Multigrid Methods for Calibrating Financial Models

A Thesis
Presented to
The Academic Faculty
by

Adam Speight

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

Department of Mathematical Sciences
Carnegie Mellon University
Pittsburgh, PA 15213
August 2005

Acknowledgements

My most sincere gratitude and fondness go to my advisor, Shlomo Ta'asan, who at every turn has given me exactly what I needed. Without his guidance, insight, and most of all patience, this work would not have been possible. My thesis committee members, Pierre Collin-Dufresne, Roy Nicolaides, and Steven Shreve, deserve special thanks for years of advice, assistance, and friendship, and probably most of all, for the things I'll never know about. I would also like to thank Dmitry Kramkov, John Lehoczky, Noel Walkington, Michael Gallmeyer, Burton Hollifield, Reha Tütüncü, William Hrusa, and Deborah Brandon. In their own way, each has been a tremendous help to me over the years.

I am greatly indebted to those who helped with logistical matters. In particular, the administrative staff in the Department of Mathematical Sciences has been invaluable: They are the best! My officemates, Barbara, Caner, and Venk, deserve much credit for granting countless favors and brightening Wean Hall a little. On a more basic level, thanks to Erin and Seth for loaning me their closet, Irina and Pasha for helping when I "needed fed," and Cliff, Kate, Louella, Phil, and Winnie for lending their good humor this year.

Finally, I would like to gratefully acknowledge the National Science Foundation for supporting part of this work under grant number DMS-9819950.

Contents

Acknowledgements **ii**

Abstract **v**

1 Introduction **1**

1.1 Examples 3

1.1.1 Derivative Pricing 3

1.1.2 Economic Methodology 5

1.2 A Calibration Strategy 8

2 Multigrid Methods for Probabilists **11**

2.1 An Example Problem 12

2.2 Representation of (Pseudo)-Differential Operators 14

2.3 Elements of Multigrid 15

2.3.1 Iterative Methods 16

2.3.2 Classical Smoothing Schemes 17

2.3.3 Grid Transfer Operators 20

2.3.4 The Principle of Defect Correction 21

2.3.5 Coarse Grid Correction 23

2.3.6 Two Grid Analysis 24

2.3.7 The Multigrid Cycle 27

2.4 Smoothing Analysis 31

2.4.1 Examples and Probabilistic Interpretation 34

2.5 Multigrid for Nonlinear Equations 37

3 Calibration: The Finite Parameter Case **39**

3.1 The Basic Problem 39

3.1.1 A First Example 40

3.1.2 Direct Solvers 41

3.2 A Multigrid Strategy 42

3.3 The Need for τ-Expansion 47

3.4 An Example from Economics 48

3.4.1 Characterizing the Value Function 50

3.4.2 Calibration 52

3.4.3 Numerical Treatment 53

3.4.4 Multigrid Treatment 55

3.4.5 Numerical Results 57

4 An Infinite Dimensional Example **60**

4.1 The Problem 61

4.1.1 A Classical Formulation 63

4.2 Linearizing the Problem 64

4.2.1 A Special Case 64

4.2.2 The General Case 66

4.2.3 Half-space analysis 67

4.3 Discretization 69

4.3.1 Grid Generation 70

4.3.2 A Finite Volume Scheme 70

4.4 Multigrid Treatment 72

4.4.1 Multigrid Components 73

4.4.2 The V-Cycle 77

4.5 A Probabilistic Interpretation 82

A Convergence History of Residuals **86**

Bibliography **87**

Abstract

To calibrate a model, one adjusts parameters so it gives correct answers to a set of questions with known answers. In this thesis, we consider the problem of calibrating a class of financial models. Specifically, we consider models common in financial economics, macroeconomics, and financial engineering based on continuous-time problems of stochastic control. We ask the question: If a model must be solved numerically, how difficult is it to calibrate? While numerical methods for solving stochastic control problems are well studied in both industry and academy, the inverse problem of parameter calibration has received comparatively little attention.

To solve the calibration problem numerically, we propose a multigrid technique that couples the calibration process with the model solver. For well-behaved problems, we find that the calibration problem can be solved for about three times the cost of solving the control problem with a fixed set of parameters. Computational evidence suggests that this holds independent of the number of parameters to be calibrated.

We illustrate this technique and its limitations with a series of examples. One example is a recent model from financial economics. This has only one parameter to calibrate but requires careful formulation to realize good numerical results. Another example we consider is a classical optimal stopping problem in two-dimensions. In this problem, the location of the stopping boundary is viewed as an infinite-dimensional parameter that must be chosen to satisfy the *smooth pasting* and *value matching* conditions. Fourier analysis of the pseudo-differential operators implicit in this problem show that the natural formulation is badly conditioned for numerical purposes. This analysis suggests a reformulation to regularize the problem without introducing distortions and a process for smoothing errors. This is used to construct a fast multigrid solver.

Chapter 1

Introduction

In the physical sciences, many instruments used to take measurements need periodic adjustments to give meaningful readings. A tare must be applied to a scale. Clocks must be synchronized. The practice of adjusting an instrument so its readings coincide with a set of benchmarks is called calibration. For example, suppose you have an unlabeled thermometer, just a tube filled with mercury that expands and contracts with temperature variations, and wish to know the temperature outside. If you do not know exactly how a thermometer works, you might proceed by postulating a simple model: If all other factors are held constant, mercury levels vary with temperature in a linear fashion. Before measuring the temperature outside, you would first need to expose the thermometer to two environments with known temperatures (water in transition to ice and steam, for example), and mark the tube appropriately. You would then have a degree of faith that the thermometer gives reasonable measurements for a range of temperatures.

You might calibrate a thermometer in one environment, say a swamp at sea level, and then wish to take measurements in a different setting, say high in the mountains. Since the simple model ignored factors such as air pressure and humidity, you may be skeptical of the thermometer's readings in the new environment. You would then have at least three options: ignore these effects and use the thermometer anyway, use a more realistic model that incorporates the effects of all potentially relevant factors, or simply recalibrate the thermometer in the new environment. The third option is often the most practical.

Calibration in the context of financial models follows the same logic: Parameters are adjusted so that a model gives correct answers to a carefully chosen set of questions with known answers. These questions are chosen so the model mimics the real-world along a few dimensions relevant to a particular issue. Once calibrated, the model can be used to

"measure" unknown quantities and re-calibrated as the environment changes. For example, an investment bank may use a model to quote prices for derivative securities, a corporation for accounting purposes to book values for complicated cash flows, an economist to measure the effects of a particular trade policy or production shock. Generic models are often used to address qualitative questions regarding existence, uniqueness, and direction: Questions that ask "which" and "whether" and hold for a whole family of models. In contrast, a calibrated model can address quantitative questions such as "how much" and "how big," where the answers depend critically on the parameters.

This thesis develops computational methods for solving and calibrating a class of models based on continuous-time stochastic control problems. These models form a basis for many modern theories of financial economics, macroeconomics, and financial engineering, and many canonical models in these fields have been solved with great success by applying analytical techniques to derive exact or approximate solutions. Too often, variations on these models, ones that better reflect empirical realities or explicitly incorporate institutional structures, are abandoned for want of closed-form solutions. It is sad to see technical limitations hobble a good idea.

The field of computational economics has emerged to remove some of these limitations. Judd [30] surveys the most common numerical techniques in economics and others from the physical sciences that may benefit economics. In [29], he suggests that "theoretical" analysis need not be limited to proving theorems and that computational methods can both complement and substitute for traditional analytical methods. You need two legs to run, but if one is broken, you can still hop with the other.

In both economics and engineering, there is a huge theoretical and applied literature covering the numerical solution of stochastic control problems. Rust [52], Judd [30], Kushner & Dupuis [35], and Fleming & Soner [25] survey methods relevant to economic models. The associated calibration problems remain largely unexplored. This work takes a step toward filling this gap.

The generic model we consider takes the form of an equation

$$L(v;\alpha) = f$$

which characterizes the solutions to a family stochastic control problems indexed by a parameter α. In particular, this typically denotes a Hamilton-Jacobi-Bellman (HJB) equation corresponding to a control problem with state variables governed by a stochastic differential equation which is driven by Brownian motion or some other Lévy process [25, 50]. The

examples below illustrate a few well-known models that assume this form.

