High-order Numerical Methods for Parabolic PDEs with Nonsmooth Data: A Perspective from Option Pricing

by

Dawei Wang

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy

Graduate Department of Computer Science
University of Toronto

High-order Numerical Methods for Parabolic PDEs with Nonsmooth Data:
A Perspective from Option Pricing

Dawei Wang
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
2024

Abstract

High-order methods for solving partial differential equations (PDEs) enjoy, at least asymptotically, better performance compared to low-order methods, when the performance is measured by the ratio of error to computational cost. However, when there is nonsmoothness in the solution, as seen in many real-world applications, it generally requires careful development to solve the PDEs to a high-order accuracy. In this thesis, we focus on developing high-order numerical methods for solving parabolic PDEs with nonsmooth data from a perspective of option pricing. We study separately the convergence behaviors when the solution contains an unknown free boundary on which it is nonsmooth (the so-called free boundary problems), and when the initial conditions are nonsmooth.

For free boundary problems, the finite difference approximations of derivatives of a nonsmooth function exhibit degenerated orders of accuracy. We propose a high-order deferred correction algorithm combined with penalty iterations, and show that the order of convergence of the solution can be increased to fourth-order by solving successively corrected finite difference systems, where the corrections are derived from the previously computed lower order solutions, and applied solely to the right-hand side of the linear system.

For handling nonsmooth initial conditions, while still obtaining high-order time stepping, we propose to discretize the initial conditions with appropriate high-order smoothing schemes, and apply BDF4 time marching initialized with two steps of an explicit third-order Runge-Kutta (RK3) method and one step of BDF3 (2RK3-BDF3-BDF4). From the Fourier analysis of the discrete system, we prove that, for nonsmooth data, the low-order errors in the high-frequency domain are exponentially damped away by BDF steps, while the persisting low-order errors in the low-frequency domain can be eliminated by smoothing techniques, e.g., convolution-based techniques.

In addition, we derive novel smoothing techniques that cancel out the low-order terms of the quantization errors in the Fourier domain arising from discretization. Furthermore, we show how

to flexibly apply smoothings to general nonsmooth (but piece-wise smooth) initial condition discretizations, not only on uniform, but also on nonuniform grids.

Abundant numerical examples are presented to numerically support the high-order convergence of the proposed algorithms in the thesis.

To my family

Acknowledgements

It has been a long and winding journey. This thesis is an outcome of the help of great people along the way who have made this possible. There are many people and many things that I feel grateful for, and I could not possibly write them all.

This work could not be done without the support and guidance of my two amazing supervisors, Prof. Christina Christara and Prof. Kirill Serkh. They are always available to provide timely help whenever I have questions, while giving me enough freedom at the same time. I cannot thank them both too much for the many long meetings they had with me clearing up my questions and correcting my papers. Prof. Christara always makes sure every detail of my work is correct. She would spend hours and hours proofreading my draft papers, making sure my results are valid, providing valuable suggestions on improvements. Her rigorous attitude towards research makes me a better researcher. She also really cares about my career development and would take the time to help improve my application materials for jobs, scholarships and conferences. Prof. Serkh gave me the opportunity to come to UofT and work with him, which I really appreciate. Discussions with Prof. Serkh about math are always interesting and inspiring. I can often learn something new from our meetings. He gave me many insightful suggestions on our work and taught me how to tackle a complex problem step by step. He is also very easy going and makes his students feel like a friend. He even introduced me to his doctor friends when I had a bone injury and needed advice. I am really grateful to both of them for agreeing to accept me and working out a solution with the department, when I decided to transfer to a different research group in my first semester. Even though I moved to Toronto with the goal to study ML and HPC but had to give it up later, my time here has not been any less worthwhile, because of them. I am so fortunate to have had both of them as my Ph.D. advisors. Words cannot express my gratitude!

I thank Prof. Kenneth Jackson for being my committee chair and hosting all my checkpoints, and for reviewing my term reports and providing valuable feedback on my work during the past four years. My thanks also go to Prof. Justin Wan and Prof. Tom Fairgrieve for agreeing to join my final oral exam committee. Their time is very much appreciated.

I thank my M.Sc. supervisors Prof. Scott MacLachlan and Prof. Ronald Haynes for introducing me to the field of numerical analysis and scientific computing. Their mentorship on my projects there made me a competent researcher and prepared me well for Ph.D. study. I especially appreciate Prof. MacLachlan for kindly navigating me through the difficult situation in my first semester, which I will always be thankful to. Without his advice, I am not sure how I would continue my study at UofT.

I owe my gratitude to Prof. Hans De Sterck. With his guidance, I published my first paper in scientific computing. His passion for research still inspires me. His level of adeptness at presenting concepts and ideas is something I only hope to reach someday. I am indebted to his mentorship during my study in Waterloo and I really appreciate his understanding of my decisions.

I was also fortunate to have two industry experiences in my field of work during my Ph.D. I thank Yan, Charles, Mathieu, Chao and Nat for giving me these internship opportunities, and teaching me many useful skills in the financial industry.

I thank my friends and labmates Bangtian, Yunhui, Yuwei, Zewen, Ray and Mohan for our conversations about research, career, life or just random things. Ph.D. study can feel hard sometimes,

not just in research itself, it is wonderful to have them to chat with from time to time.

My time in Toronto became more fun thanks to the UofT nordic ski team. Alyssa is a great coach! I really enjoyed the fun night runs and ski trips. Gliding on the snow is probably the most exciting thing to do outside of school in the Canadian winter.

I am also thankful to the Department of Computer Science for providing the funding support for my study. I thank the graduate program coordinator Kimberly Main for her assistance on my checkpoints and final oral exam scheduling.

Finally, I thank my parents for always supporting and believing in me, and for their investment in my growth and education. Although they disagree with my choices sometimes and wish that I did not study abroad, they still stand behind me. I would have not gone this far without their sacrifices.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Option pricing and parabolic PDEs

An option is a contract that gives its holder the right but not the obligation to buy or sell an underlying asset at a certain price, called the exercise price or strike price, at a future date within a specified period of time, called the expiration time or maturity time. It is an example of a derivative instrument, whose values depend on one or more securities or assets, called underlying assets. Options are used for both speculation and risk hedging. The simplest financial option is the European call (put) option that allows its holder to buy (sell) the underlying at a strike price K, only on the expiration time T. An American option is one that can be exercised at any time prior to the maturity. Other types of options include the so-called exotic or path-dependent options, whose values depend on the history of an asset price, not just its values at exercise. In this thesis, we are interested in the pricing of European and American options.

The final value of an option at expiration time is known, and its fair value $V(t, S)$ at current time t needs to be correctly determined so that it is arbitrage-free, i.e. it should not be possible to create portfolios of zero initial cost consisting of long and short positions in options and their underlyings, that have surely nonnegative values in time and make profits with a positive probability. Using this principle, an analytical valuation formula for options with the underlying price S at time t, constant risk-free interest rate r, dividend rate q, and volatility σ, was first derived in [3] as

$$V(t, S) = Se^{-q(T-t)}\mathcal{N}(d_+) - Ke^{-r(T-t)}\mathcal{N}(d_-) \tag{1.1}$$

for European call options, where $\mathcal{N}(\cdot)$ is the cumulative density function (CDF) of the standard normal distribution, and

$$d_\pm = \frac{\log(S/K) + (r - q \pm \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}.$$

(1.1) is the famous Black-Scholes formula in derivative pricing. Prices for the corresponding European put options can then be obtained from put-call parity.

No-arbitrage option prices can be also derived from the fundamental theorem of asset pricing, which states that there is no arbitrage opportunities in the market if and only if there exists at

least one equivalent martingale measure $\mathbb{Q}$ with numeraire B to the real world measure $\mathbb{P}$ such that the numeraire-normalized prices are $\mathbb{Q}$-martingales. In particular, supposing that the underlying asset price S follows some stochastic process $(S_t)_{0\leq t\leq T}$. We have in the risk-neutral measure

$$\frac{V(t,S_t)}{B(t)} = \mathbb{E}^{\mathbb{Q}}\left[\frac{V(T,S_T)}{B(T)}\middle|\mathcal{F}_t\right]$$

under certain integrability assumptions, where $\mathcal{F}_t$ is the filtration at time t, and the numeraire B is the riskless asset [41]. Assuming that the risk-free interest rate is $r(t,S_t)$, we obtain the risk-neutral option price in expectation form

$$V(t,S_t) = \mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^T r(s,S_s)ds}V(T,S_T)|\mathcal{F}_t\right], \tag{1.2}$$

which is a model-free formula. Note that S_t in the risk-neutral measure $\mathbb{Q}$ follows a relative but different price process from the real world price process. [51] The option price in the expectation form can be approximated with Monte Carlo (MC) simulation. MC simulation with (1.2) enjoys the advantage of implementation simplicity.

Alternatively, suppose that the underlying asset price process is given by

$$dS_t = \mu(t,S_t)dt + c(t,S_t)dW_t^{\mathbb{Q}}, \tag{1.3}$$

where $(W_t^{\mathbb{Q}})_{t\geq 0}$ is the Brownian motion in measure $\mathbb{Q}$, and $\mu(\cdot,\cdot)$ and $c(\cdot,\cdot)$ are deterministic functions. From the Feynman-Kac theorem and (1.2), under certain assumptions on $\mu(\cdot,\cdot)$ and $c(\cdot,\cdot)$, the option price satisfies the partial differential equation (PDE)

$$\frac{\partial V}{\partial t} + \frac{c(t,S)^2}{2}\frac{\partial^2 V}{\partial S^2} + \mu(t,S)\frac{\partial V}{\partial S} = r(t,S)V, \tag{1.4}$$

at any given time $0 \leq t \leq T$ with asset price S. In comparison to Monte Carlo simulation, with the PDE approach, both the solutions and solution derivatives, which are financially important hedging parameters, can be computed to a higher accuracy under similar computational cost. In local volatility models, $c(t,S) = \sigma(t,S)S$, and the volatility $\sigma(t,S)$ is a calibrated function such that the model-implied option prices match the observed option prices in the market. Moreover, when the initial distribution of S_t is known, the probability density function $P(t,S)$ of the underlying process S_t in (1.3) can be calculated by the Fokker-Planck equation [46]

$$\frac{\partial P(t,S)}{\partial t} = \frac{\partial^2}{\partial S^2}\left(\frac{c(t,S)^2}{2}P(t,S)\right) - \frac{\partial}{\partial S}\left(\mu(t,S)P(t,S)\right). \tag{1.5}$$

In the Black-Scholes-Merton (BSM) model, the underlying asset price follows a geometric Brownian motion (GBM) with $dS_t = \mu S_t dt + \sigma S_t dW_t$, where dW_t is the Brownian motion in real-world measure, and μ is the constant growth rate of S. This, after some arguments, gives rise to the well-known Black-Scholes PDE,

$$\frac{\partial V}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{1.6}$$

which can be viewed as a particular case of (1.4), and has the analytic solution (1.1) under the European call payoff.

To present the Black-Scholes PDE in a form convenient for numerical solution, from now on, let $t = T - t$ be the backward time, i.e., time from maturity. In this way $t = 0$ corresponds to the expiration time, and a terminal condition becomes an initial condition. *We use this time convention throughout the thesis.*
Thus, the Black-Scholes PDE (in backward time t) is

$$(\partial_t - \mathcal{L}_{BS})\, V = 0, \quad \mathcal{L}_{BS} \equiv \frac{\sigma^2 S^2}{2}\frac{\partial^2}{\partial S^2} + rS\frac{\partial}{\partial S} - r. \tag{1.7}$$

In (1.7), we used the notation ∂_t to denote partial derivative with respect to time, i.e., $\partial_t \equiv \frac{\partial}{\partial t}$. In the rest of the thesis, we will use the two notations interchangeably.

In stochastic volatility models, the variance $c(t, S)$ in (1.3) of the asset return is itself a stochastic process. For example, one of the most popular stochastic volatility models is the Heston model, which assumes that the variance ν of the asset returns follows a Feller process, and results in a two-dimensional PDE

$$\frac{\partial V}{\partial t} - rS\frac{\partial V}{\partial S} - \vartheta(\hat{\nu} - \nu)\frac{\partial V}{\partial \nu} - \frac{1}{2}\nu S^2\frac{\partial^2 V}{\partial S^2} - \varrho\iota\nu S\frac{\partial^2 V}{\partial S \partial \nu} - \frac{1}{2}\iota^2\nu\frac{\partial^2 V}{\partial \nu^2} + rV = 0, \tag{1.8}$$

where $\vartheta, \hat{\nu}, \iota, \varrho$ are parameters related to the Feller process, see e.g. [29]. Under certain assumptions, the resulting PDEs (1.4), (1.5), (1.7) and (1.8) are typically of parabolic type.

For American options, the holder has a right to exercise the contract at any time prior to maturity and receive a payoff $V^*(S)$. Consequently, there is an unknown optimal exercise boundary that needs to be determined in order to solve the problem. The American option pricing problem is a specific instance of a broader class of optimal stopping problems, which can be mathematically formulated using the Hamilton–Jacobi–Bellman equation. These equations manifest as nonlinear PDEs in many contexts. American option prices can also be described as a free boundary or obstacle problem. Within such frameworks, the American option prices can be approximated by solving a penalized Black-Scholes PDE,

$$\frac{\partial V}{\partial t} = \mathcal{L}_{BS}V + \rho \max\{V^* - V, 0\}. \tag{1.9}$$

This formulation is a nonlinear parabolic PDE [21, 60]. The penalty term, $\rho \max\{V^* - V, 0\}$, enforces the constraint associated with the optimal exercise boundary. This approach is employed in the thesis to model American option prices. Additional details about the penalty formulation will be provided in Chapter 2.

In this dissertation, we study high-order finite difference methods for the simplified Black-Scholes PDE (1.7). Although the Black-Scholes model relies on some assumptions that do not reflect the real-world market, it is still a foundation model in financial mathematics. Moreover, from a mathematical perspective, the techniques that we introduce for constructing high-order methods for option pricing under the Black-Scholes model extend readily to more general non-constant coefficient parabolic PDEs, including, for example (1.4), (1.5), and (1.8).

1.2 Typical option payoffs

Different European/American options are distinguished via their payoffs. The most commonly seen payoffs include digital call/put, call/put, bull/bear spread and butterfly spread. For convenience, define the Heaviside function

$$H(S) \equiv \begin{cases} 1, & S \geq 0, \\ 0, & \text{else.} \end{cases} \tag{1.10}$$

A digital call/put option of strike K is a cash-or-nothing binary option that gives its holder one dollar if the asset prices S falls above/below K. Their payoff functions are

$$G_K^{H_c}(S) \equiv H(S-K) \tag{1.11}$$

for a digital call, and

$$G_K^{H_p}(S) \equiv 1 - H(S-K) \tag{1.12}$$

for a digital put. As introduced earlier, a European call/put option with strike K gives its holder the right to buy/sell the underlying asset with price K at exercise, with the payoff functions

$$G_K^C(S) \equiv \max\{S-K, 0\} \tag{1.13}$$

for a call, and

$$G_K^P(S) \equiv \max\{K-S, 0\} \tag{1.14}$$

for a put. Given $0 < K_1 < K_2$, a bull spread option can be seen as a portfolio consisting of a long position in a call strike at K_1, and a short position in a call strike at K_2, with the payoff function

$$G_{K_1,K_2}^{\text{bull}}(S) = G_{K_1}^C(S) - G_{K_2}^C(S). \tag{1.15}$$

A bear spread option can be seen as a portfolio consisting of a long position in a put strike at K_2, and a short position in a put strike at K_1, with the payoff function

$$G_{K_1,K_2}^{\text{bear}}(S) = G_{K_2}^P(S) - G_{K_1}^P(S). \tag{1.16}$$

A butterfly spread option can be seen as a portfolio consisting of a long position in a call strike at $K_1 = K - \mathcal{B}$, two short positions in a call strike at $K_2 = K$, and a long position in a call strike at $K_3 = K + \mathcal{B}$, where $\mathcal{B} > 0$ and $0 < K_1 < K_2 = \frac{1}{2}(K_1 + K_3) < K_3$. The payoff function is

$$G_{K,\mathcal{B}}^B(S) = G_{K_1}^C(S) - 2G_{K_2}^C(S) + G_{K_3}^C(S). \tag{1.17}$$

As we can see, all these payoff functions are nonsmooth at the strikes, with the digital call/put option being discontinuous, and other payoffs being $\mathcal{C}^0$.

1.3 Numerical solution methods

Typically, option pricing problems are solved using low-order methods, such as first-order lattice schemes [28, 58], Monte Carlo simulation and second-order finite difference/volume methods [25, 21], among others. In computational finance, we are interested not only in the solution itself, but also the solution derivatives for hedging purposes. Moreover, the point-wise accuracy, especially around the strike, matters a lot in financial applications. As mentioned earlier, the PDE approach allows for reliable computation of solution derivatives with higher accuracy compared to MC simulation and lattice tree methods. This dissertation focuses on the PDE approach, employing finite difference methods (FDMs) in particular, for solving option pricing problems.

Second-order FDMs for solving option pricing problems, and parabolic PDEs in general, have been well studied in the literature. In the case of European options, the main difficulty to achieve stable second-order convergence comes from nonsmoothness in the initial conditions. When applying Crank-Nicolson time stepping, which is known to be unconditionally stable in the L_2 norm, it has been noticed that the numerical solutions and/or their derivatives tend to exhibit spurious oscillations and degeneration of accuracy. To restore stable point-wise second-order convergence, it is a common practice to precede Crank-Nicolson time marching with two or more steps of backward Euler method, often with half step size, known as Rannacher's startup scheme [44]. Moreover, some form of smoothing to the initial condition is usually needed depending on the placement of the nonsmooth point on the space grid [7].

American options are particular examples of free boundary problems of parabolic type. Free boundary problems, in which both the solution to a PDE and the domain on which it is defined are unknowns to be solved for, arise in numerous applications of practical importance. Many free boundary problems, both with boundaries that move over time (also called moving boundary problems) and boundaries that are invariant with time, can be reformulated as linear complementarity problems (LCPs) (see, for example, §8.5 of [10]). Besides American option pricing problems, a few other well-known examples are the elliptic obstacle problem (see, for example, [47]), and the Stefan problem [48]. For such problems, in addition to solution nonsmoothness, the unknown optimal exercise boundary (or free boundary) also needs to be carefully dealt with. As we elaborate later, the LCP can be approximated by using a penalty method. Penalty methods for solving the American option problems fall under the broader category of fixed-domain methods, offering the advantage of algorithmic simplicity. With the penalty formulation in (1.9), the optimal exercise boundary is implicit in the equation, emerging automatically as part of the solution during the PDE solution process. The convergence of the penalty method with finite volume discretization and Crank-Nicolson time stepping has been investigated in [21]. The analysis technique can be applied to FDM as well [6]. Sufficient conditions are derived in [21, 6] to ensure monotone convergence, and an adaptive time step selector in [21] is suggested to restore stable quadratic convergence.

A more detailed literature review of numerical methods for solving the European and American option pricing problems is presented in Chapters 2 and 3.

1.4 Thesis contributions and outline

In this thesis, we focus on the development of high-order finite difference methods to find accurate option prices and hedging parameters. The initial motivation behind pursuing high-order methods stems from the desire to attain high accuracy and computational efficiency.

It may be reasonable to argue that exceptionally high accuracy in the solutions of pricing models is not imperative in finance, given the potential significance of modelling errors and large variability between different models. In practical financial applications, it is often believed that the solution accuracy obtained from low-order methods is adequate. Additionally, low-order methods enjoy a well-established understanding in the existing literature, leading to the dominant use of first- or second-order accurate methods in nearly all production codes for option pricing.

Indeed, on a given mesh, low-order methods may be more cost-effective to implement. However, the narrative changes when comparing the cost of low and high-order methods required to achieve the same level of accuracy. To obtain a specified level of accuracy with low-order methods, a fine grid discretization is often necessary, incurring a large computational expense. On the other hand, high-order PDE methods offer the advantage of achieving the same accuracy on a much coarser mesh. The difference becomes especially pronounced in multiple dimensions for multi-asset problems due to the curse of dimensionality. For example, assuming that the computational cost is proportional to the degrees of freedom in the system, when the mesh size and time step are reduced by half, the computational cost increases by a factor of roughly 16 (three spatial dimensions and one time dimension). Therefore, to reduce the error by a factor of 16, the computational cost increase by a factor of 256 for a second-order method, and only by 16 for a fourth-order one.

Hence, high-order PDE methods offer more flexibility for multi-dimensional problems, ensuring satisfactory accuracy with reasonable computational costs. The development of high-order finite difference methods in this thesis, while focusing on 1D only, is a crucial step to a better understanding of high-order methods in option pricing. It lays the foundation to apply high-order methods in multiple dimensions. We acknowledge that spectral methods do exist in option pricing, such as the well-known Fourier-cosine series expansion proposed in [17]. However, this method computes the solution only at a single point.

With that being said, solutions with high level of accuracy are necessary in many other scientific and engineering applications. The problems considered in this thesis have close analogues that appear in other areas of applied science, though we do not develop these other applications in this dissertation.

This thesis focuses on parabolic PDE problems arising from option pricing. With finite difference methods, the difficulties in obtaining high-order accuracy include: (i) The payoff functions or their derivatives are often discontinuous (i.e. nonsmooth), and (ii) the solution itself may contain an unknown free boundary, on which it is nonsmooth, and (iii) the free boundary moves infinitely quickly close to expiry, which causes the solution derivatives at the free boundary blow up at $t = 0$. Solutions of European options typically only have issue (i), while in the solutions of American options, all the three problems exist. In general, nonsmoothness poses a challenge to the development of high-order methods. In this thesis, we develop methods to deal with problems (i) and (ii) separately, and leave problem (iii) for future research.

There are three main contributions in this dissertation, which are given in order in Chapters 2, 3 and 4. In Chapter 2, we develop a high-order deferred correction algorithm combined with penalty iteration to solve free boundary problems, using fourth-order finite difference methods [60]. In particular, the American call/put option prices can be modelled as a moving boundary problem, and described by a linear complementarity problem. Under the LCP framework, the problem is defined on a fixed domain so that the unknown moving boundary is handled implicitly and comes out as part of the solution. As the solution has discontinuous derivatives at the moving boundary, a direct application of high-order methods will result in degenerated order of accuracy. Using a detailed error analysis, we observe that the order of convergence of the solution can be increased to fourth-order by solving successively corrected finite difference systems, where the corrections are derived from the previously computed lower order solutions. We show that our algorithms achieve high-order accuracy in both the solution and the unknown optimal exercise boundary. This contribution solves problem (ii).

In the development of high-order methods for free boundary problems, it is assumed that the initial conditions are smooth, and that problem (iii) does not exist, allowing for the separate study of problem (ii). These assumptions, however, do not hold for American put option prices. To address this, heuristics are applied by subtracting the corresponding European option price, effectively setting the initial condition to zero. This adjustment eliminates nonsmoothness in the payoff function, preventing the appearance of problem (i). Additionally, a time stretching around $t = 0$ is implemented to alleviate problem (iii).

Given these considerations, the forthcoming contribution in Chapter 3 focuses on problem (i). Specifically, it delves into high-order time-stepping methods for solving parabolic PDEs with nonsmooth initial conditions [59]. The challenge posed by nonsmooth initial conditions lies in the significant magnitudes of high-frequency components, whose false propagation to later time steps will lead to spurious oscillations in the solution around the singularity. This phenomenon is well understood when applying the second-order Crank-Nicolson method. In the context of high-order time stepping, we prove that performing the fourth-order backward differentiation formula (BDF4) effectively dampens the low-order errors in the high frequency domain. Simultaneously, the low-order errors in the low-frequency domain coming from the discretization error of the initial conditions persist (they are not damped out by BDF schemes) and can be eliminated by smoothing. BDF4 is initialized with a third-order Runge-Kutta scheme for the first two time steps, and BDF3 for the third time step. By incorporating fourth-order smoothed discrete initial conditions, we theoretically prove that our proposed time stepping scheme achieves stable fourth-order convergence.

A fourth-order convolution-type smoothing operator proposed in [31] is applied to the initial conditions in Chapter 3, which can be thought of averaging the nodal values over nearby points. However, this approach has certain limitations: it requires a uniformly discretized space domain, necessitates the smoothing of a fixed number of points on both sides of the singular point regardless of the function's regularity, and may be computationally expensive for complex nonsmooth functions. These drawbacks are the motivations to derive a different smoothing method. In the final contribution of this thesis, presented in Chapter 4, we develop a new smoothing technique that is flexible to handle complicated nonsmooth initial conditions, without the limitations of the existing approach. Specifically, we consider the Dirac delta, Heaviside, ramp and quadratic ramp

initial conditions. The Heaviside initial condition corresponds to the cash-or-nothing payoff, and the ramp initial condition corresponds to the call/put payoff. By linear combination of these basic singularities, we can model more general nonsmoothness. From Fourier analysis of the initial conditions, we derive correction schemes to cancel out low-order errors in the discretized initial conditions. Additionally, we derive formulas to apply the high-order correction scheme to any type of nonsmoothness on a nonuniform grid, as long as the function is analytic on each side of the singular point. This novel smoothing method provides flexibility and improved computational efficiency in handling complex nonsmooth initial conditions, addressing the limitations associated with the previous approach.

In Chapter 5, abundant numerical experiments are provided to support the theoretical analysis developed throughout the thesis. First, numerical results are presented for solving American put options and free boundary problems in general. We demonstrate results for solving an elliptic and a parabolic free boundary problem. For the American put options, two scenarios are considered, one with small volatility and the other with large volatility. Then we use our algorithms to solve European options covering various payoffs including digital call, call, and butterfly spread, along with convection-diffusion equations featuring general nonsmooth initial conditions. The solutions are first computed on uniform spatial grids, and then on nonuniform grids for comparison.

Finally in Chapter 6, we conclude the thesis with some possible generalizations and future research based on my current work.

Chapter 2

High-order methods for free boundary problems and American options

American option pricing problems are particular examples of free boundary problems, the defining feature of which is that the boundary of the domain is not known a priori and has to be determined as part of the solution, introducing an additional challenge to efficiently solving a PDE. Since the solution has discontinuous derivatives at the unknown free boundary, approximations to the solution derivatives around the nonsmooth points using finite difference methods will not have the expected high order of accuracy. In this chapter, we deal with these difficulties and develop a high-order deferred correction method under the LCP framework, such that both the solution and the free boundary location can be solved to a high accuracy at the same time.

Existing methods for solving free boundary problems can be classified into two categories: the front tracking methods and the fixed domain methods. Front tracking methods directly compute an approximation to the free boundary, either at each time step in time-dependent problems, or iteratively in time-independent problems (as in, for example, [50]). While the free boundary can be tracked parametrically or as an indicator function of some set, the most common approaches are the level set method (see, for example, [49] for a survey), in which the free boundary is represented as the zero level-set of a function which obeys an evolution equation, and the phase-field method, in which the free boundary is approximated by a finite-width region where a phase-field function smoothly changes sign across the region (see, for example, [1] for a survey). The front-fixing method, in which the free boundary PDE is transformed into a nonlinear PDE with a fixed boundary, is also considered to be a front tracking method (see, for example, [64]). Whatever the particular method used, front tracking requires the construction of a separate algorithm for approximating the front, derived from the underlying equations and constraints of the free boundary problem.

In contrast, fixed domain methods reformulate the problem over the whole of a fixed domain, and solve the new equations in such a way that the position of the free boundary is returned simultaneously with the solution to the PDE, and appears a posteriori as part of the solution process. Such methods have a reputation for being robust and relatively straightforward to implement. The most widely-used fixed-domain method is the penalty method, which incorporates the inequality constraint of the LCP into the PDE by adding a nonlinear penalty term (see, for example, Ch. 1, §8 of [22]). The resulting penalized equation can be solved by successive over-relaxation (SOR), which

can be fairly expensive (see, for example, [11]); this process can be accelerated by the multigrid method (see, for example, [9]). Remarkably, under certain conditions, when Newton's method is applied to the penalized PDE, the solution converges monotonically and exhibits the rapid convergence characteristic of Newton's method. However, while the discretized equation is solved to high accuracy, the approximation of the discrete solution to the true solution of the LCP is of low order, due to the disagreement between the free boundary and the fixed computational grid.

The problem of reconciling a nonconforming boundary with a fixed computational grid has been studied extensively, particularly in the context of front tracking methods for fluids and PDEs with smooth, fixed boundaries. One of the earliest such methods is the immersed boundary method (IBM) of Peskin, in which the boundary exerts an effect on a fluid, represented on a rectangular mesh, using approximations to delta functions located on the nonconforming smooth boundary (see [42]). This method was extended by Leveque and Li to the immersed interface method (IIM), which modifies the finite difference stencil in the vicinity of the boundary to correct for error terms derived from the underlying Taylor expansions (see [34]). In the explicit jump immersed interface method (EJIIM) proposed by Wiegmann and Bube, the corrections of the IIM are applied directly in terms of the jumps in the solution and its derivatives. Importantly, when the jumps are known a priori, the corrections are applied to the right-hand side of the discretized system of equations; when they are unknown, they are simultaneously solved for and used to correct the solution in the same spirit as the IIM (see [61]). We note that the idea of applying corrections to the right-hand side of the system was also suggested earlier by Fornberg and Meyer-Spasche in [20], in which they proposed a method for eliminating the first term in the expansion of the error near the nonconforming boundary. The ghost cell method (GSM) proposed by Gibou, Fedkiw, and others [23], based on the ghost fluid method (GFM) [18], is an alternative way of applying the jump corrections, in which ghost points are defined near the boundary, and equations for their values are adjoined to the discretized system. In [23], the authors observe that a second-order scheme can be constructed in which the discretized system is symmetric, however, they also observe that the resulting finite difference matrices becomes nonsymmetric for orders higher than two (see, for example, [24]). All of the aforementioned methods assume that the jumps at the nonconforming interface are either known beforehand, or are determined by augmenting the finite difference system with additional equations.

We present a method that does not augment or alter the finite difference matrix, and does not assume that the jumps are known in advance. We describe a high order deferred correction type algorithm for computing both the solution and the free boundary of an LCP. The idea is to derive the correction from the solution itself, after it has already been computed without any correction, or with a correction of a lower order. The correction is then applied to the right-hand side, and the problem is re-solved with the same matrix to one order of accuracy higher than before. Two key ideas which we use to rigorously justify this procedure are the smoothness of the error away from the free boundary, which justifies the numerical differentiation and extrapolation of the solution to obtain the jumps, and the fact that the Green's function describing the error near the free boundary decreases like $O(h)$ as the gridsize h goes to zero, which is needed to show that the jump corrections are computed to a sufficiently high order. Since the corrections are computed separately and are applied exclusively to the right hand side, the matrix of the system to be solved is identical to the

original finite difference matrix at each correction stage. In fact, since the solution at the previous correction stage can be used as an initial guess to penalty iteration at the subsequent stage, only one or two iterations are required for all correction stages after the first. The jump corrections are computed to high order by one-sided finite differences and extrapolation, and the location of the free boundary is determined, also to high order, from the solution by a combination of Lagrange interpolation and Newton's method. The deferred correction procedure can, at least in principle, be continued to indefinitely high orders, although we only apply it to fourth-order. We also note that the principles behind our deferred correction method are completely general, in the sense that they could be applied to essentially any free boundary problem formulated as an LCP. We demonstrate the effectiveness of the method on the American pricing problem, and several other examples with a one-dimensional space component, with and without a time component in the numerical results in Chapter 5.

2.1 Preliminaries

2.1.1 The LCP formulation of free and moving boundary problems

One form of the variational inequality representation of a free boundary problem in one dimension is

$$\begin{cases} \partial_t V_a - \mathcal{L}V_a - g \geq 0, \\ V_a - V^* \geq 0, \\ (\partial_t V_a - \mathcal{L}V_a - g) \cdot (V_a - V^*) = 0, \end{cases} \tag{2.1}$$

see, for example [22], where V_a is the solution we are seeking, $V^*(S)$ is a given function, sometimes called the obstacle function or the payoff function, and $\mathcal{L}$ is a second-order differential operator

$$\mathcal{L} = p(t,S)\frac{\partial^2}{\partial S^2} + w(t,S)\frac{\partial}{\partial S} + z(t,S), \tag{2.2}$$

where p, w, z and $g = g(t,S)$ are also given functions. Problem (2.1) is called a linear complementarity problem. Note that all three relations in (2.1) need to be satisfied. The solution of (2.1) is separated into two parts by a moving boundary $S_f(t)$. The goal is to find the solution $V_a = V_a(t,S)$ such that either $V_a - V^* > 0$ and $\partial_t V_a - \mathcal{L}V_a - g = 0$, on what we call the PDE region of the solution, or $\partial_t V_a - \mathcal{L}V_a - g \geq 0$ and $V_a - V^* = 0$, on what we call the penalty region of the solution. Consider the American option pricing problems. At each time, on one side of the domain, it is always optimal to exercise the option such that $V - V^* = 0$; while on the other side, it is always optimal to keep the option so that the PDE equality holds.

In elliptic obstacle problems, the ∂_t term disappears in the above formulation, and $\mathcal{L}$ is an elliptic operator. That is, the problem becomes

$$\begin{cases} -\mathcal{L}V_a - g \geq 0, \\ V_a - V^* \geq 0, \\ (-\mathcal{L}V_a - g) \cdot (V_a - V^*) = 0, \end{cases} \tag{2.3}$$

In the American option pricing problems, $\partial_t - \mathcal{L}$ is the famous Black-Scholes operator with $\mathcal{L} = \mathcal{L}_{BS}$ and

$$\mathcal{L}_{BS} \equiv \frac{\sigma^2 S^2}{2}\frac{\partial^2}{\partial S^2} + (r-d)S\frac{\partial}{\partial S} - r, \tag{2.4}$$

where S is the underlying asset price, r is the risk-free rate, d is the dividend rate of the underlying asset, σ is the volatility, and t is the backward time from expiry. Typical payoff functions are

$$V^*(S) = \max\{S-K, 0\} \quad \text{or } V^*(S) = \max\{K-S, 0\}$$

for the American call and put options, respectively. Note that for American put and call options, the obstacle function is not time dependent. It can be shown that the solution is only piecewise smooth, and the value matching and smooth pasting conditions

$$V_a(t, S_f(t)) = V^*(S_f(t)), \quad \frac{\partial}{\partial S}V_a(t, S_f(t)) = \frac{\partial}{\partial S}V^*(S_f(t)), \tag{2.5}$$

hold at the moving boundary (see, for example [62]), while the second derivative is discontinuous at $S_f(t)$. We see that the solution is only C^1 in space.

The presence of nonsmoothness in the solution at the undetermined free boundary poses challenges to achieve high-order convergence when solving the American option pricing problems. Nonuniform-mesh techniques have been proposed to deal with this issue [40, 12]. In the work by Oosterlee and Leentvaar [40], the authors propose to use fourth-order finite differences in space and BDF4 in time, together with time-dependent grid-stretching in a predictor-corrector type scheme to attain fourth-order accuracy. However, the authors of [40] did not provide numerical results on the convergence order for American options. In this chapter, we develop a general deferred correction algorithm using fourth-order finite difference method in space and BDF4 in time for solving free and moving boundary problems.

We point out that since the free boundary $S_f(t)$ changes infinitely quickly near expiry, [5] the solution derivatives at the free boundary of the American put options blow up at $t = 0$ as seen from (A.11) in Appendix A. This singular behaviour also appears in many other free boundary problems and typically causes an extra challenge to the development of high-order methods. Therefore, in the analysis of this chapter, we assume that the solution has no infinite singularity near $t = 0$ and leave this part for future research. Instead, when solving American options for the numerical results in Chapter 5, we apply a heuristic by stretching the time grid around $t = 0$ such that the solution errors from the initial steps remain minimal and do not dominate the global error. Additionally, we subtract out the solution of the corresponding European option from the American put option, such that we obtain the same PDE but with zero initial condition, and the solution changes less dramatically near $t = 0$. In this case, the solution of the corresponding European option serves as a media solution to alleviate the singularity in the American option solution. Since the solution of European options has an analytic form, this heuristic can be applied easily. For other problems that do not have closed-form media solutions, we can also apply this heuristic by solving the media problem numerically using our high-order methods developed in Chapters 3 and 4. More details on the implementation considerations are given in the numerical results Chapter 5

2.1.2 Penalty method for solving the LCP

We solve the LCP using the penalty method as discussed in [21]. We approximate (2.1) by the penalized nonlinear PDE

$$\frac{\partial V}{\partial t} = \mathcal{L}V + g + \rho \max\{V^* - V, 0\}, \tag{2.6}$$

for moving boundary problems, and

$$\mathcal{L}V + g + \rho \max\{V^* - V, 0\} = 0, \tag{2.7}$$

for free boundary problems, where ρ is a large positive penalty parameter, and V^* is the payoff function, as defined in Section 2.1.1, which also serves as the initial condition for PDE in (2.6). In particular, the penalty formulation of American option pricing is given by (1.9). The penalty term introduces nonlinearity to the PDE and penalizes the violation of the constraint $V - V^* \geq 0$. It can be shown that $\|\frac{\partial V}{\partial t}\|_{L^2}$, $\|V\|_{H^2}$ and $\|\rho \max\{V^* - V, 0\}\|_{L^2}$ are all uniformly bounded independent of ρ [16]. Therefore, as $\rho \to \infty$ in the limit, the linear complementarity conditions are satisfied, and either $V - V^* \geq 0$ or $V^* = V + \epsilon$ for $0 < \epsilon \ll 1$, where $\epsilon = \mathcal{O}(\rho^{-1})$, see [22]. Using a finite volume discretization and applying the generalized Newton's iteration, also referred to as discrete penalty iteration, to the discretized PDE, the authors of [21] are able to prove monotonic convergence and finite termination of the algorithm under certain conditions. Moreover, second-order convergence can be obtained with an adaptive time step selector.

2.2 Discretization, jump corrections and error analysis

2.2.1 Discretization of the penalized equation

In this section, we describe the discretization of (2.7) and (2.6), which will later lead to the formulation of a penalty iteration method for solving (2.7) and (2.6), similar to the second-order penalty method introduced in [21]. Unlike [21], we use fourth-order finite difference space discretization and BDF4 time-stepping in order to obtain high-order accuracy.

Consider a discretized domain $S_0 < S_1 < \cdots < S_M$ where S_0 and S_M represent the left and right boundary respectively. Let $\tilde{V}_j^n \approx V(t_n, S_j)$ be the finite difference approximation to the true solution $V(t, S)$ of (2.6) at time t_n, and space point S_j. We drop the superscript n when time is irrelevant. The space discretization is performed on a nonuniform grid that is smoothly mapped from the parametric space. The finite difference weights can be obtained by the method of undetermined coefficients in a stable way (see, for example, [19]). We denote the generic fourth-order finite difference approximation to $\frac{\partial^2 V}{\partial S^2}$ at the interior point S_j for $1 < j < M - 1$ to be

$$D_4^2 V_j \equiv c_{-2} V_{j-2} + c_{-1} V_{j-1} + c_0 V_j + c_1 V_{j+1} + c_2 V_{j+2},$$

where we abuse notation here and denote the finite difference coefficients at the points S_{j-2}, S_{j-1}, S_j, S_{j+1}, S_{j+2} by c_{-2}, c_{-1}, c_0, c_1, c_2, respectively, for the finite difference approximation at S_j.

At the two near-boundary points S_1 and S_{M-1}, we use biased finite difference scheme such that

$$D_4^2 V_1 \equiv c_{-1}V_0 + c_0 V_1 + c_1 V_2 + c_2 V_3 + c_3 V_4,$$

and similarly $D_4^2 V_{M-1}$. Fourth-order finite difference discretization of the first derivative $\frac{\partial V}{\partial S}(t, S_j)$ can be obtained similarly using a five-point stencil, which we omit for brevity. We note that using centered second-order finite differences for the two near-boundary points S_1 and S_{M-1} does not affect the fourth-order convergence rate.

Let $\mathbf{S}$ denote the vector of the interior grid points, i.e. $\mathbf{S} = [S_1, \ldots, S_{M-1}]^T$. Assuming Dirichlet boundary conditions, the fourth-order finite differences above give us the space discretization of $\frac{\partial^2 V}{\partial S^2}$ and $\frac{\partial V}{\partial S}$

$$\frac{\partial V}{\partial S}(t, \mathbf{S}) \approx \bar{\mathbf{L}}_1 \tilde{\mathbf{V}}_{\text{aug}}, \quad \frac{\partial^2 V}{\partial S^2}(t, \mathbf{S}) \approx \bar{\mathbf{L}}_2 \tilde{\mathbf{V}}_{\text{aug}},$$

where $\tilde{\mathbf{V}}_{\text{aug}} \equiv [\tilde{V}_0,\ \tilde{V}_1,\ \ldots,\ \tilde{V}_M]^T$ is the finite difference solution vector, $\bar{\mathbf{L}}_1$ and $\bar{\mathbf{L}}_2$ are $(M-1) \times (M+1)$ matrices with the coefficients of the corresponding finite difference stencil on each row. On a uniform space grid with step size h, we have

$$\bar{\mathbf{L}}_2 = \frac{1}{12h^2} \begin{bmatrix} 10 & -15 & -4 & 14 & -6 & 1 & & & & \\ -1 & 16 & -30 & 16 & -1 & & & & & \\ & -1 & 16 & -30 & 16 & -1 & & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & -1 & 16 & -30 & 16 & -1 & & \\ & & & & -1 & 16 & -30 & 16 & -1 & \\ & & & & 1 & -6 & 14 & -4 & -15 & 10 \end{bmatrix}_{(M-1)\times(M+1)},$$

$$\bar{\mathbf{L}}_1 = \frac{1}{12h} \begin{bmatrix} -3 & -10 & 18 & -6 & 1 & & & & & \\ 1 & -8 & 0 & 8 & -1 & & & & & \\ & 1 & -8 & 0 & 8 & -1 & & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & 1 & -8 & 0 & 8 & -1 & & \\ & & & & 1 & -8 & 0 & 8 & -1 & \\ & & & & -1 & 6 & -18 & 10 & 3 \end{bmatrix}_{(M-1)\times(M+1)}.$$

Let $\mathbf{L}$ be an $(M-1) \times (M-1)$ matrix defined by

$$\mathbf{L} \equiv \mathbf{P}\mathbf{L}_2 + \mathbf{W}\mathbf{L}_1 + \mathbf{Z}, \tag{2.8}$$

where $\mathbf{L}_2$ and $\mathbf{L}_1$ are $(M-1)\times(M-1)$ matrices formed by removing the first and last columns of $\bar{\mathbf{L}}_2$ and $\bar{\mathbf{L}}_1$, respectively, and $\mathbf{P}$, $\mathbf{W}$, and $\mathbf{Z}$ are diagonal matrices with diagonal entries $[\mathbf{P}]_{jj} = p(t, S_j)$, $[\mathbf{W}]_{jj} = w(t, S_j)$, and $[\mathbf{Z}]_{jj} = z(t, S_j)$ for $j = 1, \ldots, M-1$. Then the discretization of $\mathcal{L}V + g$

becomes

$$\mathcal{L}V(t,\mathbf{S}) + g(t,\mathbf{S}) \approx \mathbf{L}\tilde{\mathbf{V}} + \mathbf{b},$$

where $\tilde{\mathbf{V}} \equiv [\tilde{V}_1,\ \tilde{V}_2,\ \ldots,\ \tilde{V}_{M-1}]^T$, and

$$\begin{aligned}\mathbf{b} =& p(t,S_0)V(t,S_0)\bar{\mathbf{L}}_2[:,1] + w(t,S_0)V(t,S_0)\bar{\mathbf{L}}_1[:,1] \\ &+ p(t,S_M)V(t,S_M)\bar{\mathbf{L}}_2[:,M+1] + w(t,S_M)V(t,S_M)\bar{\mathbf{L}}_1[:,M+1] + g(t,\mathbf{S}),\end{aligned}$$

which is a vector that incorporates the boundary conditions, where $\bar{\mathbf{L}}_1[:,j]$ and $\bar{\mathbf{L}}_2[:,j]$ denote the j-th columns of $\bar{\mathbf{L}}_1$ and $\bar{\mathbf{L}}_2$, respectively. The penalty term in (2.6) and (2.7) can be discretized by

$$\mathbf{q}(\tilde{\mathbf{V}}) \equiv \rho\boldsymbol{\mathcal{I}}_{\tilde{\mathbf{V}}}(\mathbf{V}^* - \tilde{\mathbf{V}}) \tag{2.9}$$

where $\mathbf{V}^* = [V_1^*,\ V_2^*,\ \ldots,\ V_{M-1}^*]^T$ is the vector of the payoff function values on the grid points S_1 to S_{M-1}, and $\boldsymbol{\mathcal{I}}_{\tilde{\mathbf{V}}}$ is a diagonal matrix whose diagonal entries are

$$[\boldsymbol{\mathcal{I}}_{\tilde{\mathbf{V}}}]_{i,i} = \begin{cases} 1, & V_i^* > \tilde{V}_i, \\ 0, & \text{else.} \end{cases} \tag{2.10}$$

Therefore, we obtain the discretization of the right-hand side of (2.6),

$$\mathcal{L}V(t,\mathbf{S}) + g(t,\mathbf{S}) + \rho\max\{V^*(\mathbf{S}) - V(t,\mathbf{S}), 0\} \approx \mathbf{L}\tilde{\mathbf{V}} + \mathbf{b} + \mathbf{q}(\tilde{\mathbf{V}}). \tag{2.11}$$

Assuming BDF4 uniform time discretization, and defining

$$\mathbf{A} \equiv \frac{25}{12}\mathbf{I} - k\mathbf{L}, \tag{2.12}$$

the complete discretization of (2.6) including time stepping follows the rule

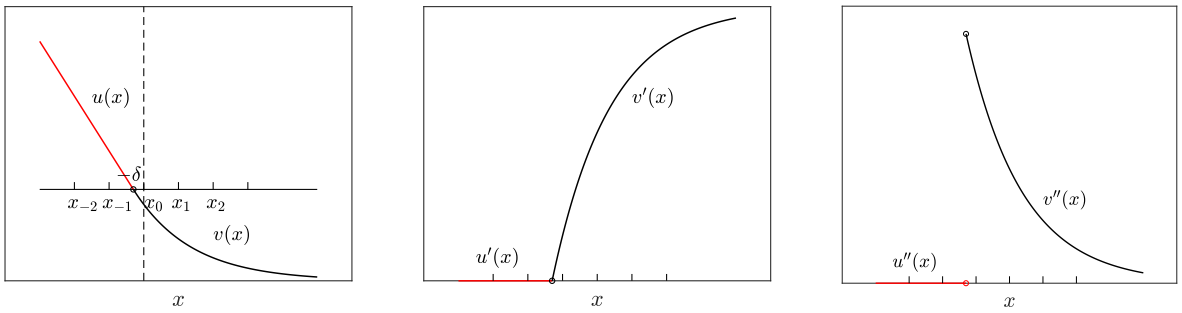
$$\mathbf{A}\tilde{\mathbf{V}}^{n+4} = 4\tilde{\mathbf{V}}^{n+3} - 3\tilde{\mathbf{V}}^{n+2} + \frac{4}{3}\tilde{\mathbf{V}}^{n+1} - \frac{1}{4}\tilde{\mathbf{V}}^{n} + k\mathbf{b}^{n+4} + k\mathbf{q}(\tilde{\mathbf{V}}^{n+4}), \tag{2.13}$$

where k is time step size, $\mathbf{I}$ is the identity matrix of size $(M-1)\times(M-1)$, and the superscript n means the n-th time step. To start BDF4, we find that using an explicit third-order Runge-Kutta (RK3) scheme for the first two steps and BDF3 for the third step works well. (This kind of starting scheme is studied in detail in Chapter 3.) Other starting schemes are also possible; see, for example, the American put option pricing problem in Section 5.1.3.
We also obtain the discretization of (2.7) as

$$\mathbf{L}\tilde{\mathbf{V}} + \mathbf{b} + \mathbf{q}(\tilde{\mathbf{V}}) = 0. \tag{2.14}$$

Systems (2.13) and (2.14) are nonlinear systems due to the presence of the penalty term, and we solve them using the penalty iteration described in [21].

We note that, at this point, we do not specify the choice of ρ in the terms $\mathbf{q}(\tilde{\mathbf{V}}^{n+4})$ and $\mathbf{q}(\tilde{\mathbf{V}})$



(a) $f(x)$ (b) $f'(x)$ (c) $f''(x)$

Figure 2.1: An example graph of a nonsmooth function with a point of discontinuity of the second derivative at $x = -\delta$

in (2.13) and (2.14), respectively. This will be discussed in Subsection 2.3.2.

2.2.2 Finite difference approximation on a nonsmooth but piecewise smooth function

As has been mentioned in the previous section, the solution of the LCP has a discontinuous second derivative at the free boundary at all times. This is a major factor that causes the degeneracy of convergence rate when using the finite difference method on uniform grids. Since the finite difference approximation is based upon Taylor expansions, a certain level of smoothness has to be assumed in order to obtain the corresponding accuracy. When this smoothness requirement is not satisfied even at a single point, the truncation error will be contaminated by an additional error, and propagated to other points in the solution through the Green's function, as we will see later. The analysis of this section is similar to the analysis of Li [36], and Wiegmann and Bube [61], except that it is applied to our particular high-order finite difference operator.

To analyze the impact of piecewise smoothness on the finite difference approximation, consider a piecewise smooth function

$$f(x) = \begin{cases} v(x), & x + \delta > 0, \\ u(x), & x + \delta \le 0, \end{cases} \tag{2.15}$$

such that $u(-\delta) = v(-\delta)$ and $u'(-\delta) = v'(-\delta)$, where δ is a positive constant. In addition, suppose that $u(x)$ and $v(x)$ admit smooth extensions, i.e., $u(x)$ is well defined and can be smoothly extended to the domain $x > -\delta$, and similarly $v(x)$ can be smoothly extended to $x < -\delta$. Let $\{x_j\}$ be a grid with $x_i < x_j$ for $i < j$, and with $x_{-1} < -\delta < x_0 = 0$. An example graph of function $f(x)$ with grid points x_{-2} to x_2 is shown in Figure 2.1. We want to approximate the second derivative of $f(x)$ at grid points around the nonsmooth position $x = -\delta$.

Second-order finite difference scheme

For notational convenience, we give a detailed derivation only for the second-order method. Derivations for the fourth-order method follow similarly. We pick the grid and location of the nonsmooth

point only for the ease of demonstration. The following derivation is generalizable to any other function of the same form, irrespective of where the nonsmooth point is located. When using a second-order finite difference method to approximate the second derivative of $f(x)$ at point $x = x_0 = 0$, we are actually computing

$$D^2 f_0 = \frac{1}{\bar{h}_0}\left[\frac{1}{h_0}u_{-1} - \left(\frac{1}{h_0}+\frac{1}{h_1}\right)v_0 + \frac{1}{h_1}v_1\right], \tag{2.16}$$

where u_j, v_j denote $u(x_j), v(x_j)$ respectively, D^2 represents the standard centred three-point finite-difference operator, $h_j = x_j - x_{j-1}$, and $\bar{h}_j = (h_j + h_{j+1})/2$. Note that the value of $u(x_{-1})$ instead of $v(x_{-1})$ is used for the left-most stencil point in (2.16). This is because the finite difference operator is applied to $f(x)$, which is equal to $u(x_{-1})$ at point x_{-1}. However, the correct (in the sense that it is second-order accurate) approximation to the second derivative at the point x_0 should be

$$D^2 v_0 = \frac{1}{\bar{h}_0}\left[\frac{1}{h_0}v_{-1} - \left(\frac{1}{h_0}+\frac{1}{h_1}\right)v_0 + \frac{1}{h_1}v_1\right], \tag{2.17}$$

where we recall the assumption that $v(s)$ has smooth extension for $x < -\delta$. Note that u_{-1} in the formula $D^2 f_0$ is replaced by v_{-1} in the formula $D^2 v_0$. The other problematic point is at $x = x_{-1} = -h_0$, where we approximate the derivative by

$$D^2 f_{-1} = \frac{1}{\bar{h}_{-1}}\left[\frac{1}{h_{-1}}u_{-2} - \left(\frac{1}{h_{-1}}+\frac{1}{h_0}\right)u_{-1} + \frac{1}{h_0}v_0\right],$$

rather than the second-order accurate finite difference

$$D^2 u_{-1} = \frac{1}{\bar{h}_{-1}}\left[\frac{1}{h_{-1}}u_{-2} - \left(\frac{1}{h_{-1}}+\frac{1}{h_0}\right)u_{-1} + \frac{1}{h_0}u_0\right].$$

The points x_{-1} and x_0 are the only problematic points for a second-order method. The degeneracy of the finite difference approximation accuracy comes from the inconsistency between the formulas for $D^2 f_0$ and $D^2 v_0$, and between $D^2 f_{-1}$ and $D^2 v_{-1}$.

The following theorem describes the relationship between $D^2 f_0$, $D^2 f_{-1}$ and $D^2 v_0$, $D^2 u_{-1}$, respectively, in terms of the jumps of $u(x)$ and $v(x)$ at point $x = -\delta$, and quantifies the degeneration of accuracy.

