k \forall k \in t(m)} \sum_{j=0}^{n-1} \left| \sum_{k \in t(m)} a_k e^{\frac{2\pi i}{b} k x_j} - f(x_j) \right|^2 . \tag{3.8}$$

The method of solution that has yielded the best results is to solve for the coefficients a_k by using a regularized Singular-Value Decomposition (SVD). Although formally the system can be solved in $O(n^2)$ operations by either Vandermonde techniques (see [70]) or related recursive techniques of orthogonal polynomials over the unit circle, (see [29]), the performance of the SVD can be significantly more accurate for large numbers of points. Throughout this thesis, this particular method of Fourier approximation will be denoted as the FC(SVD).

The function $f^c(x)$ constructed in this manner has periodicity interval b and therefore

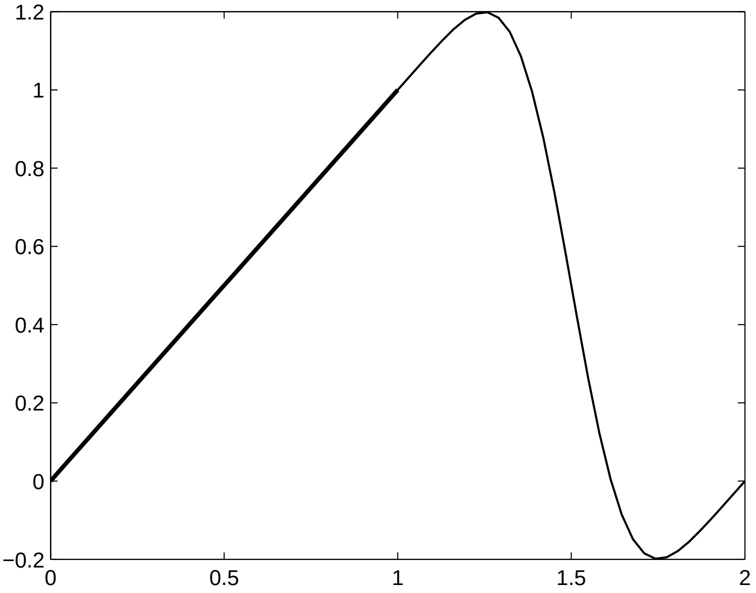


Figure 3.4: Smooth periodic function resulting from the continuation method applied to the function $f(x) = x$ over the unit interval.

lies in the space of smooth functions, $C^{\infty}[-\infty, \infty]$. In what follows we denote by $C^{\infty}_{per}[c, d]$ the space of infinitely differentiable functions defined over the real line with periodicity length $d - c$; with this notation we have $f^{c}(x) \in C^{\infty}_{per}[0, b]$.

Let us consider once again the function $f(x) = x$ defined on the unit interval. Applying the FC(SVD) approximation with $n = 32$, $m = 16$, and $b = 2$ yields Figure 3.4, which accurately approximates the function to more than six digits of accuracy with just 16 Fourier modes. In reference [28], accurate approximations were obtained using several values of b and different ratios of n and m. For m sufficiently small relative to n ($m = n/2$), spectral convergence was demonstrated to both the function and its derivatives.

The computational strengths of the continuation method include its ability to handle both evenly and unevenly spaced grids over complicated geometries in high dimensional space. Also, the continuation method loses very little accuracy due to the conditioning of the system compared to other Gibbs resolution techniques (see [50, 68, 73, 74, 127]). The continuation method will not be applied directly in this work because in this context it would lead to loss of computational efficiency. As was noted above, using an m length stencil, $O(mn)$ operation would be required to calculate the derivatives at all n points by finite-differences. In contrast, calculating the SVD for a fixed ratio of m to n, requires $O(n^3)$ operations. In Chapter 5 a variation of this Fourier continuation technique is developed that, for functions of a one-dimensional variable, yields high-order accuracy at operation counts equivalent to the FFT's $O(n\log(n))$. The application of the new technique to the solution of PDEs then requires a method to reduce the PDE to one or more differential equations in a single dimension. This is exactly what is accomplished by alternating direction methods.

Chapter 4

Background: Alternating Direction Methods

The Alternating Direction Implicit (ADI) algorithm was introduced in [105] and further developed in [44, 45, 48, 49] as well as hundreds of additional contributions. The basic idea of the method is to treat each spatial direction of the PDEs separately. A single direction is chosen and then the value of the solution at each point on a line in the specified direction is advanced in time by using only the values on the same line. The process is then repeated, but in an alternate direction. The basic algorithm is simple and computationally efficient, and thus extensions have been attempted to extend the technique to numerous PDE problems.

4.1 Basic Concepts and Computational Strengths

The ADI algorithm was developed initially for parabolic problems; following the initial contributions, we use the example of the Heat Equation to detail the main characteristics of this algorithm. Let us thus consider a simple homogenous Heat Equation over the unit cube with some initial condition and zero Dirichlet boundary conditions,

$$u_t = u_{xx} + u_{yy}, \quad u(x, y, 0) = u_0(x, y). \tag{4.1}$$

Let A_x be the matrix associated with a centered finite difference scheme over a discrete equi-spaced mesh ($n \times n$, $h = 1/n$) the second derivatives with respect to x. The matrix A_x is tridiagonal along each line of data for fixed y and, therefore, it can be inverted in $O(n^2)$ operations by means of LU decompositions (c.f., [110]). Analogously, let A_y be the corresponding matrix for evaluation of second derivatives in the y direction, which, similarly, can be efficiently inverted. As discussed in what follows, these matrices play central roles in the ADI algorithm. A discrete implicit method for stepping forward a discrete solution of Equation (4.1) by a time increment Δt is given by

$$\frac{u^{n+1}-u^n}{\Delta t} = (A_x + A_y)\frac{u^{n+1}+u^n}{2} + O(\Delta t^2) + O(h^2), \tag{4.2}$$

which can be simplified to

$$(I - \frac{\Delta t}{2}A_x + \frac{\Delta t}{2}A_y)u^{n+1} = (I + \frac{\Delta t}{2}A_x + \frac{\Delta t}{2}A_y)u^n + O(\Delta t^3) + O(\Delta t h^2). \tag{4.3}$$

While each matrix A_x and A_y can be easily inverted, the combined matrix $(I - \frac{\Delta t}{2}A_x + \frac{\Delta t}{2}A_y)$ cannot. Peaceman and Rachford (see [105]) noted that the factored product $(I - \frac{\Delta t}{2}A_x)(I - \frac{\Delta t}{2}A_y)$, which approximates $(I - \frac{\Delta t}{2}A_x + \frac{\Delta t}{2}A_y)$ to within an error of the order of Δt^2 *can be easily inverted*; using this approximation we obtain the iterative scheme,

$$u^{n+1} = (I - \frac{\Delta t}{2}A_x)^{-1}(I - \frac{\Delta t}{2}A_y)^{-1}(I + \frac{\Delta t}{2}A_x)(I + \frac{\Delta t}{2}A_y)u^n + \varepsilon(h, \Delta t), \tag{4.4}$$

where the error term, $\varepsilon(h, \Delta t)$, is given by

$$\varepsilon(h, \Delta t) = \frac{\Delta t^2}{4}A_x A_y(u^{n+1} - u^n) + O(\Delta t^3) + O(\Delta t h^2). \tag{4.5}$$

The additional error term, $\frac{\Delta t^2}{4}A_x A_y(u^{n+1} - u^n)$, is known as the splitting error; since $u^{n+1} - u^n = O(\Delta t)$, the splitting error is of order Δt^3 (see [46, 134] for ADI variations leading to further reductions of the splitting error) and we obtain,

$$\varepsilon(h, \Delta t) = O(\Delta t^3) + O(\Delta t h^2). \tag{4.6}$$

Over a rectangular domain, the operators $(I - \frac{\Delta t}{2}A_x)^{-1}$, $(I - \frac{\Delta t}{2}A_y)^{-1}$, $(I + \frac{\Delta t}{2}A_x)$, and $(I + \frac{\Delta t}{2}A_y)$ commute and a von Neumann stability analysis shows that the symmetric discrete operator combinations $(I - \frac{\Delta t}{2}A_x)^{-1}(I + \frac{\Delta t}{2}A_x)$ and $(I - \frac{\Delta t}{2}A_y)^{-1}(I + \frac{\Delta t}{2}A_y)$ have eigenvalues less than one and therefore produce a stable numerical algorithm. Coupling this stability with the bound for the local error from Equation (4.6) gives the convergence of the method. The stability analysis can be generalized to other PDEs as long as the one-dimensional operators commute. Unfortunately, however, this useful property only holds for a very restricted class of PDEs and most importantly, only for rectangular geometries (c.f., [18, 129, 140]). It has been demonstrated in explicit constructions that instabilities can occur in the noncommutative case (see [114]).

4.2 Iteration Parameters

The Laplace equation, over a rectangular domain, is an interesting example of a PDE to which the above theory can directly be applied. One can use the algorithm for the Heat Equation and advance the up to a point in which a steady state is reached: the result is, of course, a solution of the Laplace equation. Instead of taking a constant "time-step", iteration parameters, γ_j, are used which change with each step, so that at the j^{th} step the next approximation is calculated by

$$u^{j+1} = (I - \frac{\gamma_j}{2}A_x)^{-1}(I - \frac{\gamma_j}{2}A_y)^{-1}(I + \frac{\gamma_j}{2}A_x)(I + \frac{\gamma_j}{2}A_y)u^j. \tag{4.7}$$

It can be shown [118] that if the matrices A_x and A_y each has n common linearly independent eigenvectors, then in at most n iterations the solution of the discrete problem can be found. Thus this algorithm requires a total of $O(n^3)$ operations, which can be compared with $O(n^4)$ operations required by the fastest direct method. Thus, the ADI is also considered an iterative method for the solution of a linear system (c.f., [113]). In practice, a set of iteration parameters can be chosen for more general problems that are in some sense optimal (c.f., [133, 140]).

4.3 Extensions of the Method

There have been several extensions of the original theory for ADI and related techniques (e.g., [5, 77, 78, 87, 91]), but there is no general stability theory for non-rectangular domains. Despite this, significant work on the noncommutative case has been performed. By far, the largest portion of the work on the noncommutative case for alternating direction schemes has focused on the application to nonlinear differential equations. A particulary notable result is the use of scaling for non-linear systems (see [136–138]). In essence the scaling procedure chooses the iteration parameters as functions of the spatial variables, rather than fixed constants, as was previously done. Another interesting contribution to ADI theory for the non-commutative case was the development of a criterion for the convergence of an alternating direction scheme where commutativity was not assumed, but in practice the condition can be "cumbersome to verify" (see [81]). Consideration of issues relating to complex geometries is nearly absent from the plentiful ADI literature. An early paper, [47], demonstrated on a single fixed mesh, numerical result showing that the ADI could be applied successfully, with at best low-order accuracy, to a complex geometry through the use of fictitious boundary points. Despite this isolated contribution, and as mentioned above, subsequent work has either been rectangular geometries, or relied upon domain mappings, related curvilinear coordinates, domain decomposition, and their combination (e.g., [17, 54, 85, 116, 122]).

Even with this limited geometric generality, the ADI has been applied to various PDEs and innumerable applications; the manifold key developments in this area includes high-order methods (c.f., [60, 94]), application to Maxwell's equations (c.f., [142]), application to Navier-Stokes Equations (c.f., [7, 12]), application to inhomogeneous media (c.f., [116]), application in conjunction with Perfectly Matched Layer (PML) absorbing boundary conditions (c.f., [115]), time-extrapolation for higher temporal accuracy (c.f., [65]), application with spline collocation methods (c.f., [17, 42, 43]), and spectral methods (c.f., [7, 141]). The use of ADI in conjunction with higher-order compact finite difference schemes is a current research area (c.f., [86, 128, 139]).

4.4 Alternating Direction Developments Related to the Current Work

The approach presented in this thesis gives rise to significant improvements over previous approaches, including the applicability to general domains with unconditional stability for elliptic, parabolic, and hyperbolic equations, with a computing time per time-step comparable to those arising from a single step of an explicit finite-difference method. The author knows of no previous alternating direction algorithms possessing these properties, despite significant efforts in these regards. However, the algorithm developed in the present work is not the first high-order accurate alternating direction method that has been produced. A cubic-spline method presented in references [42, 43] demonstrated the capability to solve some separable elliptic partial differential equations over certain irregular geometries. The fourth-order accurate approach is based on alternating direction splittings, but does not exhibit unconditional stability, and, in fact fails to produce correct solutions for some simple geometries, such as the region contained between two squares.

A notable contribution [7], uses an approach based on alternating directions for the solution of non-linear Navier-Stokes equations. Domain decomposition was used in [7] to implement the solver in a parallel manner and high-order accuracy was obtained by means of the Multi-Domain Local Fourier Basis Methods (MDLFB) (see [82]), which extends (or continues) a function and then applies a smooth window to produce periodicity. The resulting periodic function can then be approximated to high-order in a Fourier basis. Their technique has many similarities with our FC-AD approach. While reference [7] demonstrates the power of alternating direction schemes to handle nonlinear equations efficiently and in a parallel computational infrastructure, the work is restricted to rectangular domains. The authors did note however that "The proposed algorithm has the potential for treating efficiently complex geometries. This is a work in progress," but in the decade since this paper was published, no citing paper addresses the issues relating to applying this algorithm for complex geometries. A subsequent paper, [24], by some of the same authors states that "Generally, spectral solution of equations in complex geometries employs either transformation or patching."

We conclude this section on related techniques with a reference first outlined in [55] which presents a method based on removal of discontinuities in function values and derivatives for the resolution of Gibbs phenomenon. In that paper, which does not present any results on the solution of differential equations, an expectation is expressed that the Gibbs resolution methodology presented there, coupled with a suitable splitting scheme, could yield high-order accurate solvers for partial differential equations, "only limited by the requirement that the full scheme shall be stable." The paper itself only speculated on how one might proceed to apply the algorithm in a 2D or higher dimensional setting. The effort apparently culminated in [100], which successfully applied the technique to the solution of the Helmholtz equation in one, two, and three dimensions over rectangular domains. The technique required extra points near the boundaries to obtain the desired accuracies and, as mentioned in Section 3.4 of this thesis, this requirement rules out a direct application to complex geometries.

Chapter 5

FC(Gram) Continuations

In previous chapters, a range of efforts were mentioned [7, 17, 42, 43, 55], which have sought to produce high-order, unconditionally stable alternating direction solvers for complex domains without recourse to domain mappings. The algorithms resulting from those contributions apply with unconditional stability only for simple combinations of rectangular or parallelepipedal domains. In this chapter, an approximation methodology is presented which, as is shown in Chapter 6, when used in conjunction with the idea of Alternating Directions, gives rise to unconditionally stable high-order solvers for complex geometries. This approximation methodology, which we refer to as the Gram-polynomial Fourier-continuation method (FC(Gram)), combines Fourier continuations and Gram orthogonal polynomial projections; the result is a high-order accurate Fourier-based approximation method which, for one-dimensional approximation problems on a discrete mesh of size n requires only $O(n \log(n))$ operations—as opposed to the full $O(n^3)$ cost of computing the SVD required for the FC(SVD) continuation [22, 27, 28].

5.1 The FC(Gram) Algorithm

Consider a smooth function $f \in C^k[x_\ell, x_r]$ (see Chapter 1 and in particular Figure 1.2) for some positive integer k or $k = \infty$. In our application, approximate values of the function we wish to approximate are given on a discrete grid x_j, $j = 1, \ldots, n$. With reference to Figure 1.2, we mention that the FC-AD algorithm introduced in Chapter 6 uses the FC(Gram) continuation method by applying it to the restriction of f to the domain $[x_1, x_n]$-from the slightly larger interval $[x_\ell, x_r]$. Naturally, as noted in Section 5.2.3, this restriction operation has an effect on the overall error and error bounds for the FC approximation of

f in the interval $[x_\ell, x_r]$. For clarity but without loss of generality, within Chapter 5 we replace the interval $[x_1, x_n]$ by the unit interval $[0, 1]$. The discrete grid

$$x_j = (j-1)h, \quad j = 1, \dots, n, \quad h = 1/(n-1); \tag{5.1}$$

the results of this chapter apply, via simple transformations, to the case of a general interval $[x_1, x_n]$ and associated (slightly larger) intervals $[x_\ell, x_r]$ which contains $[x_1, x_n]$

Using the FC(SVD) methods described in Section 3.5.1, a periodic continuation of f can be obtained, although at a computational cost that is significantly higher than is desirable for use as an element of a PDE solver (we mention, however, that, for other uses, including high-order surface representations [28] the use of an $O(n^3)$ continuation algorithm is perfectly adequate). In order to introduce the significantly faster (but one-dimensional) FC(Gram) method, consider a graph displaying both $f(x)$ and $f(x-d-1)$, each one over its domain of definition: $[0, 1]$ and $[1+d, 2+d]$ respectively; see Figure 5.1.

As indicated in the figure, we focus on a small portion on the right end of the graph of $f(x)$ as well as a small portion on the left end of the graph of $f(x-d-1)$, say, the portions defined by in the intervals $[1-\Delta, 1]$ and $[1+d, 1+d+\Delta]$. The FC(SVD) algorithm can be applied in these two line segments to produce a periodic function, with periodicity interval $[1-\Delta, 1+2d+\Delta]$ which simultaneously approximates $f(x)$ on the interval $[1-\Delta, 1]$, and $f(x-d-1)$ on the interval $[1+d, 1+d+\Delta]$—so that, indeed, the periodic continuation can be used to *match* $f(x)$ to $f(x-d-1)$. This new function, $f_{match} \in C^\infty_{per}[1-\Delta, 1+2d+\Delta]$, which is shown in the upper right of Figure 5.1, can be used to produce a smooth transition between the right-end of $f(x)$ and the left-end of $f(x-d-1)$. Thus the prescription

$$f^{de}(x) = \begin{cases} f(x) \quad \text{for} \quad x \in [0, 1] \\ f_{match}(x) \quad \text{for} \quad x \in (1, 1+d] \\ f^{de}(x+1+d) = f^{de}(x) \quad \text{for all } x \text{ in } \mathbb{R}, \end{cases} \tag{5.2}$$

defines a $1+d$-periodic function f^{de} (a “discontinuous extension”) that is, in fact, discontinuous, but it is equal to a smooth function up to the error arising from the FC(SVD) continuation.

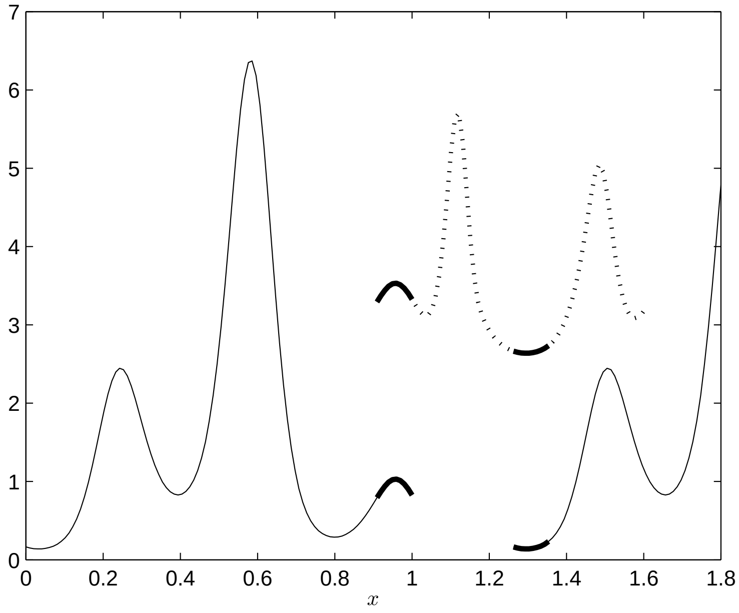


Figure 5.1: Calculation of a periodic extension of $f(x) = e^{\sin(5.4\pi x - 2.7\pi) - \cos(2\pi x)}$ using only a small number of points of the original data ($n_\Delta = 10$). Raised for visibility, the function $f_{match}(x)$ is displayed in the upper-right portion of the figure.

The Fourier continuation series may now be obtained easily. To do this, let

$$n_\Delta = \Delta/h + 1 \quad \text{and} \quad n_d = d/h + 1; \tag{5.3}$$

throughout this thesis we assume the constants Δ and d are selected in such a way that n_Δ and n_d are positive integers. The Fourier continuation for f^{de} is then obtained by sampling this function at the $n + n_d - 2$ evenly spaced points

$$x_j = (j-1)h, \quad j = 1, \ldots n + n_d - 2 \tag{5.4}$$

followed by evaluation of an approximating Fourier series by means of an FFT of the discrete f^{de} values. (Note that evenly spaced points are not used by necessity—one could use an unequally spaced FFT instead—but, rather, because this is the relevant case for the FC-AD methodology to be introduced in Chapter 6).

Out of the $n + n_d - 2$ points used in the interval $[0, 1 + d]$, the points x_j, $j = n - n_\Delta + 1, \ldots, n$ fall within the set $[1 - \Delta, 1]$ and the points $x_j + 1 + d$, $j = n - n_\Delta + 1, \ldots, n$ fall within the set $[1+d, 1+d+\Delta]$, for a total of $2n_\Delta$ points in the set $[1-\Delta, 1]\cup[1+d, 1+d+\Delta]$. Then, for a fixed ratio d/Δ, and taking Δ such that $n_\Delta \leq Cn^{1/3}$ for some constant C, the total number of operations required to construct the Fourier transform of f^{de} is of order $O(n + n_d \log(n + n_d))$. Additionally, if n_Δ is chosen so that $n_\Delta \geq Cn^\gamma$ for some $\gamma > 0$ and some constant C, then, due to the spectral convergence of the FC(SVD) algorithm [28], spectral convergence is expected for this continuation algorithm as well.

Our overall FC(Gram) continuation method results as a slight, but necessary, variation of the algorithm just described: note that, even for "reasonable" values of the number n of one-dimensional samples, the number $n_\Delta \sim O(n^{1/3})$ of discretization points contained in the boundary intervals, which was chosen small to maintain the FFT-type efficiency, is not large enough to give rise to optimally accurate SVD continuations. For reasonable values of n it is much more efficient to use small values of Δ and n_Δ and to precede the FC(SVD) calculation by a projection into a space of orthogonal polynomial, as detailed below. The advantage of this procedure is that the value of the functions which form polynomial basis are known over the entire interval. Thus FC(SVD) continuation of these basis functions can be constructed with an unlimited number of discrete data points, making the continuations very accurate. These separate continuations can be combined to form the continuation of the original function. While the direct evaluation of each individual FC(SVD) will require more computing time, these continuation functions only need to be computed once and stored for each choice of n_Δ, n_d, and the parameters of the FC(SVD) continuation: this precomputation does not affect the $O(n\log(n))$ complexity of the FC(Gram) algorithm.

To provide a complete description of the FC(Gram) algorithm, we thus let $\Delta \sim 1/n$ (typically we use $\Delta t = 9/n$) and we define the spaces

$$C^k[1-\Delta, 1] \quad \text{and} \quad C^k[1+d, 1+d+\Delta] \tag{5.5}$$

of smooth boundary functions, which we endow with the semi-positive definite discrete scalar product

$$(f, g) = \sum f(x_j)g(x_j), \tag{5.6}$$

(where the sum extends to all n_Δ discretization points x_j contained either in $[1-\Delta, 1]$ $(j = 1, \dots, n_\Delta)$ or in $[1+d, 1+d+\Delta]$ $(j = n - n_\Delta + 1, \dots, n)$, as appropriate). Taking fixed values of n_Δ (in all of our examples we use $n_\Delta = 10$), the smooth boundary functions

$f_{\mathit{left}} \in C^k[1-\Delta, 1]$, $f_{\mathit{left}}(x) = f(x)$ for $x \in [1-\Delta, 1]$, and $f_{\mathit{right}} \in C^k[1+d, 1+d+\Delta]$, $f_{\mathit{right}}(x) = f(x-d-1)$ for $x \in [1+d, 1+d+\Delta]$, are approximated with high-order accuracy (in the Δ-length boundary intervals and with an explicit error bound as displayed in Equation (5.15)) by their orthogonal projections, with respect to the scalar product (5.6), onto the subspaces of polynomials of appropriately high degrees. In our examples we use projections onto the space of polynomials of degree $\leq m$ with values such as $m = 4$, $m = 5$ and $m = 9$. (These choices of the numerical values of m are dictated, in part, by *stability* considerations, as discussed in Chapter 10).

In view of these considerations, the algorithm FC(Gram) proceeds as follows:

1. Orthonormal bases $\{P_{r,n_\Delta-1}(x)\}_{r=0}^{m}$ (the Gram polynomials [14, 19]) of the subspaces of polynomials of degree m with regard to the corresponding discrete scalar product (5.6) in both $C^k[1-\Delta, 1]$ and $C^k[1+d, 1+d+\Delta]$ are obtained.

2. The smooth boundary functions f_{left} and f_{right} are projected orthogonally onto elements f^p_{left} and f^p_{right} of the polynomial subspaces:

$$f^p_{\mathit{left}}(x) = \sum_{r=0}^{m} a^r_{\mathit{left}} P_{r,n_\Delta-1}(x) \quad \text{and} \quad f^p_{\mathit{right}}(x) = \sum_{r=0}^{m} a^r_{\mathit{right}} P_{r,n_\Delta-1}(x), \tag{5.7}$$

   where

$$a^r_{\mathit{left}} = (f_{\mathit{right}}, P_{r,n_\Delta-1}) \quad \text{and} \quad a^r_{\mathit{right}} = (f_{\mathit{right}}, P_{r,n_\Delta-1})\,. \tag{5.8}$$

3. Very accurate FC(SVD) continuations $f^{P,Q} \in C^\infty_{per}[1-\Delta, 1+2d+\Delta]$ are produced for pairs $\{P, Q\}$ of Gram polynomials, $P \in C^k[1-\Delta, 1]$ and $Q \in C^k[1+d, 1+d+\Delta]$, resulting in a continuation function $f^{P,Q} \in C^\infty_{per}[1-\Delta, 1+2d+\Delta]$ that approximates P in $[1-\Delta, 1]$ and approximates Q in $[1+d, 1+d+\Delta]$ for each pair $\{P, Q\}$; various types of polynomial pairings are admissible as are methods to effect their joint continuation; full details concerning our prescriptions in these regards are presented in Section 5.1.1. Here we note that, as prescribed in Section 5.1.1, the method we use leads to certain continuation functions $f^r_{\mathit{even}}(x)$ and $f^r_{\mathit{odd}}(x)$, $r = 0 \ldots m$.

4. The function $f_{\mathit{match}}(x)$ (see Equation (5.2) and the paragraph immediately preceding it) is obtained as the following linear combination of the FC(SVD) continuations

mentioned in point 3.:

$$f_{match}(x) = \sum_{r=0}^{m} \frac{a^r_{left} + a^r_{right}}{2} f^r_{even}(x) + \frac{a^r_{left} - a^r_{right}}{2} f^r_{odd}(x). \tag{5.9}$$

5. The function $f^{dp}(x)$ is constructed according to the formula

$$f^{dp}(x) = \begin{cases} f^p_{right}(x-1-d) \quad \text{for} \quad x \in [0,\Delta] \\ f(x) \quad \text{for} \quad x \in (\Delta, 1-\Delta) \\ f^p_{left}(x) \quad \text{for} \quad x \in [1-\Delta, 1] \\ f_{match}(x) \quad \text{for} \quad x \in (1, 1+d) \\ f^{dp}(x+1+d) = f^{dp}(x) \quad \text{for all } x \text{ in } \mathbb{R}, \end{cases} \tag{5.10}$$

and the unique $n+n_d-2$ term Fourier series, f^c, that interpolates f^{dp} at the points x_j for $j = 1 \ldots n+n_d-2$ (see Equation (5.4)) is calculated by an FFT, thus completing the approximation.

Remark 5.1.1. *The specific selection prescribed in Step 5. for f^{dp} is important for the specific use within the FC-AD methodology to provide stability; c.f., the stability analysis presented in Chapter 10; otherwise alternate and simpler functions might be considered.*

Clearly, a necessary requirement for the FC(Gram) approximation to be accurate is that the functions $f(x)$ and $f(x-d-1)$ are well approximated by their Gram-polynomial projections over the intervals $[1-\Delta, 1]$ and $[1+d, 1+d+\Delta]$ respectively. For smooth functions such as those in either of the boundary spaces defined in Equation (5.5), let p_{left} and p_{right} be the optimal m degree polynomial approximations *in the maximum norm* in the Δ-length boundary regions. The error made by approximating by the best polynomial is well known (c.f., [106]) and gives, for the present case gives, for example, equality

$$\|f_{left}(x) - p_{left}(x)\|_{L^\infty[1-\Delta,1]} = \frac{\Delta^{m+1}|f^{(m+1)}(\xi)|}{2^{2m+1}(m+1)!}, \tag{5.11}$$

for some point ξ in the Δ-length interval, with a similar equality for f_{right} and p_{right}. Since p_{left} is in the space of polynomials of degree $\leq m$, its projection, according to (5.6), into a polynomial basis is equal to itself. Thus, the error made in projecting a f_{left} into an

$L(m)$	$m=4$	$m=5$	$m=9$
$n_\Delta = 10$	1.7021	1.9261	17.849

Table 5.1: Lebesgue constants for various parameters of interest in the present section and calculated as a maximum over 30000 points (see Equation (5.13)).

orthogonal m degree basis over the interval $[1-\Delta,1]$ is bounded by

$$\begin{aligned} \|f_{\mathit{left}}(x) - f^p_{\mathit{left}}(x)\|_{L_\infty[1-\Delta,1]} \quad &\leq \quad \|f_{\mathit{left}}(x) - p_{\mathit{left}}(x)\|_{L^\infty[1-\Delta,1]} \\ &+ \sum_{r=0}^{m} \left(f_{\mathit{left}}(x) - p_{\mathit{left}}(x), P_{r,n_\Delta-1}\right) P_{r,n_\Delta-1}(x). \end{aligned} \tag{5.12}$$

Introducing the constant $L(m)$ (related to and a bound for the Lebesgue constant, c.f., [107]) for the discrete projection, which is defined by

$$L(m) = \max_{x\in[1-\Delta,1]} \sum_{r=0}^{m} (1, |P_{r,n_\Delta-1}|)\, |P_{r,n_\Delta-1}(x)|, \tag{5.13}$$

we then obtain the bound

$$\sum_{r=0}^{m} \left(f_{\mathit{left}}(x) - p_{\mathit{left}}(x), P_{r,n_\Delta-1}\right) P_{r,n_\Delta-1}(x) \leq L(m) \|f_{\mathit{left}}(x) - p_{\mathit{left}}(x)\|_{L^\infty[1-\Delta,\Delta]}, \tag{5.14}$$

and therefore the bound on the error of the projection is given by

$$\|f_{\mathit{left}}(x) - f^p_{\mathit{left}}(x)\|_{L_\infty[1-\Delta,1]} \leq (1 + L(m)) \frac{\Delta^{m+1} |f^{(m+1)}(\xi)|}{2^{2m+1}(m+1)!}. \tag{5.15}$$

A similar bound is obtained for the projection of f_{right} over the interval $[1+d, 1+d+\Delta]$. The calculated values of the bounds $L(m)$ on the Lebesgue constants for the values pertinent to this chapter are shown in Table 5.1.