To calibrate the model, a number of equations equal to the dimension of α pair the model parameters with observable data. This restriction is generically written as

$$F(v;\alpha) = b,$$

where b encodes the data with which the model should coincide. In this way, the calibration problem takes the form of the equation

$$C(\alpha) = b$$

where $C(\alpha) = F(v;\alpha)$ and the model defines v as an implicit function of α. By solving this equation, the model is *calibrated*.

1.1 Examples

We now consider several examples of calibration problems consistent with our generic formulation.

1.1.1 Derivative Pricing

The practice of calibration is ubiquitous in financial engineering. Models used by an investment bank to quote prices for derivative securities must match prices of related assets traded on liquid markets. Otherwise, the bank may be subject arbitrage. For this reason, many managers consider calibration to be an essential part of their business.

Black & Scholes Implied Volatility

The classic option-pricing model developed by Black & Scholes [10] and Merton [43] gave birth to a whole industry. Elaborations on this model are surveyed in [28, 32, 46]. One formulation posits a financial market with two assets: a risk-free money market account bearing returns at a constant rate $r > 0$, and a stock that pays no dividends with a price process that follows the geometric Brownian motion

$$\frac{dS_t}{S_t} = rdt + \sigma dW_t^Q,$$

where W^Q is a Brownian motion under a *risk-neutral* probability measure Q. The theory associates random cash flow with prices; the price of a European option that pays a random amount $f(S_T)$ at time T is $v(S_0, 0)$, where the function

$$v(x,t) = E\left[e^{-r(T-t)} f(S_T)|S_t = x\right]$$

satisfies the following parabolic partial differential equation for $x > 0$ and $t \in [0, T]$:

$$v_t - rv + rxv_x + \frac{\sigma^2 x^2}{2} v_{xx} = 0,$$
$$v(x,T) = f(x).$$

There are two parameters in this model: the interest rate r, which is often stable enough to estimate from market data, and the stock price volatility σ, which in principle could be estimated from historical stock prices using statistical techniques. However, for many stocks there is an active market for various options, and a statistical estimate of σ generally predicts theoretical prices inconsistent with prices observed in the options market. For this reason, the volatility is usually calibrated so the model prices one of the traded options correctly. The calibrated parameter σ is called the *implied volatility*.

Black & Scholes with Local Volatility

If a variety of derivative securities are available for use as calibration instruments, there are a corresponding number of implied volatilities, and it is unclear how to calibrate the Black & Scholes model. The calibration problem with a single parameter σ is overdetermined. Several approaches have been developed to remedy this situation. One is to assume that the stock price follows the process

$$\frac{dS_t}{S_t} = rdt + \sigma(S_t) dW_t^Q$$

where $\sigma(s)$ is a function of the stock price. The calibration problem then becomes to choose a function $\sigma(s),\ s > 0$ so that the model correctly prices all traded options.

In this case, the parameter to be calibrated is infinite-dimensional. This type of model can bring added flexibility to the modeling process. Usually, there are only a finite, if large, number of traded securities available for use as calibration data, and the problem is underdetermined. This setup can introduce instability and distortions that must be confronted by various regularization strategies. From a computational viewpoint, even a stable, regularized

problem of calibrating an infinite-dimensional parameter can be difficult due to the sheer size of the problem.

Nonparametric Models

Cont & Tankov [19] propose another infinite-dimensional calibration problem. Their option pricing model calibrates the risk-neutral density of the stock price process to the price of related call and put options with various strikes and maturities. In this model, discounted stock returns follow a stationary jump diffusion and are manipulated by controlling the density of the the associated Lévy measure, which is represented nonparametrically as a function of the stock price. The structure of this problem allows use of the Fast Fourier Transform. Extending this calibration technique to a broader class of models requires a different formulation and different computational technique that is appropriate for very large problems with less structure.

1.1.2 Economic Methodology

Many models in macroeconomics and finance are formulated as one or more dynamic programs. Ljungqvist & Sargent [38] survey dozens of these models.

The Lucas Model

Lucas [39] established a simple asset-pricing framework based on a general-equilibrium model. In a continuous-time version, the aggregate consumption process follows the geometric Brownian motion

$$\frac{dc_t}{c_t} = \mu dt + \sigma dW.$$

This income stream is consumed by a single agent representing a continuum of homogeneous agents who have utility functions over a cash flows δ defined by

$$U(\delta) = E\left[\int_0^\infty e^{-\beta s} u(\delta_s) ds | c_0 = x\right],$$

where the utility function is $u(x) = \frac{x^{1-\gamma}}{1-\gamma}$ and the parameter $\gamma > 1$ denotes the agents' coefficient of relative risk aversion. Optimality conditions imply that the state-price density

process, defined by

$$\xi_t = e^{-\beta t} u'(c_t),$$

gives time t prices for all cash flows δ;

$$\text{price}(\delta; [t, \infty)) = E\left[\int_t^\infty \frac{\xi_s}{\xi_t} \delta_s ds | c_t\right].$$

By regarding the sum of all corporate stock issues—the market portfolio—as a claim to the aggregate consumption process, the market portfolio's price is given by

$$S_t = \text{price}(c; [t, \infty)) = \underbrace{E\left[\int_t^\infty \frac{\xi_s}{\xi_t} c_s ds | c_t\right]}_{v(x)|_{x=c_t}}.$$

The drift and volatility of the market portfolio follow from an application of Itô's formula to $v(x)$. The interest rate and risk premium can be computed similarly.

Working with a discrete-time version of this model, Mehra & Prescott [41] asked the question, what value of γ is required to generate the risk premium observed for the market portfolio? Since this model naturally maps agents' preferences to asset prices, to recover the risk aversion parameter, they inverted the model given empirical estimates of the risk premium. In this spirit, many variations on the Lucas model have been proposed in an attempt to reconcile asset pricing theory with empirical findings from asset markets and decision theory [15, 17].

Method of Moments

Calibration is not a statistical exercise. While the practice is similar to point estimation, there is no probabilistic structure implicit in a calibration problem. Still, calibration and estimation sometimes share a similar structure and can be implemented using the same numerical techniques.

The Method of Moments can be viewed as a calibration problem [17]. Supposes that iid random variables $X_1, \ldots, X_N$ are drawn from a parametric distribution $p(x; \alpha_1, \ldots, \alpha_m)$ and that from this data a set of statistics $M_1^N, \ldots, M_m^N$ (usually sample moments) have been computed. The Method of Moments prescribes choosing parameters α so that the moments of the distribution match the sample moments. If there are more moment matching conditions than there are parameters, the Generalized Method of Moment technique applies [27].

To use the Method of Moments it is necessary to compute the moments associated with the distribution $p(x;\alpha)$. This might be done analytically, or with a Monte Carlo experiment. For many complicated models, it must usually be done numerically, which may incur the computational expense of solving a dynamic programming problem for thousands of different values of α.

Calibrated Simulation in Macroeconomics

Calibrating models based on dynamic programs is particularly important in macroeconomics. In [36], Kydland & Prescott outline a successful methodology for using macroeconomic models as laboratories for quantitatively assessing policy effects. A key step in this methodology involves calibrating parameters in a given model to stable quantities of interest in the data. Unfortunately, calibration is a nontrivial numerical problem when the model of interest does not admit solutions in closed form. Kydland and Prescott [36] have this to say:

> If a model environment is not computable, then it cannot be used for a computational experiment. This restriction can be a severe one, and the development of appropriate computable methods must therefore be given high priority.

In the context of business cycle models, Lucas (1980) [40] indicates:

> A "theory" is not a collection of assertions about the behavior of the actual economy but rather an explicit set of instructions for building a parallel or analogue system—a mechanical imitation economy... [T]he most important [force] I believe, in this area and in economics generally, consists of purely technical developments that enlarge our abilities to construct analogue economies. Here I would include both improvements in mathematical models and improvements in computational capacity.

Since 1980, there have been tremendous advances in mathematical models and computational facility. Unfortunately, the expertise required to fully leverage the computational advances is in short supply compared with the analytical virtuosity ubiquitous in the economics literature.

1.2 A Calibration Strategy

Suppose we face a generic calibration problem

$$L(v;\alpha) = f, \tag{1.1a}$$

$$F(v;\alpha) = b \tag{1.1b}$$

and regard the problem as solving an equation $C(\alpha) = b$ with v implicitly defined in terms of α by the first equation. There two fundamental issues involved with solving this equation.