Theorem 2.2.1. *Suppose $f(x)$ is given by (2.15), where $f(x) = v(x)$ for $x > -\delta$ and $f(x) = u(x)$ for $x \le -\delta$, with $u(-\delta) = v(-\delta)$ and $u'(-\delta) = v'(-\delta)$, where $u(x)$ and $v(x)$ admit smooth extensions. Consider the functions on a grid $\{x_j\}$ with $x_i < x_j$ for $i < j$, and with $x_{-1} < -\delta < x_0 = 0$. Then, $D^2 f_0$, $D^2 f_{-1}$ and $D^2 v_0$, $D^2 u_{-1}$ satisfy the relations*

$$\begin{aligned} D^2 v_0 = D^2 f_0 &- \frac{(h_0-\delta)^2}{h_0(h_0+h_1)}(u''_\delta - v''_\delta) + \frac{(h_0-\delta)^3}{3h_0(h_0+h_1)}(u'''_\delta - v'''_\delta) \\ &- \frac{(h_0-\delta)^4}{12h_0(h_0+h_1)}(u''''_\delta - v''''_\delta) + \mathcal{O}(h^3), \end{aligned} \tag{2.18}$$

and

$$D^2u_{-1} = D^2f_{-1} + \frac{\delta^2}{h_0(h_{-1}+h_0)}(u''_\delta - v''_\delta) + \frac{\delta^3}{3h_0(h_{-1}+h_0)}(u'''_\delta - v'''_\delta) + \frac{\delta^4}{12h_0(h_{-1}+h_0)}(u''''_\delta - v''''_\delta) + \mathcal{O}(h^3), \tag{2.19}$$

where $h = \max\{h_0, h_1\}$*, and the subscript* δ *denotes the quantities at the nonsmooth point* $x = -\delta$*, e.g.* $u''_\delta = u''(-\delta)$*.*

Proof. Subtracting (2.16) from (2.17), we get

$$D^2f_0 = D^2v_0 + \frac{2}{h_0(h_0+h_1)}(u_{-1} - v_{-1}). \tag{2.20}$$

Applying Taylor expansions for functions $u(x)$ and $v(x)$ around $x = -\delta$, we have

$$u_{-1} = u_\delta - (h_0-\delta)u'_\delta + \frac{(h_0-\delta)^2}{2}u''_\delta - \frac{(h_0-\delta)^3}{6}u'''_\delta + \frac{(h_0-\delta)^4}{24}u''''_\delta + \mathcal{O}((h_0-\delta)^5),$$
$$v_{-1} = v_\delta - (h_0-\delta)v'_\delta + \frac{(h_0-\delta)^2}{2}v''_\delta - \frac{(h_0-\delta)^3}{6}v'''_\delta + \frac{(h_0-\delta)^4}{24}v''''_\delta + \mathcal{O}((h_0-\delta)^5),$$

which gives

$$u_{-1} - v_{-1} = \frac{(h_0-\delta)^2}{2}(u''_\delta - v''_\delta) - \frac{(h_0-\delta)^3}{6}(u'''_\delta - v'''_\delta) + \frac{(h_0-\delta)^4}{24}(u''''_\delta - v''''_\delta) + \mathcal{O}((h_0-\delta)^5), \tag{2.21}$$

using the assumptions that $u_\delta = v_\delta$ and $u'_\delta = v'_\delta$. Substituting (2.21) into (2.20), we get (2.18). Following a similar derivation, we get (2.19). □

Fourth-order finite difference scheme

In the previous section, we use the second-order approximation as a convenient way to demonstrate the essential relations that lead to our method. In this chapter, we focus on high-order methods. Following exactly the same derivation procedure, we can arrive at similar formulas for fourth-order methods. The main difference between the second-order and fourth-order FDs is that in the fourth-order FDs there are four problematic points, namely x_{-2}, x_{-1}, x_0, x_1, instead of just two. Let the finite difference coefficients at the points x_{j-2}, x_{j-1}, x_j, x_{j+1}, x_{j+2} be denoted by c_{-2}, c_{-1}, c_0, c_1, c_2, respectively, for the finite difference approximation at x_j. We give the following theorem for fourth-order discretization.

Theorem 2.2.2. *Under the same assumptions as in Theorem 2.2.1, we have that* $D_4^2u_{-2}$*,*

$D_4^2 u_{-1}$, $D_4^2 v_0$, $D_4^2 v_1$ satisfy the relations

$$D_4^2 u_{-2} = D_4^2 f_{-2} + c_2 \frac{\delta^2}{2}(u''_\delta - v''_\delta) + c_2 \frac{\delta^3}{6}(u'''_\delta - v'''_\delta) + c_2 \frac{\delta^4}{24}(u''''_\delta - v''''_\delta) + \mathcal{O}(h^3), \tag{2.22}$$

$$\begin{aligned} D_4^2 u_{-1} = D_4^2 f_{-1} &+ \left(c_1 \frac{\delta^2}{2} + c_2 \frac{(h_1+\delta)^2}{2} \right)(u''_\delta - v''_\delta) \\ &+ \left(c_1 \frac{\delta^3}{6} + c_2 \frac{(h_1+\delta)^3}{6} \right)(u'''_\delta - v'''_\delta) \\ &+ \left(c_1 \frac{\delta^4}{24} + c_2 \frac{(h_1+\delta)^4}{24} \right)(u''''_\delta - v''''_\delta) + \mathcal{O}(h^3), \end{aligned} \tag{2.23}$$

$$\begin{aligned} D_4^2 v_0 = D_4^2 f_0 &- \left(c_{-2} \frac{(h_{-1}+h_0-\delta)^2}{2} + c_{-1} \frac{(h_0-\delta)^2}{2} \right)(u''_\delta - v''_\delta) \\ &+ \left(c_{-2} \frac{(h_{-1}+h_0-\delta)^3}{6} + c_{-1} \frac{(h_0-\delta)^3}{6} \right)(u'''_\delta - v'''_\delta) \\ &- \left(c_{-2} \frac{(h_{-1}+h_0-\delta)^4}{24} + c_{-1} \frac{(h_0-\delta)^4}{24} \right)(u''''_\delta - v''''_\delta) + \mathcal{O}(h^3), \end{aligned} \tag{2.24}$$

$$\begin{aligned} D_4^2 v_1 = D_4^2 f_1 &- c_{-2} \frac{(h_0-\delta)^2}{2}(u''_\delta - v''_\delta) + c_{-2} \frac{(h_0-\delta)^3}{6}(u'''_\delta - v'''_\delta) \\ &- c_{-2} \frac{(h_0-\delta)^4}{24}(u''''_\delta - v''''_\delta) + \mathcal{O}(h^3), \end{aligned} \tag{2.25}$$

where $h = \max\{h_{-1}, h_0, h_1\}$.

Proof. From the approximation equations, we easily see that

$$\begin{aligned} D_4^2 f_{-2} &= D_4^2 u_{-2} + c_2(v_0 - u_0), \\ D_4^2 f_{-1} &= D_4^2 u_{-1} + c_1(v_0 - u_0) + c_2(v_1 - u_1), \\ D_4^2 f_0 &= D_4^2 v_0 + c_{-2}(u_{-2} - v_{-2}) + c_{-1}(u_{-1} - v_{-1}), \\ D_4^2 f_1 &= D_4^2 v_1 + c_{-2}(u_{-1} - v_{-1}). \end{aligned}$$

Then, expressing the quantities $u_{-2} - v_{-2}$, $u_{-1} - v_{-1}$, $v_0 - u_0$, $v_1 - u_1$, by applying Taylor expansions to $u_{-2}, v_{-2}, u_{-1}, v_{-1}, v_0, u_0, v_1, u_1$ about the point $x = -\delta$, exactly as in the proof of theorem 2.2.1, we get the desired relations. □

From Theorems 2.2.1 and 2.2.2, we see that, since $c_j = \mathcal{O}(1/h^2)$ and $\delta = \mathcal{O}(h)$, dominant $\mathcal{O}(1)$ terms appear in the truncation errors. Therefore, the second derivative approximations have degenerated orders of accuracy, regardless of the order of discretization. In order to achieve the desired order of accuracy, we have to eliminate the remainder terms. This can be done by adding corrections. We call the right-hand side terms of Equations (2.22)–(2.25) that are added to $D_4^2 f_j$ the *correction terms* to the finite difference approximation of the second derivatives at x_{-2} to x_1. For example, we call $c_2 \frac{\delta^2}{2}(u''_\delta - v''_\delta) + c_2 \frac{\delta^3}{6}(u'''_\delta - v'''_\delta) + c_2 \frac{\delta^4}{24}(u''''_\delta - v''''_\delta)$ the correction terms to $D_4^2 f_{-2}$ at x_{-2}. We will also refer to $u''_\delta - v''_\delta$, $u'''_\delta - v'''_\delta$ and $u''''_\delta - v''''_\delta$ as *derivative jumps*.

Modifying the finite differences with approximate corrections

The order of the FDs at the problematic points x_{-2} to x_1 is determined by the dominant error term in the corrections. If we were able to apply the exact corrections using Equations (2.22)–(2.25), we would fully recover the fourth-order convergence of the FDs at the four problematic points x_{-2} to x_1. However, in this work, we assume that the exact free boundary location and the derivative jumps are not known a priori. They are instead approximated using a previously computed $\mathcal{O}(h^\ell)$ solution, as we describe later. Therefore, we replace the exact free boundary and derivative jumps in the correction terms by the approximated ones. The accuracies of the corrected FDs depend on the accuracy of the approximate free boundary and derivative jumps.

To see how the order of accuracy of the free boundary and derivative jumps affect the correction, suppose that the free boundary is known exactly. Then, it is obvious that $\mathcal{O}(h^\ell)$ derivative jumps will give rise to $\mathcal{O}(h^\ell)$ corrections. On the other hand, suppose that the derivative jumps are known exactly, but we are given an approximate free boundary equal to $\delta + \mathcal{O}(h^\ell)$ with $1 \le \ell \le 3$. It is important to notice that the approximate free boundary introduces an extra source of error in the correction terms. To see this, we take one point, x_{-2}, for example. The finite difference scheme with approximate correction terms becomes

$$\begin{aligned}
&D_4^2 f_{-2} + c_2 \frac{(\delta + \mathcal{O}(h^\ell))^2}{2}(u''_\delta - v''_\delta) + c_2 \frac{(\delta + \mathcal{O}(h^\ell))^3}{6}(u'''_\delta - v'''_\delta) \\
&\qquad + c_2 \frac{(\delta + \mathcal{O}(h^\ell))^4}{24}(u''''_\delta - v''''_\delta) \\
=&D_4^2 f_{-2} + \left(c_2 \frac{\delta^2}{2} + \mathcal{O}(h^{\ell-1})\right)(u''_\delta - v''_\delta) + \left(c_2 \frac{\delta^3}{6} + \mathcal{O}(h^\ell)\right)(u'''_\delta - v'''_\delta) \\
&\qquad + \left(c_2 \frac{\delta^4}{24} + \mathcal{O}(h^{\ell+1})\right)(u''''_\delta - v''''_\delta) \\
=&D_4^2 f_{-2} + c_2 \frac{\delta^2}{2}(u''_\delta - v''_\delta) + c_2 \frac{\delta^3}{6}(u'''_\delta - v'''_\delta) + c_2 \frac{\delta^4}{24}(u''''_\delta - v''''_\delta) \\
&\qquad + \mathcal{O}(h^{\ell-1})(u''_\delta - v''_\delta) + \mathcal{O}(h^\ell)(u'''_\delta - v'''_\delta) + \mathcal{O}(h^{\ell+1})(u''''_\delta - v''''_\delta) \\
=&D_4^2 u_{-2} + \mathcal{O}(h^{\ell-1})(u''_\delta - v''_\delta) + \mathcal{O}(h^\ell)(u'''_\delta - v'''_\delta) + \mathcal{O}(h^{\ell+1})(u''''_\delta - v''''_\delta) + \mathcal{O}(h^3).
\end{aligned} \tag{2.26}$$

Equation (2.26) implies that, when applying corrections using an approximate free boundary, the correction terms produce additional errors that are one order lower than the accuracy of the approximate free boundary. In order to improve the order of accuracy of the finite difference scheme by adding back the correction terms, we see that ℓ has to satisfy $\ell \ge 2$, because if $\ell = 1$, the leading order term of the corrections on the right-hand side of (2.26) is still of constant order $\mathcal{O}(h^{\ell-1}) = \mathcal{O}(1)$. Therefore, we require the approximate derivative jumps and the free boundary location to be of at least $\mathcal{O}(h)$ and $\mathcal{O}(h^2)$, respectively, in order to increase the order of accuracy of the corrected finite differences to first-order, $\mathcal{O}(h^2)$ and $\mathcal{O}(h^3)$ to increase the order of accuracy to second-order, and so on.

2.2.3 Convergence of the fourth-order finite difference space discretization and its error propagation through the Green's function

Boundary value problems

In the previous section, we derived the correction terms for the finite difference approximations of derivatives of a nonsmooth but piecewise smooth function. Unless the values of $u''_\delta - v''_\delta$, $u'''_\delta - v'''_\delta$, $u''''_\delta - v''''_\delta$ are known, we cannot make use of these formulas directly to obtain fourth-order convergence. To solve this problem, we use a deferred correction approach, and successively compute the approximate derivatives from the lower-order solutions that are already known, and make sure to match up the orders of solutions and orders of corrections. In order to decide how much accuracy is required for the derivative approximation to result in corrections of the required order, we need to understand the error behaviour.

We consider boundary value problems that are time-independent, i.e., we consider (2.3) so that we can leave the complexity of time evolution for later discussion. To analyze the error behavior of the space discretization scheme, we consider the finite difference approximation of the PDE in (2.7) given by (2.14).

The theorem below describes the error behaviour of the fourth-order finite difference scheme applied to (2.7), and how the nonsmoothness at the free boundary causes the convergence order of the fourth-order difference scheme to degenerate.

Proposition 2.2.1. *Consider the penalized PDE (2.7) with $V(S)$ being its exact solution, and the original LCP (2.3) with $V_a(S)$ being its exact solution. Suppose that the first $m+1$ points $V_a(S_0)$, $V_a(S_1)$, $\ldots$, $V_a(S_m)$ lie on the penalty region, i.e. $V_a(S_j) = V^*(S_j) = V(S_j) \pm \epsilon$ for $0 \le j \le m$, with $0 < \epsilon \ll 1$ being approximately the size of the stopping tolerance set in the penalty iteration, and $V_a(S_j) > V^*(S_j)$, $V(S_j) \pm \epsilon > V^*(S_j)$, for $m+1 \le j \le M$. Assume also that the approximate solution $\tilde{\mathbf{V}}$ of the penalty iteration exactly recovers $\boldsymbol{\mathcal{I}}_{\mathbf{V}_a}$, i.e., $\boldsymbol{\mathcal{I}}_{\tilde{\mathbf{V}}} = \boldsymbol{\mathcal{I}}_{\mathbf{V}_a}$. Then, the error, $\mathbf{e} = [V_a(S_1) - \tilde{V}_1,\ V_a(S_2) - \tilde{V}_2,\ \ldots,\ V_a(S_{M-1}) - \tilde{V}_{M-1}]^T$, of the fourth-order finite difference scheme in (2.14) for solving the penalized PDE (2.7) satisfies*

$$(\mathbf{L} - \rho\boldsymbol{\mathcal{I}}_{\mathbf{V}_a})\mathbf{e} = \boldsymbol{\gamma} + \sum_{j=m-1}^{m+2} \mathcal{O}(1)\mathbf{1}_j + \sum_{j=1}^{M-1} \mathcal{O}(h^4)\mathbf{1}_j \equiv \mathbf{r}, \tag{2.27}$$

when the grid point S_m is not exactly on the free boundary, i.e. $S_m < S_f$, where $\mathbf{1}_j$ is the j-th column of an $(M-1)\times(M-1)$ identity matrix, and $[\boldsymbol{\gamma}]_j = (\mathcal{L}V_a(S_j) + g(S_j))\mathbb{1}_{1\le j\le m}$, for $j = 1, \ldots, M-1$, where $\mathbb{1}_{1\le j\le m}$ is the indicator function, which is one when $1 \le j \le m$ and zero otherwise. When S_m is exactly on the free boundary, i.e., $S_m = S_f$, the sum in the second summation term is taken from $j = m-1$ to $m+1$.

Proof. Since $S_0, S_1, \ldots, S_m$ lie on the penalty region and we assume Dirichlet boundary conditions, $\boldsymbol{\mathcal{I}}_{\mathbf{V}_a}$ is an $(M-1)\times(M-1)$ diagonal matrix with the diagonal elements $(\boldsymbol{\mathcal{I}}_{\mathbf{V}_a})_{i,i} = 1$ for $i = 1, \ldots, m$

and $(\boldsymbol{\mathcal{I}}_{\mathbf{V}_a})_{i,i} = 0$ for $i = m+1, \ldots, M-1$. Hence, from the assumption that $\boldsymbol{\mathcal{I}}_{\tilde{\mathbf{V}}} = \boldsymbol{\mathcal{I}}_{\mathbf{V}_a}$, we have

$$\begin{aligned}\mathbf{q}(\tilde{\mathbf{V}}) &= \rho[V^*(S_1) - \tilde{V}_1,\ \ldots,\ V^*(S_m) - \tilde{V}_m,\ 0,\ \ldots, 0]^T \\ &= \rho[V_a(S_1) - \tilde{V}_1,\ \ldots,\ V_a(S_m) - \tilde{V}_m,\ 0,\ \ldots, 0]^T \\ &= \rho \boldsymbol{\mathcal{I}}_{\tilde{\mathbf{V}}} \mathbf{e},\end{aligned} \tag{2.28}$$

Assume $S_m < S_f$, i.e. the grid point S_m is not exactly on the free boundary. The proof for the case when $S_m = S_f$ is similar.

From theorem 2.2.2 for fourth-order discretization, we apply the discrete $\mathbf{L}$ operator to the true solution $\mathbf{V}_a \equiv V_a(\mathbf{S})$ to get

$$\begin{aligned}&\mathbf{L}\mathbf{V}_a + \mathbf{b} = \sum_{j=1}^{M-1} \left(\mathcal{L}V_a(S_j) + g(S_j) + \mathcal{O}(h^4)\right)\mathbf{1}_j + \sum_{j=m-1}^{m+2} \mathcal{O}(1)\mathbf{1}_j \equiv \mathcal{L}V_a(\mathbf{S}) + g(\mathbf{S}) + \boldsymbol{\theta}, \\ &\boldsymbol{\theta} \equiv \sum_{j=m-1}^{m+2} \mathcal{O}(1)\mathbf{1}_j + \sum_{j=1}^{M-1} \mathcal{O}(h^4)\mathbf{1}_j.\end{aligned} \tag{2.29}$$

Since $\mathcal{L}V_a + g = 0$ for $S_{m+1} \le S \le S_{M-1}$, and using (2.29), we have

$$\mathbf{L}\mathbf{V}_a + \mathbf{b} = \mathcal{L}V_a(\mathbf{S}) + g(\mathbf{S}) + \boldsymbol{\theta} = \boldsymbol{\gamma} + \boldsymbol{\theta}. \tag{2.30}$$

Subtracting (2.14) from (2.30) and applying (2.28), we get

$$\mathbf{L}(\mathbf{V}_a - \tilde{\mathbf{V}}) - \mathbf{q}(\tilde{\mathbf{V}}) - \boldsymbol{\gamma} - \boldsymbol{\theta} = (\mathbf{L} - \rho\boldsymbol{\mathcal{I}}_{\mathbf{V}_a})\mathbf{e} - \boldsymbol{\gamma} - \theta = \mathbf{0}.$$

Therefore, the error satisfies

$$(\mathbf{L} - \rho\boldsymbol{\mathcal{I}}_{\mathbf{V}_a})\mathbf{e} = \boldsymbol{\gamma} + \sum_{j=m-1}^{m+2} \mathcal{O}(1)\mathbf{1}_j + \sum_{j=1}^{M-1} \mathbf{O}(h^4)\mathbf{1}_j.$$

□

Proposition 2.2.1 identifies the error equation $\mathbf{e} = (\mathbf{L} - \rho\boldsymbol{\mathcal{I}}_{\tilde{\mathbf{V}}})^{-1}\mathbf{r}$. The following proposition tells us how the operator $(\mathbf{L} - \rho\boldsymbol{\mathcal{I}}_{\tilde{\mathbf{V}}})^{-1}$ behaves.

Proposition 2.2.2. *Consider the partitioning of the matrix* $\mathbf{L}$*, representing the discretization of (2.2) and defined in (2.8), into*

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix}, \tag{2.31}$$

where the submatrices $\mathbf{L}_{11}$, $\mathbf{L}_{12}$, $\mathbf{L}_{21}$, $\mathbf{L}_{22}$ *are of sizes* $m \times m$, $m \times (M-1-m)$, $(M-1-m) \times m$ *and* $(M-1-m) \times (M-1-m)$ *respectively, and* m *is the largest integer such that* $S_m \le S_f$*. Assume* $\mathbf{L}_{11}$ *and* $\mathbf{L}_{22}$ *are nonsingular, and* ρ *is a positive number such that* $\rho \gg \max_{ij}\{|L_{i,j}|\}$*. Assume also that* $\max_j\{|[\mathbf{L}_{22}^{-1}]_{1,j}|\} = \mathcal{O}(h^2)$, $\max_j\{|[\mathbf{L}_{22}^{-1}]_{2,j}|\} = \mathcal{O}(h^2)$, $\max_i\{|[\mathbf{L}_{22}^{-1}]_{i,1}|\} = \mathcal{O}(h^2)$, *and* $\max_i\{|[\mathbf{L}_{22}^{-1}]_{i,2}|\} = \mathcal{O}(h^2)$*. Let* $\boldsymbol{\mathcal{I}}$ *be a diagonal matrix such that* $(\boldsymbol{\mathcal{I}})_{i,i} = 1$ *for* $i = 1, \ldots, m$*,*

$(\boldsymbol{\mathcal{I}})_{i,i} = 0$ *for* $i = m+1, \ldots, M-1$. *Then* $(\mathbf{L} - \rho\boldsymbol{\mathcal{I}})^{-1}$ *has the approximation*

$$(\mathbf{L} - \rho\boldsymbol{\mathcal{I}})^{-1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{22}^{-1} \end{bmatrix} + \mathcal{O}\left(\frac{1}{\rho}\right) \mathbf{J}_{(M-1)\times(M-1)}, \tag{2.32}$$

where $\mathbf{J}_{(M-1)\times(M-1)}$ *denotes a* $(M-1)\times(M-1)$ *matrix of all ones.*

Remark 2.2.1. *We denote* $(\mathbf{L}_{22})^{-1}$ *by* $\mathbf{L}_{22}^{-1}$ *for notational simplicity.*

Proof. For convenience, we consider a uniform space discretization. Results on nonuniform grids follow similarly. Since $\mathbf{L} = \mathbf{P}\mathbf{L}_2 + \mathbf{W}\mathbf{L}_1 + \mathbf{Z}$ as in (2.8) and $\mathbf{P}, \mathbf{W}, \mathbf{Z}$ are diagonal matrices, we have $\mathbf{L}_{11} = \mathbf{P}_1\mathbf{L}_{2|11} + \mathbf{W}_1\mathbf{L}_{1|11} + \mathbf{Z}_1$, $\mathbf{L}_{22} = \mathbf{P}_2\mathbf{L}_{2|22} + \mathbf{W}_2\mathbf{L}_{1|22} + \mathbf{Z}_2$ and $\mathbf{L}_{12} = \mathbf{P}_1\mathbf{L}_{2|12} + \mathbf{W}_1\mathbf{L}_{1|12} + \mathbf{Z}_1$ with $\mathbf{P}_1 = \mathbf{P}[1:m, 1:m], \mathbf{P}_2 = \mathbf{P}[m+1:M-1, m+1:M-1]$, similarly for $\mathbf{W}_1, \mathbf{W}_2, \mathbf{Z}_1, \mathbf{Z}_2$, and

$$\mathbf{L}_{2|11} = \frac{1}{12h^2}\begin{bmatrix} -15 & -4 & 14 & -6 & 1 & & \\ 16 & -30 & 16 & -1 & & & \\ -1 & 16 & -30 & 16 & -1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -1 & 16 & -30 & 16 & -1 \\ & & & -1 & 16 & -30 & 16 \\ & & & & -1 & 16 & -30 \end{bmatrix}_{m\times m},$$

$$\mathbf{L}_{2|22} = \frac{1}{12h^2}\begin{bmatrix} -30 & 16 & -1 & & & & \\ 16 & -30 & 16 & -1 & & & \\ -1 & 16 & -30 & 16 & -1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & -1 & 16 & -30 & 16 & -1 \\ & & & -1 & 16 & -30 & 16 \\ & & & 1 & -6 & 14 & -4 & -15 \end{bmatrix}_{(M-1-m)\times(M-1-m)},$$

$$\mathbf{L}_{1|11} = \frac{1}{12h}\begin{bmatrix} 0 & 8 & -1 & & & & \\ -8 & 0 & 8 & -1 & & & \\ 1 & -8 & 0 & 8 & -1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -8 & 0 & 8 & -1 \\ & & & 1 & -8 & 0 & 8 \\ & & & -1 & 6 & -18 & 10 \end{bmatrix}_{m\times m},$$

$$\mathbf{L}_{1|22} = \frac{1}{12h}\begin{bmatrix} -10 & 18 & -6 & 1 & & & \\ -8 & 0 & 8 & -1 & & & \\ 1 & -8 & 0 & 8 & -1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -8 & 0 & 8 & -1 \\ & & & 1 & -8 & 0 & 8 \\ & & & & 1 & -8 & 0 \end{bmatrix}_{(M-1-m)\times(M-1-m)},$$

$$\mathbf{L}_{2|12} = \frac{1}{12h^2}\begin{bmatrix} & & \\ -1 & & \\ 16 & -1 & \end{bmatrix}_{m\times(M-1-m)}, \qquad \mathbf{L}_{1|12} = \frac{1}{12h}\begin{bmatrix} & & \\ -1 & & \\ 8 & -1 & \end{bmatrix}_{m\times(M-1-m)}.$$

Note that $\mathbf{L}_{11}, \mathbf{L}_{22}$ and $\mathbf{L}_{12}$ have the same nonzero pattern as $\mathbf{L}_{2|11}, \mathbf{L}_{2|22}$ and $\mathbf{L}_{2|12}$, and that $\mathbf{L}_{12}$ and $\mathbf{L}_{21}$ have only three nonzero entries in the lower-left and upper-right corners, respectively, and all these entries are $\mathcal{O}(1/h^2)$. Therefore, we have the nonzero patterns of $\mathbf{L}_{12}\mathbf{L}_{22}^{-1}$, $\mathbf{L}_{22}^{-1}\mathbf{L}_{21}$ and $\mathbf{L}_{12}\mathbf{L}_{22}^{-1}\mathbf{L}_{21}$ as follows:

$$\underbrace{\begin{bmatrix} & & & \\ & & & \\ \times & \times & \dots & \times \\ \times & \times & \dots & \times \end{bmatrix}_{m\times(M-1-m)}}_{\mathbf{L}_{12}\mathbf{L}_{22}^{-1}}, \qquad \underbrace{\begin{bmatrix} & & \\ & \times & \times \\ & \times & \times \\ & \vdots & \vdots \\ & \times & \times \end{bmatrix}_{(M-1-m)\times m}}_{\mathbf{L}_{22}^{-1}\mathbf{L}_{21}}, \qquad \underbrace{\begin{bmatrix} & & \\ & & \\ & \times & \times \\ & \times & \times \end{bmatrix}_{m\times m}}_{\mathbf{L}_{12}\mathbf{L}_{22}^{-1}\mathbf{L}_{21}},$$

where the symbol "$\times$" denotes a nonzero entry. In addition to the special nonzero patterns, from the fact that the nonzero entries of $\mathbf{L}_{12}$, $\mathbf{L}_{21}$ are $\mathcal{O}(1/h^2)$, and the assumptions that $\max_j\{|[\mathbf{L}_{22}^{-1}]_{1,j}|\} = \mathcal{O}(h^2)$, $\max_j\{|[\mathbf{L}_{22}^{-1}]_{2,j}|\} = \mathcal{O}(h^2)$, $\max_i\{|[\mathbf{L}_{22}^{-1}]_{i,1}|\} = \mathcal{O}(h^2)$, and $\max_i\{|[\mathbf{L}_{22}^{-1}]_{i,2}|\} = \mathcal{O}(h^2)$ we see that $\mathbf{L}_{12}\mathbf{L}_{22}^{-1}$, $\mathbf{L}_{22}^{-1}\mathbf{L}_{21}$ have nonzero entries of $\mathcal{O}(1)$, and that $\mathbf{L}_{12}\mathbf{L}_{22}^{-1}\mathbf{L}_{21}$ has nonzero entries of $\mathcal{O}(1/h^2)$. Moreover, $\mathbf{L}_{22}^{-1}\mathbf{L}_{21}\mathbf{L}_{12}\mathbf{L}_{22}^{-1}$ have nonzero entries of $\mathcal{O}(1)$ since each of its nonzero entry is composed as the sum of two terms of $\mathcal{O}(1)$.

Since $\mathbf{L}_{11}$, $\mathbf{L}_{22}$ are nonsingular, the exact inverse matrix of $\mathbf{L} - \rho\boldsymbol{\mathcal{I}}$ is

$$(\mathbf{L} - \rho\boldsymbol{\mathcal{I}})^{-1} = \begin{bmatrix} \mathbf{B} & -\mathbf{B}\mathbf{L}_{12}\mathbf{L}_{22}^{-1} \\ -\mathbf{L}_{22}^{-1}\mathbf{L}_{21}\mathbf{B} & \mathbf{L}_{22}^{-1} + \mathbf{L}_{22}^{-1}\mathbf{L}_{21}\mathbf{B}\mathbf{L}_{12}\mathbf{L}_{22}^{-1} \end{bmatrix},$$

where $\mathbf{B} = (\mathbf{L}_{11} - \rho\mathbf{I} - \mathbf{L}_{12}\mathbf{L}_{22}^{-1}\mathbf{L}_{21})^{-1}$, and $\mathbf{I}$ is the identity matrix of size $m \times m$. Since $\rho \gg \max_{ij}\{|L_{i,j}|\} = \mathcal{O}(1/h^2)$, we have $\mathbf{B} = -\frac{1}{\rho}\left(\mathbf{I} + \frac{1}{\rho}(\mathbf{L}_{12}\mathbf{L}_{22}^{-1}\mathbf{L}_{21} - \mathbf{L}_{11})\right)^{-1} = -\frac{1}{\rho}\mathbf{I} + \frac{1}{\rho}\mathcal{O}\left(\frac{1}{\rho h^2}\right)\mathbf{J}_{m\times m}$. Using the assumption that $\rho \gg \mathcal{O}(1/h^2)$, we get $\frac{1}{\rho}\mathcal{O}\left(\frac{1}{\rho h^2}\right) < \mathcal{O}\left(\frac{1}{\rho}\right)$. Therefore,

$$(\mathbf{L}-\rho\boldsymbol{\mathcal{I}})^{-1} = \begin{bmatrix} -\frac{1}{\rho}\mathbf{I} + \frac{1}{\rho}\mathcal{O}\left(\frac{1}{\rho h^2}\right)\mathbf{J}_{m\times m} & \frac{1}{\rho}\mathbf{L}_{12}\mathbf{L}_{22}^{-1} \\ \frac{1}{\rho}\mathbf{L}_{22}^{-1}\mathbf{L}_{21} & \mathbf{L}_{22}^{-1} - \frac{1}{\rho}\mathbf{L}_{22}^{-1}\mathbf{L}_{21}\mathbf{L}_{12}\mathbf{L}_{22}^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{22}^{-1} \end{bmatrix} + \mathcal{O}\left(\frac{1}{\rho}\right)\mathbf{J}_{(M-1)\times(M-1)},$$

where the last equality holds because each nonzero entry of $\mathbf{L}_{12}\mathbf{L}_{22}^{-1}$, $\mathbf{L}_{22}^{-1}\mathbf{L}_{21}$, and $\mathbf{L}_{22}^{-1}\mathbf{L}_{21}\mathbf{L}_{12}\mathbf{L}_{22}^{-1}$ is $\mathcal{O}(1)$ as shown earlier. □

Remark 2.2.2. *Note that* $\mathbf{L}_{22}$ *behaves as the fourth-order finite difference discretization of* $\mathcal{L}$ *on the grid* S_{m+1}, S_{m+2}, …, S_{M-1} *with* S_m *and* S_M *as the boundary points. As such, the assumptions in the proposition that* $\max_j\{|[\mathbf{L}_{22}^{-1}]_{1,j}|\} = \mathcal{O}(h^2)$, $\max_j\{|[\mathbf{L}_{22}^{-1}]_{2,j}|\} = \mathcal{O}(h^2)$, $\max_i\{|[\mathbf{L}_{22}^{-1}]_{i,1}|\} = \mathcal{O}(h^2)$, *and* $\max_i\{|[\mathbf{L}_{22}^{-1}]_{i,2}|\} = \mathcal{O}(h^2)$ *are typically true, under some conditions on the coefficients* $p(t,S), w(t,S), z(t,S)$ *and the grid spacing. See, for example, Proposition 2.2.3 below.*

Remark 2.2.3. *The assumption $\rho \gg \max_{ij}\{|L_{i,j}|\} = \mathcal{O}(1/h^2)$ implies that the value of ρ should be adjusted for each refinement of the grid in a way that ensures $\rho \gg 1/h^2$. However, in practice, we can also set ρ to be a fixed constant that is large enough so that the approximation (2.32) holds at all refinements. The assumption that $\rho \gg 1/h^2$ is a sufficient condition for our proof of Proposition 2.2.2, but is not a necessary one.*

Let $\mathbf{r}$ be defined as in Proposition 2.2.1 and $\mathbf{r}_{m+1:M-1}$ denote the subvector of $\mathbf{r}$ starting from entry $m+1$ to $M-1$. We then have the following theorem.

Theorem 2.2.3. *Under the same assumptions as in Propositions 2.2.1, 2.2.2 and the assumption that $\boldsymbol{\mathcal{I}}_{\tilde{\mathbf{V}}} = \boldsymbol{\mathcal{I}}_{\mathbf{V}_a}$, when using the fourth-order finite difference scheme in (2.14) to solve the penalized PDE (2.7), the error satisfies*

$$\mathbf{e} = \begin{bmatrix} \mathbf{0} \\ \mathbf{L}_{22}^{-1}\mathbf{r}_{m+1:M-1} \end{bmatrix} + \mathcal{O}\left(\frac{1}{\rho}\right)\mathbf{J}_{(M-1)\times 1},$$

where $\mathbf{J}_{(M-1)\times 1}$ is a $(M-1)\times 1$ vector of all ones.

Proof. The theorem is easily obtained from Propositions 2.2.1 and 2.2.2. □

Theorem 2.2.3 shows that the penalty method obtains the exact solution within a pre-specified tolerance $\mathcal{O}(1/\rho)$ on the penalty region, while, on the PDE region, where the solution satisfies the PDE (2.7), the error is given by $\mathbf{L}_{22}^{-1}\mathbf{r}_{m+1:M-1}$. Note also that, on a uniform grid with step size h, $\mathbf{L}_{22}^{-1}$ can be thought of as the finite difference analogue of the continuous Green's function of $\mathcal{L}$ on the PDE region, scaled by $\frac{h}{2}$.

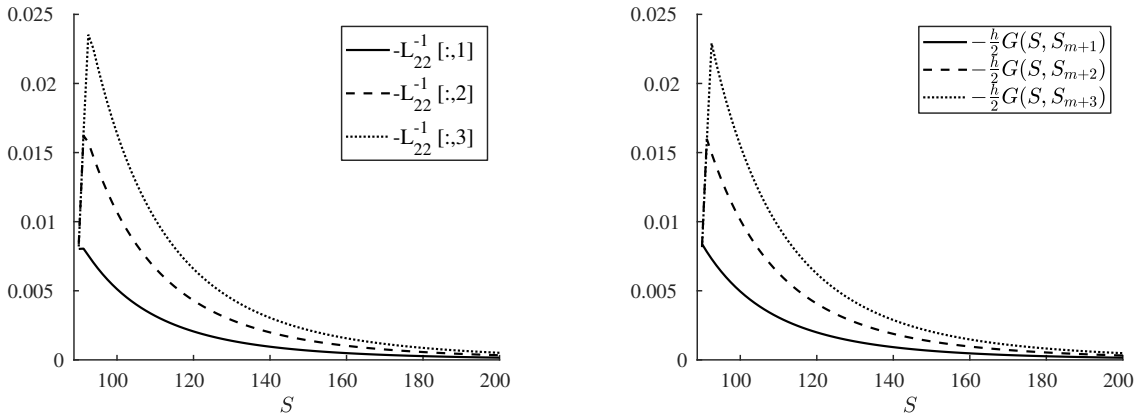
To support this conjecture, we visualize the size of entries of $\mathbf{L}_{22}^{-1}$ and its relation to the continuous Green's function. Figure 2.2 gives the first three columns of $-\mathbf{L}_{22}^{-1}$ and the corresponding $-\frac{h}{2}G(S, S_{m+j})$, for $j = 1,\ 2$ and 3, for the operator $\mathcal{L}_{BS}$ given by Equation (2.4), on an example uniform grid where the free boundary is located at $S_m < S_f = 89.748 < S_{m+1}$. We can see that $[\mathbf{L}_{22}^{-1}]_{:,j}$ and $\frac{h}{2}G(S, S_{m+j})$ behave similarly.
Therefore, we have

$$\mathbf{e} \approx \sum_{j=m+1}^{M-1} r_j \begin{bmatrix} \mathbf{0} \\ [\mathbf{L}_{22}^{-1}]_{:,j} \end{bmatrix} \approx \sum_{j=m+1}^{M-1} r_j \frac{h}{2} \begin{bmatrix} \mathbf{0} \\ G(\mathbf{S}_2, S_j) \end{bmatrix}, \tag{2.33}$$

where $G(\mathbf{S}_2, S_j)$ is a column vector of function values $G(S, S_j)$ at points $\mathbf{S}_2 \equiv \{S_{m+1}, \ldots, S_{M-1}\}$.

In order to analyze the error behavior, we turn to understanding the properties of the Green's function $G(S, S_j)$, which is easier to investigate than its discrete analogue $\mathbf{L}_{22}^{-1}$. The following proposition gives the exact expression of the Green's function to a general operator.

Proposition 2.2.3. *Suppose that $Tu(x) = 0$ is a constant-coefficient, second-order homogeneous differential equation defined on the domain $[a, b]$. Let ξ_1 and ξ_2 be the roots of the characteristic equation arising from the differential operator T. Suppose further that ξ_1 and ξ_2 are real and $\xi_1 \neq \xi_2$. Let $u(x) = c_1 e^{\xi_1 x} + c_2 e^{\xi_2 x}$ denote the general solution to this equation. Then, the Green's*



(a) The first 3 columns of $-\mathbf{L}_{22}^{-1}$ (b) The continuous Green's function $-\frac{h}{2}G(S, S_j)$

Figure 2.2: (a) The first three columns of $-\mathbf{L}_{22}^{-1}$ on an example uniform grid of size $h = 1.25$; (b) The scaled continuous Green's function $-\frac{h}{2}G(S, S_j)$ for the operator $\mathcal{L}_{BS}$ at S_{m+1}, S_{m+2}, S_{m+3} and for $S \geq S_{m+1}$. The free boundary location is $S_f = 89.748$.

function for the operator T is

$$G(x,\bar{x}) = \begin{cases} \dfrac{e^{(\xi_2-\xi_1)b} - e^{(\xi_2-\xi_1)\bar{x}}}{(\xi_2-\xi_1)e^{\xi_2\bar{x}}\left(e^{(\xi_2-\xi_1)b} - e^{(\xi_2-\xi_1)a}\right)}\left(e^{\xi_2 x} - e^{(\xi_2-\xi_1)a+\xi_1 x}\right), & a \leq x < \bar{x}, \\ \dfrac{e^{(\xi_2-\xi_1)a} - e^{(\xi_2-\xi_1)\bar{x}}}{(\xi_2-\xi_1)e^{\xi_2\bar{x}}\left(e^{(\xi_2-\xi_1)b} - e^{(\xi_2-\xi_1)a}\right)}\left(e^{\xi_2 x} - e^{(\xi_2-\xi_1)b+\xi_1 x}\right), & \bar{x} \leq x \leq b. \end{cases} \tag{2.34}$$

Moreover, for any $x \in [a, b]$, we have

$$G(x,\bar{x}) = \mathcal{O}(\bar{x} - a), \quad \text{as } \bar{x} \to a.$$

Proof. The computation of the Green's function follows the standard procedure and we omit it. When $\bar{x} \leq x \leq b$, we have

$$e^{(\xi_2-\xi_1)a} - e^{(\xi_2-\xi_1)\bar{x}} = e^{(\xi_2-\xi_1)a}\left(1 - e^{(\xi_2-\xi_1)(\bar{x}-a)}\right) \approx (\xi_1-\xi_2)(\bar{x}-a)e^{(\xi_2-\xi_1)a} = \mathcal{O}(\bar{x}-a),$$

as $\bar{x} \to a$. When $a \leq x \leq \bar{x}$, we have

$$e^{\xi_2 x} - e^{(\xi_2-\xi_1)a+\xi_1 x} = e^{\xi_2 a}\left(e^{\xi_2(x-a)} - e^{\xi_1(x-a)}\right) \approx e^{\xi_2 a}(\xi_2-\xi_1)(x-a) \leq e^{\xi_2 a}(\xi_2-\xi_1)(\bar{x}-a) = \mathcal{O}(\bar{x}-a),$$

as $\bar{x} \to a$. Therefore, we see that $G(x,\bar{x}) = \mathcal{O}(\bar{x} - a)$ as $\bar{x} \to a$ for all $a \leq x \leq b$. □

From Proposition 2.2.3, with $a = S_f$, and $\bar{x} = S_{m+1}$ or $\bar{x} = S_{m+2}$, we have, for $S > S_f$, $G(S, S_{m+1}) = \mathcal{O}(S_{m+1} - S_f) = \mathcal{O}(h)$, and $G(S, S_{m+2}) = \mathcal{O}(S_{m+2} - S_f) = \mathcal{O}(h)$ as $h \to 0$. For a visual demonstration of property, Figure 2.3 gives an illustration of Green's function for a hypothetical second-order differential equation on two successive grid refinements.
The following theorem is a key point in the success of the method presented in Section 2.3.

Theorem 2.2.4. *Under the same assumptions as in Theorem 2.2.3, we have that the error e_i at*

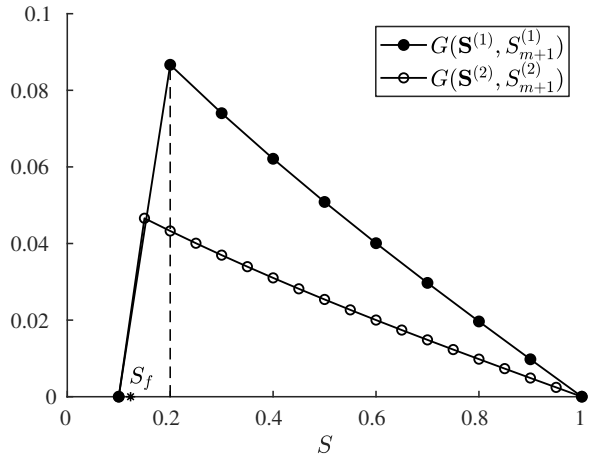


Figure 2.3: Illustration of Green's functions on two successive grid refinements $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}$, for the second-order differential operator $-u'' + u$ in $[0.1, 1]$ with $S_f = 0.123$.

S_i, for $i \geq m+1$, is

$$e_i \approx \mathcal{O}(h)G(S_i, S_{m+1}) + \mathcal{O}(h)G(S_i, S_{m+2}) = \mathcal{O}(h^2), \tag{2.35}$$

and its components for $i \geq m+2$ are samples of a smooth function of the form $\mathcal{O}(h)G(S, S_{m+1}) + \mathcal{O}(h)G(S, S_{m+2})$.

Proof. We first note that $r_{m+1} = \mathcal{O}(1)$, $r_{m+2} = \mathcal{O}(1)$, and $r_j = \mathcal{O}(h^4)$, for $j \geq m+3$. Therefore, from relation (2.33), we have

$$\mathbf{e} \approx \mathcal{O}(h) \begin{bmatrix} \mathbf{0} \\ G(\mathbf{S}_2, S_{m+1}) \end{bmatrix} + \mathcal{O}(h) \begin{bmatrix} \mathbf{0} \\ G(\mathbf{S}_2, S_{m+2}) \end{bmatrix} + \sum_{j=m+3}^{M-1} \mathcal{O}(h^5) \begin{bmatrix} \mathbf{0} \\ G(\mathbf{S}_2, S_j) \end{bmatrix}. \tag{2.36}$$

Since the summation terms in (2.36) are negligible compared to the $\mathcal{O}(h)$-coefficient terms, and since $G(\mathbf{S}_2, S_j) = \mathcal{O}(h)$ for $j \geq m+1$, as shown in Proposition 2.2.3, relation (2.35) is proved. Further, from Proposition 2.2.3, it is clear that the Green's functions $G(S, S_j)$ are piecewise smooth with first-derivative jumps at points S_j, and smooth for $S \geq S_{m+2}$. This proves the "smoothness" of the error e_i for $i \geq m+2$. □

Remark 2.2.4. *The matrix $\mathbf{L}$ in (2.8) representing the discretization of (2.2) by a high-order method does not satisfy the M-property, and it is not diagonally dominant. However, by numerical experiments for the cases of coefficient functions in (2.2) and grid spacings considered in this chapter, we noticed that each of $-\mathbf{L}$ and $-\mathbf{L} + \rho\mathcal{I}$ have non-negative inverses, i.e. they are monotone. While we do not derive conditions under which the matrices are monotone, we note that a high-order method does not preclude monotonicity [4, 8]. Notice also that Figure 2.2, that plots the first three columns of $-\mathbf{L}_{22}^{-1}$, which is the dominant part of $(-\mathbf{L} + \rho\mathcal{I})^{-1}$, supports the conjecture that $-\mathbf{L} + \rho\mathcal{I}$ is monotone. We also note that, while the M-property and diagonal dominance of the matrix are common arguments used in the convergence study of the penalty iteration [21, 6], in both these references, it is argued that such conditions are only sufficient and not necessary.*

Initial value problems

When the time variable is included, the analysis becomes more involved. However, the conclusions are similar to the ones for boundary value problems. In this section, we study the single step error behavior when we solve (2.6) using fourth-order finite difference discretization and BDF4 time-stepping. While the stability analysis is important, we observed empirically that our time-stepping scheme is generally stable in practice. Hence, we leave the stability analysis of the time-stepping scheme for future research, and only focus on single step error behavior.

Consider the original LCP given by (2.1), and the corresponding penalized PDE (2.6). As in Proposition 2.2.1 we have relation (2.30), it is easy to see that, in the case of time-dependent problems, we have

$$\partial_t \mathbf{V}_a = \mathbf{L}\mathbf{V}_a + \mathbf{b} + \boldsymbol{\gamma} + \boldsymbol{\theta}, \qquad \text{with } [\boldsymbol{\gamma}(t)]_j \equiv ((\partial_t - \mathcal{L})V_a(t, S_j) - g(t, S_j))\, \mathbb{1}_{1\le j\le m(t)}, \tag{2.37}$$

where $m(t)$ is the node index at time t such that $S_{m(t)} \le S_f(t) < S_{m(t)+1}$.

Consider the BDF4 discretization applied to Equations (2.37) starting at the fourth time step. We have, for the exact LCP solution,

$$\mathbf{A}\mathbf{V}_a^{n+4} = 4\mathbf{V}_a^{n+3} - 3\mathbf{V}_a^{n+2} + \frac{4}{3}\mathbf{V}_a^{n+1} - \frac{1}{4}\mathbf{V}_a^{n} + k\mathbf{b}^{n+4} + k(\boldsymbol{\gamma} + \boldsymbol{\theta} + \boldsymbol{\beta}), \tag{2.38}$$

where $\boldsymbol{\beta}$ is the truncation error of the BDF4 time-stepping scheme applied to (2.37). Note that the fully discrete system that we are actually solving is Equation (2.13). The following proposition gives the relationship between the solution V_a of the exact LCP and the solution $\tilde{\mathbf{V}}^{n+4}$ of the fully discrete system (2.13). To simplify the notation, in the proposition as well as the theorem following, we drop the superscript $n+4$ from the computed solution $\tilde{\mathbf{V}}^{n+4}$. Note that we have also dropped the superscript from $\boldsymbol{\gamma}, \boldsymbol{\theta}$ and $\boldsymbol{\beta}$ for simplicity.

Proposition 2.2.4. *Let V_a be the solution to the exact LCP (2.1), and V be the solution to the continuous penalized PDE (2.6). Assume that the penalty terms in the fully discrete equations reflect the correct behavior, i.e., $\mathcal{I}_{\tilde{\mathbf{V}}} = \mathcal{I}_{\mathbf{V}_a}$ at each time step. Then, we have*

$$\left(\frac{25}{12k}\mathbf{I} - \mathbf{L} + \rho\mathcal{I}_{\mathbf{V}_a}\right)\mathbf{e}^{n+4} = \frac{1}{k}\left(4\mathbf{e}^{n+3} - 3\mathbf{e}^{n+2} + \frac{4}{3}\mathbf{e}^{n+1} - \frac{1}{4}\mathbf{e}^{n}\right) + (\boldsymbol{\gamma} + \boldsymbol{\theta} + \boldsymbol{\beta}), \tag{2.39}$$

where $\mathbf{e} = \mathbf{V}_a - \tilde{\mathbf{V}}$ is error in the solution.

Proof. First, notice that $\rho\mathcal{I}_{\tilde{\mathbf{V}}}(\mathbf{V}^* - \tilde{\mathbf{V}}) = \rho\mathcal{I}_{\mathbf{V}_a}(\mathbf{V}_a - \tilde{\mathbf{V}})$, since

$$\rho\mathcal{I}_{\tilde{\mathbf{V}}}(\mathbf{V}^*-\tilde{\mathbf{V}}) = \rho\mathcal{I}_{\tilde{\mathbf{V}}}(\mathbf{V}^*-\mathbf{V}_a+\mathbf{V}_a-\tilde{\mathbf{V}}) = \rho\mathcal{I}_{\mathbf{V}_a}(\mathbf{V}^*-\mathbf{V}_a)+\rho\mathcal{I}_{\tilde{\mathbf{V}}}(\mathbf{V}_a-\tilde{\mathbf{V}}) = \rho\mathcal{I}_{\tilde{\mathbf{V}}}(\mathbf{V}_a-\tilde{\mathbf{V}}) = \rho\mathcal{I}_{\mathbf{V}_a}(\mathbf{V}_a-\tilde{\mathbf{V}}).$$

Using this identity, we subtract Equation (2.13) from (2.38) and rearrange to get Equation (2.39). □

Corresponding to Proposition 2.2.1 in the previous section which describes the error equation for time-independent free boundary problems, Proposition 2.2.4 gives an expression of the error evolution for solving time-dependent free boundary problems using a fourth-order finite difference

scheme and BDF4. For convenience of discussion, define

$$\mathbf{L}_k \equiv \frac{25}{12k}\mathbf{I} - \mathbf{L}, \text{ and } \mathbf{r}_k^{n+3} \equiv \frac{1}{k}\left(4\mathbf{e}^{n+3} - 3\mathbf{e}^{n+2} + \frac{4}{3}\mathbf{e}^{n+1} - \frac{1}{4}\mathbf{e}^{n}\right) + (\boldsymbol{\gamma} + \boldsymbol{\theta} + \boldsymbol{\beta}). \tag{2.40}$$

In addition, assume that, at the $(n+4)$-th time step, the free boundary is located in between S_m and S_{m+1}, i.e., $S_m \leq S_f(t_{n+4}) < S_{m+1}$, on the space grid $\{S_0, S_1, \ldots, S_m, \ldots S_{M-1}\}$. Similar to the discussion for boundary value problems, we divide the matrix $\mathbf{L}_k$ into four block submatrices

$$\mathbf{L}_k = \begin{bmatrix} [\mathbf{L}_k]_{11} & [\mathbf{L}_k]_{12} \\ [\mathbf{L}_k]_{21} & [\mathbf{L}_k]_{22} \end{bmatrix}$$

where the block matrices $[\mathbf{L}_k]_{11}$, $[\mathbf{L}_k]_{12}$, $[\mathbf{L}_k]_{21}$, $[\mathbf{L}_k]_{22}$ are of sizes $m \times m$, $m \times (M-1-m)$, $(M-1-m) \times m$ and $(M-1-m) \times (M-1-m)$, respectively. Corresponding to Theorem 2.2.3 for time-independent problems, the error decomposition for time-dependent problems is given by Theorem 2.2.5.

Theorem 2.2.5. *Assume that $[\mathbf{L}_k]_{11}$ and $[\mathbf{L}_k]_{22}$ are nonsingular, and ρ is a positive number such that $\rho \gg \max_{ij}\{|[\mathbf{L}_k]_{i,j}|\}$. Assume also that $\max_j\{|[[\mathbf{L}_k]_{22}^{-1}]_{1,j}|\} = \mathcal{O}(h^2)$, $\max_j\{|[[\mathbf{L}_k]_{22}^{-1}]_{2,j}|\} = \mathcal{O}(h^2)$, $\max_i\{|[[\mathbf{L}_k]_{22}^{-1}]_{i,1}|\} = \mathcal{O}(h^2)$, and $\max_i\{|[[\mathbf{L}_k]_{22}^{-1}]_{i,2}|\} = \mathcal{O}(h^2)$. Further, assume that $\mathcal{I}_{\tilde{\mathbf{V}}} = \mathcal{I}_{\mathbf{V}_a}$ at each time step. When using BDF4 time-stepping and the fourth-order finite difference scheme to solve the penalized PDE (2.6), the solution error at the $(n+4)$-th time step satisfies*

$$\mathbf{e}^{n+4} = \begin{bmatrix} \mathbf{O} \\ ([\mathbf{L}_k]_{22})^{-1}\,[\mathbf{r}_k^{n+3}]_{m+1:M-1} \end{bmatrix} + \mathcal{O}\left(\frac{1}{\rho}\right)\mathbf{J}_{(M-1)\times(M-1)}.$$

Remark 2.2.5. *We denote $([\mathbf{L}_k]_{22})^{-1}$ by $[\mathbf{L}_k]_{22}^{-1}$ for notational simplicity.*

Proof. From Proposition 2.2.4, we know that the error is the solution to

$$(\mathbf{L}_k + \rho\mathcal{I}_{\tilde{\mathbf{V}}})\mathbf{e}^{n+4} = \mathbf{r}_k^{n+3}.$$

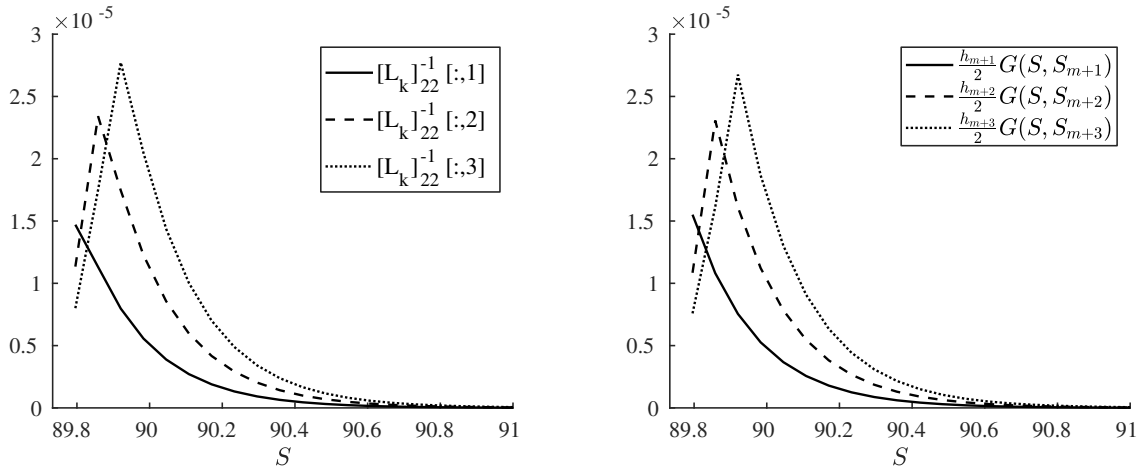
Therefore, we get

$$\mathbf{e}^{n+4} = (\mathbf{L}_k + \rho\mathcal{I}_{\tilde{\mathbf{V}}})^{-1}\mathbf{r}_k^{n+3}.$$

Then by making use of Proposition 2.2.2 applied to $\mathbf{L}_k$, the theorem is proved. □

Remark 2.2.6. *Note that similar to Proposition 2.2.2, the assumptions on $[\mathbf{L}_k]_{22}^{-1}$ are typically true in practice.*

Theorem 2.2.5 shows that the errors in the approximate solutions of moving boundary problems behave in a similar way to the solution of free boundary problems. The solution on the penalty region is computed exactly within a tolerance, while on the PDE region, in addition to the truncation errors, the solution errors from previous time steps also contribute to the solution error at the current time step. The error propagation is governed by $[\mathbf{L}_k]_{22}^{-1}$, which depends on both the time stepping and space discretization schemes. Similar to the discussion of boundary value problems,



(a) The first 3 columns of $[\mathbf{L}_k]_{22}^{-1}$ (b) The continuous Green's function $\frac{h_j}{2}G(S, S_j)$

Figure 2.4: (a) The first three columns of $[\mathbf{L}_k]_{22}^{-1}$ on an example nonuniform grid; (b) The scaled continuous Green's function $\frac{h_j}{2}G(S, S_j)$ for the operator $\mathcal{L}_{BS}$ at S_{m+1}, S_{m+2} and S_{m+3}. The free boundary location is $S_f = 89.748$. Note that the zero value on the left of the free boundary is not included.

it can be treated as the discrete analogue of the Green's function to the continuous operator on the PDE region.