5.1.1 Useful Gram Polynomial Pairs and Accurate FC(SVD) Continuations of Gram Polynomials

Recall that Point 3. in the description of the FC(Gram) algorithm (above in Section 5.1) does not prescribe either how to select pairs $\{P, Q\}$ of Gram polynomials, $P \in C^k[1-\Delta, 1]$ and $Q \in C^k[1+d, 1+d+\Delta]$ for subsequent continuation via the FC(SVD) approach, or how the corresponding continuations are to be performed; in this section we present the details of such choices and their FC(SVD) continuations. Our choice of polynomial pairs is made in such a way as to require SVDs of the smallest possible dimensionality. We thus use polynomial pairs of the form $\{P, Q\}$ with $P(x) = P_{r,n_\Delta-1}(x)$ and $Q(x) = P_{r,n_\Delta-1}(x-d-\Delta)$ for each $r \leq m$; see Point 1. in the prescription of the FC(Gram) algorithm. We will call this an even pair, since an FC(SVD) continuation can be produced, which will be denoted by f^r_{even}, that uses only even Fourier coefficients; see Equation (5.22). Similarly, the pair $\{P, -Q\}$ can be represented by a FC(SVD) continuation f^r_{odd}, shown in Equation (5.25), that incorporates odd Fourier coefficients only. The set of all continuations of these pairs of Gram polynomials contains $2m+2$ elements that, clearly, can be used to form the matching function f_{match}—since

$$\frac{f^r_{even} + f^r_{odd}}{2} \approx \begin{cases} P_{r,n_\Delta-1}(x) \quad \text{for} \quad x \in [1-\Delta, 1] \\ 0 \quad \text{for} \quad x \in [1+d, 1+d+\Delta] \end{cases}, \tag{5.16}$$

$$\frac{f^r_{even} - f^r_{odd}}{2} \approx \begin{cases} 0 \quad \text{for} \quad x \in [1-\Delta, 1] \\ P_{r,n_\Delta-1}(x-d-\Delta) \quad \text{for} \quad x \in [1+d, 1+d+\Delta] \end{cases}, \tag{5.17}$$

while the set of all $P_{r,n_\Delta-1}(x)$ can be used to approximate f_{left} in the interval $[1-\Delta, 1]$ and the set of all $P_{r,n_\Delta-1}(x-d-\Delta)$ can be used to approximate f_{right} in the interval $[1+d, 1+d+\Delta]$.

To produce the functions f^r_{even}, each one of which should clearly be periodic with period $2d+2\Delta$,

$$f^r_{even}(x) = \sum_{k \in t(g)} a^r_k e^{\frac{\pi i}{d+\Delta}(x-1+\Delta)k}, \tag{5.18}$$

(where the quantity $t(g)$ is defined below Equation (3.2)) we apply the FC(SVD) method to match the polynomial pair $\{P_{r,n_\Delta-1}(x), P_{r,n_\Delta-1}(x-d-\Delta)\}$ at the points $\hat{x}_j$ and $\hat{x}_j + d + \Delta$ for $j = 1 \dots \Upsilon$ where

$$\hat{x}_j = 1 - \Delta + \frac{(j-1)\Delta}{\Upsilon - 1}, \quad j = 1, \dots, \Upsilon. \tag{5.19}$$

Note that g in Equation (5.18) equals the number of Fourier modes used in the FC(SVD) continuation.

We obtain the Fourier coefficients a^r_k by solving, in sense of least squares and by means of an SVD, the system of equations formed by the left-interval equations

$$f^r_{even}(\hat{x}_j) \approx \sum_{k\in t(g)} \hat{a}^r_k e^{\frac{\pi i}{d+\Delta}\hat{x}_j k} = P_{r,n_\Delta-1}(\hat{x}_j), \quad j = 1\ldots\Upsilon, \tag{5.20}$$

together with the right-interval equations

$$f^r_{even}(\hat{x}_j) = \sum_{k\in t(g)} (-1)^k \hat{a}^r_k e^{\frac{\pi i}{d+\Delta}\hat{x}_j k} = P_{r,n_\Delta-1}(\hat{x}_j), \quad j = 1\ldots\Upsilon. \tag{5.21}$$

In view of the uniqueness of solution for this least-squares problem, it is easy to show that the odd-indexed coefficients vanish, so that, indeed, the overall system is reduced to a least-squares problem for the even-indexed coefficients:

$$f^r_{even}(\hat{x}_j) = \sum_{\substack{k\in t(g)\\ k\ even}} \hat{a}^r_k e^{\frac{\pi i}{d+\Delta}\hat{x}_j k} = P_{r,n_\Delta-1}(\hat{x}_j), \quad j = 1\ldots\Upsilon. \tag{5.22}$$

Similarly, an odd continuation is calculated with the left-interval equations

$$f^r_{odd}(\hat{x}_j) \approx \sum_{k\in t(g)} \hat{b}^r_k e^{\frac{\pi i}{d+\Delta}\hat{x}_j k} = P_{r,n_\Delta-1}(\hat{x}_j), \quad j = 1\ldots\Upsilon, \tag{5.23}$$

and the right-interval equations

$$f^r_{odd}(\hat{x}_j) = \sum_{k\in t(g)} (-1)^k \hat{b}^r_k e^{\frac{\pi i}{d+\Delta}\hat{x}_j k} = -P_{r,n_\Delta-1}(\hat{x}_j), \quad j = 1\ldots\Upsilon. \tag{5.24}$$

These constraints can similarly be reduced by the previous uniqueness arguments to

$$f^r_{odd}(\hat{x}_j) = \sum_{\substack{k\in t(g)\\ k\ odd}} \hat{b}^r_k e^{\frac{\pi i}{d+\Delta}\hat{x}_j k} = P_{r,n_\Delta-1}(\hat{x}_j), \quad j = 1\ldots\Upsilon. \tag{5.25}$$

The functions f^r_{even} and f^r_{odd} just constructed allow us to obtain the matching function f_{match} by means of the expression given in Equation 5.9.

A requirement for the accuracy of the FC(Gram) approximation is that the continuations of Gram polynomials must themselves be accurate approximations of the polynomials. In our examples, the number g of Fourier modes used is typically set to equal $\Upsilon/2$ (c.f., [28]). In order to achieve approximations of the desired accuracy, as noted in the previous section,

	$Error$		$Error$
f^0_{even}	$5.6 \cdot 10^{-17}$	f^0_{odd}	$1.7 \cdot 10^{-16}$
f^1_{even}	$6.1 \cdot 10^{-16}$	f^1_{odd}	$5.6 \cdot 10^{-16}$
f^2_{even}	$2.9 \cdot 10^{-15}$	f^2_{odd}	$2.5 \cdot 10^{-15}$
f^3_{even}	$1.4 \cdot 10^{-14}$	f^3_{odd}	$1.6 \cdot 10^{-14}$
f^4_{even}	$1.4 \cdot 10^{-13}$	f^4_{odd}	$5.6 \cdot 10^{-14}$
f^5_{even}	$4.2 \cdot 10^{-13}$	f^5_{odd}	$5.3 \cdot 10^{-13}$
f^6_{even}	$5.5 \cdot 10^{-12}$	f^6_{odd}	$1.8 \cdot 10^{-12}$
f^7_{even}	$1.8 \cdot 10^{-11}$	f^7_{odd}	$3.2 \cdot 10^{-11}$
f^8_{even}	$4.4 \cdot 10^{-10}$	f^8_{odd}	$1.4 \cdot 10^{-10}$
f^9_{even}	$2.2 \cdot 10^{-9}$	f^9_{odd}	$4.0 \cdot 10^{-9}$

Table 5.2: Maximum errors (evaluated as maxima over 1800 points in the set $[1-\Delta, 1] \cup [1+d, 1+d+\Delta]$) that result from the even and odd continuations of the Gram Polynomials for the parameter values $n_\Delta = 10$, $d/\Delta = 26/9$, $g = 63$, and $\Upsilon = 150$.

we select appropriately large values of Υ. The number g of Fourier continuation modes, on the other hand, must lie below $2n_\Delta(1 + d/\Delta)$ to prevent aliasing. The explicit values of these parameters used for the numerical examples in this thesis are given in Remark 5.1.3. When the method is used as intended, as explained in Remark 5.1.2, these parameter values give rise to errors that are smaller than those implicit in other portions of the FC(Gram) approximations. Figure 5.2 shows the calculated basis for $r \leq 3$.

Remark 5.1.2. *We see from Table 5.2 that the error in the continuation of the pairs of Gram polynomials increases with the polynomial degree. In practice this has no negative effect on the actual convergence of these continuations to the original functions f_{left} and f_{right} for the small values of Δ encountered in our applications, since, as follows from Equation (5.15), the scalar product (f, P_{r,n_Δ}) decreases as $O(\Delta^r)$. For example, even though an error of $4.21 \cdot 10^{-13}$ is made for the even continuation of $P_{5,9}(x)$, the coefficient is of order $O(\Delta^5)$ and thus all the values in Table 5.2 result in errors that, as indicated above and as demonstrated in Section 5.3, are smaller than those implicit in other portions of the algorithm. The error can be made to decrease spectrally fast [28] by increasing the parameters n_Δ, g, and Υ since g can be chosen proportional to n_Δ and then, for fixed d/Δ, as g and Υ increase, spectral accuracy is obtained.*

Remark 5.1.3. *The calculation of the FC(SVD) continuations of the Gram polynomials depends on a small number of parameters: d/Δ, n_Δ, g, and Υ. A single set of values of these parameters, namely $n_\Delta = 10$, $d/\Delta = 26/9$, $g = 63$, and $\Upsilon = 150$, can be used for*

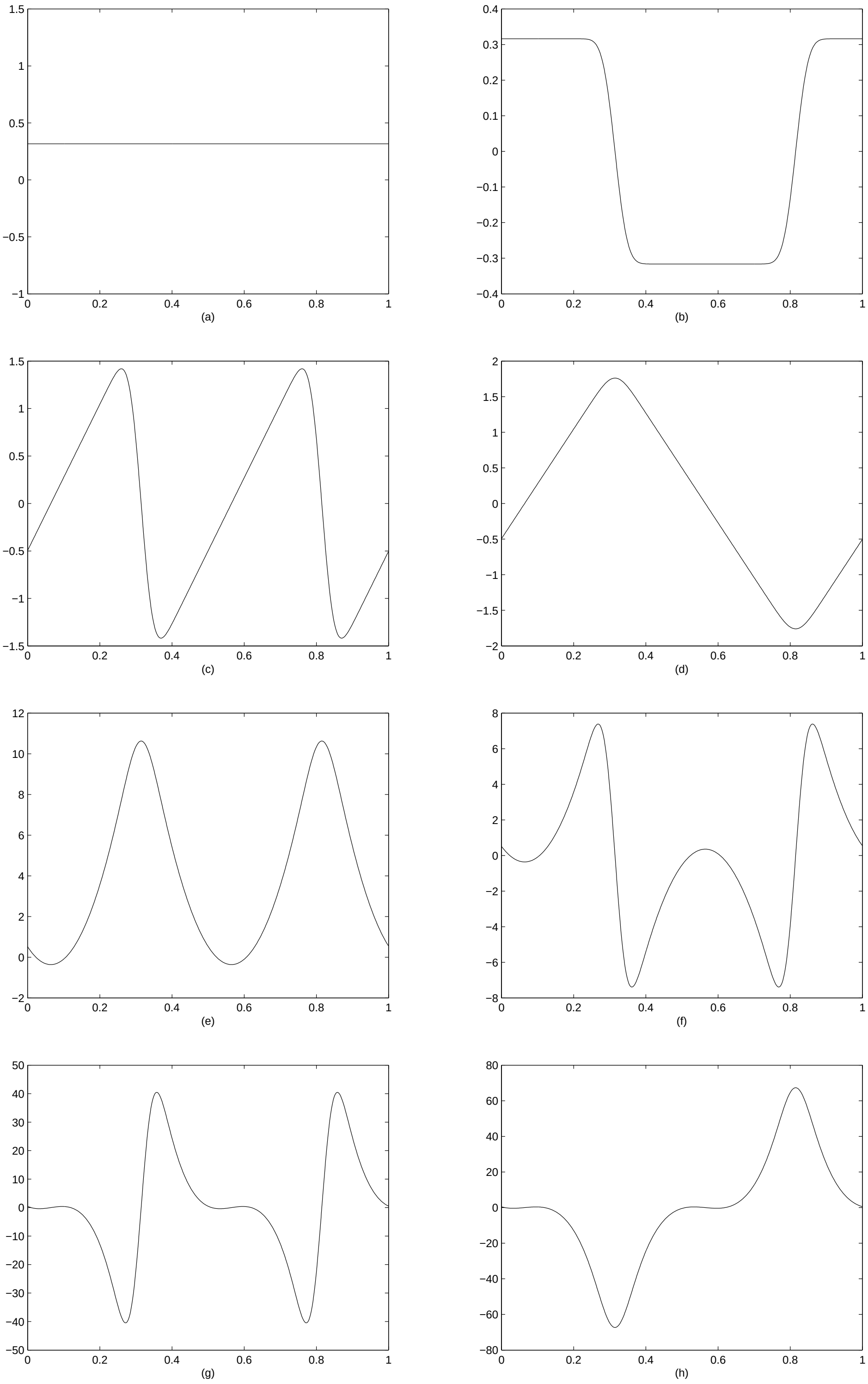


Figure 5.2: Display of the calculated functions: (a) f^0_{even}, (b) f^0_{odd}, (c) f^1_{even}, (d) f^1_{odd}, (e) f^2_{even}, (f) f^2_{odd}, (g) f^3_{even}, (h) f^3_{odd} scaled over the unit interval and sampled at 1000 discrete points for display purposes.

general applications and arbitrary accuracy, up to essentially the level of machine precision roundoff. These values were used, in particular, for all of the numerical examples presented in this thesis. In view of the near-universality of these parameters, it becomes possible to obtain generic continuation functions f^r_{even} and f^r_{odd} ($0 \leq r \leq 9$) that can be used to deal with all continuation problems that may arise in an application of the PDE solver proposed in this thesis. The Fourier coefficients of these functions were thus calculated and their values at 35 points were stored in a 20×35 double precision matrix F. (Seventy points x_j lie in the interval $1 - \Delta \leq x_j < 1 + 2d + \Delta$; only half of the corresponding values $f^r_{even}(x_j)$ and $f^r_{odd}(x_j)$ need to be stored, however, since the values of the function f^r_{even} (resp. f^r_{odd}) in the interval $[1 + d, 1 + 2d + \Delta]$ equal the corresponding value (resp. minus the corresponding values) over the interval $[1 - \Delta, 1]$.) This matrix can be applied to calculate the FC(Gram) continuation of any function. The effect of using the fixed matrix F on the error of the FC(Gram) approximation is considered in Remark 5.2.2 below. The calculation leading to the matrix F was performed in Maple's high precision environment to eliminate all (small) errors arising from the conditioning of the FC(SVD) linear system. It was found that with the parameters given above, 48 digits of arithmetic precision were sufficient to guarantee all matrix values were obtained with 16 correct digits (double precision) which were eventually stored. (Interestingly, the computational time required by these one-time-only high-precision computations amounted to a few tens of minutes.) All other computations leading to results in this thesis were performed in the standard double precision environment. In particular, the maximum errors reported in Table 5.2 resulted from use of the values stored in the matrix F, followed by a double precision inverse FFT to produce the approximating Fourier series which was eventually subtracted from the actual Gram-polynomial values to obtain the approximation error at a large number of comparison points.

Remark 5.1.4. *Since n_Δ is taken to be fixed (see Remark 5.1.3 above), the associated value of Δ may be such that the various cases in Equation (5.10) may give rise to conflicting definitions. Indeed, if $n < 2n_\Delta$ then the segments $[0, \Delta]$ and $[1 - \Delta, 1]$ will overlap at least at a point. There are two options to resolve this. One is that mesh refinement within the FC-AD framework may be performed as shown in Section 6.2 to avoid the occurrence of such small values of n. The other is to first approximate the function f over all $n < 2n_\Delta$ points by a m degree polynomial which ensures that $f^p_{left}(x) = f^p_{right}(x - 1 - d)$ over their* *common domain of definition. For $n \leq n_\Delta$, while mesh refinement still leads to stable and accurate solutions, our numerics indicate that latter option should not be chosen as it does not maintain the stability of FC-AD algorithms.*

5.2 Accuracy of the Approximation

In this section we provide an estimate of the error resulting from the FC(Gram) approximation of a smooth function $f \in C^k[0,1]$, where k is either a sufficiently large positive integer or $k = \infty$. To obtain our error bound we evaluate the errors arising from each of the error-generating elements of the method, namely steps 2., 3., and 5. of the algorithm described in Section 5.1. A certain blending of f_{match} and f used in this section, which results from use of a smooth windowing function (see Figure 5.3) is an important element of the analysis. As pointed out in Remark 5.2.1, however, the final results apply, up to a small correction, even when the blending procedure is not used. In a couple of instances within our analysis it will prove necessary to perform such blending operations on functions defined previously in this text which, like f^{de}, differ by little from a smooth function. The resulting smooth functions will be distinguished by superimposing a bar to the function name so that, e.g., the smooth, blended version of f^{de} becomes $\overline{f^{de}}$; see Equation (5.31) below.

Error arising from step 2. of the FC(Gram) algorithm. Step 2. of the FC(Gram) algorithm projects the boundary functions f_{left} and f_{right} arising from f onto spaces of polynomials of degree $\leq m$. Recall that for the prescriptions in Section 5.1, the polynomial projections use n_Δ discrete evenly spaced points in Δ-sized intervals, resulting in the error bound shown in Equation (5.15) for the interval $[1-\Delta, 1]$. Including the equivalent result for the interval $[1+d, 1+d+\Delta]$, we have

$$\|f_{\mathit{left}}(x) - f^p_{\mathit{left}}(x)\|_{L_\infty[1-\Delta,1]} \leq (1 + L(m)) \frac{\Delta^{m+1} M}{2^{2m+1}(m+1)!}, \tag{5.26}$$

and

$$\|f_{\mathit{right}}(x) - f^p_{\mathit{right}}(x)\|_{L_\infty[1+d,1+d+\Delta]} \leq (1 + L(m)) \frac{\Delta^{m+1} M}{2^{2m+1}(m+1)!}, \tag{5.27}$$

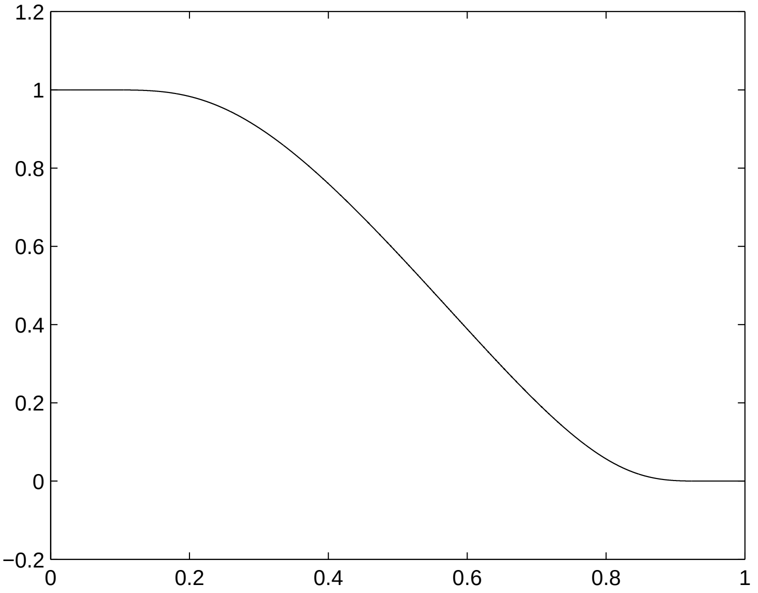


Figure 5.3: Sample smooth windowing function $w(x) = e^{\frac{2e^{1/x}}{x-1}}$ over the unit interval.

where M is the largest value of $f^{m+1}(x)$ over the intervals, $[1-\Delta, 1]$ and $[0, \Delta]$ (whose values coincide with those of the corresponding derivatives of f_{left} and f_{right} on the intervals $[1-\Delta, 1]$ and $[1+d, 1+d+\Delta]$ respectively). The relevant values of the parameter $L(m)$ are shown in Table 5.1.

Error arising from step 3. of the FC(Gram) algorithm. An additional component of the error in the FC(Gram) algorithm arises from the inaccuracies in the FC(SVD) continuations of the Gram polynomial pairs as defined in Section 5.1.1. In a more general setting the existence of such continuations and the convergence properties of FC(SVD) continuations have been considered in references [28]. This error in FC(SVD) continuation of the Gram polynomial pairs can be measured by noting how well f_{match} approximates the polynomials f^p_{left} and f^p_{right} over the domains $[1-\Delta, 1]$ and $[1+d, 1+d+\Delta]$ respectively. Since, by Equation (5.9), the function f_{match} is a linear combination of the functions f^r_{even} and f^r_{odd}, for $r = 0, \ldots m$, the error is a linear combination of the errors listed in Table 5.2 with coefficients given by Equation (5.8). An orthonormal basis was selected in step 1., with the coefficients a^r_{left} and a^r_{right}

$$|a^r_{left}| \leq \sqrt{n_\Delta}\|f(x)\|_{L^\infty[1-\Delta,1]} \quad \text{for} \quad r = 0, \dots, m, \tag{5.28}$$

and

$$|a^r_{right}| \leq \sqrt{n_\Delta}\|f(x-1-d)\|_{L^\infty[1+d,1+d+\Delta]} \quad \text{for} \quad r = 0, \dots, m. \tag{5.29}$$

Therefore,

$$\begin{aligned} \left\|f_{match}(x) - f^p_{left}(x)\right\|_{L_\infty[1-\Delta,1]} &\leq S_0(m) \\ \left\|f_{match}(x) - f^p_{right}(x)\right\|_{L_\infty[1+d,1+d+\Delta]} &\leq S_0(m), \end{aligned} \tag{5.30}$$

where $S_0(m)$ depends on d/Δ, n_Δ, g and Υ: $S_0(m) = S_0(m, d/\Delta, n_\Delta, g, \Upsilon)$. In particular if η is the maximum error in Table 5.2 for $r \leq m$, then $S_0(m) = \eta m \sqrt{n_\Delta}\|f(x)\|_{L^\infty[0,1]}$. The value of $S_0(m)$ decays spectrally as the values of g, Υ, and n_Δ are increased (c.f. [28]) since it results from convergent FC(SVD) continuations. As previously noted in Remark 5.1.2, the errors arising in step 3. are negligible in practice.

A smooth approximation $\overline{f^{de}}$ of the discontinuous function f^{de} can be obtained using a smooth windowing function (see Figure 5.3), scaled to a Δ-length interval:

$$\overline{f^{de}} = \begin{cases} f(x) + w(x/\Delta)(f_{match}(x+1+d) - f(x)) \quad \text{for} \quad x \in [0, \Delta] \\ f(x) \quad \text{for} \quad x \in (\Delta, 1-\Delta) \\ f(x) + w((1-x)/\Delta)(f_{match}(x) - f(x)) \quad \text{for} \quad x \in [1-\Delta, 1+d+\Delta] \end{cases}, \tag{5.31}$$

and directly from the property $0 \leq w(x) \leq 1$ and from the bounds (5.30), (5.26), and (5.27), it follows that

$$\left\|f(x) - \overline{f^{de}}(x)\right\|_{L_\infty[0,1]} \leq S_0(m) + (1 + L(m))\frac{\Delta^{m+1}M}{2^{2m+1}(m+1)!}. \tag{5.32}$$

Finally considering the definition of f^{dp} in Equation (5.10) yields,

$$\left\|f^{dp}(x) - \overline{f^{de}}(x)\right\|_{L_\infty[0,1]} \leq S_0(m) + (1 + L(m))\frac{\Delta^{m+1}M}{2^{2m+1}(m+1)!}. \tag{5.33}$$

Error arising from step 5. of the FC(Gram) algorithm. Step 5. of the algorithm produces the full FC(Gram) approximation f^c by applying an FFT to f^{dp} and thus providing a Discrete Fourier Transform (DFT) approximation. This gives rise to additional errors except at the discretization points, x_j, $j = 1, \ldots, n$, since the DFT is interpolatory. In order to obtain a bound for the error arising from this Fourier series approximation we consider the exact Fourier series of the smooth and $(1+d)$-periodic function $\overline{f^{de}}$

$$\overline{f^{de}}(x) = \frac{1}{1+d} \sum_{k-\infty}^{\infty} c_k^e e^{-\Psi_k x}. \tag{5.34}$$

and its DFT

$$\mathrm{DFT}\left(\overline{f^{de}}\right)(x) = \frac{1}{1+d} \sum_{k \in t(n+n_d-2)} c_k e^{-\Psi_k x}, \tag{5.35}$$

with $\Psi_k = \frac{2\pi i k}{1+d}$ and where h is given by (5.1). The coefficients c_k in Equation (5.35) do not match the exact Fourier coefficients of the function $\overline{f^{de}}$, which are given by

$$c_k^e = \int_0^{1+d} \overline{f^{de}}(x) e^{\Psi_k x} dx. \tag{5.36}$$

The error E_{tot} introduced by approximating the function $\overline{f^{de}}$ by the function DFT $\left(\overline{f^{de}}\right)$ can be decomposed into the error E_{trunc} arising from the series truncation, and the error arising by approximating the coefficients c_k^e by c_k. As is known [31, 83], however, the combined error E_{tot} can be bounded in terms of the truncation error: there is a constant C_I such that

$$|E_{tot}| \leq C_I \log(N) |E_{trunc}|, \tag{5.37}$$

where N is the number of data points used for the evaluation of the DFT. To obtain a bound on E_{tot} we thus seek a corresponding bound on the truncation error E_{trunc}.

We clearly have

$$|E_{trunc}| \leq \sum_{k \notin t(n+n_d-2)} |c_k^e|. \tag{5.38}$$

In order to bound the magnitude of the coefficients c_k^e, we apply integration by parts q times to Equation (5.36) which yields

$$c_k^e = \frac{(-1)^q}{\Psi_k^q} \int_0^{1+d} \overline{f^{de}}^{(q)}(x) e^{\Psi_k x} dx,$$

where the boundary terms of the integration by parts procedure vanish since $\overline{f^{de}}$ is a smooth periodic function. Considering the derivatives of the function $\overline{f^{de}}(x)$, we note that over the interval $[1, 1+d]$, $\overline{f^{de}}^{(q)}(x) = f^{(q)}_{match}(x)$, but $f_{match}(x)$ is a linear combination of $f^r_{even}(x)$ and $f^r_{odd}(x)$ for $r = 0, \ldots, m$. It can be seen easily that the latter functions, in turn, are scaled versions of a set of C^∞ functions that are independent of d, Δ and h. From Remark 5.1.3 we have $d = 26/9\Delta$, and we therefore see that for all $q \geq 0$ $f^{(q)}_{match}(x)$ is bounded by Δ^{-q} times a value independent of h, d, and Δ. Similarly, $w^{(q)}(x)$ has bounded derivatives for any $q \geq 0$, so that both $d^q/dx^q\,(w(x/\Delta))$ over the interval $[0, \Delta]$ and $d^q/dx^q\,(w((1-x)/\Delta))$ over the interval $[1-\Delta, 1]$ are bounded by Δ^{-q} times a value independent of h, d, and Δ. Thus if $f \in C^q[0,1]$, by application of the chain rule for differentiation, there exist constants F_q and M_q independent of h, d, and Δ such that

$$\left|\frac{d^q}{dx^q}\overline{f^{de}}(x)\right| \leq \begin{cases} M_q\Delta^{-q} & \text{for} \quad x \in [0, \Delta] \\ F_q \quad \text{for} & x \in [\Delta, 1-\Delta] \\ M_q\Delta^{-q} & \text{for} \quad x \in [1-\Delta, 1+d]. \end{cases} \tag{5.39}$$

Equation (5.39) then gives rise to the following bound for the coefficients c^e_k

$$|c^e_k| \leq \frac{(1+d)^q}{(2\pi k)^q}F_q + \frac{(d+2\Delta)(1+d)^q}{(2\pi k\Delta)^q}M_q. \tag{5.40}$$

From the definition of the function t just below Equation (3.2) note that for $k \notin t(n + n_d - 2)$ we have $|k| \geq (1+d)/2h$. Clearly for all $q \geq 1$ and $n \geq 1$ we also have

$$\sum_{k=n}^{\infty}\frac{1}{k^q} \leq \frac{1}{n^q} + \int_n^\infty \frac{1}{k^q}dk = \frac{1}{n^q} + \frac{1}{(q-1)(n)^{q-1}} \leq \frac{q}{(q-1)n^q}. \tag{5.41}$$

Therefore from Equations (5.40) and (5.38), we obtain the bound

$$|E_{trunc}| \leq \frac{q(1+d)h^{q-1}}{(q-1)\pi^q}F_q + \frac{q(d/\Delta+2)(1+d)h^{q-1}\Delta}{(q-1)(\pi\Delta)^q}M_q. \tag{5.42}$$

Then from the bound (5.37) it follows that

$$\begin{aligned} \left\|\mathrm{DFT}\left(\overline{f^{de}}\right)(x) - \overline{f^{de}}(x)\right\|_{L_\infty[0,1]} \quad \leq \quad & C_I \log(n+n_d-2)\frac{q(1+d)h^{q-1}}{(q-1)\pi^q}F_q \\ & + C_I \log(n+n_d-2)\frac{q(d/\Delta+2)(1+d)h^{q-1}\Delta}{(q-1)(\pi\Delta)^q}M_q. \end{aligned} \tag{5.43}$$

We can thus state the main results of this section, which present as a series of three remarks.

Remark 5.2.1. *A windowing function was used as part of the preceding analysis but is not part of prescription of the FC(Gram) method as described in Section 5.1. Following that prescription exactly involves the calculation error resulting from a Fourier series of f^{dp} instead of $\overline{f^{de}}$. Noting the result (5.33) and considering again the bound (5.37) immediately gives*

$$\left\| f^c(x) - DFT\left(\overline{f^{de}}\right)(x) \right\|_{L_\infty[0,1]} \leq C_I \log(n+n_d-2) \left[S_0(m) + (1+L(m)) \frac{\Delta^{m+1} M}{2^{2m+1}(m+1)!} \right], \tag{5.44}$$

from which we obtain a bound for the error of the FC(Gram) approximation as prescribed in Section 5.1:

$$\|f(x) - f^c(x)\|_{L_\infty[0,1]} \leq \log(n+n_d-2)\left(\zeta_1 + \zeta_2 + \zeta_3 + \zeta_4\right), \tag{5.45}$$

with

$$\zeta_1 = (C_I+1)(1+L(m)) \frac{\Delta^{m+1} M}{2^{2m+1}(m+1)!}, \tag{5.46}$$

$$\zeta_2 = (C_I+1) S_0(m), \tag{5.47}$$

$$\zeta_3 = C_I \frac{q(1+d)h^{q-1}}{(q-1)\pi^q} F_q, \tag{5.48}$$

and

$$\zeta_4 = C_I \frac{q(d/\Delta+2)(1+d)h^{q-1}\Delta}{(q-1)(\pi\Delta)^q} M_q. \tag{5.49}$$

Remark 5.2.2. *Recall from Equation (5.3), that $\Delta = (n_\Delta - 1)h$. Using this value for Δ in equation (5.49) gives*

$$\zeta_4 = C_I \frac{q(d/\Delta+2)(1+d)}{(q-1)(\pi(n_\Delta-1))^q} M_q, \tag{5.50}$$

which clearly does not tend to zero as h and consequently d and Δ are refined. This bound is therefore not ideal. Consider the alternate definition $\Delta = I(n_\Delta - 1)h$, for some positive integer I. One could still project the functions f_{left} and f_{right} into the Gram polynomials using n_Δ points in step 2. of the FC(Gram) algorithm (just use every Ith point to do the projection). Therefore the terms ζ_2, ζ_3 will not be affected by the change, nor will the parameter M_q be changed but the value of ζ_4 will be decreased by a factor of nearly I^{q-1} while ζ_1 is increased by a factor of I^{m+1}. Therefore there exists a procedure to enforce that *the sum* $(\zeta_1+\zeta_2+\zeta_3+\zeta_4) \leq \varepsilon$ *for any positive value* ε *and proceeds as follows: choose parameters* n_Δ, g, *and* Υ *for fixed values of* m *and* d/Δ *such that* $\zeta_2 \leq \varepsilon/4$. *Next the parameter* I *may be chosen such that* $\zeta_4 \leq \varepsilon/4$. *The remaining terms,* ζ_1 *and* ζ_3 *decay as* h^{m+1} *and* h^{q-1} *respectively and therefore for small enough* h *will produce the prescribed error. It is numerically observed (see Section 5.3) that taking* $I = 1$, *as is prescribed by the FC(Gram) approximation, is sufficient to ensure that the non-vanishing errors (relating to the bounds* ζ_2 *and* ζ_3*) are indistinguishable from machine precision roundoff errors and* $O(h^{m+1})$ *convergence is observed. The logarithmic term,* $\log(n+n_d-2)$, *is bounded by a small number for all reasonable values of* n.