The first issue involves the mathematical nature of the mapping $C(\alpha)$. We call this mapping *well-behaved* if it satisfies the conditions for Newton's method to converge quadratically. In particular, we require that C can be inverted to recover a unique solution $\alpha(b)$ for all b under consideration, that the solution $\alpha(b)$ varies smoothly in b, and that C^{-1} be Lipschitz continuous near b. The mapping must either satisfy a monotonicity property to ensure global convergence or a good starting point for the iteration must be known. If not well-behaved, the calibration problem will likely be difficult or impossible to solve.

A common pathology is an unstable function $\alpha(b)$. If the data b changes slightly and the model is re-calibrated, the calibrated parameters may move erratically. In this case, the model must be re-parameterized or regularized before computational techniques can be used profitably. Good results usually require a deep understanding of the model at hand, and while computational methods cannot fix an ill-behaved problem, they can be used to help diagnose pathologies and help the modeler to pose an alternative, well-behaved problem.

The second issue involves the computational difficulty of inverting a well-behaved map $C(\alpha)$. If α is of small dimension (up to ten or more) and a good starting point is chosen, several competitive methods can effectively solve $C(\alpha) = b$, some requiring only evaluations of the function C. The computational cost of these methods is typically proportional to $cM\log(M)$ where c is the cost of computing $C(\alpha)$ for a particular α and M is the dimension of α. For models that must be solved numerically, the bulk of this cost comes from solving equation (1.1a).

The most common numerical approach to the calibration problem involves writing a solver for equation (1.1a) and using this to construct the map $C(\alpha)$. This map is treated as a black-box and passed to a commercial equation solver. Usually, the bulk of the cost comes from solving equation (1.1a) repeatedly.

We propose a technique that solves the model and the calibration problem simultaneously. Our main finding is that well-behaved calibration problems of the form (1.1) can be solved for

about three or four times the computational cost of solving equation (1.1a) for a particular value of α. This property holds independent of the dimension of α and holds even for some well-posed problems with infinite-dimensional α.

This thesis consists of a series of examples illustrating a technique for solving calibration problems. This technique incorporates conventional methods for solving stochastic control problems, standard multigrid methods for solving elliptic PDEs, specialized multigrid methods for solving problems in optimal shape design, and several refinements we developed for applying these techniques to calibration problems.

The main value of this work is to provide a technological extension to calibration methodologies already prominent in financial engineering and economics. Large problems of the form (1.1a) that do not admit analytical solution or simplification are right at the limits of our computational ability, and due to the *curse of dimensionality*[1], are likely to remain so for the foreseeable future. While it may seem that solving the full calibration problem is prohibitively expensive, we show this is not true for many problems in finance of practical interest: Most well-behaved models that can be solved numerically can also be calibrated.

Our computational framework is based on *multigrid* methods [14, 57]. These numerical techniques were originally developed to solve Poisson-type equations but have since been extended to bear on many classes of problems. In finance, multigrid methods have been successfully applied to solve HJB equations [3] and the American option pricing problem [16]. Chapter 2 presents an introduction to multigrid methods geared toward an audience trained in probability. Chapter 3 outlines a multigrid strategy for solving calibration problems with finite-dimensional parameters. Our approach follows Ta'asan [55], using multigrid strategies to replace equation (1.1a) with a lower-dimensional proxy that is both easier to solve and gives a correct solution to the calibration problem. We develop some refinements to this technique and illustrate its application and performance with three detailed examples.

Chapter 4 treats an optimal stopping problem formulated as an infinite-dimensional calibration problem. Techniques originally developed for designing optimal airfoil shapes [7, 51, 56] are used to analyze the problem and design a multigrid solver. Fourier analysis reveals a conditioning problem in a natural discretization of the classical problem formulation. By analyzing the pseudo-differential operators derived from the problem, we are able to design a nondistortionary regularization scheme that renders the problem well-posed. In the process, we discover an interesting relationship between the modern formulation of optimal stopping

[1]The curse of dimensionality [9, 53] refers the growth of a hyper-volume as a function of dimensionality. For example, to represent the unit cube in $\mathbb{R}^d$ with N grid points per dimension requires N^d total grid points.

problems (based on viscosity solutions to variational inequalities) an its classical formulation (based on the smooth pasting condition). This leads to a probabilistic interpretation of the pseudo-differential operators in our analysis that gives insight into the nature of the smooth pasting condition.

Chapter 2

Multigrid Methods for Probabilists

Here we present a brief survey of the basic principles of an efficient numerical strategy for solving partial differential equations of the elliptic type and related problems. Though the basic idea dates back to the sixties, multigrid methods were not considered as an efficient computational strategy until the seventies when they were applied to solve elliptic boundary-value problems [11, 47]. They have since been extended to solve more general equations and applied in many disciplines.

This discussion will focus on equations related to stochastic control, elliptic Hamilton-Jacobi-Bellman (HJB) equations in particular. A numerical treatment of this topic consists of *i)* discretizing, or approximating an HJB equation with a finite-dimensional equation, and *ii)* solving the resulting discrete equations. A masterful account detailing the use of probabilistic techniques to design robust, discrete approximations to solutions of stochastic control problems can be found in Kushner & Dupuis [25, 35]. Using multigrid techniques to solve the discrete HJB equations they prescribe is discussed by Akian [3]. An accessible introduction to multigrid methods can be found in [14] and a more advanced account in [57].

To fix notation, the generic problem under consideration is to find a function $\bar{v}$ of a certain class $\mathcal{L}(\Omega)$ over a connected domain $\Omega \subset \mathbb{R}^d$ that uniquely satisfies the (potentially nonlinear) elliptic equation

$$L(v) = f$$

for a given function f. Any boundary conditions are implicit in this notation. In the context of HJB equations, the operator L corresponds to the generator, or controlled family of generators, of a continuous-time Markov process and the forcing function, f, to a cost or utility function.

Associated with this problem is a family of discrete problems indexed by $n = 0, 1, 2, \ldots$ with grid spacings $h_n = h_0 2^{-n}$ characteristic of a sequence of nested *grids*, denoted by

$$\Omega_{h_n} \subset \{(x_1, \ldots, x_d) = (h_n j_1, \ldots, h_n j_d), j \in \mathbf{Z}^d\}.$$

Associated with each grid, there is a grid function space $\mathcal{L}^h(\Omega_h)$ embedded in $\mathcal{L}(\Omega)$, an operator L^h, and a grid function f_h such that the discrete equation

$$L^h(v^h) = f^h$$

has a unique solution $\bar{v}^h$. We assume the sequence $\bar{v}^{h_n}$ converges to $\bar{v}$ in an appropriate sense. In particular, HJB equations from stochastic control problems often do not have unique solutions in classical or Sobolov function spaces. In this case, we understand the solution $\bar{v}$ to be defined as a monotone limit of the $\bar{v}^{h_n}$ and interpret differential operators in a weak or formal sense consistent with the theory of viscosity solutions.

To ease exposition, the analysis in this chapter deals only with linear equations until Section 2.5, though notation and algorithms are presented in a way that applies to nonlinear problems as well.

2.1 An Example Problem

To begin, we consider the problem of finding a smooth function v over $\Omega = [0, 1]$ that satisfies the equation $L(v) = f$, where

$$L(v) = -\beta v + \mu v_x + \frac{\sigma^2}{2} v_{xx} \tag{2.1a}$$

and f is a smooth function that vanishes at the boundaries. To close the problem, we impose the Dirichlet boundary conditions

$$v(0) = v(1) = 0. \tag{2.1b}$$

The Feynman-Kac theorem characterizes the solution of these equations in terms of the stochastic process

$$X_t = X_0 + \mu t + \sigma W_t,$$

where W_t is a standard Brownian motion. The solution to equation (2.1a)–(2.1b) is

$$\bar{v}(x) = E\left[\int_0^{T_{\partial\Omega}} -e^{-\beta t} f(X_t)dt | X_0 = x\right],$$

where $T_{\partial\Omega} = \inf\{t : W_t \notin \Omega\}$ is the first time W hits the boundary of Ω.

A Family of Discretizations

One prominent scheme for approximating solutions to differential equations is to replace derivatives with finite-difference quotients. In the context of equation (2.1a)-(2.1b), one possible discretization is

$$-\beta v_j + \mu\Big[\frac{v_{j+1} - v_{j-1}}{2h}\Big] + \frac{\sigma^2}{2}\Big[\frac{v_{j+1} - 2v_j + v_{j-1}}{h^2}\Big] = f(x_j),$$
$$\text{for } j = 1, \ldots, N_h - 1 \quad \text{and} \quad v_0 = v_{N_h} = 0, \tag{2.2}$$

where $\Omega_h = \{x_j^h = hj : j = 0, 1, \ldots, N_h\}$ and $N_h = h^{-1}$. The grid function v may be regarded either as a vector in $\mathbb{R}^{N_h+1}$ or as a continuous function over $[0, 1]$ defined by linear interpolation of the nodes (x_j, v_j).