In the following, we consider a concrete example of the Black-Scholes operator, i.e., let $\mathcal{L} = \mathcal{L}_{BS}$. Instead of studying $\mathbf{L}_k$, which is hard to analyze, we investigate the Green's function for the continuous operator $\mathcal{L}_k = \frac{25}{12k} - \mathcal{L}$ on the PDE region for a fixed time step size k. In Figure 2.4, we show the comparison of the graphs of $G(\mathbf{S}_2, S_j)$ on the PDE region and $[\mathbf{L}_k]_{22}^{-1}$ for the first three Green's functions on an example grid, where the free boundary is located at $S_f = 89.748$. Again, we see that they have the same shape with similar magnitudes. By performing the usual variable transformation $S = Ke^x$ to $\mathcal{L}_k$, we can get a transformed operator $\mathcal{L}_{k,x} = -\frac{\partial^2}{\partial x^2} - (\kappa - 1)\frac{\partial}{\partial x} + (\kappa + \frac{25}{6k\sigma^2})$, whose Green's function is given by Equation (2.34) in Proposition 2.2.3, with $\xi_1 = \frac{-(\kappa-1)+\sqrt{(\kappa+1)^2+4\lambda}}{2}$, $\xi_2 = \frac{-(\kappa-1)-\sqrt{(\kappa+1)^2+4\lambda}}{2}$, where $\lambda = \frac{25}{6k\sigma^2}$, and $\kappa = \frac{2r}{\sigma^2}$. Let $x_{m+1} = \log(S_{m+1}/K)$ and $x_f = \log(S_f/K)$. When $x_{m+1} - x_f$ is small enough, we have that $G(x, x_{m+1}) = \mathcal{O}(x_{m+1} - x_f)$ by Proposition 2.2.3. Since

$$x_{m+1} - x_f = \log(S_{m+1}/K) - \log(S_f/K) = \log\left(1 + \frac{S_{m+1} - S_f}{S_f}\right) \approx \frac{S_{m+1} - S_f}{S_f} = \mathcal{O}(h),$$

when $S_{m+1} \to S_f > 0$, the Green's function $G(S, S_{m+1}) = \mathcal{O}(h)$. Using the same argument as in Section 2.2.3, we claim that the error in the solution of Equation (2.13) is of order $\mathcal{O}(h^2)$ and smooth for $S \geq S_{m+2}$.

Therefore, we have the following theorem.

Theorem 2.2.6. *Under the same assumptions as in Theorem 2.2.5, we have that the error at S_i, at the $(n+4)$-th time step, is $e_i^{n+4} \approx \mathcal{O}(h^2)$, and it is smooth for $i \geq m+2$.*

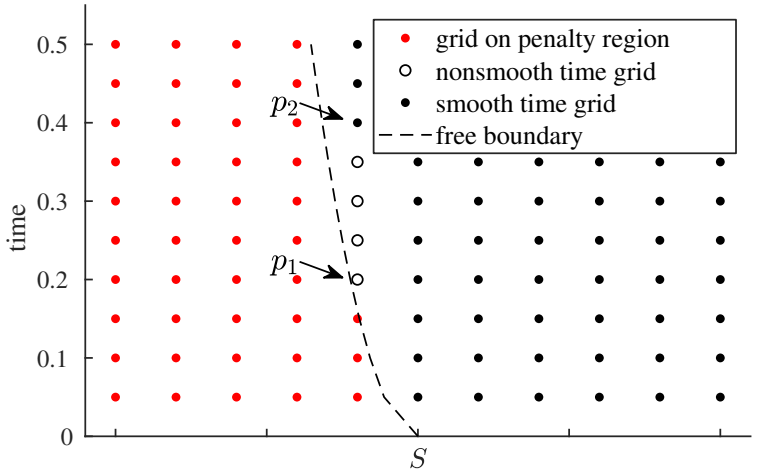


Figure 2.5: An example layout of grid points in the time and space domain. The dashed line is the free boundary. The red points to the left of the free boundary are on the penalty region, and the points to the right of the free boundary are on the PDE region. Unlike at the solid black points, BDF4 has degenerated accuracy at the hollow black points because it involves solution points that lie on different sides of the free boundary.

2.2.4 Grid crossing

With BDF4 time-stepping, the time derivative of the solution at some point is computed by a linear combination of the solutions at the four points directly prior to the current point. However, one of more of the prior points may not lie on the same side of the moving boundary.

To see this, we can look at an example grid shown in Figure 2.5. The black hollow points, for example p_1, are problematic points for the BDF4 time-stepping scheme, because their computation depends on one or more points on the other side of the moving boundary. Therefore, the BDF4 scheme at the black hollow points may exhibit degenerated accuracy.

On the other hand, for the black solid points, such as p_2, the time derivative of the solution with the BDF4 scheme uses only points on the same side of the moving boundary. Hence, the BDF4 scheme at the black solid points is fourth-order accurate. In general, we can see that the number of problematic points depends on how quickly the moving boundary moves relative to the grid spacing. On a fixed grid, a slow-moving free boundary will have a smaller number of black hollow points.

At minimum, we require the time discretization to be accurate enough so as not to affect the convergence order in space. Despite the complicated behavior of the time derivative discretization, we do not explicitly deal with the loss of accuracy in the time derivative in our algorithm. Instead, we apply a time variable transformation to change the shape of the free boundary and try to reduce the number of problematic points in the time stepping. This time transformation together with appropriate space stretching turn out to be good enough to maintain high-order accuracy.

2.2.5 Extrapolating the numerical solution

In this section, we discuss how to approximate the derivative jumps and the free boundary location using a given numerical solution, and analyze some related technical details. We present the details for one free boundary. In the case there are multiple free boundaries, we assume that the finite difference stencils can only cross one free boundary at most. If this is not true, we apply a grid stretching so that we obtain enough grid point between free boundaries and this assumption holds.

Approximation of the solution derivatives and associated derivative jumps

In this subsection, we discuss how to approximate the derivative jumps $\Delta V''_{S_f} \equiv V''_{S_f,-} - V''_{S_f,+}$, $\Delta V'''_{S_f} \equiv V'''_{S_f,-} - V'''_{S_f,+}$, and $\Delta V''''_{S_f} \equiv V''''_{S_f,-} - V''''_{S_f,+}$, where the "$-$" and the "$+$" in the subscripts denote values in the penalty and the PDE regions, respectively.

In the penalty region, since we are given the obstacle function V^*, and we know that $V = V^*$, we can evaluate $V''_{S_f,-}$, $V'''_{S_f,-}$ and $V''''_{S_f,-}$ exactly.

We now turn to the PDE region and discuss the approximation of $V''_{S_f,+}$, $V'''_{S_f,+}$ and $V''''_{S_f,+}$. Recall that $S_m \leq S_f < S_{m+1}$, as shown on an example uniform grid in Figure 2.6. From Theorem 2.2.4, we know that $e_i \approx \mathcal{O}(h^2)$ and e_i is smooth, for $i \geq m+2$. Therefore, by having $O(h^2)$ accurate values $\tilde{V}_{m+2}, \tilde{V}_{m+3}, \tilde{V}_{m+4}$, we construct $O(h^2)$ accurate second derivative values $\tilde{V}''_{m+2}, \tilde{V}''_{m+3}, \tilde{V}''_{m+4}$. To approximate the second derivative of the solution at the free boundary $S_f < S_{m+1}$ to $\mathcal{O}(h^2)$ accuracy, we extrapolate the solution derivative at S_f using the computed $\tilde{V}''_{m+2}, \tilde{V}''_{m+3}, \tilde{V}''_{m+4}$ by

$$\tilde{V}''(S) = \sum_{i=2}^{4} \left(\tilde{V}''_{m+i} \prod_{j=2, j\neq i}^{4} \frac{S - S_{m+j}}{S_{m+i} - S_{m+j}} \right),$$

i.e., by using a quadratic Lagrange polynomial. The obtained approximate derivative at S_f is of $\mathcal{O}(h^2)$ accuracy. Higher-order derivatives are computed in the same way. Hence, from the analysis in Section 2.2.2, we see that the correction terms computed using the approximate derivative jumps will be of $\mathcal{O}(h^2)$ accurate, which is more than the required $\mathcal{O}(h)$ accuracy to increase the order of the corrected finite differences. Similarly, we use $\tilde{V}_{m+2}, \tilde{V}_{m+3}, \tilde{V}_{m+4}, \tilde{V}_{m+5}$ and a cubic polynomial for the third-order approximation, and $\tilde{V}_{m+2}, \tilde{V}_{m+3}, \tilde{V}_{m+4}, \tilde{V}_{m+5}, \tilde{V}_{m+6}$ and a quartic polynomial for the fourth-order approximation of the solution derivatives at the free boundary.

We stress here that the choice of the interpolation points starting from S_{m+2} (skipping S_{m+1}) to maintain the convergence order is guided by our analysis of the error behavior in the solution given by Equation (2.36). Similar interpolation scheme can be found in [37], where the authors are only able to justify their results empirically from numerical experiments. Note that we could have used one degree less Lagrange interpolation in each case, but our choice is dictated by the fact that we want the extrapolation error to be of even lower order than the solution error. We also note that, while extrapolation may introduce additional errors, it is used to avoid areas where the error is non-smooth and of order incompatible to the solution. Further, we emphasize that we extrapolate less than $2h$ away from the smooth data points.

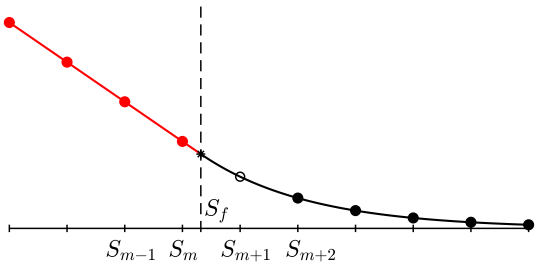


Figure 2.6: An example grid on which a free boundary problem is defined. Point S_f is the free boundary location.

Approximation of the free boundary location

To approximate the free boundary location, we apply the smooth pasting condition of the derivative at the free boundary in Equation (2.5), which we repeat here for convenience:

$$\frac{\partial V}{\partial S}(t, S_f(t)) = \frac{\partial V^*}{\partial S}(S_f(t)). \tag{2.41}$$

Assume that we have already calculated the approximate derivatives $\tilde{V}'_{m+2}$, $\tilde{V}'_{m+3}$, $\tilde{V}'_{m+4}$, $\tilde{V}'_{m+5}$ and $\tilde{V}'_{m+6}$ at S_{m+2}, S_{m+3}, S_{m+4}, S_{m+5}, S_{m+6}, respectively, with certain accuracy, using solution values starting from S_{m+2} and on. We then apply the quartic Lagrange polynomial to fit the derivative by

$$\tilde{V}'(S) = \sum_{i=2}^{6} \left(\tilde{V}'_{m+i} \prod_{j=2, j\neq i}^{6} \frac{S - S_{m+j}}{S_{m+i} - S_{m+j}} \right).$$

Then, the approximate free boundary is obtained by Newton's root finding algorithm such that

$$\tilde{V}'(S) - \frac{\partial V^*}{\partial S}(S) \approx 0.$$

The approximate free boundary obtained in this way is of the same order as the numerical solution. Hence, from the analysis in Section 2.2.2, by using $\mathcal{O}(h^2)$ accurate solution, the approximate free boundary is $\mathcal{O}(h^2)$, and the correction terms computed are $\mathcal{O}(h)$, as is required to increase the order of the corrected finite differences. Similarly, using $\mathcal{O}(h^3)$ accurate solution, the correction terms computed are $\mathcal{O}(h^2)$, and so on. Note that we decided to use the smooth pasting condition (2.41) to locate the free boundary instead of the value matching condition $V(t, S_f(t)) = V^*(S_f(t))$. The reason for this is that the value matching equation has a zero derivative at the root, resulting in a double root. Therefore, Newton's root-finding method is slow, if value matching is used. However, we note that, for problems where the value matching equation holds, while the smoothing pasting condition does not hold, e.g. cases of variance gamma, jump diffusion, American options, we do not rule out that solving the value matching equation may work well. We leave this for further research.

2.3 Algorithm

2.3.1 A fourth-order deferred correction algorithm for solving free boundary problems

We are now ready to present a fourth-order deferred correction finite difference algorithm for solving free and moving boundary problems. To start, we first present the algorithm for solving free boundary problems where no time variable is involved. Recall that the penalized equation for solving free boundary problems is given by Equation (2.7), and the respective discrete equations are given by the nonlinear system (2.14), solved by a generalized Newton's iteration as described in [21], which is also referred to as discrete penalty iteration.

The main idea of our algorithm is to use a deferred correction technique to eliminate the lower-order errors in the finite difference approximation introduced by piecewise smoothness in the solution. We illustrate our correction scheme by considering only the leading order terms of the corrections in Equations (2.22)–(2.25), that is, those associated with the jump in the second derivative. The other terms are corrected in the same manner and we omit the discussion.

Recall that the fourth-order FD discretization of $\frac{\partial^2 V}{\partial S^2}$ and $\frac{\partial V}{\partial S}$ is $\bar{\mathbf{L}}_2\tilde{\mathbf{V}}_{\text{aug}}$ and $\bar{\mathbf{L}}_1\tilde{\mathbf{V}}_{\text{aug}}$, respectively, at the interior nodes of a grid $S_0 < S_1 < \ldots < S_M$, where $\bar{\mathbf{L}}_2$ and $\bar{\mathbf{L}}_1$ are $M-1 \times (M+1)$ second- and first-derivative, respectively, FD coefficient matrices, and $\tilde{\mathbf{V}}_{\text{aug}} = [\tilde{V}_0, \tilde{V}_1, \ldots, \tilde{V}_M]^T$, as defined in Section 2.2.1. Suppose that $S_m \le S_f < S_{m+1}$, as shown in Figure 2.6. The second derivative jump at the free boundary is pre-computed to be $\Delta V''_{S_f}$ (either approximate or exact). From Theorem 2.2.2, making $\delta = S_{m+1} - S_f$, and picking appropriate FD coefficients, we see that the correction terms corresponding to the second derivative jumps at nodes S_{m-1}, S_m, S_{m+1} and S_{m+2}, are computed by

$$C''_{1,0} = \frac{(S_{m+1}-S_f)^2}{2}\bar{\mathbf{L}}_2(m-1, m+2)\Delta V''_{S_f}, \tag{2.42}$$

$$C''_{2,0} = \left(\frac{(S_{m+1}-S_f)^2}{2}\bar{\mathbf{L}}_2(m, m+2) + \frac{(S_{m+2}-S_f)^2}{2}\bar{\mathbf{L}}_2(m, m+3)\right)\Delta V''_{S_f}, \tag{2.43}$$

$$C''_{3,0} = -\left(\frac{(S_f-S_{m-1})^2}{2}\bar{\mathbf{L}}_2(m+1, m) + \frac{(S_f-S_m)^2}{2}\mathbf{L}_2(m+1, m+1)\right)\Delta V''_{S_f}, \tag{2.44}$$

$$C''_{4,0} = -\frac{(S_f-S_m)^2}{2}\bar{\mathbf{L}}_2(m+2, m+1)\Delta V''_{S_f}, \tag{2.45}$$

where $\bar{\mathbf{L}}_2(i,j)$ denotes the (i,j) entry of the coefficient matrix $\bar{\mathbf{L}}_2$, and $\Delta V''_{S_f} \equiv \lim_{S\uparrow S_f(t)} V''(t,S) - \lim_{S\downarrow S_f(t)} V''(t,S)$. Note that $C''_{j,0}$, $j = 1,\ldots,4$, correspond to $\mathcal{O}(1)$ error terms in (2.22)-(2.25). The correction terms corresponding to the third and fourth derivative jumps are computed similarly, giving rise to $C''_{j,1}$, $j = 1,\ldots,4$ (corresponding to $\mathcal{O}(h)$ error terms in (2.22)-(2.25)) and $C''_{j,2}$, $j = 1,\ldots,4$ (corresponding to $\mathcal{O}(h^2)$ error terms in (2.22)-(2.25)), respectively. Then, the total correction terms for the FD approximation of the second derivative at nodes S_{m-1}, S_m, S_{m+1} and

S_{m+2} are

$$C_1'' = C_{1,0}'' + C_{1,1}'' + C_{1,2}'', \tag{2.46}$$
$$C_2'' = C_{2,0}'' + C_{2,1}'' + C_{2,2}'', \tag{2.47}$$
$$C_3'' = C_{3,0}'' + C_{3,1}'' + C_{3,2}'', \tag{2.48}$$
$$C_4'' = C_{4,0}'' + C_{4,1}'' + C_{4,2}''. \tag{2.49}$$

By replacing the $\bar{\mathbf{L}}_2$ entries with the corresponding $\bar{\mathbf{L}}_1$ entries, we can similarly calculate the correction terms C_j' to the first derivative approximations, for $j = 1, 2, 3, 4$, at nodes S_{m-1}, S_m, S_{m+1} and S_{m+2}, respectively.

With these correction entries in hand, instead of solving Equation (2.14), we solve a modified system

$$\mathbf{L}\tilde{\mathbf{V}} + \mathbf{b} + \rho\mathcal{I}_{\tilde{\mathbf{V}}}(\mathbf{V}^* - \tilde{\mathbf{V}}) + \mathbf{a}_2 + \mathbf{a}_1 = \mathbf{0}, \tag{2.50}$$

with correction terms $\mathbf{a}_2$ and $\mathbf{a}_1$, where

$$\mathbf{a}_2 = [0,\ \ldots,\ 0,\ p(S_{m-1})C_1'',\ p(S_m)C_2'',\ p(S_{m+1})C_3'',\ p(S_{m+2})C_4'',\ 0,\ \ldots,\ 0]^T, \tag{2.51}$$
$$\mathbf{a}_1 = [0,\ \ldots,\ 0,\ w(S_{m-1})C_1',\ w(S_m)C_2',\ w(S_{m+1})C_3',\ w(S_{m+2})C_4',\ 0,\ \ldots,\ 0]^T. \tag{2.52}$$

Ideally, we would know the exact derivative jumps and apply them to correct the FDs when discretizing the PDE. However, these jumps are not known a priori in the setting of this work. Therefore, we make use of the approximate solution derivatives and free boundary location that we get, as described in Section 2.2.5, to compute the approximate correction terms. Applying these correction terms increases the order of the finite difference approximation. As a result, it also increases the order of the new approximate solution when we solve the discrete system again with the corrected FDs. We give the details in Algorithm 1.

Remark 2.3.1. *In fact, the algorithm already reaches fourth-order accuracy with two corrections. However, from numerical results presented later, it turns out the third correction improves the error noticeably. Therefore, we present the algorithm with three corrections. One could theoretically go beyond three corrections, however, the order cannot be further improved due to the order of finite differences being used. Based on numerical experiments, we conjecture that further corrections, as long as they involve second, third and fourth derivative jumps, would converge and may slightly improve the error (not the order), but possibly not enough to justify the extra cost. Furthermore, we believe that iterating would converge to the solution of the problem with the jumps incorporated into the left-hand side matrix (extrapolated problem) and solved for simultaneously with the vector of unknowns.*

We also note that, for a fourth-order discretization method, it does not make sense to generate jumps for higher order derivatives, and proceed with more corrections.

Remark 2.3.2. *We presented the algorithm for a fourth-order discretization method and applied three corrections, but the idea can be extended to higher order methods. For a $\zeta-$order discretiza-*

Algorithm 1 A fourth-order FD algorithm for solving free boundary problems

1: **Phase 1**:
Solve (2.14) to obtain an approximate solution $\tilde{\mathbf{V}}^{(0)}$ of $\mathcal{O}(h^2)$ using penalty iteration.
Find S_m and S_{m+1} as in Proposition 2.2.1.
Compute the approximate free boundary $\tilde{S}_f^{(0)}$ to $\mathcal{O}(h^2)$ accuracy, using Newton's method with initial guess $(S_m + S_{m+1})/2$, as in Section 2.2.5.
Approximate $V''(S_f)$ at $\tilde{S}_f^{(0)}$ to obtain $\tilde{V}_{S_f}^{''(0)} = V''(S_f) + O(h)$, as in Section 2.2.5.

2: **Phase 2**:
Compute the FD corrections $\mathbf{a}_1$, $\mathbf{a}_2$ using $\tilde{S}_f^{(0)}$ and $V_{S_f}^{*}{}'' - \tilde{V}_{S_f}^{''(0)}$.
Solve (2.50) to obtain an approximate solution $\tilde{\mathbf{V}}^{(1)}$ of $\mathcal{O}(h^3)$ using penalty iteration with initial guess $\tilde{\mathbf{V}}^{(0)}$.
Compute the approximate free boundary $\tilde{S}_f^{(1)}$ to $\mathcal{O}(h^3)$ accuracy, using Newton's method with initial guess $\tilde{S}_f^{(0)}$.
Approximate $V''(S_f)$, $V'''(S_f)$ at $\tilde{S}_f^{(1)}$ to obtain $\tilde{V}_{S_f}^{''(1)} = V''(S_f) + \mathcal{O}(h^2)$, $\tilde{V}_{S_f}^{'''(1)} = V'''(S_f) + \mathcal{O}(h)$.

3: **Phase 3**:
Compute the FD corrections $\mathbf{a}_1$, $\mathbf{a}_2$ using $\tilde{S}_f^{(1)}$ and $V_{S_f}^{*}{}'' - \tilde{V}_{S_f}^{''(1)}$, $V_{S_f}^{*}{}''' - \tilde{V}_{S_f}^{'''(1)}$.
Solve (2.50) to obtain an approximate solution $\tilde{\mathbf{V}}^{(2)}$ of $\mathcal{O}(h^4)$ using penalty iteration with initial guess $\tilde{\mathbf{V}}^{(1)}$.
Compute the approximate free boundary $\tilde{S}_f^{(2)}$ to $\mathcal{O}(h^4)$ accuracy, using Newton's method with initial guess $\tilde{S}_f^{(1)}$.
Approximate $V''(S_f)$, $V'''(S_f)$, $V''''(S_f)$ at $\tilde{S}_f^{(2)}$ to obtain $\tilde{V}_{S_f}^{''(2)} = V''(S_f) + \mathcal{O}(h^3)$, $\tilde{V}_{S_f}^{'''(2)} = V'''(S_f) + \mathcal{O}(h^2)$, $\tilde{V}_{S_f}^{''''(2)} = V''''(S_f) + \mathcal{O}(h)$.

4: **Phase 4**:
Compute the FD corrections $\mathbf{a}_1$, $\mathbf{a}_2$ using $\tilde{S}_f^{(2)}$ and $V_{S_f}^{*}{}'' - \tilde{V}_{S_f}^{''(2)}$, $V_{S_f}^{*}{}''' - \tilde{V}_{S_f}^{'''(2)}$, $V_{S_f}^{*}{}'''' - \tilde{V}_{S_f}^{''''(2)}$.
Solve (2.50) to obtain an approximate solution $\tilde{\mathbf{V}}^{(3)}$ using penalty iteration with initial guess $\tilde{\mathbf{V}}^{(2)}$.
Compute the approximate free boundary $\tilde{S}_f^{(3)}$, using Newton's method with initial guess $\tilde{S}_f^{(2)}$.

tion method, we could apply $\zeta - 1$ corrections, the first correction (Phase 2) involving the second derivative, and proceeding so that each subsequent correction involves one additional derivative of higher order, up to the $\zeta-$order derivative.

2.3.2 A fourth-order deferred correction algorithm for solving moving boundary problems

When solving moving boundary problems, we also need to consider time discretization. In order to show the flow of computations as the correction phases and timesteps proceed, we introduce a double index, with n denoting the timestep and ℓ (in parentheses) the correction phase. We assume that Equation (2.6) has been discretized in time by BDF4, except for the first three time steps, and in space by standard fourth-order FDs, resulting in the nonlinear system (2.13).

At this point, we make a note regarding the choice of ρ in the discrete problem. As discussed in [21], it may be appropriate to adjust the value of ρ for each refinement of the grid in a way so that the error arising from the approximation of the LCP by the penalized nonlinear PDE reduces at the same rate as the discretization error. However, it is more practical to set a small enough target relative error tolerance *tol* in the approximation of the LCP by the penalized nonlinear PDE. Following the same arguments as in [21], and under similar boundedness assumptions, we essentially scale ρ as k^{-1} in (2.13), or, equivalently, solve, with a fixed ρ, the nonlinear system

$$\mathbf{A}\tilde{\mathbf{V}}^{n+4,(\ell)} = \tilde{\mathbf{y}}^{n+4,(\ell)} + \rho\boldsymbol{\mathcal{I}}_{\tilde{\mathbf{V}}^{n+4,(\ell)}}(\mathbf{V}^* - \tilde{\mathbf{V}}^{n+4,(\ell)}), \tag{2.53}$$

where

$$\mathbf{A} = \left(\frac{25}{12}\mathbf{I} - k\mathbf{L}\right), \quad \tilde{\mathbf{y}}^{n+4,(\ell)} = 4\tilde{\mathbf{V}}^{n+3,(\ell)} - 3\tilde{\mathbf{V}}^{n+2,(\ell)} + \frac{4}{3}\tilde{\mathbf{V}}^{n+1,(\ell)} - \frac{1}{4}\tilde{\mathbf{V}}^{n,(\ell)} + k\tilde{\mathbf{b}}^{n+4}. \tag{2.54}$$

When we apply corrections to Equation (2.53), we solve a modified system

$$\mathbf{A}\tilde{\mathbf{V}}^{n+4,(\ell)} = \tilde{\mathbf{y}}^{n+4,(\ell)} + \rho\boldsymbol{\mathcal{I}}_{\tilde{\mathbf{V}}^{n+4,(\ell)}}(\mathbf{V}^* - \tilde{\mathbf{V}}^{n+4,(\ell)}) + k(\mathbf{a}_1 + \mathbf{a}_2). \tag{2.55}$$

with correction terms $\mathbf{a}_1$ and $\mathbf{a}_2$ computed in the same way as for free boundary problems discussed in the previous section. Only slight modifications to Algorithm 1 are required to include BDF4 time-stepping and solve moving boundary problems. Our high-order finite difference method for solving moving boundary problems is given in Algorithm 2.

Remark 2.3.3. *Although we have given the BDF time stepping formula assuming uniform time step size, Algorithm 2 is presented in a more general way, referring to general time points t_{n+3}, t_{n+2}, t_{n+1}, t_n, as, in cases when the solution evolves rapidly or exhibits nonsmoothness in certain time periods, a variable or adaptive timestep BDF can be used, to maintain small local errors at all time steps. It is also important to note that, to produce quantities for the ℓth phase at t_{n+4}, data from the ℓth phase of timesteps t_{n+3}, t_{n+2}, t_{n+1} and t_n are used.*

Algorithm 2 A fourth-order FD algorithm for solving moving boundary problems

1: **for** each time step t_n **do**

2: **Phase 1** ($\ell = 0$):

Solve (2.53) to obtain an approximate solution $\tilde{\mathbf{V}}^{n,(0)}$ of $\mathcal{O}(h^2)$ using penalty iteration with initial guess $\tilde{\mathbf{V}}^{n-1,(0)}$.

Find S_m, S_{m+1} at t_n as in Proposition 2.2.1.

Compute the approximate free boundary $\tilde{S}_f^{n,(0)}$ to $\mathcal{O}(h^2)$ accuracy, using Newton's method with initial guess $(S_m + S_{m+1})/2$ as in Section 2.2.5.

Approximate $V''(t_n, S_f)$ at $\tilde{S}_f^{n,(0)}$ to obtain $\tilde{V}_{S_f}^{''(0)} = V''(t_n, S_f) + O(h)$, as in Section 2.2.5.

3: **Phase 2** ($\ell = 1$):

Compute the FD corrections $\mathbf{a}_1$, $\mathbf{a}_2$ using $\tilde{S}_f^{n,(0)}$ and $V_{S_f}^{*\,''} - \tilde{V}_{S_f}^{''n,(0)}$.

Solve (2.55) to obtain an approximate solution $\tilde{\mathbf{V}}^{n,(1)}$ of $\mathcal{O}(h^3)$ using penalty iteration with initial guess $\tilde{\mathbf{V}}^{n,(0)}$.

Compute the approximate free boundary $\tilde{S}_f^{n,(1)}$ to $\mathcal{O}(h^3)$ accuracy, using Newton's method with initial guess $\tilde{S}_f^{n,(0)}$.

Approximate $V''(t_n, S_f)$, $V'''(t_n, S_f)$ at $\tilde{S}_f^{n,(1)}$ to obtain $\tilde{V}_{S_f}^{''n,(1)} = V''(t_n, S_f) + \mathcal{O}(h^2)$, $\tilde{V}_{S_f}^{'''n,(1)} = V'''(t_n, S_f) + \mathcal{O}(h)$.

4: **Phase 3** ($\ell = 2$):

Compute the FD corrections $\mathbf{a}_1$, $\mathbf{a}_2$ using $\tilde{S}_f^{n,(1)}$ and $V_{S_f}^{*\,''} - \tilde{V}_{S_f}^{''n,(1)}$, $V_{S_f}^{*\,'''} - \tilde{V}_{S_f}^{'''n,(1)}$.

Solve (2.55) to obtain an approximate solution $\tilde{\mathbf{V}}^{n,(2)}$ of $\mathcal{O}(h^4)$ using penalty iteration with initial guess $\tilde{\mathbf{V}}^{n,(1)}$.

Compute the approximate free boundary $\tilde{S}_f^{n,(2)}$ to $\mathcal{O}(h^4)$ accuracy, using Newton's method with initial guess $\tilde{S}_f^{n,(1)}$.

Approximate $V''(t_n, S_f)$, $V'''(t_n, S_f)$, $V''''(t_n, S_f)$ at $\tilde{S}_f^{n,(2)}$ to obtain $\tilde{V}_{S_f}^{''n,(2)} = V''(t_n, S_f) + \mathcal{O}(h^3)$, $\tilde{V}_{S_f}^{'''n,(2)} = V'''(t_n, S_f) + \mathcal{O}(h^2)$, $\tilde{V}_{S_f}^{''''n,(2)} = V''''(t_n, S_f) + \mathcal{O}(h)$.

5: **Phase 4** ($\ell = 3$):

Compute the FD corrections $\mathbf{a}_1$, $\mathbf{a}_2$ using $\tilde{S}_f^{n,(2)}$ and $V_{S_f}^{*\,''} - \tilde{V}_{S_f}^{''n,(2)}$, $V_{S_f}^{*\,'''} - \tilde{V}_{S_f}^{'''n,(2)}$, $V_{S_f}^{*\,''''} - \tilde{V}_{S_f}^{''''n,(2)}$.

Solve (2.55) to obtain an approximate solution $\tilde{\mathbf{V}}^{n,(3)}$ using penalty iteration with initial guess $\tilde{\mathbf{V}}^{n,(2)}$.

Compute the approximate free boundary $\tilde{S}_f^{n,(3)}$, using Newton's method with initial guess $\tilde{S}_f^{n,(2)}$.

6: **end for**

Chapter 3

High-order time stepping for parabolic PDEs and European options

When solving European options, the main difficulty in obtaining high-order methods comes from the fact that the payoff functions in financial contracts are often nonsmooth. Such payoffs can cause a degenerated accuracy of numerical schemes as well as spurious oscillations in the approximate solutions and/or derivatives [54, 25]. While second-order methods for such problems have been extensively used and studied, high-order methods in both the time and space domains have received less attention in the literature. In this chapter, we analyze the effect of nonsmooth initial conditions on the solution of parabolic PDEs using Fourier analysis, and show that the adverse behaviors caused by nonsmooth initial conditions can be addressed by a backward differentiation formula (BDF) time-stepping combining with some smoothing of the initial conditions. We then propose several high-order methods in time and space that do not exhibit degenerated accuracy and spurious oscillations.

In the following, we give a background of existing methods for solving parabolic PDEs with nonsmooth initial conditions. In [43], the authors investigated the convergence rate behavior of PDE methods for pricing problems with nonsmooth payoffs, and proposed various smoothing procedures (averaging the initial data, shifting the grid and a projection method) combined with a special time-stepping method suggested by Rannacher [44] to restore the expected quadratic convergence. In [43], it is shown that both the Rannacher startup procedure preceding the Crank-Nicolson method and a smoothing of the initial data are necessary in order to obtain second-order convergence. In fact, with Crank-Nicolson time stepping (and the diagonal Padé schemes in general), the nonsmoothness in the initial condition causes two sources of errors: the low-order error in the high-frequency Fourier domain, and the quantization error due to the placement of the nonsmooth point on the numerical grid. The convergence behavior of Crank-Nicolson and Rannacher time-marching methods is studied in detail in [25], where the authors applied Fourier analysis to show that several implicit backward Euler steps preceding Crank-Nicolson time stepping, with suitable grid alignment of the nonsmooth point, can act as a damping device and restore the global second-order convergence. To understand the quantization error, Christara and Leung [7] analyzed the effect of the placement of the nonsmooth point relative to the grid, and of various types of smoothings of the initial conditions on the accuracy and stability of second-order numerical methods used for

solving a model convection-diffusion equation.

Higher order methods in both time and space for solving parabolic problems with nonsmooth initial data are not well investigated in the literature. Most existing studies apply grid stretching schemes with high-order discretizations to obtain high-accuracy solutions. With either grid stretching or locally refined meshes, the grid sizes around the singularity are much smaller than on the smooth region. This provides a heuristic for improving the solution accuracy around the singularity. In [40], the authors apply a standard fourth-order FD in space with a smoothly stretched grid around the strike, and BDF4 in time. To initialize BDF4, they employ the combination of two Crank-Nicolson steps and one BDF3 steps. With an appropriately chosen grid stretching parameter, their numerical results empirically demonstrate fourth-order convergence in the option prices of a European vanilla call, while the convergence orders of the calculated Δ and Γ are degenerated and inconsistent. Furthermore, no theoretical guarantees of convergence and stability are provided. Indeed, the authors in [53] observe that only third-order convergence is obtained with the reference method in [40] when initializing BDF4 with two Crank-Nicolson and one BDF3 step, on a uniform space grid discretized with standard fourth-order FD, and fourth-order convergence can be restored only when initializing BDF4 with the exact solutions, which is consistent with our convergence analysis in this chapter. To avoid the wide stencils of standard high-order FD methods, methods using high-order compact (HOC) schemes, usually on uniform grids, are also commonly applied, see, e.g. [53, 13, 14, 15]. In [13, 14, 15], the authors construct HOC schemes on a uniform grid to price more complicated models with stochastic volatility and jumps in multiple dimensions. To match the fourth-order accuracy in the space discretization, a fourth-order smoothing operator [31] is applied to the nonsmooth payoff functions. Compared to the standard FD methods, the construction of the HOC coefficients can be restrictive and quite tedious. Moreover, these methods are typically only second-order accurate in time. To obtain highly accurate time-stepping schemes, the authors in [12] apply an exact in time exponential time integration method, combined with a high-order FD scheme on a locally refined mesh in space, though it is relatively inefficient to approximate the matrix exponential and vector product. Other lines of work based on the weighted essentially non-oscillatory (WENO) discretization schemes are also proposed to solve option pricing problems with nonsmoothness in the solutions or terminal conditions [52, 39]. These schemes are known to be of a high accuracy in smooth solution regimes, while in regions with discontinuities or large gradients, there is an automatic switch to a one-sided high-order reconstruction, which prevents the creation of spurious oscillations.

In this chapter, we propose a simple-to-implement fourth-order method to solve parabolic PDEs with nonsmooth initial conditions. Our method applies BDF4 time stepping initialized with two steps of an explicit third-order Runge-Kutta (RK3) and one step of BDF3 schemes (we can also initialize with three steps of RK3 method). We prove that RK3 generates low-order errors for nonsmooth data in the high-frequency domain that can be damped away by BDF4, while low-order errors in the low-frequency domain are due to the propagation of low-order quantization errors. To deal with the low-order quantization errors of discretizing nonsmooth initial conditions, we derive explicit formulas for fourth-order smoothing of the Dirac delta, Heaviside and ramp initial conditions, from the smoothing operators suggested in [31], and use these to eliminate the low-order errors of the initial condition discretization in the Fourier domain. Given a high-order

initial condition discretization, the time-stepping scheme combining RK3 in the first two time steps, BDF3 in the third time step, and BDF4 onwards is guaranteed to be globally fourth-order in time. Our analysis can be easily generalized to even higher-order time-stepping schemes in the BDF and Runge-Kutta families of methods.

This chapter is organized as follows: In Section 3.1, we describe the model convection-diffusion equation and the various nonsmooth initial conditions that our convergence analysis is based on. In Section 3.2, we introduce the high-order discretization schemes we use. In Section 3.3, we write the error of BDF4 in the Fourier domain as the sum of two terms, namely, the high- and low-frequency components, and study their convergence. In Section 3.4, we analyze the error of RK3 in the Fourier domain, and show that it has a nonconvergent high-frequency component, which, when RK3 is followed by BDF4 (or any other BDF method), is damped exponentially. In Section 3.5, we derive explicit expressions for the smoothed discretization of the initial conditions. In Section 3.6, we bring back all errors to the time domain and demonstrate fourth-order convergence of our method.

3.1 Preliminaries

We are interested in the convection-diffusion equation

$$\frac{\partial v}{\partial t} = \epsilon \frac{\partial^2 v}{\partial x^2} - a \frac{\partial v}{\partial x} \tag{3.1}$$

over $x_L \le x \le x_R$ and $0 < t < T$. To simplify convergence study, we employ the standard practice and consider PDE (3.1) on the infinite space domain $-\infty < x < \infty$ for von Neumann stability analysis. Our numerical results show that the conclusions obtained for (3.1) on the infinite space domain generalize well to problems defined on finite domains.

To investigate the behavior of (3.1) with nonsmooth initial conditions, we consider three types of initial conditions

$$v(0,x) = \delta(x), \tag{3.2}$$

$$v(0,x) = H(x) \equiv \begin{cases} 1, & x \ge 0, \\ 0, & x < 0, \end{cases} \tag{3.3}$$

$$v(0,x) = C(x) \equiv \max(x,0) = xH(x), \tag{3.4}$$

which correspond to the Dirac delta, Heaviside and ramp functions, respectively. The singularities of Heaviside and ramp functions form the basis of many other nonsmooth functions. For example, the bump function can be constructed from a linear combination of ramp functions, as

$$v(0,x) = \max(x - \mathcal{B}, 0) - 2\max(x,0) + \max(x + \mathcal{B}, 0), \tag{3.5}$$

where $\mathcal{B} > 0$ is a constant. We see that the payoff of a digital call option is a shifted Heaviside function, and the payoff of a call option is a shifted ramp function. Using linear combinations of the shifted Heaviside and ramp functions, we can get many other types of payoffs, such as the

bull/bear spread as in (1.15), (1.16), and butterfly spread as in (1.17).

The exact solutions corresponding to the three initial conditions (3.2), (3.3) and (3.4) are, respectively,

$$v_\delta(t,x) \equiv \frac{1}{\sqrt{4\pi\epsilon t}} \exp\left(-\frac{(x-at)^2}{4\epsilon t}\right) = \frac{(\sqrt{2}\zeta)^{-1}}{2}\ \mathrm{i}^{-1}\operatorname{erfc}\left(-\frac{x-at}{\sqrt{2}\zeta}\right), \tag{3.6}$$

$$v_H(t,x) \equiv \int_{-\infty}^{x} \frac{1}{\sqrt{4\pi\epsilon t}} \exp\left(-\frac{(y-at)^2}{4\epsilon t}\right) dy = \frac{(\sqrt{2}\zeta)^{0}}{2}\ \mathrm{i}^{0}\operatorname{erfc}\left(-\frac{x-at}{\sqrt{2}\zeta}\right), \tag{3.7}$$

$$v_C(t,x) \equiv \int_{-\infty}^{x}\int_{-\infty}^{z} \frac{1}{\sqrt{4\pi\epsilon t}} \exp\left(-\frac{(y-at)^2}{4\epsilon t}\right) dydz = \frac{(\sqrt{2}\zeta)^{1}}{2}\ \mathrm{i}^{1}\operatorname{erfc}\left(-\frac{x-at}{\sqrt{2}\zeta}\right), \tag{3.8}$$

where $\zeta = \sqrt{2\epsilon t}$, $\mathrm{i}^{-1}\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}}e^{-x^2}$, $\mathrm{i}^{0}\operatorname{erfc}(x) = \operatorname{erfc}(x)$, $\mathrm{i}^{1}\operatorname{erfc}(x) = \int_x^\infty \operatorname{erfc}(z)dz$, and $i = \sqrt{-1}$. The exact solution of (3.1) with the bump initial condition (3.5) can be calculated from the same linear combination of the exact solutions of (3.1) with the corresponding ramp functions as the initial conditions. The first and second derivatives of (3.6) are

$$\begin{aligned}
\frac{\partial v_\delta}{\partial x} &= -\frac{x-at}{(\sqrt{2}\zeta)^3}\ \mathrm{i}^{-1}\operatorname{erfc}\left(-\frac{x-at}{\sqrt{2}\zeta}\right), \\
\frac{\partial^2 v_\delta}{\partial x^2} &= \left(\frac{2(x-at)^2}{(\sqrt{2}\zeta)^5} - \frac{1}{(\sqrt{2}\zeta)^3}\right)\ \mathrm{i}^{-1}\operatorname{erfc}\left(-\frac{x-at}{\sqrt{2}\zeta}\right),
\end{aligned} \tag{3.9}$$

and the first and second derivatives of (3.7) and (3.8) are, respectively,

$$\frac{\partial v_H}{\partial x} = v_\delta, \quad \frac{\partial^2 v_H}{\partial x^2} = \frac{\partial v_\delta}{\partial x}, \tag{3.10}$$

$$\frac{\partial v_C}{\partial x} = v_H, \quad \frac{\partial^2 v_C}{\partial x^2} = \frac{\partial v_H}{\partial x}. \tag{3.11}$$

We take $\epsilon = 1$ in the following. We are interested in approximating the solution and its derivatives to a high-order accuracy.

We define the Fourier transform pair of a generic function $v(t,x)$ as

$$\hat{v}(t,\omega) \equiv \int_{-\infty}^{\infty} v(t,x)e^{-i\omega x}dx, \quad v(t,x) \equiv \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{v}(t,\omega)e^{i\omega x}d\omega.$$

The Fourier transformed model problem (3.1) in the frequency domain becomes $\hat{v}_t = -(\omega^2 + ia\omega)\hat{v}$, and has the exact solution

$$\hat{v}(t,\omega) = e^{-(\omega^2+ia\omega)t}\hat{v}(0,\omega), \tag{3.12}$$

where $\hat{v}(0,\omega)$ is the Fourier transform of any of the initial conditions defined in (3.2) to (3.4). When it is clear from context, we drop ω in the frequency notation and simply write $\hat{v}(t)$ for convenience.

When discretizing the space domain to a grid $\{x_j\}$, for $j = \ldots, -1, 0, 1, \ldots$, the nonsmooth point does not necessarily lie exactly on a grid point. To accommodate this, we introduce a parameter $\alpha \in (0,1]$ and denote $x_j = (j + (1-\alpha))h$ as the grid points, where h is the uniform spatial stepsize. The nonsmooth point is fixed at $x = 0$. For general α, the delta initial condition is typically

discretized as [25, 7]

$$\delta_\alpha(x_j) \equiv \begin{cases} \frac{1-\alpha}{h}, & j = -1, \\ \frac{\alpha}{h}, & j = 0, \\ 0, & \text{else}, \end{cases} \tag{3.13}$$

which is equivalent to second order smoothing in [31]. The discretization of (3.3), (3.4) and (3.5) can be simply sampled from the continuous respective function. However, it turns out that the naive discretization of initial conditions may lead to deterioration of the convergence rate of a high-order method, due to their low-order representation in the frequency domain. Moreover, the alignment of the nonsmooth point on the grid also plays a role in the convergence order. To deal with nonsmooth initial condition discretization, we apply a high-order convolution-type smoothing; see Section 3.5. In this chapter, we focus on high-order time stepping analysis. A more detailed study of nonsmooth initial conditions is presented in Chapter 4.

We employ the Fourier transform pair on the discrete space domain $\{x_j\}$ for $-\infty < j < \infty$ to study the convergence behavior of the discretization. Again, numerical results in Chapter 5 show that the conclusions from Fourier analysis of the model PDE (3.1) generalize well to problems defined on finite domains. With the alignment $\alpha = 1$ and with $\theta \equiv \omega h$, the semi-discrete Fourier transform pair [56] is

$$\hat{V}(\omega h) = \hat{V}(\theta) = h \sum_{j=-\infty}^{\infty} V_j e^{-i\omega x_j} = h \sum_{j=-\infty}^{\infty} V_j e^{-ij\theta}, \tag{3.14}$$

$$V_j = \frac{1}{2\pi} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{V}(\omega h) e^{i\omega x_j} d\omega = \frac{1}{2\pi h} \int_{-\pi}^{\pi} \hat{V}(\theta) e^{ij\theta} d\theta, \tag{3.15}$$

Note that direct application of the semi-discrete Fourier transform is only valid for the case of Dirac delta initial condition, because the summation in (3.14) of the semi-discrete Fourier transform of the solutions V_H and V_C corresponding to the Heaviside and the ramp initial conditions, respectively, is divergent. For this reason, the Fourier transform pair for the Heaviside and ramp initial conditions we use here is

$$\hat{V}(\omega h) = \hat{V}(\theta) = h \sum_{j=-\infty}^{\infty} (e^{-\eta j h} V_j) e^{-i\omega x_j} = h \sum_{j=-\infty}^{\infty} (e^{-\eta j h} V_j) e^{-ij\theta},$$

$$e^{-\eta j h} V_j = \frac{1}{2\pi} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{V}(\omega h) e^{i\omega x_j} d\omega = \frac{1}{2\pi h} \int_{-\pi}^{\pi} \hat{V}(\theta) e^{ij\theta} d\theta,$$

which requires to multiply both sides of (3.16) by $e^{-\eta j h}$, for $\eta > 0$, before the semi-discrete inverse Fourier transform can be applied. As a result, $\hat{V}$ is not the direct Fourier transform of the solutions in the cases of the Heaviside and the ramp initial conditions. However, for notational simplicity, we do not explicitly differentiate the two situations. Readers should understand the meaning of $\hat{V}$ from its context.

3.2 Discretization

In practice, when numerically solving for the solution, we need to work on a finite space domain, either because the original problem is already defined on a finite space, or because we must truncate an infinite domain for computation. Consider a discretized domain $x_0 < x_1 < \cdots < x_M$ where x_0 and x_M represent the left and right boundary, respectively. Note that we are using different grid indices here for discretization on a finite domain, as compared to the notations in the previous section where the indices j of x_j goes from $-\infty$ to ∞ for Fourier analysis. Let $V_j^n \approx v(t_n, x_j)$ be the FD approximation to the true solution $v(t_n, x_j)$, where $t_n = nk$ is the n-th time step, and $k = \frac{T}{N}$ is the time step size with a total of N time steps. We drop the superscript n when time is irrelevant. On a uniform grid with stepsize h, the fourth-order FD approximation to $\frac{\partial^2 V}{\partial S^2}(t, x_j)$ is given by the operator

$$D_4^2 V_j \equiv \frac{1}{12h^2}(-V_{j-2} + 16V_{j-1} - 30V_j + 16V_{j+1} - V_{j+2}),$$

for $2 \le j \le M-2$. At the boundaries, we only need to apply a second-order discretization to maintain fourth-order accuracy [4],

$$D_4^2 V_1 \equiv \frac{1}{h^2}(V_0 - 2V_1 + V_2),$$

for $j = 1$, and similarly for $j = M-1$. The fourth-order FD approximation to $\frac{\partial V}{\partial S}(t, x_j)$ is given by the operator

$$D_4 V_j \equiv \frac{1}{12h}(V_{j-2} - 8V_{j-1} + 8V_{j+1} - V_{j+2}),$$

for $2 \le j \le M-2$, and

$$D_4 V_1 \equiv \frac{1}{h}(-V_0 + V_2).$$

for $j = 1$, and similarly for $j = M-1$. For convenience of later discussion, we denote $\mathcal{D}_h$ to be $\mathcal{D}_h \equiv D_4^2 - aD_4$, so that

$$\mathcal{D}_h V_j = \frac{-V_{j-2} + 16V_{j-1} - 30V_j + 16V_{j+1} - V_{j+2}}{12h^2} - a\frac{V_{j-2} - 8V_{j-1} + 8V_{j+1} - V_{j+2}}{12h}.$$

for $2 \le j \le M-2$, while slightly different relations hold for $j = 1$ and $j = M-1$. Hence, with the space discretization of (3.1), we obtain an ordinary differential equation (ODE) system

$$\frac{dV_j}{dt} = \mathcal{D}_h V_j, \quad 1 \le j \le M-1.$$

When using BDF4 time-stepping, with time step size k, we have, for the $(n+4)$-th time step,

$$\frac{\frac{25}{12}V_j^{(n+4)} - 4V_j^{(n+3)} + 3V_j^{(n+2)} - \frac{4}{3}V_j^{(n+1)} + \frac{1}{4}V_j^{(n)}}{k} = \mathcal{D}_h V_j^{(n+4)}, \tag{3.16}$$

for $2 \leq j \leq M-2$. For later convenience, we define

$$\alpha_4 = \frac{25}{12},\ \alpha_3 \ = -4,\ \alpha_2 = 3,\ \alpha_1 = -\frac{4}{3},\ \alpha_0 = \frac{1}{4}. \tag{3.17}$$

Hence, (3.16) becomes

$$\sum_{l=0}^{4} \alpha_l V_j^{(n+l)} = k\mathcal{D}_h V_j^{(n+4)}. \tag{3.18}$$

3.3 Fourier analysis of the discrete system arising from BDF4

In this section, we investigate the Fourier transform of the discrete system (3.16). Applying Fourier transform (3.15) to (3.18), we get

$$\sum_{l=0}^{4} \alpha_l \hat{V}^{(n+l)}(\theta) = \mu \hat{V}^{(n+4)}(\theta), \tag{3.19}$$

where $\alpha_4, \alpha_3, \alpha_2, \alpha_1, \alpha_0$ are the BDF4 coefficients as defined in (3.17), and

$$\begin{aligned} \mu &= -\frac{\bar{d}}{12}\left(2\cos(2\theta) - 32\cos\theta + 30\right)) - i\frac{ad}{12}\left(16\sin(\theta) - 2\sin(2\theta)\right) \\ &= -\frac{\bar{d}}{3}(1-\cos\theta)(7-\cos\theta) - i\frac{ad}{3}\sin(\theta)(4-\cos\theta), \end{aligned} \tag{3.20}$$

where $d = \frac{k}{h}$ and $\bar{d} = \frac{k}{h^2}$. Note that a is fixed for a given problem, and $k = dh$ for some constant d. By re-arranging the terms, we can write (3.19) as

$$\sum_{l=0}^{4} \hat{\alpha}_l \hat{V}^{(n+l)}(\theta) = 0, \tag{3.21}$$

with the coefficients

$$\begin{aligned} &\hat{\alpha}_4 = \frac{25}{12} + \frac{\bar{d}}{3}(1-\cos\theta)(7-\cos\theta) + i\frac{ad}{3}\sin(\theta)(4-\cos\theta) = \alpha_4 - \mu, \\ &\hat{\alpha}_3 = \alpha_3 = -4,\ \hat{\alpha}_2 = \alpha_2 = 3,\ \hat{\alpha}_1 = \alpha_1 = -\frac{4}{3},\ \hat{\alpha}_0 = \alpha_0 = \frac{1}{4}. \end{aligned} \tag{3.22}$$

The corresponding characteristic polynomial of the difference equation (3.21) is

$$\sum_{l=0}^{4} \hat{\alpha}_l \xi^l = \rho(\xi) - \mu\xi^4,\ \text{where } \rho(\xi) = \sum_{l=0}^{4} \alpha_l \xi^l. \tag{3.23}$$

Considering the recurrence relation

$$\hat{V}^{(n+4)} = -\frac{\alpha_3 \hat{V}^{(n+3)} + \alpha_2 \hat{V}^{(n+2)} + \alpha_1 \hat{V}^{(n+1)} + \alpha_0 \hat{V}^{(n)}}{\hat{\alpha}_4}, \tag{3.24}$$

we find the generic expression of $\hat{V}^{(n+4)}$ given the four starting values $\hat{V}^{(n+l)}$ for $l = 0, 1, 2, 3$. To study the convergence behavior, we write the BDF4 iteration as a one-step method by

$$\bar{V}^{(n+1)} \equiv \begin{bmatrix} \hat{V}^{(n+4)} \\ \hat{V}^{(n+3)} \\ \hat{V}^{(n+2)} \\ \hat{V}^{(n+1)} \end{bmatrix} = \begin{bmatrix} -\frac{\alpha_3}{\hat{\alpha}_4} & -\frac{\alpha_2}{\hat{\alpha}_4} & -\frac{\alpha_1}{\hat{\alpha}_4} & -\frac{\alpha_0}{\hat{\alpha}_4} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{V}^{(n+3)} \\ \hat{V}^{(n+2)} \\ \hat{V}^{(n+1)} \\ \hat{V}^{(n)} \end{bmatrix} \equiv R\bar{V}^{(n)}, \tag{3.25}$$

where $R = R(\mu)$ is a function of μ denoting the iteration matrix. The spectral radius of R indicates the convergence behavior of the iteration. Note that there is one-to-one correspondence between the roots of the characteristic polynomial $\rho(\xi) - \mu\xi^4$ and the eigenvalues of the companion matrix R.