Remark 5.2.3. *It was noted at the beginning of Section 5.1 that the FC-AD PDE solver involves restriction of the function* f *from a certain interval* $[x_\ell, x_r]$ *to a certain interval* $[x_1, x_n]$ *which, in the context of this section, was assumed (without loss of generality) to equal the interval* $[0,1]$. *In order to extend our error bounds to yield bounds on the error of the approximation of* f *by* f^c *over an interval slightly larger than* $[0,1]$, *we need only consider the errors that arise in steps 2. and 3. where the accuracy of the orthogonal polynomial projection must be considered over a slightly larger interval and consequently the accuracy of the FC(SVD) continuations of Gram polynomial pairs must also be considered in the marginally extended interval. Consequently only the parameters* ζ_1 *and* ζ_2 *in Equation (5.45) are changed from the case presented above. Recall from Chapter 1 (see Figures 1.1 and 1.2) that* $x_1 - x_\ell < h$ *and* $x_r - x_n < h$ *and then let*

$$\epsilon = \max(x_1 - x_\ell, x_r - x_n) \quad \textit{and} \quad \Delta^\epsilon = \Delta + \epsilon. \tag{5.51}$$

Then following exactly the same bounding procedures presented above but extrapolating to a slightly larger interval yields the results:

$$\zeta_1 = (C_I + 1)(1 + L^\epsilon(m)) \frac{(\Delta^\epsilon)^{m+1} M}{2^{2m+1}(m+1)!} \tag{5.52}$$

and

$$\zeta_2 = (C_I + 1) S_0^\epsilon(m), \tag{5.53}$$

where the value of $L^\epsilon(m)$ *for all* $\epsilon \in [0,h]$ *is shown as a function of* ϵ *in Figure 5.4. The effect of the extrapolation is significant since* $L^\epsilon(m)$ *increases rapidly with* ϵ*; note that,*

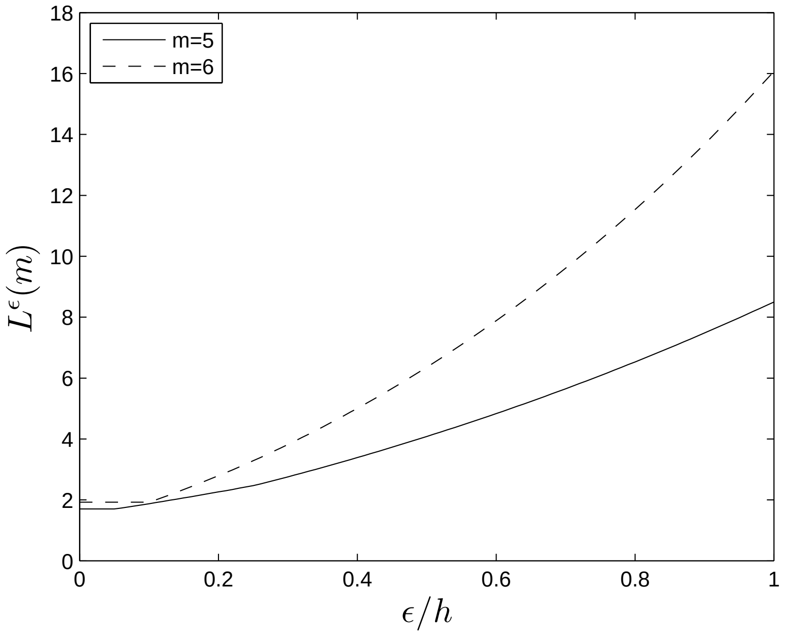


Figure 5.4: Coefficient $L^\epsilon(m)$ for the polynomial error bound in Equation (5.52) and calculated as a maximum over 30000 points (see Equation (5.13)).

amongst the cases considered in this section, and the effect is most significant for $L^\epsilon(9)$*. The figure lists only* $m = 4$ *and* $m = 5$ *since due to the stability considerations, examined in Chapter 10, only these values of* m *will be used within the FC-AD methodology. In our FC(Gram) algorithm,* $L^\epsilon(m)$ *is independent of the mesh and therefore a constant in the convergence analysis of the FC(Gram) continuation approximation. The value of* $S_0^\epsilon(m)$ *is determined in the same manner as* $S_0(m)$ *(see text below Equation (5.30)) except that the values in Table 5.3 are used instead of those in Table 5.2. Table 5.3 assumes the worst possible case* $\epsilon = h$*.*

5.3 Numerical Examples

Results for the continuation of the function $f(x) = e^{\sin(2.7\pi x)+\cos(\pi x)}$ are shown in Figure 5.5 for the FC(Gram) continuation just introduced, as well as Barycenter Chebyshev interpolation [135], and linear interpolation. The maximum error produced by each method is shown for a range of values of n. The error was calculated at $20n$ discrete data points. Note that the error of the Chebyshev interpolation decreases faster than any power of h, while the

	Error		*Error*
f^0_{even}	$5.6 \cdot 10^{-17}$	f^0_{odd}	$2.9 \cdot 10^{-15}$
f^1_{even}	$4.8 \cdot 10^{-14}$	f^1_{odd}	$5.6 \cdot 10^{-16}$
f^2_{even}	$6.7 \cdot 10^{-15}$	f^2_{odd}	$1.2 \cdot 10^{-13}$
f^3_{even}	$2.2 \cdot 10^{-12}$	f^3_{odd}	$2.3 \cdot 10^{-14}$
f^4_{even}	$4.8 \cdot 10^{-13}$	f^4_{odd}	$6.0 \cdot 10^{-12}$
f^5_{even}	$1.2 \cdot 10^{-10}$	f^5_{odd}	$2.2 \cdot 10^{-12}$

Table 5.3: Maximum errors over a slightly extended interval (evaluated as maxima over 2000 points in the set $[1-\Delta, 1+1/h] \cup [1+d-1/h, 1+d+\Delta]$) that result from the even and odd continuations of the Gram Polynomials for the parameter values $n_\Delta = 10$, $d/\Delta = 26/9$, $g = 63$, and $\Upsilon = 150$.

FC(Gram) used here is high-order but not spectrally accurate. We point out that, as discussed earlier in Chapter 3 however, the Chebyshev method requires data points arranged in a special fashion, which makes it unsuitable for solution of PDEs in general domains. If an FC(Gram) algorithm based on approximations by Gram polynomials of degree five at the left and right ends of the approximation interval is used (m=5), the FC(Gram) continuation cannot be better than sixth-order accurate. If a ninth-degree polynomial is used instead (m=9), then tenth-order convergence results. Figure 5.5 shows the results $m = 9$ on the top and the results with $m = 5$ on the bottom.

Another interesting test case is provided by the Runge function,

$$f(x) = \frac{1}{25(2x-1)^2}, \tag{5.54}$$

here scaled to the domain $x \in [0, 1]$. The Runge function is a classic example of the difficulties that can be encountered when a non-periodic function is interpolated over an evenly spaced discretization of points. Figure 5.6 shows the maximum interpolation error of the FC(Gram) approximation of the Runge function for $m = 9$ and $m = 5$ again compared with Barycenter Chebyshev interpolation, and linear interpolation.

In this case the superior performance of the continuation method is clearly demonstrated in the tenth-order case, while the oversampled version (sixth-order) also displays an excellent performance. Notice also that with nearly 1000 points, the limiting resolution of the FC(Gram) continuation is within approximately one order of magnitude from machine precision.

The error made in second derivative after approximating the Runge function with

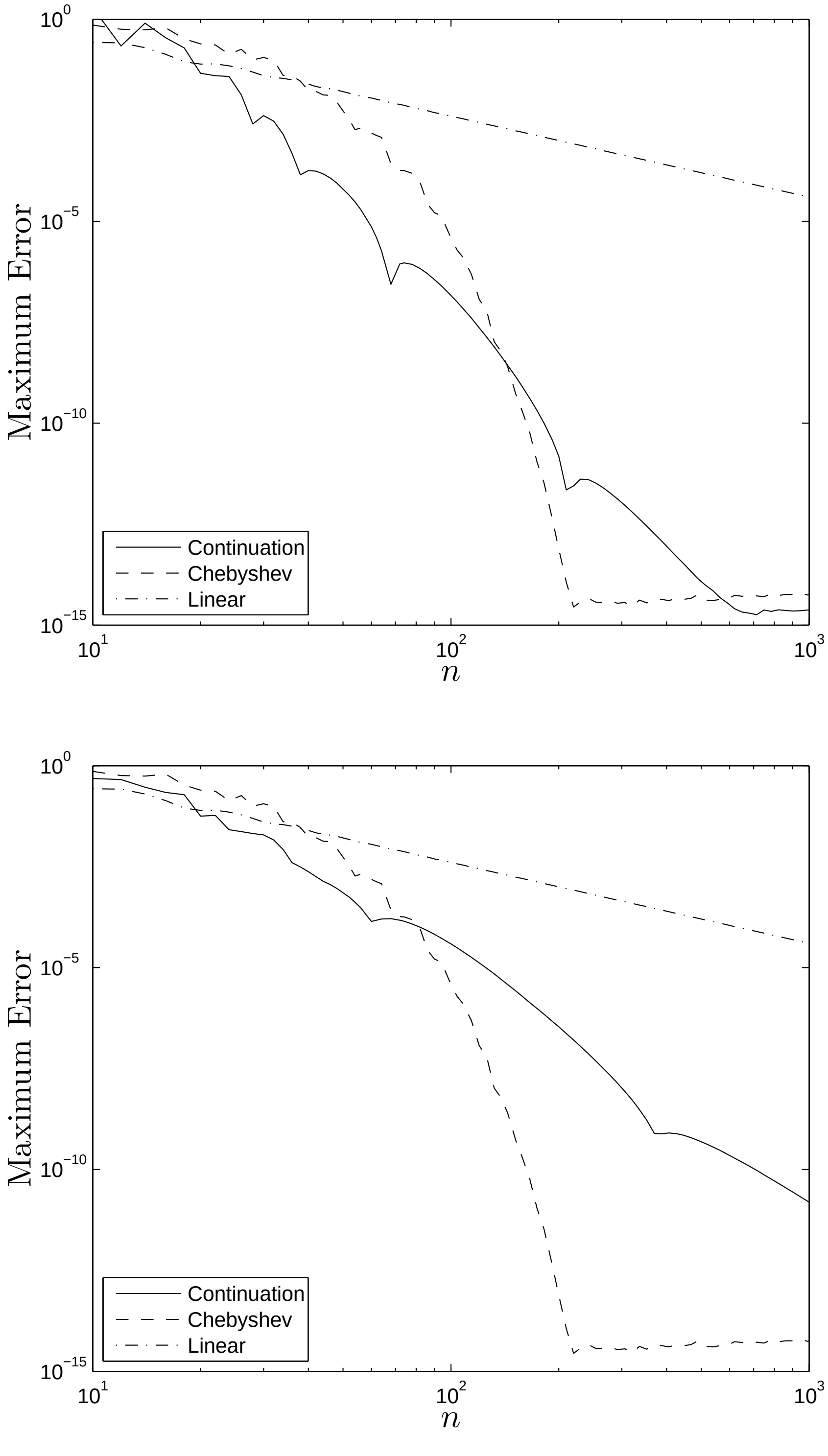
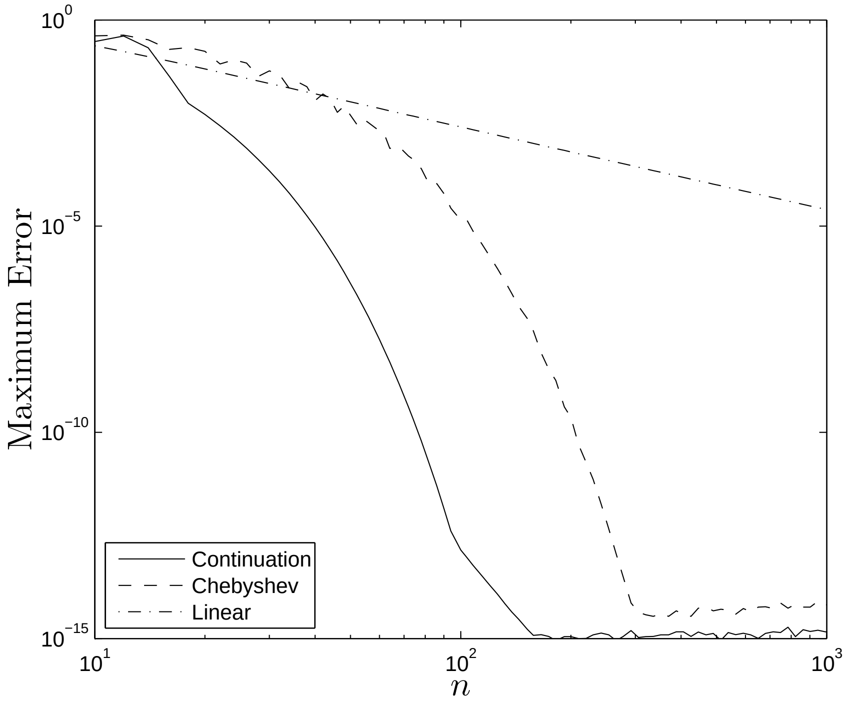


Figure 5.5: Maximum interpolation error for the function $f(x) = e^{\sin(2.7\pi x)+\cos(\pi x)}$ with a variety of interpolation schemes including a tenth-order accurate FC(Gram) continuation on top and a sixth-order version on the bottom.



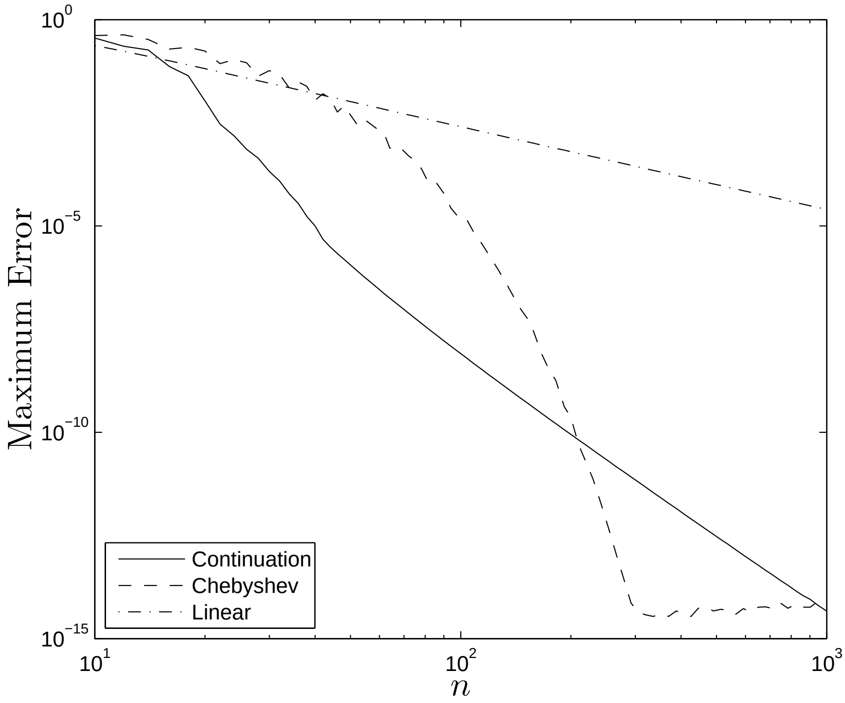


Figure 5.6: Maximum interpolation error for the function $f(x) = \frac{1}{25(2x-1)^2}$ with a variety of interpolation schemes, including a tenth-order accurate FC(Gram) continuation on top and a sixth-order version on the bottom.

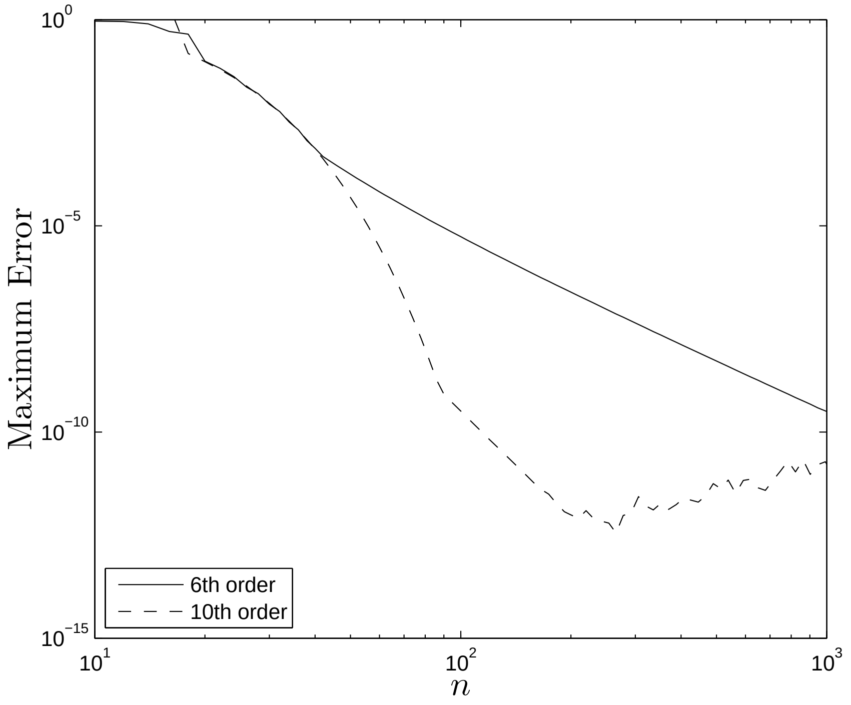


Figure 5.7: Maximum interpolation error of the second derivative for the function $f(x) = \frac{1}{25(2x-1)^2}$ with a sixth-order accurate FC(Gram) continuation and a tenth-order version.

FC(Gram) continuation of orders six and ten are shown in Figure 5.7. It shows initially spectral but ultimately fourth- and eight-order accurate approximations of the second derivative, respectively, in which cases the largest error occurs near the boundary. Note that the tenth-order scheme hits a limiting threshold and then the error grows as the mesh is refined, so that at 1000 points approximately five digits have been lost. For a traditional finite difference method, standard Fourier series, or even Chebyshev approximation one would expect to lose approximately six digits with as many points.

Chapter 6

FC-AD Algorithm for the Heat Equation

The high-order accuracy of the FC(Gram) continuation approximation was demonstrated in the last chapter. In this chapter, a certain FC-AD PDE solver is introduced. The FC-AD solver results as a combination of the FC(Gram) continuation method presented in the previous chapter with the classical Alternating Direction framework. To illustrate the methodology, here we present the FC-AD algorithm in its application as a solver for the Heat Equation. A splitting is developed that reduces the problem to one of solving ODEs over the simple 1D geometries shown in Figure 1.2. A detailed algorithm is presented which, using the FC(Gram) continuations to solve Alternating Direction ODEs produces a stable numerical scheme. A theoretical discussion of the stability properties of this algorithm is presented in Chapter 10. The unconditional stability and high-order accuracy of the FC-AD methodology is illustrated in this section through a variety of numerical results. Additional FC-AD algorithms for Elliptic and Hyperbolic problems in two and three dimensions are presented in Chapters 7 and 8, respectively.

6.1 Alternating Direction Splitting for the Heat Equation

To introduce the FC-AD algorithm we consider, the 2D Heat Equation with Dirichlet boundary data:

$$\begin{aligned}
&u_t = k(u_{xx} + u_{yy}) + Q(x,y,t), \quad (x,y,t) \in \quad \Omega \times (0,T], \\
&u(x,y,t) = G(x,y,t), \quad (x,y) \in \partial\Omega, \quad t \in (0,T], \\
&u(x,y,0) = u_0(x,y), \quad (x,y) \in \Omega, \\
&k \in \mathbb{R}, \quad k > 0,
\end{aligned} \tag{6.1}$$

where $\Omega \subset \mathbb{R}^2$ is a smoothly bounded domain, and where the functions Q, G, and u_0 are given smooth functions. Let $t^n = n\Delta t$ and further let u^n and $Q^{n+\frac{1}{2}}$ be equal to $u(x,y,t^n)$ and $Q(x,y,(n+1/2)\Delta t)$ respectively. Using a centered finite difference scheme to discretize the time derivative gives

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{k}{2}\frac{\partial^2}{\partial x^2}(u^{n+1} + u^n) + \frac{k}{2}\frac{\partial^2}{\partial y^2}(u^{n+1} + u^n) + Q^{n+\frac{1}{2}} + E_1(x,y,\Delta t), \tag{6.2}$$

where

$$\begin{aligned}
E_1(x,y,\Delta t) \quad \leq \quad & \frac{\Delta t^2}{24}\|u_{ttt}(x,y,t)\|_{L^\infty(\Omega\times(t^n,t^{n+1}))} \\
& + \frac{k\Delta t^2}{8}\|u_{ttxx}(x,y,t)\|_{L^\infty(\Omega\times(t^n,t^{n+1}))} \\
& + \frac{k\Delta t^2}{8}\|u_{ttyy}(x,y,t)\|_{L^\infty(\Omega\times(t^n,t^{n+1}))},
\end{aligned} \tag{6.3}$$

results from use of Taylor series expansions with the Cauchy form of the remainder. Collecting the terms for u^n and u^{n+1} in the above equations gives

$$\begin{aligned}
\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2} - \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)u^{n+1} \quad = \quad & \left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2} + \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)u^n \\
& + \Delta t Q^{n+\frac{1}{2}} + \Delta t E_1(x,y,t).
\end{aligned} \tag{6.4}$$

Factoring (6.4) into directional components yields

$$\begin{aligned}\left(1-\frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)\left(1-\frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)u^{n+1} &= \left(1+\frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)\left(1+\frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)u^{n} \\ &\quad +\frac{k^2\Delta t^2}{4}\frac{\partial^2}{\partial x^2}\frac{\partial^2}{\partial y^2}(u^{n+1}-u^n) \qquad (6.5)\\ &\quad +\Delta t Q^{n+\frac{1}{2}}+\Delta t E_1(x,y,t).\end{aligned}$$

Again using Taylor series with the Cauchy form of the remainder, we see that

$$E_2(x,y,t)=\frac{k^2\Delta t^2}{4}\frac{\partial^2}{\partial x^2}\frac{\partial^2}{\partial y^2}(u^{n+1}-u^n)\le\frac{k^2\Delta t^3}{4}\|u_{txxyy}(x,y,t)\|_{L^\infty(\Omega\times(t^n,t^{n+1}))}; \qquad (6.6)$$

note that $E_2(x,y,t)$ is of order $O(\Delta t^3)$. In order to solve for u^{n+1} in Equation (6.5), operators of the form $\left(1-\frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)$ and $\left(1-\frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)$ must be inverted. The application of the inverse operators on a function f gives as a result the solution of the one-dimensional boundary value problem

$$-\alpha^2u''+u=f, \quad u(a)=B_\ell, \quad u(b)=B_r, \qquad (6.7)$$

with

$$\alpha^2=\frac{k\Delta t}{2}, \qquad (6.8)$$

for given boundary values B_ℓ and B_r, prescribed at locations a and b respectively (see Remark 6.1.1). The points a and b are the intersection of the line over which the ODE is to be solved with the domain boundary.

Definition 6.1.1. *For notational convenience, we define*

$$\left(1-\frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)^{-1}=\left(1-\frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)^{-1}_{a,b;B_\ell,B_r} \qquad (6.9)$$

and

$$\left(1-\frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)^{-1}=\left(1-\frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)^{-1}_{a,b;B_\ell,B_r} \qquad (6.10)$$

as the operators that produce the solution u of the boundary value problem (6.7) *from the right hand side f with boundary values B_ℓ and B_r and boundary locations a and b.*

A numerical algorithm (based on the FC(Gram) continuation method presented in the previous chapter) for the solution of ODEs such as Equation (6.7) is presented in Section 6.3.

In the special case $Q = 0$, Peaceman and Rachford obtained an approximation, $\tilde{u}^{n+1}$, to u^n, the solution to Equation (6.5), using a scheme of the form

$$\begin{aligned}\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)\tilde{u}^{n+\frac{1}{2}} &= \left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)\tilde{u}^{n}\\ \left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)\tilde{u}^{n+1} &= \left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)\tilde{u}^{n+\frac{1}{2}},\end{aligned}$$

(see also [125]). In order to account for the presence of the inhomogeneity Q, we use instead

$$\left(1 - \tfrac{k\Delta t}{2}\tfrac{\partial^2}{\partial x^2}\right)\tilde{u}^{n+\frac{1}{2}} = \left(1 + \tfrac{k\Delta t}{2}\tfrac{\partial^2}{\partial y^2}\right)\tilde{u}^{n} + \tfrac{\Delta t}{2}Q^{n+\frac{1}{4}} \tag{6.11}$$

$$\left(1 - \tfrac{k\Delta t}{2}\tfrac{\partial^2}{\partial y^2}\right)\tilde{u}^{n+1} = \left(1 + \tfrac{k\Delta t}{2}\tfrac{\partial^2}{\partial x^2}\right)\tilde{u}^{n+\frac{1}{2}} + \tfrac{\Delta t}{2}Q^{n+\frac{3}{4}}, \tag{6.12}$$

where $Q^{n+\frac{1}{4}}$ and $Q^{n+\frac{3}{4}}$ are equal to $Q(x, y, (n{+}1/4)\Delta t)$ and $Q(x, y, (n{+}3/4)\Delta t)$ respectively.

It is easy to check that a solution of Equations (6.11) and (6.12) provide an approximate solution of Equation (6.5). To show this we multiply Equation (6.11) by $\left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)$ and Equation (6.12) by $\left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)$—noting that the operators $\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)$ and $\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)$ commute and subtracting the two resulting equations, we obtain

$$\begin{aligned}\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)\tilde{u}^{n+1} \quad = \quad & \left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)\left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)\tilde{u}^{n}\\ & + \frac{\Delta t}{2}\left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)Q^{n+\frac{1}{4}}\\ & + \frac{\Delta t}{2}\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)Q^{n+\frac{3}{4}}.\end{aligned} \tag{6.13}$$

We now introduce the splitting

$$Q^{n+\frac{1}{2}} = \frac{1}{2}\left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)Q^{n+\frac{1}{4}} + \frac{1}{2}\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)Q^{n+\frac{3}{4}} + E_3(x, y, t), \tag{6.14}$$

where, applying a Taylor series expansion with the Cauchy form of the remainder to the function Q we obtain the bound

$$E_3(x, y, t) \leq \frac{\Delta t^2}{16}\|Q_{tt}(x, y, t)\|_{L^\infty(\Omega\times(t^n,t^{n+1}))} + \frac{k\Delta t^2}{8}\|Q_{txx}(x, y, t)\|_{L^\infty(\Omega\times(t^n,t^{n+1}))}. \tag{6.15}$$

It follows that, as claimed above, a solution of Equations (6.11) and (6.12) is also a solution to Equation (6.5) with error equal to

$$\Delta t E_1(x,y,t) + E_2(x,y,t) + \Delta t E_3(x,y,t). \tag{6.16}$$

Boundary values for $\tilde{u}^{n+\frac{1}{2}}$ in Equation (6.11) and for $\tilde{u}^{n+1}$ in Equation (6.12) are all that are needed in order to complete the scheme.

Remark 6.1.1. *The boundary values for $\tilde{u}^{n+1}$ are simply given by the value $G(x,y,t^{n+1})$ for the appropriate boundary points $(x,y) \in \partial\Omega$. The boundary values of $\tilde{u}^{n+\frac{1}{2}}$ for the traditional ADI on a square domain are given by*

$$\begin{aligned} \tilde{u}^{n+\frac{1}{2}}(x,y) \;\; = \;\; & \frac{1}{2}(1+\frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2})G(x,y,t^n) + \frac{1}{2}(1-\frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2})G(x,y,t^{n+1}) \\ & + \frac{\Delta t}{4}(Q(x,y,t^{n+\frac{1}{4}}) - Q(x,y,t^{n+\frac{3}{4}})), \quad (x,y) \in \partial\Omega, \end{aligned} \tag{6.17}$$

a relation that follows directly from Equations (6.11) and (6.12), *c.f., [125]. In the case of a complex domain, the spatial derivatives of G are not known a priori. In order to handle complex domains, the FC-AD algorithm uses the boundary values*

$$\tilde{u}^{n+\frac{1}{2}}(x,y) = G(x,y,(n+1/2)\Delta t). \tag{6.18}$$

This approximation introduces an additional time discretization error E_4 that satisfies

$$\begin{aligned} E_4(x,y,t) \;\; \leq \;\; & \frac{k\Delta t^2}{4}\|u_{tyy}(x,y,t)\|_{L^\infty(\Omega\times(t^n,t^{n+1}))} + \frac{\Delta t^2}{4}\|u_{tt}(x,y,t)\|_{L^\infty(\Omega\times(t^n,t^{n+1}))} \\ & + \frac{\Delta t^2}{8}\|Q_t(x,y,t)\|_{L^\infty(\Omega\times(t^n,t^{n+1}))}. \end{aligned} \tag{6.19}$$

Accounting for the time discretization errors (6.16) *and the error* (6.18) *on the boundary values, the overall error arising from one step of our Heat-Equation FC-AD algorithm is of order $O(\Delta t^2)$. This bound thus predicts an overall FC-AD error of $O(\Delta t)$ Despite this argument, our numerical experiments, shown in Section 6.5, indicate that in practice the overall accuracy of the algorithm remains $O(\Delta t^2)$. In any case, Richardson Extrapolation can also be employed to increase the accuracy in time as has been shown in reference [65] and which we demonstrate in Chapter 8 for the Wave Equation.*

For both algorithmic and analysis reasons it is convenient to introduce the notation

$$w^n = \left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)\tilde{u}^n, \tag{6.20}$$

and

$$w^{n+\frac{1}{2}} = \left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)\tilde{u}^{n+\frac{1}{2}}. \tag{6.21}$$

In terms of these new variables and the notation of Definition 6.1.1, the algorithm becomes

$$\begin{aligned} w^{n+\frac{1}{2}} &= \left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)^{-1}\left(w^n + \frac{\Delta t}{2}Q^{n+\frac{1}{4}}\right) \\ w^{n+1} &= \left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)^{-1}\left(w^{n+\frac{1}{2}} + \frac{\Delta t}{2}Q^{n+\frac{3}{4}}\right). \end{aligned} \tag{6.22}$$

The operators $\left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial x^2}\right)^{-1}$ and $\left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)\left(1 - \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right)^{-1}$ acting on a function f give the equations

$$\begin{aligned} &-\alpha^2 u''(x) + u(x) = f(x), \quad u(a) = B_\ell, \quad u(b) = B_r, \\ &\alpha^2 u''(x) + u(x) = q(x), \end{aligned} \tag{6.23}$$

for some function q where again $\alpha^2 = \frac{k\Delta t}{2}$. By adding the two equations in (6.23), it is observed that

$$q(x) = 2u(x) - f(x), \tag{6.24}$$

where $u(x)$ is the solution to Equation (6.7), and therefore each half-step of (6.22) only requires one to solve Equation (6.7) and then simply apply Equation (6.24).

6.2 Mesh Structure

The discrete mesh for the FC-AD methodology has already been introduced in Chapter 1 (in particular see Figures 1.1 and 1.2). It involves simply overlaying the geometry with a Cartesian mesh and using, in addition, all boundary points (denoted by x_ℓ and x_r in Figure 6.1) lying at the intersection of the domain boundary with all Cartesian horizontal and vertical lines. Note that on a single line (see Figure 6.1), n interior points are positioned in an arbitrary manner with respect to x_ℓ and x_r, with the only limitation that $x_1 - x_\ell$ and

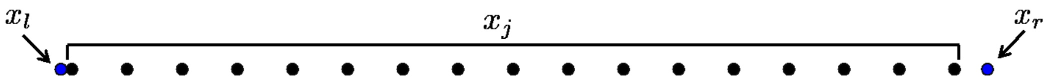


Figure 6.1: Sample discretization for the ODEs resulting from an alternating direction splitting of a PDE: n discretization points x_j are shown in addition to the boundary points x_ℓ and x_r,

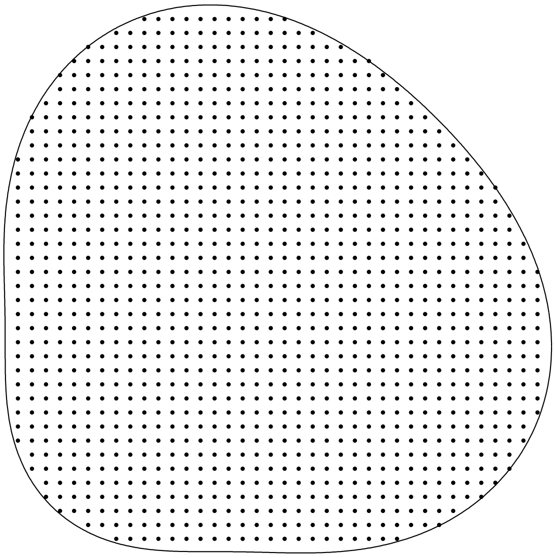


Figure 6.2: Spatial PDE domain Ω, showing the spatial discretization $\mathcal{D}_\Omega$.