The discretization (2.2) is second-order accurate but may become unstable for large h. We contrast this approximation by considering an alternative discretization that uses a one-sided approximation for v_x that is stable provided $\mu > 0$,

$$-\beta v_j + \mu\Big[\frac{v_{j+1} - v_j}{h}\Big] + \frac{\sigma^2}{2}\Big[\frac{v_{j+1} - 2v_j + v_{j-1}}{h^2}\Big] = f(x_j). \tag{2.3}$$

For grid points on the interior of the domain, the operators on the left side of these equations, denoted[1] L_2^h and L_1^h, can be represented compactly in *stencil* notation by

$$L_2^h = -\beta[0, 1, 0] + \frac{\mu}{2h}[-1, 0, 1] + \frac{\sigma^2}{2h^2}[1, -2, 1],$$
$$L_1^h = -\beta[0, 1, 0] + \frac{\mu}{h}[0, -1, 1] + \frac{\sigma^2}{2h^2}[1, -2, 1].$$

[1]The subscripts denote that L_2^h is accurate up to second order while L_1^h is a first-order scheme.

Analytical Solutions

If the forcing function takes the form $f(x) = -c\sin(\pi k x)$, we can analytically solve the discrete and continuous example problem in the case where $\mu = 0$. The solution to the continuous problem (2.1) is

$$v(x) = c\Big(\beta + \frac{\sigma^2}{2}\pi^2 k^2\Big)^{-1} \sin(\pi k x).$$

The solution to the discrete problem is[2]

$$v_j^h = c\Big(\beta + \sigma^2 \frac{1-\cos(\pi k h)}{h^2}\Big)^{-1} \sin(\pi k x_j).$$

Formulating the continuous and discrete problems this way allows the solutions to be represented in terms of a Fourier decomposition. Fourier techniques, both formal and rigorous, enable much of the proceeding analysis.

2.2 Representation of (Pseudo)-Differential Operators

Consider the class of linear operators on functions over $\mathbb{R}^d$ that are diagonalized by the Fourier modes. This means that an operator L is characterized by its action on the basis functions $e^{\iota k^\top x}$ ($\iota = \sqrt{-1}$)

$$L e^{\iota k^\top x} = \hat{L}(k) e^{\iota k^\top x}, \quad k \in \mathbb{R}^d. \tag{2.4}$$

The function $\hat{L}(k)$ is called the symbol of L. We also consider discrete operators L^h over an unbounded domain $\Omega_h = \{(x_1, \ldots, x_d) = (h_n j_1, \ldots, h_n j_d), j \in \mathbf{Z}^d\}$ of the form

$$L^h e^{\iota\theta^\top x/h} = \hat{L}^h(\theta) e^{\iota\theta^\top x/h}, \quad \theta \in \Theta = [-\pi, \pi]^d.$$

The frequency k is identified with θ/h at the discrete level, so the grid Ω_h can support functions with frequencies up to $\pm\pi/h$. For small h, we expect

$$\hat{L}(k) \approx \hat{L}^h(\theta/h).$$

[2]Both discretizations (2.2) and (2.3) are the same since $\mu = 0$.

The operators in the example problem extended to the domains $\Omega = \mathbb{R}$ and $\Omega_h = \{x_j = hj : j = 0, \pm 1, \pm 2, \ldots\}$, are represented at the continuous level by

$$Lv = -\beta v + \mu v_x + \frac{\sigma^2}{2} v_{xx}$$
$$\hat{L}(k) = -\beta + \iota k\mu - k^2 \frac{\sigma^2}{2}$$

and at the discrete level by

$$\hat{L}_2^h(\theta) = -\beta + \frac{\mu\iota \sin(\theta)}{h} - \frac{\sigma^2}{h^2}\big((1 - \cos(\theta)\big) \tag{2.5}$$
$$\hat{L}_1^h(\theta) = -\beta + \frac{\mu}{h}\Big(\iota \sin(\theta) - (1 - \cos(\theta)\Big) - \frac{\sigma^2}{h^2}\big((1 - \cos(\theta)\big). \tag{2.6}$$

Operators that admit the representation in (2.4) belong to the class of *pseudo-differential* operators. For operators that do not admit this representation a more general operator with space-dependent symbol is defined by

$$Le^{\iota k^\top x} = \hat{L}(k, x)e^{\iota k^\top x}.$$

Differential operators with constant coefficients are simply pseudo-differential operators with a symbol that is polynomial in the frequency k. Operators with non-polynomial symbols are of interest as well: the class of operators that satisfy $\hat{L}(\theta, x) < 0$ essentially corresponds to the generators of stochastic differential equations driven by Lévy Processes [18].

2.3 Elements of Multigrid

A multigrid algorithm weaves together a fast solver from several well-known techniques, many of which perform very poorly in isolation. The central idea behind multigrid algorithms is that if a grid function representing an error is smooth relative to a *fine grid* then, without any essential loss of information, it can be represented on a *coarser grid* and the smooth error can be eliminated by solving a smaller-dimensional equation. In order to apply this idea, we need a process to assure that errors are smooth. In this light, a generic multigrid scheme consists of two parts: *i)* a process to smooth errors (eliminate high-frequency components), and *ii)* a scheme to correct the smooth component of errors by solving a coarser (smaller-dimensional) system of equations. Essentially, the details of a multigrid algorithm simply consist of specifying a meaning for "smooth" and "coarse."

2.3.1 Iterative Methods

Multigrid belongs to a class of algorithms known as iterative methods. For example, the solution $\bar{x}$ to an invertible linear system

$$Ax = f \tag{2.7}$$

can often be found by iterating on a fixed point equation

$$x^{n+1} \leftarrow Mx^n + b \tag{2.8}$$

for a matrix M and vector b. One way to define the iteration is to split the matrix A into two parts, $A = B + C$, and rewrite (2.7) as $Bx = f - Cx$. An iteration of the form (2.8) can then be defined by taking $b = B^{-1}f$ and $M = -B^{-1}C$.

If $\bar{x}$ is a fixed point of this iteration and $Mx + b$ is a contraction map, then the iteration converges to $\bar{x}$ at a rate governed by the contraction factor $\|M\|$,

$$\left\|x^{n+1} - \bar{x}\right\| \leq \|M(x^n - \bar{x})\| \leq \|M\| \, \|x^n - \bar{x}\|$$

where $\|x\|$ is any norm on R^N and the induced matrix norm is defined as

$$\|M\| = \max_{\|x\|=1} \|Mx\| .$$

A convergence factor for the iteration that is independent of a particular norm is the spectral radius, denoted

$$\varrho(M) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } M\}.$$

The following result taken from [26][Section 10.1] is useful.

Proposition 1. *The spectral radius is characterized by*

$$\varrho(M) = \inf_{\|\cdot\|} \|M\|$$

where the infimum is taken over all induced matrix norms.
The iteration (2.8) *converges from any starting point* x^0 *if and only if*

$$\varrho(M) < 1.$$

2.3.2 Classical Smoothing Schemes

In recent years much attention has been given to developing fast converging iterative methods for large sparse linear equations. Most classical iterative schemes exhibit slow convergence rates but many have strong smoothing properties. These are of primary interest in the context of multigrid. Working with the generic problem of finding $v \in \mathbb{R}^{N+1}$ that solves a linear equation $Av = f$, we consider three classical iterative schemes.