Let $\hat{E}^{(n)} \equiv \hat{v}(t_n) - \hat{V}^{(n)}$ for $l \geq 0$, and the truncation error

$$\varepsilon^{(n)} \equiv \frac{1}{\hat{\alpha}_4} \sum_{l=0}^{4} \hat{\alpha}_l \hat{v}(t_{n+l}). \tag{3.26}$$

Define $\bar{E}^{(n)} \equiv [\hat{E}^{(n+3)}, \hat{E}^{(n+2)}, \hat{E}^{(n+1)}, \hat{E}^{(n)}]^T$, and $\bar{\varepsilon}^{(n)} \equiv [\varepsilon^{(n)}, 0, 0, 0]^T$. Then, we can see from the iterative scheme (3.25) that

$$\bar{E}^{(n+1)} = R\bar{E}^{(n)} + \bar{\varepsilon}^{(n)},$$

and therefore,

$$\bar{E}^{(n+1)} = R^{n+1}\bar{E}^{(0)} + \sum_{l=0}^{n} R^l \bar{\varepsilon}^{(n-l)},$$

given an initial approximate $\bar{V}^{(0)}$ and the corresponding $\bar{E}^{(0)}$.

Note that $\hat{V}, \hat{E}, \bar{E}, \bar{V}$ and R are (vector-)functions of ω and h. For the convenience of later discussion, for any fixed $h \in (0, 1)$, when $\omega \neq 0$, we define

$$\beta \equiv \frac{\log|\omega|}{\log(1/h)}, \tag{3.27}$$

so that ω and h are related by $|\omega| = h^{-\beta}$. Since $\omega \in [-\pi/h, \pi/h]$, we get that $\beta \leq 1 + \frac{\log \pi}{\log(1/h)} \equiv \beta_{\max}$. The exact solution $\hat{v}(t_N)$ at $t_N \equiv T = Nk$ is

$$\hat{v}(t_N) = e^{-(\omega^2 + ia\omega)Nk}\hat{v}(0) = e^{-(\omega^2 + ia\omega)T}\hat{v}(0).$$

For later discussion, we define $z \equiv (\omega^2 + ia\omega)k$. We see that, as $h \to 0$, we have $\hat{v}(t_N) \to e^{-\infty} = 0$ exponentially in the frequency range $|\omega| = h^{-\beta}$ with $\beta > 0$. Moreover, the exact solution for all t_n decays exponentially to zero when $\beta > \frac{1}{2}$. The goal is to study the stability and convergence of the BDF4 solution by investigating the behavior of $\hat{V}^{(N)}$ obtained from the recurrence relation (3.24). In the following discussion, we consider the frequencies $|\omega| = h^{-\beta}$ with $\beta < \frac{1}{2}$, together with the frequency $\omega = 0$, as being in the low-frequency regime, and the frequencies $|\omega| = h^{-\beta}$ with

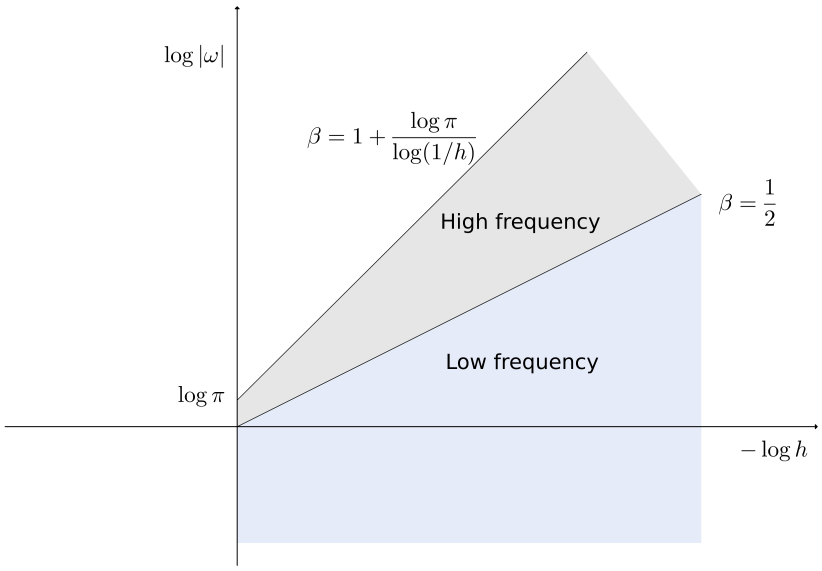


Figure 3.1: High- and low-frequency regions arising in BDF4. Note that $|\omega| = h^{-\beta}$.

$\frac{1}{2} \le \beta \le \beta_{\max}$ as being in the high-frequency regime, as shown in Figure 3.1. We show later that the convergence performance of the approximate $\hat{V}^{(n)}$ behaves differently in the high and low-frequency domain.

We use the following lemmas to prove our main theorem.

Lemma 3.3.1. *Let β be defined by (3.27), and μ be given by (3.20). As $h \to 0$, we have $|\mu| \to \infty$ when $\frac{1}{2} < \beta \le \beta_{\max}$, and $|\mu| \to 0$ when $\beta < \frac{1}{2}$.*

Proof. When $1 < \beta \le \beta_{\max}$, we have $\omega h \to \pm\infty$ as $h \to 0$. Hence, the real part $\mathrm{Re}(\mu) = -\frac{d}{3h}(1-\cos(\theta))(7-\cos(\theta)) \to -\infty$, and the imaginary part $|\mathrm{Im}(\mu)| = \left|-\frac{ad}{3}\sin(\theta)(4-\cos\theta)\right|$ is bounded above. It is obvious then that $|\mu| \to \infty$. Similarly for $\beta = 1$.

When $\beta < 1$, we have $\omega h \to 0$. In this case, $\mathrm{Im}(\mu) \to 0$. Moreover,

$$\begin{aligned}
\lim_{h\to 0} \mathrm{Re}(\mu) &= \lim_{h\to 0}\left[-\frac{d}{3h}(1-\cos(\theta))(7-\cos(\theta))\right] \\
&= -\frac{d}{3}\lim_{h\to 0}\frac{(1-\cos h^{1-\beta})(7-\cos h^{1-\beta})}{h} \\
&= -\frac{d}{3}\lim_{h\to 0}\frac{(1-\beta)h^{-\beta}\sin h^{1-\beta}(8-2\cos h^{1-\beta})}{1} \\
&= -2(1-\beta)d\lim_{h\to 0}\frac{\sin h^{1-\beta}}{h^{\beta}} \\
&= -2(1-\beta)d\lim_{h\to 0} h^{1-2\beta}.
\end{aligned}$$

We see that if $\frac{1}{2} < \beta < 1$, we have $\lim_{h\to 0}\mathrm{Re}(\mu) = -\infty$. Hence, $\mu \to -\infty$ in the complex plane. If $\beta < \frac{1}{2}$, we have $\lim_{h\to 0}\mathrm{Re}(\mu) = 0$. Hence, we get $\mu \to 0$. □

Lemma 3.3.2 ([32])**.** *Suppose that a linear multistep method (ρ, σ) is strongly $A(\theta)$-stable $(0 \le \theta \le \pi/2)$. Then there exist positive constants r, γ, C, such that $\forall \mu \in S_\theta \equiv \{z \in \bar{\mathbb{C}} | z = 0$ or $z =$*

∞ *or* $-\theta_1 \le Arg\, z \le \theta_1, 0 \le \theta_1 < \theta\}$, *we have*

$$\|R(\mu)^n\| \le Ce^{-\gamma n}, \quad \text{if } |\mu| \ge r;$$
$$\|R(\mu)^n\| \le Ce^{\gamma n Re(\mu)}, \quad \text{if } |\mu| \le r.$$

Proof. See Lemma 3 in [32]. □

Combining Lemmas 3.3.1 and 3.3.2, we see that when $\frac{1}{2} < \beta \le \beta_{\max}$, we have $\|R(\mu)^n\| \le Ce^{-\gamma n}$; and when $\beta < \frac{1}{2}$, we have $\|R(\mu)^n\| \le Ce^{\gamma n \mathrm{Re}(\mu)}$. For the case when $\beta = \frac{1}{2}$, we can see from the proof of Lemma 3.3.1 that $\mathrm{Re}(\mu) \to -d$ and $|\mu| \to d$ as $h \to 0$. Thus, if $|\mu| \ge r$, we have $\|R(\mu)^n\| \le Ce^{-\gamma n}$, and, if $|\mu| \le r$, we have $||R(\mu)^n|| \le Ce^{-\gamma n d}$, which decays exponentially as well.

Lemma 3.3.3. *Let* $\varepsilon^{(n)}$ *be defined by (3.26),* β *as in (3.27), and* μ *as in (3.20). We have*

$$|\varepsilon^{(n)}| \le \frac{12}{25}\left(\frac{20}{3}|z|^5 + |\mu + z||e^{-4z}|\right)|\hat{v}(t_n)|.$$

Proof. For notation convenience, we look at $\varepsilon^{(n-4)}$. We first note that

$$\begin{aligned}\hat{\alpha}_4\varepsilon^{(n-4)} &= \sum_{l=0}^{4}\hat{\alpha}_{4-l}\hat{v}(t_{n-l}) \\ &= \sum_{l=0}^{4}\alpha_{4-l}\hat{v}(t_{n-l}) + z\hat{v}(t_n) - (\mu+z)\hat{v}(t_n) \\ &= \sum_{l=0}^{4}\alpha_{4-l}\hat{v}(t_{n-l}) - k\hat{v}_t(t_n) - (\mu+z)\hat{v}(t_n).\end{aligned}$$

Applying Taylor expansion to $\hat{v}(t_{n-l})$ and get

$$\hat{v}(t_{n-l}) = \hat{v}(t_n) - lk\hat{v}_t(t_n) + \frac{l^2k^2}{2}\hat{v}_{tt}(t_n) - \frac{l^3k^3}{6}\hat{v}_{ttt}(t_n) + \frac{l^4k^4}{24}\hat{v}_{tttt}(t_n) + \int_{t_n}^{t_{n-l}}\frac{(t_{n-l}-t)^4}{24}\hat{v}_{ttttt}(t)dt.$$

From the properties of BDF4 coefficients and the fact that $\hat{v}(t) = e^{-(\omega^2+ia\omega)t}\hat{v}(0)$ is infinitely smooth, we have

$$\sum_{l=0}^{4}\alpha_{4-l}\hat{v}(t_{n-l}) - k\hat{v}_t(t_n) = \frac{-(\omega^2+ia\omega)^5}{24}\sum_{l=1}^{4}\alpha_{4-l}\int_{t_n}^{t_{n-l}}(t_{n-l}-t)^4e^{-(\omega^2+ia\omega)t}\hat{v}(0)dt,$$

and

$$
\begin{aligned}
|\varepsilon^{(n-4)}| &= \left| \frac{1}{\alpha_4-\mu} \left(\frac{-(\omega^2+ia\omega)^5}{24} \sum_{l=1}^{4} \alpha_{4-l} \int_{t_n}^{t_{n-l}} (t_{n-l}-t)^4 e^{-(\omega^2+ia\omega)t} \hat{v}(0)dt - (\mu+z)\hat{v}(t_n) \right) \right| \\
&\le \frac{1}{\alpha_4} \left(\frac{|\omega^2+ia\omega|^5}{24} \left| \sum_{l=1}^{4} \alpha_{4-l} \int_{t_{n-l}}^{t_n} (t-t_{n-l})^4 e^{-(\omega^2+ia\omega)t} \hat{v}(0)dt \right| + |\mu+z||\hat{v}(t_n)| \right) \\
&\le \frac{1}{\alpha_4} \left(\frac{|\omega^2+ia\omega|^5}{24} \sum_{l=2,4} \alpha_{4-l} \left| \int_{t_{n-l}}^{t_n} (t-t_{n-l})^4 e^{-(\omega^2+ia\omega)t} \hat{v}(0)dt \right| + |\mu+z||\hat{v}(t_n)| \right) \\
&\le \frac{1}{\alpha_4} \left(\frac{|\omega^2+ia\omega|^5 k^4}{24} \sum_{l=2,4} \alpha_{4-l} l^4 \left| \int_{t_{n-l}}^{t_n} e^{-(\omega^2+ia\omega)t} \hat{v}(0)dt \right| + |\mu+z||\hat{v}(t_n)| \right) \\
&\le \frac{1}{\alpha_4} \left(\frac{|\omega^2+ia\omega|^5 k^4}{24} \sum_{l=2,4} \alpha_{4-l} l^4 \cdot lk\, |\hat{v}(t_{n-l})| + |\mu+z||e^{-(\omega^2+ia\omega)4k} \hat{v}(t_{n-4})| \right) \\
&\le \frac{1}{\alpha_4} \left(\frac{|\omega^2+ia\omega|^5 k^5}{24} \sum_{l=2,4} \alpha_{4-l} l^5 + |\mu+z||e^{-(\omega^2+ia\omega)4k}| \right) |\hat{v}(t_{n-4})| \\
&= \frac{12}{25} \left(\frac{20}{3}|z|^5 + |\mu+z||e^{-(\omega^2+ia\omega)4k}| \right) |\hat{v}(t_{n-4})|.
\end{aligned}
$$

□

Theorem 3.3.4. *For the iteration scheme (3.19), there exist some positive constants γ, C_1, C_2, C_3 such that*

$$
\begin{aligned}
|\hat{E}^{(n)}| \le\ & C_1 e^{-\gamma n} \max_{0\le l\le 3} |\hat{V}^{(l)}| \mathbb{1}_{\{\frac{1}{2}\le\beta\le\beta_{\max}\}} \\
& + \left(C_2 e^{\gamma n Re(\mu)} \max_{0\le l\le 3} |\hat{E}^{(l)}| + C_3 h^4 |\omega|^{\chi_5} |\nu|^{\bar{\gamma}(n-1)} |\hat{v}(0)| \right) \mathbb{1}_{\{\beta<\frac{1}{2},\omega=0\}},
\end{aligned}
\tag{3.28}
$$

for $n \ge 4$, where $\nu = e^{-(\omega^2+ia\omega)k}$, $\bar{\gamma} = \min(\gamma, 1)$, and $\chi_5 \equiv 5(1+H(\beta))$.

Proof. When $\frac{1}{2} \le \beta \le \beta_{\max}$, the exact solution $\hat{v}(t_n) = e^{-nz}\hat{v}(0)$ converges to 0 exponentially. Hence, from Lemma 3.3.2, we have

$$
|\hat{E}^{(n)}| \le \|\bar{E}^{(n)}\| \approx \|R^n \bar{V}^{(0)}\| \le \|R^n\| \cdot \|\bar{V}^{(0)}\| \le C_1 e^{-\gamma n} \max_{0\le l\le 3} |\hat{V}^{(l)}|.
$$

When $\beta < \frac{1}{2}$, we note that

$$
\mathrm{Re}(\mu) = -\omega^2 k + \frac{d}{90}\omega^6 h^5 + \mathcal{O}(\omega^8 h^7),
$$

and

$$
\left| \frac{d}{90}\omega^6 h^5 + \mathcal{O}(\omega^8 h^7) \right| = \left| \frac{d}{90} h^{5-6\beta} + \mathcal{O}(h^{7-8\beta}) \right| \le Ch^2
$$

for some positive constant C. Hence, with $0 \le l \le n-1 \le \frac{T}{k}$, we have

$$\begin{aligned} e^{\gamma l \mathrm{Re}(\mu)} &\le e^{-\gamma l \omega^2 k} \cdot e^{\gamma l |\frac{d}{90}\omega^6 h^5 + \mathcal{O}(\omega^8 h^7)|} \\ &\le e^{-\gamma l \omega^2 k} \cdot e^{\gamma l C h^2} \\ &\le e^{-\gamma l \omega^2 k}(1 + C l h^2) \\ &= |\nu|^{\gamma l}(1 + C l h^2). \end{aligned}$$

Here and in the following, the constants C at each step are not necessarily the same. Moreover, recalling that $z = (\omega^2 + ia\omega)k$, we have

$$|z| \le \begin{cases} C\omega^2 h, & \beta \ge 0, \\ C|\omega| h, & \beta < 0, \end{cases} \quad \text{and} \quad |\mu + z| \le \begin{cases} C\omega^6 h^5, & \beta \ge 0, \\ C|\omega|^5 h^5, & \beta < 0. \end{cases}$$

From Lemmas 3.3.2 and 3.3.3, we have

$$\begin{aligned} |\hat{E}^{(n)}| \le \|\bar{E}^{(n)}\| &= \|R^n \bar{E}^{(0)} + \sum_{l=0}^{n-1} R^l \bar{\varepsilon}^{(n-1-l)}\| \\ &\le \|R^n\| \cdot \|\bar{E}^{(0)}\| + \sum_{l=0}^{n-1} \|R^l\| \cdot |\varepsilon^{(n-1-l)}| \\ &\le C e^{\gamma n \mathrm{Re}(\mu)} \|\bar{E}^{(0)}\| + C \frac{12}{25} \left(\frac{20}{3}|z|^5 + |\mu + z||e^{-4z}| \right) |\hat{v}(0)| \sum_{l=0}^{n-1} e^{\gamma l \mathrm{Re}(\mu)} |\nu|^{n-1-l} \\ &\le C e^{\gamma n \mathrm{Re}(\mu)} \|\bar{E}^{(0)}\| + C \left(\frac{20}{3}|z|^5 + |\mu + z| \right) |\hat{v}(0)| \sum_{l=0}^{n-1} |\nu|^{\gamma l + n - 1 - l}(1 + C l h^2) \\ &\le C e^{\gamma n \mathrm{Re}(\mu)} \|\bar{E}^{(0)}\| + C |\omega|^{\chi_5} h^5 |\hat{v}(0)| |\nu|^{\bar{\gamma}(n-1)} \sum_{l=0}^{n-1} (1 + C l h^2) \\ &\le C e^{\gamma n \mathrm{Re}(\mu)} \|\bar{E}^{(0)}\| + C |\omega|^{\chi_5} h^5 |\hat{v}(0)| |\nu|^{\bar{\gamma}(n-1)} n \\ &\le C e^{\gamma n \mathrm{Re}(\mu)} \max_{0 \le l \le 3} |\hat{E}^{(l)}| + C |\omega|^{\chi_5} h^5 |\hat{v}(0)| \frac{|\nu|^{\bar{\gamma}(n-1)}}{h} \\ &= C e^{\gamma n \mathrm{Re}(\mu)} \max_{0 \le l \le 3} |\hat{E}^{(l)}| + C h^4 |\omega|^{\chi_5} |\nu|^{\bar{\gamma}(n-1)} |\hat{v}(0)|. \end{aligned}$$

□

Remark 3.3.1. *Theorem 3.3.4 shows that the high-frequency error of BDF4 decays exponentially, while the low-frequency error involves the error from the three steps of initialization, and a fourth-order component.*

Remark 3.3.2. *For one step iteration, we have* $|\hat{E}^{(n)}| \le \|\bar{E}^{(n-3)}\| = \|R\bar{E}^{(n-4)} + \varepsilon^{(n-4)}\|$. *From the proof of Theorem 3.3.4, we see that, for* $n \ge 4$, *the local error of BDF4 satisfies*

$$|\hat{E}^{(n)}| \le C_2 e^{\gamma Re(\mu)} \max_{1 \le l \le 4} |\hat{E}^{(n-l)}| + C_3 h^5 |\omega|^{\chi_5} |\nu|^{n-4} |\hat{v}(0)| \tag{3.29}$$

in the low-frequency region, which is what we would expect for the local error.

Remark 3.3.3. *We study the convergence behavior of BDF4 time stepping in particular, but it is easy to see that the proof process does not rely on the order of the BDF method, and the conclusions can be similarly extended to other methods in the BDF family. In general, for the p-th order BDF method with $p \leq 6$, we have*

$$|\hat{E}^{(n)}| \leq C_1 e^{-\gamma n} \max_{0\leq l\leq p-1} |\hat{V}^{(l)}| \mathbb{1}_{\{\frac{1}{2}\leq\beta\leq\beta_{\max}\}} + \left(C_2 e^{\gamma n Re(\mu)} \max_{0\leq l\leq p-1} |\hat{E}^{(l)}| + C_3 h^p |\omega|^{\chi_{p+1}} |\nu|^{\bar{\gamma}(n-1)} |\hat{v}(0)| \right) \mathbb{1}_{\{\beta<\frac{1}{2},\omega=0\}},$$

and the low-frequency local error applying one step of the p-th order BDF method satisfies

$$|\hat{E}^{(n)}| \leq C_2 e^{\gamma Re(\mu)} \max_{1\leq l\leq p} |\hat{E}^{(n-l)}| + C_3 h^{p+1} |\omega|^{\chi_{p+1}} |\nu|^{n-p} |\hat{v}(0)|,$$

for $n \geq p$, where $\chi_{p+1} = (p+1)(1+H(\beta))$. For example, the low-frequency error applying BDF3 converges as $\mathcal{O}(h^3|\omega|^{\chi_4}|\nu|^{\bar{\gamma}(n-1)}|\hat{v}(0)|)$ globally, and as $\mathcal{O}(h^4|\omega|^{\chi_4}||\nu|^{n-3}\hat{v}(0)|)$ locally, assuming that BDF3 is initialized with the exact solutions.

3.4 Initializing BDF4

Third order methods are sufficient to initialize the first three time steps in order to obtain global fourth-order convergence. We propose to initialize BDF4 with a classic third-order explicit Runge-Kutta method (RK3) for the first two time steps, and a third order backward differentiation formula (BDF3) for the third time step (denoted as 2RK3-BDF3-BDF4 in the following). In this section, we carry out an analysis of such initialization scheme for BDF4.

BDF3 for solving the third time step follows the update rule

$$\frac{\frac{11}{6}V_j^{(3)} - 3V_j^{(2)} + \frac{3}{2}V_j^{(1)} - \frac{1}{3}V_j^{(0)}}{k} = \mathcal{D}_h V_j^{(3)}. \tag{3.30}$$

We have already studied the convergence behavior of BDF methods in the preceding section. The RK3 method used in the first two steps is given by the Butcher tableau

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & 1/6 & 2/3 & 1/6 \end{array}.$$

We study its convergence in the following subsection.

3.4.1 Fourier analysis of RK3 applied to nonsmooth data

Recall that the semi-discrete ODE system we are solving is $\frac{dV_j}{dt} = \mathcal{D}_h V_j$. Without loss of generality, we consider applying RK3 to the first time step and compute the solution at x_j by computing

$$\begin{aligned} f_1 &= \mathcal{D}_h V_j^{(0)}, \\ f_2 &= \mathcal{D}_h \left(V_j^{(0)} + \frac{k}{2} f_1 \right), \\ f_3 &= \mathcal{D}_h (V_j^{(0)} - k f_1 + 2k f_2), \end{aligned}$$

and

$$V_j^{(1)} = V_j^{(0)} + \frac{k}{6}(f_1 + 4f_2 + f_3) = V_j^{(0)} + k\mathcal{D}_h V_j^{(0)} + \frac{k^2}{2}\mathcal{D}_h^2 V_j^{(0)} + \frac{k^3}{6}\mathcal{D}_h^3 V_j^{(0)}.$$

Defining the operator

$$\mathcal{K}_{k,h} \equiv 1 + k\mathcal{D}_h + \frac{k^2}{2}\mathcal{D}_h^2 + \frac{k^3}{6}\mathcal{D}_h^3,$$

one step of RK3 is simply

$$V_j^{(1)} = \mathcal{K}_{k,h} V_j^{(0)}. \tag{3.31}$$

We see that $\mathcal{K}_{k,h} V_j^{(0)}$ is similar to a truncated Taylor expansion of $v(k, x_j)$ around $t = 0$ with the time derivatives $\dot{v}(0, x_j), \ddot{v}(0, x_j)$ and $\dddot{v}(0, x_j)$ replaced by $\mathcal{D}_h V_j^{(0)}$, $\mathcal{D}_h^2 V_j^{(0)}$ and $\mathcal{D}_h^3 V_j^{(0)}$, respectively. If the initial data $V^{(0)}$ were smooth enough in space, since $\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2} - a\frac{\partial v}{\partial x}$, then $\mathcal{D}_h V_j^{(0)}$, $\mathcal{D}_h^2 V_j^{(0)}$ and $\mathcal{D}_h^3 V_j^{(0)}$ would simply be the fourth-order FD approximations of $\dot{v}(0, x_j), \ddot{v}(0, x_j)$ and $\dddot{v}(0, x_j)$, respectively. Hence, $V_j^{(1)} = \mathcal{K}_{k,h} V_j^{(0)}$ would be a fourth-order approximation of first time step solution $v(k, x_j)$.

For nonsmooth initial conditions, the convergence order analysis is more involved. We study this through the analysis of the Fourier transform of $\mathcal{K}_{k,h}$. The Fourier transform $\mathcal{F}(\mathcal{D}_h)$ of $\mathcal{D}_h$ is straightforward; see Section 3.3. To derive $\mathcal{F}(\mathcal{D}_h^2)$ and $\mathcal{F}(\mathcal{D}_h^3)$, we note that $\mathcal{D}_h = D_4^2 - aD_4$, $\mathcal{D}_h^2 = D_4^4 - aD_4^2 D_4 - aD_4 D_4^2 + a^2 D_4 D_4$, and $\mathcal{D}_h^3 = D_4^6 - a(D_4^4 D_4 + D_4^2 D_4 D_4^2 + D_4 D_4^4) + a^2(D_4^2 D_4 D_4 + D_4 D_4^2 D_4 + D_4 D_4 D_4^2) - a^3 D_4 D_4 D_4$, which give the relations

$$\mathcal{D}_h^2 V_j = \left(\frac{1}{h^4} D^{(2,0)} - a\frac{1}{h^3} D^{(2,1)} + a^2 \frac{1}{h^2} D^{(2,2)} \right)^T V_{j-4:j+4},$$

and

$$\mathcal{D}_h^3 V_j = \left(\frac{1}{h^6} D^{(3,0)} - a\frac{1}{h^5} D^{(3,1)} + a^2 \frac{1}{h^4} D^{(3,2)} - a^3 \frac{1}{h^3} D^{(3,3)} \right)^T V_{j-6:j+6},$$

where $D^{(i,j)}$ are column vectors with entries given in the tables

$D^{(2,0)}$	$\frac{1}{144}$	$\frac{-32}{144}$	$\frac{316}{144}$	$\frac{-992}{144}$	$\frac{1414}{144}$	$\frac{-992}{144}$	$\frac{316}{144}$	$\frac{-32}{144}$	$\frac{1}{144}$
$D^{(2,1)}$	$\frac{-1}{72}$	$\frac{24}{72}$	$\frac{-158}{72}$	$\frac{248}{72}$	0	$\frac{-248}{72}$	$\frac{158}{72}$	$\frac{-24}{72}$	$\frac{1}{72}$
$D^{(2,2)}$	$\frac{1}{144}$	$\frac{-16}{144}$	$\frac{64}{144}$	$\frac{16}{144}$	$\frac{-130}{144}$	$\frac{16}{144}$	$\frac{64}{144}$	$\frac{-16}{144}$	$\frac{1}{144}$

and

$D^{(3,0)}$	$\frac{-1}{1728}$	$\frac{48}{1728}$	$\frac{-858}{1728}$	$\frac{7024}{1728}$	$\frac{-27279}{1728}$	$\frac{58464}{1728}$	$\frac{-74796}{1728}$	$\frac{58464}{1728}$	$\frac{-27292}{1728}$	$\frac{7024}{1728}$	$\frac{-858}{1728}$	$\frac{48}{1728}$	$\frac{-1}{1728}$
$D^{(3,1)}$	$\frac{1}{576}$	$\frac{-40}{576}$	$\frac{572}{576}$	$\frac{-3512}{576}$	$\frac{9093}{576}$	$\frac{-9744}{576}$	0	$\frac{9744}{576}$	$\frac{-9093}{576}$	$\frac{3512}{576}$	$\frac{-572}{576}$	$\frac{40}{576}$	$\frac{-1}{576}$
$D^{(3,2)}$	$\frac{-1}{576}$	$\frac{32}{576}$	$\frac{-350}{576}$	$\frac{1504}{576}$	$\frac{-1791}{576}$	$\frac{-1536}{576}$	$\frac{4284}{576}$	$\frac{-1536}{576}$	$\frac{-1791}{576}$	$\frac{1504}{576}$	$\frac{-350}{576}$	$\frac{32}{576}$	$\frac{-1}{576}$
$D^{(3,3)}$	$\frac{1}{1728}$	$\frac{-24}{1728}$	$\frac{192}{1728}$	$\frac{-488}{1728}$	$\frac{-387}{1728}$	$\frac{1584}{1728}$	0	$\frac{-1584}{1728}$	$\frac{387}{1728}$	$\frac{499}{1728}$	$\frac{-192}{1728}$	$\frac{24}{1728}$	$\frac{-1}{1728}$

where the notation $V_{j_1:j_2}$ denotes the slice of the vector V. Therefore, we get

$$
\begin{aligned}
\mathcal{F}[\mathcal{D}_h](\omega) =& -\frac{\cos(2\theta) - 16\cos\theta + 15}{6h^2} - ia\frac{8\sin(\theta) - \sin(2\theta)}{6h},\\
\mathcal{F}[\mathcal{D}_h^2](\omega) =& \frac{707 + \cos(4\theta) - 32\cos(3\theta) + 316\cos(2\theta) - 992\cos\theta}{72h^4}\\
&+a^2\frac{-65 + \cos(4\theta) - 16\cos(3\theta) + 64\cos(2\theta) + 16\cos\theta}{72h^2}\\
&-ia\frac{\sin(4\theta) - 24\sin(3\theta) + 158\sin(2\theta) - 248\sin(\theta)}{36h^3},\\
\mathcal{F}[\mathcal{D}_h^3](\omega) =& \frac{-37398-\cos(6\theta)+48\cos(5\theta)-858\cos(4\theta)+7024\cos(3\theta)-27279\cos(2\theta)+58464\cos\theta}{864h^6}\\
&+a^2\frac{2142-\cos(6\theta)+32\cos(5\theta)-350\cos(4\theta)+1504\cos(3\theta)-1791\cos(2\theta)-1536\cos\theta}{288h^4}\\
&-ia\frac{-\sin(6\theta) + 40\sin(5\theta) - 572\sin(4\theta) + 3512\sin(3\theta) - 9093\sin(2\theta) + 9744\sin(\theta)}{288h^5}\\
&-ia^3\frac{-\sin(6\theta) + 24\sin(5\theta) - 192\sin(4\theta) + 488\sin(3\theta) + 387\sin(2\theta) - 1584\sin(\theta)}{864h^3},
\end{aligned}
$$

where we recall that $\theta = \omega h$. The RK3 iteration for the first time step in the frequency domain is

$$\hat{V}^{(1)}(\omega) = \left(1 + k\mathcal{F}[\mathcal{D}_h](\omega) + \frac{k^2}{2}\mathcal{F}[\mathcal{D}_h^2](\omega) + \frac{k^3}{6}\mathcal{F}[\mathcal{D}_h^3](\omega)\right)\hat{V}^{(0)}(\omega).$$

Similar to the discussion in the previous section, we study the convergence of one RK3 iteration for different magnitudes of ω with respect to h. Applying Maclaurin series expansion to $\mathcal{F}[\mathcal{D}_h](\omega)$,

$\mathcal{F}[\mathcal{D}_h^2](\omega)$ and $F[\mathcal{D}_h^3](\omega)$, we have

$$\begin{aligned}
\mathcal{F}[\mathcal{D}_h](\omega) &= -(\omega^2 + ia\omega) + \frac{1}{90}(\omega^6 + i3a\omega^5)h^4 + \cdots, \\
\mathcal{F}[\mathcal{D}_h^2](\omega) &= (\omega^4 - \frac{1}{45}\omega^8 h^4 + \cdots) - a^2(\omega^2 - \frac{1}{15}\omega^6 h^4 + \cdots) - ia(-2\omega^3 + \frac{4}{45}\omega^7 h^4 + \cdots) \\
&= (\omega^2 + ia\omega)^2 + \frac{1}{45}(-\omega^8 + 3a^2\omega^6 - i4a\omega^7)h^4 + \cdots, \\
\mathcal{F}[\mathcal{D}_h^3](\omega) &= (-\omega^6 + \frac{1}{30}\omega^{10}h^4 + \cdots) + a^2(3\omega^4 - \frac{7}{30}\omega^8 h^4 + \cdots) \\
&\quad - ia(3\omega^5 - \frac{1}{6}\omega^9 h^4 + \cdots) + ia^3(\omega^3 - \frac{1}{10}\omega^7 h^4 + \cdots) \\
&= -(\omega^2 + iaw)^3 + \frac{1}{30}(\omega^{10} - 7a^2\omega^8 + i5a\omega^9 - i3a^3\omega^7)h^4 + \cdots.
\end{aligned}$$

The results match our expectation by noticing that the exact frequency satisfies

$$\frac{\partial^n}{\partial t^n}\hat{v}(t,\omega) = (-1)^n(\omega^2 + ia\omega)^n\hat{v}(t,\omega),$$

and

$$\begin{aligned}
\hat{v}(t_1,\omega) =& e^{-(\omega^2+ia\omega)k}\hat{v}(t_0,\omega) \\
=& \left(1 - (\omega^2 + ia\omega)k + \frac{k^2}{2}(\omega^2 + ia\omega)^2 - \frac{k^3}{6}(\omega^2 + ia\omega)^3 + \ldots\right)\hat{v}(t_0,\omega).
\end{aligned}$$

Given $\hat{V}^{(0)}(\omega) = \hat{v}(t_0,\omega) + \hat{E}^{(0)}(\omega)$, where $\hat{E}^{(0)}$ is the frequency error at the initial time step, which is intrinsic to the initial condition discretization, we see that the error in $\hat{V}^{(1)}$ from one RK3 iteration is simply

$$\begin{aligned}
&\hat{E}^{(1)} = \hat{V}^{(1)} - \hat{v}(t_1) \\
&= \left(1 + k\mathcal{F}[\mathcal{D}_h](\omega) + \frac{k^2}{2}\mathcal{F}[\mathcal{D}_h^2](\omega) + \frac{k^3}{6}\mathcal{F}[\mathcal{D}_h^3](\omega)\right)\hat{E}^{(0)} + (-1)^{j+1}\sum_{j=4}^{\infty}\frac{(\omega^2 k + ia\omega k)^j}{j!}\hat{v}(t_0) \\
&+\left\{\frac{1}{90}\left((\omega^6 + i3a\omega^5)k + (-\omega^8 + 3a^2\omega^6 - i4a\omega^7)k^2 + \frac{1}{2}(\omega^{10} - 7a^2\omega^8 + i5a\omega^9 - i3a^3\omega^7)k^3\right)h^4\right. \\
&\left. + \cdots\right\}\hat{v}(t_0).
\end{aligned} \tag{3.32}$$

The error behaves differently in the high- and low-frequency regimes. We discuss this below. Recall that $k = dh \sim h$.

1. First consider the case $|\omega| = h^{-\beta}$ with $\beta < \frac{1}{2}$, i.e. $\omega^2 h \to 0$ as $h \to 0$. We see from (3.32) that the error $\hat{E}^{(1)}$ from one step of RK3 is comprised of $\hat{E}^{(0)}$ multiplied by a constant order coefficient, plus the remaining $\mathcal{O}(|\omega|^{\chi_4}h^4)$ terms. Therefore, we have

$$|\hat{V}^{(1)} - \hat{v}(t_1)| \le C_1'|\hat{E}^{(0)}| + C_2'|\omega|^{\chi_4}h^4|\hat{v}(t_0)|, \tag{3.33}$$

where C_i' and C_2' are some positive constants. Since $\hat{E}^{(0)}$ is multiplied by a constant order coefficient, it cannot be eliminated by RK3 time stepping. This explains why smoothing is necessary so that $\hat{E}^{(0)}$ is of high order as well.

2. Consider $|\omega| = h^{-\beta}$ with $\frac{1}{2} \le \beta \le \beta_{\max}$, i.e. $\omega^2 h \not\to 0$ as $h \to 0$. We see from (3.32) that the error $\hat{V}^{(1)} - \hat{v}(t_1) \not\to 0$ as $h \to 0$. Therefore, RK3 time stepping is not convergent in the high-frequency region $\omega = h^{-\beta}$ with $\frac{1}{2} \le \beta \le \beta_{\max}$, which lies exactly in the high-frequency exponential damping region of BDF4 scheme starting from the fourth time step, see Equation (3.28). As a result, even though RK3 is not convergent in a single time step in the high-frequency domain, the combination of RK3 as the initialization scheme and BDF4 for the general steps gives the expected $\mathcal{O}(|\omega|^{\chi_4} h^4)$ order of convergence.

Therefore, in summary, one step of RK3 time stepping gives

$$|\hat{E}^{(n)}| \le (\text{nonconvergent error}) \cdot \mathbb{1}_{\{\frac{1}{2} \le \beta \le \beta_{\max}\}} + \left(C_1' |\hat{E}^{(n-1)}| + C_2' h^4 |\omega|^{\chi_4} |\hat{v}(t_{n-1})| \right) \mathbb{1}_{\{\beta < \frac{1}{2}, \omega = 0\}}. \tag{3.34}$$

Remark 3.4.1. *Relation (3.34) shows that the high-frequency error of RK3 is not convergent, while the low-frequency error involves the error from the previous step and a fourth-order component. Combining RK3 and BDF4 (or any BDF method), the nonconvergent error of RK3 in the high-frequency domain is damped exponentially by BDF.*

In the low-frequency region, given $\hat{E}^{(0)}$, with two steps of RK3 initialization scheme, we have

$$|\hat{E}^{(1)}| \le C_1' |\hat{E}^{(0)}| + C_2' |\omega|^{\chi_4} h^4 |\hat{v}(t_0)|,$$
$$|\hat{E}^{(2)}| \le C_1' |\hat{E}^{(1)}| + C_2' |\omega|^{\chi_4} h^4 |\hat{v}(t_1)| \le C_3' |\hat{E}^{(0)}| + |\omega|^{\chi_4} h^4 (C_4' |\hat{v}(t_0)| + C_5' |\hat{v}(t_1)|),$$

where C_i' for $i = 1, \ldots, 5$ are positive constants, and with BDF3 at the third time step, using the results from Remark 3.3.3, we have

$$|\hat{E}^{(3)}| \le C_2 \max_{0 \le j \le 2} |\hat{E}^{(j)}| + C_3 h^4 |\omega|^{\chi_4} |\hat{v}(0)|.$$

We see that the final convergence behavior is determined by the accuracy of $\hat{E}^{(0)}$, which also needs to be of high order. We discuss this in the next section.

Remark 3.4.2. *We note that implicit schemes are usually the methods of choice when the initial condition is nonsmooth. For example, we have also implemented a third-order fully implicit Runge-Kutta method, the Radau IIA method [27], as the start up method for BDF steps and obtained the expected fourth-order convergence. An interesting contribution of this chapter is to show that, contrary to the common practices, an explicit Runge-Kutta method can also be applied as the start up scheme, when combined with BDF methods. Compared to implicit Runge-Kutta methods, explicit methods are cheaper and easier to implement. When the condition number of the discretized PDEs are extremely large, such as in chemical reactions problems, implicit methods may be more appropriate, but they are not the focus of this thesis.*

$V_j^{(0)}$	Dirac delta	Heaviside	Ramp
$j<-3$	0	0	0
$j=-3$	$\frac{(\alpha-1)^3}{36h}$	$-\frac{(\alpha-1)^4}{144}$	$\frac{(\alpha-1)^5 h}{720}$
$j=-2$	$-\frac{(11\alpha^3-30\alpha^2+24\alpha-4)}{36h}$	$\frac{(11\alpha^4-40\alpha^3+48\alpha^2-16-4)}{144}$	$\frac{(-11\alpha^5+50\alpha^4-80\alpha^3+40\alpha^2+20\alpha-20)h}{720}$
$j=-1$	$\frac{(14\alpha^3-27\alpha^2+15)}{18h}$	$\frac{(-7\alpha^4+18\alpha^3-30\alpha+18)}{36}$	$\frac{(14\alpha^5-45\alpha^4+150\alpha^2-180\alpha+51)h}{360}$
$j=0$	$\frac{(-14\alpha^3+15\alpha^2+12\alpha+2)}{18h}$	$\frac{(7\alpha^4-10\alpha^3-12\alpha^2-4\alpha+37)}{36}$	$-\frac{(14\alpha^5-25\alpha^4-40\alpha^3-20\alpha^2+370\alpha-350)h}{360}$
$j=1$	$-\frac{(-11\alpha^3+3\alpha^2+3\alpha+1)}{36h}$	$\frac{(-11\alpha^4+4\alpha^3+6\alpha^2+4\alpha+145)}{144}$	$-\frac{(-11\alpha^5+5\alpha^4+10\alpha^3+10\alpha^2+725\alpha-1439)h}{720}$
$j=2$	$-\frac{\alpha^3}{36h}$	$\frac{\alpha^4+144}{144}$	$-\frac{(\alpha^5+720\alpha-2160)h}{720}$
$j>2$	0	1	$(j+1-\alpha)h$

Table 3.1: Fourth-order smoothed discrete Dirac delta, Heaviside and ramp initial conditions

3.5 High-order smoothing of the initial conditions

Due to the nonsmoothness in the Dirac delta, Heaviside and ramp initial conditions, to achieve global fourth-order convergence, we still need to make sure that the initial condition is discretized to a high-order in the frequency domain. In this chapter, we perform initial condition smoothing using the smoothing operator suggested in [31]. In particular, a fourth-order smoothing operator Φ_4 is given by the inverse Fourier transform of

$$\hat{\Phi}_4(\omega) = \left(\frac{\sin(\omega/2)}{\omega/2}\right)^4 \left[1+\frac{2}{3}\sin^2(\omega/2)\right].$$

The smoothed initial condition is then computed from

$$\tilde{v}^{(0)}_{\text{Kreiss}}(x) = \frac{1}{h}\int_{-3h}^{3h} \Phi_4(s/h)v(t_0, x-s)ds. \tag{3.35}$$

We calculated explicit formulas for the fourth-order discrete Dirac delta, Heaviside and ramp initial conditions arising after applying the smoothing operator (3.35), and present them in Table 3.1. For the derivation of the explicit formulas, and a discussion of smoothing orders other than 4, see Appendix C.1. Using the smoothed initial condition discretizations given in Table 3.1, we guarantee that the initial conditions are fourth-order accurate in the frequency domain, and hence, $\hat{E}^{(0)} = \mathcal{O}(|\omega|^p h^4)$ for some positive constant p. Note that an appropriate linear combination of the smoothed ramp functions gives us the smoothed discretization of the bump function.

Remark 3.5.1. *A visualization of the Φ_4-smoothed initial conditions can be found in Figure 4.1, where alternative smoothings are derived.*

3.6 Solution error analysis

Now that we have analyzed the solution behavior in the Fourier frequency domain, we can perform inverse Fourier transform to recover the actual solution error. Assume that the initial conditions are already smoothed to high-order in the frequency domain, such that $|\hat{E}^{(0)}| = \mathcal{O}(|\omega|^p h^4)$ for some positive constant p. For the Dirac delta initial condition, we have

$$
\begin{aligned}
&E^{(n)}(x_j) = \frac{1}{2\pi}\int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{E}^{(n)}(\omega) e^{i\omega x_j} d\omega \\
\leq &\frac{1}{2\pi}\int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} C_1 e^{-\gamma n} \max_{0\leq l\leq 3} |\hat{V}^{(l)}| \mathbb{1}_{\{\frac{1}{2}\leq\beta\leq\beta_{\max}\}} e^{i\omega x_j} d\omega \\
&+ \frac{1}{2\pi}\int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \left(C_2 e^{\gamma n \mathrm{Re}(\mu)} \max_{0\leq l\leq 3} |\hat{E}^{(l)}| + C_3 h^4 |\omega|^{\chi_5} |\nu|^{\bar{\gamma}(n-1)} |\hat{V}(0)| \right) \mathbb{1}_{\{\beta<\frac{1}{2},\omega=0\}} e^{i\omega x_j} d\omega \\
= &C_1 e^{-\gamma n} \frac{\cos(\pi x_j/h) - \cos(x_j/\sqrt{h})}{i\pi x_j} \max_{0\leq l\leq 3} |\hat{V}^{(l)}| \\
&+ C_2 \frac{1}{2\pi}\int_{-\sqrt{\frac{1}{h}}}^{\sqrt{\frac{1}{h}}} e^{\gamma n \mathrm{Re}(\mu)} \max_{0\leq l\leq 3} |\hat{E}^{(l)}| e^{i\omega x_j} d\omega + C_3 \frac{1}{2\pi}\int_{-\sqrt{\frac{1}{h}}}^{\sqrt{\frac{1}{h}}} h^4 |\omega|^{\chi_5} |\nu|^{\bar{\gamma}(n-1)} |\hat{V}(0)| e^{i\omega x_j} d\omega \\
\leq &C_1 e^{-\gamma n} \max_{0\leq l\leq 3} |\hat{V}^{(l)}| + C_2 h^4 \int_{-\sqrt{\frac{1}{h}}}^{\sqrt{\frac{1}{h}}} |\omega|^p e^{\gamma n \mathrm{Re}(\mu)} e^{i\omega x_j} d\omega + C_3 h^4 |\hat{V}(0)| \int_{-\sqrt{\frac{1}{h}}}^{\sqrt{\frac{1}{h}}} |\omega|^{\chi_5} e^{-\bar{\gamma}\omega^2(n-1)k} e^{i\omega x_j} d\omega \\
\approx &C_1 e^{-\gamma n} \max_{0\leq l\leq 3} |\hat{V}^{(l)}| + \left(C_2 \int_{-\infty}^{\infty} |\omega|^p e^{\gamma n \mathrm{Re}(\mu)} e^{i\omega x_j} d\omega + C_3 |\hat{V}(0)| \int_{-\infty}^{\infty} |\omega|^{\chi_5} e^{-\bar{\gamma}\omega^2(n-1)k} e^{i\omega x_j} d\omega \right) h^4 \\
\leq &C_1 e^{-\gamma n} \max_{0\leq l\leq 3} |\hat{V}^{(l)}| + (C_2 + C_3 |\hat{V}(0)|) h^4,
\end{aligned}
$$

where the constants C_j with the same index are not necessarily equal. From the derivation, we see that fourth-order convergence is obtained if the discretized initial condition is smoothed to fourth-order. For the Heaviside and ramp initial conditions, a minor difference is that inverse Fourier transform does not directly give the solution. Instead, we have

$$
e^{-\eta x_j} E^{(n)}(x_j) = \frac{1}{2\pi}\int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{E}^{(n)}(\omega) e^{i\omega x_j} d\omega,
$$

which gives

$$
E^{(n)}(x_j) \leq e^{\eta x_j} C_1 e^{-\gamma n} \max_{0\leq l\leq 3} |\hat{V}^{(l)}| + e^{\eta x_j} (C_2 + C_3 |\hat{V}(0)|) h^4.
$$

We see that the solution error contains an error from the initial steps which is damped exponentially, and a term which is $\mathcal{O}(h^4)$.

Chapter 4

Novel high-order smoothing techniques

In the previous chapter, we developed a high-order time stepping method for parabolic PDEs with nonsmooth initial conditions. To achieve the final fourth-order convergence, we applied a convolution-type fourth-order smoothing to the initial condition. However, this smoothing method as presented in [31] has certain limitations that cause difficulties when applied to initial conditions with nonsmooth points too close to the boundary, or when dealing with nonuniform grids. In this chapter, we present novel smoothing techniques that cancel out the low-order terms of the quantization errors in the Fourier domain arising from discretization. These techniques usually require fewer points to be smoothed than the convolution-type smoothings used in Chapter 3, and can be flexibly applied even when the nonsmooth point is very close to the boundary. Furthermore, we extend the smoothing procedure to nonuniform grids. Moreover, with an additive approach, we develop smoothing formulas for more general nonsmooth but piece-wise smooth functions.

Consider again the model convection-diffusion PDE (3.1) under initial condition

$$v(0,x) = g(x).$$

Following [31], we assume that g is a tempered distribution such that its Fourier transform $\hat{g}$ is locally integrable and satisfies a growth condition

$$|\hat{g}(\omega)| \le C(|\omega| + 1)^{\beta}, \tag{4.1}$$

where β is some real number, and C is a positive constant. It has been shown in [31] that an approximate solution obtained from a suitable order-χ explicit difference method can be $\mathcal{O}(h^{\chi})$ accurate only when the initial condition $\hat{g}$ is smooth enough such that g satisfies the growth condition (4.1) with $\beta < -\chi - 3$.

Under nonsmooth initial conditions such as the Dirac delta, Heaviside, and ramp functions, the smoothness requirement on β is usually not satisfied. Hence, a difference operator may falsely propagate the large high-frequency components in the initial condition and cause spurious oscillations in the approximate solution around the nonsmooth point, leading to a degenerated order of accuracy. This behaviour is well investigated for the Crank-Nicolson (CN) method. Although

the second-order CN is well known to be unconditionally (as far as time stepsize is concerned) stable in L_2-norm, the convergence order may be less than two for nonsmooth initial data, and the point-wise error can be even worse as in the case of spurious oscillations. It was first suggested in [44] to restore the second-order convergence by replacing the first CN step with two backward Euler steps, often referred to as Rannacher timestepping.

As mentioned in the previous chapter, with the CN time-stepping (and the diagonal Pade schemes in general), the nonsmoothness in the initial condition causes two sources of errors: the lower order error in the high-frequency Fourier domain, and the quantization error due to the placement of the nonsmooth point on the numerical grid. The two sources of errors can be eliminated either together or separately. Convolution-type smoothing operators M_h of order χ are proposed in [31], such that the approximate solution obtained from an explicit difference method with the smoothed initial condition $M_h g$ is of $\mathcal{O}(h^\chi)$ accuracy. The essential idea is that the Fourier transform of the smoothing operator M_h is an order-χ approximation of the identity (correcting quantization error), and can at the same time reduce the high-frequency components in the initial condition that are falsely propagated by the difference operator (correcting high-frequency error). Therefore, with such smoothing operators of suitable orders applied to the initial condition, both the frequency errors and the quantization errors are dealt with together. It is often seen in practice to combine Rannacher timestepping with the convolution-type smoothing of initial conditions. However, it is worth pointing out that Rannacher timestepping is not necessary when a convolution-type smoothing operator of suitable order is applied to the initial data, as also empirically observed (though not mathematically analyzed) in [15]. One can also deal with the two sources of errors separately. In the previous chapter, we proposed a fourth-order 2RK3-BDF3-BDF4 time stepping method that can automatically damp out the high-frequency components in the nonsmooth initial conditions, due to the stiff-decay property of BDF. The quantization error is handled separately by using a fourth-order convolution-type smoothing operator suggested in [31]. Some limitations of such a smoothing operator as presented in [31] include: that the space domain has to be uniformly discretized; that a fixed number of points on both sides of the singular point have to be smoothed regardless of the regularity of the nonsmooth function; and that the convolution operation can be expensive to compute for complicated nonsmooth functions.

It is the objective of this chapter to extend the application of the smoothing operators to more complicated scenarios that without these restrictions. To this end, we suggest using an additive modification strategy to perform the smoothing operation. We derive explicit smoothing modifications to the basic types of nonsmoothness exhibited by the Dirac delta, Heaviside, ramp and quadratic ramp functions, and so on, and then show that any general nonsmoothness on piecewise analytic functions can be expressed as linear combinations of the basic types of discontinuities. Therefore, we can calculate the smoothing modifications for very general nonsmooth initial conditions of high complexity, including smoothing on nonuniform grids. Moreover, we derive new smoothing modifications that can reduce the quantization error alone to any high order, with the extra flexibility to choose the points of modification, such that the smoothing scheme is able to deal with the extreme situation where the nonsmooth point is located very close to the boundary.

This chapter is organized as follows. In Section 4.1, we introduce the four basic types of nonsmooth initial conditions. In Section 4.2, we briefly review the existing smoothing technique

proposed by [31]. In Section 4.3, we derive the semi-discrete Fourier transform of the four discrete initial conditions, that reveals the low-order quantization errors, and formulate new smoothing modifications that eliminate the low-order quantization errors. In Section 4.4, we go beyond this observation and propose smoothing modifications that are exact in the frequency domain, i.e. eliminating all the quantization errors. In Section 4.5, we apply the additive modification strategy to derive smoothing modifications to the initial condition discretization on nonuniform grids. In Section 4.6, we derive a general formula that provides the smoothing modifications for any general nonsmoothness, including discretization on nonuniform grids.

4.1 Preliminaries

We investigate four basic types of nonsmooth initial conditions. In addition to the Dirac delta function $\delta(x)$ given by (3.2), the Heaviside function $H(x)$ given by (3.3), and the ramp function $C(x)$ given by (3.4), we also consider the quadratic ramp function

$$v(0,x) = Q(x) \equiv \frac{x^2}{2} H(x). \tag{4.2}$$

We show later that an appropriate linear combination of each of the four basic functions gives an approximation to more general singularities around a point.

The numerical solution is computed on a grid $\{x_j\}$, for $j = \ldots, -1, 0, 1, \ldots$, with grid stepsizes $h_j \equiv x_j - x_{j-1}$ and $x_{-1} < 0 \leq x_0$. The nonsmooth point x_K is fixed at $x = 0$ and does not necessarily lie exactly on a grid point[1]. To accommodate this, we introduce a parameter $\alpha \in (0, 1]$ such that $\alpha \equiv 1 - \frac{x_0}{x_0 - x_{-1}}$. We see that $\alpha = 1$ corresponds to the case where the nonsmooth point lies exactly at the grid point x_0. On a general grid $\{x_j\}$, the delta function can be discretized as

$$\delta_\alpha(x_j) \equiv \begin{cases} \frac{1}{h_0} \mathbb{1}_{\alpha < \frac{1}{2}}, & j = -1, \\ \frac{1}{h_0}(1 - \mathbb{1}_{\alpha < \frac{1}{2}}), & j = 0, \\ 0, & \text{else}, \end{cases} \tag{4.3}$$

where $\mathbb{1}_{\alpha < \frac{1}{2}}$ is the indicator function that is 1 when $\alpha < \frac{1}{2}$, and 0 otherwise. The discretization of (3.3), (3.4) and (4.2) can be simply sampled from the continuous respective functions so that

$$H_\alpha(x_j) \equiv \begin{cases} 1, & j \geq 0, \\ 0, & \text{else}, \end{cases} \tag{4.4}$$

$$C_\alpha(x_j) \equiv x_j H_\alpha(x_j), \tag{4.5}$$

$$Q_\alpha(x_j) \equiv \frac{x_j^2}{2} H_\alpha(x_j).. \tag{4.6}$$

As shown in Chapter 3, discretizations of nonsmooth initial conditions may lead to deterioration of the convergence rate of a high order method, due to two issues: (a) the nonnegligible high-frequency components in the initial conditions may be falsely propagated to later time steps, and

[1] Note that that the subscript K in x_K is not an index, but a notation to distinguish the nonsmooth point.

(b) the representation of the exact initial conditions in the frequency domain is of low order. The latter is partly related to the alignment of the nonsmooth point on the grid as we show later.