$x_r - x_n$ are both less than or equal to h, where h is the spacing of the interior points.

In order to facilitate reference to the grid points in the interior of the domain, we introduce some additional notation. Let the bounded open set Ω (which is the domain of the PDE, see Figure 6.2) be contained in $[a_x, b_x] \times [a_y, b_y]$. Let (x_i, y_j) be a point in a Cartesian mesh over the rectangle $[a_x, b_x] \times [a_y, b_y]$. Then we define the interior mesh points $\mathcal{D}_\Omega$ as the set

$$\mathcal{D}_\Omega = \{(x_i, y_j) \in [a_x, b_x] \times [a_y, b_y] : (x_i, y_j) \in \Omega\}\,. \tag{6.25}$$

For a given point $(x_i, y_j) \in \mathcal{D}_\Omega$, let n_{x_i} be the number of points in $\mathcal{D}_\Omega$ on the same vertical line as (x_i, y_j), and let n_{y_j} be the corresponding number of points in $\mathcal{D}_\Omega$ on the same horizontal line.

We introduce some spaces associated with these meshes, which will be useful in the

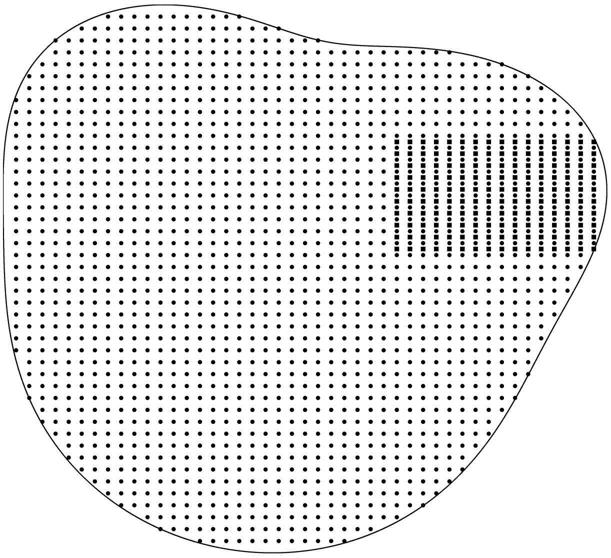


Figure 6.3: Sample geometry and discretization for our FC-AD algorithm demonstrating a local refinement designed to maintain high-order accuracy near portions of the boundary for which the sampling is not sufficiently fine.

stability discussion presented in Chapter 10. We define the vector space $\ell_2(n) = \mathbb{R}^n$ with the Euclidean norm; in our constructions n may be either n_{x_i} or n_{y_j} for some x_i or y_j. Also letting N_Ω be the total number of points in $\mathcal{D}_\Omega$, we define the space $\ell_2(\mathcal{D}_\Omega) = \mathbb{R}^{D_\Omega}$; that is, the set of all functions from D_Ω to $\mathbb{R}$ with the usual Euclidean norm: for a given $\theta \in \ell_2(\mathcal{D}_\Omega)$, $\theta_{ij} = \theta(x_i, y_j)$ the norm of θ is given by

$$\|\theta\|_{\ell_2(\mathcal{D}_\Omega)} = \sqrt{\sum_{(x_i,y_j)\in\mathcal{D}_\Omega} \frac{\theta_{ij}^2}{N_\Omega}}. \tag{6.26}$$

The mesh structure and definitions above will be used for all PDEs considered in this thesis.

For some domains Ω, the mesh lines may cross the boundary more than twice (see Figure 8.1). In such a case, each segment of data is treated separately. Additionally, consider the geometry in Figure 6.3 with the evenly spaced discrete Cartesian mesh shown. Lines of data are available in this geometry in both spatial directions over which the ODEs (6.22) may be solved. But it may occur that an insufficient number of points are available for high-order accuracy in the FC(Gram) continuation along a given line (see Remark 5.1.4). In this case, local refinement may be used, as is shown on the right portion of Figure 6.3. This simply involves refining the mesh in a single spatial direction, but that refinement must be extended into the mesh some distance so that high-order FC(Gram) approximations can be obtained along all lines of data in the mesh (once again, see Remark 5.1.4). The boundary conditions needed for the fine scale mesh in the interior of Ω may be interpolated from the results obtained on the coarser mesh. This local refinement strategy should also be applicable to capture fine details in the solution in an adaptive framework as well. Clearly a more detailed study of mesh refinement for the FC-AD and its potential uses would be beneficial, but results obtained thus far have maintained the accuracy and stability of the overall approach. In the next chapter we present an example in which this type of mesh refinement was used.

6.3 FC-ODE Algorithm: Solution of ODEs by Means of Fourier Continuation

In view of the discussion presented in the preceding sections, the last remaining steps required to implement the FC-AD for the Heat Equation is to obtain an accurate approximation of the solution of ODEs of the form

$$-\alpha^2 u''(x) + u(x) = f(x), \quad u(x_\ell) = B_\ell, \quad u(x_r) = B_r, \tag{6.27}$$

and, using that solution, to determine a corresponding approximation for the function q in Equation (6.24). It is not difficult to obtain such high-order approximations for u and q; a number of additional steps must be followed, however, to insure the stability of the associated FC-AD algorithm—as described below in this section, and studied in detail in Chapter 10. The first step in our approximate solution of Equation (6.27) approximates f by a continuation Fourier series f^c

$$f^c(x) = \sum_{k \in t(n+n_d-2)} a_k e^{2\pi i \frac{xk}{(x_r - x_\ell)(1+d)}} \tag{6.28}$$

as discussed the previous chapter. A particular solution to

$$-\alpha^2 v''(x) + v(x) = f^c(x), \quad v(x_\ell) = B_\ell, \quad v(x_r) = B_r, \tag{6.29}$$

is easily obtained: including the appropriate solution of the associated homogeneous problem, the solution v of (6.29) is given by

$$v(x) = \sum_{k \in t(n+n_d-2)} \frac{a_k}{1 + \frac{4\alpha^2\pi^2k^2}{(x_r-x_\ell)^2(1+d)^2}} e^{2\pi i \frac{xk}{(x_r-x_\ell)(1+d)}} + c_1 h_1(x) + c_2 h_2(x), \tag{6.30}$$

where c_1 and c_2 are constants chosen to fit the boundary conditions in (6.27) and where the homogeneous solutions to (6.27) are given by,

$$h_1(x) = e^{x/|\alpha|} \quad \text{and} \quad h_2(x) = e^{-x/|\alpha|}. \tag{6.31}$$

As shown in Section 6.1, in the context of the FC-AD algorithm we have $\alpha^2 = k\Delta t/2$: $\alpha \to 0$ as $\Delta t \to 0$. However, we note that as $\alpha \to 0$, $v(x)$ does not converge to $u(x)$ as it would be the case were one to use Finite Difference Approximations; instead

$$u(x) - v(x) \to f(x) - f^c(x) \neq 0 \quad \text{as} \quad \alpha \to 0 \quad \text{for } x \text{ away from } x_\ell \text{ and } x_r. \tag{6.32}$$

Remark 6.3.1. *In view of the lack of convergence to zero as shown in Equation* (6.32)*, use of a scheme such as* (6.29) *as part of our FC-AD algorithm would result in certain types of conditional convergence—e.g., lack of convergence as* Δt *tends to 0 much faster than* h *tends to zero. In order to eliminate this difficulty we introduce corrections ensuring that the corrected solution* u_j *satisfies*

$$u(x_j) - u_j \to 0 \quad \textit{as} \quad \alpha \to 0 \quad \textit{for all } x. \tag{6.33}$$

A simple correction that restores the aforementioned convergence to zero as $\alpha \to 0$, (and, additionally, gives rise to *high-order accuracy*) is obtained by means of *low-order* accurate Finite Difference techniques—as indicated in what follows. The correction η_j is obtained as the solution of the equation

$$-\alpha^2\frac{\eta_{j+1}-2\eta_j+\eta_{j-1}}{h^2}+\eta_j=f(x_j)-f^c(x_j), \tag{6.34}$$

with boundary conditions $\eta_0 = 0$, $\eta_{n_\Delta+1} = 0$, $\eta_{n-n_\Delta} = 0$, and $\eta_{n+1} = 0$ (A banded LU Decomposition allows η to be calculated in $O(n_\Delta)$ operations (c.f., [110])). Clearly, the alternate choice of $\eta_j = f(x_j) - f^c(x_j)$, which would suffice for convergence to 0, lacks a connection to the actual ODE and would result in a lack of convergence to solutions of the PDE as the spatial discretization tends to zero. Notice that the values η_j are essentially "all error" (they arise from the error $f(x_j) - f^c(x_j)$ on the right hand side of Equation (6.34)) that we are required to keep to ensure convergence of the FC-AD algorithm as the spatial and temporal mesh sizes converge to zero independently). It is easy to check that the corrected solution $v(x_j) + \eta_j$ satisfies $u(x_j) - (v(x_j) + \eta_j) \to 0$ as $\alpha \to 0$, as required.

Additional steps are needed in order to ensure the stability of the full FC-AD algorithm. The steps we list later in this section, which where obtained through a significant amount of experimentation, involve a series of orthogonal projections that, as is established in Section 10, give rise to a stable and high-order accurate numerical scheme. To present the stability restoring steps we introduce the following two important definitions:

Definition 6.3.1. *For a given function f defined in $[x_\ell, x_r]$, the open boundary projection f^p of f is defined as follows: an orthonormal Gram polynomial [14] basis $\{P_s^{o,1}(x)\}_{s=0}^m$ is formed by polynomials of degree $\le m$ that are orthogonal with respect to the discrete scalar product $(f,g)_{o,\ell} = \sum f(x_j)g(x_j)$ over the set of points*

$$\{x_j, j = 1, \ldots, n_\Delta\}. \tag{6.35}$$

An additional orthonormal Gram polynomial basis $\{P_s^{o,2}\}_{s=0}^m$ is formed orthogonal with respect to the discrete scalar product $(f,g)_{o,r} = \sum f(x_j)g(x_j)$ over the set of points

$$\{x_j, j = n - n_\Delta + 1, \ldots, n\}. \tag{6.36}$$

The function f^p is then defined by

$$f^p(x) = \begin{cases} \sum_{s=0}^m (f, P_s^{o,1})_{o,\ell} P_s^{o,1}(x) \quad \textit{for} \quad x \in [x_\ell, x_{n_\Delta}] \\ f(x) \quad \textit{for} \quad x \in (x_{n_\Delta}, x_{n-n_\Delta+1}) \\ \sum_{s=0}^m (f, P_s^{o,2})_{o,r} P_s^{o,2}(x) \quad \textit{for} \quad x \in [x_{n-n_\Delta+1}, x_r]. \end{cases} \tag{6.37}$$

Clearly, on the boundary segment $[x_\ell, x_{n_\Delta}]$ (resp. $[x_{n-n_\Delta+1}, x_r]$), f^p equals the projection of f according to the scalar product $(\cdot,\cdot)_{o,\ell}$ (resp. $(\cdot,\cdot)_{o,r}$).

Definition 6.3.2. *For a given function f defined in $[x_\ell, x_r]$, the closed boundary projection f^b of f is defined as follows: an orthonormal polynomial basis $\{P_s^{c,1}(x)\}_{s=0}^m$ is formed by polynomials of degree $\leq m$ that are orthogonal with respect to the discrete scalar product $(f,g)_{c,\ell} = \sum f(x_j)g(x_j)$ over the set of points*

$$\{x_\ell\} \cup \{x_j, j = 1, \dots, n_\Delta\}.$$

An additional orthonormal polynomial basis $\{P_s^{c,2}\}_{s=0}^m$ is formed orthogonal with respect to the discrete scalar product $(f,g)_{c,r} = \sum f(x_j)g(x_j)$ over the set of points

$$\{x_j, j = n - n_\Delta + 1, \dots, n\} \cup \{x_r\}.$$

The function f^b is then defined by

$$f^b(x) = \begin{cases} \sum_{s=0}^m (f, P_s^{c,1})_{c,\ell} P_s^{c,1}(x) \quad \textit{for} \quad x \in [x_\ell, x_{n_\Delta}] \\ f(x) \quad \textit{for} \quad x \in (x_{n_\Delta}, x_{n-n_\Delta+1}) \\ \sum_{s=0}^m (f, P_s^{c,2})_{c,r} P_s^{c,2}(x) \quad \textit{for} \quad x \in [x_{n-n_\Delta+1}, x_r]. \end{cases} \tag{6.38}$$

Clearly, on the boundary segment $[x_\ell, x_{n_\Delta}]$ (resp. $[x_{n-n_\Delta+1}, x_r]$), f^b equals the projection of f according to the scalar product $(\cdot,\cdot)_{c,\ell}$ (resp. $(\cdot,\cdot)_{c,r}$).

We are now ready to introduce our final, consistent and stability-corrected solution u_j mentioned earlier in this section and presented in point 4. below. To obtain u_j we proceed as follows:

1. Construct the open boundary projection v^p of v according to Definition 6.3.1.

2. Construct the closed boundary projection v^b of v according to Definition 6.3.2.

3. Project the calculated finite difference solution η_j into the open Gram polynomial basis (see Defintion 6.3.1) to obtain the projection

$$\eta_j^p = \begin{cases} \sum_{s=0}^{m}(\eta, P_s^{o,1})_{o,\ell} P_s^{o,1}(x_j) \quad \text{for} \quad j = 1, \dots, n_\Delta \\ 0 \quad \text{for} \quad j = n_\Delta + 1, \dots, n - n_\Delta \\ \sum_{s=0}^{m}(\eta, P_s^{o,2})_{o,r} P_s^{o,2}(x_j) \quad \text{for} \quad j = n - n_\Delta + 1, \dots, n, \end{cases} \tag{6.39}$$

where $P_s^{o,1}$ and $P_s^{o,2}$ are as defined in Definition 6.3.1 and the inner products notations $(\eta, g)_{o,\ell}$ and $(\eta, g)_{o,r}$ are slight modifications of those used in Definition 6.3.1: here these symbols are taken to denote $\sum \eta_j g(x_j)$ over the sets of points (6.35) or (6.36) as appropriate.

Our stability corrected solution u_j, presented in step 4. below, results as a combination of the projections mentioned in steps 1. through 3. The main goal leading to the specific combination of projections we use is to preserve, at the same time, the validity of (6.33) and the stability of the algorithm; we obtained our final expression for the solution u_j by seeking to achieve such dual goal through a combination of heuristic arguments and a degree of experimentation. In particular we found that use of the closed boundary projection v^b gives rise to stability but, owing to the inclusion of the boundary points x_ℓ and x_r in the projection scheme, $v^b(x_j)$ actually does not tend to $v(x_j)$ as $\alpha \to 0$: use of this approach would not give rise to a fully convergent overall PDE solver. While $v^p(x_j)$ does tend to $v(x_j)$ as $\alpha \to 0$, it happens not to give rise to stable numerics when used in conjunction with the FC-AD methodology. We found that a linear combination in terms of a variable coefficient $\chi = \chi(\alpha, h)$

$$\chi = \min(25\alpha^2/h^2, 1) \tag{6.40}$$

induces stability while giving rise to convergence of the boundary values. The specific form (6.40) and in particular, the constant (25) we use, were obtained through experimentation, and was found to insure unconditional stability as the resulting ODE solver is used as part of the overall FC-AD method. We note that both v^p and v^b are high-order accurate approximations of v and therefore so is their linear combination. The projection in Step 3. was similarly undertaken for stability considerations and since η_j is an error term, this projection has no significant effect on the overall error. Thus:

4. The solution u_j is constructed as

$$u_j = \eta_j - \eta_j^p + (1-\chi)v^p(x_j) + \chi v^b(x_j). \tag{6.41}$$

Once the approximate solution u_j to the differential equation (6.27) has been obtained, our approximation g_j to the function q is then given by

$$q_j \approx 2u_j - f(x_j). \tag{6.42}$$

All the elements are now in place for the implementation of our stable and high-order accurate FC-AD methodology for the Heat Equation; elliptic and hyperbolic PDEs are considered, in turn, in Chapters 7 and 8. An analysis of the error arising from this correction and all other approximations is given in Section 6.4

6.4 Accuracy of the Solution to ODEs

A theoretical study of the accuracy of the FC(Gram) approximation to a smooth function was presented in Section 5.2 and a bound was given in Equation (5.45) for the corresponding approximation error, including the error arising from "accurate extrapolation slightly beyond the interior mesh", as discussed in Remark 5.2.3. The expected $O(h^{m+1})$ convergence was also demonstrated with a variety of numerical results in Section 5.3.

In the previous portion of Chapter 6, in turn, we introduced a method, outlined in Section 6.3, which, based on the FC(Gram) approach, can be used to produce numerical solutions u_j for ODEs of the type

$$-\alpha^2 u''(x) + u(x) = f(x), \quad u(x_\ell) = B_\ell, \quad u(x_r) = B_r, \tag{6.43}$$

that arise in connection with Alternating Direction PDE solvers. In the present section we present a study of the error that results as these methods are used to produce such numerical ODE solutions; an additional portion of our theoretical discussion is presented in Chapter 10 where certain singular-values that relate to the stability of the overall FC-AD approach are studied.

The error analysis presented in this section results from an application of the maximum principle [59, 98]. Let $f \in C^k[x_\ell, x_r]$ and $u \in C^{k+2}[x_\ell, x_r]$ for sufficiently large k, and let f^c be the FC(Gram) approximation to f. Further, call $A(f,h)$ the error implicit in a FC(Gram) continuation:

$$A(f,h) = \|f(x) - f^c(x)\|_{L^\infty[x_\ell,x_r]}. \tag{6.44}$$

In order to determine the error of the approximate solution u_j given by Equation (6.41) we consider the Green's function associated with the differential operator in Equation (6.43). For clarity, and without loss of generality, in this section we let $x_\ell = 0$ and $x_r = 1$ so that the Green's function for zero Dirichlet data is given by

$$G(x,\tilde{x}) = \begin{cases} \frac{2e^{1/|\alpha|}}{|\alpha|(e^{2/|\alpha|}-1)} \sinh\left(\frac{\tilde{x}}{|\alpha|}\right) \sinh\left(\frac{1-x}{|\alpha|}\right) & \text{for} \quad \tilde{x} < x \\ \frac{2e^{1/|\alpha|}}{|\alpha|(e^{2/|\alpha|}-1)} \sinh\left(\frac{1-\tilde{x}}{|\alpha|}\right) \sinh\left(\frac{x}{|\alpha|}\right) & \text{for} \quad x < \tilde{x}. \end{cases} \tag{6.45}$$

Since the function v (defined in Equation (6.30)) is the exact solution of

$$-\alpha^2 v''(x) + v(x) = f^c(x), \quad v(x_\ell) = B_\ell, \quad v(x_r) = B_r, \tag{6.46}$$

it follows that

$$u(x) - v(x) = \int_0^1 \left(f(\tilde{x}) - f^c(\tilde{x})\right) G(x,\tilde{x}) d\tilde{x}. \tag{6.47}$$

Clearly this quantity is bounded by

$$\|u(x) - v(x)\|_{L^\infty[0,1]} \leq A(f,h) \int_0^1 |G(x,\tilde{x})| \, d\tilde{x}. \tag{6.48}$$

Noting that $G(x,\tilde{x}) \geq 0$ for $x, \tilde{x} \in [0,1]$ and that, further,

$$0 \leq \int_0^1 G(x,\tilde{x}) d\tilde{x} = 1 - \frac{\sinh\left(\frac{1-x}{|\alpha|}\right) + \sinh\left(\frac{x}{|\alpha|}\right)}{\sinh\left(\frac{1}{|\alpha|}\right)} \leq 1, \tag{6.49}$$

we obtain

$$\|u(x) - v(x)\|_{L^\infty[0,1]} \leq A(f,h). \tag{6.50}$$

Thus the error in the ODE solution is bounded by the maximum error in the FC approximation of the right-hand-side.

In order to correct for consistency in the algorithm, certain discrete values η_j were introduced as the finite difference solution to the ODE (6.34) in Section 6.3. From the definition of $A(f,h)$ (Equation (6.44)), we have

$$|\eta_j| \leq |f(x_j) - f^c(x_j)| \leq A(f,h) \quad \text{for} \quad j = 1, \ldots, n, \tag{6.51}$$

where the first inequality follows directly from the discrete maximum principle (see [98]), and therefore

$$\max_{j=0,\ldots,n} |u(x) - v(x) - \eta_j| \leq 2A(f,h). \tag{6.52}$$

We now consider the error resulting from the projections introduced in Section 6.3, which, as will be discussed in Chapter 10, relate to stability issues. Let u^p and u^b be the open and closed boundary projections of the function u according to the Definitions 6.3.1 and 6.3.2 respectively. The approximation error in these projections can be bounded as in Chapter 5. Recalling the definitions of ϵ and Δ_ϵ from Equation (5.51), we have

$$\|u(x) - u^p(x)\|_{L_\infty[0,1]} \leq (1 + L^\epsilon(m)) \frac{(\Delta^\epsilon)^{m+1} M_u}{2^{2m+1}(m+1)!}, \tag{6.53}$$

and

$$\|u(x) - u^b(x)\|_{L_\infty[0,1]} \leq (1 + L_B^\epsilon(m)) \frac{(\Delta^\epsilon)^{m+1} M_u}{2^{2m+1}(m+1)!}, \tag{6.54}$$

where M_u is the maximum of the $(m+1)$th derivative of u over the set $[0, x_{n_\Delta}] \cup [x_{n-n_\Delta+1}, 1]$. The new constant $L_B^\epsilon(m)$ in Equation (6.54) is a quantity analogous to $L^\epsilon(m)$ which uses a set of discrete sampling points that includes the relevant boundary point ($u(x_\ell)$ or $u(x_r)$) in the boundary projection (see Definition 6.3.2). (Note that, by symmetry, the same value $L_B^\epsilon(m)$ applies to both possible configurations, namely, a) adding a point x_ℓ or b) adding a point x_r.) All the possible values of the constant $L_B^\epsilon(m)$ are plotted in Figure 6.4; compare the corresponding values of $L^\epsilon(m)$ shown in Figure 5.4.

In order to bound the magnitude of the values $\eta_j - \eta_j^p$ (calculated as part of the stability corrected solution u_j where the projection is η_j^p is defined in Equation (6.39)) we use the relation

$$\left(\sum_j y_j^2\right)^{1/2} \leq \sqrt{N} \max_j |y_j| \leq \sqrt{N} \left(\sum_j y_j^2\right)^{1/2} : \tag{6.55}$$

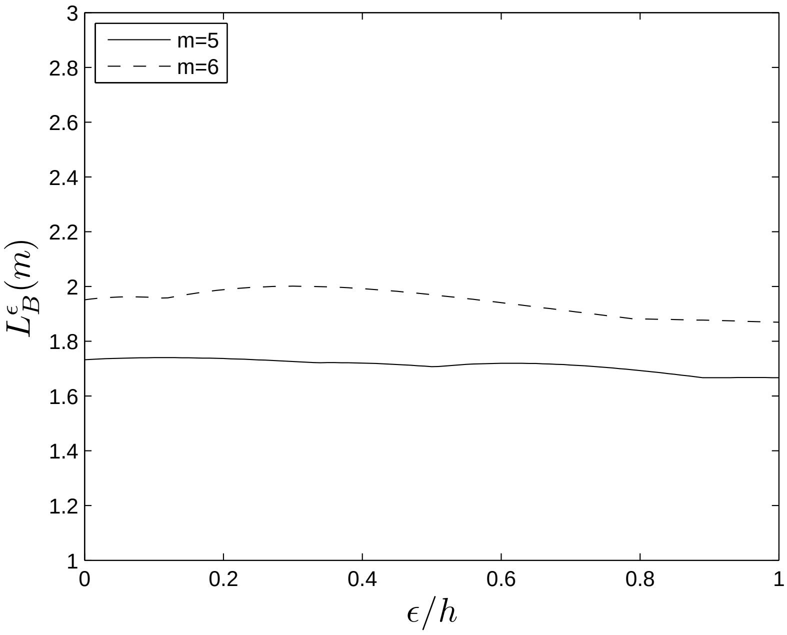


Figure 6.4: Coefficient $L_B^\epsilon(m)$ for the polynomial error bound in Equation (6.54), calculated as a maximum over 30000 points (see Equation (5.13)).

It follows that $\eta_j - \eta_j^p$ is bounded by

$$\max_{j=1,\ldots,n} |\eta_j - \eta_j^p| \leq \sqrt{n_\Delta} A(f,h), \tag{6.56}$$

since η_j^p is calculated by an orthogonal projection for which

$$\left(\sum_{j=1}^{n_\Delta} \left(\eta_j - \eta_j^p \right)^2 \right)^{1/2} \leq \left(\sum_{j=1}^{n_\Delta} (\eta_j)^2 \right)^{1/2} \tag{6.57}$$

and

$$\left(\sum_{j=n-n_\Delta+1}^{n} \left(\eta_j - \eta_j^p \right)^2 \right)^{1/2} \leq \left(\sum_{j=n-n_\Delta+1}^{n} (\eta_j)^2 \right)^{1/2} \tag{6.58}$$

hold. (Note from Section 6.3 that $\eta_j = \eta_j^p = 0$ for $j = n_\Delta + 1, \ldots, n - n_\Delta$). The error in the approximation u_j, defined in Equation (6.41) as

$$u_j = \eta_j - \eta_j^p + (1-\chi) v^p(x_j) + \chi v^b(x_j), \tag{6.59}$$

can then be bounded by combining the previous bounds. With some algebra using Equations (6.50), (6.53), (6.54), and (6.56) we obtain

$$\max_{j=1,\ldots,n} |u(x_j) - u_j| \leq (\sqrt{n_\Delta + 1} + 2\sqrt{n_\Delta} + 1)A(f,h) \tag{6.60}$$

$$+(1 + L^\epsilon(m) + L^\epsilon_B(m))\frac{(\Delta^\epsilon)^{m+1} M_u}{2^{2m+1}(m+1)!}. \tag{6.61}$$

Since

$$q_j \approx 2u_j - f(x_j), \tag{6.62}$$

and

$$q(x) \approx 2u(x) - f(x), \tag{6.63}$$

then clearly from the bound (6.60) we obtain

$$\max_{j=1,\ldots,n} |q(x_j) - q_j| \leq 2(\sqrt{n_\Delta + 1} + 2\sqrt{n_\Delta} + 1)A(f,h) \tag{6.64}$$

$$+2(1 + L^\epsilon(m) + L^\epsilon_B(m))\frac{(\Delta^\epsilon)^{m+1} M_u}{2^{2m+1}(m+1)!}. \tag{6.65}$$

Since $A(f,h) = O(h^{m+1})$ we conclude that $\max_{j=1,\ldots,n} |u(x_j) - u_j| = O(h^{m+1})$ and therefore $\max_{j=1,\ldots,n} |q(x_j) - q_j| = O(h^{m+1})$, as desired.

6.5 Numerical Results for the Heat Equation

In this section we present our first example of application of the FC-AD algorithm (6.22). Here the method is applied to solve the PDE given in (6.1) where right-hand-side and boundary conditions were chosen in such a way that the exact solution of the problem is given by

$$u(x,y,t) = \sin(\pi(9x^2 + 4y^2 + 2t)), \tag{6.66}$$

with $k = 1$. The boundary of the computational domain is the parameterized curve:

$$x(\theta) = a_x(10\sin^2(2\theta) + 3\cos^3(2\theta) + 40)\cos(\theta), \tag{6.67}$$

$$y(\theta) = a_y(10\sin^2(2\theta) + 3\cos^3(2\theta) + 40)\sin(\theta),$$

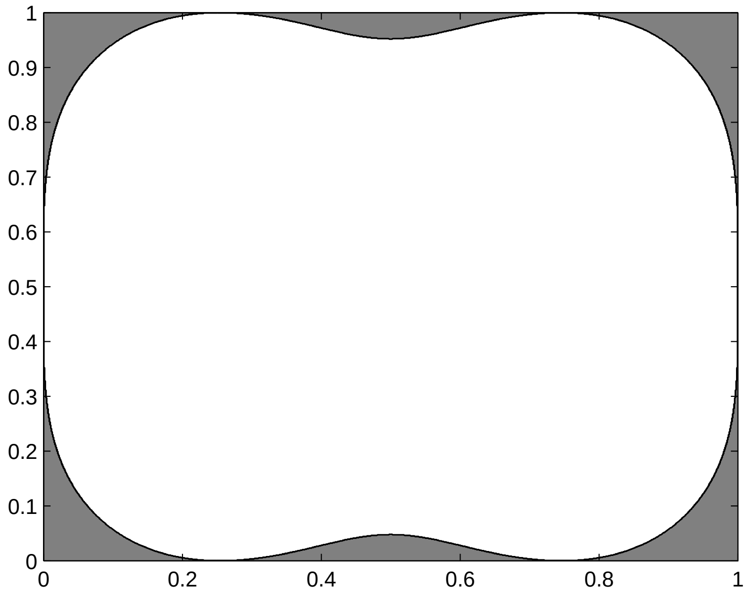


Figure 6.5: Domain used for the solution of the Heat Equation, where the boundary is defined in Equation (6.67).

for $\theta \in [0, 2\pi]$. The constants a_x, and a_y where chosen so that domain fits in the unit square; the resulting domain is depicted in Figure 6.5.

As shown in Chapter 10, we need to set $m \leq 4$ in order for the algorithm to be unconditionally stable; we set $m = 4$ and therefore expect fifth-order spatial convergence. In order to demonstrate the convergence with respect to the time step, in the first example a fine spatial discretization was fixed ($h_x = h_y = 1.0 \cdot 10^{-3}$). The calculation was preformed for multiple values of Δt to a final time $T = 1.0$. The maximum error calculated over the discrete mesh $\mathcal{D}_\Omega$ at and for all time $\leq T$ is shown as a function of Δt in Figure 6.6: this graph demonstrates a second-order convergence rate in time until the limiting accuracy of the spatial resolution is reached. In a second test, we use a fixed time-step of $\Delta t = 10^{-5}$, and we compute the maximum error for all points in space and all time-steps up to a final time of $T = 0.01$; the errors thus obtained are shown in Figure 6.7 as a function of the spatial discretization mesh-size along with a fifth-order slope line: clearly the expected fifth order convergence is achieved.

To further demonstrate the unconditional stability, the time-step $\Delta t = 10^{-4}$ was chosen and the value of h was refined. The maximum error at any step in time is displayed in

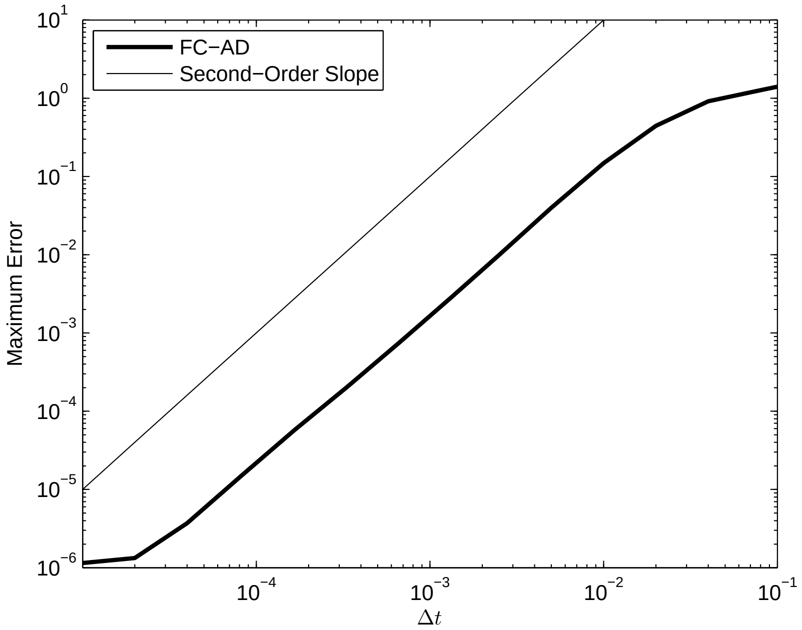


Figure 6.6: Time convergence for the Heat Equation on the domain depicted in Figure 6.5.