1. The *Jacobi* iteration updates an iterate v^n by choosing the jth component of v^{n+1} to solve the jth row of $Av = f$ in terms of the components $v^n_{0:(j-1)}$ and $v^n_{(j+1):N}$. By splitting the matrix into its diagonal, upper, and lower triangular parts, $A = D+U+L$, the Jacobi iteration can be written as

$$v^{n+1} = D^{-1}(f - (L+U)v^n).$$

2. The *damped Jacobi* iteration updates v^n in the same direction as the Jacobi iteration, but only goes a distance $\xi \in (0,1)$ in that direction. This iteration can be written as

$$v^{n+1} = (1-\xi)v^n + \xi\big(D^{-1}(f - (L+U)v^n)\big).$$

3. The *Gauss-Seidel* scheme sweeps over each grid point $j = 0, 1, \ldots, N$ and adjusts v^{n+1}_j to solve the jth equation in terms of the components $v^{n+1}_{0:(j-1)}$ that have already been updated and the $v^n_{(j+1):N}$ that still need updating. Said differently, when sweeping through $j = 0, 1 \ldots, N$, the component v^{n+1}_j is adjusted to solve the jth equation:

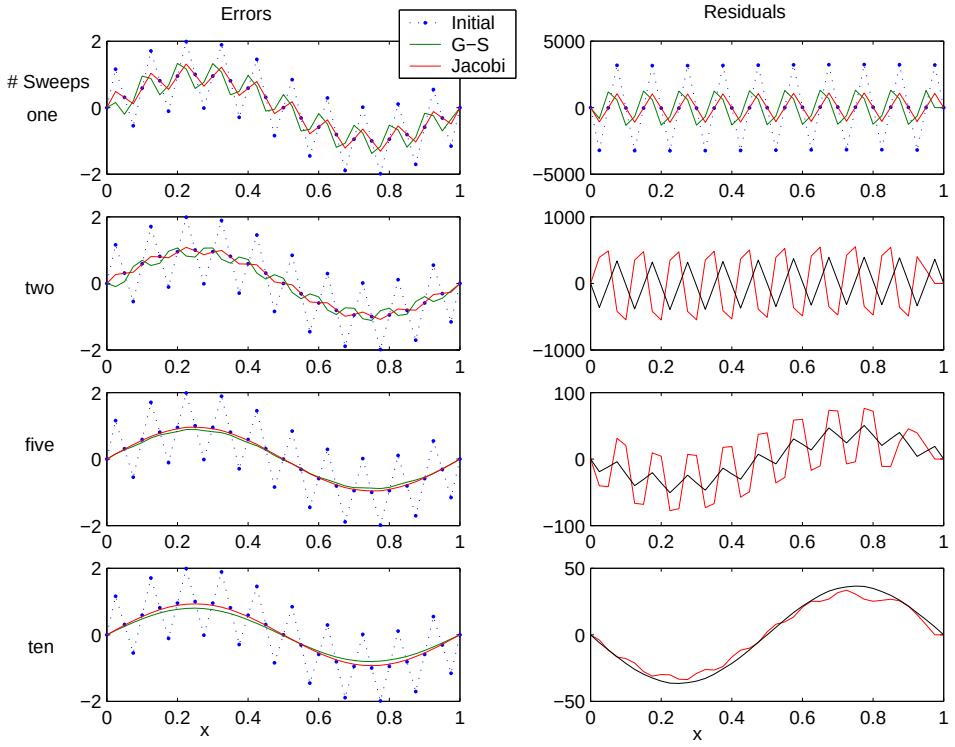
$$A(v^{n+1}_0, \ldots, v^{n+1}_{j-1}, \cdot, v^n_{j+1}, \ldots, v^n_N) = f_j.$$

   We can also write this as

$$(L+D)v^{n+1} = f - Uv^n, \quad \text{or} \quad v^{n+1} = D^{-1}(f - Uv^n - Lv^{n+1}).$$

   In contrast to the Jacobi iteration, the Gauss-Seidel scheme depends on the ordering of the equations. For some problems, changing the order in which equations are updated can drastically affect the behavior of this scheme. For this reason, it is also more difficult to analyze than the Jacobi iteration.

Figure 2.1: Shown are the effects of the damped Jacobi (with $\xi = 2/3$) and Gauss-Seidel iterations on the model problem. The initial error was set to $\sin(2\pi x) + \sin(20\pi x)$, and $N = 40$ grid points were used. After a few iterations, the errors are smooth relative to the fine grid. Since the equation is elliptic, the residuals $f - Lv$ are smoothed as well.



Example

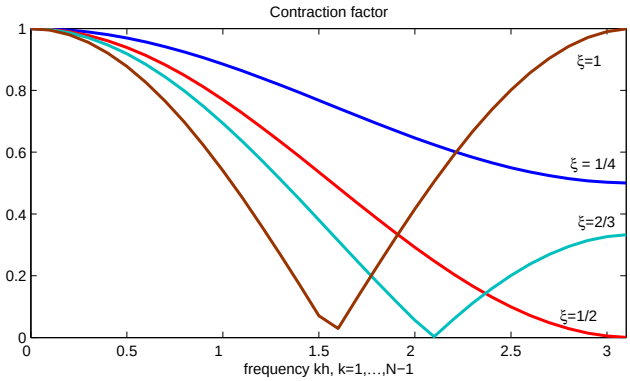
The damped Jacobi iteration can be fully characterized for a special case of the test problem (2.2). To simplify the analysis we assume the forcing function f and drift μ are both zero. This makes the solution to both the continuous and discrete problems trivial so that a candidate solution v is also the error.

We adopt the convention that $w^{\theta/h}$ is the wave defined by

$$w_j^{\theta/h} = \sin(\theta x_j/h) = \sin(\theta j), \quad \theta = kh, \quad k = 1, \ldots, N^h - 1,$$

and begin by writing the equations (2.2) compactly as a linear equation $L^h v^h = 0$. Then, for wave numbers $k = 1, \ldots, N-1$ and frequencies $\theta_k = \pi kh$, the vector $w^{\theta/h}$ is an eigenvector

Figure 2.2: The amplification factor $G(S^h)$ for the damped Jacobi iteration. Choosing $\xi = 2/3$ controls the worst-case error reduction among the high frequencies. Notice that without damping ($\xi = 1$) the Jacobi iteration fails to effectively reduce errors of highest frequency where $\theta \approx \pi$.



of L^h with an associated eigenvalue

$$\hat{L}^h(\theta) = -\beta + \sigma^2 \frac{\cos(\theta) - 1}{h^2}.$$

If the initial error is $v^0 = w^{\theta/h}$ then the application of one damped Jacobi sweep produces an error of $v^1 = \hat{S}^h(\theta)v^0$ where the *amplification factor* is

$$\hat{S}^h(\theta) = 1 - \xi \frac{\beta h^2/\sigma^2 + 1 - \cos(\theta)}{\beta h^2/\sigma^2 + 1} \approx 1 - 2\xi \sin^2(\frac{1}{2}\theta). \tag{2.9}$$

The approximation holds for small h provided $\sigma^2 > 0$ and is exact for $\beta = 0$.

As shown in Figure 2.2, the amplification factor is near one for all low-frequency modes $\theta \in [0, \pi/2]$ no matter what damping factor is used. This indicates that the Jacobi method converges slowly. However, for a well-chosen damping factor, we see that high-frequency modes with $\theta \in [\pi/2, \pi]$ are damped significantly. We define a *smoothing factor* associated with S^h to be

$$G(S^h) = \max_{\theta \in [\pi/2, \pi]} \hat{S}^h(\theta).$$

For $\beta = 0$, the damping factor that minimizes $G(S^h)$ is $\xi = 2/3$, which gives a smoothing factor of $1/3$. Figure 2.1 shows the effects of the damped Jacobi and Gauss-Seidel iterations starting with an error that has both high and low-frequency components. Both methods are efficient smoothers, and both converge slowly.