4.2 Review of the smoothing technique in [31]

To solve the two problems, a convolution-type smoothing scheme was proposed in [31], in which they define the operators M_h on a uniform grid of stepsize h to be such that

$$M_h g = \Phi_h * g, \quad \Phi_h(x) = h^{-1}\Phi(h^{-1}x)$$

where $*$ denotes convolution operation, Φ is a tempered distribution on $\mathbb{R}$ such that Φ is bounded, and

$$\hat{\Phi}(\omega) = 1 + \mathcal{O}(\omega^\chi), \quad \text{as} \quad \omega \to 0, \tag{4.7}$$

which implies that M_h approximates the identity with order χ, while

$$\hat{\Phi}(\omega) = \mathcal{O}(|\omega - 2j\pi|^\chi), \quad \text{as} \quad \omega \to 2j\pi, \tag{4.8}$$

uniformly for integers j, which implies that high-frequency components in $\hat{g}$ get reduced by M_h. The fourth-order smoothing operator given by (3.35) is a particular example that satisfies these conditions. For a discussion of smoothing operators M_h^χ with orders χ other than 4, see Appendix C.1.

In the following, we relate the discretization of the initial conditions and their representations in the Fourier domain. We show in detail the Fourier transforms of the discrete nonsmooth initial conditions (4.3), (4.4), (4.5) and (4.6), and reveal the problems with such discretizations. We then propose alternative discretization schemes, which, together with high-order time stepping and space discretization as in Chapter 3, can be used to solve the model convection-diffusion (3.1) to a high order. We mostly focus our discussion on the basic initial conditions: the Dirac delta, Heaviside, ramp and quadratic ramp functions, and later use linear combinations of these initial conditions to represent more complicated nonsmooth functions.

4.3 Fourier analysis and smoothing modifications of the discrete initial conditions

To see why the discretizations (4.3), (4.4), (4.5) and (4.6) lead to low-order errors in the frequency domain, we apply semi-discrete Fourier transform to the initial conditions, and compare them with the Fourier transform of the exact continuous initial conditions. We first study the solution on a uniform grid, i.e., when $h_j = h$ for all j, where h is the uniform spatial stepsize. From the definition of α, the grid points are $x_j = (j + (1 - \alpha))h$.

4.3.1 Fourier transform of the discrete initial conditions

For the discrete Dirac delta function (4.3), its semi-discrete Fourier transform is

$$\hat{v}^{(0)}_{\delta,h,\alpha}(\omega) \equiv h \sum_{j=-\infty}^{\infty} e^{-i\omega x_j} \delta_\alpha(x_j) = e^{i\alpha\omega h} \mathbb{1}_{\alpha<\frac{1}{2}} + e^{-i(1-\alpha)\omega h}(1 - \mathbb{1}_{\alpha<\frac{1}{2}}).$$

Applying Taylor expansion to $e^{i\alpha\omega h}$ and $e^{-i(1-\alpha)\omega h}$, and using the fact that $\hat{v}^{(0)}_{\delta} = 1$, we obtain

$$\begin{aligned}\hat{v}^{(0)}_{\delta,h,\alpha}(\omega) = \hat{v}_\delta(t=0) &+ \left(i\alpha\omega h - \frac{\alpha^2}{2}\omega^2 h^2 - i\frac{\alpha^3}{6}\omega^3 h^3 + \frac{\alpha^4}{24}\omega^4 h^4 + \mathcal{O}(\omega^5 h^5)\right) \mathbb{1}_{\alpha<\frac{1}{2}} \\ &+ \left(-i(1-\alpha)\omega h - \frac{(1-\alpha)^2}{2}\omega^2 h^2 + i\frac{(1-\alpha)^3}{6}\omega^3 h^3 + \frac{(1-\alpha)^4}{24}\omega^4 h^4 + \mathcal{O}(\omega^5 h^5)\right) \mathbb{1}_{\alpha\geq\frac{1}{2}}.\end{aligned} \tag{4.9}$$

We see from (4.9) that the discrete delta function (4.3) is a first-order approximation to the true delta function in the frequency domain. When aligning the nonsmooth point $x = 0$ with a grid point, i.e. $\alpha = 1$, the semi-discrete Fourier representation is accurate to a high order. Moreover, we see that the presence of the indicator functions in the discretization (4.3) always results in coefficient of the dominant error term in (4.9) that are less than $\frac{1}{2}$.

Different from the Dirac delta function, the Fourier transforms of the Heaviside, ramp and quadratic ramp functions are not convergent. Even though the Fourier transform does exist in the distribution sense, the summation of the corresponding semi-discrete Fourier transform is not convergent. Therefore, instead of applying the Fourier transform directly to $H(x)$, $C(x)$ and $Q(x)$, we compute the Fourier transform of $e^{-\eta x}H(x)$, $e^{-\eta x}C(x)$ and $e^{-\eta x}Q(x)$ with some real positive number $\eta > 0$. This is the same as the usual Fourier transform but with the frequency being complex numbers as we demonstrate below.

For the discrete Heaviside function (4.4), its semi-discrete Fourier transform is

$$\hat{v}^{(0)}_{H,h,\alpha}(\kappa) \equiv h \sum_{j=-\infty}^{\infty} e^{-i\kappa x_j} H_\alpha(x_j) = \frac{h e^{-i(1-\alpha)\kappa h}}{1 - e^{-i\kappa h}},$$

where $\kappa = \omega - i\eta$. Applying Taylor expansion to $e^{-i(1-\alpha)\kappa h}$ and using the fact that $\hat{v}^{(0)}_{H} = \frac{1}{i\kappa}$, we obtain

$$\hat{v}^{(0)}_{H,h,\alpha} = \hat{v}_H(t=0) + \left(\alpha - \frac{1}{2}\right) h + i\frac{1}{2}\left(\left(\alpha - \frac{1}{2}\right)^2 - \frac{1}{12}\right)\kappa h^2 + \frac{\alpha(1-\alpha)(2\alpha-1)}{12}\kappa^2 h^3 + \mathcal{O}(\kappa^3 h^4). \tag{4.10}$$

The details of showing (4.10) are given in Appendix B.2.1.

For the discrete ramp initial condition (4.5), its semi-discrete Fourier transform is

$$\hat{v}^{(0)}_{C,h,\alpha} \equiv h \sum_{j=-\infty}^{\infty} e^{-i\kappa x_j} \max(x_j, 0) = h^2 e^{-i\kappa(1-\alpha)h} \left(\frac{1}{(1 - e^{-i\kappa h})^2} - \frac{\alpha}{1 - e^{-i\kappa h}}\right).$$

Applying Taylor expansion, we get

$$\hat{v}^{(0)}_{C,h,\alpha} = \hat{v}_C(t=0) - \frac{1}{2}\left(\alpha^2 - \alpha + \frac{1}{6}\right)h^2 + i\frac{1}{6}\alpha(1-\alpha)(2\alpha-1)\kappa h^3 + \mathcal{O}(\kappa^2 h^4). \tag{4.11}$$

The details of showing (4.11) are given in Appendix B.2.2. We see that for both the discrete Heaviside and ramp functions, aligning the nonsmooth point $x = 0$ with a grid point, i.e. $\alpha = 1$, introduces lower-order error terms in the Fourier transform.

For the discrete quadratic ramp initial condition (4.6), its semi-discrete Fourier transform is

$$\hat{v}^{(0)}_{Q,h,\alpha} \equiv h\sum_{j=-\infty}^{\infty} e^{-i\kappa x_j}\frac{x_j^2}{2}H_\alpha(x_j) = \frac{h^3}{2}\frac{e^{-i\kappa(1-\alpha)h}}{(1-e^{-i\kappa h})^3}\left((1-\alpha)^2 - (2\alpha^2-2\alpha-1)e^{-i\kappa h} + \alpha^2 e^{-i2\kappa h}\right).$$

Applying Taylor expansion, we get

$$\hat{v}^{(0)}_{Q,h,\alpha} = \hat{v}_Q(t=0) + \frac{\alpha}{12}\left(2\alpha^2 - 3\alpha + 1\right)h^3 + i\frac{\alpha^2}{8}\left(\alpha^2 - 2\alpha + 1\right)\kappa h^4 + \mathcal{O}(\kappa^2 h^5). \tag{4.12}$$

The details of showing (4.12) are given in Appendix B.2.3.

4.3.2 Smoothing modifications of the discrete initial conditions

In this section, we derive explicit modifications to the discrete initial conditions (4.3), (4.4), (4.5) and (4.6), so that their Fourier transforms are fourth-order accurate. For demonstration, consider the discretized Dirac delta function (4.3). We add modifications $c^{[4]}_{\delta,-2}, c^{[4]}_{\delta,-1}, c^{[4]}_{\delta,0}, c^{[4]}_{\delta,1}$ to $\delta_\alpha(x_{-2}), \delta_\alpha(x_{-1}), \delta_\alpha(x_0), \delta_\alpha(x_1)$, respectively, to apply high-order smoothing, where the superscript "[4]" implies fourth-order smoothing, and similarly for other numbers. In order to eliminate the low-order terms in (4.9), we need to have

$$\begin{aligned}
&h\sum_{j=-2}^{1} c^{[4]}_{\delta,j} e^{-i\omega(j+(1-\alpha))h} \\
=&-\left(i\alpha\omega h - \frac{\alpha^2}{2}\omega^2h^2 - i\frac{\alpha^3}{6}\omega^3h^3 + \frac{\alpha^4}{24}\omega^4h^4 + \mathcal{O}(\omega^5h^5)\right)\mathbb{1}_{\alpha<\frac{1}{2}} \\
&-\left(-i(1-\alpha)\omega h - \frac{(1-\alpha)^2}{2}\omega^2h^2 + i\frac{(1-\alpha)^3}{6}\omega^3h^3 + \frac{(1-\alpha)^4}{24}\omega^4h^4 + \mathcal{O}(\omega^5h^5)\right)\mathbb{1}_{\alpha\geq\frac{1}{2}}.
\end{aligned} \tag{4.13}$$

Applying Taylor expansion to the left-hand side of (4.13) and combining terms, we obtain

$$\begin{aligned}
h\sum_{j=-2}^{1} c^{[4]}_{\delta,j} e^{-i\omega(j+(1-\alpha))h} =& (c^{[4]}_{\delta,-2} + c^{[4]}_{\delta,-1} + c^{[4]}_{\delta,0} + c^{[4]}_{\delta,1})h \\
&+ i((1+\alpha)c^{[4]}_{\delta,-2} + \alpha c^{[4]}_{\delta,-1} - (1-\alpha)c^{[4]}_{\delta,0} - (2-\alpha)c^{[4]}_{\delta,1})\omega h^2 \\
&+ \left(-\frac{(1+\alpha)^2}{2}c^{[4]}_{\delta,-2} - \frac{\alpha^2}{2}c^{[4]}_{\delta,-1} - \frac{(1-\alpha)^2}{2}c^{[4]}_{\delta,0} - \frac{(2-\alpha)^2}{2}c^{[4]}_{\delta,1}\right)\omega^2h^3 \\
&+ i\left(-\frac{(1+\alpha)^3}{6}c^{[4]}_{\delta,-2} - \frac{\alpha^3}{6}c^{[4]}_{\delta,-1} + \frac{(1-\alpha)^3}{6}c^{[4]}_{\delta,0} + \frac{(2-\alpha)^3}{6}c^{[4]}_{\delta,1}\right)\omega^3h^4 + \mathcal{O}(\omega^5h^5).
\end{aligned} \tag{4.14}$$

To match the terms between the right-hand sides of (4.13) and (4.14), we need

$$\begin{cases} c^{[4]}_{\delta,-2} + c^{[4]}_{\delta,-1} + c^{[4]}_{\delta,0} + c^{[4]}_{\delta,1} & = 0, \\ (1+\alpha)c^{[4]}_{\delta,-2} + \alpha c^{[4]}_{\delta,-1} - (1-\alpha)c^{[4]}_{\delta,0} - (2-\alpha)c^{[4]}_{\delta,1} & = \frac{1}{h}(-\alpha \mathbb{1}_{\alpha<0.5} + (1-\alpha)\mathbb{1}_{\alpha\geq 0.5}), \\ -\frac{(1+\alpha)^2}{2}c^{[4]}_{\delta,-2} - \frac{\alpha^2}{2}c^{[4]}_{\delta,-1} - \frac{(1-\alpha)^2}{2}c^{[4]}_{\delta,0} - \frac{(2-\alpha)^2}{2}c^{[4]}_{\delta,1} & = \frac{1}{2h}(\alpha^2 \mathbb{1}_{\alpha<0.5} + (1-\alpha)^2\mathbb{1}_{\alpha\geq 0.5}), \\ -\frac{(1+\alpha)^3}{6}c^{[4]}_{\delta,-2} - \frac{\alpha^3}{6}c^{[4]}_{\delta,-1} + \frac{(1-\alpha)^3}{6}c^{[4]}_{\delta,0} + \frac{(2-\alpha)^3}{6}c^{[4]}_{\delta,1} & = \frac{1}{6h}(\alpha^3 \mathbb{1}_{\alpha<0.5} - (1-\alpha)^3\mathbb{1}_{\alpha\geq 0.5}). \end{cases}$$

Solving the equations for $c^{[4]}_{\delta,-2}, c^{[4]}_{\delta,-1}, c^{[4]}_{\delta,0}$ and $c^{[4]}_{\delta,1}$, we get

$$\begin{cases} c^{[4]}_{\delta,-2} & = -\frac{1}{6h}(\alpha^3 - 3\alpha^2 + 2\alpha), \\ c^{[4]}_{\delta,-1} & = \frac{1}{2h}(\alpha^3 - 2\alpha^2 - \alpha) + \frac{1}{h}\mathbb{1}_{\alpha\geq 0.5}, \\ c^{[4]}_{\delta,0} & = \frac{1}{2h}(-\alpha^3 + \alpha^2 + 2\alpha) - \frac{1}{h}\mathbb{1}_{\alpha\geq 0.5}, \\ c^{[4]}_{\delta,1} & = \frac{1}{6h}(\alpha^3 - \alpha). \end{cases}$$

Let $c^{[4]}_{\delta,j} = 0$ for $j \geq 2$ and $j \leq -3$. We obtain a fourth-order smoothed discretization of the delta initial condition

$$\delta^{[4]}_\alpha(x_j) \equiv \delta_\alpha(x_j) + c^{[4]}_{\delta,j}, \tag{4.15}$$

and in the Fourier domain

$$\hat{\delta}^{[4]}_\alpha = \hat{v}_\delta(t=0) + \mathcal{C}_4\omega^4 h^4 + \mathcal{O}(\omega^5 h^5), \quad \text{with} \quad \mathcal{C}_4 = \frac{\alpha(\alpha^2-1)(2-\alpha)}{24}.$$

Note that $c^{[4]}_{\delta,j} = c^{[4]}_{\delta,j}(\alpha)$ is a function of α. Moreover, the final smoothed discretization $\hat{\delta}^{[4]}_\alpha$ for $\alpha < 0.5$ and $\alpha \geq 0.5$ are the same, and with the same leading order coefficient $\mathcal{C}_4$ in the frequency error. Therefore, in the following, we drop the indicator function and always let $\delta_\alpha(x_{-1}) = \frac{1}{h_0}$, that is,

$$\delta_\alpha(x_j) \equiv \begin{cases} \frac{1}{h_0}, & j = -1, \\ 0, & \text{else.} \end{cases} \tag{4.16}$$

From now on, we use (4.16) instead of (4.3) for the discretization of the delta function. Another point to note is that with fourth-order smoothing, the coefficient of $\mathcal{O}(\omega^4 h^4)$ is dependent on the alignment α. Hence, we may not see stable fourth-order convergence with grid refinements. To obtain stable fourth-order convergence, we just need to modify one more grid value, e.g. at x_2, and apply fifth-order smoothing; see Appendix C.

It is worth mentioning that the points of modification are not unique. Varying the points of modification changes the magnitude of the constant coefficient in front of leading order term. It is usually the best to center the singularity around the smoothing points so that the constant is small. Moving the smoothing points away from the singularity makes the coefficient larger. Applying smoothing to the points that are too far away from the singularity is not advisable, since

$c_{\delta,j}^{[4]}$	$x_{-3}:x_0$	$x_{-2}:x_1$	$x_{-1}:x_2$	$x_0:x_3$
$j<-3$	0	0	0	0
$j=-3$	$\frac{-\alpha^3+\alpha}{6h}$	0	0	0
$j=-2$	$\frac{\alpha^3+\alpha^2-2\alpha}{2h}$	$\frac{-\alpha^3+3\alpha^2-2\alpha}{6h}$	0	0
$j=-1$	$\frac{-\alpha^3-2\alpha^2+\alpha}{2h}$	$\frac{\alpha^3-2\alpha^2-\alpha}{2h}$	$\frac{-\alpha^3+6\alpha^2-11\alpha}{6h}$	0
$j=0$	$\frac{\alpha^3+3\alpha^2+2\alpha}{6h}$	$\frac{-\alpha^3+\alpha^2+2\alpha}{2h}$	$\frac{\alpha^3-5\alpha^2+6\alpha}{2h}$	$\frac{-\alpha^3+9\alpha^2-26\alpha}{6h}$
$j=1$	0	$\frac{\alpha^3-\alpha}{6h}$	$\frac{-\alpha^3+4\alpha^2-3\alpha}{2h}$	$\frac{\alpha^3-8\alpha^2+19\alpha}{2h}$
$j=2$	0	0	$\frac{\alpha^3-3\alpha^2+2\alpha}{6h}$	$\frac{-\alpha^3+7\alpha^2-14\alpha}{2h}$
$j=3$	0	0	0	$\frac{\alpha^3-6\alpha^2+11\alpha}{6h}$
$j>3$	0	0	0	0
$\mathcal{C}_4$	$\frac{-\alpha^4+2\alpha^3+\alpha^2-2\alpha}{24}$	$\frac{-\alpha^4-2\alpha^3+\alpha^2+2\alpha}{24}$	$\frac{-\alpha^4+6\alpha^3-11\alpha^2+6\alpha}{24}$	$\frac{-\alpha^4+10\alpha^3-35\alpha^2+50\alpha}{24}$

Table 4.1: Fourth-order smoothing modifications to discrete Dirac delta function (4.16) along with the leading order coefficient $\mathcal{C}_4$ of the $\mathcal{O}(\omega^4 h^4)$ term of the error in its Fourier transform representation.

the constant coefficient of the leading order term becomes too large, even though the leading term is in the correct order. A list of fourth-order smoothing modifications based on (4.16) to different sets of points around the singularity are given in Table 4.1. The ability to choose different points of modification gives us more flexibility to perform a smoothing operation regardless of the location of the singularity. In the extreme case that, for example, the singularity is very near the left boundary, we can simply let the smoothing points be the neighboring points to the right of (including) the left boundary.

Following exactly the same procedure to eliminate the low-order terms in (4.10), (4.11) and (4.12), we can derive the smoothing modifications of the Heaviside, ramp and quadratic ramp functions. We list the results in Tables 4.2 and 4.3. For completeness, the derivation of fourth- and fifth-order smoothings for the Heaviside, ramp and quadratic ramp functions are given in Appendix C. In order to visualize the effect of different smoothings for the initial conditions considered, we present Figure 4.1, where we plot the initial conditions and their smoothed versions.

It is important to highlight that the convolution-type smoothing operator, as expressed in (3.35), can also be considered as an additive modification applied to the initial conditions. This perspective provides more flexibility in deriving smoothing procedures for handling general nonsmoothness, and/or on nonuniform grids, as we will demonstrate later.

4.4 Exact-in-frequency discretization

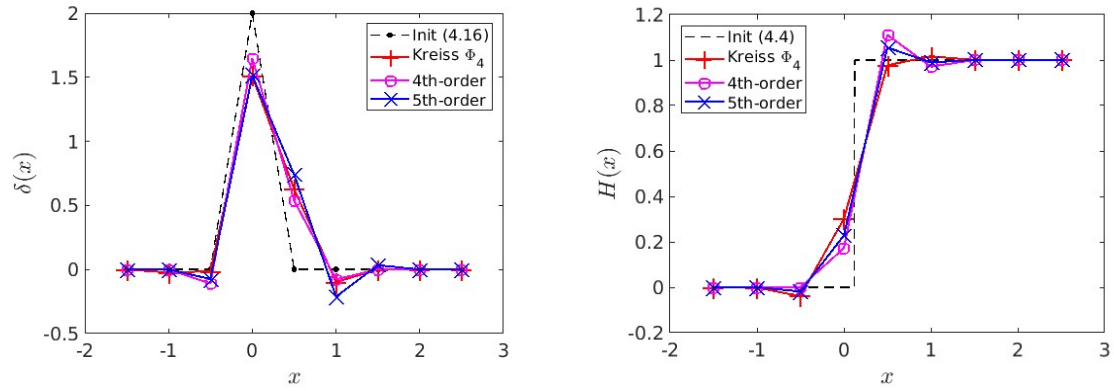
It is obvious that by modifying more grid values, we can obtain arbitrarily high-order discretization. A natural question to ask is: Can we eliminate all the low-order terms by modifying all grid values, so that the discretization is exact in the frequency domain. The answer is Yes! To achieve this,

$c_{H,j}^{[4]}$	$x_{-2} : x_0$	$x_{-1} : x_1$	$x_0 : x_2$
$j < -2$	0	0	0
$j = -2$	$\frac{-4\alpha^3+6\alpha^2-1}{24}$	0	0
$j = -1$	$\frac{2\alpha^3-6\alpha+3}{6}$	$\frac{-4\alpha^3+18\alpha^2-24\alpha+9}{24}$	0
$j = 0$	$\frac{-4\alpha^3-6\alpha^2+1}{24}$	$\frac{2\alpha^3-6a^2+1}{6}$	$\frac{-4a^3+30\alpha^2-72\alpha+31}{24}$
$j = 1$	0	$\frac{-4\alpha^3+6\alpha^2-1}{24}$	$\frac{2\alpha^3-12\alpha^2+18\alpha-7}{6}$
$j = 2$	0	0	$\frac{-4\alpha^3+18\alpha^2-24\alpha+9}{24}$
$j > 2$	0	0	0
$\mathcal{C}_4$	$i\frac{30\alpha^4-60\alpha^2+11}{720}$	$i\frac{30\alpha^4-120\alpha^3+120\alpha^2-19}{720}$	$i\frac{30\alpha^4-240\alpha^3+660\alpha^2-720\alpha+251}{720}$

Table 4.2: Fourth-order smoothing modifications to discrete Heaviside function (4.4) along with the leading order coefficient $\mathcal{C}_4$ of the $\mathcal{O}(\kappa^3 h^4)$ term of the error in its Fourier transform representation.

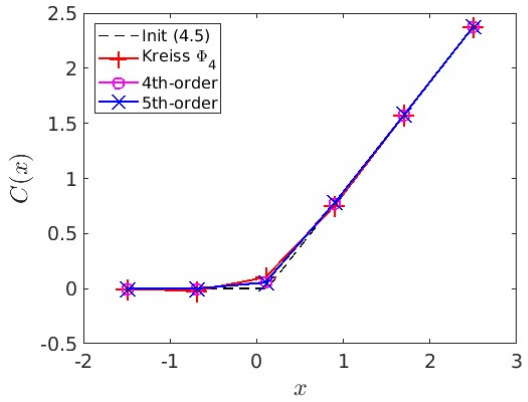
	$c_{C,j}^{[4]}$		$c_{Q,j}^{[4]}$	
	$x_{-1} : x_0$	$x_0 : x_1$	x_{-1}	x_0
$j < -1$	0	0	0	0
$j = -1$	$\frac{(1-2\alpha^3+6\alpha^2-5\alpha)h}{12}$	0	$\frac{\alpha(-2\alpha^2+3\alpha-1)h^2}{12}$	0
$j = 0$	$\frac{(2\alpha^3-\alpha)h}{12}$	$\frac{(-2\alpha^3+12\alpha^2-11\alpha+2)h}{12}$	0	$\frac{\alpha(-2\alpha^2+3\alpha-1)h^2}{12}$
$j = 1$	0	$\frac{(2\alpha^3-6\alpha^2+5\alpha-1)h}{12}$	0	0
$j > 1$	0	0	0	0
$\mathcal{C}_4$	$\frac{10\alpha^4-20\alpha^3+10\alpha-1}{240}$	$\frac{10\alpha^4-60\alpha^3+120\alpha^2-90\alpha+19}{240}$	$i\frac{\alpha^2(1-\alpha^2)}{24}$	$i\frac{\alpha(1-\alpha)^2(2-\alpha)}{24}$

Table 4.3: Fourth-order smoothing modifications to discrete ramp and quadratic ramp functions (4.5), (4.6) along with the leading order coefficient $\mathcal{C}_4$ of the $\mathcal{O}(\kappa^2 h^4)$ and $\mathcal{O}(\kappa h^4)$ terms of the errors, in their Fourier transform representations.

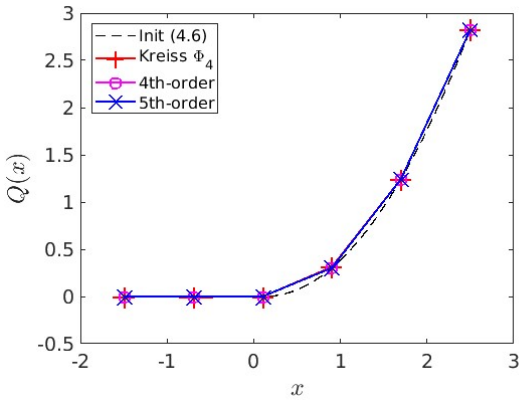


(a) Delta

(b) Heaviside



(c) Ramp



(d) Quadratic ramp

Figure 4.1: Discrete values on the grid points of the Dirac delta, Heaviside, ramp and quadratic ramp functions connected by broken lines without smoothing (Dirac delta function taken from (4.16)), in comparison with the respective values of the fourth-order Kreiss smoothing given in Table 3.1, of our new fourth-order smoothings given in Tables 4.1, 4.2 and 4.3, and of our new fifth-order smoothings given in Tables C.1, C.2 and C.3.

we need the semidiscrete Fourier transform of the discrete initial condition to be exactly equal to $\hat{v}_\delta(t=0)=1$, i.e.

$$h\sum_{j=-\infty}^{\infty}\delta_j e^{-i\omega x_j}=1.$$

Let $\theta\equiv\omega h$, and multiply both sides by $\frac{1}{h}e^{i(1-\alpha)\omega h}$, we get

$$\sum_{j=-\infty}^{\infty}\delta_j e^{-ij\theta}=\frac{1}{h}e^{i(1-\alpha)\theta}.$$

Therefore, we see that the discrete values δ_j are the inverse discrete-time Fourier transform (DTFT) of $\frac{1}{h}e^{i(1-\alpha)\theta}$, and can be easily computed by

$$\delta_j=\frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{e^{i(1-\alpha)\theta}}{h}e^{ij\theta}d\theta=\frac{\sin\left(\frac{\pi x_j}{h}\right)}{\pi x_j}.$$

Similarly, for the Heaviside function, we need

$$h\sum_{j=-\infty}^{\infty}\left(H_j e^{-\eta x_j}\right)e^{-i\omega x_j}=\frac{1}{i\kappa},$$

which can be re-arranged to get the DTFT form

$$\sum_{j=-\infty}^{\infty}\left(H_j e^{-\eta jh}\right)e^{-ij\theta}=\frac{e^{i(1-\alpha)\kappa h}}{i\kappa h}.$$

Applying inverse DTFT, we get

$$H_j=\frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{e^{i(1-\alpha)\kappa h}}{i\kappa h}e^{ij\theta+\eta jh}d\theta=H_\alpha(x_j)+\mathrm{Im}\left(\frac{\Gamma(0,-\eta x_j+i\frac{\pi x_j}{h})}{\pi}\right),$$

where $\mathrm{Im}(\cdot)$ means taking the imaginary part, and

$$\Gamma(\nu,z)=\int_z^\infty t^{\nu-1}e^{-t}dt$$

is the incomplete Gamma function. Therefore, we see that the infinite smoothing modifications are

$$c_{H,j}^{[\infty]}=\mathrm{Im}\left(\frac{\Gamma(0,-\eta x_j+i\frac{\pi x_j}{h})}{\pi}\right).$$

For the ramp function, we similarly get

$$C_j=\frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{e^{i(1-\alpha)\kappa h}}{-\kappa^2 h}e^{ij\theta+\eta jh}d\theta=x_j\left(H_\alpha(x_j)-\mathrm{Im}\left(\frac{\Gamma(-1,-\eta x_j+i\frac{\pi x_j}{h})}{\pi}\right)\right).$$

Therefore, we see that

$$c_{C,j}^{[\infty]}=-x_j\mathrm{Im}\left(\frac{\Gamma(-1,-\eta x_j+i\frac{\pi x_j}{h})}{\pi}\right).$$

For the quadratic ramp function, we have

$$Q_j=\frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{e^{i(1-\alpha)\kappa h}}{-i\kappa^3 h}e^{ij\theta+\eta jh}d\theta=\frac{x_j^2}{2}\left(H_\alpha(x_j)+\mathrm{Im}\left(\frac{\Gamma(-2,-\eta x_j+i\frac{\pi x_j}{h})}{\pi}\right)\right),$$

and

$$c_{Q,j}^{[\infty]}=\frac{x_j^2}{2}\mathrm{Im}\left(\frac{\Gamma(-2,-\eta x_j+i\frac{\pi x_j}{h})}{\pi}\right).$$

We present the formulas and leave the study of the results to readers who are pursuing even higher order methods.

4.5 Smoothing modifications on nonuniform grids

When the space discretization is nonuniform, the smoothing modification formulas derived above requires adjustment. Suppose that the nonuniform discretization comes from a smooth, strictly monotonic mapping $\psi : \mathbb{R} \to \mathbb{R}$ in the parameter space ξ, such that $x_j = \psi(\xi_j)$, where ξ_j are uniform discretization points in the parameter space with grid size $h_\xi = \xi_{j+1} - \xi_j$. Without loss of generality, again assume that the nonsmooth point in the nonuniform grid is at $x_K = 0$ and $\xi_K \equiv \phi(x_K)$ is the nonsmooth point in the uniform grid, where we define $\phi \equiv \psi^{-1}$. The nonsmooth point alignment value changes from α in the nonuniform grid to $\alpha_\xi \equiv \frac{\xi_K - \xi_{-1}}{\xi_0 - \xi_{-1}} = \frac{\phi(x_K) - \phi(x_{-1})}{\phi(x_0) - \phi(x_{-1})}$ in the uniform grid. In this work, we only consider the nonuniform discretization that comes from a smooth mapping.

Since solving the original problem (3.1) in x-space is equivalent to solving the equation with variable transformation $x = \phi^{-1}(\xi) = \psi(\xi)$ in ξ space, rather than calculating the smoothing modifications on nonuniform grid, it is easier to simply calculate the smoothing modifications on the transformed initial conditions on a uniform grid under variable transformation.

For the Dirac delta initial condition (4.16), We have

$$\delta_\alpha(x_j) = \frac{h_\xi}{h_0} \cdot \begin{cases} \frac{1}{h_\xi}, & j = -1 \\ 0, & \text{else} \end{cases} = \left(\frac{h_\xi}{h_0}\right) \cdot \delta_{\alpha_\xi}(\xi_j).$$

Therefore, the ℓth-order smoothing modification for the delta initial condition on the nonuniform grid $\{x_j\}$ is

$$\delta_\alpha^{[\ell]}(x_j) = \delta_\alpha(x_j) + \frac{h_\xi}{h_0} c_{\delta,j}^{[\ell]}(\alpha_\xi, h_\xi)$$

For the Heaviside initial condition, the discretization (4.4) stays the same under variable transformation. Hence, the ℓth-order smoothing modifications under grid mapping are simply

$$H_\alpha^{[\ell]}(x_j) = H_\alpha(x_j) + c_{H,j}^{[\ell]}(\alpha_\xi).$$

The ramp initial condition (4.5) under variable transformation $x_j = \psi(\xi_j)$ becomes

$$C_\alpha(x_j) = \begin{cases} \psi(\xi_j), & j \geq 0, \\ 0, & \text{else.} \end{cases}$$

We have

$$\begin{aligned} \psi(\xi) &= \psi(\xi_K) + \psi'(\xi_K)(\xi - \xi_K) + \frac{\psi''(\xi_K)}{2}(\xi - \xi_K)^2 + \frac{\psi'''(\xi_K)}{6}(\xi - \xi_K)^3 + \dots \\ &= \psi'(\xi_K)(\xi - \xi_K) + \frac{\psi''(\xi_K)}{2}(\xi - \xi_K)^2 + \mathcal{O}((\xi - \xi_K)^3), \end{aligned} \tag{4.17}$$

where we have used the fact that $\psi(\xi_K) = x_K = 0$. Of the terms in (4.17), the nonsmoothness arising from the constant, linear and quadratic terms need modifications. Therefore, the ℓth-order

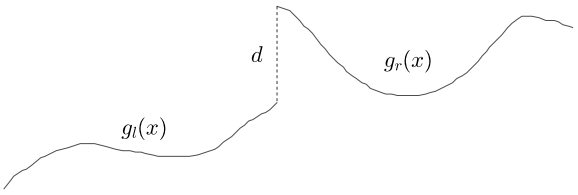


Figure 4.2: An example function with general nonsmoothness

smoothing modifications are

$$C_\alpha^{[\ell]}(x_j) = C_\alpha(x_j) + \psi'(\xi_K)c_{C,j}^{[\ell]}(\alpha_\xi, h_\xi) + \psi''(\xi_K)c_{Q,j}^{[\ell]}(\alpha_\xi, h_\xi).$$

Similarly, for the Q function, we have

$$\begin{aligned}\frac{1}{2}x^2 = \frac{1}{2}\psi(\xi)^2 &= \frac{1}{2}\left(\psi'(\xi_K)(\xi-\xi_K) + \frac{\psi''(\xi_K)}{2}(\xi-\xi_K)^2 + \frac{\psi'''(\xi_K)}{6}(\xi-\xi_K)^3 + \dots\right)^2 \\ &= \frac{1}{2}\psi'(\xi_K)^2(\xi-\xi_K)^2 + \mathcal{O}((\xi-\xi_K)^3).\end{aligned}$$

Therefore, the ℓth-order smoothing modifications are

$$Q_\alpha^{[\ell]}(x_j) = Q_\alpha(x_j) + \psi'(\xi_K)^2 c_{Q,j}^{[\ell]}(\alpha_\xi, h_\xi).$$

Note that when $x_K \neq 0$, $C_\alpha(x_j) = \psi(\xi_j) - \psi(\xi_K)$, and $Q_\alpha(x_j) = \frac{1}{2}(\psi(\xi_j) - \psi(\xi_K))^2$, for $j \geq 0$. Hence, the above derivations still hold.

4.6 Smoothing modifications to general nonsmoothness

The above analysis can also be extended to any general nonsmoothness when the initial conditions are piecewise analytic aside from the nonsmooth points. Without loss of generality, suppose the nonsmooth point is at $x_K = 0$. Consider the initial condition that has nonsmoothness as shown in Figure 4.2, where the amount of jump discontinuity is of size d, and $g_l(x), g_r(x)$ on the left, right of the nonsmooth point are both analytic with smooth expansions on both sides of x_K. Applying Taylor expansion to both $g_l(x)$ and $g_r(x)$ around x_K, we get

$$\begin{aligned}g_l(x) &= g_l(x_K) + g_l'(x_K)x + \frac{g_l''(x_K)}{2}x^2 + \mathcal{O}(x^3), \\ g_r(x) &= g_r(x_K) + g_r'(x_K)x + \frac{g_r''(x_K)}{2}x^2 + \mathcal{O}(x^3).\end{aligned}$$

The jump discontinuity between $g_r(x_K)$ and $g_l(x_K)$ is obvious. The discontinuity between $g_r'(x_K)x$ and $g_l'(x_K)x$ can be expressed as

$$g_l'(x_K)x + (g_r'(x_K) - g_l'(x_K))\max\{x, 0\}.$$

Similarly, the discontinuity between $\frac{g_r''(x_K)}{2}x^2$ and $\frac{g_l''(x_K)}{2}x^2$ can be expressed as

$$\frac{g_l''(x_K)}{2}x^2 + (g_r''(x_K) - g_l''(x_K))Q(x).$$

Therefore, the ℓth-order smoothing modifications of the general nonsmothness on a uniform grid are

$$g_\alpha^{[\ell]}(x_j) = g(x_j) + (g_r(x_K) - g_l(x_K))c_{H,j}^{[\ell]}(\alpha) + (g_r'(x_K) - g_l'(x_K))c_{C,j}^{[\ell]}(\alpha) + (g_r''(x_K) - g_l''(x_K))c_{Q,j}^{[\ell]}(\alpha),$$

When the space grid is generated from a smooth mapping $x = \psi(\xi)$, since $g(x) = g(\psi(\xi))$, the smoothing modifications are simply

$$\begin{aligned} g_\alpha^{[\ell]}(x_j) = g(x_j) &+ (g_r(x_K) - g_l(x_K))c_{H,j}^{[\ell]}(\alpha_\xi) + (g_r'(x_K)\psi'(\xi_K) - g_l'(x_K)\psi'(\xi_K))c_{C,j}^{[\ell]}(\alpha_\xi, h_\xi) \\ &+ (g_r''(x_K)\psi'(\xi_K)^2 + g_r'(x_K)\psi''(\xi_K) - g_l''(x_K)\psi'(\xi_K)^2 - g_l'(x_K)\psi''(\xi_K))c_{Q,j}^{[\ell]}(\alpha_\xi, h_\xi), \end{aligned} \tag{4.18}$$

where $x_K = \psi(\xi_K)$ is the nonsmooth point on the nonuniform grid. It is apparent that (4.18) includes the case of uniform grids when $\phi(x) = x$.

Chapter 5

Numerical results

With the analysis and algorithms in place, we present numerical results in this chapter to demonstrate that we indeed achieve high-order convergence and hence numerically verify the correctness of our analysis. Although this dissertation is concerned more with option pricing applications, we develop our algorithms with generic parabolic PDEs in mind. These algorithms can be equally applied to other areas of interest. We show numerical results for solving simple parabolic PDEs, typical option pricing problems, as well as other more general parabolic PDEs.

5.1 Applications to free boundary problems and American option pricing

In this section, we demonstrate the effectiveness of our high-order deferred correction algorithms for solving free boundary problems with several examples. We begin with a simple one-dimensional elliptic obstacle problem in Section 5.1.1. We then consider a time-dependent problem in Section 5.1.2. In this problem, the free boundary position $x_f(t)$ has first derivative singularity at $t = 0$, but the solution itself is smooth everywhere. For both problems, we show that the convergence rate at each solve phase in Algorithms 1 and 2, respectively, is exactly as predicted. Finally, in Section 5.1.3, we apply our Algorithm 2 with some additional considerations to the American option pricing problem. For this problem, both the free boundary and the solution itself have first derivative singularity at $t = 0$. The numerical results show the expected convergence rate at each solve phase. [1]

5.1.1 An elliptic obstacle problem

Consider the boundary value problem defined in the LCP form

$$\begin{aligned} -f'' + f + 1 &\geq 0, \\ f - f^* &\geq 0, \\ (-f'' + f + 1 = 0) \vee (f - f^* &= 0), \end{aligned} \tag{5.1}$$

[1] In the figures of this section, we denote M by N_x, and N by N_t.

on the domain $x \in [-1, 1]$, where $f^*(x) = x$, with boundary conditions

$$f(-1) = -1, \ f(1) = e - 1.$$

The exact solution to this problem is the piecewise smooth function

$$f(x) = \begin{cases} e^x - 1, & 0 < x \leq 1, \\ x, & -1 \leq x \leq 0. \end{cases}$$

It is obvious that the solution is smooth on both $[-1, 0]$ and $[0, 1]$, separately. At the point $x = 0$, the solution satisfies the value matching and smooth pasting conditions, that is,

$$\lim_{x\to 0^-} f(x) = \lim_{x\to 0^+} f(x) = f(0) = 0, \text{ and } \lim_{x\to 0^-} f'(x) = \lim_{x\to 0^+} f'(x) = f'(0) = 1.$$

However, the solution $f(x)$ has a discontinuous second derivative at $x = 0$, which means $f(x) \in C^1 \backslash C^2$. To apply Algorithm 1, we write (5.1) in the penalty form

$$-f'' + f + 1 - \rho \max(f^* - f, 0) = 0, \tag{5.2}$$

with Dirichlet boundary conditions at $x = -1$ and 1, where ρ is a penalty constant, taken to be $\rho = 1 \times 10^{12}$ in the numerical experiments.

We use this example to demonstrate that the four solve phases with deferred corrections in Algorithm 1 improve the convergence rates as expected. In Table 5.1, we can see that, away from the free boundary $x = 0$, the convergence orders at the first, second, third and fourth solves are 2, 3, 4 and 5, respectively. The free boundary approximation also follows the same successive increase of convergence order, as shown in Table 5.2. To demonstrate the computational efficiency of our algorithm, we have plotted the log-log graph of the solution errors versus the computational complexity, represented by grid size in space multiplied by the total number of penalty iterations, as shown in Figure 5.1. Note that in the first solve phase with no corrections, we only use a rough initial guess of a constant function $f = 1$, even though a better initial guess could be chosen. As a result, the first solve phase requires several penalty iterations to converge and takes up the major computational cost of the algorithm. In the second to fourth solve phases, only a single iteration is required for each solve phase, because the solution of the previous solve phase provides a good initial guess for Newton's method.

Remark 5.1.1. *Note that in both Tables 5.1 and 5.2, we see that fifth-order convergence is obtained after applying three corrections. We think the reason is that, even though we are using a fourth-order method, the error from the nonsmoothness of the solution at the free boundary has been reduced to $\mathcal{O}(h^5)$ with a large constant coefficient. We conjecture that this error is larger than the $\mathcal{O}(h^4)$ error arising from the fourth-order finite difference scheme, thus the convergence appears to be fifth-order. Furthermore, we remark that, if the two near-boundary points S_1 and S_{M-1} are discretized by centered second-order finite differences, the fifth-order convergence at the final solve phase is not observed anymore, and we see fourth-order convergence as expected.*

N	$x = 0.2$							
	1st solve (no correction)				2nd solve (one correction)			
	niters	value	error	conv	niters	value	error	conv
30	7	0.221530668	1.28e-04	-	8	0.221431233	2.85e-05	-
60	12	0.221434987	3.22e-05	1.99	13	0.221396803	5.96e-06	2.26
120	23	0.221410846	8.09e-06	1.99	24	0.221401484	1.27e-06	2.22
240	44	0.221404784	2.03e-06	2.00	45	0.221402568	1.90e-07	2.75
480	86	0.221403265	5.07e-07	2.00	87	0.221402733	2.53e-08	2.91
N	3rd solve (two corrections)				4th solve (three corrections)			
	niters	value	error	conv	niters	value	error	conv
30	9	0.221415708	1.29e-05	-	10	0.221400689	2.07e-06	-
60	14	0.221403511	7.53e-07	4.10	15	0.221402701	5.68e-08	5.19
120	25	0.221402801	4.30e-08	4.13	26	0.221402757	1.62e-09	5.13
240	46	0.221402760	2.29e-09	4.23	47	0.221402758	4.77e-11	5.09
480	88	0.221402758	9.83e-11	4.54	89	0.221402758	1.06e-12	5.50

Table 5.1: Convergence results of solutions at point $x = 0.2$ for each solve phase in Algorithm 1, when solving the penalized PDE (5.2) of a one-dimensional free boundary obstacle problem with free boundary at $x = 0$. Uniform grid spacing is used. Note that "niters" for the second to fourth solve includes the total number of iterations from all previous solve phases.

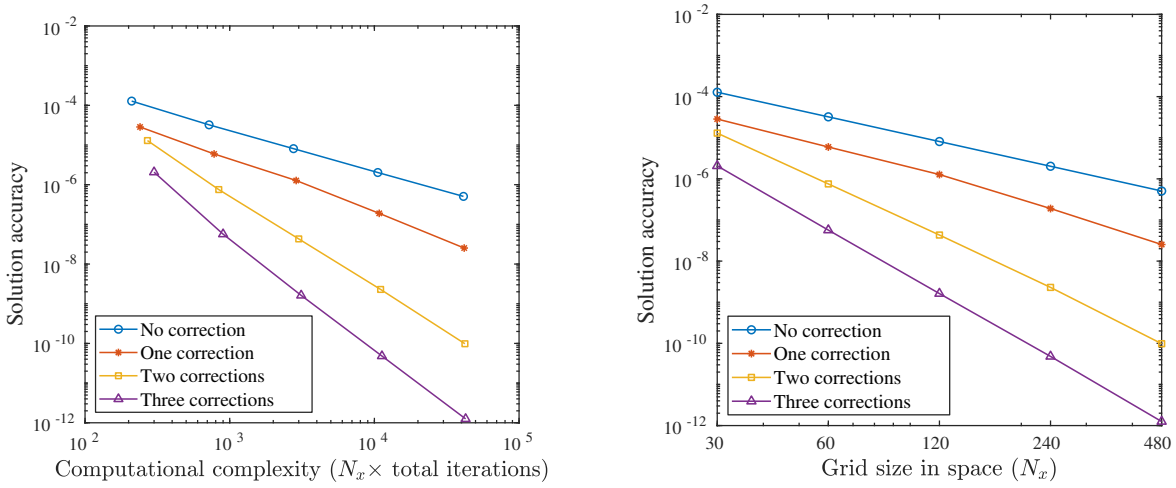
N	1st solve		2nd solve		3rd solve		4th solve	
	error	conv	error	conv	error	conv	error	conv
30	1.82e-02	-	2.20e-03	-	2.20e-04	-	1.87e-05	-
60	4.69e-03	1.96	2.51e-04	3.13	1.15e-05	4.26	4.12e-07	5.50
120	1.19e-03	1.98	2.84e-05	3.15	5.85e-07	4.30	6.55e-09	5.97
240	2.97e-04	2.00	2.93e-06	3.28	2.58e-08	4.50	8.38e-11	6.29
480	7.33e-05	2.02	2.17e-07	3.75	1.02e-09	4.67	1.03e-11	3.02

Table 5.2: Convergence results of the free boundary approximation for each solve phase in Algorithm 1, when solving the penalized PDE (5.2) of a one-dimensional free boundary obstacle problem with the exact free boundary at $x = 0$. Uniform grid spacing is used.

5.1.2 A simple moving boundary problem

In this second example, we introduce the time variable and consider a time-dependent free boundary problem. Consider the LCP

$$\begin{aligned} f_t - \frac{1}{2\sqrt{t}} f'' + \frac{1}{2\sqrt{t}} &\geq 0, \\ f - f^* &\geq 0, \\ \left(f_t - \frac{1}{2\sqrt{t}} f'' + \frac{1}{2\sqrt{t}} = 0 \right) &\vee (f - f^* = 0), \end{aligned} \tag{5.3}$$



(a) (b)

Figure 5.1: Log-log plot of solution error at point $x = 0.2$ versus computational complexity (a), and grid size in space (b), using results of Table 5.1, when solving the penalized PDE (5.2) of a one-dimensional free boundary obstacle problem. The computational complexity is represented by the grid size times the total number of penalty iterations.

on the domain $(t, x) \in [0, 0.5] \times [-2, 2]$, where $f^*(t, x) = x$. The solution satisfies the Dirichlet boundary conditions

$$f(t, -2) = -2, \; f(t, 2) = e^{2+\sqrt{t}} - \sqrt{t} - 1,$$

and the initial condition

$$f(0, x) = \begin{cases} e^x - 1, & 0 \le x \le 2, \\ x, & -2 \le x < 0. \end{cases}$$

The exact solution to (5.3) is

$$f(t, x) = \begin{cases} e^{x+\sqrt{t}} - \sqrt{t} - 1, & x_f(t) \le x \le 2, \\ x, & -2 \le x < x_f(t), \end{cases}$$

where $x_f(t)$ is the moving free boundary

$$x_f(t) = -\sqrt{t}.$$

The value matching and smooth pasting conditions at the free boundary $x = x_f(t)$ follow naturally. Again, we see that $f(\cdot, x) \in C^1 \backslash C^2$ on $[-2, 2]$, but it is smooth on $[-2, x_f(t)]$ and $[x_f(t), 2]$, separately. To apply Algorithm 2, we write (5.3) in penalty form

$$f_t - \frac{1}{2\sqrt{t}} f'' + \frac{1}{2\sqrt{t}} - \rho \max(f^* - f, 0) = 0, \tag{5.4}$$

where ρ is a penalty constant, taken to be $\rho = 1 \times 10^8$ in the numerical experiments.

Since the free boundary $x_f(t) = -\sqrt{t}$, its location changes rapidly near time $t = 0$. This will cause a problem in the BDF4 time-stepping scheme because many grid points will cross the free boundary in the initial time steps (see Section 2.2.4). Hence, BDF4 degenerates to only first-order convergence due to piecewise smoothness in the solution across the free boundary. To avoid this situation, we perform a time-variable transformation $t = \tau^2$ so that the free boundary changes more slowly, and fewer points will cross the free boundary in the initial time steps. Although this does not completely solve the problem, it is accurate enough for the algorithm to achieve high-order convergence, as shown in the numerical results.

We remark that for this problem, where the free boundary is known to change rapidly near $t = 0$, the $t = \tau^2$ transformation is effective. For problems for which we do not know when the free boundary changes rapidly, additional knowledge about the free boundary behavior needs to be used in order to pick an appropriate time transformation, or we can apply an adaptive time-step selector, which is left for future research.

To start BDF4, we use the exact solutions for the first three time steps. As described in Algorithm 2, the initial guess for the first solve of each time step after the first three time steps comes from the solution of the previous time step, and the initial guess for each subsequent solve in the same time step comes from the previous solve phase. For this problem, we simply use a uniform grid in space.

In Table 5.3, we record the convergence results at $x = -0.37$ and at $x = 0$. The point $x = -0.37$ is slightly to the right of the first grid point right of the final-time free boundary location on the coarsest grid $N_x = 20$. The point $x = 0$ is the initial free boundary location. We see that the solutions at both points gain the expected order of convergence after each correction. To demonstrate the computational efficiency of our algorithm, we plot the log-log graph of the solution errors versus the computational complexity represented by the grid size in space multiplied by the total number of penalty iterations, as shown in Figure 5.2. It is clear that except for very mild accuracy, the four-correction scheme outperforms the rest.

Remark 5.1.2. *Note that the solutions after solving with corrections have larger errors on the coarsest grid $N_x = 20$. This is due to large extrapolation errors of free boundary and derivatives approximations when the space step size near the free boundary is large, which occurs on a uniform coarse grid. This can be avoided by applying grid stretching around the free boundary.*

5.1.3 American option pricing

We use Algorithm 2 to solve the American put option pricing problem. For convenience of discussion, we repeat the penalty formulation of the problem

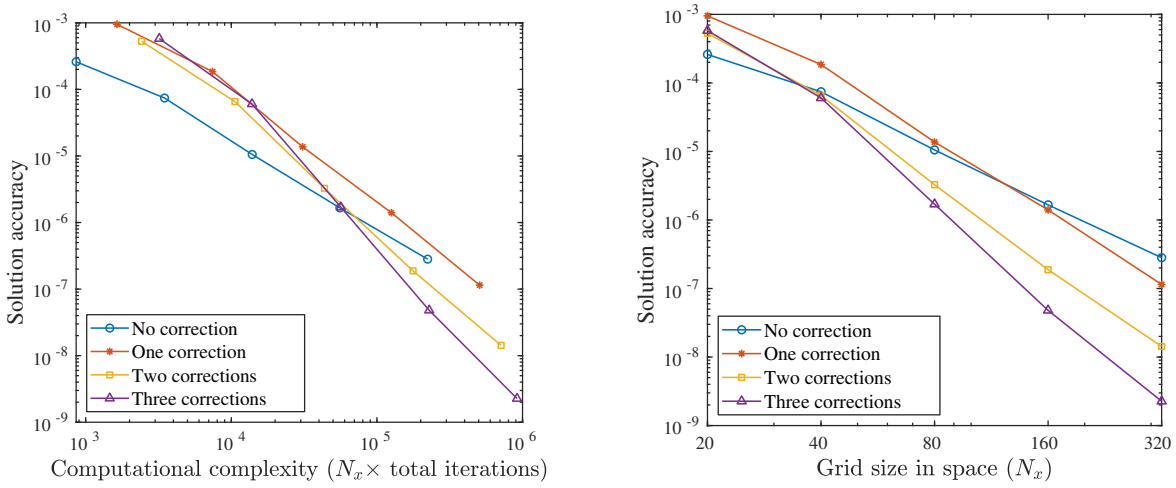
$$\partial_t V = \mathcal{L}_{BS} V + \rho \max\{V^* - V, 0\},$$

where $V^*(S) = \max\{K - S, 0\}$ is the payoff of the American put option struck at K. In the numerical experiments, the penalty constant ρ is chosen to be $\rho = 1 \times 10^8$. The initial condition is

$$V(0, S) = V^*(S).$$

(M,N)	$x=-0.37$							
	1st solve (no correction)				2nd solve (one correction)			
	niters	value	error	conv	niters	value	error	conv
(20,40)	43	-0.307227	1.04e-03	-	82	-0.304930	1.26e-03	-
(40,80)	87	-0.306241	8.16e-06	7.00	186	-0.306075	1.47e-04	3.10
(80,160)	174	-0.306227	9.13e-06	-0.16	386	-0.306205	1.33e-05	3.46
(160,320)	348	-0.306220	1.94e-06	2.24	786	-0.306216	1.67e-06	3.00
(320,640)	697	-0.306218	3.19e-07	2.60	1586	-0.306218	1.44e-07	3.53
(M,N)	3rd solve (two corrections)				4th solve (three corrections)			
	niters	value	error	conv	niters	value	error	conv
(20,40)	121	-0.306334	1.61e-04	-	160	-0.305883	3.03e-04	-
(40,80)	265	-0.306147	7.44e-05	1.11	347	-0.306163	5.90e-05	2.36
(80,160)	545	-0.306215	3.35e-06	4.47	707	-0.306216	1.77e-06	5.06
(160,320)	1106	-0.306218	2.20e-07	3.93	1429	-0.306218	5.78e-08	4.94
(320,640)	2227	-0.306218	1.12e-08	4.29	2866	-0.306218	2.79e-09	4.37
(M,N)	$x=0$							
	1st solve (no correction)				2nd solve (one correction)			
	niters	value	error	conv	niters	value	error	conv
(20,40)	43	0.320748	2.60e-04	-	82	0.321960	9.52e-04	-
(40,80)	87	0.320934	7.43e-05	1.81	186	0.321194	1.85e-04	2.36
(80,160)	174	0.320998	1.05e-05	2.82	386	0.321022	1.37e-05	3.76
(160,320)	348	0.321007	1.66e-06	2.66	786	0.321010	1.40e-06	3.29
(320,640)	697	0.321008	2.82e-07	2.56	1586	0.321008	1.14e-07	3.62
(M,N)	3rd solve (two corrections)				4th solve (three corrections)			
	niters	value	error	conv	niters	value	error	conv
(20,40)	121	0.321537	5.29e-04	-	160	0.321588	5.80e-04	-
(40,80)	265	0.321074	6.56e-05	3.01	347	0.321069	6.04e-05	3.26
(80,160)	545	0.321011	3.27e-06	4.33	707	0.321010	1.70e-06	5.15
(160,320)	1106	0.321008	1.88e-07	4.12	1429	0.321008	4.80e-08	5.15
(320,640)	2227	0.321008	1.43e-08	3.72	2866	0.321008	2.27e-09	4.41

Table 5.3: Convergence results of solutions at points $x=-0.37$ and $x=0$ for each solve phase in Algorithm 2, when solving the penalized PDE (5.4) of a moving boundary problem with the exact moving boundary $x_f(t)=-\sqrt{t}$. Note that "niters" for the second to fourth solve includes the total number of iterations from all previous solve phases.