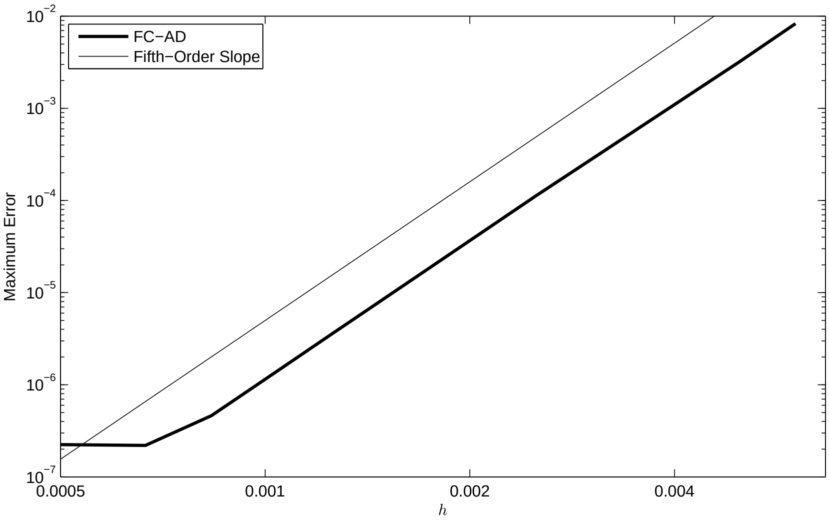


Figure 6.7: Spatial convergence for the Heat Equation on the domain depicted in Figure 6.5.

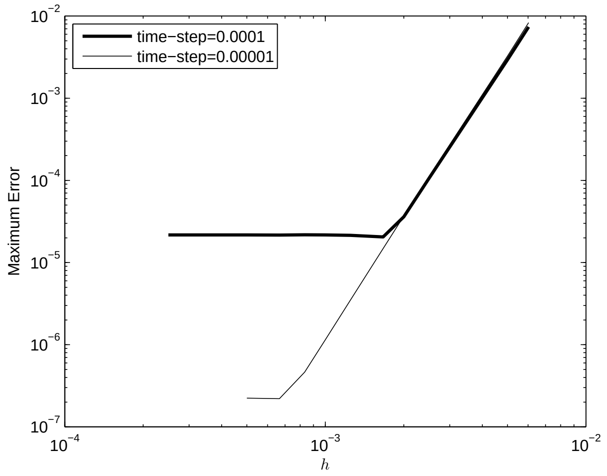


Figure 6.8: Unconditional stability demonstrated by refining the spatial discretization for fixed $\Delta t = 10^{-4}$ and $\Delta t = 10^{-5}$.

Figure 6.8 along with the results for $\Delta t = 10^{-5}$. The stability limit for an explicit method is governed by the distance between the two closest points of the domain including the boundary of the domain. Therefore the time step for the previous problem would have to be significantly less than $2.57 \cdot 10^{-7}$ unless techniques similar to those mentioned in Section 2.1 are used for which the limit for Δt would be approximately $2.57 \cdot 10^{-7}$. The FC-AD algorithm was also run one thousand iterations with $\Delta t = 100$ for the previous problem. Since our example problem does not have a steady state solution, due to the time varying source Q, no accuracy was expected in the approximation but the stability of the approximate solution was nontheless observed.

Additionally to demonstrate the unconditional stability, Δt was refined, for the fixed spatial resolutions of $h = 5.0 \cdot 10^{-3}$, $h = 3.3 \cdot 10^{-3}$, and $h = 2.5 \cdot 10^{-3}$. The maximum error at any time-step calculated to a final time $t = 0.01$ is displayed in Figure 6.9. The Figure 6.9 represents the calculation of millions of time steps over tens of thousands of unknowns, all of which can be easily handled on a standard desktop computer. Note in Figure 6.9, that a slight cusp is observed in the computational results. In fact this cusp occurs at precisely the location where χ (see Equation (6.41)) switches from 1 to $25\alpha^2/h^2$, providing evidence of the necessity of this parameter for unconditional convergence. Beyond this cusp a strict reduction in the maximum error is observed.

To demonstrate the computational speed of the method, and its essentially linear cost,

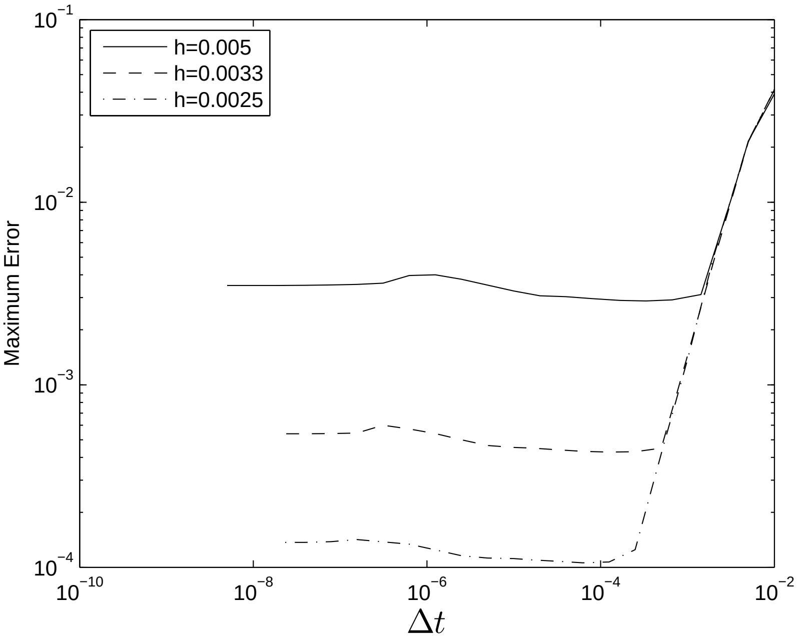


Figure 6.9: Unconditional stability demonstrated by refining the time step Δt for fixed spatial resolutions of $h = 5.0 \cdot 10^{-3}$, $h = 3.3 \cdot 10^{-3}$, and $h = 2.5 \cdot 10^{-3}$.

in Figure 6.10 we present the computational time required by the algorithm per time step as a function of the number of spatial unknowns (Here we used our C implementation of the algorithm, compiled with gcc, on a single 2.33 GHz processing core). We note that the algorithm requires approximately one second per time step with one million unknowns, where each iteration corresponds to a complete pass in both spatial directions. The implemented code used approximately 170 MB of memory with approximately 4 million unknowns. The memory usage of the algorithm was not optimized in any way and we estimate that it could be reduced by 60% with a minimal reduction in speed.

Naturally, the FC-AD methods apply to three-dimensional problems as well. For the Heat Equation, several Alternating Direction type splittings are available including the Locally One-Dimensional (LOD) scheme [99] which is based on Crank-Nicholson time-stepping. Here we present results obtained in three dimensions by Backward Euler time-stepping, resulting in a sixth-order spatially accurate scheme with first-order accuracy in time (noting again that, as seen in Section 8.2, Richardson Extrapolation can be used to increase temporal accuracy—something we do not do here for the sake of simplicity). The FC-AD was applied to the Heat Equation with parameters chosen such that the exact solution is given

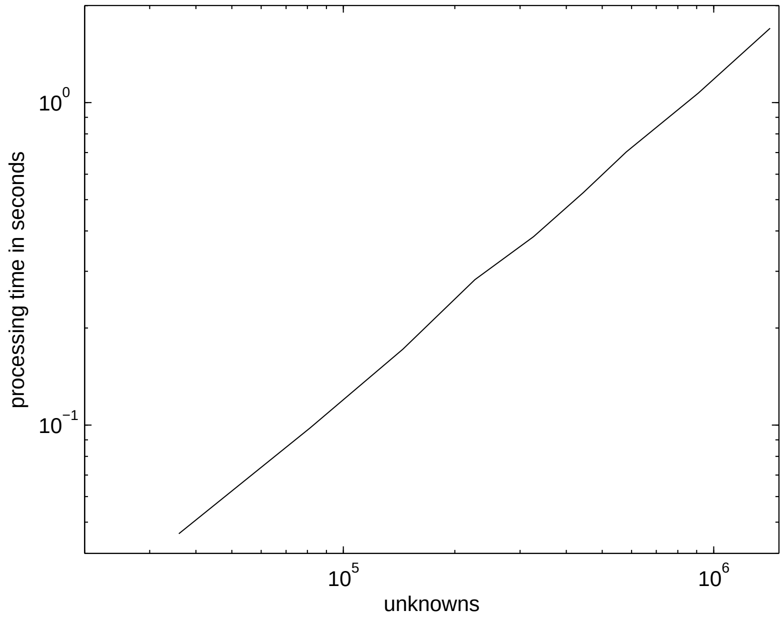


Figure 6.10: Processing time for a single time-step of the Heat Equation on a single processing core of a 2.33 GHz Intel Core 2 Duo processor.

by

$$u(x, y, z, t) = e^{-3\pi^2 t} \sin(\pi x) \sin(\pi y) \sin(\pi z) \tag{6.68}$$

on the domain consisting of the complement of the sphere of radius $r = 0.125$ in the unit cube. The spatial resolution was fixed at $h = \frac{1}{60}$ and the computation repeated for various time-steps. The results are shown in Figure 6.11 demonstrating the first-order time accuracy as well as stability well above typical stability limits.

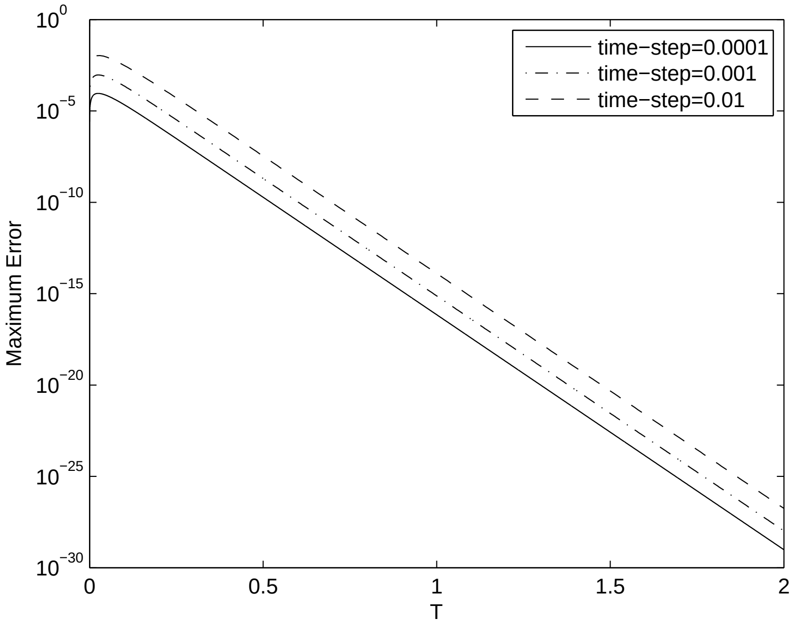


Figure 6.11: Maximum error from a full three-dimensional simulation of the Heat Equation for fixed $h = \frac{1}{60}$ and various time-steps demonstrating first-order temporal accuracy and stability well above typical stability limits. The domain consisted of the volume contained within a unit cube but outside a sphere with radius $r = 0.125$ that is centered in the cube.

Chapter 7

FC-AD Algorithm for the Poisson Equation

The FC-AD methodology that was developed in the previous chapter will now be extended to the Poisson Equation. The most important aspect of this extension is the development of iteration parameters akin to those mentioned in Section 4.2.

7.1 Derivation of Technique for the Poisson Equation

We consider the Poisson Equation

$$\begin{aligned} -u_{xx} - u_{yy} &= Q(x,y), \quad (x,y) \in \Omega \\ u(x,y) &= G(x,y), \quad (x,y) \in \partial\Omega. \end{aligned} \tag{7.1}$$

Since the Poisson Equation can be viewed as the steady state solution to the Heat Equation, the splitting scheme in Equation (6.5) could be directly applied in this context. For efficiency, however, it has long been recognized that introducing sequences of iteration parameters (effectively changing Δt for each time-step in Heat Equation solver), can give rise to significantly faster convergence to the solution of the Poisson Equation (c.f., [119]). With the iteration parameters γ_j the iteration schemes (6.11) and (6.12) adapted for the Poisson Equation becomes

$$\begin{aligned} \tilde{u}^{j+\frac{1}{2}} &= \left(1 - \gamma_j \frac{\partial^2}{\partial x^2}\right)^{-1} \left(1 + \gamma_j \frac{\partial^2}{\partial y^2}\right) \tilde{u}^j + \gamma_j \left(1 - \gamma_j \frac{\partial^2}{\partial x^2}\right)^{-1} Q \\ \tilde{u}^{j+1} &= \left(1 - \frac{\gamma_j}{2} \frac{\partial^2}{\partial y^2}\right)^{-1} \left(1 + \gamma_j \frac{\partial^2}{\partial x^2}\right) \tilde{u}^{j+\frac{1}{2}} + \gamma_j \left(1 - \gamma_j \frac{\partial^2}{\partial y^2}\right)^{-1} Q, \end{aligned} \tag{7.2}$$

where the boundary conditions for Equation (7.2) are given by $G(x, y)$. As for the Heat Equation it is convenient to rewrite the alternating-direction scheme in terms of the variables

$$w^j = \left(1 + \gamma_j \frac{\partial^2}{\partial y^2}\right) \tilde{u}^j \quad \text{and} \quad w^{j+\frac{1}{2}} = \left(1 + \gamma_j \frac{\partial^2}{\partial x^2}\right) \tilde{u}^{j+\frac{1}{2}}, \tag{7.3}$$

so that Equation (7.2) becomes

$$\begin{aligned} w^{j+\frac{1}{2}} &= \left(1 + \gamma_j \frac{\partial^2}{\partial x^2}\right) \left(1 - \gamma_j \frac{\partial^2}{\partial x^2}\right)^{-1} \left(w^j + \gamma_j Q\right) \\ w^{j+1} &= \left(1 + \gamma_{j+1} \frac{\partial^2}{\partial y^2}\right) \left(1 - \frac{\gamma_j}{2} \frac{\partial^2}{\partial y^2}\right)^{-1} \left(w^{j+\frac{1}{2}} + \gamma_j Q\right). \end{aligned} \tag{7.4}$$

The determination of optimal iteration parameters, γ_j, have been extensively researched in the finite difference case (c.f., [133, 140]). We propose a simple choice of iteration parameters that provide surprisingly efficient performance. Our choice was made on the basis of the following considerations. The result of applying the operator $\left(1 + \gamma_j \frac{\partial^2}{\partial x^2}\right)\left(1 - \gamma_j \frac{\partial^2}{\partial x^2}\right)^{-1}$ with periodic boundary conditions to a given periodic function μ given by a Fourier series

$$\mu(x) = \sum_{k \in t(N)} a_k e^{iPxk}, \tag{7.5}$$

of period $2\pi/P$, clearly we have

$$\left(1 + \gamma_j \frac{\partial^2}{\partial x^2}\right) \left(1 - \gamma_j \frac{\partial^2}{\partial x^2}\right)^{-1} \mu(x) = \sum_{k \in t(N)} \frac{1 - \gamma_j P^2 k^2}{1 + \gamma_j P^2 k^2} a_k e^{iPxk}; \tag{7.6}$$

a graph of the multiplier $\frac{1-\gamma_j P^2 k^2}{1+\gamma_j P^2 k^2}$ is shown in Figure 7.1. If we consider the function $\mu(x)$ to be the error in our approximate solution, then components of the error in Fourier modes with index $k \approx \frac{1}{P\sqrt{\gamma_j}}$ will be reduced significantly while the error in modes with index k far from $\frac{1}{P\sqrt{\gamma_j}}$ will remain nearly the same in magnitude. Our choice of iteration parameters follows directly from insuring that every possible Fourier mode is reduced by at least a factor of ε at least once. We thus use

$$\gamma_{j+1} = \phi \gamma_j \quad \text{with} \quad \phi = (1 - \varepsilon)^2 / (1 + \varepsilon)^2, \tag{7.7}$$

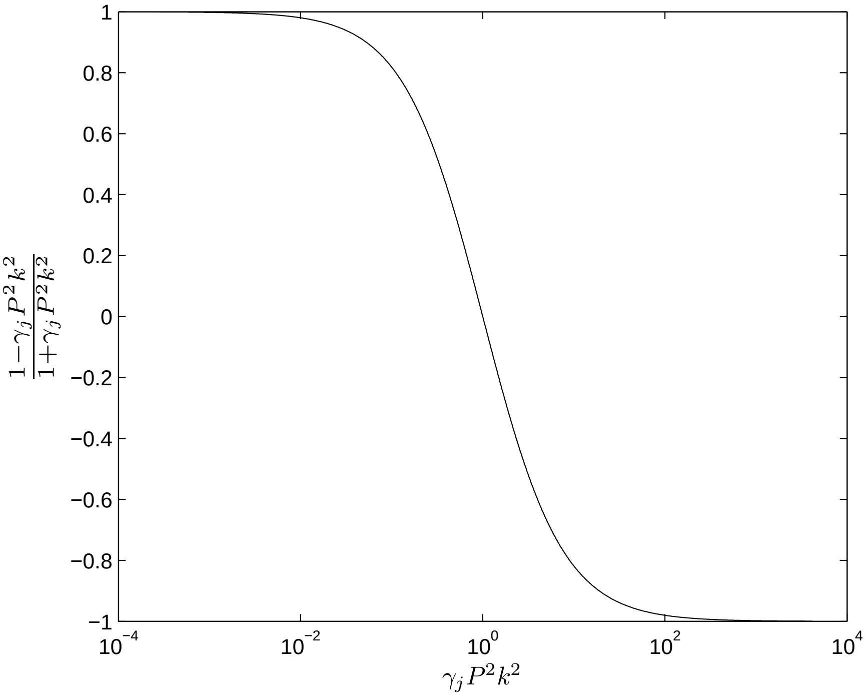


Figure 7.1: Behavior of the multiplier $\frac{1-\gamma_j P^2 k^2}{1+\gamma_j P^2 k^2}$.

giving a total of

$$N_I = -2\log(2P/h)/\log(\phi), \tag{7.8}$$

iteration steps. Although the exact sequence could be varied, in particular the initial step γ_0 (here we take $\gamma_0 = \frac{\pi^2}{10P^2}$ which we have found consistently works very well for the examples we have considered).

The accuracy of the final result and the number of iterations depends on the choice of ε. A smaller choice of ε requires more iterations and in turn provides greater accuracy if the spatial resolution of mesh allows it. Very good results have been obtained with very few iterations as will be shown in the next section.

7.2 Numerical Results for the Poisson Equation

Mesh Refinement Example In this section we present a fairly simple example which demonstrates, at the same time, the properties of the FC-AD algorithm and the mesh refinement strategy mentioned in Section 6.2. In detail, we consider Laplace's Equation in a circle with boundary condition

$$G(\theta) = e^{q\cos(\theta)}. \tag{7.9}$$

The exact solution is then given in terms of the Fourier coefficients of the function $G(\theta)$ as

$$u(x, y) = \sum_{k=-\infty}^{\infty} a_k r^k e^{2\pi i k\theta}, \tag{7.10}$$

where

$$G(\theta) = \sum_{k=-\infty}^{\infty} a_k e^{2\pi i k\theta}. \tag{7.11}$$

This problem was solved on a coarse mesh requiring refinement as shown in Figure 7.2. The results of this experiment are shown in Table 7.1 for several values of q (see Equation 7.9). The high order accuracy of the algorithm leads to high-quality solutions from a rather coarse mesh. Decreasing the spacing of the coarse mesh shows the convergence of the algorithm, in particular for $q = 5$, reducing the mesh by a factor of 4 yields reduction in error by a factor of about $(1/4)^{5.4}$ ($(1/4)^5$ is expected), demonstrating the high-order convergence of the algorithm.

A challenge problem for spectral embedding methods presented in [22] provides an in-

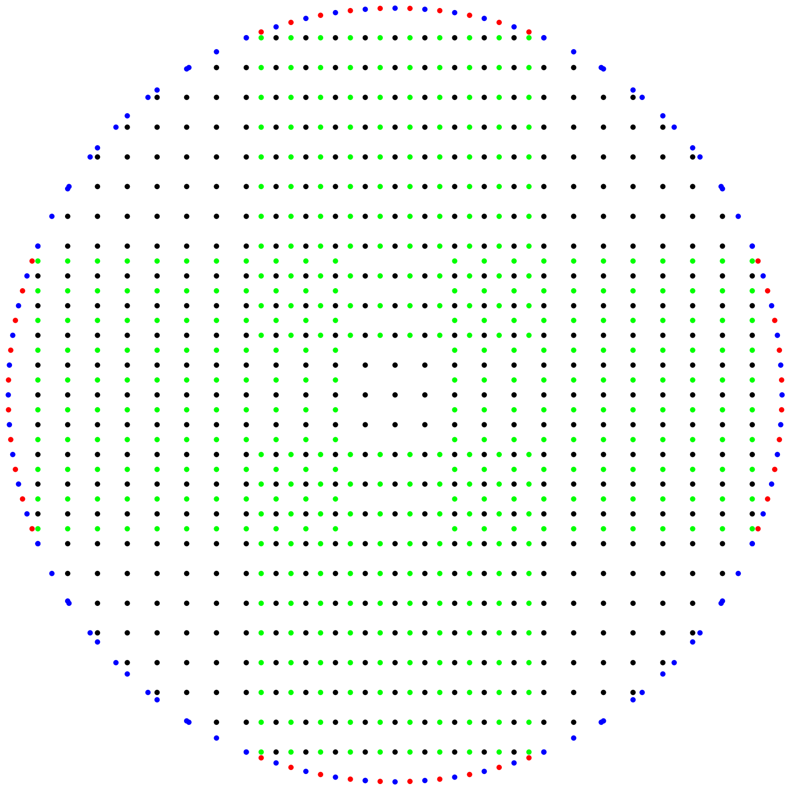


Figure 7.2: Refined coarse mesh on a circle showing the primary discretization points in black, refinement interior points in green, boundary points in blue, with additional boundary points required by the refinement shown in red.

$q = 1$	$q = 2$	$q = 3$	$q = 5$
$3.39 \cdot 10^{-6}$	$2.94 \cdot 10^{-4}$	$5.72 \cdot 10^{-3}$	$2.73 \cdot 10^{-2}$

Table 7.1: Maximum errors produced in the solution of Laplace Equation on a circle with the mesh shown in Figure 7.2.

teresting test problem for our FC-AD Poisson solver. The solution in the challenge problem is given by

$$\Psi(x, y) = \log\{\cosh(y - \pi) - \cos(x)\} - \log\{\cosh(y + \pi) - \cos(x)\}, \tag{7.12}$$

and the PDE domain is the region between $-2\pi \leq x \leq 2\pi$ and the curves

$$|\Psi(x, y)| = \log\{\cosh(3\pi) - 1\} - \log\{\cosh(\pi) - 1\}. \tag{7.13}$$

This domain is shown in Figure 7.3 with black dots on the figure denoting the location of the singularities in the solution just outside the domain of interest. These singularities, which prevent convergence of a series expansion over the complete rectangle, present a challenging configuration for spectral methods—since convergence of a spectral series within the actual PDE domain often implies convergence over the complete rectangle. Figure 7.4 shows numerical results produced by our FC-AD solver, with Dirichlet boundary data given by (7.12) on the entire boundary of the domain. In order to demonstrate the number of iterations required to reach a given accuracy, a variety of values of the parameter ε were used for each of three mesh spacings. For each one of the values $\varepsilon = 0.2$, 0.1, 0.05, 0.025, 0.01, 0.005, 0.0025, 0.001, and 0.0005, and for each one of three different meshes, demonstrated clearly that use of the iteration parameters given by Equation (7.7) produces rapidly decreasing errors as ε is decreased. Maximum errors, for each value of ε and each mesh, are shown in Figure 7.4 as a function of the number of iterations. Note that with only about 150 iterations, the solution has been obtained with an accuracy of 6 digits for each mesh size despite there being over one million unknowns for the finest mesh. This example clearly demonstrates the usefulness of the iteration parameters presented in the previous section; in particular we note that the number of iterations needed to reach a given accuracy is nearly independent of the number of unknowns. A more systematic study and optimization of the choice of these iteration parameters could in principle reduce the required number of iterations even further. Additionally the ultimate resolutions for several values of h, including those used above, are shown in Figure 7.5 demonstrating the fifth-order convergence of the method.

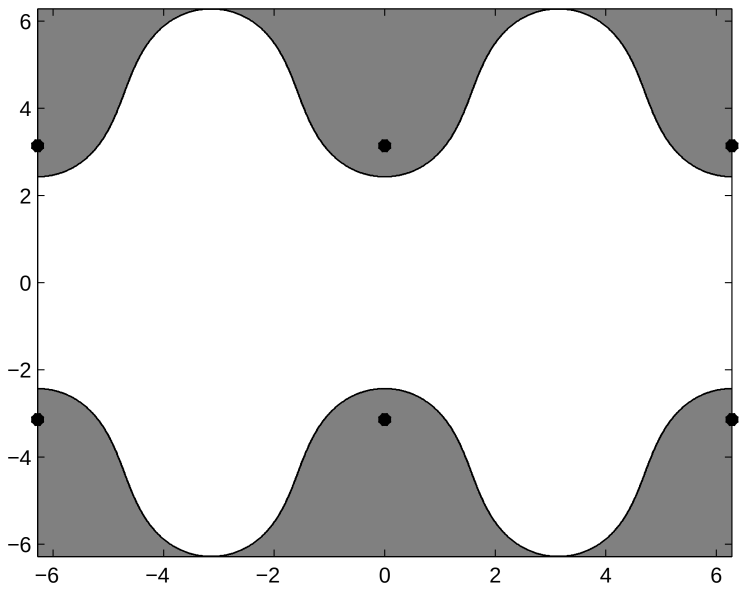


Figure 7.3: Domain used for solution of the Laplace Equation given by the region within $-2\pi \leq x \leq 2\pi$ and the curves $|\Psi(x, y)| = \log\{\cosh(3\pi) - 1\} - \log\{\cosh(\pi) - 1\}$. The black dots denote the locations of singularities of the function in (7.12).

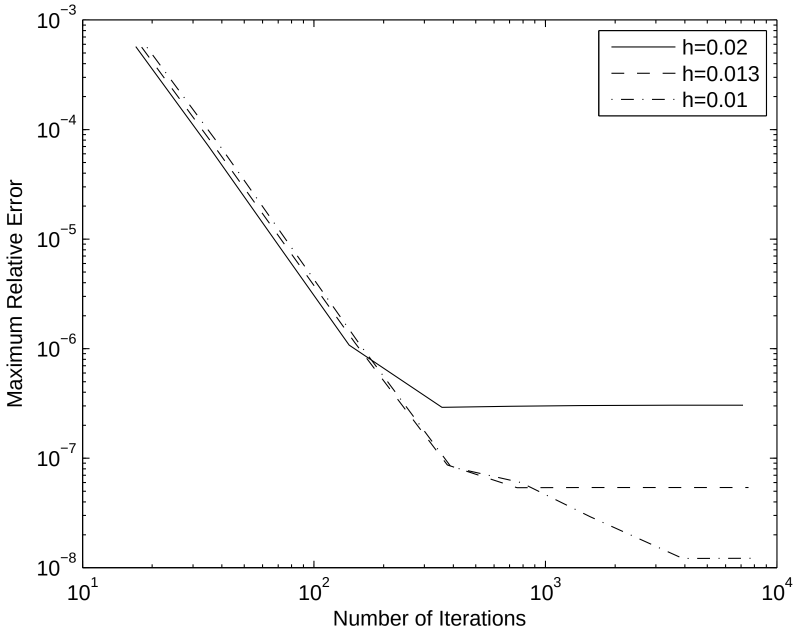


Figure 7.4: Convergence results for the Laplace Equation with Dirichlet boundary data imposed on the boundary of the region shown in Figure 7.3. The true solution is given by Equation (7.12). Results shown for three mesh spacings $h = 0.02$, $h = 0.013$, $h = 0.01$ which resulted in 274068, 617399, and 1098396 unknowns respectively.

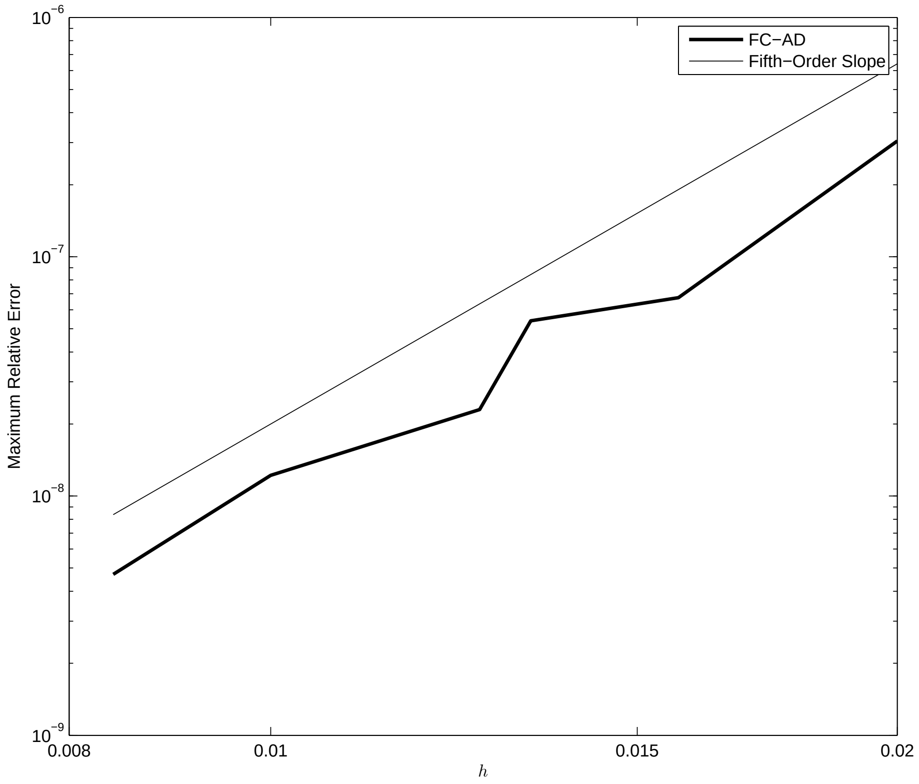


Figure 7.5: Convergence results for the Laplace Equation with Dirichlet boundary data imposed on the boundary of the region shown in Figure 7.3, where the true solution is given by Equation (7.12), demonstrating the expected fifth-order convergence of the FC-AD algorithm.

Chapter 8

FC-AD Algorithm for the Wave Equation

8.1 Derivation of Wave Equation FC-AD Algorithm

Let us now consider the Wave Equation in 2D with Dirichlet boundary data:

$$\begin{aligned}
&u_{tt} = k^2(u_{xx} + u_{yy}) + Q(x, y, t), \quad in \;\; \Omega \times (0, T], \\
&u(x, y, t) = G(x, y, t), \quad (x, y) \in \partial\Omega, \quad t \in (0, T], \\
&u(x, y, 0) = W(x, y), \quad (x, y) \in \Omega, \\
&u_t(x, y, 0) = V(x, y), \quad (x, y) \in \Omega,
\end{aligned} \tag{8.1}$$

where $\Omega \subset \Re^2$ is a smoothly bounded domain, and the functions Q, G, W, and V are given smooth functions. A second-order accurate time discretization following the pattern used previously leads to long time instability due to the occurrence of eigenvalues with small imaginary components in the iterative scheme. Instead a first order scheme is used that remains stable in our implementations. As shown in Section 8.2, Richardson extrapolation can be used to regain second order and even higher orders of accuracy in time (compare [65]).