2.3.3 Grid Transfer Operators

A multigrid scheme relates functions between grids of various levels of refinement. In the context of our one-dimensional example problem, a *prolongation* operator is a map $I_{2h}^{h} : \mathcal{L}^{2h} \to \mathcal{L}^{h}$ that takes functions on one grid to the next finest grid. For second-order HJB equations, an appropriate choice is linear interpolation defined by

$$(I_{2h}^{h} v^{2h})(x) = \begin{cases} v^{2h}(x) & \text{for } x \in \Omega_{2h}, \\ \frac{1}{2}\big(v^{2h}(x+h) + v^{2h}(x-h)\big) & \text{for } x \in \Omega_h - \Omega_{2h}. \end{cases}$$

A *restriction* operator is a map $I_{h}^{2h} : \mathcal{L}^{h} \to \mathcal{L}^{2h}$ that takes functions on one grid to the next coarser grid. In this work, we consider two choices of restriction operators, the simplest of which is the *injection* map

$$(I_{h}^{2h} v^{h})(x) = v^{h}(x)$$

where x is a point on the coarse grid. Another choice is the *full-weighing* map, where for x on the interior of the coarse grid, we have

$$(I_{h}^{2h} v^{h})(x) = \frac{1}{4} v^{h}(x-h) + \frac{1}{2} v^{h}(x) + \frac{1}{4} v^{h}(x-h),$$

and points on the boundary are defined by injection. The injection and full-weighting operators can be denoted compactly in stencil form by

$$\begin{aligned} &[0, 1, 0] \quad \text{(injection)}, \\ &\frac{1}{4}[1, 2, 1] \quad \text{(full-weighting)}, \end{aligned}$$

These grid transfer operators are invariant on the two-dimensional subspaces defined by $W^{\theta/h} = \operatorname{span}(w^{\theta/h}, w^{\theta'/h})$, where $\theta \in [0, \pi/2)$ is a low-frequency wave with the complementary harmonic frequency $\theta' = \theta - \pi$. The linear interpolation operator is characterized by

$$I_{2h}^{h} w^{2\theta/2h} = \underbrace{\cos^2(\theta/2)}_{\hat{I}_{2h}^{h}(\theta/h)} w^{\theta/h} + \underbrace{\sin^2(\theta/2)}_{\hat{I}_{2h}^{h}(\theta'/h)} w^{\theta'/h}. \tag{2.10}$$

The restriction operators are characterized in the frequency domain by

$$I_h^{2h}(aw^{\theta/h} + bw^{\theta'/h}) = \big(\hat{I}_h^{2h}(\theta)a + \hat{I}_{2h}^{h}(\theta')b\big)w^{2\theta/2h},$$

where $(\hat{I}_h^{2h}(\theta),\ \hat{I}_h^{2h}(\theta')) = (1,1)$ for injection and $(\hat{I}_h^{2h}(\theta),\ \hat{I}_h^{2h}(\theta')) = (\hat{I}_{2h}^{h}(\theta), \hat{I}_{2h}^{h}(\theta'))$ for the case of full-weighting. The latter relationship reflects an intimate pairing between the full-weighting and linear interpolation operators, which, up to a constant of proportionality, are adjoint to one another.

The notation suggests that the transfer operators behave like the identity map. Indeed, they should be thought of as satisfying

$$I_h^{2h} I_{2h}^{h} \approx I_{2h}^{2h}, \quad \text{and} \quad I_{2h}^{h} I_h^{2h} = I_h^h + \text{high-frequencies.}$$

where I_h^h is the identity map on Ω_h.

2.3.4 The Principle of Defect Correction

Before developing a multigrid algorithm, we study an iterative scheme that illustrates the basic principle. Suppose we have two invertible matrices A and B and are interested in finding $\bar{x}$ that solves

$$Ax = f. \tag{2.11}$$

Now suppose that solving this equation is difficult but solving the equation

$$Bx = g$$

is relatively easy for any vector g. In particular, we think of equation (2.11) as the problem of interest and the equation $Bx = f$ as a low-accuracy approximation.

The idea of *defect correction* is to define an iterative scheme

$$\begin{aligned} Bx^{n+1} &= f - Ax^n + Bx^n, \qquad \text{or} \\ x^{n+1} &= B^{-1}(f - Ax^n + Bx^n) \\ &= B^{-1}f + (I - B^{-1}A)x^n. \end{aligned}$$

The solution $\bar{x}$ is a stable point of this iteration, and as shown earlier, the largest eigenvalue of $I - B^{-1}A$ governs the convergence rate of the iteration. In particular, this iteration

converges provided $\varrho(I - B^{-1}A) < 1$. If the contraction factor is small and B^{-1} cheap to apply, then this scheme defines a fast solver.

A rearrangement of this iteration shows that $\bar{x}$ is the unique solution of

$$Bx = f + \underbrace{(B - A)\bar{x}}_{\tau}. \tag{2.12}$$

Thus, the high-accuracy equation has the same solution as the low-accuracy equation provided the right side is perturbed by an appropriate vector τ, which we call the *defect*. If we knew τ, we could simply solve (2.12) and recover $\bar{x}$ at the cost of solving the low-accuracy equation. Unfortunately, there is no free lunch: τ is unknown and must be computed. In this light, the defect correction scheme can be thought of as a strategy for approximating both $\bar{x}$ and τ together as the iteration progresses.

If the condition $\varrho(I - B^{-1}A) < 1$ fails, the defect correction iteration may still converge if started at a well-chosen point. For example, take $M = I - B^{-1}A$ and $f = 0$, and consider the iteration

$$x^{n+1} = Mx^n.$$

Suppose a few eigenvalues of M are large and the rest are much smaller than one. Suppose further that the eigenspace $\mathcal{X}^{\text{big}}$ corresponding to the large eigenvalues is orthogonal to the complementary eigenspace $\mathcal{X}^{\text{small}}$.[3] A point chosen in this subspace will converge at a rate governed by the largest of the small eigenvalues.

If the eigenspaces $\mathcal{X}^{\text{big}}$ and $\mathcal{X}^{\text{small}}$ are not orthogonal, the iteration is not invariant in $\mathcal{X}^{\text{small}}$. In this case the following iteration can be used.

$$x^{n+1} = M\big(\text{Proj}(x^n|\mathcal{X}^{\text{small}})\big) \tag{2.13}$$

The strategy here is to split the problem into two pieces: one that exploits the defect correction principle on the relevant subspace, and another that keeps the iterates on that subspace. To implement this algorithm, the subspaces must be identified and a projection operator defined. A simple, two-grid algorithm is basically an implementation of this construction.

[3]This is assured if M is symmetric.

2.3.5 Coarse Grid Correction

With the defect correction scheme in mind, we now develop a basic multigrid algorithm for solving the equation

$$L^h v^h = f^h, \quad \text{on a grid } \Omega_h.$$

The operator L^{2h} is used to approximate L^h on the space of low-frequency functions. We proceed by assuming the equation

$$L^{2h} v^{2h} = f^{2h}$$

has a unique solution over Ω_{2h} for any coarse grid function f^{2h}.

The *Full Approximation Scheme* (FAS) of Brandt assumes that an initial guess v^h is given and the errors $v^h - \bar{v}^h$ and residuals $f^h - L^h v^h$ are smooth relative to the fine grid Ω_h. Since the errors are smooth, we seek a smooth correction $\widetilde{v}^h$, which can be represented on a coarser grid. The FAS updates v^h by

$$v^h \leftarrow v^h + \underbrace{I_{2h}^h (v^{2h} - I_h^{2h} v^h)}_{\widetilde{v}^h}.$$

where v^{2h} solves the coarse grid equation

$$L^{2h} v^{2h} = f^{2h}, \tag{2.14}$$

and the coarse grid right side is given by

$$\begin{aligned} f^{2h} &= I_h^{2h}(f^h - L^h v^h) + L^{2h}(I_h^{2h} v^h) \\ &= I_h^{2h} f^h + \underbrace{L^{2h}(I_h^{2h} v^h) - I_h^{2h}(L^h v^h)}_{\tau}. \end{aligned} \tag{2.15}$$

In compact form, the FAS scheme defines a coarse grid correction operator $v^h \leftarrow CGC(v^h)$ where

$$CGC(v^h) = v^h + I_{2h}^h \Big((L^{2h})^{-1} \big[I_h^{2h}(f^h - L^h v^h) + L^{2h}(I_h^{2h} v^h) \big] - I_h^{2h} v^h \Big). \tag{2.16}$$

The τ-adjustment to the right side vector in (2.15) ensures the coarse-grid problem will correct smooth components of the errors, leaving only high-frequency (nonsmooth) errors

that cannot be represented on the coarse grid.

The coarse-grid correction defined by the FAS satisfies the fixed-point property

$$\bar{v}^h = CGC(\bar{v}^h);$$

however it does not contract for all v^h. Indeed, it may not contract for *any* v^h in the discrete L_h^2 norm. Instead, the CGC operator may exchange low-frequency errors for high-frequency errors, which can be removed cheaply by application of a smoother. Thus, the smoother and coarse-grid correction play complementary roles.

As with defect correction, if the τ in (2.15) were computed using the solution $\bar{v}^h$ rather than the current iterate v^h, the coarse-grid correction would produce a correction

$$v^h \leftarrow v^h + I_{2h}^h I_h^{2h}(\bar{v}^h - v^h) = (I - I_{2h}^h I_h^{2h})v^h + I_{2h}^h I_h^{2h}\bar{v}^h,$$

which is correct up to the spurious high-frequency harmonics introduced by the grid transfer operators.