(a) (b)

Figure 5.2: Log-log plot of solution errors at point $x = 0$ versus computational complexity (a), and grid size in space (b), using results of Table 5.3, when solving the penalized PDE (5.4) of a moving boundary problem with the exact moving boundary $x_f(t) = -\sqrt{t}$. The computational complexity is represented by the grid size times the total number of penalty iterations.

The left boundary of the domain is $S_{\min} = 0$ and we truncate the right end of the domain at $S = S_{\max}$ with Dirichlet boundary conditions $V(t, 0) = K$ and $V(t, S_{\max}) = 0$. Because our focus in Algorithm 2 is to deal with the space discontinuity at the free boundary, to reduce other complications such as the solution derivatives at the free boundary blowing up at $t = 0$ - see (A.11) and problem (iii) in Chapter 1 - we compute the difference between an American and a European option; this is referred to as the singularity-separating method in [65]. A European put option value V^E with the same volatility σ, bank interest r, dividend q, and strike price K satisfies the Black-Scholes equation $\partial_t V^E = \mathcal{L}_{BS} V^E$, with the initial condition $V^E(0, S) = \max\{K - S, 0\}$, and is given by a known explicit formula. The difference of the solutions $V^{\text{diff}} = V - V^E$ has a zero initial condition. Therefore, instead of solving for the original American option price, we solve for V^{diff}, which satisfies the equation

$$\partial_t V^{\text{diff}} = \mathcal{L}_{BS} V^{\text{diff}} + \rho \max\{(V^* - V^E) - V^{\text{diff}}, 0\},$$

with a zero initial condition, and then add the European option price back to obtain the final American option price.

Here we remark that the boundary condition $V(t, S_{\max}) = 0$ is only approximate for both European and American options. For a specific $S_{\max}$, the European and American option values are both positive and different from each other. An appropriately chosen $S_{\max}$ would make this difference negligible compared to the discretization error.

We apply Algorithm 2 developed in Chapter 2 together with the 2RK3-BDF3-BDF4 time-stepping scheme proposed in Chapter 3 to solve for the solution. We test the algorithm on two example problems with different volatilities, $\sigma = 0.2$ and $\sigma = 0.8$ as the examples in [21]. Both examples have the other parameters the same: zero dividend payment, interest rate $r = 0.1$, strike

price $K = 100$ and expiration time $T = 0.25$. We truncate the infinite domain at $S_{\max} = 10K = 1000$ for the problem with smaller volatility $\sigma = 0.2$, and at $S_{\max} = 13K = 1300$ for the larger volatility $\sigma = 0.8$. As it turns out, a larger $S_{\max}$ is not only necessary for the accuracy of solution with a larger volatility, it is also important for observing the convergence results of our algorithm. For the space stretching, we propose the stretching function

$$\xi(S) = \left(S - \frac{\sqrt{\pi}}{2} \frac{1-\beta}{\beta} \alpha \text{erfc}\left(\frac{S-K}{\alpha} \right) \right) C_1 + C_2,$$

to stretch the space grid, where

$$C_1 = 1/\left[(S_{\max} - S_{\min}) - \frac{\sqrt{\pi}}{2} \frac{1-\beta}{\beta} \alpha \left(\text{erfc}\left(\frac{S_{\max}-K}{\alpha} \right) - \text{erfc}\left(\frac{S_{\min}-K}{\alpha} \right) \right) \right],$$
$$C_2 = \left[\frac{\sqrt{\pi}}{2} \frac{1-\beta}{\beta} \alpha \text{erfc}\left(\frac{S_{\min}-K}{\alpha} \right) - S_{\min} \right] C_1,$$

and where α and β are parameters controlling the density of the stretching. In the numerical experiments, the parameters α and β are chosen to be $\alpha = 125/6 \approx 20.83$ and $\beta = 1/20$ for $\sigma = 0.2$, and $\alpha = 65, \beta = 1/8$ for $\sigma = 0.8$. The stretching function $\xi(S)$ above introduces grid points with density $1/\beta$, on a region of width 6α centered around the point K, while maintaining the density of 1 elsewhere. Our choice of parameter α can be motivated by the fact that the moving boundary moves more with a higher volatility than with a lower one, and thus the stretching must cover a longer region when the volatility is high. More specifically, for $\sigma = 0.2$, the range of the optimal exercise boundary movement is approximately 10, starting from $S = 100$ at $t = 0$, to $S = 89.7$ at expiry $t = 0.25$. Therefore, the moving free boundary is within $\frac{10}{3\alpha} \approx \frac{1}{6}$ of the length of the stretched region, away from the stretching center, during the whole time period. For $\sigma = 0.8$, the range of the optimal exercise boundary movement is around 48, starting from $S = 100$ at $t = 0$, to $S = 51.8$ at expiry $t = 0.25$. Therefore, the moving free boundary is within $\frac{51.8}{3\alpha} \approx \frac{1}{4}$ of the length of the stretched region, away from the stretching center, during the whole time period.

In addition, since the solution of the American option price has singular second and higher space derivatives at the strike at expiry, meaning that the solution is not smooth, we apply a couple of customizations to Algorithm 2. First, as in the previous example, we know that the free boundary changes rapidly near $t = 0$, so we use the time variable transformation $t = \tau^2$ for the American option example as well (for both $\sigma = 0.2$ and 0.8). Previous studies [21] indicated that, for the American option problem, adaptive time stepping is needed to maintain the order of convergence of the method used. Later, [45] suggested that the quadratic transformation in time acts equivalently to restore the order of convergence. We leave the study of general adaptive high-order time stepping for future work, and apply a simple $t = \tau^2$ transformation in this thesis. Second, we do not apply the correction scheme in Algorithm 2 for the first several time steps, as the correction scheme is heavily dependent on the smoothness of the solution. In the numerical experiments, the number of time steps in which the correction scheme is skipped is chosen to be $t_{skip} = 12$ for both $\sigma = 0.2$ and 0.8. Since we apply time and space stretching near the strike at expiry, the errors from the skipped corrections in the first few time steps are sufficiently small so as not to affect the high-order convergence.

The numerical results are given in Table 5.4. We can see clear convergence rate improvements and error reduction with corrections at each solve phase, with the smaller volatility exhibiting even faster error reduction. The second solve phase, corresponding to one correction, only slightly changes the error. The third- and fourth-solve phases, corresponding to two and three corrections, respectively, reduce the error significantly. The final results after the fourth solve phase exhibit a reduction of error by nearly 100 times compared to the no-correction phase.

To demonstrate the computational efficiency of our algorithm, we have shown in Figure 5.3 the solution accuracy versus the computational complexity represented by the grid size in space multiplied with the total number of penalty iterations. We can see that the three-correction algorithm is slightly more expensive if high accuracy is not the goal. However, when a high accuracy solution is desired, the three-correction algorithm is more efficient.

As a comparison, we also include the results from our previous publication [60], which uses a slightly different starting scheme for time-stepping. Instead of applying RK3 for the first two time steps, we replace the second RK3 step with a three-level fourth-order method [35] as in [60],

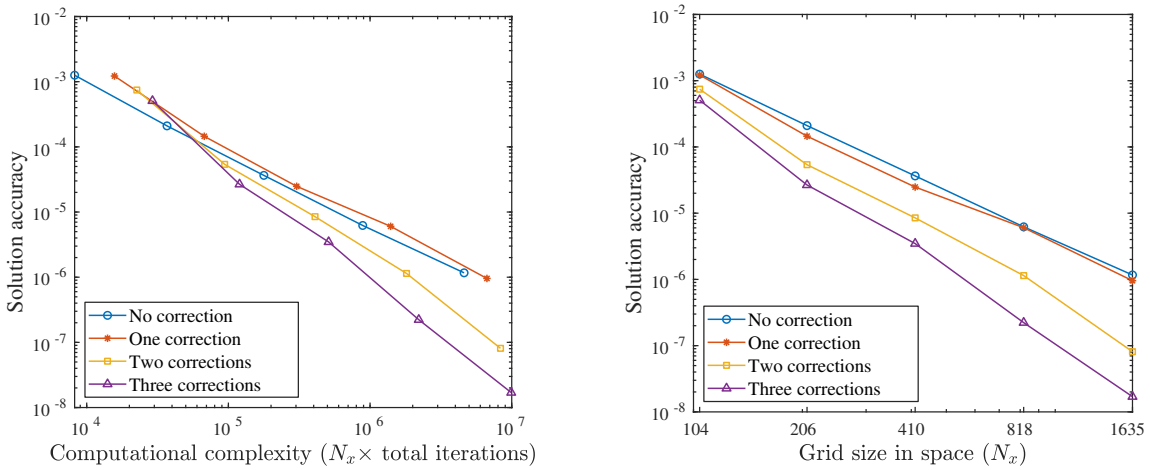
$$\left(\mathbf{I}-\frac{k}{3}\mathbf{L}\right)\tilde{\mathbf{V}}^{n+2}=\tilde{\mathbf{V}}^{n}+\frac{k}{3}\mathbf{L}\left(4\tilde{\mathbf{V}}^{n+1}+\tilde{\mathbf{V}}^{n}\right)+\frac{k}{3}(\mathbf{b}^{n+2}+4\mathbf{b}^{n+1}+\mathbf{b}^{n})+2\mathbf{q}(\tilde{\mathbf{V}}^{n+2}), \tag{5.5}$$

and the remaining time steps are the same as in Chapter 3. With the new starting scheme for time stepping, the convergence results applying Algorithm 2 are given in Table 5.5. We see that it gives very similar results as in Table 5.4. Changing the second step from RK3 to (5.5) decreases the error slightly. We conjecture that the slightly smaller error may be due to the implicit nature of (5.5), which has a similar damping property as BDF methods, thus reducing the error more effectively than explicit Runge-Kutta methods.

Remark 5.1.3. *For this example, we see that the second solve does not improve the convergence much. This is because we have applied enough stretching to reduce the leading-order error term due to the second-derivative jump in the first solve, even without correction. Moreover, we observe that our algorithm for* $\sigma = 0.2$ *performs better than for* $\sigma = 0.8$*. For a larger volatility* $\sigma = 0.8$*, the optimal exercise boundary moves more quickly and ranges over a larger part of the domain within the same time span. This has at least two negative effects: (a) We need to increase* $S_{\max}$ *to maintain good accuracy (see, for example, [30, 63]) and (b) We need to stretch the space grid over a larger interval. These two facts, in turn, result in (i) having larger space stepsizes all over the domain including the stretching area, (ii) the free boundary moving farther away from the stretching center (which is the initial free boundary), and (iii) the grid-crossing effect happening more frequently. The first two contribute to larger extrapolation errors, and the third introduces additional errors. One way to avoid large extrapolation errors is to implement a time-dependent grid stretching that follows the free boundary movement, see e.g. [40] where a predictor-corrector scheme is applied. We leave this to future work.*

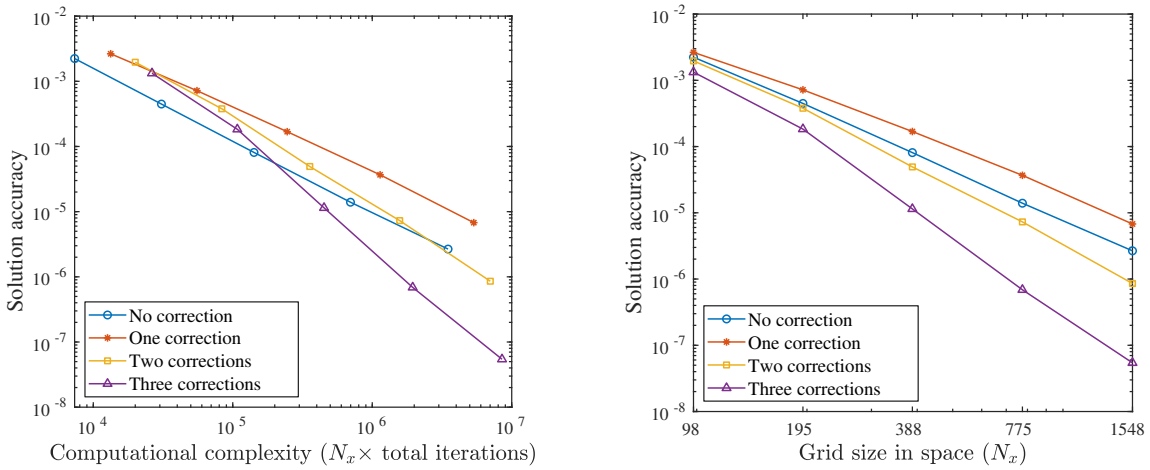
(M,N)	$\sigma = 0.2,\ T = 0.25$							
	1st solve (no correction)				2nd solve (one correction)			
	niters	value	error	conv	niters	value	error	conv
(53,30)	37	3.068596419	-	-	71	3.068709290	-	-
(104,60)	79	3.069852823	1.26e-03	-	151	3.069929517	1.22e-03	-
(206,120)	179	3.070062605	2.10e-04	2.58	328	3.070074722	1.45e-04	3.07
(410,240)	434	3.070099107	3.65e-05	2.52	740	3.070099511	2.48e-05	2.55
(818,480)	1085	3.070105325	6.22e-06	2.55	1708	3.070105542	6.03e-06	2.04
(1635,960)	2820	3.070106492	1.17e-06	2.41	4088	3.070106499	9.57e-07	2.66
(M,N)	3rd solve (two corrections)				4th solve (three corrections)			
	niters	value	error	conv	niters	value	error	conv
(53,30)	104	3.069298485	-	-	136	3.069568422	-	-
(104,60)	217	3.070043097	7.45e-04	-	280	3.070076401	5.08e-04	-
(206,120)	457	3.070097013	5.39e-05	3.79	582	3.070103025	2.66e-05	4.25
(410,240)	997	3.070105501	8.49e-06	2.67	1244	3.070106498	3.47e-06	2.94
(818,480)	2211	3.070106643	1.14e-06	2.89	2699	3.070106721	2.23e-07	3.96
(1635,960)	5096	3.070106724	8.10e-08	3.82	6067	3.070106738	1.70e-08	3.71
(M,N)	$\sigma = 0.8,\ T = 0.25$							
	1st solve (no correction)				2nd solve (one correction)			
	niters	value	error	conv	niters	value	error	conv
(50,30)	34	14.676098302	-	-	64	14.675303453	-	-
(98,60)	75	14.678332647	2.23e-03	-	136	14.677946806	2.64e-03	-
(195,120)	158	14.678780201	4.48e-04	2.32	284	14.678664815	7.18e-04	1.88
(388,240)	366	14.678861201	8.10e-05	2.47	633	14.678833483	1.69e-04	2.09
(775,480)	901	14.678875155	1.40e-05	2.54	1469	14.678870264	3.68e-05	2.20
(1548,960)	2258	14.678877821	2.67e-06	2.39	3453	14.678877046	6.78e-06	2.44
(M,N)	3rd solve (two corrections)				4th solve (three corrections)			
	niters	value	error	conv	niters	value	error	conv
(50,30)	94	14.676478652	-	-	126	14.677353497	-	-
(98,60)	204	14.678441994	1.96e-03	-	269	14.678682759	1.33e-03	-
(195,120)	428	14.678820822	3.79e-04	2.37	553	14.678866173	1.83e-04	2.86
(388,240)	919	14.678870113	4.93e-05	2.94	1163	14.678877616	1.14e-05	4.00
(775,480)	2030	14.678877401	7.29e-06	2.76	2516	14.678878305	6.89e-07	4.05
(1548,960)	4536	14.678878257	8.56e-07	3.09	5505	14.678878359	5.43e-08	3.66

Table 5.4: Convergence results of an American put option at $S = 100$, $T = 0.25$ with $K = 100$, $r = 0.1$, $q = 0$, for $\sigma = 0.2$ and $\sigma = 0.8$, and for each solve phase in Algorithm 2. Note that "niters" for the second to fourth solve includes the total number of iterations from all previous solve phases. The "error" columns are calculated by the difference between two successive grid resolutions.



(a) (b)

Figure 5.3: Log-log plot of solution changes at the strike point K at the final time T versus computational complexity (a), and grid size in space (b), using results of solving American option prices in Table 5.4 for $\sigma = 0.2$. The computational complexity is represented by the grid size in space times the total number of penalty iterations.



(a) (b)

Figure 5.4: Log-log plot of solution changes at the strike point K at the final time T versus computational complexity (a), and grid size in space (b), using results of solving American option prices in Table 5.4 for $\sigma = 0.8$. The computational complexity is represented by the grid size in space times the total number of penalty iterations.

(M,N)	$\sigma = 0.2,\ T = 0.25$							
	1st solve (no correction)				2nd solve (one correction)			
	niters	value	error	conv	niters	value	error	conv
(53,30)	37	3.068602382	-	-	71	3.068715191	-	-
(104,60)	79	3.069855016	1.25e-03	-	151	3.069931750	1.22e-03	-
(206,120)	179	3.070062874	2.08e-04	2.59	328	3.070075013	1.43e-04	3.09
(410,240)	434	3.070099140	3.63e-05	2.52	740	3.070099544	2.45e-05	2.55
(818,480)	1085	3.070105329	6.19e-06	2.55	1708	3.070105547	6.00e-06	2.03
(1635,960)	2820	3.070106489	1.16e-06	2.42	4088	3.070106496	9.49e-07	2.66
(M,N)	3rd solve (two corrections)				4th solve (three corrections)			
	niters	value	error	conv	niters	value	error	conv
(53,30)	104	3.069304376	-	-	136	3.069574295	-	-
(104,60)	217	3.070045347	7.41e-04	-	280	3.070078659	5.04e-04	-
(206,120)	457	3.070097280	5.19e-05	3.83	582	3.070103304	2.46e-05	4.36
(410,240)	997	3.070105534	8.25e-06	2.65	1244	3.070106531	3.23e-06	2.93
(818,480)	2211	3.070106647	1.11e-06	2.89	2699	3.070106725	1.94e-07	4.06
(1635,960)	5096	3.070106721	7.34e-08	3.92	6067	3.070106734	9.32e-09	4.38
(M,N)	$\sigma = 0.8,\ T = 0.25$							
	1st solve (no correction)				2nd solve (one correction)			
	niters	value	error	conv	niters	value	error	conv
(50,30)	34	14.676127404	-	-	64	14.675332474	-	-
(98,60)	75	14.678320134	2.19e-03	-	136	14.677934361	2.60e-03	-
(195,120)	158	14.678780547	4.60e-04	2.25	284	14.678665160	7.31e-04	1.83
(388,240)	366	14.678861193	8.06e-05	2.51	633	14.678833475	1.68e-04	2.12
(775,480)	902	14.678875150	1.40e-05	2.53	1471	14.678870259	3.68e-05	2.19
(1548,960)	2258	14.678877820	2.67e-06	2.39	3453	14.678877045	6.79e-06	2.44
(M,N)	3rd solve (two corrections)				4th solve (three corrections)			
	niters	value	error	conv	niters	value	error	conv
(50,30)	94	14.676507458	-	-	126	14.677382768	-	-
(98,60)	204	14.678429412	1.92e-03	-	269	14.678670286	1.29e-03	-
(195,120)	428	14.678821171	3.92e-04	2.29	553	14.678866522	1.96e-04	2.71
(388,240)	919	14.678870105	4.89e-05	3.00	1163	14.678877609	1.11e-05	4.15
(775,480)	2033	14.678877396	7.29e-06	2.75	2520	14.678878300	6.91e-07	4.00
(1548,960)	4536	14.678878257	8.61e-07	3.08	5505	14.678878359	5.89e-08	3.55

Table 5.5: Convergence results for solving the same American put options applying Algorithm 2 as in Table 5.4. Different from Table 5.4, instead of using RK3 in the second time step, we replace it with the fourth-order implicit scheme given by 5.5. Note that "niters" for the second to fourth solve includes the total number of iterations from all previous solve phases. The "error" columns are calculated by the difference between two successive grid resolutions.

5.2 Applications to parabolic PDEs and European option pricing

In this section, we consider the model parabolic PDE 3.1 and the applications to European options, and solve them by combining the time stepping scheme developed in Chapter 3 with the smoothing techniques in Chapters 3 and 4. We note that the convergence analysis in Chapter 3 for the model PDE 3.1 relies on Fourier analysis which assumes $x \in (-\infty, \infty)$, while the numerical results in this section are demonstrated on a truncated domain. The numerical experiments in this section indicate that the conclusions we obtained still hold on the truncated domain.

5.2.1 The model convection-diffusion equation

In this section, we provide numerical results for solving the model problem (3.1) with nonsmooth initial conditions (3.2), (3.3), (3.4), and (3.5), using the methods developed in Chapter 3. We consider a truncated space domain on $x \in [-4, 4]$, with exact Dirichlet boundary conditions.

In practice, it may be inconvenient to maintain the grid alignment value α to a fixed number. For this reason, we consider cases where the grid alignment changes for each refinement, by slightly shifting the nonsmooth point to $x = 0.123$ while keeping the space domain $x \in [-4, 4]$ unchanged. We apply the fourth-order convolution-type smoothed initial condition discretizations given in Table 3.1, for the Dirac delta, Heaviside and ramp initial conditions, respectively, with $a = 2, T = 1$. To also verify the correctness of our convergence analysis on the effect of smoothed and unsmoothed initial data, we show convergence results both with and without the smoothing of initial conditions. Tables 5.6, 5.7 and 5.8 show that directly applying the discrete initial conditions (3.13), (4.4) and (4.5) leads to low-order and inconsistent convergence, while with the smoothing modifications, we restore stable fourth-order convergence. We have also listed in the tables the convergence results of the solution derivatives. The results clearly show stable fourth-order accuracy.

To demonstrate the intrinsic high-frequency damping properties of BDF time stepping, we solve the model convection-diffusion (3.1) under the delta and Heaviside initial conditions with our 2RK3-BDF3-BDF4 method and with the Crank-Nicolson (CN) method, respectively, and compare the solutions. We apply the same fourth-order FD discretization in space, and the original initial condition discretizations given by (3.13), (4.4) without any smoothing modifications to make sure we are only looking at the effect of different time-stepping schemes. Figure 5.5 shows comparisons between the numerical solutions to the model problem with $a = 2$ and $T = 0.1$. We choose $h = 0.0211$, $d = \frac{k}{h} = 0.1185$ for solving the PDE with delta initial condition, and $h = 0.0123$, $d = \frac{k}{h} = 0.2033$ for the Heaviside initial condition. As seen in Figure 5.5, CN time stepping by itself fails to converge in L_∞, and generates oscillatory solutions. After replacing the first two steps of CN approximation by four half-timestep backward Euler time marching (CN-Rannacher), the oscillations disappear [25]. On the other hand, due to the high-frequency damping property of BDF4, we observe that no spurious oscillations occur in the solutions and solution derivatives with the 2RK3-BDF3-BDF4 method.

Finally, to show that our method can be applied to solve PDEs with more complicated nonsmooth initial conditions constructed from the three basic nonsmooth functions, in Table 5.9, we present convergence results for solving the model PDE with the bump initial condition (3.5). In this table, we list the maximum error across all gridpoints (as an approximation to the ∞-norm

N	α	v			v'		v''	
		value	error	conv	error	conv	error	conv
		w/o smoothing						
20	0.3075	0.1045846	8.12e-04	-	3.27e-04	-	1.37e-03	-
40	0.6150	0.1040104	2.33e-04	1.80	2.30e-04	0.50	5.67e-04	1.27
80	0.2300	0.1038231	4.62e-05	2.33	4.23e-05	2.45	1.11e-04	2.35
160	0.4600	0.1037930	1.61e-05	1.52	1.60e-05	1.41	4.01e-05	1.47
320	0.9200	0.1037781	1.18e-06	3.77	1.21e-06	3.72	2.98e-06	3.75
		w/ smoothing in Table 3.1						
20	0.3075	0.103747622	8.13e-05	-	4.28e-04	-	9.05e-04	-
40	0.6150	0.103775024	1.36e-06	5.90	2.67e-05	4.00	3.47e-05	4.70
80	0.2300	0.103776728	1.53e-07	3.15	1.69e-06	3.98	2.19e-06	3.99
160	0.4600	0.103776867	9.42e-09	4.02	1.06e-07	3.99	1.34e-07	4.03
320	0.9200	0.103776874	7.06e-10	3.74	6.60e-09	4.01	8.66e-09	3.96

Table 5.6: Convergence results at the nonsmooth point $x = 0.123$, $T = 1$, for solving the model problem (3.1) with the *Dirac delta initial condition*, taking $a = 2$. The grid alignment value α is different on each grid refinement level as given in the table, and the number of space intervals $M = N$.

of the error). The results clearly demonstrate fourth-order convergence of the solution, with slight degeneration in the solution derivatives.

5.2.2 European option pricing on uniform grids

In this section, we solve the Black-Scholes PDE in (1.7) for European option pricing. Although one can convert (1.7) to a constant-coefficients PDE, we solve the original Black-Scholes PDE (1.7) directly. The numerical results using the methods developed in Chapter 3 for the constant-coefficient model PDE (3.1) demonstrate the general applicability of our methods. We consider three types of European options: digital call, call and butterfly spread, given in Equations (1.11), (1.13) and (1.17), respectively, corresponding to the shifted Heaviside, ramp and bump initial conditions we discussed for the model PDE. The parameters we use in the numerical experiments are: strike $K = 100$, $\mathcal{B} = 19.75$, expiry time $T = 0.5$, interest rate $r = 2\%$, zero dividend. The volatility σ is either 0.2 or 0.8 as given in the tables and figures. The semi-infinite spatial domain is truncated to (S_a, S_b) with $S_a = 0$ and $S_b = 6K$, and exact Dirichlet conditions are applied.

Tables 5.10, 5.11 and 5.12 show the results of solving digital call, call and butterfly spread options, respectively, with variable α. We also list the convergence of the options' Δ and Γ at the single strike K for the digital call and call options, and at all the three kink points of the butterfly spread payoff function. In all experiments, we apply the 2RK3-BDF3-BDF4 time-stepping scheme, with fourth-order smoothings to the initial conditions as given in Table 3.1. Fourth-order convergence is obtained for the option prices and the calculated Δ and Γ.

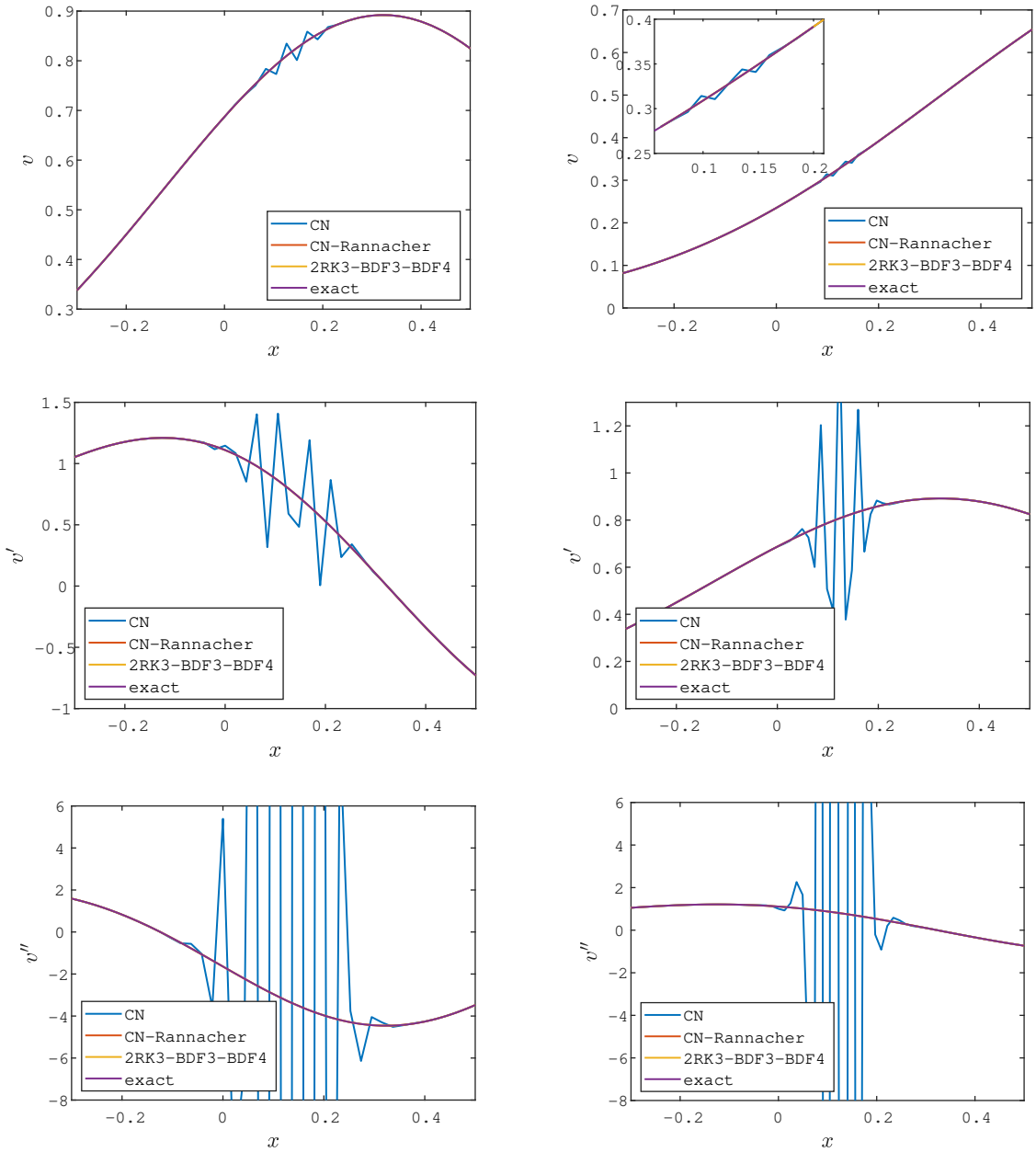
To demonstrate the intrinsic high-frequency damping properties of BDF time stepping, we compare the solutions for solving the digital call, call options using 2RK3-BDF3-BDF4 and CN time-stepping methods. We apply the same fourth-order FD discretization in space, and the original initial condition discretizations given by (14), (15) without any smoothing modifications to make

N	α	v			v'		v''	
		value	error	conv	error	conv	error	conv
		w/o smoothing						
20	0.3075	0.0700980	8.56e-03	-	8.26e-03	-	3.46e-03	-
40	0.6150	0.0808755	2.23e-03	1.94	2.31e-03	1.84	1.30e-03	1.41
80	0.2300	0.0758418	2.81e-03	-0.34	2.81e-03	-0.28	1.40e-03	-0.11
160	0.4600	0.0784314	2.18e-04	3.69	2.13e-04	3.72	9.84e-05	3.83
320	0.9200	0.0797423	1.09e-03	-2.32	1.09e-03	-2.36	5.43e-04	-2.47
		w/ smoothing in Table 3.1						
20	0.3075	0.078481631	1.73e-04	-	5.83e-05	-	4.11e-04	-
40	0.6150	0.078638802	1.10e-05	3.98	3.94e-06	3.89	2.59e-05	3.99
80	0.2300	0.078648921	6.90e-07	3.99	2.70e-07	3.87	1.63e-06	3.99
160	0.4600	0.078649561	4.33e-08	4.00	1.68e-08	4.01	1.02e-07	4.00
320	0.9200	0.078649601	2.71e-09	4.00	1.05e-09	4.00	6.40e-09	4.00

Table 5.7: Convergence results at the nonsmooth point $x = 0.123$, $T = 1$, for solving the model problem (3.1) with the *Heaviside initial condition*, taking $a = 2$. The grid alignment value α is different on each grid refinement level as given in the table, and the number of space intervals $M = N$.

N	α	v			v'		v''	
		value	error	conv	error	conv	error	conv
		w/o smoothing						
20	0.3075	0.0504901	2.38e-04	-	1.74e-04	-	1.50e-04	-
40	0.6150	0.0504040	1.50e-04	0.67	1.39e-04	0.33	6.34e-05	1.24
80	0.2300	0.0502581	3.56e-06	5.39	3.92e-06	5.15	3.13e-06	4.34
160	0.4600	0.0502651	1.05e-05	-1.57	1.05e-05	-1.43	5.30e-06	-0.76
320	0.9200	0.0502515	3.00e-06	1.81	3.01e-06	1.81	1.52e-06	1.80
		w/ smoothing in Table 3.1						
20	0.3075	0.050162772	9.04e-05	-	1.28e-04	-	5.99e-05	-
40	0.6150	0.050248754	5.58e-06	4.02	8.18e-06	3.96	1.56e-06	5.26
80	0.2300	0.050254186	3.49e-07	4.00	5.18e-07	3.98	4.87e-08	5.00
160	0.4600	0.050254519	2.18e-08	4.00	3.24e-08	4.00	1.85e-09	4.72
320	0.9200	0.050254540	1.36e-09	4.00	2.03e-09	4.00	6.76e-11	4.77

Table 5.8: Convergence results at the nonsmooth point $x = 0.123$, $T = 1$, for solving the model problem (3.1) with the *ramp initial condition*, taking $a = 2$. The grid alignment value α is different on each grid refinement level as given in the table, and the number of space intervals $M = N$.



Dirac delta Heaviside

Figure 5.5: Comparison of numerical solutions and the calculated derivatives around the non-smooth point from solving the model PDE (3.1) with $a = 2$ under Dirac delta and Heaviside initial conditions, using CN, CN-Rannacher and 2RK3-BDF3-BDF4 time stepping. We see that CN-Rannacher, 2RK3-BDF3-BDF4 methods are indistinguishable from the exact, while CN exhibits oscillations.

N	α	v		v'		v''	
		max error	conv	max error	conv	max error	conv
		w/o smoothing					
20	0.3075	4.10e-03	-	1.78e-03	-	3.24e-03	-
40	0.6150	1.48e-03	1.47	7.31e-04	1.28	5.21e-04	2.64
80	0.2300	2.03e-04	2.87	1.27e-04	2.52	2.78e-04	0.91
160	0.4600	1.02e-04	0.99	1.32e-04	-0.06	2.51e-04	0.15
320	0.9200	1.30e-05	2.97	2.41e-05	2.46	4.73e-05	2.41
		w/ smoothing obtained from Table 3.1					
20	0.3075	1.35e-03	-	9.93e-04	-	2.16e-03	-
40	0.6150	8.81e-05	3.93	1.06e-04	3.23	3.06e-04	2.81
80	0.2300	5.48e-06	4.01	9.70e-06	3.45	2.57e-05	3.58
160	0.4600	3.41e-07	4.01	8.12e-07	3.58	1.97e-06	3.70
320	0.9200	2.18e-08	3.96	6.94e-08	3.55	1.58e-07	3.64

Table 5.9: Convergence results for maximum error and first and second derivatives, when solving the model problem (3.1) with the *bump initial condition* of spread $\mathcal{B}$, taking $a = 2$, $T = 1$. There are three nonsmooth points at $K - \mathcal{B}$, K, and $K + \mathcal{B}$, with $K = 0.123, \mathcal{B} = 1.321$. The grid alignment value α is different on each grid refinement level as given in the table, and the number of space intervals $M = N$.

sure we are only looking at the effect of different time-stepping schemes. We choose $h = 0.5$ and $d = \frac{k}{h} = 0.01$ for both examples. The results are plotted in Figure 5.6. We see that no spurious oscillations occur in the solutions with 2RK3-BDF3-BDF4 time stepping as expected, due to the high-frequency damping properties of BDF methods.

5.2.3 European option pricing on nonuniform grids

In this example, we apply the time stepping scheme developed in Chapter 3 and the smoothing scheme developed in Chapter 4 to solve the same Black-Scholes PDE under the same butterfly spread payoff (1.17) as in Table 5.12, but on a nonuniform grid in space such that more points are concentrated around the three nonsmooth points, K_1, K_2 and K_3. We apply the nonuniform grid stretching

$$\xi = \phi(S) \equiv \frac{1}{3}\left(\frac{\sinh^{-1}(\rho(S-K_1)) - c_a}{c_b - c_a} + \frac{\sinh^{-1}(\rho(S-K_2)) - c_a}{c_b - c_a} + \frac{\sinh^{-1}(\rho(S-K_3)) - c_a}{c_b - c_a}\right), \tag{5.6}$$

where $c_a = \frac{1}{3}\sinh^{-1}(\rho(S_a-K_1)) + \frac{1}{3}\sinh^{-1}(\rho(S_a-K_2)) + \frac{1}{3}\sinh^{-1}(\rho(S_a-K_3))$, $c_b = \frac{1}{3}\sinh^{-1}(\rho(S_b-K_1)) + \frac{1}{3}\sinh^{-1}(\rho(S_b-K_2)) + \frac{1}{3}\sinh^{-1}(\rho(S_b-K_3))$, and $\xi \in [0,1]$ is uniformly discretized to generate nonuniform grid in physical space. To apply smoothing to the discrete butterfly payoff function on the resulting nonuniform grid, we use the formula (4.18) with smoothing modifications of fifth order given in Table C.3.

The stretching intensity applied is $\rho = 0.42$ in the numerical experiment. Table 5.13 shows the point-wise convergence results at K_1, K_2 and K_3, with and without the smoothing correction. The

(M,N)	α	V			Δ		Γ	
		value	error	conv	error	conv	error	conv
		w/o smoothing						
(40,20)	0.6667	0.568113749	7.43e-02	-	5.35e-04	-	4.21e-04	-
(80,40)	0.3333	0.459075097	3.58e-02	1.05	3.72e-04	0.53	1.91e-04	1.14
(160,80)	0.6667	0.512425925	1.74e-02	1.04	7.59e-05	2.29	8.83e-05	1.11
(320,160)	0.3333	0.486264480	8.76e-03	0.99	1.89e-05	2.01	4.37e-05	1.01
(640,320)	0.6667	0.499382679	4.36e-03	1.01	4.71e-06	2.00	2.18e-05	1.00
		w/ smoothing in Table 3.1						
(40,20)	0.6667	0.500060559	4.55e-03	-	1.23e-03	-	6.22e-05	-
(80,40)	0.3333	0.495628236	6.93e-04	2.71	1.73e-04	2.83	6.17e-06	3.33
(160,80)	0.6667	0.495075147	5.29e-05	3.71	1.08e-05	4.00	6.79e-07	3.18
(320,160)	0.3333	0.495028135	3.46e-06	3.94	6.89e-07	3.98	4.50e-08	3.91
(640,320)	0.6667	0.495025124	2.20e-07	3.98	4.32e-08	4.00	2.90e-09	3.96

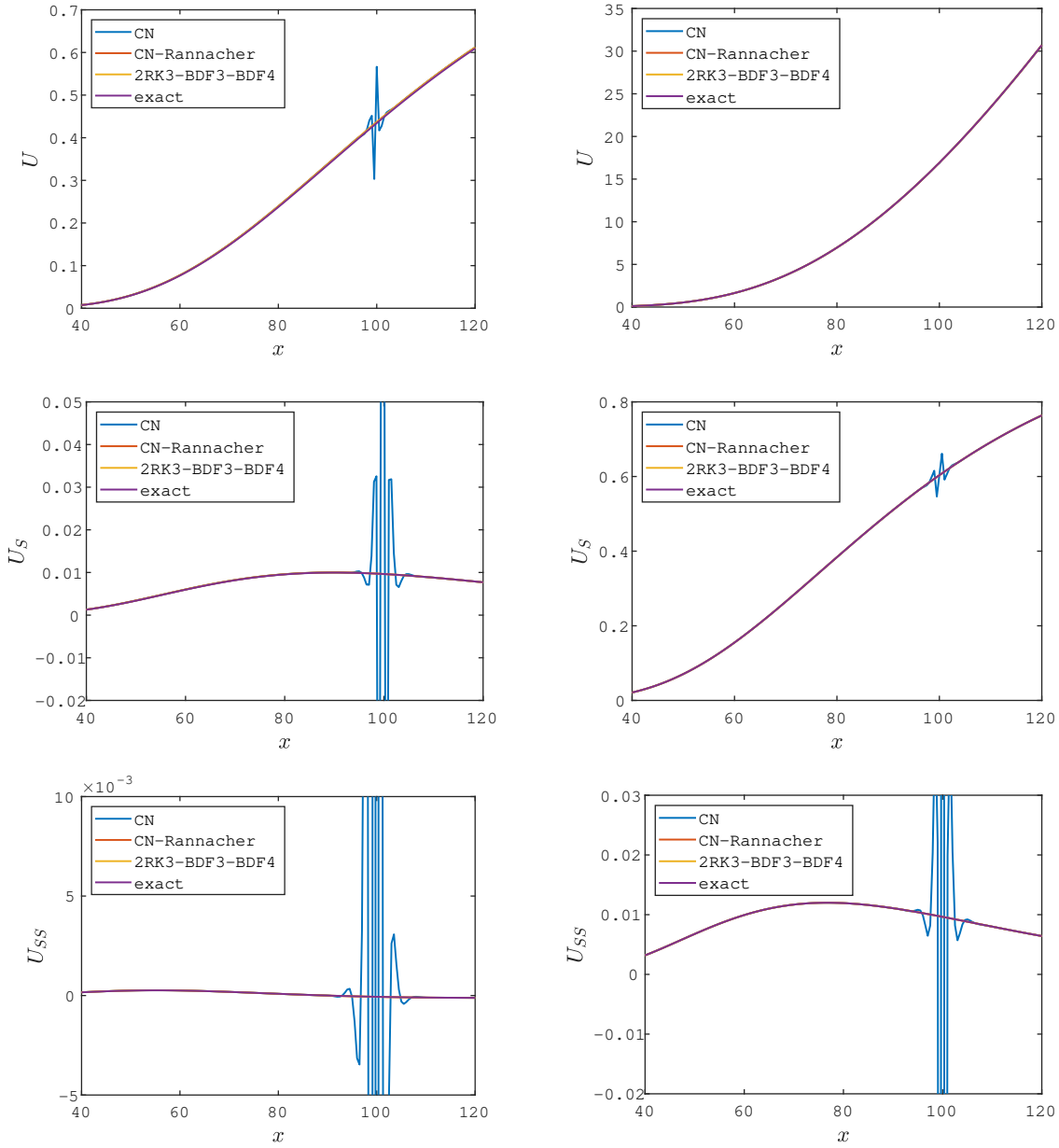
Table 5.10: Convergence results for the price V and its Δ and Γ at the strike $K = 100$, for solving the European digital call option, taking $\sigma = 0.2$. The grid alignment value α varies on each grid refinement level as given in the table.

(M,N)	α	V			Δ		Γ	
		value	error	conv	error	conv	error	conv
		w/o smoothing						
(40,20)	0.6667	22.706368799	4.61e-02	-	4.66e-05	-	1.45e-05	-
(80,40)	0.3333	22.670339162	1.00e-02	2.20	5.39e-05	-0.21	3.33e-06	2.12
(160,80)	0.6667	22.663038447	2.70e-03	1.89	1.10e-05	2.29	9.06e-07	1.88
(320,160)	0.3333	22.660991961	6.50e-04	2.05	3.20e-06	1.78	2.19e-07	2.05
(640,320)	0.6667	22.660507365	1.66e-04	1.97	7.54e-07	2.09	5.58e-08	1.97
		w/ smoothing in Table 3.1						
(40,20)	0.6667	22.659843234	5.96e-04	-	4.44e-05	-	1.59e-06	-
(80,40)	0.3333	22.660297803	3.78e-05	3.98	3.00e-06	3.89	3.67e-08	5.44
(160,80)	0.6667	22.660338994	2.61e-06	3.85	2.02e-07	3.90	2.76e-09	3.73
(320,160)	0.3333	22.660341607	1.67e-07	3.97	1.26e-08	4.00	1.86e-10	3.89
(640,320)	0.6667	22.660341776	1.05e-08	3.99	7.87e-10	4.00	1.14e-11	4.02

Table 5.11: Convergence results for the price V and its Δ and Γ at the strike $K = 100$, for solving the European call option, taking $\sigma = 0.8$. The grid alignment value α varies on each grid refinement level as given in the table.

(M,N)	α	V			Δ		Γ	
		value	error	conv	error	conv	error	conv
		$K_1 = 80.25$ (w/o smoothing)						
(40,20)	0.35000	4.108437670	7.78e-02	-	2.81e-02	-	3.21e-03	-
(80,40)	0.70000	4.173066559	5.38e-03	3.85	8.85e-03	1.67	2.44e-04	3.72
(160,80)	0.40000	4.169334971	8.57e-03	-0.67	1.36e-03	2.70	1.76e-04	0.48
(320,160)	0.80000	4.158899648	1.96e-03	2.13	2.70e-04	2.33	1.18e-05	3.89
(640,320)	0.60000	4.161549262	6.98e-04	1.49	5.92e-05	2.19	1.20e-05	-0.02
		$K_1 = 80.25$ (w/ smoothing obtained from Table 3.1)						
(40,20)	0.35000	4.376718594	1.30e-01	-	2.08e-02	-	4.65e-03	-
(80,40)	0.70000	4.176683715	1.22e-02	3.42	4.26e-03	2.29	3.03e-04	3.94
(160,80)	0.40000	4.161766412	7.92e-04	3.94	3.65e-04	3.55	1.79e-05	4.08
(320,160)	0.80000	4.160905971	5.05e-05	3.97	2.25e-05	4.02	1.15e-06	3.96
(640,320)	0.60000	4.160854034	3.18e-06	3.99	1.41e-06	3.99	7.23e-08	3.99
		$K_2 = 100$ (w/o smoothing)						
(40,20)	0.66667	9.096127563	2.38e-01	-	3.50e-02	-	4.30e-04	-
(80,40)	0.33333	9.338618860	9.71e-02	1.29	3.85e-03	3.19	1.65e-04	1.38
(160,80)	0.66667	9.415269963	2.22e-02	2.13	9.68e-04	1.99	1.13e-04	0.54
(320,160)	0.33333	9.431223427	6.43e-03	1.79	7.37e-05	3.72	2.29e-05	2.31
(640,320)	0.66667	9.436768375	8.96e-04	2.84	2.05e-05	1.85	9.74e-06	1.23
		$K_2 = 100$ (w/ smoothing obtained from Table 3.1)						
(40,20)	0.66667	9.174877799	1.71e-01	-	5.18e-02	-	2.02e-03	-
(80,40)	0.33333	9.425482368	8.49e-03	4.33	5.34e-04	6.60	2.71e-04	2.90
(160,80)	0.66667	9.436777317	6.97e-04	3.61	6.86e-05	2.96	1.91e-05	3.83
(320,160)	0.33333	9.437610473	4.71e-05	3.89	4.96e-06	3.79	1.25e-06	3.94
(640,320)	0.66667	9.437661793	2.97e-06	3.99	3.12e-07	3.99	7.89e-08	3.98
		$K_3 = 119.75$ (w/o smoothing)						
(40,20)	0.98333	4.246070790	6.30e-01	-	6.19e-03	-	2.36e-03	-
(80,40)	0.96667	4.725424517	1.49e-01	2.08	3.03e-03	1.03	3.76e-04	2.65
(160,80)	0.93333	4.849020081	2.58e-02	2.53	5.97e-04	2.34	4.26e-05	3.14
(320,160)	0.86667	4.870124235	4.73e-03	2.45	1.82e-04	1.72	1.38e-06	4.95
(640,320)	0.73333	4.874586104	2.70e-04	4.13	4.08e-05	2.16	2.30e-06	-0.74
		$K_3 = 119.75$ (w/ smoothing obtained from Table 3.1)						
(40,20)	0.98333	4.813475069	6.06e-02	-	1.73e-02	-	1.09e-03	-
(80,40)	0.96667	4.868931521	5.80e-03	3.39	1.58e-03	3.46	9.69e-05	3.49
(160,80)	0.93333	4.874470837	3.92e-04	3.89	1.06e-04	3.90	6.82e-06	3.83
(320,160)	0.86667	4.874832706	2.46e-05	3.99	6.73e-06	3.97	4.58e-07	3.90
(640,320)	0.73333	4.874854735	1.53e-06	4.01	4.22e-07	3.99	2.89e-08	3.98

Table 5.12: Convergence results for the price V and its Δ and Γ at the strikes $K_1 = 80.25$, $K_2 = 100$, $K_3 = 119.75$, for solving the butterfly spread option, taking $\sigma = 0.2$. The grid alignment values α vary for all three singular points on each grid refinement level as given in the table. Uniform grid in space is used.



Digital option Call option

Figure 5.6: Comparison of numerical solutions and the calculated Δ, Γ of the European digital call and call options, with volatility $\sigma = 0.2$, with CN, CN-Rannacher and 2RK3-BDF3-BDF4 methods. We see that CN-Rannacher, 2RK3-BDF3-BDF4 methods are indistinguishable from the exact, while CN exhibits oscillations.

results demonstrate clear stable fourth-order convergence in the solution and solution derivatives at all points. Moreover, compared to the results in Table 5.12 on a uniform grid, by using a nonuniform grid with points concentrated around the singular points, we observe more stable convergence rates and higher accuracy with less total number of grid points.

5.2.4 A general nonsmooth example on nonuniform grids

In this final example, we again apply the time stepping scheme developed in Chapter 3 and the smoothing scheme developed in Chapter 4, this time to solve the model PDE (3.1) with a more general nonsmooth initial condition. Consider the initial condition

$$v(0,x) = \begin{cases} -\sin\left(\frac{\pi x}{2}\right), & x < 0.123, \\ 1 + e^{-x}\cos\left(\frac{\pi x}{2}\right), & x \geq 0.123, \end{cases}$$

defined on $[x_a, x_b] \equiv [-4, 4]$. The boundary conditions are fixed to be $v(t, x_a) = v(0, x_a)$ and $v(t, x_b) = v(0, x_b)$. We solve the solution to time $T = 0.25$.

It is worth pointing out that the definition of boundary conditions in this example is different from other examples. Instead of imposing the exact solutions of the PDE defined on infinite space domain at the truncated boundaries, here we force the boundary conditions to be fixed values and stay unchanged during the whole time period. Under such definition of boundary conditions, the unknown exact solution of the current problem is not the same as that of the model PDE (3.1) defined on infinite domain. However, the numerical results below show that the methods developed in Chapters 3 and 4 apply equally well to the current problem. These results, in addition to the results of the previous examples, suggest that our analysis based on the model PDE (3.1) defined on infinite domain can be generalized to solve PDEs on finite domains.

The space discretization is performed on nonuniform grids with points concentrated around the singularities. We apply grid stretching around the nonsmooth point $x_K = 0.123$ such that the Taylor expansions in the derivation of smoothing modification formula in Table 5.14 are accurate enough. In the numerical experiments, we use fourth-order smoothings given in Tables 4.1, 4.2 and 4.3, where we center the nonsmooth point as much as possible. The grid stretching function we employ is

$$\xi = \phi(x) \equiv \frac{\sinh^{-1}(\rho(x - x_K)) - c_a}{c_b - c_a}, \tag{5.7}$$

where $c_a = \sinh^{-1}(\rho(x_a - x_K))$, $c_b = \sinh^{-1}(\rho(x_b - x_K))$, so that $\xi \in [0, 1]$ is uniformly discretized to generate a nonuniform grid in x, ρ is the stretching intensity and is set to be $\rho = 1.25$ in the numerical test, see [9]. Table 5.14 shows the results. The effect of our smoothing modification schemes on the high and stable order of convergence is evident. For performance comparison, we also solve the problem with smoothing modifications obtained from Table 3.1 with the convolution-type smoothing. We observe similar convergence results of the two types of smoothing modifications, but the convolution-type smoothings require the existence of three grid points on each side of the nonsmooth point. Such constraint is even more restraining for higher-order methods, see the discussion in Appendix C.1.

(M,N)	α	V			Δ		Γ	
		value	error	conv	error	conv	error	conv
		$K_1 = 80.25$ (w/o smoothing)						
(26,20)	0.76429	4.158494852	2.10e-03	-	1.36e-03	-	1.40e-04	-
(52,40)	0.50057	4.163822369	2.97e-03	-0.50	2.47e-04	2.46	6.22e-07	7.81
(104,80)	0.99915	4.160036706	8.14e-04	1.87	4.15e-05	2.57	1.08e-05	-4.12
(208,160)	0.99829	4.160619117	2.31e-04	1.82	5.28e-06	2.98	2.44e-06	2.15
(416,320)	0.99657	4.160780005	7.03e-05	1.72	5.41e-07	3.29	4.88e-07	2.32
		$K_1 = 80.25$ (w/ 5th-order smoothing modifications in Table C.3)						
(26,20)	0.76429	4.150151101	1.08e-02	-	5.97e-04	-	9.84e-05	-
(52,40)	0.50057	4.160237780	6.14e-04	4.14	4.09e-05	3.87	1.87e-06	5.72
(104,80)	0.99915	4.160816033	3.43e-05	4.16	2.61e-06	3.97	2.47e-08	6.24
(208,160)	0.99829	4.160848201	2.10e-06	4.03	1.65e-07	3.98	1.19e-10	7.70
(416,320)	0.99657	4.160850168	1.29e-07	4.02	1.05e-08	3.98	3.77e-12	4.98
		$K_2 = 100$ (w/o smoothing)						
(26,20)	0.98529	9.475449303	3.78e-02	-	5.80e-04	-	3.34e-04	-
(52,40)	0.96810	9.445075059	7.41e-03	2.35	3.09e-05	4.23	2.58e-05	3.69
(104,80)	0.93481	9.438314466	6.49e-04	3.51	2.92e-05	0.08	8.22e-06	1.65
(208,160)	0.86895	9.437727287	6.20e-05	3.39	1.01e-05	1.53	1.27e-06	2.70
(416,320)	0.73762	9.437640155	2.51e-05	1.30	3.08e-06	1.72	6.17e-08	4.36
		$K_2 = 100$ (w/ 5th-order smoothing modifications in Table C.3)						
(26,20)	0.98529	9.454236621	1.66e-02	-	6.09e-04	-	2.04e-04	-
(52,40)	0.96810	9.438908604	1.24e-03	3.74	4.37e-05	3.80	7.39e-06	4.79
(104,80)	0.93481	9.437747473	8.22e-05	3.92	2.83e-06	3.95	4.46e-07	4.05
(208,160)	0.86895	9.437670480	5.18e-06	3.99	1.76e-07	4.01	2.78e-08	4.00
(416,320)	0.73762	9.437665630	3.30e-07	3.97	1.12e-08	3.98	1.75e-09	3.99
		$K_3 = 119.75$ (w/o smoothing)						
(26,20)	0.20282	4.898365327	2.35e-02	-	4.03e-04	-	1.62e-04	-
(52,40)	0.43267	4.879549509	4.69e-03	2.32	1.55e-04	1.38	5.34e-06	4.93
(104,80)	0.87020	4.875123068	2.67e-04	4.14	2.89e-05	2.42	1.68e-06	1.67
(208,160)	0.73931	4.874939548	8.35e-05	1.68	3.29e-06	3.13	6.25e-08	4.75
(416,320)	0.47845	4.874870636	1.46e-05	2.52	7.30e-07	2.17	1.28e-07	-1.04
		$K_3 = 119.75$ (w/ 5th-order smoothing modifications in Table C.3)						
(26,20)	0.20282	4.888975250	1.41e-02	-	2.90e-04	-	1.24e-04	-
(52,40)	0.43267	4.875967711	1.11e-03	3.67	1.78e-05	4.03	2.89e-06	5.43
(104,80)	0.87020	4.874930596	7.45e-05	3.90	1.12e-06	3.98	1.71e-07	4.08
(208,160)	0.73931	4.874860746	4.68e-06	3.99	7.06e-08	3.99	1.08e-08	3.99
(416,320)	0.47845	4.874856364	2.98e-07	3.97	4.26e-09	4.05	6.63e-10	4.02

Table 5.13: Convergence results for the price V and its Δ and Γ at the strikes $K_1 = 80.25$, $K_2 = 100$, $K_3 = 119.75$, for solving the butterfly spread option, taking $\sigma = 0.2$. The grid alignment values α vary for all three singular points on each grid refinement level as given in the table. The nonuniform grid (5.6) is used.