Our FC-AD splitting of the Wave Equation follows from the following time discretization:

$$\frac{u^{n+1} - 2u^n + u^{n-1}}{\Delta t^2} = k^2 \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) u^{n+1} + Q^{n+1/2} + E_1(x, y, t), \tag{8.2}$$

where

$$\begin{aligned} E_1(x,y,t) \quad \leq \quad & k^2\Delta t \left\| u_{txx} \right\|_{L^\infty(\Omega\times(t^n,t^{n+1}))} + k^2\Delta t \left\| u_{tyy} \right\|_{L^\infty(\Omega\times(t^n,t^{n+1}))} \\ & + \frac{\Delta t^2}{12} \left\| u_{tttt} \right\|_{L^\infty(\Omega\times(t^n,t^{n+1}))} . \end{aligned} \tag{8.3}$$

Equation (8.2) can be re-expressed in the form

$$\left(1 - k^2\Delta t^2\frac{\partial^2}{\partial x^2} - k^2\Delta t^2\frac{\partial^2}{\partial y^2}\right)u^{n+1} = 2u^n - u^{n-1} + \Delta t^2 Q^n + \Delta t E_1(x,y,t). \tag{8.4}$$

The operator on the left hand side of Equation (8.4) may be split creating an additional error

$$E_2(x,y,t) = \frac{k^4\Delta t^4}{4}\frac{\partial^2}{\partial x^2}\frac{\partial^2}{\partial y^2}u^{n+1} \leq \frac{k^4\Delta t^4}{4}\|u_{xxyy}(x,y,t)\|_{L^\infty(\Omega\times(t^n,t^{n+1}))}, \tag{8.5}$$

inverting the split operator we obtain the alternating direction algorithm

$$\tilde{u}^{n+1} = \left(1 - k^2\Delta t^2\frac{\partial^2}{\partial y^2}\right)^{-1}\left(1 - k^2\Delta t^2\frac{\partial^2}{\partial x^2}\right)^{-1}(2u^n - u^{n-1} + \Delta t^2 Q^{n+1/2}). \tag{8.6}$$

The boundary conditions for each inverse operation are given by $G(x,y,t^{n+1})$. As in the case of the Heat Equation (see Remark 6.1.1), this choice of boundary data yields additional error, E_3, which is bounded by

$$E_3 \leq \frac{k^2\Delta t^2}{4}\|u_{yy}(x,y,t)\|_{L^\infty(\Omega\times(t^n,t^{n+1}))}; \tag{8.7}$$

which despite being an order of Δt larger than any other term, in practice turns out not to degrade the temporal accuracy order.

Note that the solution of Equation (8.6) is computed by solving ODEs of the same form as (6.27) with

$$\alpha = k\Delta t, \tag{8.8}$$

by means of the FC-ODE algorithm described in Section 6.3. We have demonstrated the unconditional stability of the algorithm for a variety of geometries by both explicit calculation of the eigenvalues of the full two-dimensional system and by performing millions of time steps (which, in fact, is equivalent to using the power method for evaluation of the largest eigenvalue). We have not yet been able to reduce the stability analysis to properties

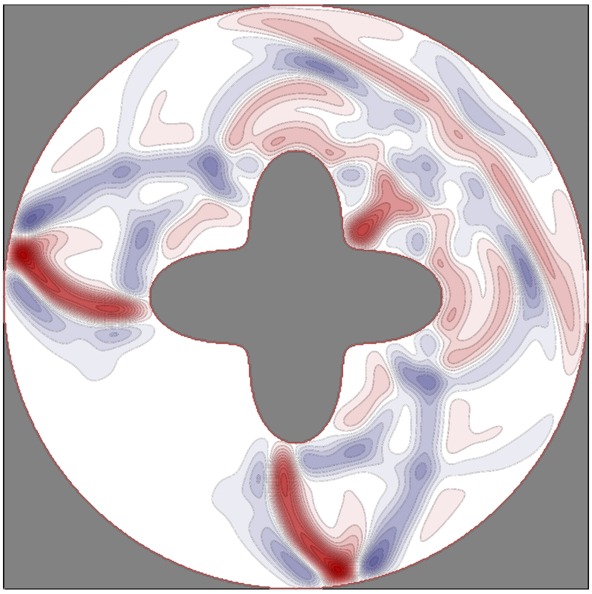


Figure 8.1: Domain test problem for the Wave Equation.

of the one-dimensional operators for the Wave Equation as we do for the Heat Equation. We do emphasize, however, that after very many computational runs for a variety of two- and three-dimensional geometries, we have never observed an instability—for any time-step/spatial-mesh-size whatsoever.

8.2 Numerical Results for the Wave Equation

In order to demonstrate the convergence of the FC-AD algorithm we consider the geometry shown in Figure 8.1. The external boundary is simply a circle centered at $x = 0.5$ and $y = 0.5$. The inner boundary curve can be parameterized in terms of $\theta \in [0, 2\pi]$ as

$$x(\theta) = \frac{(1 + \cos(2\theta))\cos(\theta)}{8} - \frac{1}{2}, \quad y(\theta) = \frac{(1 + \cos(2\theta))\sin(\theta)}{8} - \frac{1}{2}, \tag{8.9}$$

Gaussian initial data of the form,

$$u(x, y, 0) = e^{-1000\{(x-0.75)^2+(y-0.75)^2\}}, \tag{8.10}$$

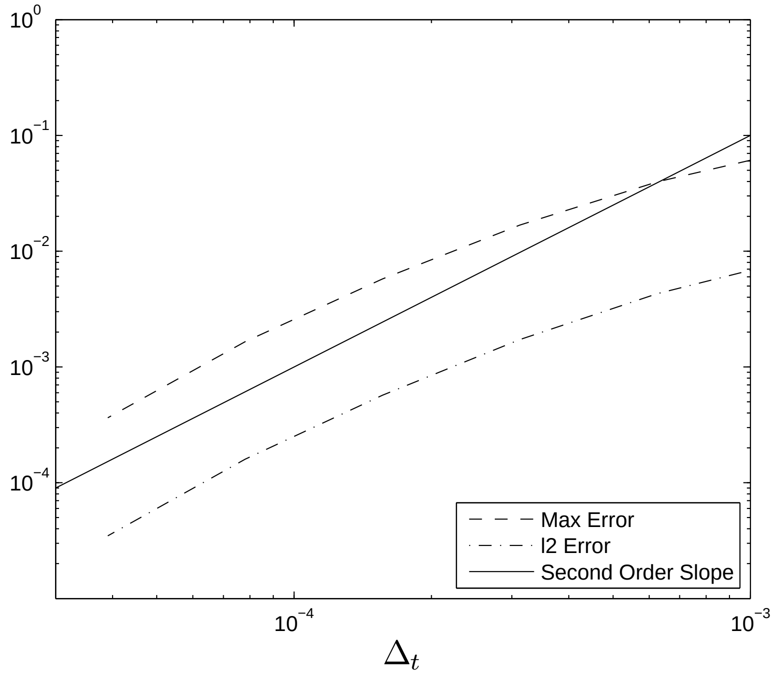


Figure 8.2: Maximum error as a function of the time-step for the Wave Equation with Gaussian initial data, using $h = 0.002$.

was used. The boundary condition $g(x, y, t)$, the forcing function $f(x, y, t)$ and the time derivative at $t = 0$, $u_t(x, y, 0)$, were all taken to equal zero, and we chose $k = 1$. The solution was calculated until a final time $T = 1$ at which point the wave resulting from the pulse had traveled through most of the domain. A convergence study was performed to determine the accuracy. For the accuracy in time, the spatial resolution was fixed at $h_x = h_y = 0.002$. Time steps ranged from $\Delta t = 1/800$ to $\Delta t = 1/25600$ and were then compared to the result with $\Delta t = 1/51200$. The time convergence displayed in Figure 8.2 demonstrates the expected second order accuracy in time. To study the convergence as the spatial discretization is refined, the value $\Delta t = 1/3000$ was fixed and the value of $h_x = h_y$ was allowed to vary from 0.008 to 0.003 and compared with the reference solution calculated with $h_x = h_y = 0.001$. The result is shown in Figure 8.3 demonstrating the high-order convergence in space of the FC-AD algorithm.

Additionally to show unconditional stability, the Wave Equation was solved with $k = 1$ over a domain with the boundary defined by the function $x^4 + y^4 = 1$. The functions Q, G,

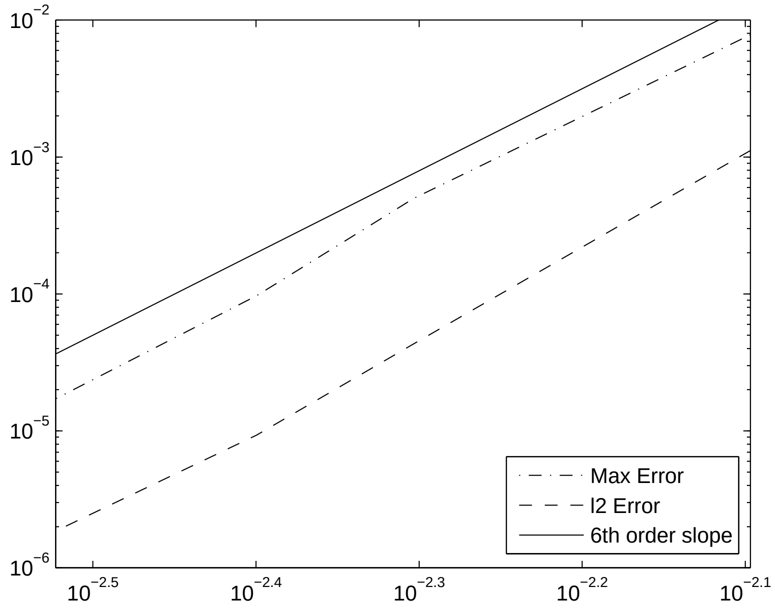


Figure 8.3: Maximum error as a function of the spatial resolution h for the Wave Equation with Gaussian initial data, using $\Delta t = 1/3000$.

W, and V were taken so that the exact solution would be

$$u(x, y, t) = \sin\left(\sqrt{85}\pi(x - t)\right) + \sin\left(\sqrt{85}\pi(y - t)\right). \tag{8.11}$$

For a fixed mesh of $h = 0.01667$ and a fixed final time $T = 1$, various time steps were chosen ranging from 1 to $3.33 \cdot 10^{-7}$ and the maximum error at any time step during the calculation is shown in Figure 8.4.

The solution was subsequently obtained for the same problem using Richardson Extrapolation (see [65]), on the basis of solutions obtained with Δt equal to $5 \cdot 10^{-4}$, $2.5 \cdot 10^{-4}$ and $1.667 \cdot 10^{-4}$ and $1.25 \cdot 10^{-4}$ to produce a solution with errors proportional to Δt^4. The spatial resolution was refined from $h = 0.01667$ to $h = 0.001$. The errors obtained are displayed in Figure 8.5.

To demonstrate the benefits that arise from use of high-order Richardson extrapolation, in Figure 8.6 we show the errors resulting from calculations using fourth-order Richardson extrapolation and a fifth-order spatial FC-AD scheme, in which both the time-step and the spatial mesh-size are refined simultaneously. The error is primarily dependent on the

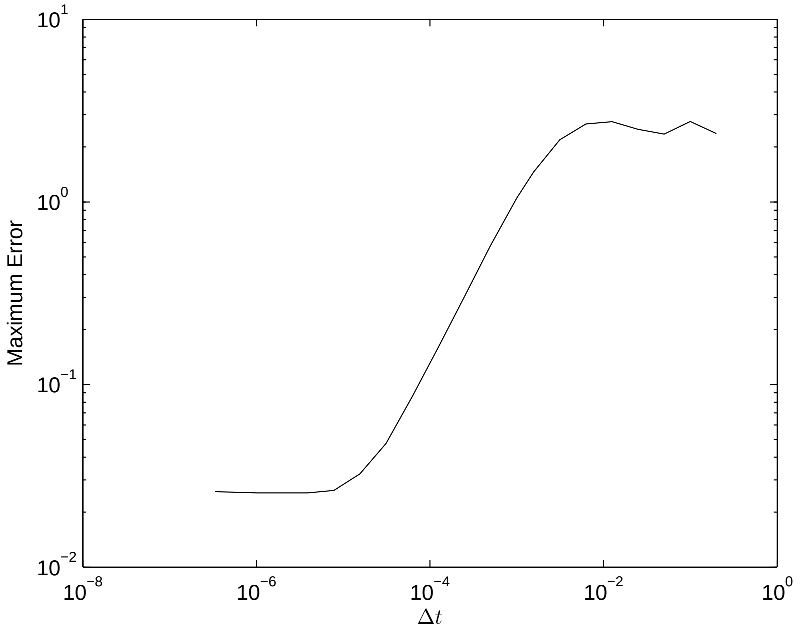


Figure 8.4: Solution to the Wave Equation with solution given in Equation (8.11) with fixed spatial resolution of $h = 0.01667$ for a range of values for Δt. The domain boundary is defined by $x^4 + y^4 = 1$ and the maximum error for each time-step is reported.

time-step and therefore fourth-order convergence is observed. The error could be improved by implementing a restarted Richardson Extrapolation (see [65]). We note some care must be take in the extrapolation so that the parameter χ (see Equation 6.40) is either one or less than one for each time level, as appropriate, in order to obtain the full fourth-order convergence.

In order to demonstrate that these techniques also apply to three-dimensional complex domains, as already shown for the Heat Equation, a sample calculation is shown in Figure 8.7 where a Gaussian pulse was initiated at the location $x = 0.5$, $y = 0.75$, and $z = 0.75$ with zero boundary conditions on both the surface of the sphere and the surrounding cube. The projection of the solution into the three planes $x = 0.5025$, $y = 0.5025$, and $z = 0.5025$ are shown in Figure 8.7. The unconditional stability and accuracy of the algorithm are maintained in this three-dimensional context.

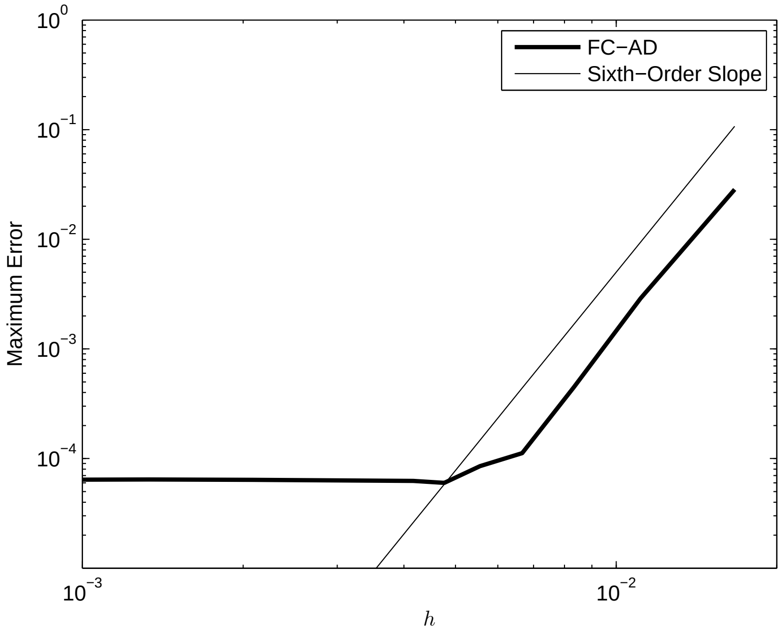


Figure 8.5: Numerical errors arising in the solution of a Wave Equation, with exact solution given in Equation (8.11), as a function of the spatial mesh-size h. Fourth-order Richardson Extrapolation was used. The domain boundary is defined by $x^4 + y^4 = 1$ and the maximum error at any time step is reported. Sixth-order convergence is shown in the region where the error is determined by the spatial discretization.

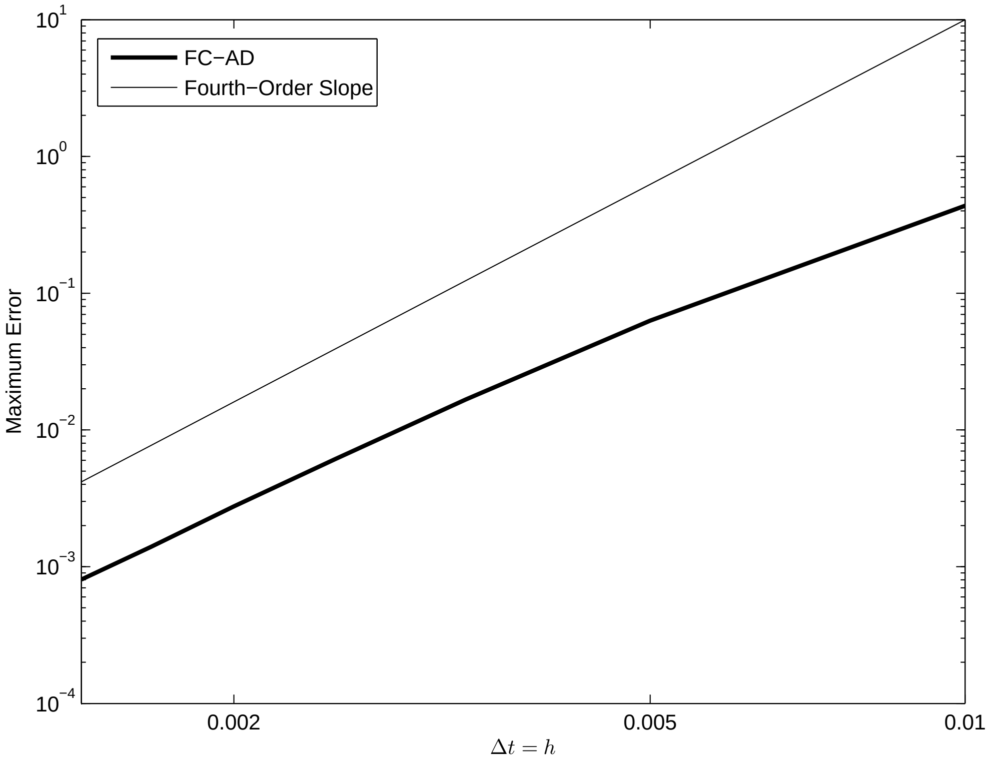


Figure 8.6: Numerical errors arising in the solution of a Wave Equation, with exact solution given in Equation (8.11), as a function of the spatial mesh-size. Fourth-order Richardson Extrapolation was used. The coarsest time-step of the extrapolation was taken to equal h; the other time-steps used were $h/2$, $h/3$, and $h/4$. Fourth-order convergence is observed as h and therefore Δt are refined.

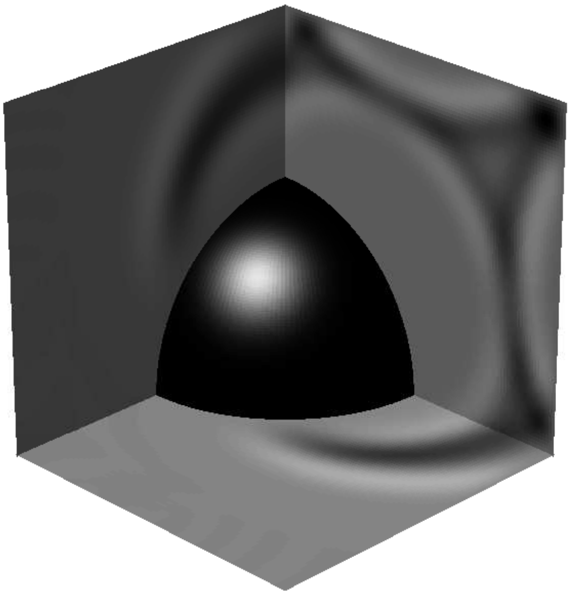


Figure 8.7: Solution to the Wave Equation with Gaussian initial data in a domain consisting of the complement of a sphere within a cube. The gray-scale on the three planar sections $x = 0.5025$, $y = 0.5025$, and $z = 0.5025$ display the planer values of the solution.

Chapter 9

Comparative Discussion

In this chapter, we compare the performance of the FC-AD methodology to that of other methods for the solution of PDEs. In doing so we attempt to take into account general properties of the solution techniques rather than explore the manifold nuances of each individual algorithm in the literature. Our focus will be on the FC-AD's avoidance of pollution errors (which encumber all Finite-Difference and Finite-Element approaches), and its efficient unconditionally stable implementation which arises primarily from the Alternating Direction framework and use of the FFT. We will not make a direct comparison to Pseudospectral methods due since the geometric limitations of those methods, described in Chapter 3, are well established.

In order to facilitate the task at hand, we perform comparisons for particularly simple one- and two-dimensional geometries. These comparisons show that even for configurations for which finite-difference methods show their best performance (e.g., a fourth-order finite difference method applied to a one dimensional interval with an assumption of periodicity) the FC-AD approach exhibits significant advantages over the classical approaches; we show further that the performance of the FC-AD methodology remains unaffected as one moves to more complex problems: e.g., the FC-AD accuracy and performance on a square domain remain unchanged as the problem is ported to a more complex curved geometry. While we mainly compare the FC-AD to the finite difference approach, we expect a similarly favorable comparison with the finite-element method holds. We note however that we would expect significantly more degradation in the performance for FDM methods relative to FEM for more complex situations than considered in this section (see Chapter 2).

9.1 Transient and Time-harmonic Wave Propagation: Pollution Error

As discussed in Section 2.3, the error produced in representing wave propagation by means of finite-differences or finite-elements is compounded over the length of the domain—resulting in a requirement of increasingly finer spatial discretizations, as the number of waves in the domain increases, to produce constant overall errors. The accumulation of error is inherent to both FEM and FDM, as shown in [11] and [84] respectively. In Figure 9.1 we display both the error in a spatially second-order explicit finite-difference scheme and the error arising from our FC-AD algorithm for a simple spatially Wave Equation for which the solution is $\sin(2\pi w(x-t))$ over the unit intervals in time and space. The time step used for both algorithms was taken sufficiently small as to have no significant affect on the error. The maximum error for all time-steps at all discretization points is reported in the figure for 15 and 20 PPW spatial discretizations fixed for the FC-AD method and 25 and 400 PPW for the second-order finite-difference method. This represent an extreme case where finite-differences perform near optimally in view of the periodicity of the solution and regular spatial grid used. Note the FC-AD method cannot benefit from the periodicity because it extends the domain and recalculates a periodic extension on this new domain as shown in Figure 9.3. The FC-AD algorithm yields similar performance for an arbitrarily shaped domain in two- and three-dimensional space.

A similar graph is shown in Figure 9.2 comparing the FC-AD method with fourth-order finite-difference method. The obvious conclusion is that low-order methods are exceedingly costly for problem whose spatial dimensions span many wavelengths. This contention is also supported by Figure 9.4, which shows the points per wavelength required by the finite-difference and FC-AD methods to reach 1% maximum error. The improvement in the performance from second- to fourth-order finite difference methods is remarkable but even with fourth-order finite difference methods the growth in the required number of points per wavelength makes 3D calculations prohibitive. This is particularly significant since comparable FDM for complex geometries are just now reaching fourth-order accuracy (see Section 2.1): fourth order methodologies in two- and three-dimensions are not quite established as yet. In contrast, for the FC-AD methodology the number of points per wavelength required for a given accuracy quickly ceases to grow—and therefore offers the method a sig-

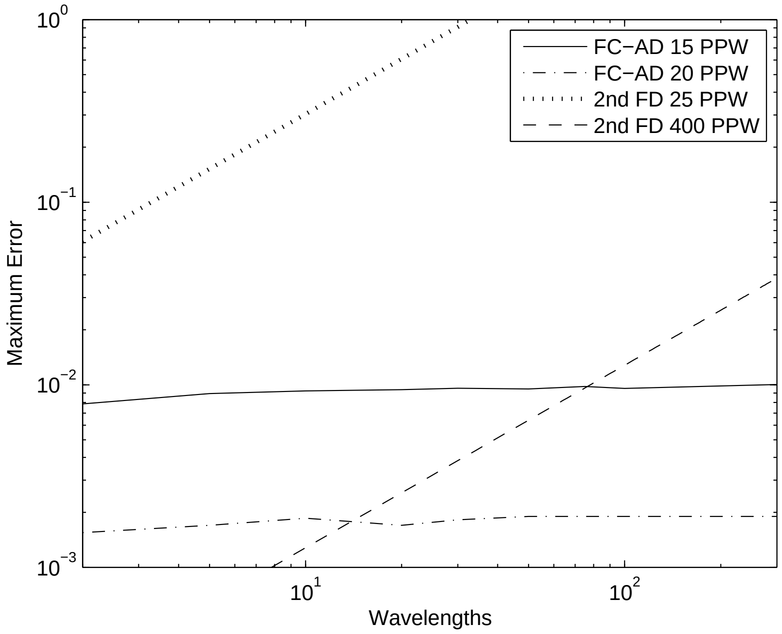


Figure 9.1: Maximum errors arising from applications of FC-AD and second-order finite-difference algorithms for increasing number of wavelengths and various fixed numbers of PPW.

nificant advantage over alternative approaches.

9.2 Comparison for Heat Type Equations

While the pollution error of the previous section is not as significant a source of error for diffusive equations, the conditionally stability of explicit schemes and the additional cost of implicit methods are. In a 2D square domain, consider the Heat Equation with $k = 1$ and solution given by

$$u(x, y, t) = \cos(20\pi(2t - x + y)), \tag{9.1}$$

—representing some sort of a thermal cycle. A spatially second-order accurate explicit finite-difference method was used to solve to accuracies of 10% and 1%. The time-step was chosen in order to maintain stability in the finite-difference algorithm, we used $\Delta t = h^2/4$. The required spatial resolutions are shown in Table 9.1 along with a lower bound for the computational cost required to produce a 0.1% error. The results were obtained on a 3.4 GHz Pentium D.

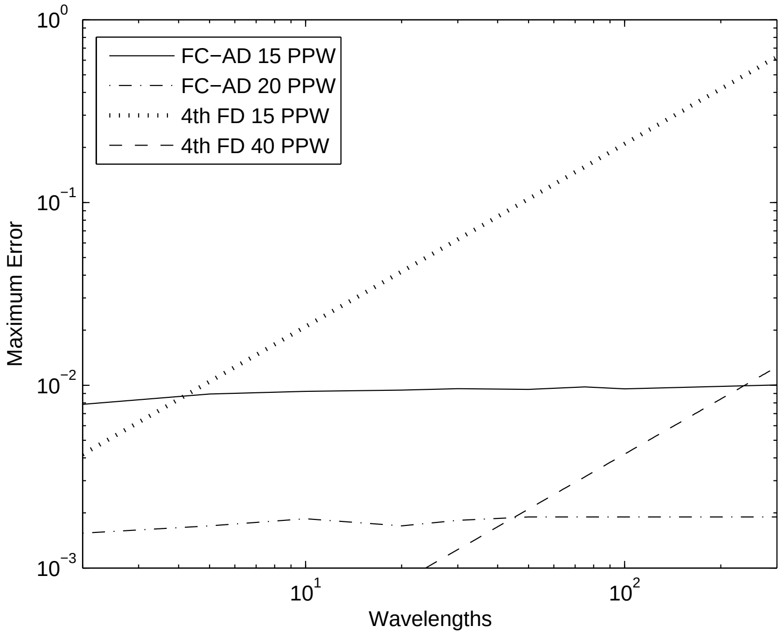


Figure 9.2: Maximum errors arising from applications of FC-AD and fourth-order finite-difference algorithms for increasing number of wavelengths and various fixed numbers of PPW.

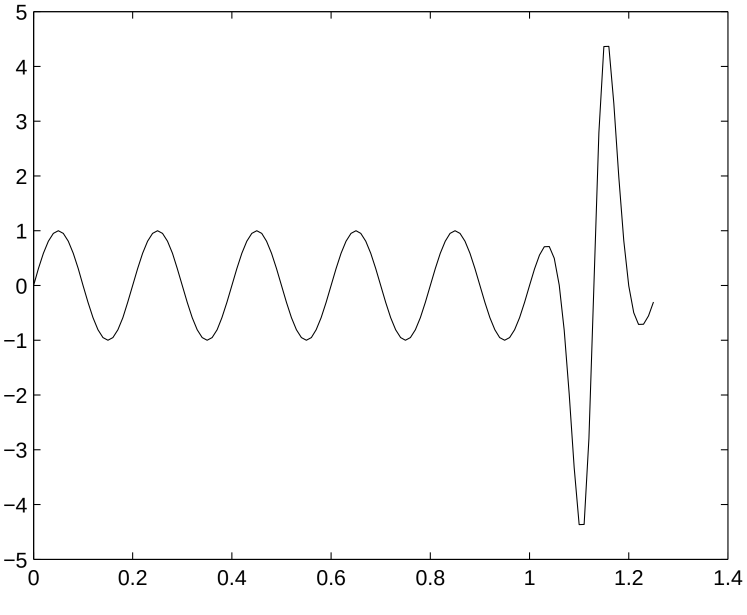


Figure 9.3: Extension of otherwise periodic function $\sin(10\pi x)$.

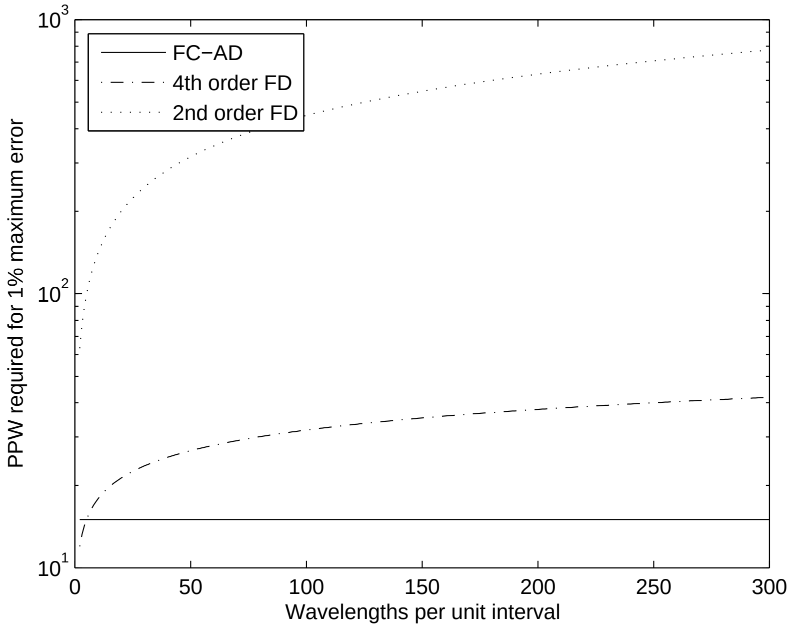


Figure 9.4: Estimated PPW required to obtain a 1% error in the solution of the Wave Equation with solution $\sin(2W\pi(x-t))$ to a final time $T = 1$ where W is the number of wavelengths.

Accuracy	h	Δt	*Comp. Time*
10%	$1.52 \cdot 10^{-2}$	$5.7 \cdot 10^{-5}$	33 secs
1%	$4.9 \cdot 10^{-3}$	$6.1 \cdot 10^{-6}$	2,940 secs
0.1%	$< 1.67 \cdot 10^{-3}$	$< 7.0 \cdot 10^{-7}$	$> 213,000$ secs

Table 9.1: Computational times required to produce various accuracies by means of an explicit second-order finite-difference solver for the Heat Equation in a square domain (see Equation (9.1)). Computations performed on a 3.4 GHz Pentium D processor.

Accuracy	h	Δt	*Comp. Time*
10%	$7.9 \cdot 10^{-3}$	$7.3 \cdot 10^{-3}$	3 secs
1%	$4.6 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	24 secs
0.1%	$2.9 \cdot 10^{-3}$	$7.0 \cdot 10^{-4}$	212 secs

Table 9.2: Computational times required to produce various accuracies by means of the FC-AD algorithm for the Heat Equation in a square domain (see Equation (9.1)). Computations performed on a 3.4 GHz Pentium D processor.

Accuracy	h	Δt	*Comp. Time*
9.3%	$7.9 \cdot 10^{-3}$	$7.3 \cdot 10^{-3}$	3.2 secs
0.852%	$4.6 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	25 secs
0.0793%	$2.9 \cdot 10^{-3}$	$7.0 \cdot 10^{-4}$	220 secs

Table 9.3: Computational results for the FC-AD algorithm applied to the Heat Equation over the domain bounded by $x^4+y^4 = 1$ using the parameters from Table 9.2. Computations performed on a 3.4 GHz Pentium D processor. These results show only minor variations in computational time and accuracy versus those produced by the FC-AD algorithm for the square domain.