Remark 1. *A common mistake is to use $f^{2h} = I_h^{2h} f^h$ with no τ-correction in the coarse grid problem. In this case, the solution $\bar{v}^{2h}$ does not depend on the approximation $\bar{v}^h$ and fine-grid accuracy does not obtain for the coarse-grid problem. The correction $\widetilde{v}^h = I_{2h}^h(\bar{v}^{2h} - I_h^{2h}\bar{v}^h)$ is not zero even though the iteration starts at the actual solution. This violates the fixed-point property that any iterative scheme should have.*

2.3.6 Two Grid Analysis

The FAS depends critically upon the assumption that the errors and residuals are smooth enough to be represented accurately on the coarse grid. As we have seen, high frequency errors can often be efficiently removed by a procedure such as the Gauss-Seidel or damped Jacobi iterations. Once the error is smooth, the coarse grid correction update can introduce some high-frequency error. An effective algorithm must therefore smooth errors both before and after the coarse grid correction is applied. The two-grid operator integrates these ideas.

In compact form the two-grid operator reads

$$TG(v^h) = \left(S^s \circ CGC \circ S^s\right) v^h.$$

For the model problem (2.2) we can perform a rigorous analysis of the two-grid operator in

* **Algorithm:** $v^h \leftarrow TG(v^h, f^h, s)$

1. Apply s pre-smoothing sweeps, $v^h \leftarrow S^s v^h$.
2. Compute f^{2h} according to (2.15) and solve the coarse grid equation (2.14) for v^{2h} using an exact solver with $I_h^{2h} v^h$ as a starting point.
3. Update $v^h \leftarrow v^h + I_{2h}^h (v^{2h} - I_h^{2h} v^h)$.
4. Apply s post-smoothing sweeps, $v^h \leftarrow S^s v^h$ and return v^h.

the case when the drift $\mu = 0$ and forcing function $f(x) = 0$. Since this problem is linear and homogeneous, it admits only the trivial solution $\bar{v}^h = 0$, and the coarse grid correction (2.16) simplifies to

$$CGC(v^h) = (I - I_{2h}^h (L^{2h})^{-1} I_h^{2h} L^h) v^h. \tag{2.17}$$

The smoother to be used is s sweeps of the damped Jacobi iteration from Section 2.3.2. As we have seen, this smoother damps each component of the error according to

$$\left(S^h\right)^s w^{\theta/h} = \left(\hat{S}^h(\theta)\right)^s w^{\theta/h}$$

where the symbol $\hat{S}^h(\theta)$ is given in (2.9).
The symbol of L^h is given by

$$\hat{L}^h(\theta) = \frac{\sigma^2(1 - \cos(\theta))}{h^2},$$

so when f^{2h} takes the form $cw^{2\theta/2h}$, the coarse grid equation is solved by

$$v^{2h} = (L^{2h})^{-1} f^{(2h)} = c(\hat{L}^{2h}(2\theta))^{-1} w^{2\theta/2h}.$$

The spectral characterization of these components shows that the coarse-grid correction operator is invariant on the subspace $W^{hk} = \text{span}(w^{\theta/h}, w^{\theta'/h})$, where $\theta' = \theta - \pi$. We consider each subspace $W^{\theta/h}$ individually, writing $v^h = aw^{\theta/h} + bw^{\theta'/h}$. On each subspace, the coarse grid correction has the effect

$$\widehat{CGC}(\theta) \begin{pmatrix} a \\ b \end{pmatrix} = \left[I - \begin{pmatrix} \hat{I}_{2h}^h(\theta) \\ \hat{I}_{2h}^h(\theta') \end{pmatrix} \frac{(\hat{I}_h^{2h}(\theta),\ \hat{I}_h^{2h}(\theta'))}{\hat{L}^{2h}(2\theta)} \begin{bmatrix} \hat{L}^h(\theta) & 0 \\ 0 & \hat{L}^h(\theta') \end{bmatrix} \right] \begin{pmatrix} a \\ b \end{pmatrix}$$

and the action of the two-grid scheme with s pre and post-relaxation sweeps is given by

$$\widehat{TG}(\theta)\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \hat{S}^h(\theta) & 0 \\ 0 & \hat{S}^h(\theta') \end{pmatrix}^s \widehat{CGC}(\theta) \begin{pmatrix} \hat{S}^h(\theta) & 0 \\ 0 & \hat{S}^h(\theta') \end{pmatrix}^s \begin{pmatrix} a \\ b \end{pmatrix}.$$

The spectral radius of TG governs the convergence rate of the two-grid scheme. Since TG is invariant under each two-dimensional subspace, $W^{\theta/h}$, we consider each pair of eigenvalues corresponding to $\widehat{TG}(\theta)$ for $\theta \in [0, \pi/2)$. Figure 2.3 shows the eigenvalues for various combinations of smoothing sweeps and wave numbers. Using either injection or full-weighting for the restriction operator, we see that using $s = 2$ pre and post-relaxation sweeps results in a two-grid scheme with a contraction factor better than 1/10. This is considered textbook multigrid performance.

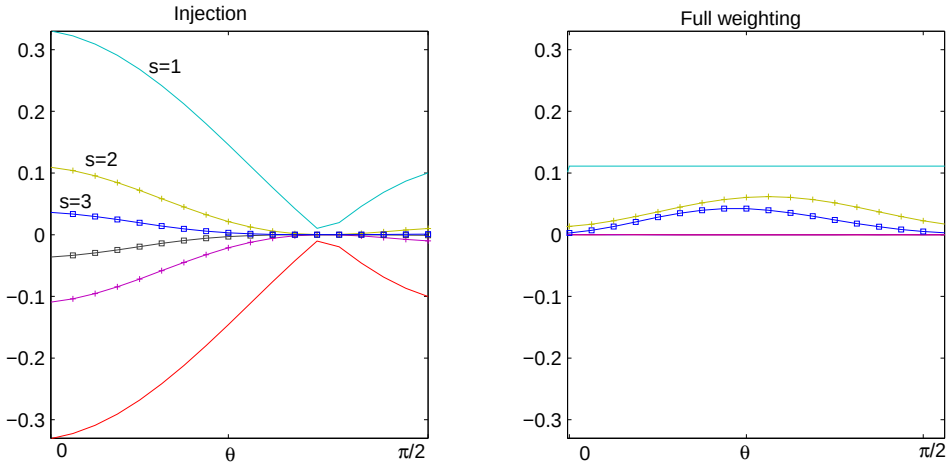
Figure 2.3 also shows that the choice of grid transfer operators significantly affects the convergence rate of the two grid iteration. The eigenvalues of the coarse-grid correction (with no smoothing sweeps) are $(1, -1)$ when using the injection operator and $(1, 0)$ when using full-weighting. With full-weighting, zero is an eigenvalue for every frequency, and the coarse grid correction effectively removes errors of low-frequency. With injection, the coarse grid correction does not eliminate low-frequency errors. Rather it transorms them into high-frequency errors, which must be damped by a smoother. Interpreted in the context of defect correction, the smoother plays the role of projecting the error on the subspace of low-frequency functions and the coarse-grid correction removes the defect on this subspace but may cause the iterate to leave that subspace. Individually, the smoother and coarse grid correction perform poorly, but together they are effective as shown in Figure 2.4.

To properly gauge the convergence rate of the iteration, the subspaces on which TG is invariant must be identified and the behavior analyzed there. For larger, more complex problems, there may be no low-dimensional, invariant subspaces. In these cases, various approximations must be made to predict the performance of the two-grid operator. One crude approximation that is often effective ignores the effects of the grid-transfer operators and simply ensures that the coarse-grid operator L^{2h} is sufficiently close to L^h across all frequencies. Using equation (2.17) but ignoring the effects of grid transfer operators gives the convergence criteria

$$\max_{\theta} \left| 1 - \frac{\hat{L}^h(\theta)}{\hat{L}^{2h}(2\theta)} \right| < 1, \tag{2.18}$$

where the maximum is taken over all low-frequency θ for which the smoother fails to damp

Figure 2.3: Eigenvalue pairs for the $\widehat{TG}$ matrix for frequencies $\theta \in [0, \pi/2]$ and $s = 1, 2, 3$ smoothing sweeps.