$N(M)$	v			v'			v''		
	value	diff	conv	value	diff	conv	value	diff	conv
	w/o smoothing								
20	-0.157225412	-	-	0.929543617	-	-	0.829520352	-	-
40	-0.124985674	3.22e-02	-	0.962497095	3.30e-02	-	0.804577888	-2.49e-02	-
80	-0.107916178	1.71e-02	0.92	0.979509539	1.70e-02	0.95	0.787637784	-1.69e-02	0.56
160	-0.116193463	-8.28e-03	1.04	0.971047394	-8.46e-03	1.01	0.795563301	7.93e-03	1.10
320	-0.111990213	4.20e-03	0.98	0.975291206	4.24e-03	1.00	0.791442086	-4.12e-03	0.94
640	-0.109872626	2.12e-03	0.99	0.977416833	2.13e-03	1.00	0.789340606	-2.10e-03	0.97
	w/ smoothing modifications in Tables 4.1, 4.2 and 4.3								
20	-0.108780069	-	-	0.979692987	-	-	0.785025247	-	-
40	-0.108506644	2.73e-04	-	0.978853356	-8.40e-04	-	0.787838963	2.81e-03	-
80	-0.108489688	1.70e-05	4.01	0.978804771	-4.86e-05	4.11	0.787949563	1.11e-04	4.67
160	-0.108488568	1.12e-06	3.92	0.978801724	-3.05e-06	4.00	0.787957176	7.61e-06	3.86
320	-0.108488504	6.42e-08	4.12	0.978801536	-1.87e-07	4.02	0.787957579	4.02e-07	4.24
640	-0.108488500	3.72e-09	4.11	0.978801525	-1.17e-08	4.00	0.787957604	2.49e-08	4.01
	w/ smoothing modifications obtained from Table 3.1								
20	-0.108832549	-	-	0.979726053	-	-	0.785599787	-	-
40	-0.108508740	3.24e-04	-	0.978857520	-8.69e-04	-	0.787810194	2.21e-03	-
80	-0.108489736	1.90e-05	4.09	0.978804921	-5.26e-05	4.05	0.787948382	1.38e-04	4.00
160	-0.108488577	1.16e-06	4.04	0.978801744	-3.18e-06	4.05	0.787957056	8.67e-06	3.99
320	-0.108488504	7.25e-08	4.00	0.978801538	-2.06e-07	3.95	0.787957571	5.15e-07	4.07
640	-0.108488500	4.60e-09	3.98	0.978801525	-1.27e-08	4.03	0.787957603	3.15e-08	4.03

Table 5.14: Convergence results at the nonsmooth point $x = 0.123$ for solving the model problem (3.1) with a general nonsmooth initial condition, taking $a = 2$. Our method using (4.18) achieves fourth-order convergence of the solution and derivatives with smoothing modifications from Tables 4.1, 4.2 and 4.3, and from Table 3.1 as comparison. The nonuniform grid (5.7) is used.

Chapter 6

Concluding remarks

6.1 Conclusions

In this thesis, we studied high-order methods for solving European and American option pricing problems. We investigated the negative effects of nonsmoothness in the initial conditions and/or the solution itself, that cause trouble to obtain high-accuracy solutions, and proposed remedies to restore high-order accuracy. For American options, we first studied the error behaviour for solving free boundary problems using the standard fourth-order finite differences and BDF4 time-stepping scheme. Based on the analysis, we presented high-order deferred correction algorithms. Our algorithms utilize the penalty method and assume no prior knowledge of the exact free boundary location and derivative jumps at the free boundary. Our method does not modify the finite difference stencils and the arising matrix, but applies the corrections to the right-hand side. The penalty iteration converges in a few iterations. From the analysis of the error behaviors when solving free boundary problems, we showed that our deferred correction algorithms can successively increase the solution order of convergence from $\mathcal{O}(h^2)$ to $\mathcal{O}(h^3)$, and from $\mathcal{O}(h^3)$ to $\mathcal{O}(h^4)$ after applying each successive correction.

For European options, we applied Fourier analysis to a model convection-diffusion PDE and proved that the exponential damping of high-frequency error components using BDF schemes makes the schemes a good combination with RK3 as the starting scheme for nonsmooth data, and guarantees fourth-order convergence and stability, assuming the nonsmooth initial conditions are discretized to a high-order. Our analysis can be easily extended to even higher order methods in the BDF family.

To deal with the low-order quantization errors in the frequency domain introduced by the discretization of the nonsmooth initial conditions, we use smoothing techniques from [31]. To simplify implementation, we have provided explicit formulas for the discretization of the Dirac delta, Heaviside and ramp initial conditions as a high-order smoothing mechanism, so that the discrete initial conditions are fourth-order accurate in the frequency domain. We theoretically proved that with the proposed 2RK3-BDF3-BDF4 time stepping scheme and smoothing, fourth-order convergence can be obtained for the model convection-diffusion PDE with nonsmooth initial conditions.

Furthermore, we derived novel high-order smoothing modifications for a variety of nonsmooth

initial conditions that can be used as alternatives to [31]. Moreover, we suggested to use an additive modification formulation of the smoothing procedure. Based on the new formulation, we derived high-order smoothing modifications for general nonsmooth, piecewise analytic functions as initial conditions of parabolic PDEs, on uniform or nonuniform grids. Combined with the high-order 2RK3-BDF3-BDF4 time stepping scheme, our method achieves stable fourth-order convergence. Moreover, with the flexibility to apply smoothing to functions on nonuniform grids, we are able to achieve even better accuracy by concentrating more grid points around the nonsmooth location.

The numerical results on constant-coefficient convection-diffusion PDEs, and European digital call, call and butterfly spread options show stable fourth-order convergence, and verify the correctness of our analysis. Furthermore, the calculated solution derivatives exhibit fourth-order accuracy. Though the work in this thesis is developed with option pricing problems in mind, we note that the contributions of this thesis are not only applicable to option pricing problems, but also to more general parabolic PDEs with nonsmooth initial data, as also shown in the numerical experiments on various parabolic PDEs problems, and with more general nonsmooth initial data and nonmsooth solutions.

6.2 Generalizations and future work

The work in this thesis offers a fundamental framework that can be readily applied to solve numerous other interesting problems that have been regarded as challenging in the existing literature. Primarily, while our current focus has been on addressing issues within a single spatial dimension, the analytical framework established in this thesis can be extended to multi-dimensional settings. Specifically, the deferred correction algorithm introduced in Chapter 2 for one-dimensional scenarios can be adapted to applications in two spatial dimensions, albeit with some remaining difficulties to be resolved when mixed derivatives are involved. One typical such example is the elliptic obstacle problem in two dimensions, which is still an active area of research. The major difference in the algorithm when applied to two-dimensional problems is how the extrapolation scheme should be designed. Two-dimensional extrapolation has already been extensively studied in the literature (see, for example, [24, 37, 61]). Initial results on the extension to 2D elliptic obstacle problems have been obtained, and will be presented separately in a follow-up work. Regarding the high-order time stepping methods we developed for one-dimensional European options, we expect them to perform well in multiple dimensions with some alterations to the existing algorithm. Applying our methodologies to more complicated models, such as European options involving multiple underlyings and the Heston model as defined by (1.8), holds particular interest and deserves further exploration.

In a different context, all the algorithms in the thesis are formulated based on the standard fourth-order finite difference in space, chosen for its simplicity in analysis. However, this choice is not imperative. It is worthwhile to extend our methods and conduct analysis using generic parabolic difference operators, see e.g. [55]. We posit that, under reasonable assumptions, our findings in this work will likely generalize effectively to other types of difference operators, such as high-order compact finite difference schemes that are known for resulting in discretization matrices with smaller bandwidth, and other implicit L-stable time-stepping schemes in general formulation.

Throughout this thesis, we have performed the convergence studies with only uniform time-step

sizes, and applied a time-variable transformation heuristic to deal with the rapid change of solutions in particular examples as in Sections 2.2.4, 5.1.2, 5.1.3. A better way of handling this difficulty is to apply the time stepping with variable or adaptive step sizes. For example, by estimating and controlling the local error at each step, one can choose a suitable step size for the next step such that the global error maintains high order, and then apply the time stepping with variable step-size formula, see e.g. [2, 33].

Finally, when solving the American options, various heuristics were employed, including a stretching operation around $t = 0$. These measures were implemented to mitigate dominance of solution errors near $t = 0$ such that the global errors maintain fourth order. Despite yielding impressive high-order results, there is room for improvement. The solution of American options introduces an infinite singularity in time near expiry, posing challenges in achieving perfect fourth-order convergence. Addressing this issue remains an unresolved problem.

Appendix A

Solution derivatives of American put option at the free boundary

A.1 Second derivative at the free boundary

The American put option solution $v(t,s)$ on the continuation region satisfies the following free boundary equation (note that t here denotes the time to maturity)

$$\frac{\partial v}{\partial t} = \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 v}{\partial s^2} + rs\frac{\partial v}{\partial s} - rv, \quad (t,s) \in (0,T] \times (s_f, +\infty), \tag{A.1}$$

where s_f is the free boundary satisfying the smooth pasting conditions

$$v(t,s_f) = K - s_f, \quad \frac{\partial v}{\partial s}(t,s_f) = -1. \tag{A.2}$$

While the second derivative is discontinuous across the free boundary. From (A.1), for a point s that is infinitely close to $s_f(t)$ at some time t, we have

$$\begin{aligned}
\lim_{s\downarrow s_f(t)} \frac{\partial^2 v}{\partial s^2}(t,s) &= \lim_{s\downarrow s_f(t)} \frac{2}{\sigma^2 s^2}\left(\frac{\partial v}{\partial t}(t,s) - rs\frac{\partial v}{\partial s}(t,s) + rv(t,s)\right) \\
&= \frac{2}{\sigma^2 s_f^2}\left(\lim_{s\downarrow s_f(t)} \frac{\partial v}{\partial t}(t,s) - rs_f\frac{\partial v}{\partial s}(t,s_f) + rv(t,s_f)\right) \\
&= \frac{2}{\sigma^2 s_f^2}\left(\lim_{s\downarrow s_f(t)} \frac{\partial v}{\partial t}(t,s) - rs_f(-1) + r(K - s_f)\right) \\
&= \frac{2}{\sigma^2 s_f^2}\left(\lim_{s\downarrow s_f(t)} \frac{\partial v}{\partial t}(t,s_f) - rK\right).
\end{aligned}$$

For the time derivative $\lim_{s\downarrow s_f(t)} \frac{\partial v}{\partial t}(t,s_f)$, it turns out, this term is actually zero. First, we can show that $\frac{\partial v}{\partial t}(t,s_f(t)) = 0$ at the free boundary [26]. From the boundary condition (A.2)

$$\frac{\partial v}{\partial t}(t,s_f) + \dot{s}_f\frac{\partial v}{\partial s}(t,s_f) = \frac{\partial v}{\partial t}(t,s_f) + \dot{s}_f(-1) = -\dot{s}_f,$$

where $\dot{s}_f$ means taking derivative of the free boundary s_f with respect to time t. This immediately gives

$$\frac{\partial v}{\partial t}(t, s_f) = 0. \tag{A.3}$$

From [38, Lemma 5], we know that $\frac{\partial v}{\partial t}(t, s)$ is continuous across and vanish at the free boundary. Hence, we also have

$$\lim_{s \downarrow s_f(t)} \frac{\partial v}{\partial t}(t, s_f) = 0.$$

Therefore, we get

$$\lim_{s \downarrow s_f(t)} \frac{\partial^2 v}{\partial s^2}(t, s) = \frac{2rK}{\sigma^2 s_f^2}. \tag{A.4}$$

We will see later that $\frac{\partial^2 v}{\partial s^2} v(t, s)$ has a jump discontinuity at $s_f(t)$ and

$$\frac{\partial^2 v}{\partial s^2}(t, s_f(t) = \frac{rK}{\sigma^2 s_f^2}.$$

For notational convenience, we introduce the multi-index notation. Let $\alpha = (\alpha_1, \alpha_2)$ be a tuple of 2 nonnegative integers. Given a multi-index α, define

$$v^{(\alpha)} \equiv D^\alpha v(t, s) \equiv \left(\frac{\partial}{\partial t}\right)^{\alpha_1} \left(\frac{\partial}{\partial s}\right)^{\alpha_2} v(t, s),$$

assuming the derivatives are valid. Define $v_B^{(\alpha)} \equiv D^\alpha v(t, s_f(t))$, and $v_{\downarrow B}^{(\alpha)} \equiv \lim_{s \downarrow s_f(t)} D^\alpha v(t, s_f(t))$, e.g. $v_B^{(0,2)} = \frac{\partial^2 v}{\partial s^2}(t, s)$, $v_{\downarrow B}^{(0,2)} = \lim_{s \downarrow s_f(t)} \frac{\partial^2 v}{\partial s^2}(t, s)$.

A.2 Higher derivatives at the free boundary

Given that the solution is smooth enough, we can also derive higher derivatives at the free boundary. Similar results can be found in [38]. Take the total derivative with respect to t on both sides of (A.3), we get

$$v_{\downarrow B}^{(2,0)} + \dot{s}_f v_{\downarrow B}^{(1,1)} = 0. \tag{A.5}$$

From (A.4), we obtain $\frac{1}{2}\sigma^2 s_f^2 v_{\downarrow B}^{(0,2)} = rK$, and taking total derivative with respect to t, we get

$$\begin{aligned}
&\sigma^2 s_f \dot{s}_f v_{\downarrow B}^{(0,2)} + \frac{1}{2}\sigma^2 s_f^2 (v_{\downarrow B}^{(s,s,t)} + \dot{s}_f v_{\downarrow B}^{(0,3)}) \\
=&\frac{1}{2}\sigma^2 s_f^2 v_{\downarrow B}^{(s,s,t)} + \frac{1}{2}\sigma^2 s_f^2 \dot{s}_f v_{\downarrow B}^{(0,3)} + \sigma^2 s_f \dot{s}_f \frac{2rK}{\sigma^2 s_f^2} \\
=&\frac{1}{2}\sigma^2 s_f^2 v_{\downarrow B}^{(s,s,t)} + \frac{1}{2}\sigma^2 s_f^2 \dot{s}_f v_{\downarrow B}^{(0,3)} + 2rK \frac{\dot{s}_f}{s_f} = 0.
\end{aligned} \tag{A.6}$$

Taking partial derivative with respect to t on (A.1) and taking the limit $s \downarrow s_f$, we get

$$v_{\downarrow B}^{(2,0)} - \frac{1}{2}\sigma^2 s_f^2 v_{\downarrow B}^{(s,s,t)} - r s_f v_{\downarrow B}^{(s,t)} = 0. \tag{A.7}$$

where we have used the fact $v_{\downarrow B}^{(t)} = 0$ from (A.3). Taking partial derivative with respect to s to (A.1) and taking the limit $s \downarrow s_f$, we get

$$\begin{aligned} & v_{\downarrow B}^{(1,1)} - \sigma^2 s_f v_{\downarrow B}^{(0,2)} - \frac{1}{2}\sigma^2 s_f^2 v_{\downarrow B}^{(0,3)} - r v_{\downarrow B}^{(s)} - r s_f v_{\downarrow B}^{(0,2)} + r v_{\downarrow B}^{(s)} \\ =& v_{\downarrow B}^{(1,1)} - (r+\sigma^2) s_f v_{\downarrow B}^{(0,2)} - \frac{1}{2}\sigma^2 s_f^2 v_{\downarrow B}^{(0,3)} \\ =& v_{\downarrow B}^{(1,1)} - \frac{1}{2}\sigma^2 s_f^2 v_{\downarrow B}^{(0,3)} - (r+\sigma^2)\frac{2rK}{\sigma^2 s_f} = 0. \end{aligned} \tag{A.8}$$

Taking two partial derivatives with respect to s to (A.1) and plugging in s_f, we obtain

$$\begin{aligned} & v_{\downarrow B}^{(1,2)} - (r+\sigma^2) s_f v_{\downarrow B}^{(0,3)} - (r+\sigma^2) v_{\downarrow B}^{(0,2)} - \frac{1}{2}\sigma^2 s_f^2 v_{\downarrow B}^{(0,4)} - \sigma^2 s_f v_{\downarrow B}^{(0,3)} \\ =& -\frac{1}{2}\sigma^2 s_f^2 v_{\downarrow B}^{(0,4)} - (r+2\sigma^2) s_f v_{\downarrow B}^{(0,3)} + v_{\downarrow B}^{(1,2)} - (r+\sigma^2)\frac{2rK}{\sigma^2 s_f^2} = 0 \end{aligned} \tag{A.9}$$

Denote $\mathcal{X}_1 = v_{\downarrow B}^{(1,1)}, \mathcal{X}_2 = v_{\downarrow B}^{(2,0)}, \mathcal{X}_3 = v_{\downarrow B}^{(1,2)}, \mathcal{X}_4 = v_{\downarrow B}^{(0,3)}, \mathcal{X}_5 = v_{\downarrow B}^{(0,4)}$, we have five equations with five unknowns from Equations A.5, A.6, A.7, A.8 and A.9

$$\begin{cases} \dot{s}_f \mathcal{X}_1 + \mathcal{X}_2 = 0, \\ \mathcal{X}_1 - \frac{1}{2}\sigma^2 s_f^2 \mathcal{X}_4 = (r+\sigma^2)\frac{2rK}{\sigma^2 s_f}, \\ -r s_f \mathcal{X}_1 + \mathcal{X}_2 - \frac{\sigma^2 s_f^2}{2}\mathcal{X}_3 = 0, \\ \frac{1}{2}\sigma^2 s_f^2 \mathcal{X}_3 + \frac{1}{2}\sigma^2 s_f^2 \dot{s}_f \mathcal{X}_4 = -2rK\frac{\dot{s}_f}{s_f}, \\ \mathcal{X}_3 - (r+2\sigma^2) s_f \mathcal{X}_4 - \frac{1}{2}\sigma^2 s_f^2 \mathcal{X}_5 = (r+\sigma^2)\frac{2rK}{\sigma^2 s_f^2}. \end{cases} \tag{A.10}$$

Solving the linear system gives

$$\begin{aligned} v_{\downarrow B}^{(1,1)} &= -\frac{2rK}{\sigma^2 s_f}\frac{\dot{s}_f}{s_f}, \\ v_{\downarrow B}^{(2,0)} &= \frac{2rK}{\sigma^2}\left(\frac{\dot{s}_f}{s_f}\right)^2, \\ v_{\downarrow B}^{(1,2)} &= \frac{4rK}{\sigma^4 s_f^2}\frac{\dot{s}_f}{s_f}\left(\frac{\dot{s}_f}{s_f} + r\right), \\ v_{\downarrow B}^{(0,3)} &= -\frac{4rK}{\sigma^4 s_f^3}\left(\frac{\dot{s}_f}{s_f} + (r+\sigma^2)\right), \\ v_{\downarrow B}^{(0,4)} &= \frac{4rK}{\sigma^4 s_f^4}\left[(r+\sigma^2) + \frac{2}{\sigma^2}\left(\frac{\dot{s}_f}{s_f} + (r+\sigma^2)\right)^2\right]. \end{aligned} \tag{A.11}$$

Since the free boundary $s_f(t)$ moves infinitely quickly near expiry [5] and $\dot{s}_f(t)$ goes to ∞ at $t = 0$. We see that all the higher spatial derivatives above order 2 blows up near $t = 0$.

Appendix B

Fourier transform of initial conditions

B.1 Fourier transform of the analytic solutions

For reference in Chapter 3, the Fourier transform of the Heaviside, ramp and quadratic ramp functions are, respectively,

$$\hat{v}_H(t=0) = \int_{-\infty}^{\infty} e^{-i\omega x} e^{-\eta x} H(x) dx = \int_0^{\infty} e^{-i\omega x - \eta x} dx = \frac{1}{i\kappa}$$
$$\hat{v}_C(t=0) = \int_{-\infty}^{\infty} e^{-i\omega x} e^{-\eta x} \max(x,0) dx = \int_0^{\infty} e^{-i\omega x - \eta x} x dx = -\frac{1}{\kappa^2},$$
$$\hat{v}_Q(t=0) = \int_{-\infty}^{\infty} e^{-i\omega x} e^{-\eta x} Q(x) dx = \int_0^{\infty} \frac{1}{2} e^{-i\omega x - \eta x} x^2 dx = -\frac{1}{i\kappa^3},$$

where $\kappa = \omega - i\eta$ for some $\eta > 0$. Note that the above transformations can be considered as the usual Fourier transforms but with frequency in the complex domain. For example, for the Heaviside function, we have

$$\int_{-\infty}^{\infty} e^{-i\kappa x} H(x) dx = \frac{1}{i\kappa}.$$

The corresponding inverse Fourier transform is just

$$e^{-\eta x} H(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega x} \hat{v}_H^{(0)} d\omega,$$

which means

$$H(x) = \frac{1}{2\pi} e^{\eta x} \int_{-\infty}^{\infty} e^{i\omega x} \hat{v}_H^{(0)} d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega x + \eta x} \hat{v}_H^{(0)} d\omega = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\kappa x} \hat{v}_H^{(0)} d\omega.$$

The Fourier transform of solutions $e^{-\eta x}v_H(t,x)$, $e^{-\eta x}v_C(t,x)$ and $e^{-\eta x}v_Q(t,x)$ are respectively

$$\begin{aligned}
\mathcal{F}[e^{-\eta x}v_H(x)](\omega) &= \frac{e^{-i\kappa\mu-\frac{1}{2}\sigma^2\kappa^2}}{i\kappa} = \frac{e^{-(ia\kappa+\kappa^2)t}}{i\kappa},\\
\mathcal{F}[e^{-\eta x}v_C(x)](\omega) &= -\frac{e^{-i\kappa\mu-\frac{1}{2}\sigma^2\kappa^2}}{\kappa^2} = -\frac{e^{-(ia\kappa+\kappa^2)t}}{\kappa^2},\\
\mathcal{F}[e^{-\eta x}v_Q(x)](\omega) &= -\frac{e^{-i\kappa\mu-\frac{1}{2}\sigma^2\kappa^2}}{i\kappa^3} = -\frac{e^{-(ia\kappa+\kappa^2)t}}{i\kappa^3}.
\end{aligned}$$

B.2 Fourier transform of the discrete Heaviside, ramp and quadratic ramp functions

B.2.1 The discrete Heaviside function

On a discretized grid $x_j = (j+(1-\alpha))h$ with a general alignment value α, the semi-discrete Fourier transform of the discrete Heaviside function (4.4) is

$$\hat{v}^{(0)}_{H,h,\alpha}(\kappa) \equiv h\sum_{j=-\infty}^{\infty} e^{-i\kappa x_j}H(x_j) = h\sum_{j=0}^{\infty} e^{-i\kappa(j+(1-\alpha))h} = \frac{he^{-i(1-\alpha)\kappa h}}{1-e^{-i\kappa h}}.$$

The Taylor expansion of $\frac{h}{1-e^{-i\kappa h}}$ is

$$\begin{aligned}
\frac{h}{1-e^{-i\kappa h}} &= \frac{h}{1-\left(1-i\kappa h+\frac{(-i\kappa h)^2}{2}+\frac{(-i\kappa h)^3}{6}+\frac{(-i\kappa h)^4}{24}+\frac{(-i\kappa h)^5}{120}+\frac{(-i\kappa h)^6}{720}+\mathcal{O}(\kappa^7h^7)\right)}\\
&= \frac{h}{i\kappa h-\frac{i^2\kappa^2h^2}{2}-\frac{-i^3\kappa^3h^3}{6}-\frac{i^4\kappa^4h^4}{24}-\frac{-i^5\kappa^5h^5}{120}-\frac{i^6\kappa^6h^6}{720}+\mathcal{O}(\kappa^7h^7)}\\
&= \frac{1}{i\kappa+\frac{\kappa^2h}{2}-i\frac{\kappa^3h^2}{6}-\frac{\kappa^4h^3}{24}+i\frac{\kappa^5h^4}{120}+\frac{\kappa^6h^5}{720}+\mathcal{O}(\kappa^7h^6)}\\
&= \frac{1}{i\kappa}\cdot\frac{1}{1-i\frac{\kappa h}{2}-\frac{\kappa^2h^2}{6}+i\frac{\kappa^3h^3}{24}+\frac{\kappa^4h^4}{120}-i\frac{\kappa^5h^5}{720}+\mathcal{O}(\kappa^6h^6)}\\
&= \frac{1}{i\kappa}\left(1+i\frac{\kappa h}{2}-\frac{\kappa^2h^2}{12}-\frac{\kappa^4h^4}{720}-i\frac{\kappa^5h^5}{32}+\mathcal{O}(\kappa^6h^6)\right)\\
&= \frac{1}{i\kappa}+\frac{1}{2}h+\frac{i\kappa}{12}h^2+\frac{i\kappa^3}{720}h^4-\frac{\kappa^4}{32}h^5+\mathcal{O}(\kappa^5h^6).
\end{aligned}$$

Applying Taylor expansion to both $\frac{h}{1-e^{-i\kappa h}}$ and $e^{-i(1-\alpha)\kappa h}$, and using the fact that $\hat{v}^{(0)}_H = \frac{1}{i\kappa}$, we obtain

$$\begin{aligned}
&\hat{v}^{(0)}_{H,h,\alpha}\\
&=\left(\frac{1}{i\kappa}+\frac{h}{2}+\frac{i\kappa h^2}{12}+\frac{i\kappa^3h^4}{720}+\mathcal{O}(\kappa^4h^5)\right)\left(1-i(1-\alpha)\kappa h-\frac{(1-\alpha)^2\kappa^2h^2}{2}+i\frac{(1-\alpha)^3\kappa^3h^3}{6}+\frac{(1-\alpha)^4\kappa^4h^4}{24}+\mathcal{O}(\kappa^5h^5)\right)\\
&=\hat{v}^{(0)}_H+\left(\alpha-\frac{1}{2}\right)h+\frac{i}{2}\left(\alpha^2-\alpha+\frac{1}{6}\right)\kappa h^2+\frac{\alpha(1-\alpha)(2\alpha-1)}{12}\kappa^2h^3+i\left(-\frac{\alpha^4}{24}+\frac{\alpha^3}{12}-\frac{\alpha^2}{24}+\frac{1}{720}\right)\kappa^3h^4+\mathcal{O}(\kappa^4h^5)\\
&=\hat{v}^{(0)}_H+\left(\alpha-\frac{1}{2}\right)h+\frac{i}{2}\left(\left(\alpha-\frac{1}{2}\right)^2-\frac{1}{12}\right)\kappa h^2+\frac{\alpha(1-\alpha)(2\alpha-1)}{12}\kappa^2h^3+i\left(-\frac{\alpha^4}{24}+\frac{\alpha^3}{12}-\frac{\alpha^2}{24}+\frac{1}{720}\right)\kappa^3h^4+\mathcal{O}(\kappa^4h^5),
\end{aligned}$$

where the coefficient in front of $\kappa^4 h^5$ term is $-\frac{1}{32} - \frac{\alpha}{720} + \frac{\alpha^3}{72} - \frac{\alpha^4}{48} + \frac{\alpha^5}{120}$.

In order to achieve fourth-order convergence, we need to get rid of the lower order terms in h. To attempt to do that, we notice that picking the special alignment values $\alpha = 1$ and $\alpha = \frac{1}{2}$ will not cancel out all the lower-order terms. In particular, when $\alpha = 1$,

$$\hat{v}^{(0)}_{H,h,1}(\kappa) = \frac{h}{1 - e^{-i\kappa h}}.$$

which is of first-order accurate since

$$\hat{v}^{(0)}_{H,h,1} = \hat{v}^{(0)}_H + \frac{1}{2}h + \frac{i\kappa}{12}h^2 + \frac{i\kappa^3}{720}h^4 + \mathcal{O}(\kappa^4 h^5).$$

When $\alpha = \frac{1}{2}$, it is of second-order accurate since

$$\hat{v}^{(0)}_{H,h,\frac{1}{2}} = h\sum_{j=0}^{\infty} e^{-i\kappa(j+1/2)h} H(x_j) = \hat{v}^{(0)}_H - i\frac{1}{24}\kappa h^2 + \mathcal{O}(\kappa^3 h^4).$$

Note that

$$\frac{1}{24}\left(e^{i\kappa\frac{h}{2}} - e^{-i\kappa\frac{h}{2}}\right) = i\frac{1}{24}\kappa h + \mathcal{O}(\kappa^3 h^3).$$

Hence,

$$h\sum_{j=0}^{\infty} e^{-i\kappa(j+1/2)h} H(x_j) + \frac{h}{24}\left(e^{-i\kappa(-h/2)} - e^{-i\kappa(h/2)}\right) = \hat{v}^{(0)}_H + \mathcal{O}(\kappa^3 h^4).$$

Therefore, if we define

$$\tilde{H}_{\frac{1}{2}}(x_j) \equiv H(x_j) + \begin{cases} \frac{1}{24}, & j = -1 \\ -\frac{1}{24}, & j = 0 \\ 0, & \text{else} \end{cases} = \begin{cases} 0, & j < -1 \\ \frac{1}{24}, & j = -1 \\ \frac{23}{24}, & j = 0 \\ 1, & j > 0. \end{cases} \tag{B.1}$$

this immediately gives us

$$h\sum_{j=0}^{\infty} e^{-i\kappa(j+1/2)h} \tilde{H}_{\frac{1}{2}}(x_j) = \hat{v}^{(0)}_H + \mathcal{O}(h^4)$$

as desired. This is the motivation for constructing the modification schemes for smoothing operations.

Note that placing the nonsmoothing point in the midway of grid points leads to some complications in grid refinement. It is best to align the nonsmooth point on the grid. This can be done by picking $\alpha = 1$, and following the same strategy to remove lower order terms by adding

$$h\left(-\frac{1}{2} - \frac{1}{24}(e^{i\kappa h} - e^{-i\kappa h})\right).$$

Therefore, if we define

$$\tilde{H}_1(x_j) \equiv H(x_j) + \begin{cases} -\frac{1}{24}, & j=-1 \\ -\frac{1}{2}, & j=0 \\ \frac{1}{24}, & j=1 \\ 0, & \text{else} \end{cases} = \begin{cases} 0, & j<-1 \\ -\frac{1}{24}, & j=-1 \\ \frac{1}{2}, & j=0 \\ \frac{25}{24}, & j=1 \\ 1, & j>1. \end{cases} \tag{B.2}$$

we would have

$$h\sum_{j=0}^{\infty} e^{-i\kappa jh}\tilde{H}_1(x_j) = \hat{v}_H^{(0)} + \mathcal{O}(h^4)$$

as desired.

B.2.2 The discrete ramp function

On a discretized grid $x_j = (j+(1-\alpha))h$ with a general alignment value α, the semi-discrete Fourier transform of the discrete ramp function (4.5) is

$$\begin{aligned} \hat{v}_{C,h,\alpha}^{(0)} &\equiv h\sum_{j=-\infty}^{\infty} e^{-i\kappa x_j}\max(x_j,0) = h\sum_{j=0}^{\infty} e^{-i\kappa x_j}x_j = h\sum_{j=0}^{\infty} e^{-i\kappa(j+(1-\alpha))h}(j+(1-\alpha))h \\ &= h^2 e^{-i\kappa(1-\alpha)h}\left(\frac{e^{-i\kappa h}}{(1-e^{-i\kappa h})^2} + \frac{1-\alpha}{1-e^{-i\kappa h}}\right) \\ &= h^2 e^{-i\kappa(1-\alpha)h}\left(\frac{1}{(1-e^{-i\kappa h})^2} - \frac{\alpha}{1-e^{-i\kappa h}}\right). \end{aligned}$$

Applying Taylor expansion as in the previous section, we have

$$\frac{1}{1-e^{-i\kappa h}} = \frac{1}{i\kappa h} + \frac{1}{2} + \frac{i\kappa}{12}h + \frac{i\kappa^3}{720}h^3 - \frac{\kappa^4}{32}h^4 + \mathcal{O}(\kappa^5 h^5),$$

and

$$\begin{aligned} \frac{1}{(1-e^{-i\kappa h})^2} &= \frac{1}{\left(i\kappa h + \frac{\kappa^2h^2}{2} - i\frac{\kappa^3h^3}{6} - \frac{\kappa^4h^4}{24} + i\frac{\kappa^5h^5}{120} + \frac{\kappa^6h^6}{720} + \mathcal{O}(\kappa^7h^7)\right)^2} \\ &= \frac{1}{(i\kappa h)^2}\cdot\frac{1}{\left(1-\left(i\frac{\kappa h}{2} + \frac{\kappa^2h^2}{6} - i\frac{\kappa^3h^3}{24} - \frac{\kappa^4h^4}{120} + i\frac{\kappa^5h^5}{720} + \mathcal{O}(\kappa^6h^6)\right)\right)^2} \\ &= -\frac{1}{\kappa^2h^2}\frac{1}{1-\left(i\kappa h + \frac{7}{12}\kappa^2h^2 - i\frac{1}{4}\kappa^3h^3 - \frac{31}{360}\kappa^4h^4 + i\frac{\kappa^5h^5}{40} + \mathcal{O}(\kappa^6h^6)\right)} \\ &= -\frac{1}{\kappa^2h^2}\left(1 + i\kappa h - \frac{5}{12}\kappa^2h^2 - i\frac{1}{12}\kappa^3h^3 + \frac{1}{240}\kappa^4h^4 - i\frac{\kappa^5h^5}{720} + \mathcal{O}(\kappa^6h^6)\right). \end{aligned}$$

Hence,

$$h^2\left(\frac{1}{(1-e^{-i\kappa h})^2}-\frac{\alpha}{1-e^{-i\kappa h}}\right)=$$
$$-\frac{1}{\kappa^2}-i(1-\alpha)\frac{h}{\kappa}+\left(\frac{5}{12}-\frac{\alpha}{2}\right)h^2+i\frac{1-\alpha}{12}\kappa h^3-\frac{1}{240}\kappa^2h^4+i\frac{1-\alpha}{720}\kappa^3h^5+\mathcal{O}(\kappa^4h^6),$$

and

$$\begin{aligned}\hat{v}^{(0)}_{C,h,\alpha}&=h^2e^{-i\kappa(1-\alpha)h}\left(\frac{1}{(1-e^{-i\kappa h})^2}-\frac{\alpha}{1-e^{-i\kappa h}}\right)\\&=\left(-\frac{1}{\kappa^2}-i(1-\alpha)\frac{h}{\kappa}+\left(\frac{5}{12}-\frac{\alpha}{2}\right)h^2+i\frac{1-\alpha}{12}\kappa h^3-\frac{1}{240}\kappa^2h^4+i\frac{1-\alpha}{720}\kappa^3h^5+\mathcal{O}(\kappa^4h^6)\right)\\&\cdot\left(1-i(1-\alpha)\kappa h-\frac{(1-\alpha)^2}{2}\kappa^2h^2+i\frac{(1-\alpha)^3}{6}\kappa^3h^3+\frac{(1-\alpha)^4}{24}\kappa^4h^4-i\frac{(1-\alpha)^5}{120}\kappa^5h^5+\mathcal{O}(\kappa^6h^6)\right)\\&=\hat{v}^{(0)}_C-\frac{1}{2}\left(\alpha^2-\alpha+\frac{1}{6}\right)h^2+\frac{i}{6}\alpha(1-\alpha)(2\alpha-1)\kappa h^3+\frac{1}{240}\left(30\alpha^2(1-\alpha)^2-1\right)\kappa^2h^4+\mathcal{O}(\kappa^3h^5),\end{aligned}$$

where the coefficient in front of the κ^3h^5 term is $i\left(-\frac{\alpha}{180}+\frac{\alpha^3}{18}-\frac{\alpha^4}{12}+\frac{\alpha^5}{30}\right)$.

Again, we observe that by picking either $\alpha=1$ or $\alpha=\frac{1}{2}$ will not cancel out all the lower-order errors. In particular, when $\alpha=1$,

$$\hat{v}^{(0)}_{C,h,\alpha=1}=\hat{v}^{(0)}_C-\frac{1}{12}h^2+\mathcal{O}(\kappa^2h^4).$$

We still need to get rid of the low-order term $-\frac{h^2}{12}$. Notice that $x_0=0$, this can be achieved by simply adding $\frac{h^2}{12}$ such that

$$\hat{v}^{(0)}_{C,h,\alpha=1}+\frac{h^2}{12}=h\sum_{j=-\infty}^{\infty}e^{-i\kappa x_j}C_1(x_j)+\frac{h^2}{12}e^{-i\kappa x_0}=\hat{v}^{(0)}_C+\mathcal{O}(h^4).$$

Therefore, if we define

$$\tilde{C}_1(x_j)\equiv\max(x_j,0)+\begin{cases}\frac{h}{12}, & j=0\\ 0, & \text{else.}\end{cases}=\begin{cases}0, & j\le-1\\ \frac{h}{12}, & j=0\\ jh, & j\ge1\end{cases}\tag{B.3}$$

we would have

$$h\sum_{j=-\infty}^{\infty}e^{-i\kappa x_j}\tilde{C}_1(x_j)==\hat{v}^{(0)}_C+\mathcal{O}(h^4).$$

as desired. On the other hand, if we choose $\alpha=1/2$,

$$\hat{v}^{(0)}_{C,h,\alpha=\frac{1}{2}}=\hat{v}^{(0)}_C+\frac{1}{24}h^2+\mathcal{O}(\kappa^2h^4).$$

We need to get rid of the low-order term $\frac{1}{24}h^2$. This can be achieved by noticing that

$$-\frac{1}{48}(e^{i\kappa h/2}+e^{-i\kappa h/2}) = -\frac{1}{24}+\frac{1}{192}\kappa^2h^2+\mathcal{O}(h^4).$$

Hence, we can define

$$\tilde{C}_{1/2}(x_j) \equiv \max(x_j,0) + \begin{cases} -\frac{h}{48}, & j=-1,0 \\ 0, & \text{else,} \end{cases} = \begin{cases} 0, & j<-1 \\ -\frac{h}{48}, & j=-1 \\ \frac{23h}{48}, & j=0 \\ (j+\frac{1}{2})h, & j>0. \end{cases} \tag{B.4}$$

and have

$$h\sum_{j=0}^{\infty} e^{-i\kappa x_j}\tilde{C}_{\frac{1}{2}}(x_j) = \hat{v}_C^{(0)} + \mathcal{O}(h^4).$$

as desired.

B.2.3 The discrete quadratic ramp function

For the discrete quadratic ramp function (4.6), its semi-discrete Fourier transform is

$$\hat{v}_{Q,h,\alpha}^{(0)} \equiv h\sum_{j=-\infty}^{\infty} e^{-i\kappa x_j}\frac{x_j^2}{2}H_\alpha(x_j) = \frac{h^3}{2}\frac{e^{-i\kappa(1-\alpha)h}}{(1-e^{-i\kappa h})^3}\left((1-\alpha)^2-(2\alpha^2-2\alpha-1)e^{-i\kappa h}+\alpha^2e^{-i2\kappa h}\right).$$

We derive the Taylor expansion of $\hat{v}_{Q,h,\alpha}^{(0)}$. First, since

$$\begin{aligned}
\frac{1}{(1-e^{-i\kappa h})^3} &= \frac{1}{\left(i\kappa h+\frac{\kappa^2h^2}{2}-i\frac{\kappa^3h^3}{6}-\frac{\kappa^4h^4}{24}+i\frac{\kappa^5h^5}{120}+\frac{\kappa^6h^6}{720}+\mathcal{O}(\kappa^7h^7)\right)^3} \\
&= \frac{1}{(i\kappa h)^3}\cdot\frac{1}{\left(1-\left(i\frac{\kappa h}{2}+\frac{\kappa^2h^2}{6}-i\frac{\kappa^3h^3}{24}-\frac{\kappa^4h^4}{120}+i\frac{\kappa^5h^5}{720}+\mathcal{O}(\kappa^6h^6)\right)\right)^3} \\
&= \frac{1}{(i\kappa h)^3}\cdot\frac{1}{1-\left(i\frac{3\kappa h}{2}+\frac{5\kappa^2h^2}{4}-i\frac{3\kappa^3h^3}{4}-\frac{43\kappa^4h^4}{120}+i\frac{23\kappa^5h^5}{160}+\mathcal{O}(\kappa^6h^6)\right)} \\
&= \frac{i}{\kappa^3h^3}\left(1+i\frac{3\kappa h}{2}-\kappa^2h^2-i\frac{3\kappa^3h^3}{8}+\frac{19\kappa^4h^4}{240}-i\frac{11\kappa^5h^5}{80}+\mathcal{O}(\kappa^6h^6)\right).
\end{aligned}$$

Therefore, we have

$$
\begin{aligned}
&\hat{v}^{(0)}_{Q,h,\alpha} = \frac{h^3}{2} e^{-i\kappa(1-\alpha)h} \frac{(1-\alpha)^2 - (2\alpha^2 - 2\alpha - 1)e^{-i\kappa h} + \alpha^2 e^{-i2\kappa h}}{(1-e^{-i\kappa h})^3} \\
=&\frac{h^3}{2}\left(1 - i(1-\alpha)\kappa h - \frac{(1-\alpha)^2\kappa^2 h^2}{2} + i\frac{(1-\alpha)^3\kappa^3 h^3}{6} + \frac{(1-\alpha)^4\kappa^4 h^4}{24} - i\frac{(1-\alpha)^5\kappa^5 h^5}{120} + \mathcal{O}(\kappa^6 h^6)\right) \\
&\cdot \left[(1-\alpha)^2 - (2\alpha^2 - 2\alpha - 1)\left(1 - i\kappa h - \frac{\kappa^2 h^2}{2} + i\frac{\kappa^3 h^3}{6} + \frac{\kappa^4 h^4}{24} - i\frac{\kappa^5 h^5}{120} + \mathcal{O}(\kappa^6 h^6)\right)\right. \\
&\qquad \left. + \alpha^2\left(1 - i\kappa h - \frac{\kappa^2 h^2}{2} + i\frac{\kappa^3 h^3}{6} + \frac{\kappa^4 h^4}{24} - i\frac{\kappa^5 h^5}{120} + \mathcal{O}(\kappa^6 h^6)\right)^2\right] \\
&\cdot \frac{i}{\kappa^3 h^3}\left(1 + i\frac{3\kappa h}{2} - \kappa^2 h^2 - i\frac{3\kappa^3 h^3}{8} + \frac{19\kappa^4 h^4}{240} - i\frac{11\kappa^5 h^5}{80} + \mathcal{O}(\kappa^6 h^6)\right) \\
=&\frac{i}{2\kappa^3}\left(2 - i(\frac{\alpha^3}{3} - \frac{\alpha^2}{2} + \frac{\alpha}{6})\kappa^3 h^3 + (\frac{\alpha^4}{4} - \frac{\alpha^3}{2} + \frac{\alpha^2}{4} - \frac{1}{120})\kappa^4 h^4 - i(\frac{1}{240} - \frac{\alpha^3}{6} + \frac{\alpha^4}{4} - \frac{\alpha^5}{10})\kappa^5 h^5 + \mathcal{O}(\kappa^6 h^6)\right) \\
=&\frac{i}{\kappa^3}\left(1 - i(\frac{\alpha^3}{6} - \frac{\alpha^2}{4} + \frac{\alpha}{12})\kappa^3 h^3 + (\frac{\alpha^4}{8} - \frac{\alpha^3}{4} + \frac{\alpha^2}{8} - \frac{1}{240})\kappa^4 h^4 - i(\frac{1}{480} - \frac{\alpha^3}{12} + \frac{\alpha^4}{8} - \frac{\alpha^5}{20})\kappa^5 h^5 + \mathcal{O}(\kappa^6 h^6)\right) \\
=&\hat{v}^{(0)}_Q + \frac{\alpha}{12}(2\alpha^2 - 3\alpha + 1)h^3 + i\frac{1}{8}(\alpha^4 - 2\alpha^3 + \alpha^2 - \frac{1}{30})\kappa h^4 + (\frac{1}{480} - \frac{\alpha^3}{12} + \frac{\alpha^4}{8} - \frac{\alpha^5}{20})\kappa^2 h^5 + \mathcal{O}(\kappa^3 h^6).
\end{aligned}
$$

Appendix C

Derivation of smoothing modifications

In this appendix, we provide the details for deriving the explicit smoothing formulas with both the smoothing operator (3.35) proposed by [31] and the novel smoothing modification technique introduced in this thesis. We prove the results in Table 3.1 and the remaining formulas in Tables 4.1, 4.2 and 4.3 whose derivations are omitted in the main text. Moreover, we derive the fifth-order smoothing modifications of the Dirac delta, Heavisde, ramp and quadratic ramp functions. All the formulas with fifth-order modifications are listed in Tables C.1, C.2 and C.3. The formulas in these tables should be compared with the fourth-order modification formulas in Tables 4.1, 4.2 and 4.3.

C.1 Convolution-type smoothing [31]

In [31], the authors suggested some special smoothing operators Φ that satisfy both (4.7) and (4.8). The operator Φ_χ of order χ can be chosen such that its Fourier transform

$$\hat{\Phi}_\chi(\omega) = \frac{\mathcal{P}_\chi(\sin(\omega/2))}{(\omega/2)^\chi},$$

where $\mathcal{P}_\chi(\sin(\omega/2))$ are polynomials in $\sin(\omega/2)$ of lowest degree so that

$$\mathcal{P}_\chi(\sin(\omega/2)) = (\omega/2)^\chi + \mathcal{O}(\omega^{2\chi}). \tag{C.1}$$

In general, $\mathcal{P}_\chi$ has the form

$$\mathcal{P}_\chi(\sin(\omega/2)) = \sum_{j=0}^{\nu} c_j \sin^{\chi+j}\frac{\omega}{2},$$

where $\nu = \chi - 1$ for χ odd, and $\nu = \chi - 2$ for χ even, and the coefficients c_j are determined such that the order condition (C.1) is satisfied. In particular, we can show

$$\mathcal{P}_4(\sin(\omega/2)) = \sin^4\frac{\omega}{2} + \frac{2}{3}\sin^6\frac{\omega}{2},$$

$c_{\delta,j}^{[5]}$	$x_{-4}:x_0$	$x_{-3}:x_1$	$x_{-2}:x_2$	$x_{-1}:x_3$	$x_0:x_4$
$j<-4$	0	0	0	0	0
$j=-4$	$\frac{\alpha^4+2\alpha^3-\alpha^2-2\alpha}{24h}$	0	0	0	0
$j=-3$	$\frac{-\alpha^4-3\alpha^3+\alpha^2+3\alpha}{6h}$	$\frac{\alpha^4-2\alpha^3-\alpha^2+2\alpha}{24h}$	0	0	0
$j=-2$	$\frac{\alpha^4+4\alpha^3+\alpha^2-6\alpha}{4h}$	$\frac{-\alpha^4+\alpha^3+4\alpha^2-4\alpha}{6h}$	$\frac{\alpha^4-6\alpha^3+11\alpha^2-6\alpha}{24h}$	0	0
$j=-1$	$\frac{-\alpha^4-5\alpha^3-5\alpha^2+5\alpha}{6h}$	$\frac{\alpha^4-5\alpha^2}{4h}$	$\frac{-\alpha^4+5\alpha^3-5\alpha^2-5\alpha}{6h}$	$\frac{\alpha^4-10\alpha^3+35\alpha^2-50\alpha}{24h}$	0
$j=0$	$\frac{\alpha^4+6\alpha^3+11\alpha^2+6\alpha}{24h}$	$\frac{-\alpha^4-\alpha^3+4\alpha^2+4\alpha}{6h}$	$\frac{\alpha^4-4\alpha^3+\alpha^2+6\alpha}{4h}$	$\frac{-\alpha^4+9\alpha^3-26\alpha^2+24\alpha}{6h}$	$\frac{\alpha^4-14\alpha^3+71\alpha^2-154\alpha}{24h}$
$j=1$	0	$\frac{\alpha^4+2\alpha^3-\alpha^2-2\alpha}{24h}$	$\frac{-\alpha^4+3\alpha^3+\alpha^2-3\alpha}{6h}$	$\frac{\alpha^4-8\alpha^3+19\alpha^2-12\alpha}{4h}$	$\frac{-\alpha^4+13\alpha^3-59\alpha^2+107\alpha}{6h}$
$j=2$	0	0	$\frac{\alpha^4-2\alpha^3-\alpha^2+2\alpha}{24h}$	$\frac{-\alpha^4+7\alpha^3-14\alpha^2+8\alpha}{6h}$	$\frac{\alpha^4-12\alpha^3+49\alpha^2-78\alpha}{4h}$
$j=3$	0	0	0	$\frac{\alpha^4-6\alpha^3+11\alpha^2-6\alpha}{24h}$	$\frac{-\alpha^4+11\alpha^3-41\alpha^2+61\alpha}{6h}$
$j=4$	0	0	0	0	$\frac{\alpha^4-10\alpha^3+35\alpha^2-50\alpha}{24h}$
$j>4$	0	0	0	0	0
$\mathcal{C}_5$	$i\frac{\alpha^5+5\alpha^4+5\alpha^3-5\alpha^2-6\alpha}{120}$	$i\frac{\alpha^5-5\alpha^3+4\alpha}{120}$	$i\frac{\alpha^5-5\alpha^4+5\alpha^3+5\alpha^2-6\alpha}{120}$	$i\frac{\alpha^5-10\alpha^4+35\alpha^3-50\alpha^2+24\alpha}{120}$	$i\frac{\alpha^5-15\alpha^4+85\alpha^3-225\alpha^2+274\alpha}{120}$

Table C.1: Fifth-order smoothing modifications to discrete Dirac delta function (4.16) along with the leading order coefficient $\mathcal{C}_5$ of the $\mathcal{O}(\omega^5 h^5)$ term of the error in its Fourier transform representation.

$c_{H,j}^{[5]}$	$x_{-3}:x_0$	$x_{-2}:x_1$	$x_{-1}:x_2$	$x_0:x_3$
$j<-3$	0	0	0	0
$j=-3$	$\frac{30\alpha^4-60\alpha^2+11}{720}$	0	0	0
$j=-2$	$\frac{-30\alpha^4-40\alpha^3+120\alpha^2-21}{240}$	$\frac{30\alpha^4-120\alpha^3+120\alpha^2-19}{720}$	0	0
$j=-1$	$\frac{30\alpha^4+80\alpha^3-60\alpha^2-240\alpha+131}{240}$	$\frac{-30\alpha^4+80\alpha^3+60\alpha^2-240\alpha+109}{240}$	$\frac{30\alpha^4-240\alpha^3+660\alpha^2-720\alpha+251}{720}$	0
$j=0$	$\frac{-30\alpha^4-120\alpha^3-120\alpha^2+19}{720}$	$\frac{30\alpha^4-40\alpha^3-120\alpha^2+21}{240}$	$\frac{-30\alpha^4+200\alpha^3-360\alpha^2+59}{240}$	$\frac{30\alpha^4-360\alpha^3+1560\alpha^2-2880\alpha+1181}{720}$
$j=1$	0	$\frac{-30\alpha^4+60\alpha^2-11}{720}$	$\frac{30\alpha^4-160\alpha^3+180\alpha^2-29}{240}$	$\frac{-30\alpha^4+360\alpha^3-1140\alpha^2+1440\alpha-531}{240}$
$j=2$	0	0	$\frac{-30\alpha^4+120\alpha^3-120\alpha^2+19}{720}$	$\frac{30\alpha^4-280\alpha^3+840\alpha^2-960\alpha+341}{240}$
$j=3$	0	0	0	$\frac{-30\alpha^4+240\alpha^3-660\alpha^2+720\alpha-251}{720}$
$j>3$	0	0	0	0
$\mathcal{C}_5$	$\frac{6\alpha^5+15\alpha^4-10\alpha^3-30\alpha^2-17}{720}$	$\frac{6\alpha^5-15\alpha^4-10\alpha^3+30\alpha^2-40}{720}$	$\frac{6\alpha^5-45\alpha^4+110\alpha^3-90\alpha^2-9}{720}$	$\frac{6\alpha^5-75\alpha^4+350\alpha^3-750\alpha^2+720\alpha-260}{720}$

Table C.2: Fifth-order smoothing modifications to discrete Heaviside function (4.4) along with the leading order coefficient $\mathcal{C}_5$ of $\mathcal{O}(\kappa^4 h^5)$ term of the error in its Fourier transform representation.