The calculations were also performed using the FC-AD algorithm as detailed in Chapter 6 for 10%, 1%, and 0.1% and the corresponding timings, for the same hardware as Table 9.1, are shown in Table 9.2. Notice the dramatic difference in computational cost driven in particular by the difference in the number of time steps required for accuracies less than 10%. A single iteration of the full FC-AD algorithm was in fact approximately 3 to 5 times slower than a single step of the explicit finite-difference scheme on equivalent sized meshes (can be up to about 7 on square domains with prime numbers of points). Despite this the FC-AD algorithm was significantly faster even at the 10% error level. The computations in Table 9.2 were repeated with the FC-AD algorithm on the non-rectangular domain defined by $x^4 + y^4 = 1$ with the exact same spatial and temporal resolutions. The result of this experiment are presented in Table 9.3: the performance of the FC-AD approach for this geometry is acutally better than that obtained for the rectangular geometry considered in Table 9.2.

As mentioned in Section 2.4, an unconditionally stable finite-difference method was also applied with Crank-Nicolson time-stepping in order to obtain an implicit second-order accurate scheme in both time and space. The CG algorithm without preconditioning was used to solve the linear system at each time step using the solution at the previous time step as the initial guess. The tolerance of the CG algorithm was chosen to be as large as possible while still maintaining the desired accuracy level. It was found that for $h = 1.52 \cdot 10^{-2}$, the best performance came at a time step about 4.4 times larger than the explicit case but 9 CG iterations were needed on average per time step resulting in approximately double the computational time. By decreasing h to $1.25 \cdot 10^{-2}$, the number of time steps required for 10% accuracy were further reduced by another factor of 4 and only 8 CG iterations were required. Thus with these parameters, 10% accuracy was reached in approximately 60% of the total time of the explicit FDM which is still considerably more than was required for the FC-AD.

For higher accuracies, we were not able to improve the overall computational cost relative to the explicit version, even though we could take significantly larger time steps, due to the large number of CG iterations required. Similarly it is shown in [126] that to reach the same error tolerance, the required number of CG iterations, without preconditioning, grows proportional to h^{-1}. We note that a preconditioned CG iteration for the finite-difference method would have to improve the performance by at least two orders of magnitude in order to obtain speeds approximately those obtained by the FC-AD to 1% error. Unconditional stability and the absence of pollution error are two significant advantages of the FC-AD algorithm—which, in our opinion, makes it preferable to alternative algorithms for a wide range of computational PDE problems.

Chapter 10

Stability and Singular-Value Decompositions

In this Chapter we establish the stability of FC-AD algorithms introduced in Chapters 6 and 7, that is, the FC-AD algorithms for the Heat Equation and the Poisson Equation. Our analysis relates stability to the sizes of certain singular values. The conclusions of our analysis are drawn from a numerical evaluation of such singular values for a complete range of values of all parameters involved, with exception of the numbers n_{x_i} and n_{y_j} (see Section 6.2) of discretization points used in the various horizontal and vertical lines respectively: these are taken to span a large range but, naturally, not the complete (infinite) range of possible values.

10.1 Reduction to a Singular-Value Problem for the FC-ODE Solver

To start our study we introduce the following definition concerning the FC-ODE solver presented in Section 6.3.

Definition 10.1.1. *Let an n point discretization of the interval $[x_\ell, x_r]$, as described in Section 6.2 and depicted in Figure 6.1, be given. Let L^1 be the linear map from $\ell_2(n) \to \ell_2(n)$ that relates the n discrete values $f_j = f(x_j)$ of the function f to the approximate solution $u_i = (L^1 f)_i = \sum_{j=1}^{n} L^1_{ij} f_j$, given by the algorithm described in Section 6.3, of the one-dimensional boundary value problem*

$$-\alpha^2 u''(x) + u(x) = f(x), \quad u(x_\ell) = 0, \quad u(x_r) = 0. \tag{10.1}$$

Further let L^2 be the linear map from $\ell_2(n) \to \ell_2(n)$ defined by $L^2 f = (2L^1 - I)f$. Note that $q_j = \sum_{j=1}^{n} L^2_{ij} f_j$ is an approximation of $q(x) = 2u(x) - f(x)$. The linear maps L_1 and L_2 are then functions of the parameters n, h, n_Δ, m, d/Δ, g, Υ, $x_1 - x_\ell$, $x_r - x_n$ and α, as discussed in Chapters 5 and 6.

Remark 10.1.1. *L^1 is a discrete approximation to the operator $\left(1 - \alpha^2 \frac{\partial^2}{\partial x^2}\right)^{-1}$ with zero Dirichlet boundary data (see Equation* (6.9)*) while L_2 is a discrete approximation to the operator $\left(1 + \alpha^2 \frac{\partial^2}{\partial x^2}\right)\left(1 - \alpha^2 \frac{\partial^2}{\partial x^2}\right)^{-1}$.*

Definition 10.1.2. *Recalling the notations introduced in Section 6.2, we define the operators*

$$\begin{aligned} &\mathcal{L}^1_x : \ell_2(\mathcal{D}_\Omega) \to \ell_2(\mathcal{D}_\Omega), \\ &\mathcal{L}^1_y : \ell_2(\mathcal{D}_\Omega) \to \ell_2(\mathcal{D}_\Omega), \\ &\mathcal{L}^2_x : \ell_2(\mathcal{D}_\Omega) \to \ell_2(\mathcal{D}_\Omega), \quad \textit{and} \\ &\mathcal{L}^2_y : \ell_2(\mathcal{D}_\Omega) \to \ell_2(\mathcal{D}_\Omega) \end{aligned}$$

by

$$\begin{aligned} \mathcal{L}^1_x[\theta](x_i, y_j) &= \textstyle\sum_{k=1}^{n_{y_j}} L^1_{ik}\theta_{kj}, \\ \mathcal{L}^1_y[\theta](x_i, y_j) &= \textstyle\sum_{k=1}^{n_{x_i}} L^1_{jk}\theta_{ik}, \\ \mathcal{L}^2_x[\theta](x_i, y_j) &= \textstyle\sum_{k=1}^{n_{y_j}} L^2_{ik}\theta_{kj}, \quad \textit{and} \\ \mathcal{L}^2_y[\theta](x_i, y_j) &= \textstyle\sum_{k=1}^{n_{x_i}} L^2_{jk}\theta_{ik}. \end{aligned}$$

The operator norms $\|\mathcal{L}^1_x\|_{\ell_2(\mathcal{D}_\Omega)}$, $\|\mathcal{L}^1_y\|_{\ell_2(\mathcal{D}_\Omega)}$, $\|\mathcal{L}^2_x\|_{\ell_2(\mathcal{D}_\Omega)}$, and $\|\mathcal{L}^2_y\|_{\ell_2(\mathcal{D}_\Omega)}$, which result from the norm of the space $\ell_2(\mathcal{D}_\Omega)$ (Equation (6.26)), depend on the entire spatial discretization and, yet, they can be bounded by the largest singular-value of any the linear map L_1 or L_2 respectively along any line of data in $\mathcal{D}_\Omega$. It is easy to check the validity of this fact: for each of the four operators $\mathcal{L}^1_x$, $\mathcal{L}^1_y$, $\mathcal{L}^2_x$, and $\mathcal{L}^2_y$, each line of data (in x or y as appropriate) gives rise to one of a complete set of invariant subspaces for the operators, and therefore each one of the full operator norms is bounded by the maximum over all of theses subspaces.

For the Heat Equation, recalling the Equations (6.20) and (6.21), let $u(t^n)$, be the exact solution at time $t^n = n\Delta t$ calculated from the initial condition u^0. As was done in [81], we will examine our stability in the transformed variables $\left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right) u(t^n)$ on the discrete mesh. Let the operator $Eu(t^n)$ be defined as the evaluation of $\left(1 + \frac{k\Delta t}{2}\frac{\partial^2}{\partial y^2}\right) u(t^n)$ on $\mathcal{D}_\Omega$. We can then define $\epsilon^n = Eu(t^n) - w^n$ where w^n is the approximation to $Eu(t^n)$ calculated by the FC-AD algorithm from the same initial condition u^0 . Further let $\tau^n(h, \Delta t)$ be the error that is produced in obtaining w^{n+1} by the FC-AD algorithm from $Eu(t^n)$. Then by applying the FC-AD algorithm for one full time step (going from w^n to w^{n+1}) the new error, ϵ^{n+1}, can be bounded by

$$\|\epsilon^{n+1}\|_{\ell_2(\mathcal{D}_\Omega)} \leq \|\tau_n(n, \Delta t)\|_{\ell_2(\mathcal{D}_\Omega)} + \|\mathcal{L}_x^2\|_{\ell_2(\mathcal{D}_\Omega)} \|\mathcal{L}_y^2\|_{\ell_2(\mathcal{D}_\Omega)} \|\epsilon^n\|_{\ell_2(\mathcal{D}_\Omega)}, \tag{10.2}$$

where the discrete $\ell_2(\mathcal{D}_\Omega)$ norm was defined in Equation (6.26). The result is that for

$$\|\mathcal{L}_x^2\|_{\ell_2(\mathcal{D}_\Omega)} \leq 1 \quad \text{and} \tag{10.3}$$

the error is not amplified from an application of the FC-AD: the stability of the algorithm is ensured if Equation (10.3) is satisfied.

For the Poisson Equation the operators that must be applied at each step of the algorithm are of the form $\left(1 + \gamma_{j+1} \frac{\partial^2}{\partial x^2}\right) \left(1 - \gamma_j \frac{\partial^2}{\partial x^2}\right)^{-1}$ (see Equation (7.2)). By simple algebra we note

$$\begin{aligned} \left(1 + \gamma_{j+1} \frac{\partial^2}{\partial x^2}\right) \left(1 - \gamma_j \frac{\partial^2}{\partial x^2}\right)^{-1} &= \frac{\gamma_{j+1}}{\gamma_j} \left(1 + \gamma_j \frac{\partial^2}{\partial x^2}\right) \left(1 - \gamma_j \frac{\partial^2}{\partial x^2}\right)^{-1} \\ &\quad + \left(1 - \frac{\gamma_{j+1}}{\gamma_j}\right) \left(1 - \gamma_j \frac{\partial^2}{\partial x^2}\right)^{-1}, \end{aligned} \tag{10.4}$$

which in our discrete setting with zero Dirichlet data is simply the linear combination

$$\frac{\gamma_{j+1}}{\gamma_j} L_2 + \left(1 - \frac{\gamma_{j+1}}{\gamma_j}\right) L_1, \tag{10.5}$$

of the linear maps given in Definition 10.1.1 with $\alpha^2 = \gamma$. Therefore, for the Poisson Equation, the stability is governed by all four of the operator norms $\|\mathcal{L}_x^1\|_{\ell_2(\mathcal{D}_\Omega)}$, $\|\mathcal{L}_y^1\|_{\ell_2(\mathcal{D}_\Omega)}$, $\|\mathcal{L}_x^2\|_{\ell_2(\mathcal{D}_\Omega)}$, and $\|\mathcal{L}_y^2\|_{\ell_2(\mathcal{D}_\Omega)}$. We note that if $\gamma_{j+1} \leq \gamma_j$ (a property satisfied by the sequence given in Equation (7.7)) and the operator norms are all less than one, the FC-AD algorithm is stable. Thus the problem of establishing stability for the FC-AD algorithms for both the Heat Equation and the Poisson Equation has been reduced to the problem of showing that the operator norms $\|\mathcal{L}_x^1\|_{\ell_2(\mathcal{D}_\Omega)}$, $\|\mathcal{L}_y^1\|_{\ell_2(\mathcal{D}_\Omega)}$, $\|\mathcal{L}_x^2\|_{\ell_2(\mathcal{D}_\Omega)}$, and $\|\mathcal{L}_y^2\|_{\ell_2(\mathcal{D}_\Omega)}$ are less than or equal to one. This task, in turn, can be accomplished by showing that the norms of the linear maps L_1 and L_2 are less than or equal to one—or, equivalently, by showing that the largest singular value of these one-dimensional operators is less than or equal to one.

10.2 Evaluation of Singular-Values

In what follows we will numerically demonstrate that the singular values of L_1 and L_2, and therefore those of $\|\mathcal{L}_x^1\|_{\ell_2(\mathcal{D}_\Omega)}$, $\|\mathcal{L}_y^1\|_{\ell_2(\mathcal{D}_\Omega)}$, $\|\mathcal{L}_x^2\|_{\ell_2(\mathcal{D}_\Omega)}$, and $\|\mathcal{L}_y^2\|_{\ell_2(\mathcal{D}_\Omega)}$, are indeed less than one for all real values of $\alpha^2 = \frac{k\Delta t}{2}$ and relevant parameter values, as discussed in what follows.

The linear maps L_1 and L_2 are functions of the parameters n, h, n_Δ, m, d/Δ, g, Υ, $x_1 - x_\ell$, $x_r - x_n$ and α as explained in Chapters 5 and 6. In this thesis the values of n_Δ, d/Δ, g, and Υ have been fixed (see Remark 5.1.3); further, it is easy to check that the parameter $L = x_r - x_\ell$ can be scaled to $L = 1$ without loss of generality, and therefore we need only consider the effect of the parameters n, α, m, $\frac{x_1 - x_\ell}{h}$, and $\frac{x_r - x_n}{h}$. In Figures 10.1 and 10.2 we present the largest singular value of the linear map L_1 as a function of $(x_1 - x_\ell)/h$, for four different values of α and for several representative values of n with $x_r - x_n = 0$ and $m = 5$ (sixth-order accuracy); we see that the stability conditions

$$\|L_1\|_{\ell_2} \leq 1 \quad \text{and} \quad \|L_2\|_{\ell_2} \leq 1 \tag{10.6}$$

are satisfied: in all cases shown the largest singular value is less than one.

Remark 10.2.1. *In fact, in a similar manner we have verified that the stability conditions* (10.6) *hold for all $n \leq 200$ and many larger values of n, and, by means of adequately refined discretizations of the parameter ranges, for all other values of the parameters defining the algorithm. In what follows we present a number of additional figures that display a representative sample of the numerical experiments that we used for this verification.*

Note from Figures 10.1 and 10.2 that as n increases the dependence of the largest singular value on $(x_1 - x_\ell)/h$ decreases. Although the value of the largest singular value is quite close to one for small α, this quantity is essentially independent of n and $(x_1 - x_\ell)/h$.

Figure 10.3 shows a more general result for which $n = 50$ has been fixed but the result is shown as a function of both $(x_1 - x_\ell)/h$ and $(x_r - x_n)/h$ with $m = 5$ (sixth-order accuracy). The scale shows the one minus the largest singular value of L_1. The largest singular value occurs with both $(x_1 - x_\ell)/h = 1$ and $(x_r - x_n)/h = 1$ and, as before, that value changes

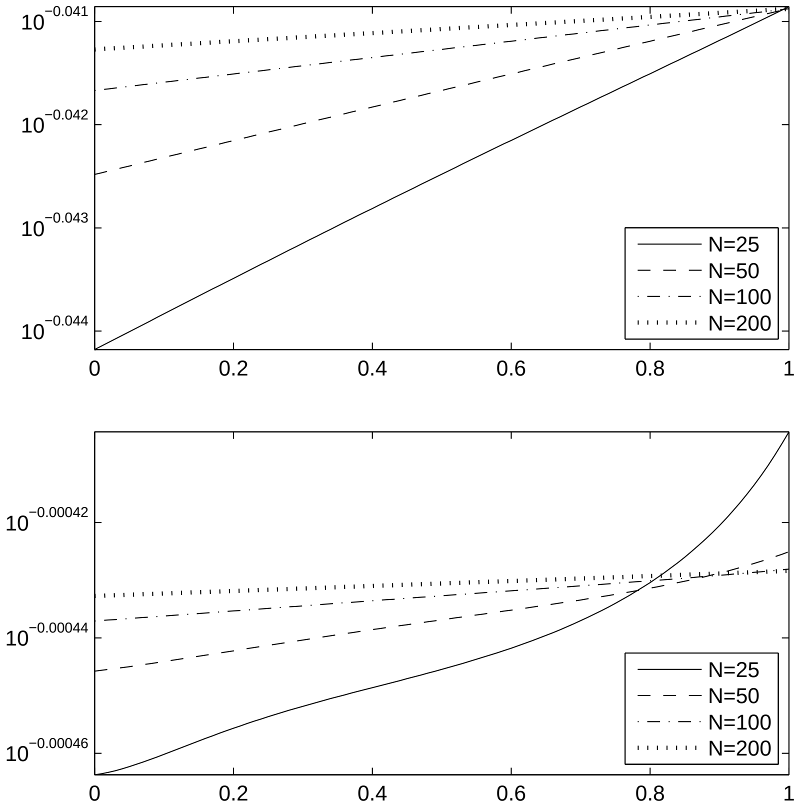


Figure 10.1: Largest singular value of the linear map L_1 as a function of $(x_1 - x_\ell)/h$ with $x_r - x_n = 0$ and $m = 5$ (sixth-order accuracy). The top and bottom plots assume $\alpha = 0.1$, and $\alpha = 0.01$, respectively.

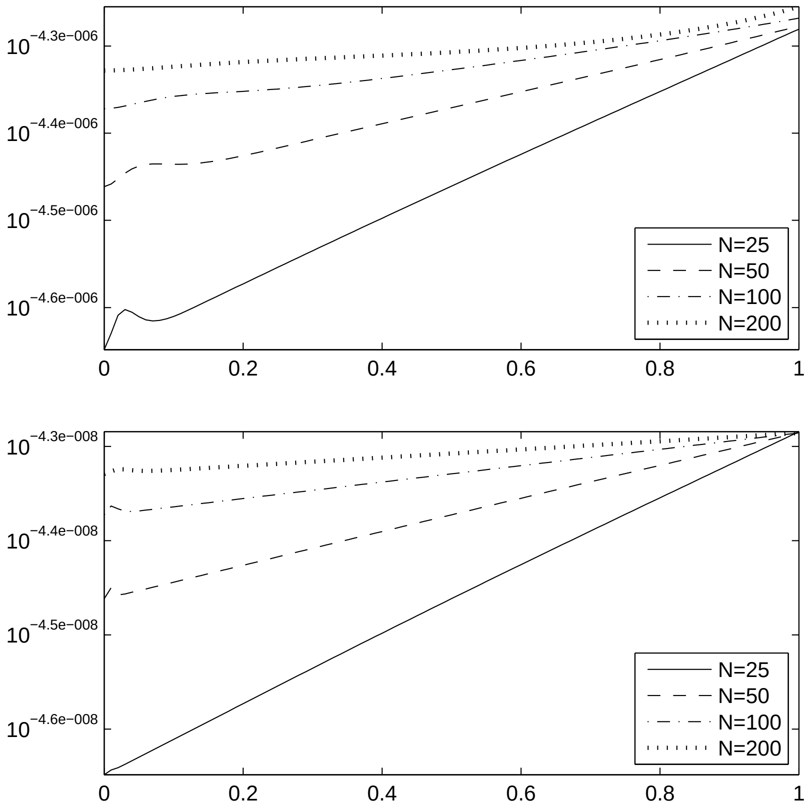


Figure 10.2: Largest singular value of the linear map L_1 as a function of $(x_1 - x_\ell)/h$ with $x_r - x_n = 0$ and $m = 5$ (sixth-order accuracy). The top and bottom plots assume $\alpha = 0.001$, and $\alpha = 0.0001$, respectively.

very little with $(x_1 - x_\ell)/h$ and $(x_r - x_n)/h$.

A more complete picture is given by Figure 10.4 which allows α to vary across a wide range of relevant values. For each value of α, the largest singular value was calculated from all possible combinations of a discrete set of 121 combinations of values for $(x_1 - x_\ell)/h$ and $(x_r - x_n)/h$. This process is shown for several values of n. In order to accurately display the result, the quantity one minus the largest singular value of L_1 is plotted in the figure. Figure 10.4 clearly demonstrates how little the largest singular value is affected by the value of n. Similar calculations were performed for a variety of other values of the parameters and in all cases the singular values were found to be less than one.

All of the calculations mentioned above have been repeated for the linear map L_2 and all the singular values remain less than 1 when $m = 4$. In particular, in Figure 10.5 we present L_2 data analogous to that presented in Figure 10.4 for the operator L_1: for each value of α, the largest singular value was calculated from the 121 combinations of values for $(x_1 - x_\ell)/h$ and $(x_r - x_n)/h$. Again, as in Figure 10.4, the process was repeated for several values of n.

The above calculations have been repeated with a larger values for m (degree of the polynomials used in the FC(Gram) approximation) and clearly show that the bound on the singular-values no longer holds. For example, if one were to attempt to apply the FC-AD algorithm with an polynomial approximation of one order higher than the prescribed (without changing any other parameter), the result is an algorithm that is unstable for a certain range of values of α: the algorithm becomes conditionally stable. Computational results are shown in Figure 10.6 and a corresponding instability for the full FC-AD solution have been observed in this case. Thus, for unconditional stability the current version of our FC-AD heat-equation and Poisson-equation solver is restricted to orders of accuracy ≤ 5. In the case of the Wave Equation, for which only the operators $\mathcal{L}_x^1$ and $\mathcal{L}_y^1$ are used, we have observed that the order of accuracy of the FC-AD algorithm can be ≤ 6 with unconditional stability.

We do not have a method to show directly the stability of the our FC-AD splitting of the Wave Equation (see Section 8.1) independently of the geometry as was done for the Heat Equation and the Poisson Equation. This is due to the specific stability requirement of the time stepping scheme. Using parameter values as prescribed above, our numerical results have never given rise to instability, for any value of Δt, despite millions of time-steps on

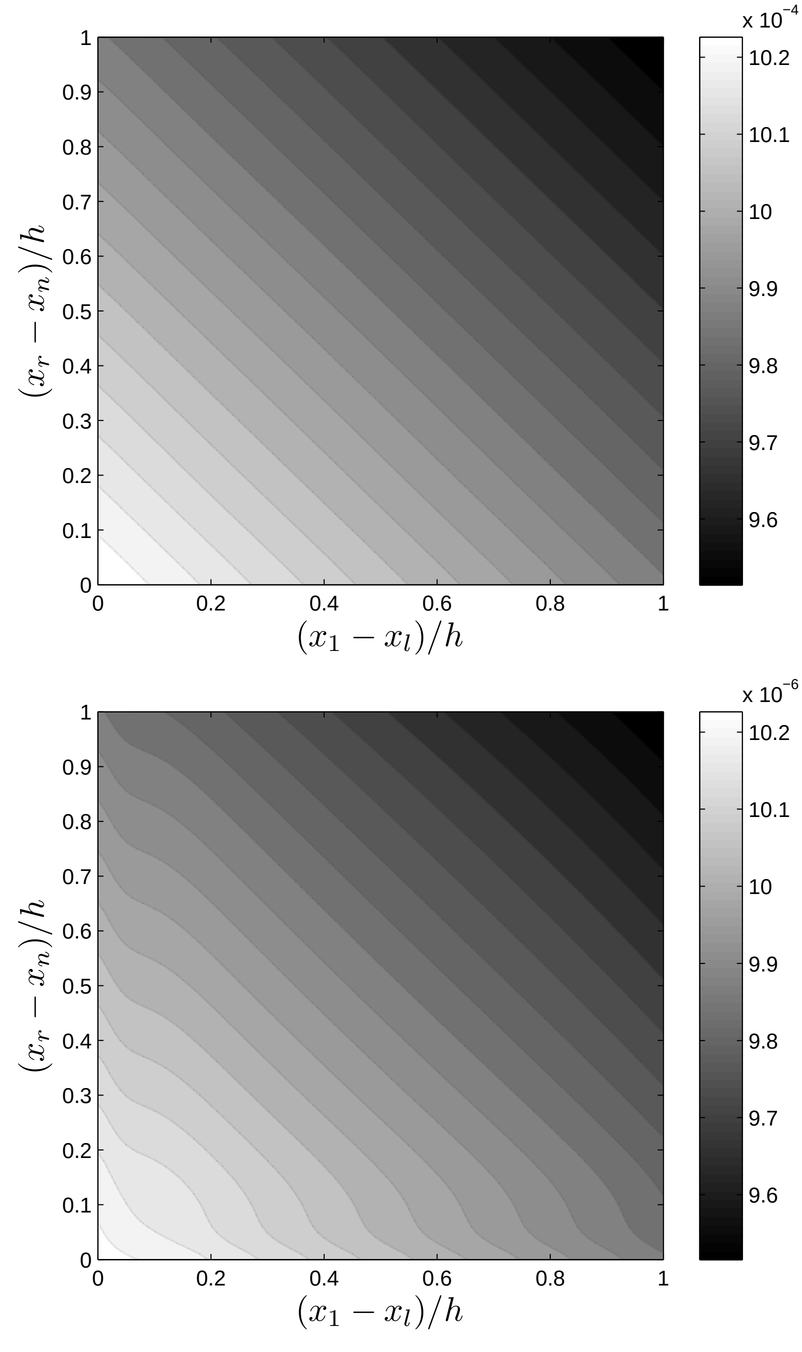


Figure 10.3: One minus the largest singular value of the linear map L_1 with $n = 50$, $m = 5$ (sixth-order accuracy); $\alpha = 0.01$ and $\alpha = 0.001$ in the top and bottom graphs, respectively.

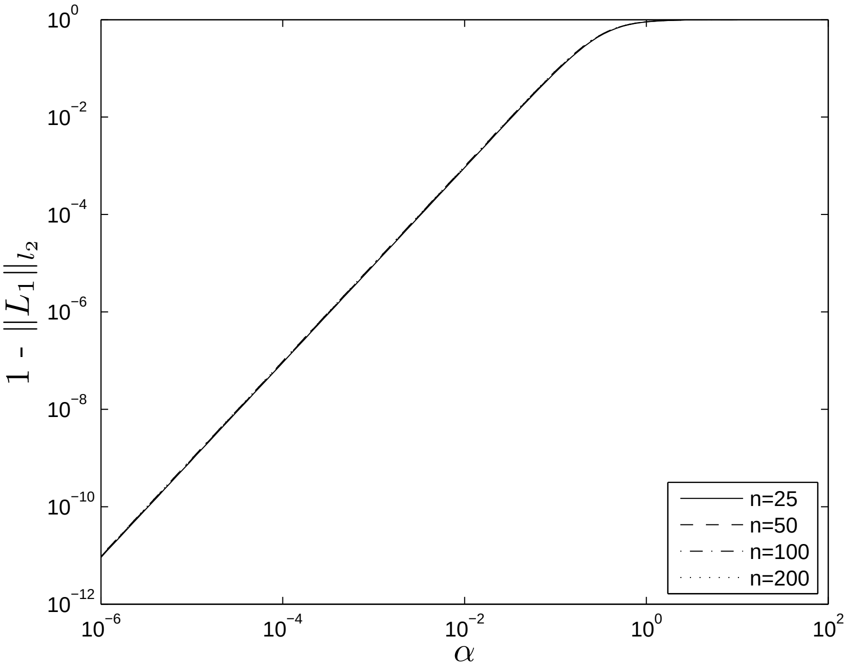


Figure 10.4: One minus the largest singular value of the linear map L_1 maximized over a discrete sampling of $(x_1 - x_\ell)/h$ and $(x_r - x_n)/h$ as a function of α with $m = 5$ (sixth-order accuracy).

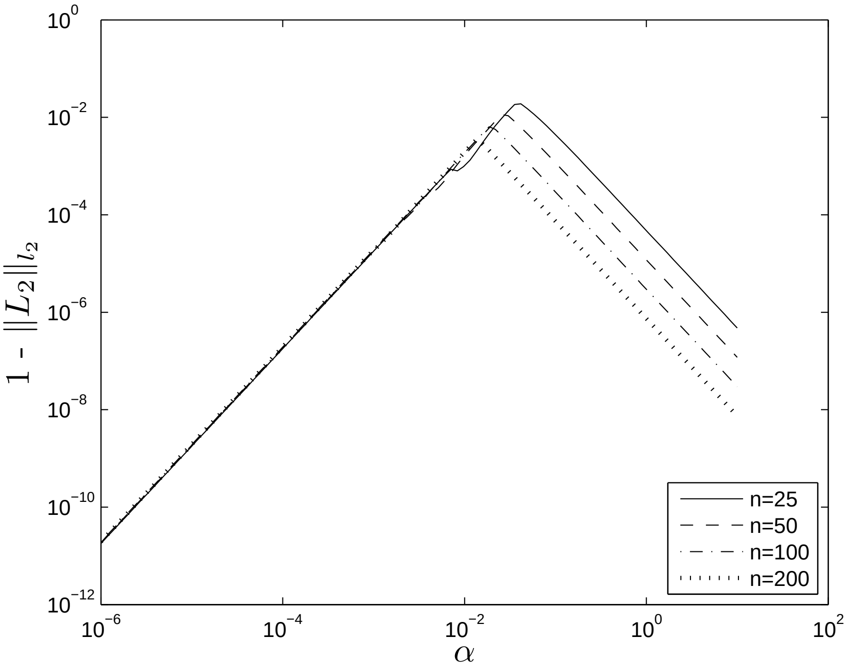


Figure 10.5: One minus the largest singular value of the linear map L_2 with $m = 4$ (fifth-order accuracy) maximized over a discrete sampling of $(x_1 - x_\ell)/h$ and $(x_r - x_n)/h$ as a function of α for several values of n.

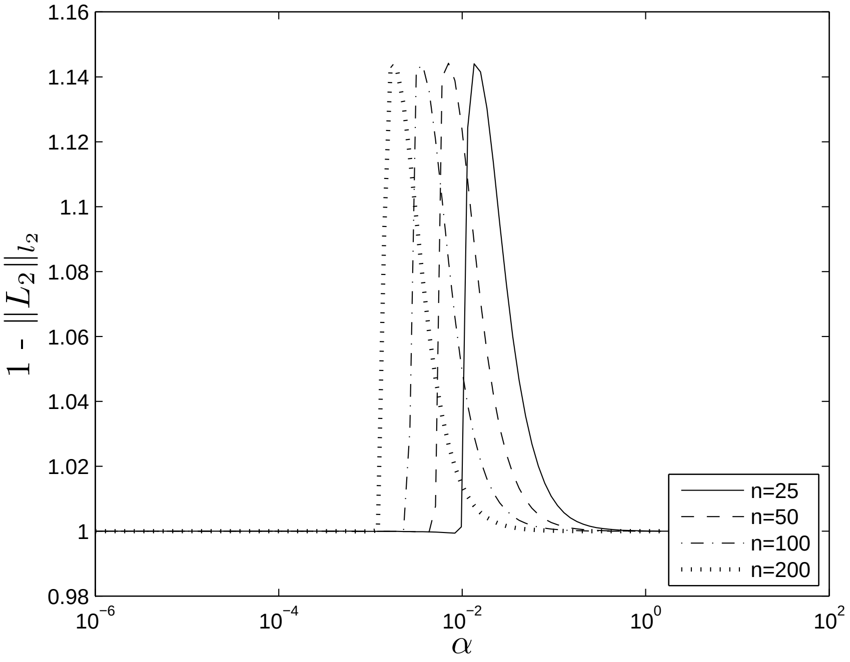


Figure 10.6: One minus the largest singular value of the linear map L_2 with $m = 5$ maximized over a discrete sampling of $(x_1 - x_\ell)/h$ and $(x_r - x_n)/h$ as a function of α for several values of n. Note that, for the present case $m = 5$, the stability condition $\|L_2\|_{\ell_2} \leq 1$ is violated. The FC-AD algorithm is unconditionally stable for spatial accuracy orders $m \leq 4$ for the Heat and Poisson Equations.

a range of geometries and spatial resolutions, as described in Chapter 8. For a sufficiently small sample geometry direct calculations of eigenvalues of the FC-AD iterative process have demonstrated stability regardless of the value chosen for the time-step.

10.3 A Note on Consistency and Convergence

In view of Lax Equivalence Theorem (c.f., [111]), consideration of the consistency of FC-AD algorithm would be a natural step towards a full convergence proof for the method. The matter is more delicate, however: the boundary correction and filtering steps in the FC-AD method give rise to truncation errors such that strict consistency for a single time step cannot be established. Instead, the convergence exhibited by the manifold examples, including those presented in Section 6.5, appears to arise from a cooperative handling of errors over subsequent time-steps, which does not give rise to error accumulation.

This is demonstrated in Figure 10.7 through a simple one-dimensional problem, essentially identical to a time discretization of a one-dimensional Heat Equation, using the FC-ODE solver at each time-step. The initial condition was given by

$$Z_0(x) = cos(\sqrt{10}\pi x), \tag{10.7}$$

and each subsequent iteration was given by

$$-\alpha^2 Z''_{n+1} + Z_{n+1} = Z_n, \tag{10.8}$$

with boundary conditions

$$Z_{n+1}(x_\ell) = \frac{1}{1+10\alpha^2\pi^2} Z_n(x_\ell) \quad \text{and} \quad Z_{n+1}(x_r) = \frac{1}{1+10\alpha^2\pi^2} Z_n(x_r), \tag{10.9}$$

thus the true solution at any time step is just a scaling of the initial condition. For each α the sequence was run up to a point for which the exact solution equals $0.8Z_0$. These calculations were performed using the FC-ODE algorithm with 20 total points in the spatial discretization and repeated for a dense sampling of values for the parameter $(x_1 - x_\ell)/h$ with $(x_1 - x_\ell)/h = (x_r - x_n)/h$. The maximum error over any set of boundary points is reported in Figure 10.7. The figure shows the error generated in the first iteration as

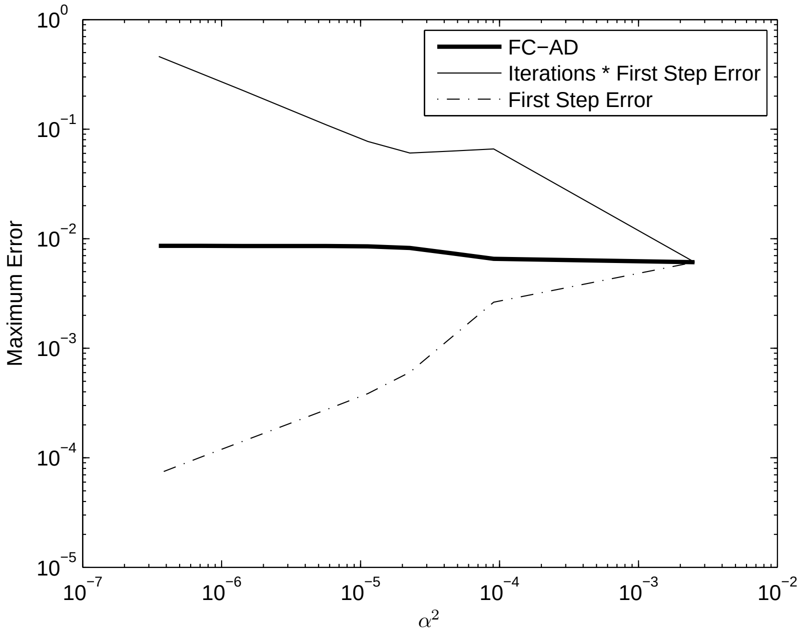


Figure 10.7: Sample calculations demonstrating the complexity inherent in the consistency property of the FC-AD algorithm. The error in the first step of an algorithm decays as α and not α^2, but this error does not accumulate over many iterations.

well as that amount multiplied by the total number of iterations required to reach the final solution. The latter diverges as the error after the first iteration decreases as α instead of as α^2. Despite this fact, the actual total error produced by the algorithm remains bounded and converges to a value dependent on the mesh spacing (hence can be reduced by spatial refinement).

Thus, a convergence proof strictly following the lines of the classical Lax Equivalence theorem does not seem possible for our approach, and, indeed, a convergence theory that fully explains the properties demonstrated by the algorithm has not yet been obtained. In particular, we note that the boundary conditions used for the Heat Equation produce a local error of $O(\Delta t^2)$ (See Remark 6.1.1) and yet result in a global error of $O(\Delta t^2)$ as shown in Chapter 6.We attribute both of these effects to the special handling of the boundary conditions within the FC-ODE algorithm.

Much progress and a significant and partially complete, if not fully adequate explanatory convergence result, can be obtained by using the ideas inherent in the Lax Equivalence Theorem. Indeed, for the Heat Equation, for example, the component of the local error $\tau^n(h, \Delta t)$ resulting from the time discretization is of the order of that given in Equation (6.19): it is not worse than $O(\Delta t^2)$. The result of the time discretization is a series of ODEs, which according to the results of Section 6.4, retains the order of accuracy of the FC(Gram) approximation—which was demonstrated in Chapter 5 to be essentially $O(h^{m+1})$ where, for the Heat Equation, we have $m = 4$. Thus, after $T/\Delta t$ iterations, we have that $\|\epsilon^{T\Delta t}\|_{\ell_2(\mathcal{D}_\Omega)}$ is of the order of $T\Delta t + Th^5/\Delta t$ and therefore as long $h^5 \rightarrow 0$ faster than Δt (not a very restrictive requirement), we have $\|\epsilon^{T\Delta t}\|_{\ell_2(\mathcal{D}_\Omega)} \rightarrow 0$ and thus the method is convergent.

While the estimate above is adequate for most purposes, our numerical tests indicate the condition is not required for convergence. This was shown in Section 6.5, where Δt was refined for fixed h and it was shown that the error did not grow as may be suggested by the estimate above (see Figure 6.9). Additionally we show a numerical example where we refine Δt proportionally to h^6 for the same problem considered in Figure 6.9. The result is shown in Figure 10.8 and demonstrates the continued convergence even in this case. Note that for $\Delta t \geq 10^{-3}$ the error is primarily due to the time-step. For $\Delta t \leq 10^{-4}$ the error is due to the spatial resolution.

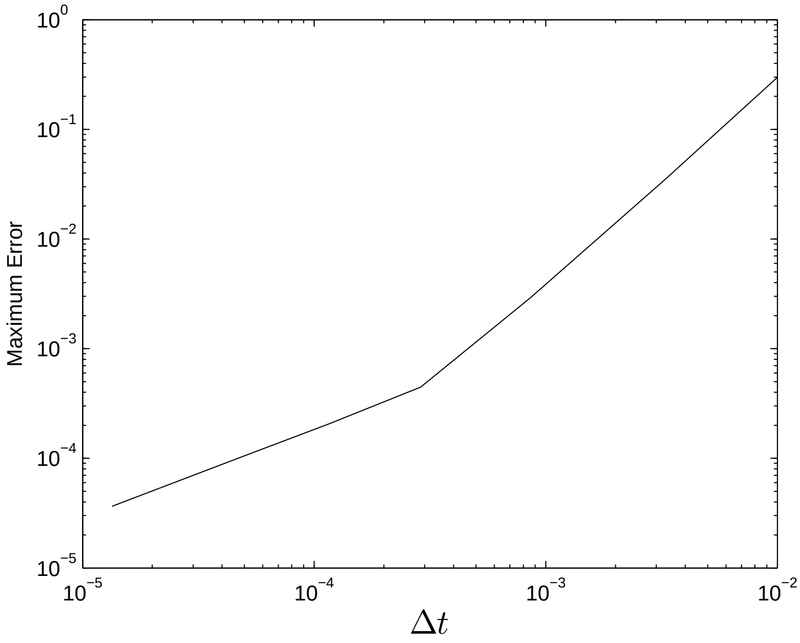


Figure 10.8: Demonstration of convergence for the case where Δt is taken proportional to h^6. The first calculation was performed with $\Delta t = 1.0 \cdot 10^{-2}$ (right-side of the figure) and $h = 6.0 \cdot 10^{-3}$. The last calculation was performed with $\Delta t = 1.3 \cdot 10^{-5}$ and $h = 2.0 \cdot 10^{-3}$ (left-side of the figure).

Chapter 11

Conclusions

A new class of fast and high-order accurate solvers for partial differential equations on general domains has been introduced. These methods combine the splitting of classical ADI schemes with a newly introduced Fourier continuation method. The resulting FC-AD algorithms have been demonstrated to yield accurate solutions for complex domains for PDEs of Parabolic, Hyperbolic and Elliptic type in two and three dimensions. The computational cost scales nearly linearly (FFT speeds) with the number of unknowns, and the methods have inherently low memory requirements that are proportional to the number of unknowns. These properties allow for rather large computations in short computational times, even in a single processing core. These methods represent, to the author's knowledge, by far the highest order spatially accurate unconditionally stable Alternating Direction schemes available for complex domains.

Bibliography

[1] S. Abarbanel and A. Ditkowski. Asymptotically stable fourth-order accurate schemes for the diffusion equation on complex shapes. *J. Comput. Phys.*, 133:279–288, 1997.

[2] S. Abarbanel, A. Ditkowski, and A. Yefet. Bounded error schemes for the wave equation on complex domains. *J. Sci. Comput.*, 26:67–81, 2006.

[3] S.S. Abarbanel and A.E. Chertock. Strict stability of high-order compact implicit finite-difference schemes: The role of boundary conditions for hyperbolic PDEs, I. *J. Comput. Phys.*, 160:42–66, 2000.

[4] S.S. Abarbanel, A.E. Chertock, and A. Yefet. Strict stability of high-order compact implicit finite-difference schemes: The role of boundary conditions for hyperbolic PDEs, II. *J. Comput. Phys.*, 160:67–87, 2000.

[5] G. Alefeld. On the convergence of the Peaceman-Rachford iterative method. *Numer. Math.*, 26:409–419, 1976.

[6] A. Averbuch, M. Israeli, and L. Vozovoi. A fast Poisson solver of arbitrary order accuracy in rectangular regions. *SIAM J. Sci. Comput.*, 19(3):933–952, May 1998.

[7] A. Averbuch and L. Vozovoi. Two-dimensional parallel solver for the solution of Navier-Stokes equations with constant and variable coefficients using ADI on cells. *Parallel Comput.*, 24:673–699, 1998.

[8] A. Averbuch, L. Vozovoi, and M. Israeli. On a fast direct elliptic solver by a modified Fourier method. *Numer. Algorithms*, 15:287–313, 1997.

[9] I. Babuska and T. Strouboulis. *The Finite Element Method and its Reliability.* Clarendon Press, Oxford, 2001.

[10] I. Babuska and T. Strouboulis. *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics.* Cambridge University Press, Cambridge, second edition, 2002.

[11] I.A. Babuska and S.A. Sauter. Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM J. Numer. Anal.*, 34(6):2392–2423, 1997.

[12] K.J. Badcock and B.E. Richards. Implicit time-stepping methods for the Navier-Stokes equations. *AIAA Journal*, 34(3):555–559, 1996.

[13] K. Balakrishnan, R. Sureshkumar, and P.A. Ramachandran. An operator splitting-radial basis function method for the solution of transient nonlinear Poisson problems. *Comput. Math. Appl.*, 43:289–304, 2002.

[14] R.W. Barnard, G. Dahlquist, K. Pearce, L. Reichel, and K.C. Richards. Gram polynomials and the kummer function. *J. Approx. Theory*, 94:128–143, 1998.

[15] G. Beylkin. On the fast Fourier transform of functions with singularities. *Appl. Comp. Harm. Anal.*, 2:363–381, 1995.

[16] B. Bialecki and G. Fairweather. Orthogonal spline collocation methods for partial differential equations. *J. Comput. Appl. Math.*, 128:55–82, 2001.

[17] B. Bialecki and R.I. Fernandes. An alternating-direction implicit orthogonal spline collocation scheme for nonlinear parabolic problems on rectangular polygons. *SIAM J. Sci. Comput.*, 28(3):1054–1077, 2006.

[18] G. Birkhoff and R.S. Varga. Implicit alternating direction methods. *T. Am. Math. Soc.*, 92(1):13–24, Jul. 1959.

[19] Å. Björck. *Numerical Methods for Least Squares Problems*, chapter 8, page 323. Society for Industrial and Applied Mathematics, Philadelphia, 1996.

[20] J.P. Boyd. A fast algorithm for chebyshev, Fourier, and sinc interpolation onto an irregular grid. *J. Comput. Phys.*, 103(2):243–257, 1992.

[21] J.P. Boyd. *Chebyshev and Fourier Spectral Methods: Second Revised Edition.* Dover Publications, New York, 1999.

[22] J.P. Boyd. A comparison of numerical algorithms for Fourier extension of the first, second, and third kinds. *J. Comput. Phys.*, 178:118–160, 2002.

[23] J.P. Boyd. Trouble with Gegenbauer reconstruction for defeating Gibbs phenomenon: Runge phenomenon in the diagonal limit of Gegenbauer polynomial approximations. *J. Comput. Phys.*, 204:253–264, 2005.

[24] E. Braverman, B. Epstein, M. Israeli, and A. Averbuch. A fast spectral subtractional solver for elliptic equations. *J. Sci. Comput.*, 21(1):91–128, Aug. 2004.

[25] E. Braverman, M. Israeli, and Averbuch A. A hierarchical 3-d direct Helmholtz solver by domain decomposition and modified Fourier method. *SIAM J. Sci, Comput.*, 26(5):1504–1524, 2005.

[26] E. Braverman, M. Israeli, A. Averbuch, and L. Vozovoi. A fast 3D Poisson solver of arbitrary order accuracy. *J. Comput. Phys.*, 144:109–136, 1998.

[27] O.P. Bruno. Fast, high-order, high-frequency integral methods for computational acoustics and electromagnetics. In M. Ainsworth, P. Davies, D. Duncan, P. Martin, and B. Rynne, editors, *Topics in Computational Wave Propagation Direct and Inverse Problems Series*, volume 31 of *Lecture Notes in Computational Science and Engineering*, 2003.

[28] O.P. Bruno, Y. Han, and M.M. Pohlman. Accurate, high-order representation of complex three-dimensional surfaces via Fourier-continuation analysis. *J. Comput. Phys.*, 227:1094–1125, 2007.

[29] M.J. Cantero, L. Moral, and L. Velazquez. Five-diagonal matrices and zeros of orthogonal polynomials on the unit circle. *Linear Algebra Appl.*, 362:29–56, 2003.

[30] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Fundamentals in Single Domains.* Scientific Computation. Springer, Berlin, 2006.

[31] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods Fundamentals in Single Domains*, chapter 5, page 272. Scientific Computation. Springer, Berlin, 2006.

[32] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods Fundamentals in Single Domains*, chapter 1, page 27. Scientific Computation. Springer, Berlin, 2006.

[33] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*. Scientific Computation. Springer, Berlin, 2007.

[34] M. Carpenter, D. Gottlieb, and S. Abarbanel. Time-stable boundary conditions for difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes. *J. Comput. Phys.*, 111:220236, 1994.

[35] M.H. Carpenter, D. Gottlieb, and A. Saul. The stability of numerical boundary treatments for compact high-order finite-difference schemes. *J. Comput. Phys.*, 108:272–295, 1993.

[36] R.-S. Chen, E. Yung, C.H. Chan, D.X. Wang, and D.G. Fang. Application of the SSOR preconditioned CG algorithm to the vector FEM for 3-D full-wave analysis of electromagnetic-field boundary-value problems. *IEEE T. Microw. Theory.*, 50(4):1165–1172, 2002.

[37] Z. Chen. *Finite Element Methods and Their Applications*. Scientific Computation. Springer, Berlin, 2005.

[38] Z. Chen. *Finite Element Methods and Their Applications*, chapter 1, pages 55–58. Scientific Computation. Springer, Berlin, 2005.

[39] C. Clavero, J.L. Gracia, and J.C. Jorge. A uniformly convergent alternating direction hodie finite difference scheme for 2d time-dependent convection-diffusion problems. *IMA. J. Numer. Anal.*, 26:155–172, 2006.

[40] T. Colonius and S.K. Lele. Computational aeroacoustics: Progress on nonlinear problems of sound generation. *Prog. Aerosp. Sci.*, 40:345–416, 2004.

[41] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, 19:297–301, 1965.

[42] K.D. Cooper, K.M. McArthur, and P.M. Prenter. Alternating direction collocation for irregular regions. *Numer. Meth. Part. D. E.*, 12:147–159, 1996.

[43] K.D. Cooper and P.M. Prenter. Alternating direction for separable elliptic partial differential equations. *SIAM J. Numer. Anal.*, 28(3):711–727, Jun. 1991.

[44] J. Douglas, Jr. Alternating direction methods for three space variables. *Numer. Math.*, 4:41–63, 1962.

[45] J. Douglas, Jr. and J.E. Gunn. A general formulation of alternating direction methods. Part I. Parabolic and hyperbolic problems. *Numer. Math.*, 6:428–453, 1964.

[46] J. Douglas, Jr. and S. Kim. Improved accuracy for locally one-dimensional methods for parabolic equations. *Math. Models Methods Appl. Sci.*, 11(9):1563–1579, 2001.

[47] J. Douglas, Jr. and D.W. Peaceman. Numerical solution of two-dimensional heat-flow problems. *A.I.Ch.E. J.*, 1(4):505–512, 1955.

[48] J. Douglas, Jr. and C.M. Pearcy. On convergence of alternating direction procedures in the presence of singular operators. *Numer. Math.*, 5:175–184, 1963.

[49] J. Douglas, Jr. and H.H. Rachford, Jr. On the numerical solution of heat conduction problems in two and three space variables. *T. Am. Math. Soc.*, 82(2):421–439, Jul. 1956.

[50] T.A. Driscoll and B. Fornberg. A padé-based algorithm for overcoming the Gibbs phenomenon. *Numer. Algorithms.*, 26:77–92, 2001.

[51] A.J. Duijndam and M.A. Schonewille. Nonuniform fast Fourier transform. *Geophysics*, 64(2):539–551, 1999.

[52] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM J. Sci. Comput.*, 14:1368–1393, 1993.

[53] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data, ii. *Appl. Comp. Harm. Anal.*, 2:85–100, 1995.

[54] W.R. Dyksen. Tensor product generalized ADI methods for separable elliptic problems. *SIAM J. Numer. Anal.*, 24(1):59–75, Feb. 1987.

[55] K.S. Eckhoff. On a high order numerical method for solving partial differential equations in complex geometries. *J. Sci. Comput.*, 12(2):119–138, 1997.

[56] K.S. Eckhoff. On a high order numerical method for functions with singularities. *Math. Comput.*, 67(223):1063–1087, July 1998.

[57] M. Elghaoui and R. Pasquetti. A spectral embedding method applied to the advection-diffusion equation. *J. Comput. Phys.*, 125:464–476, 1996.

[58] M. Elghaoui and R. Pasquetti. Mixed spectral-boundary element embedding algorithms for the Navier-Stokes equations in the vorticity-stream function formulation. *J. Comput. Phys.*, 153:82–100, 1999.

[59] L.C. Evans. *Partial Differential Equations*, chapter 2, pages 54–62. American Mathematical Society, Providence, 1998.

[60] G. Fairweather and A.R. Mitchell. A high accuracy alternating direction method for the wave equation. *J. I. Math. Appl.*, 1:309–316, 1965.

[61] J.A. Fessler and B.P. Sutton. Nonuniform fast Fourier transforms using Min-Max interpolation. *IEEE T. Signal Proces.*, 51(2), 2003.

[62] J. Fish and T. Belytschko. *A First Course in Finite Elements*, chapter 1, page 3. John Wiley and Sons Ltd., West Sussex, England, 2007.

[63] J. Fish and T. Belytschko. *A First Course in Finite Elements*, chapter 1, page 1. John Wiley and Sons Ltd., West Sussex, England, 2007.

[64] B. Fornberg. *A Practical Guide to Pseudospectral Methods.* Cambridge monographs on applied and computational mathematics. Cambridge University Press, Cambridge, 1996.

[65] B. Fornberg, J. Zuev, and J. Lee. Stability and accuracy of time-extrapolated ADI-FDTD methods for solving wave equations. *J. Comput. Appl. Math.*, 200:178–192, 2007.

[66] J. Geer. Rational trigonometric approximations to piece-wise smooth periodic functions. *J. Sci. Comput.*, 10:325–356, 1995.

[67] J. Geer and N. Banerjee. Exponentially accurate approximations using Fourier series paritial sums. *J. Sci. Comput.*, 12:253–287, 1997.

[68] A. Gelb and J. Tanner. Robust reprojection methods for the resolution of the Gibbs phenomenon. *Appl. Comput. Harmon. Anal.*, 20:3–25, 2006.

[69] R. Glowinski, T.-W. Pan, and J. Periaux. A fictitious domain method for Dirichlet problem and applications. *Comput. Method. Apply. M.*, 111:283–303, 1994.

[70] G.H. Golub and C.F. Van Loan. *Matrix Computations*, chapter 4, pages 183–188. The John Hopkins University Press, Baltimore, third edition, 2006.

[71] J. Gong and J. Nordstrom. A stable and efficient hybrid scheme for viscous problems in complex geometries. *J. Comput. Phys.*, 226:1291–1309, 2007.

[72] D. Gottlieb and S.A. Orszag. *Numerical Analysis of Spectral Methods: Theory and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, 1977.

[73] D. Gottlieb and C.-W. Shu. On the Gibbs phenomenon and its resolution. *SIAM Rev.*, 39(4):644–668, 1997.

[74] D. Gottlieb and C.-W. Shu. A general theory for the resolution of the Gibbs phenomenon. In *Tricomi's Ideas and Contemporary Applied Mathematics*, volume 147 of *Atti dei Convegni Lincei*, Rome, 1998. Academia Nazionale dei Lincei.

[75] D. Gottlieb, C.-W. Shu, A. Solomonoff, and H. Vandeven. On Gibbs phenomenon i: Recovering exponential accuracy from the Fourier partial sum of a nonperiodic analytic function. *J. Comput. Appl. Math.*, 43:81–98, 1992.

[76] D. Gottlieb and E. Tadmor. Recovering pointwise values of discontinuous data within spectral accuracy. In E.M. Murman and S.S. Abarbanel, editors, *Progress and Supercomputing in Computational Fluid Dynamics*, pages 357–375, Boston, 1985. Birkhauser.

[77] W.H. Guilinger, Jr. The Peaceman-Rachford method for small mesh increments. *J. Math. Anal. Appl.*, 11:261–277, 1965.

[78] J.E. Gunn. On the two-stage iterative method of Douglas for mildly nonlinear elliptic difference equations. *Numer. Math.*, 6:243–249, 1964.

[79] T. Hagstrom and G. Hagstrom. Grid stabilization of high-order one-sided differencing i:first-order hyperbolic systems. *J. Comput. Phys.*, 223:316–340, 2007.

[80] W.D. Henshaw and D.W. Schwendeman. An adaptive numerical scheme for high-speed reactive flow on overlapping grids. *J. Comput. Phys.*, 191:420–447, 2003.

[81] W.H. Hundsdorfer and Verwer J.G. Stability and convergence of the Peaceman-Rachford ADI method for initial-boundary value problems. *Math. Comput.*, 53(187):81–101, Jul. 1989.

[82] M. Israeli, L. Vozozoi, and A. Averbuch. Spectral multidomain technique with local Fourier basis. *J. Sci. Comput.*, 8(2):135–149, 1993.

[83] D. Jackson. *The Theory of Approximation*, volume 11 of *American Mathematical Society Colloquium Publications*, chapter 4, pages 119–123. American Mathematical Society, New York, 1930.

[84] L. Jameson. High order schemes for resolving waves: Number of points per wavelength. *J. Sci. Comput.*, 15(4):417–433, 2000.

[85] N.V. Kantartzis, T.T. Zygiridis, and T.D. Tsiboukis. An unconditionally stable higher order ADI-FDTD technique for the dispersionless analysis of generalized 3-D EMC structures. *IEEE T. MAGN.*, 40(2):1436–1439, March 2004.

[86] S. Karaa and J. Zhang. High order ADI method for solving unsteady convection-diffusion problems. *J. Comput. Phys.*, 198:1–9, 2004.

[87] M. Lees. Alternating direction methods for hyperbolic differential equations. *J. Soc. Ind. Appl. Math.*, 10(4):610–616, Dec. 1962.

[88] J.R. Li and L. Greengard. High order marching schemes for the wave equation in complex geometry. *J. Comput. Phys.*, 198:295–309, 2004.

[89] M.N. Linnick and H.F. Fasel. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *J. Comput. Phys.*, 204:157–192, 2005.

[90] B. Lombard, J. Piraux, C. Gelis, and J. Virieux. Free and smooth boundaries in 2-d finite-difference schemes for transient elastic waves. *Geophys. J. Int.*, 172:252–261, 2008.

[91] R.E. Lynch and J.R. Rice. Convergence rates of ADI methods with smooth initial error. *Math. Comput.*, 22(102):311–335, Apr. 1968.

[92] R.E. Lynch, J.R. Rice, and D.H. Thomas. Direct solution of partial difference equations by tensor product methods. *Numer. Math.*, 6:185–199, 1964.

[93] A. Majda, J. McDonough, and S. Osher. The Fourier method for nonsmooth initial data. *Math. Comput.*, 32(144):1041–1081, Oct. 1978.

[94] A.R. Mitchell and G. Fairweather. Improved forms of the alternatin direction methods of Douglas, Peaceman and Rachford for solving parabolic and elliptic equations. *Numer. Math.*, 6:285–292, 1964.

[95] M.S. Mock and P.D. Lax. The computational of discontinuous solutions of linear hyperbolic equations. *Commun. Pur. Appl. Math.*, 31:423–430, 1978.

[96] R.K. Mohanty and M.K. Jain. An unconditionally stable alternating direction implicit scheme for the two space dimensional linear hyperbolic equation. *Numer. Meth. Part. D. E.*, 17:684–688, 2001.

[97] K.W. Morton and D.F. Mayers. *Numerical Solution of Partial Differential Equations*, chapter 3, page 77. Cambridge University Press, Cambridge, second edition, 2005.

[98] K.W. Morton and D.F. Mayers. *Numerical Solution of Partial Differential Equations*, chapter 2, pages 33–38. Cambridge University Press, Cambridge, second edition, 2005.

[99] K.W. Morton and D.F. Mayers. *Numerical Solution of Partial Differential Equations*, chapter 3, pages 70–71. Cambridge University Press, Cambridge, second edition, 2005.

[100] O.F. Næss and K.S. Eckhoff. A modified Fourier Galerkin method for the Poisson and Helmholtz equations. *J. Sci. Comput.*, 17:529–539, 2002.

[101] T. Namiki. A new FDTD algorithm based on alternating-direction implicit method. *IEEE T. Microw. Theory.*, 47(10):2003–2007, Oct. 1999.

[102] P. Olsson. Summation by parts, projections and stability i. *Math. Comp.*, 64:10351065, 1995.

[103] P. Olsson. Summation by parts, projections and stability ii. *Math. Comp.*, 64:14731493, 1995.

[104] S.A. Orszag. Spectral methods for problems in complex geometries. *J. Comput. Phys.*, 37:70–92, 1980.

[105] D.W. Peaceman and H.H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Ind. Appl. Math.*, 3(1):28–41, March 1955.

[106] George M. Phillips. *Interpolation and Approximation by Polynomials*, chapter 2, pages 87–100. CMS Books in Mathematics. Springer, New York, 2003.

[107] George M. Phillips. *Interpolation and Approximation by Polynomials*, chapter 2, pages 100–116. CMS Books in Mathematics. Springer, New York, 2003.

[108] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes: The Art of Scientific Computing*, chapter 12, pages 640–719. Cambridge University Press, Philadelphia, third edition, 2007.

[109] A. Quarteroni, S. Riccardo, and F. Saleri. *Numerical Mathematics*, chapter 7, pages 331–332. Texts in Applied Mathematics. Springer, New York, 2000.

[110] A. Quarteroni, S. Riccardo, and F. Saleri. *Numerical Mathematics*, chapter 3, pages 90–93. Texts in Applied Mathematics. Springer, New York, 2000.

[111] R.D. Richtmyer and K.W Morton. *Difference methods for initial-value problems.* Robert E. Krieger Publishing Co., Inc., Malabar, FL, reprint of the second edition, 1994.

[112] Y. Saad. *Iterative Methods for Sparse Linear Systems.* SIAM, Philadelphia, second edition, 2003.

[113] Y. Saad and H. A. van der Vorst. Iterative solution of linear systems in the 20th century. *J. Comput. Appl. Math.*, 123:1–33, 2000.

[114] M. Schatzman. Stability of the Peaceman-Rachford approximation. *J. Funct. Anal.*, 162:219–255, 1999.

[115] S. Schmidt and G. Lazzi. Extension and validation of a perfectly matched layer formulation for the unconditionally stable D-H FDTD method. *IEEE Microw. Wirel. Co.*, 13(8):345–347, 2003.

[116] Y. Shi, L. Li, and C.H. Liang. The ADI multi-domain pseudospectral time-domain algorithm for 2-d arbitrary inhomogeneous media. *J. of Electromagn. Waves and Appl.*, 19(4):543–558, 2005.

[117] R.K. Shukla, M. Tatineni, and X. Zhong. Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier-Stokes equations. *J. Comput. Phys.*, 224:1064–1094, 2007.

[118] Bulirsch R. Stoer J. *Introduction to Numerical Analysis*, chapter 8, pages 601–602. Springer, New York, second edition, 2004.

[119] Bulirsch R. Stoer J. *Introduction to Numerical Analysis*, chapter 8, page 597. Springer, New York, second edition, 2004.

[120] B. Strand. Summation by parts for finite difference approximations for d/dx. *J. Comput. Phys.*, 110:4767, 1994.

[121] G. Strang and G.J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, Englewood Cliffs, N.J., 1973.

[122] W.T. Strickland and S.H. Davis. Transient response of the fluid surrounding a hot wire. *Industrial and Engineering Chemistry Fundamentals*, 5(1):38–42, 1966.

[123] J.C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations.* SIAM, Philadelphia, second edition, 2004.

[124] J.C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*, chapter 1, pages 34–36. SIAM, Philadelphia, second edition, 2004.

[125] J.C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*, chapter 7, pages 172–177. SIAM, Philadelphia, second edition, 2004.

[126] J.C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*, chapter 14, page 383. SIAM, Philadelphia, second edition, 2004.

[127] J. Tanner. Optimal filter and mollifier for piecewise smooth spectral data. *Math. Comp.*, 75:767–790, 2006.

[128] Z.F. Tian and Y.B. Ge. A fourth-order compact ADI method for solving two-dimensional unsteady convection-diffusion problems. *J. Comput. Appl. Math.*, 198:268–286, 2007.

[129] R.S. Varga. *Matrix Iterative Analysis.* Series in Automatics Computation. Prentice-Hall, Englewood Cliffs, N.J., 1962.

[130] L. Vozovoi, M. Israeli, and A. Averbuch. Analysis and application of Fourier-Gegenbauer method to stiff differential equations. *SIAM J. Numer. Anal.*, 33(5):1844–1863, Oct. 1996.

[131] L. Vozovoi, M. Israeli, and A. Averbuch. Multidomain local Fourier method for PDEs in complex geometries. *J. Comput. Appl. Math.*, 66:543–555, 1996.

[132] L. Vozovoi, A. Weill, and M. Israeli. Spectrally accurate solution of nonperiodic differential equations by the Fourier-Gegenbauer method. *SIAM J. Numer. Anal.*, 34(4):1451–1471, Aug. 1997.

[133] E.L. Wachspress. *Iterative Solution of Elliptic Systems and Application to the Neutron Diffusion Equations of Reactor Physics.* Prentice-Hall, Englewood Cliffs, N.J., 1971.

[134] S. Wang, F.L. Teixeira, and J. Chen. An iterative adi-fdtd with reduced splitting error. *IEEE Microw. Wireless Compon. Lett.*, 15(2):92–94, 2005.

[135] J.A.C. Weideman and S.C. Reddy. A MATLAB differentiation matrix suite. *ACM T. MATH. SOFTWARE*, 26(4):465–519, December 2000.

[136] O.B. Widlund. On the rate of convergence of an alternating direction implicit method in a noncommutative case. *Math. Comput.*, 20(96):500–515, Oct. 1966.

[137] O.B. Widlund. On difference methods for parabolic equations and alternating direction implicit methods for elliptic equations. *IBM J. Res. and Dev.*, 11(2):239–243, 1967.

[138] O.B. Widlund. On the effects of scaling of the Peaceman-Rachford method. *Math. Comput.*, 25(113):33–41, Jan. 1971.

[139] D. You. A high-order Padé ADI method for unsteady convection-diffusion equations. *J. Comput. Phys.*, 214:1–11, 2006.

[140] D.M. Young. *Iterative Solution of Large Linear Systems.* Academic Press, New York, 1971.

[141] G. Zhao and Q.H. Liu. The unconditionally stable pseudospectral time-domain (pstd) method. *IEEE Microw. Wirel. Co.*, 13(11):475–477, 2003.

[142] F. Zheng, Z. Chen, and J. Zhang. A finite-difference time-domain method without the Courant stability conditions. *IEEE Microw. Guided. W.*, 9(11):441–443, Nov. 1999.