	$c^{[5]}_{C,j}$			$c^{[5]}_{Q,j}$	
	$x_{-2}:x_0$	$x_{-1}:x_1$	$x_0:x_2$	$x_{-1}:x_0$	$x_0:x_1$
$j<-2$	0	0	0	0	0
$j=-2$	$\frac{(10\alpha^4-20\alpha^3+10\alpha-1)h}{240}$	0	0	0	0
$j=-1$	$\frac{(-10\alpha^4+60\alpha^2-60\alpha+11)h)}{120}$	$\frac{(10\alpha^4-60\alpha^3+120\alpha^2-90\alpha+19)h}{240}$	0	$\frac{(10\alpha^4-40\alpha^3+50\alpha^2-20a+1)h^2}{240}$	0
$j=0$	$\frac{(10\alpha^4+20\alpha^3-10\alpha-1)h}{240}$	$\frac{(-10\alpha^4+40\alpha^3-20\alpha+1)h}{120}$	$\frac{(10\alpha^4-100\alpha^3+360\alpha^2-310\alpha+59)h}{240}$	$\frac{(-10\alpha^4+10\alpha^2-1)h^2}{240}$	$\frac{(10\alpha^4-80\alpha^3+110\alpha^2-40\alpha+1)h^2}{240}$
$j=1$	0	$\frac{(10\alpha^4-20\alpha^3+10\alpha-1)h}{240}$	$\frac{(-10\alpha^4+80\alpha^3-180\alpha^2+140\alpha-29)h}{120}$	0	$\frac{(-10\alpha^4+40\alpha^3-50\alpha^2+20\alpha-1)h^2}{240}$
$j=2$	0	0	$\frac{(10\alpha^4-60\alpha^3+120\alpha^2-90\alpha+19)h}{240}$	0	0
$j>2$	0	0	0	0	0
$\mathcal{C}_5$	$i\frac{-6\alpha^5+20\alpha^3-11\alpha}{720}$	$i\frac{-6\alpha^5+30\alpha^4-40\alpha^3+19\alpha-3}{720}$	$i\frac{-6\alpha^5+60\alpha^4-220\alpha^3+360\alpha^2-251\alpha+54}{720}$	$\frac{-2\alpha^5+5\alpha^4-5\alpha^2-\alpha+1}{240}$	$\frac{-2\alpha^5+15\alpha^4-40\alpha^3+45\alpha^2-21\alpha+2}{240}$

Table C.3: Fifth-order smoothing modifications to discrete ramp and quadratic ramp functions (4.5), (4.6) along with the leading order coefficients $\mathcal{C}_5$ of the $\mathcal{O}(\kappa^3h^5)$ and $\mathcal{O}(\kappa^2h^5)$ terms of the errors, in their Fourier transform representations.

as we have seen in (3.35), and

$$\mathcal{P}_5(\sin(\omega/2)) = \sin^5\frac{\omega}{2} + \frac{5}{6}\sin^7\frac{\omega}{2} + \frac{47}{72}\sin^9\frac{\omega}{2}.$$

To derive the explicit formulas of the smoothed functions, we first obtain the inverse Fourier transform $\Phi_\chi = \mathcal{F}^{-1}[\hat{\Phi}_\chi]$, and then apply the convolution-type smoothing

$$M_h^{(\chi)}g(x) = \int \Phi_\chi(y)g(x-hy)dy, \tag{C.2}$$

where the integration limits are $(-\chi+\frac{1}{2}, \chi-\frac{1}{2})$ for χ odd and $(-\chi+1, \chi-1)$ for χ even.

Lemma C.1.1. *Let*

$$\hat{B}^{\chi,m}(\omega) = \frac{\sin^{\chi+m}(\omega/2)}{(\omega/2)^\chi},$$

where χ and m are nonnegative integers with $\chi \geq 1$ and m being even. The inverse Fourier transform $B^{\chi,m} = \mathcal{F}^{-1}[\hat{B^{\chi,m}}]$ is

$$B^{\chi,m}(x) = \frac{(-1)^{m/2}}{2^m(\chi-1)!}\sum_{k=0}^{\chi+m}\binom{\chi+m}{k}(-1)^k\left(x-k+\frac{\chi+m}{2}\right)_+^{\chi-1}, \tag{C.3}$$

where $(x-\cdot)_+^\chi = (x-\cdot)^\chi H(x-\cdot)$ is the shifted one-sided power function.

Proof. The lemma can be proved following similar procedures for the case when $m=0$ in [57]. □

Applying Lemma C.1.1, we can easily show that

$$\begin{aligned}\Phi_4(x) = \frac{1}{36}(&-(x-3)_+^3 + 12(x-2)_+^3 - 39(x-1)_+^3 + 56(x)_+^3 \\ &- 39(x+1)_+^3 + 12(x+2)_+^3 - (x+3)_+^3),\end{aligned} \tag{C.4}$$

which is a piece-wise polynomial of degree-3 and vanishes outside $(-3,3)$, and

$$\begin{aligned}\Phi_5(x) = \frac{1}{27648}\times \\ \big(&-47(x-4.5)_+^4 + 663(x-3.5)_+^4 - 4524(x-2.5)_+^4 + 14748(x-1.5)_+^4 - 25842(x-0.5)_+^4 \\ &+25842(x+0.5)_+^4 - 14748(x+1.5)_+^4 + 4524(x+2.5)_+^4 - 663(x+3.5)_+^4 + 47(x+4.5)_+^4\big)\,,\end{aligned} \tag{C.5}$$

which is a piece-wise polynomial of degree-4 and vanishes outside $(-4.5,4.5)$. Notice the wider support of $\Phi_5(x)$ in comparison to the $\Phi_4(x)$. Plugging (C.4) into (C.2), we obtain the fourth-order smoothing modification formulas given in Table 3.1. Plugging (C.5) into (C.2), we obtain the fifth-order smoothing modification formulas, and so on. Higher-order smoothings can be similarly derived. Once the convolving operators Φ's are explicitly written out using Lemma C.1.1, the final smoothing formulas can be trivially obtained from the integration (C.2) since Φ's are piece-wise polynomials. Due to the length of the final expressions of fifth order smoothing, we omit to display them in the thesis. We point out that the difference between the convolution-type smoothing and the new smoothing we proposed, in the number of grid values that need to be smoothed, becomes even more dramatic for higher order smoothings due to the growing support of the convolving operator Φ. For the fifth-order smoothing, for example, the convolution involves 9 grid points and requires the existence of 10 grid points in total, with 4 or 5 points on each side of the nonsmooth point depending on the location of the discontinuity. This poses constraints on the location of the nonsmooth point and on the grid size in space. In contrast, the novel smoothings in Tables C.1, C.2 and C.3 we proposed only requires to modify 4 points at most, providing more flexibility to the discretization.

C.2 The Dirac delta function

C.2.1 Fifth-order smoothing of the Dirac delta function

We already derived in detail a fourth-order smoothing for the discrete delta function in the main text. To obtain fifth-order smoothing, we just need modify one more grid value, e.g. at x_2, such that

$$\begin{cases} c^{[5]}_{\delta,-2} + c^{[5]}_{\delta,-1} + c^{[5]}_{\delta,0} + c^{[5]}_{\delta,1} + c^{[5]}_{\delta,2} &= 0, \\ (1+\alpha)c^{[5]}_{\delta,-2} + \alpha c^{[5]}_{\delta,-1} - (1-\alpha)c^{[5]}_{\delta,0} - (2-\alpha)c^{[5]}_{\delta,1} - (3-\alpha)c^{[5]}_{\delta,2} &= \frac{1}{h}(-\alpha\mathbb{1}_{\alpha<0.5} + (1-\alpha)\mathbb{1}_{\alpha\geq 0.5}), \\ -\frac{(1+\alpha)^2}{2}c^{[5]}_{\delta,-2} - \frac{\alpha^2}{2}c^{[5]}_{\delta,-1} - \frac{(1-\alpha)^2}{2}c^{[5]}_{\delta,0} - \frac{(2-\alpha)^2}{2}c^{[5]}_{\delta,1} - \frac{(3-\alpha)^2}{2}c^{[5]}_{\delta,2} &= \frac{1}{2h}(\alpha^2\mathbb{1}_{\alpha<0.5} + (1-\alpha)^2\mathbb{1}_{\alpha\geq 0.5}), \\ -\frac{(1+\alpha)^3}{6}c^{[5]}_{\delta,-2} - \frac{\alpha^3}{6}c^{[5]}_{\delta,-1} + \frac{(1-\alpha)^3}{6}c^{[5]}_{\delta,0} + \frac{(2-\alpha)^3}{6}c^{[5]}_{\delta,1} + \frac{(3-\alpha)^3}{6}c^{[5]}_{\delta,2} &= \frac{1}{6h}(\alpha^3\mathbb{1}_{\alpha<0.5} - (1-\alpha)^3\mathbb{1}_{\alpha\geq 0.5}), \\ \frac{(1+\alpha)^4}{24}c^{[5]}_{\delta,-2} + \frac{\alpha^4}{24}c^{[5]}_{\delta,-1} + \frac{(1-\alpha)^4}{24}c^{[5]}_{\delta,0} + \frac{(2-\alpha)^4}{24}c^{[5]}_{\delta,1} + \frac{(3-\alpha)^4}{24}c^{[5]}_{\delta,2} &= \frac{1}{24h}(-\alpha^4\mathbb{1}_{\alpha<0.5} - (1-\alpha)^4\mathbb{1}_{\alpha\geq 0.5}), \end{cases}$$

which gives the values of $c_{-2}^{[5]}$ to $c_{2}^{[5]}$

$$\begin{cases} c_{\delta,-2}^{[5]} = \frac{1}{24h}(\alpha^4 - 6\alpha^3 + 11\alpha^2 - 6\alpha), \\ c_{\delta,-1}^{[5]} = \frac{1}{6h}(-\alpha^4 + 5\alpha^3 - 5\alpha^2 - 5\alpha) + \frac{1}{h}\mathbb{1}_{\alpha\geq 0.5}, \\ c_{\delta,0}^{[5]} = \frac{1}{4h}(\alpha^4 - 4\alpha^3 + \alpha^2 + 6\alpha) - \frac{1}{h}\mathbb{1}_{\alpha\geq 0.5}, \\ c_{\delta,1}^{[5]} = \frac{1}{6h}(-\alpha^4 + 3\alpha^3 + \alpha^2 - 3\alpha), \\ c_{\delta,2}^{[5]} = \frac{1}{24h}(\alpha^4 - 2\alpha^3 - \alpha^2 + 2\alpha). \end{cases} \tag{C.6}$$

Let $c_{\delta,j}^{[5]} = 0$ for $j \geq 3$ and $j \leq -3$. We get a fifth-order smoothed discretization of the delta initial condition $\delta_{\alpha}^{[5]}(x_j) \equiv \delta_{\alpha}(x_j) + c_{\delta,j}^{[5]}$, and in the Fourier domain

$$\hat{\delta}_{\alpha}^{[5]} = \hat{v}_{\delta}(t=0) + i\frac{\alpha(\alpha-1)(\alpha^3 - 4\alpha^2 + \alpha + 6)}{120}\omega^5 h^5 + \mathcal{O}(\omega^6 h^6).$$

C.2.2 Smoothing of an alternative discrete Dirac delta function

We show in this section that the final smoothed discretization does not depend on the original discretization, due to linearity of the smoothing procedure. In second-order methods, the delta initial condition is typically discretized as [25, 7]

$$\delta_{\alpha}(x_j) \equiv \begin{cases} \frac{1-\alpha}{h}, & j = -1, \\ \frac{\alpha}{h}, & j = 0, \\ 0, & \text{else}, \end{cases} \tag{C.7}$$

which is equivalent to the second-order smoothing in [31], whose semi-discrete Fourier transform is

$$\hat{v}_{\delta,h,\alpha}^{(0)} \equiv h\sum_{j=-\infty}^{\infty} e^{-i\omega x_j}\delta_{\alpha}(x_j) = (1-\alpha)e^{i\omega\alpha h} + \alpha e^{-i\omega(1-\alpha)h}.$$

Applying Taylor expansion and using the fact that $\hat{v}_{\delta}^{(0)} = 1$, we have

$$\hat{v}_{\delta,h,\alpha}^{(0)} = \hat{v}_{\delta}^{(0)} + \alpha(1-\alpha)\left(-\frac{1}{2}\omega^2 h^2 - i\frac{1}{6}(2\alpha-1)\omega^3 h^3 + \frac{1}{24}(3\alpha^2 - 3\alpha + 1)\omega^4 h^4 + \mathcal{O}(\omega^5 h^5)\right). \tag{C.8}$$

Therefore, we can see that when $\alpha = 1$, the discrete version $\hat{v}_{\delta,h,\alpha}^{(0)} = 1$ approximates the true Fourier transform of delta function exactly.

In order to cancel out the low-order terms in (C.8), we need to have

$$\begin{aligned} &h\sum_{j=-2}^{1} c_{\delta,j}^{[4]} e^{-i\omega(j+(1-\alpha))h} \\ =&\frac{1}{2}\alpha(1-\alpha)\omega^2 h^2 + i\frac{1}{6}\alpha(1-\alpha)(2\alpha-1)\omega^3 h^3 - \frac{1}{24}\alpha(1-\alpha)(3\alpha^2 - 3\alpha + 1)\omega^4 h^4 + \mathcal{O}(\omega^5 h^5). \end{aligned} \tag{C.9}$$

Applying Taylor expansion to the left-hand side of (C.9) and combining terms, we have

$$\begin{aligned} h\sum_{j=-2}^{1} c_{\delta,j}^{[4]} e^{-i\omega(j+(1-\alpha))h} &= (c_{\delta,-2}^{[4]} + c_{\delta,-1}^{[4]} + c_{\delta,0}^{[4]} + c_{\delta,1}^{[4]})h \\ &+ i((1+\alpha)c_{\delta,-2}^{[4]} + \alpha c_{\delta,-1}^{[4]} - (1-\alpha)c_{\delta,0}^{[4]} - (2-\alpha)c_{\delta,1}^{[4]})\omega h^2 \\ &+ \left(-\frac{(1+\alpha)^2}{2}c_{\delta,-2}^{[4]} - \frac{\alpha^2}{2}c_{\delta,-1}^{[4]} - \frac{(1-\alpha)^2}{2}c_{\delta,0}^{[4]} - \frac{(2-\alpha)^2}{2}c_{\delta,1}^{[4]}\right)\omega^2 h^3 \\ &+ i\left(-\frac{(1+\alpha)^3}{6}c_{\delta,-2}^{[4]} - \frac{\alpha^3}{6}c_{\delta,-1}^{[4]} + \frac{(1-\alpha)^3}{6}c_{\delta,0}^{[4]} + \frac{(2-\alpha)^3}{6}c_{\delta,1}^{[4]}\right)\omega^3 h^4 + \cdots . \end{aligned} \tag{C.10}$$

for some coefficients $c_{\delta,-2}^{[4]}, c_{\delta,-1}^{[4]}, c_{\delta,0}^{[4]}, c_{\delta,1}^{[4]}$ to be determined. To match the terms between the right-hand sides of (C.9) and (C.10), we need

$$\begin{cases} c_{\delta,-2}^{[4]} + c_{\delta,-1}^{[4]} + c_{\delta,0}^{[4]} + c_{\delta,1}^{[4]} & = 0, \\ (1+\alpha)c_{\delta,-2}^{[4]} + \alpha c_{\delta,-1}^{[4]} - (1-\alpha)c_{\delta,0}^{[4]} - (2-\alpha)c_{\delta,1}^{[4]} & = 0, \\ -\frac{(1+\alpha)^2}{2}c_{\delta,-2}^{[4]} - \frac{\alpha^2}{2}c_{\delta,-1}^{[4]} - \frac{(1-\alpha)^2}{2}c_{\delta,0}^{[4]} - \frac{(2-\alpha)^2}{2}c_{\delta,1}^{[4]} & = \frac{1}{2h}\alpha(1-\alpha), \\ -\frac{(1+\alpha)^3}{6}c_{\delta,-2}^{[4]} - \frac{\alpha^3}{6}c_{\delta,-1}^{[4]} + \frac{(1-\alpha)^3}{6}c_{\delta,0}^{[4]} + \frac{(2-\alpha)^3}{6}c_{\delta,1}^{[4]} & = \frac{1}{6h}\alpha(1-\alpha)(2\alpha-1). \end{cases}$$

Solving the equations for $c_{\delta,-2}^{[4]}, c_{\delta,-1}^{[4]}, c_{\delta,0}^{[4]}$ and $c_{\delta,1}^{[4]}$, we get

$$\begin{cases} c_{\delta,-2}^{[4]} & = -\frac{1}{6h}(\alpha^3 - 3\alpha^2 + 2\alpha), \\ c_{\delta,-1}^{[4]} & = \frac{1}{2h}(\alpha^3 - 2\alpha^2 + \alpha), \\ c_{\delta,0}^{[4]} & = \frac{1}{2h}(-\alpha^3 + \alpha^2), \\ c_{\delta,1}^{[4]} & = \frac{1}{6h}(\alpha^3 - \alpha). \end{cases}$$

Let $c_{\delta,j}^{[4]} = 0$ for $j \geq 2$ and $j \leq -3$. We get a fourth-order smoothed discretization of the delta initial condition $\delta_\alpha^{[4]}(x_j) \equiv \delta_\alpha(x_j) + c_{\delta,j}^{[4]}$, where we see that $c_{\delta,j}^{[4]} = c_{\delta,j}^{[4]}(\alpha)$ is a function of α.
For the fifth-order smoothing, we need

$$\begin{aligned} h\sum_{j=-2}^{2} c_{\delta,j}^{[5]} e^{-i\omega(j+(1-\alpha))h} &= (c_{\delta,-2}^{[5]} + c_{\delta,-1}^{[5]} + c_{\delta,0}^{[5]} + c_{\delta,1}^{[5]} + c_{\delta,2}^{[5]})h \\ &+ i((1+\alpha)c_{\delta,-2}^{[5]} + \alpha c_{\delta,-1}^{[5]} - (1-\alpha)c_{\delta,0}^{[5]} - (2-\alpha)c_{\delta,1}^{[5]} - (3-\alpha)c_{\delta,2}^{[5]})\omega h^2 \\ &+ \left(-\frac{(1+\alpha)^2}{2}c_{\delta,-2}^{[5]} - \frac{\alpha^2}{2}c_{\delta,-1}^{[5]} - \frac{(1-\alpha)^2}{2}c_{\delta,0}^{[5]} - \frac{(2-\alpha)^2}{2}c_{\delta,1}^{[5]} - \frac{(3-\alpha)^2}{2}c_{\delta,2}^{[5]}\right)\omega^2 h^3 \\ &+ i\left(-\frac{(1+\alpha)^3}{6}c_{\delta,-2}^{[5]} - \frac{\alpha^3}{6}c_{\delta,-1}^{[5]} + \frac{(1-\alpha)^3}{6}c_{\delta,0}^{[5]} + \frac{(2-\alpha)^3}{6}c_{\delta,1}^{[5]} + \frac{(3-\alpha)^3}{6}c_{\delta,2}^{[5]}\right)\omega^3 h^4 \\ &+ \left(\frac{(1+\alpha)^4}{24}c_{\delta,-2}^{[5]} + \frac{\alpha^4}{24}c_{\delta,-1}^{[5]} + \frac{(1-\alpha)^4}{24}c_{\delta,0}^{[5]} + \frac{(2-\alpha)^4}{24}c_{\delta,1}^{[5]} + \frac{(3-\alpha)^4}{24}c_{\delta,2}^{[5]}\right)\omega^4 h^5 + \cdots . \end{aligned} \tag{C.11}$$

such that

$$\begin{cases} c^{[5]}_{\delta,-2} + c^{[5]}_{\delta,-1} + c^{[5]}_{\delta,0} + c^{[5]}_{\delta,1} + c^{[5]}_{\delta,2} & = 0, \\ (1+\alpha)c^{[5]}_{\delta,-2} + \alpha c^{[5]}_{\delta,-1} - (1-\alpha)c^{[5]}_{\delta,0} - (2-\alpha)c^{[5]}_{\delta,1} - (3-\alpha)c^{[5]}_{\delta,2} & = 0, \\ -\frac{(1+\alpha)^2}{2}c^{[5]}_{\delta,-2} - \frac{\alpha^2}{2}c^{[5]}_{\delta,-1} - \frac{(1-\alpha)^2}{2}c^{[5]}_{\delta,0} - \frac{(2-\alpha)^2}{2}c^{[5]}_{\delta,1} - \frac{(3-\alpha)^2}{2}c^{[5]}_{\delta,2} & = \frac{1}{2h}\alpha(1-\alpha), \\ -\frac{(1+\alpha)^3}{6}c^{[5]}_{\delta,-2} - \frac{\alpha^3}{6}c^{[5]}_{\delta,-1} + \frac{(1-\alpha)^3}{6}c^{[5]}_{\delta,0} + \frac{(2-\alpha)^3}{6}c^{[5]}_{\delta,1} + \frac{(3-\alpha)^3}{6}c^{[5]}_{\delta,2} & = \frac{1}{6h}\alpha(1-\alpha)(2\alpha-1), \\ \frac{(1+\alpha)^4}{24}c^{[5]}_{\delta,-2} + \frac{\alpha^4}{24}c^{[5]}_{\delta,-1} + \frac{(1-\alpha)^4}{24}c^{[5]}_{\delta,0} + \frac{(2-\alpha)^4}{24}c^{[5]}_{\delta,1} + \frac{(3-\alpha)^4}{24}c^{[5]}_{\delta,2} & = -\frac{1}{24h}\alpha(1-\alpha)(3\alpha^2-3\alpha+1), \end{cases}$$

which gives the values of $c^{[5]}_{-2}$ to $c^{[5]}_{2}$

$$\begin{cases} c^{[5]}_{\delta,-2} = \frac{1}{24h}(\alpha^4 - 6\alpha^3 + 11\alpha^2 - 6\alpha), \\ c^{[5]}_{\delta,-1} = \frac{1}{6h}(-\alpha^4 + 5\alpha^3 - 5\alpha^2 + \alpha), \\ c^{[5]}_{\delta,0} = \frac{1}{4h}(\alpha^4 - 4\alpha^3 + \alpha^2 + 2\alpha), \\ c^{[5]}_{\delta,1} = \frac{1}{6h}(-\alpha^4 + 3\alpha^3 + \alpha^2 - 3\alpha), \\ c^{[5]}_{\delta,2} = \frac{1}{24h}(\alpha^4 - 2\alpha^3 - \alpha^2 + 2\alpha). \end{cases} \tag{C.12}$$

Let $c^{[5]}_{\delta,j} = 0$ for $j \geq 3$ and $j \leq -3$. We get a fifth-order smoothed discretization of the delta initial condition $\delta^{[5]}_{\alpha}(x_j) \equiv \delta_\alpha(x_j) + c^{[5]}_{\delta,j}$. From the results of both the fourth- and fifth-order smoothing to (C.7), we see that its final smoothed values are exactly the same as the final fourth- and fifth-order smoothed values to (4.3). This supports the linearity of the smoothing procedure.

C.3 The Heaviside function

For the discrete Heaviside function (4.4), consider adding modifications $c^{[4]}_{H,-1}$, $c^{[4]}_{H,0}$, $c^{[4]}_{H,1}$ to $H_\alpha(x_{-1})$, $H_\alpha(x_0)$, $H_\alpha(x_1)$, respectively. In order to cancel out the low order terms in (4.10), we need to have

$$\begin{aligned} h\sum_{j=-1}^{1} c^{[4]}_{H,j} e^{-i\kappa(j+(1-\alpha))h} = &-\left(\alpha - \frac{1}{2}\right)h - i\frac{1}{2}\left(\left(\alpha-\frac{1}{2}\right)^2 - \frac{1}{12}\right)\kappa h^2 \\ &- \frac{\alpha(1-\alpha)(2\alpha-1)}{12}\kappa^2 h^3 - i\left(-\frac{\alpha^4}{24} + \frac{\alpha^3}{12} - \frac{\alpha^2}{24} + \frac{1}{720}\right)\kappa^3 h^4 + \mathcal{O}(\kappa^4 h^5). \end{aligned} \tag{C.13}$$

Similarly, we can apply Taylor expansion to the left-hand side of (C.13) and combine terms to get

$$\begin{aligned} h\sum_{j=-1}^{1} c^{[4]}_{H,j} e^{-i\kappa(j+(1-\alpha))h} = &(c^{[4]}_{H,-1} + c^{[4]}_{H,0} + c^{[4]}_{H,1})h \\ &+ i(\alpha c^{[4]}_{H,-1} - (1-\alpha)c^{[4]}_{H,0} - (2-\alpha)c^{[4]}_{H,1})\kappa h^2 \\ &+ \left(-\frac{\alpha^2}{2}c^{[4]}_{H,-1} - \frac{(1-\alpha)^2}{2}c^{[4]}_{H,0} - \frac{(2-\alpha)^2}{2}c^{[4]}_{H,1}\right)\kappa^2 h^3 + \cdots. \end{aligned} \tag{C.14}$$

To match the terms between the right-hand sides of (C.13)] and (C.14), we need

$$\begin{cases} c_{H,-1}^{[4]} + c_{H,0}^{[4]} + c_{H,1}^{[4]} & = -\left(\alpha - \frac{1}{2}\right), \\ \alpha c_{H,-1}^{[4]} - (1-\alpha) c_{H,0}^{[4]} - (2-\alpha) c_{H,1}^{[4]} & = -\frac{1}{2}\left(\left(\alpha - \frac{1}{2}\right)^2 - \frac{1}{12}\right), \\ -\frac{\alpha^2}{2} c_{H,-1}^{[4]} - \frac{(1-\alpha)^2}{2} c_{H,0}^{[4]} - \frac{(2-\alpha)^2}{2} c_{H,1}^{[4]} & = -\frac{1}{12}\alpha(1-\alpha)(2\alpha-1), \end{cases}$$

for some coefficients $c_{H,-1}^{[4]}, c_{H,0}^{[4]}, c_{H,1}^{[4]}$ to be determined. Solving the equations for $c_{H,-1}^{[4]}, c_{H,0}^{[4]}$ and $c_{H,1}^{[4]}$, we get

$$\begin{cases} c_{H,-1}^{[4]} & = -\frac{1}{6}\alpha^3 + \frac{3}{4}\alpha^2 - \alpha + \frac{3}{8}, \\ c_{H,0}^{[4]} & = \frac{1}{3}\alpha^3 - \alpha^2 + \frac{1}{6}, \\ c_{H,1}^{[4]} & = -\frac{1}{6}\alpha^3 + \frac{1}{4}\alpha^2 - \frac{1}{24}. \end{cases}$$

Let $c_{H,j}^{[4]} = 0$ for $j \geq 1$ and $j \leq -1$. We get a fourth-order discretization of the Heaviside initial condition $H_\alpha^{[4]}(x_j) \equiv H_\alpha(x_j) + c_{H,j}^{[4]}$ for a general α alignment. In particular, plugging $\alpha = 1$ into the formulas, we get the initial condition modification when the point of nonsmoothness $x = 0$ lies exact on grid point, as shown in Appendix B.2.1 in (B.2). When the point of nonsmoothness remain midway between two grid points, the modification is given by (B.1) in Appendix B.2.1.

Again, we can modify one additional grid value to get fifth-order smoothing so that the coefficient of $\mathcal{O}(h^4)$ term is independent of the alignment α. Suppose we choose to modify the value at x_{-2}, then

$$\begin{aligned} h\sum_{j=-2}^{1} c_{H,j}^{[5]} e^{-i\kappa(j+(1-\alpha))h} &= (c_{H,-2}^{[5]} + c_{H,-1}^{[5]} + c_{H,0}^{[5]} + c_{H,1}^{[5]})h \\ &+ i((1+\alpha)c_{H,-2}^{[5]} + \alpha c_{H,-1}^{[5]} - (1-\alpha)c_{H,0}^{[5]} - (2-\alpha)c_{H,1}^{[5]})\kappa h^2 \\ &+ \left(-\frac{(1+\alpha)^2}{2} c_{H,-2}^{[5]} - \frac{\alpha^2}{2} c_{H,-1}^{[5]} - \frac{(1-\alpha)^2}{2} c_{H,0}^{[5]} - \frac{(2-\alpha)^2}{2} c_{H,1}^{[5]}\right)\kappa^2 h^3 \\ &+ i\left(-\frac{(1+\alpha)^3}{6} c_{H,-2}^{[5]} - \frac{\alpha^3}{6} c_{H,-1}^{[5]} + \frac{(1-\alpha)^3}{6} c_{H,0}^{[5]} + \frac{(2-\alpha)^3}{6} c_{H,1}^{[5]}\right)\kappa^3 h^4 + \cdots. \end{aligned} \tag{C.15}$$

To match the terms between (C.13) and (C.15), we need

$$\begin{cases} c_{H,-2}^{[5]} + c_{H,-1}^{[5]} + c_{H,0}^{[5]} + c_{H,1}^{[5]} & = -\left(\alpha - \frac{1}{2}\right), \\ (1+\alpha)c_{H,-2}^{[5]} + \alpha c_{H,-1}^{[5]} - (1-\alpha)c_{H,0}^{[5]} - (2-\alpha)c_{H,1}^{[5]} & = -\frac{1}{2}\left(\left(\alpha - \frac{1}{2}\right)^2 - \frac{1}{12}\right), \\ -\frac{(1+\alpha)^2}{2} c_{H,-2}^{[5]} - \frac{\alpha^2}{2} c_{H,-1}^{[5]} - \frac{(1-\alpha)^2}{2} c_{H,0}^{[5]} - \frac{(2-\alpha)^2}{2} c_{H,1}^{[5]} & = -\frac{1}{12}\alpha(1-\alpha)(2\alpha-1), \\ -\frac{(1+\alpha)^3}{6} c_{H,-2}^{[5]} - \frac{\alpha^3}{6} c_{H,-1}^{[5]} + \frac{(1-\alpha)^3}{6} c_{H,0}^{[5]} + \frac{(2-\alpha)^3}{6} c_{H,1}^{[5]} & = \frac{\alpha^4}{24} - \frac{\alpha^3}{12} + \frac{\alpha^2}{24} - \frac{1}{720}, \end{cases}$$

which gives the values of $c_{H,-2}^{[5]}$ to $c_{H,1}^{[5]}$. Let $c_{H,j}^{[5]} = 0$ for $j \leq -3$ and $j \geq 2$. and we get a fifth-order smoothed discretization of the Heaviside initial condition $H_\alpha^{[5]}(x_j) \equiv H_\alpha(x_j) + c_{H,j}^{[5]}$.

C.4 The ramp function

For the discrete ramp initial condition (4.5), consider adding modifications $c^{[4]}_{C,-1}$, $c^{[4]}_{C,0}$ to $C_\alpha(x_{-1})$, $C_\alpha(x_0)$, respectively. In order to cancel out the low order terms in (4.11), we need

$$\begin{aligned} & h\sum_{j=-1}^{0} c^{[4]}_{C,j} e^{-i\kappa(j+(1-\alpha))h} \\ =& \frac{1}{12}(6\alpha^2-6\alpha+1)h^2 - i\frac{1}{6}\alpha(1-\alpha)(2\alpha-1)\kappa h^3 - \frac{1}{240}\left(30\alpha^2(1-\alpha)^2-1\right)\kappa^2h^4 + \mathcal{O}(\kappa^3h^5), \end{aligned} \tag{C.16}$$

Applying Taylor expansion to the left-hand side of (C.16), we get

$$h\sum_{j=-1}^{0} c^{[4]}_{C,j} e^{-i\kappa(j+(1-\alpha))h} = (c^{[4]}_{C,-1}+c^{[4]}_{C,0})h + i(\alpha c^{[4]}_{C,-1} - (1-\alpha)c^{[4]}_{C,0})\kappa h^2 + \mathcal{O}(\kappa^2h^3). \tag{C.17}$$

To match the terms between the right-hand sides of (C.16) and (C.17), we need to have

$$\begin{cases} c^{[4]}_{C,-1} + c^{[4]}_{C,0} & = \frac{1}{2}(\alpha^2-\alpha+1/6)h, \\ \alpha c^{[4]}_{C,-1} - (1-\alpha)c^{[4]}_{C,0} & = -\frac{1}{6}\alpha(1-\alpha)(2\alpha-1)h. \end{cases}$$

Solving the equations for $c^{[4]}_{C,-1}$ and $c^{[4]}_{C,0}$, we get

$$\begin{cases} c^{[4]}_{C,-1} & = \frac{h}{12}(1-2\alpha^3+6\alpha^2-5\alpha), \\ c^{[4]}_{C,0} & = \frac{h}{12}(2\alpha^3-\alpha). \end{cases}$$

Therefore, we get a fourth-order discretization of the ramp (call-type) initial condition $C^{[4]}_\alpha(x_j) \equiv \max(x_j,0) + c^{[4]}_{C,j}(x_j)$. for any general α alignment. In Appendix B.2.2, Equations (B.3) and (B.4), we develop the discretization modifications for the special alignment values $\alpha = 1$ and 0.5, respectively.

To get one order higher smoothness, we add modification to one more point

$$\begin{aligned} h\sum_{j=-1}^{1} c^{[5]}_{C,j} e^{-i\kappa(j+(1-\alpha))h} &= (c_{-1}+c_0+c_1)h \\ &\quad + i(\alpha c^{[5]}_{C,-1} - (1-\alpha)c^{[5]}_{C,0} - (2-\alpha)c^{[5]}_{C,1})\kappa h^2 \\ &\quad + \left(-\frac{\alpha^2}{2}c^{[5]}_{C,-1} - \frac{(1-\alpha)^2}{2}c^{[5]}_{C,0} - \frac{(2-\alpha)^2}{2}c^{[5]}_{C,1}\right)\kappa^2h^3 + \cdots. \end{aligned} \tag{C.18}$$

To match the terms between the right-hand sides of (C.16) and (C.18), we need to have

$$\begin{cases} c^{[5]}_{C,-1} + c^{[5]}_{C,0} + c^{[5]}_{C,1} & = \frac{1}{2}(\alpha^2-\alpha+1/6)h, \\ \alpha c^{[5]}_{C,-1} - (1-\alpha)c^{[5]}_{C,0} - (2-\alpha)c^{[5]}_{C,1} & = -\frac{1}{6}\alpha(1-\alpha)(2\alpha-1)h, \\ -\frac{\alpha^2}{2}c^{[5]}_{C,-1} - \frac{(1-\alpha)^2}{2}c^{[5]}_{C,0} - \frac{(2-\alpha)^2}{2}c^{[5]}_{C,1} & = -\frac{1}{240}(30\alpha^2(1-\alpha)^2-1)h. \end{cases}$$

Solving the equations for $c^{[5]}_{C,-1}$, $c^{[5]}_{C,0}$ and $c^{[5]}_{C,1}$ gives

$$\begin{cases} c^{[5]}_{C,-1} = \frac{h}{240}\left(10\alpha^4 - 60\alpha^3 + 120\alpha^2 - 90\alpha + 19\right), \\ c^{[5]}_{C,0} = \frac{h}{120}\left(-10a^4 + 40\alpha^3 - 20\alpha + 1\right), \\ c^{[5]}_{C,1} = \frac{h}{240}\left(10\alpha^4 - 20\alpha^3 + 10\alpha - 1\right). \end{cases} \tag{C.19}$$

Let $c^{[5]}_{C,j} = 0$ for $j \le -2$ and $j \ge 2$. We get a fifth-order smoothed discretization of the ramp initial condition $C^{[5]}_{\alpha}(x_j) \equiv \max(x_j, 0) + c^{[5]}_{C,j}$.

C.5 The quadratic ramp function

For the quadratic ramp function, we repeat here the semi-discrete Fourier transform (4.12) of the discretization

$$\hat{v}^{(0)}_{Q,h,\alpha} = \hat{v}_Q(t=0) + \frac{\alpha}{12}\left(2\alpha^2 - 3\alpha + 1\right)h^3 + i\frac{\alpha^2}{8}\left(\alpha^2 - 2\alpha + 1\right)\kappa h^4 + \mathcal{O}(\kappa^2 h^5). \tag{C.20}$$

The fourth-order convergence can be achieved by adding a single modification to any point, such that

$$hc^{[4]}_{Q,j}e^{-i(j+(1-\alpha))\kappa h} = -\frac{\alpha}{12}\left(2\alpha^2 - 3\alpha + 1\right)h^3 + \mathcal{O}(\kappa h^4),$$

which gives

$$c^{[4]}_{Q,j} = -\frac{\alpha}{12}\left(2\alpha^2 - 3\alpha + 1\right)h^2,$$

Let $c^{[4]}_{Q,0} = 0$ for $j \ne 0$. We have $Q^{[4]}_{\alpha}(x_j) = Q_{\alpha}(x_j) + c^{[4]}_{Q,j}$. We also see that the same value $c^{[4]}_{Q,0}$ can be added to any point to achieve the fourth-order accuracy, but will result in different leading order coefficients in the errors. To get rid of the dependency on the alignment α, we can modify one more point and let $Q^{[5]}_{\alpha}(x_j) = Q_{\alpha}(x_j) + c^{[5]}_{Q,j}$, where

$$\begin{cases} c^{[5]}_{Q,-1} = \frac{1}{24}(\alpha^4 - 4\alpha^3 + 5\alpha^2 - 2\alpha)h^2, \\ c^{[5]}_{Q,0} = \frac{1}{24}\alpha^2(1-\alpha^2)h^2, \end{cases}$$

and $c^{[5]}_{Q,j} = 0$ for $j \le -2$ and $j \ge 1$.

Bibliography

[1] D. M. Anderson, G. B. McFadden, and A. A. Wheeler. "Diffuse-interface methods in fluid mechanics". In: *Annual review of fluid mechanics, Vol. 30*. Vol. 30. Annu. Rev. Fluid Mech. Annual Reviews, Palo Alto, CA, 1998, pp. 139–165. ISBN: 0-8243-0730-5. DOI: 10.1146/annurev.fluid.30.1.139. URL: https://doi.org/10.1146/annurev.fluid.30.1.139.

[2] Uri M. Ascher and Linda R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998, pp. xviii+314. ISBN: 0-89871-412-5. DOI: 10.1137/1.9781611971392. URL: https://doi.org/10.1137/1.9781611971392.

[3] Fischer Black and Myron Scholes. "The pricing of options and corporate liabilities". In: *J. Polit. Econ.* 81.3 (1973), pp. 637–654. ISSN: 0022-3808,1537-534X. DOI: 10.1086/260062. URL: https://doi.org/10.1086/260062.

[4] J. H. Bramble and B. E. Hubbard. "On a finite difference analogue of an elliptic boundary problem which is neither diagonally dominant nor of non-negative type". In: *J. Math. and Phys.* 43 (1964), pp. 117–132. ISSN: 0097-1421.

[5] Xinfu Chen and John Chadam. "A mathematical analysis of the optimal exercise boundary for American put options". In: *SIAM J. Math. Anal.* 38.5 (2007), pp. 1613–1641. ISSN: 0036-1410,1095-7154. DOI: 10.1137/S0036141003437708. URL: https://doi.org/10.1137/S0036141003437708.

[6] Christina Christara and Duy-Minh Dang. "Adaptive and high-order methods for valuing American options". In: *J. Comput. Finance* 14.4 (2011), pp. 73–113.

[7] Christina C. Christara and Nat Chun-Ho Leung. "Analysis of quantization error in financial pricing via finite difference methods". In: *SIAM J. Numer. Anal.* 56.3 (2018), pp. 1731–1757. ISSN: 0036-1429,1095-7170. DOI: 10.1137/17M1139655. URL: https://doi.org/10.1137/17M1139655.

[8] Philippe G. Ciarlet. "Discrete maximum principle for finite-difference operators". In: *Aequationes Math.* 4 (1970), pp. 338–352. ISSN: 0001-9054,1420-8903. DOI: 10.1007/BF01844166. URL: https://doi.org/10.1007/BF01844166.

[9] Nigel Clarke and Kevin Parrott. "Multigrid for American option pricing with stochastic volatility". In: *Applied Mathematical Finance* 6.3 (1999), pp. 177–195.

[10] John Crank. *Free and moving boundary problems*. Oxford Science Publications. The Clarendon Press, Oxford University Press, New York, 1984, pp. x+425. ISBN: 0-19-853357-8.

[11] J. N. Dewynne et al. “Some mathematical results in the pricing of American options”. In: *European J. Appl. Math.* 4.4 (1993), pp. 381–398. ISSN: 0956-7925,1469-4425. DOI: 10.1017/S0956792500001194. URL: https://doi.org/10.1017/S0956792500001194.

[12] Mehzabeen Jumanah Dilloo and Désiré Yannick Tangman. “A high-order finite difference method for option valuation”. In: *Comput. Math. Appl.* 74.4 (2017), pp. 652–670. ISSN: 0898-1221,1873-7668. DOI: 10.1016/j.camwa.2017.05.006. URL: https://doi.org/10.1016/j.camwa.2017.05.006.

[13] Bertram Düring and Michel Fournié. “High-order compact finite difference scheme for option pricing in stochastic volatility models”. In: *J. Comput. Appl. Math.* 236.17 (2012), pp. 4462–4473. ISSN: 0377-0427,1879-1778. DOI: 10.1016/j.cam.2012.04.017. URL: https://doi.org/10.1016/j.cam.2012.04.017.

[14] Bertram Düring and Christof Heuer. “High-order compact schemes for parabolic problems with mixed derivatives in multiple space dimensions”. In: *SIAM J. Numer. Anal.* 53.5 (2015), pp. 2113–2134. ISSN: 0036-1429,1095-7170. DOI: 10.1137/140974833. URL: https://doi.org/10.1137/140974833.

[15] Bertram Düring and Alexander Pitkin. “High-order compact finite difference scheme for option pricing in stochastic volatility jump models”. In: *J. Comput. Appl. Math.* 355 (2019), pp. 201–217. ISSN: 0377-0427,1879-1778. DOI: 10.1016/j.cam.2019.01.043. URL: https://doi.org/10.1016/j.cam.2019.01.043.

[16] C. M. Elliott and J. R. Ockendon. *Weak and variational methods for moving boundary problems*. Vol. 59. Research Notes in Mathematics. Pitman (Advanced Publishing Program), Boston, Mass.-London, 1982, pp. iii+213. ISBN: 0-273-08503-4.

[17] F. Fang and C. W. Oosterlee. “A novel pricing method for European options based on Fourier-cosine series expansions”. In: *SIAM J. Sci. Comput.* 31.2 (2008), pp. 826–848. ISSN: 1064-8275,1095-7197. DOI: 10.1137/080718061. URL: https://doi.org/10.1137/080718061.

[18] Ronald P. Fedkiw et al. “A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)”. In: *J. Comput. Phys.* 152.2 (1999), pp. 457–492. ISSN: 0021-9991,1090-2716. DOI: 10.1006/jcph.1999.6236. URL: https://doi.org/10.1006/jcph.1999.6236.

[19] Bengt Fornberg. “Calculation of weights in finite difference formulas”. In: *SIAM Rev.* 40.3 (1998), pp. 685–691. ISSN: 0036-1445,1095-7200. DOI: 10.1137/S0036144596322507. URL: https://doi.org/10.1137/S0036144596322507.

[20] Bengt Fornberg and Rita Meyer-Spasche. “A finite difference procedure for a class of free boundary problems”. In: *J. Comput. Phys.* 102.1 (1992), pp. 72–77. ISSN: 0021-9991,1090-2716. DOI: 10.1016/S0021-9991(05)80006-3. URL: https://doi.org/10.1016/S0021-9991(05)80006-3.

[21] P. A. Forsyth and K. R. Vetzal. “Quadratic convergence for valuing American options using a penalty method”. In: *SIAM J. Sci. Comput.* 23.6 (2002), pp. 2095–2122. ISSN: 1064-8275,1095-7197. DOI: 10.1137/S1064827500382324. URL: https://doi.org/10.1137/S1064827500382324.

[22] Avner Friedman. *Variational principles and free-boundary problems*. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1982, pp. ix+710. ISBN: 0-471-86849-3.

[23] Frederic Gibou et al. "A second-order-accurate symmetric discretization of the Poisson equation on irregular domains". In: *J. Comput. Phys.* 176.1 (2002), pp. 205–227. ISSN: 0021-9991,1090-2716. DOI: 10.1006/jcph.2001.6977. URL: https://doi.org/10.1006/jcph.2001.6977.

[24] Frédéric Gibou and Ronald Fedkiw. "A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem". In: *J. Comput. Phys.* 202.2 (2005), pp. 577–601. ISSN: 0021-9991,1090-2716. DOI: 10.1016/j.jcp.2004.07.018. URL: https://doi.org/10.1016/j.jcp.2004.07.018.

[25] Michael B. Giles and Rebecca Carter. "Convergence analysis of Crank-Nicolson and Rannacher time-marching". In: *J. Comput. Finance* 9.4 (2006), pp. 89–112.

[26] Jonathan Goodman and Daniel N. Ostrov. "On the early exercise boundary of the American put option". In: *SIAM J. Appl. Math.* 62.5 (2002), pp. 1823–1835. ISSN: 0036-1399,1095-712X. DOI: 10.1137/S0036139900378293. URL: https://doi.org/10.1137/S0036139900378293.

[27] E. Hairer and G. Wanner. *Solving ordinary differential equations. II*. Second. Vol. 14. Springer Series in Computational Mathematics. Stiff and differential-algebraic problems. Springer-Verlag, Berlin, 1996, pp. xvi+614. ISBN: 3-540-60452-9. DOI: 10.1007/978-3-642-05221-7. URL: https://doi.org/10.1007/978-3-642-05221-7.

[28] Steve Heston and Guofu Zhou. "On the rate of convergence of discrete-time contingent claims". In: *Math. Finance* 10.1 (2000), pp. 53–75. ISSN: 0960-1627,1467-9965. DOI: 10.1111/1467-9965.00080. URL: https://doi.org/10.1111/1467-9965.00080.

[29] Steven L. Heston. "A closed-form solution for options with stochastic volatility with applications to bond and currency options". In: *Rev. Financ. Stud.* 6.2 (1993), pp. 327–343. ISSN: 0893-9454,1465-7368. DOI: 10.1093/rfs/6.2.327. URL: https://doi.org/10.1093/rfs/6.2.327.

[30] Raul Kangro and Roy Nicolaides. "Far field boundary conditions for Black-Scholes equations". In: *SIAM J. Numer. Anal.* 38.4 (2000), pp. 1357–1368. ISSN: 0036-1429,1095-7170. DOI: 10.1137/S0036142999355921. URL: https://doi.org/10.1137/S0036142999355921.

[31] H.-O. Kreiss, V. Thomée, and O. Widlund. "Smoothing of initial data and rates of convergence for parabolic difference equations". In: *Comm. Pure Appl. Math.* 23 (1970), pp. 241–259. ISSN: 0010-3640,1097-0312. DOI: 10.1002/cpa.3160230210. URL: https://doi.org/10.1002/cpa.3160230210.

[32] Marie-Noëlle Le Roux. "Semi-discrétisation en temps pour les équations d'évolution paraboliques lorsque l'opérateur dépend du temps". In: *RAIRO Anal. Numér.* 13.2 (1979), pp. 119–137. ISSN: 0399-0516,0516-2777. DOI: 10.1051/m2an/1979130201191. URL: https://doi.org/10.1051/m2an/1979130201191.

[33] Marie-Noëlle Le Roux. "Variable step size multistep methods for parabolic problems". In: *SIAM J. Numer. Anal.* 19.4 (1982), pp. 725–741. ISSN: 0036-1429. DOI: 10.1137/0719051. URL: https://doi.org/10.1137/0719051.

[34] Randall J. LeVeque and Zhi Lin Li. "The immersed interface method for elliptic equations with discontinuous coefficients and singular sources". In: *SIAM J. Numer. Anal.* 31.4 (1994), pp. 1019–1044. ISSN: 0036-1429. DOI: 10.1137/0731054. URL: https://doi.org/10.1137/0731054.

[35] Ming Li and Tao Tang. "A compact fourth-order finite difference scheme for unsteady viscous incompressible flows". In: *J. Sci. Comput.* 16.1 (2001), pp. 29–45. ISSN: 0885-7474,1573-7691. DOI: 10.1023/A:1011146429794. URL: https://doi.org/10.1023/A:1011146429794.

[36] Zhilin Li. "A fast iterative algorithm for elliptic interface problems". In: *SIAM J. Numer. Anal.* 35.1 (1998), pp. 230–254. ISSN: 0036-1429,1095-7170. DOI: 10.1137/S0036142995291329. URL: https://doi.org/10.1137/S0036142995291329.

[37] Mark N Linnick and Hermann F Fasel. "A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains". In: *J. Comput. Phys.* 204.1 (2005), pp. 157–192.

[38] Pierre van Moerbeke. "An optimal stopping problem with linear reward". In: *Acta Math.* 132 (1974), pp. 111–151. ISSN: 0001-5962,1871-2509. DOI: 10.1007/BF02392110. URL: https://doi.org/10.1007/BF02392110.

[39] C. W. Oosterlee, J. C. Frisch, and F. J. Gaspar. "TVD, WENO and blended BDF discretizations for Asian options". In: *Comput. Vis. Sci.* 6.2-3 (2004), pp. 131–138. ISSN: 1432-9360,1433-0369. DOI: 10.1007/s00791-003-0117-9. URL: https://doi.org/10.1007/s00791-003-0117-9.

[40] Cornelis W Oosterlee, Coenraad CW Leentvaar, and Xinzheng Huang. "Accurate American option pricing by grid stretching and high order finite differences". In: *Delft University of Technology, The Netherlands, Technical Report* (2005).

[41] Andrea Pascucci. *PDE and martingale methods in option pricing*. Vol. 2. Bocconi & Springer Series. Springer, Milan; Bocconi University Press, Milan, 2011, pp. xviii+719. ISBN: 978-88-470-1780-1. DOI: 10.1007/978-88-470-1781-8. URL: https://doi.org/10.1007/978-88-470-1781-8.

[42] Charles S Peskin. "Flow patterns around heart valves: a numerical method". In: *J. Comput. Phys.* 10.2 (1972), pp. 252–271.

[43] David M Pooley, Kenneth R Vetzal, and Peter A Forsyth. "Convergence remedies for non-smooth payoffs in option pricing". In: *J. Comput. Finance* 6.4 (2003), pp. 25–40.

[44] Rolf Rannacher. "Finite element solution of diffusion problems with irregular data". In: *Numer. Math.* 43.2 (1984), pp. 309–327. ISSN: 0029-599X,0945-3245. DOI: 10.1007/BF01390130. URL: https://doi.org/10.1007/BF01390130.

[45] Christoph Reisinger and Alan Whitley. "The impact of a natural time change on the convergence of the Crank–Nicolson scheme". In: *IMA J. Numer. Anal.* 34.3 (2014), pp. 1156–1192.

[46] H. Risken. *The Fokker-Planck equation*. Second. Vol. 18. Springer Series in Synergetics. Methods of solution and applications. Springer-Verlag, Berlin, 1989, pp. xiv+472. ISBN: 3-540-50498-2. DOI: 10.1007/978-3-642-61544-3. URL: https://doi.org/10.1007/978-3-642-61544-3.

[47] José-Francisco Rodrigues. *Obstacle problems in mathematical physics*. Vol. 134. North-Holland Mathematics Studies. Notas de Matemática, 114. [Mathematical Notes]. North-Holland Publishing Co., Amsterdam, 1987, pp. xvi+352. ISBN: 0-444-70187-7.

[48] L. I. Rubinstein. *The Stefan problem*. Vol. Vol. 27. Translations of Mathematical Monographs. Translated from the Russian by A. D. Solomon. American Mathematical Society, Providence, RI, 1971, pp. viii+419.

[49] J. A. Sethian and Peter Smereka. "Level set methods for fluid interfaces". In: *Annual review of fluid mechanics, Vol. 35*. Vol. 35. Annu. Rev. Fluid Mech. Annual Reviews, Palo Alto, CA, 2003, pp. 341–372. ISBN: 0-8243-0735-6. DOI: 10.1146/annurev.fluid.35.101101.161105. URL: https://doi.org/10.1146/annurev.fluid.35.101101.161105.

[50] Akriti Sharma and Ramsharan Rangarajan. "A shape optimization approach for simulating contact of elastic membranes with rigid obstacles". In: *International Journal for Numerical Methods in Engineering* 117.4 (2019), pp. 371–404.

[51] Steven E. Shreve. *Stochastic calculus for finance. II*. Springer Finance. Continuous-time models. Springer-Verlag, New York, 2004, pp. xx+550. ISBN: 0-387-40101-6.

[52] Chi-Wang Shu. "Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws". In: *Advanced numerical approximation of nonlinear hyperbolic equations (Cetraro, 1997)*. Vol. 1697. Lecture Notes in Math. Springer, Berlin, 1998, pp. 325–432. DOI: 10.1007/BFb0096355. URL: https://doi.org/10.1007/BFb0096355.

[53] D. Y. Tangman, A. Gopaul, and M. Bhuruth. "Numerical pricing of options using high-order compact finite difference schemes". In: *J. Comput. Appl. Math.* 218.2 (2008), pp. 270–280. ISSN: 0377-0427,1879-1778. DOI: 10.1016/j.cam.2007.01.035. URL: https://doi.org/10.1016/j.cam.2007.01.035.

[54] Domingo Tavella and Curt Randall. *Pricing financial instruments: The finite difference method*. Vol. 13. John Wiley & Sons, 2000.

[55] Vidar Thomée. "Parabolic difference operators". In: *Math. Scand.* 19 (1966), pp. 77–107. ISSN: 0025-5521,1903-1807. DOI: 10.7146/math.scand.a-10797. URL: https://doi.org/10.7146/math.scand.a-10797.

[56] Lloyd N. Trefethen. *Spectral methods in MATLAB*. Vol. 10. Software, Environments, and Tools. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000, pp. xviii+165. ISBN: 0-89871-465-6. DOI: 10.1137/1.9780898719598. URL: https://doi.org/10.1137/1.9780898719598.

[57] Michael Unser. "Splines: A perfect fit for signal and image processing". In: *IEEE Signal processing magazine* 16.6 (1999), pp. 22–38.

[58] John B. Walsh. "The rate of convergence of the binomial tree scheme". In: *Finance Stoch.* 7.3 (2003), pp. 337–361. ISSN: 0949-2984,1432-1122. DOI: 10.1007/s007800200094. URL: https://doi.org/10.1007/s007800200094.

[59] Dawei Wang, Christina C. Christara, and Kirill Serkh. "Analysis of high-order time-stepping schemes for nonsmooth initial conditions in financial pricing". In: *Numerical Analysis Technical Reports, Department of Computer Science, University of Toronto* (2023), pp. 1–29. URL: https://www.cs.toronto.edu/NA/reports.html#Analysis_of_high_order_time.

[60] Dawei Wang, Kirill Serkh, and Christina Christara. "A high-order deferred correction method for the solution of free boundary problems using penalty iteration, with an application to American option pricing". In: *J. Comput. Appl. Math.* 432 (2023), Paper No. 115272, 26. ISSN: 0377-0427,1879-1778. DOI: 10.1016/j.cam.2023.115272. URL: https://doi.org/10.1016/j.cam.2023.115272.

[61] Andreas Wiegmann and Kenneth P Bube. "The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions". In: *SIAM J. Numer. Anal.* 37.3 (2000), pp. 827–862.

[62] Paul Wilmott, Sam Howison, and Jeff Dewynne. *The mathematics of financial derivatives. A student introduction*. Cambridge University Press, Cambridge, 1995, pp. xiv+317. ISBN: 0-521-49699-3; 0-521-49789-2. DOI: 10.1017/CBO9780511812545. URL: https://doi.org/10.1017/CBO9780511812545.

[63] Heath Windcliff, Peter A Forsyth, and Ken R Vetzal. "Analysis of the stability of the linear boundary condition for the Black-Scholes equation". In: *J. Comput. Finance* 8 (2004), pp. 65–92.

[64] Lixin Wu and Yue-Kuen Kwok. "A front-fixing finite difference method for the valuation of American options". In: *Journal of Financial Engineering* 6.4 (1997), pp. 83–97.

[65] You-lan Zhu, Xiaonan Wu, and I-Liang Chern. *Derivative securities and difference methods*. Springer Finance. Springer-Verlag, New York, 2004, pp. xviii+513. ISBN: 0-387-20842-9. DOI: 10.1007/978-1-4757-3938-1. URL: https://doi.org/10.1007/978-1-4757-3938-1.