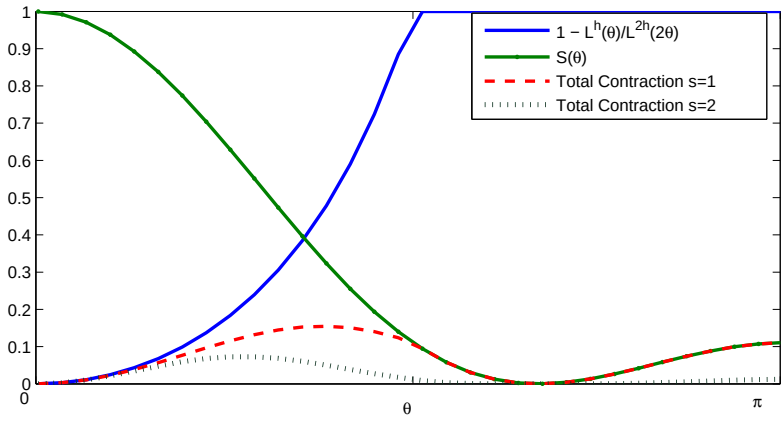
the error. If this property fails, the coarse grid problem cannot be expected to approximate the fine grid problem no matter what grid transfer operators are used. If this property holds robustly, a variety of grid transfer operators can be used. If it barely holds, a judicious choice of the grid-transfer operators may be necessary or a more expensive smoother may be needed. As shown in Figure 2.4, coarse-grid correction loses effectiveness as θ approaches $\pi/2$. Figure 2.2 shows that the smoother compensates by damping frequencies in this range. Figure 2.3 shows that a proper choice of grid-transfer operators controls the introduction of spurious high-frequency error. This means fewer smoothing sweeps are needed to make the two-grid scheme effective. Condition 2.18 ignores this fact.

We pause now to comment about what had been gained by passing to a coarser grid. For a one-dimensional problem, there are about half as many points on the coarse grid as on the fine grid. Generally, on a d-dimensional grid the relative computational burden between grids is about $|\Omega_h| \,/\, |\Omega_{2h}| \approx 2^{-d}$. Since the smoothing process and residual computations require only $O(|\Omega_h|)$ operations, passing the remaining work to the coarse grid problem can dramatically reduce the total computational cost provided the spectral radius of the two-grid operator is small.

2.3.7 The Multigrid Cycle

The two-grid scheme is a powerful idea. To make it useful, several issues need to be addressed. First, an efficient solver for the coarse grid problem must be available. Second, even if the

Figure 2.4: The symbol $1 - \frac{\hat{L}^h(\theta)}{\hat{L}^{2h}(2\theta)}$ of the approximate CGC operator compared with the symbol of the smoother S. We see that for middle-range frequencies, $\theta \approx \pi/2$, both the smoother and coarse grid correction components begin to lose effectiveness. Taken together, they reinforce one another in this critical range to give an amplification factor that is acceptable across all frequencies.



two-grid scheme has a favorable contraction factor, a good starting point must be chosen to ensure fast convergence. This is particularly important for nonlinear problems. Finally, we are often interested in solving a continuous equation $Lv = f$ and do not know *a priori* how fine the approximation needs to be to ensure a given level of accuracy. We discuss these issues in turn.

When solving an equation $L^h v^h = f^h$, the FAS prescribes solving an equation of the form $L^{2h} v^{2h} = f^{2h}$. One way to solve this equation is to work with an even coarser grid, Ω_{4h}, and use the FAS to define an equation of the form $L^{4h} v^{4h} = f^{4h}$. Several iterations of the two-grid scheme can be applied with Ω_{2h} playing the role of the fine grid. This coarsening strategy can continue until a coarsest grid Ω_{h_0}—with as few as three grid points per dimension—is reached where the equation can be solved very cheaply. Note that some functions in the low-frequency range on Ω_h are in the high-frequency range relative to the grid Ω_{2h}. Thus, applying a smoother on the coarse grid may damp errors that could not be cheaply removed on the fine grid. The *multigrid cycle* is a recursive extension of the two-grid operator.

The most popular specifications for the multigrid cycle are the V-cycle ($\eta = 1$) and the W-cycle ($\eta = 2$). These are shown in Figure 2.5. The following standard result for linear equations gives conditions under which the W-cycle exhibits contraction factors rates that mimic those of the two-grid operator.

Theorem 1 ([57] Theorem 3.2.1)**.** *Assume that for any $n \geq 0$, L^{h_n} are linear operators and*

* **Algorithm:** $v^h \leftarrow MG^h_{h_0}(v_h, f^h, \eta, s)$

1. If $h = h_0$, solve $L^{h_0} v^{h_0} = f^{h_0}$ on the coarsest grid and return v^{h_0}, else
2. Apply s pre-smoothing sweeps to v^h.
3. Define the coarse-grid right side f^{2h} according to (2.15), initialize $v^{2h} = I^{2h}_h v^h$, and repeat the following η times:
$$v^{2h} = MG^{2h}_{h_0}(v^{2h}, f^{2h}, \eta, s).$$
4. Update $v^h \leftarrow v^h + I^h_{2h}(v^{2h} - I^{2h}_h v^h)$.
5. Apply s post-smoothing sweeps to v^h and return v^h.

the following two-grid estimates hold[4]

$$\left\|MG^{h_n}_{2h_n}\right\| \le \xi, \quad \left\|I^h_{2h}(S)^s\right\| \left\|(S)^s I^{2h}_h\right\| \le C.$$

Then we have $\left\|MG^{h_n}_{h_0}\right\| \le \eta_n$ *where*

$$\eta_1 = \xi, \quad \eta_{n+1} = \xi + C\eta_n^\eta.$$

In particular, if $\eta = 2$ *and* $4C\eta \le 1$ *then for any* n *we have*

$$\left\|M^{h_n}_{h_0}\right\| \le \eta := \frac{1 - \sqrt{1 - 4C\xi}}{2C}.$$

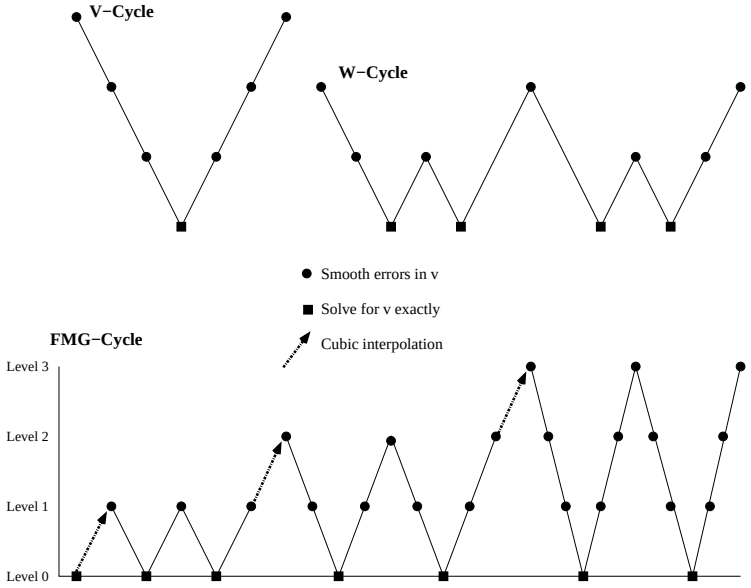
Often, the contraction properties of the two-grid operator and the smoother are sufficient to predict multigrid performance. Analogous results for V-Cycles are more delicate and problem dependent, but h-independent contraction rates can be assessed numerically and are often observed in practice.

[4]We overload notation here by using the norm to indicate contraction factors of the multigrid cycle. More precisely, the contraction factor is

$$\left\|MG^h_{h_0}\right\| = \max_{v^h} \frac{\left\|MG^h_{h_0}(v^h) - \bar{v}^h\right\|}{\|v^h - \bar{v}^h\|}.$$

This also assumes that the norms on various levels are compatible, such as the discrete L^2 norm.

Figure 2.5: Shown are a V-cycle with four levels, a W-cycle with three levels, and an FMG cycle with four levels that performs two V-cycles per outer iteration.



The Full Multigrid Cycle

If the multigrid cycle defines an appropriate iteration for solving $L^h v^h = f^h$, a suitable initial guess is still needed to start the process. One candidate starting point is an appropriately interpolated[5] function $\Pi_{2h}^h v^{2h}$ where v^{2h} solves the coarser equation $L^{2h} v^{2h} = I_h^{2h} f^h$. To solve this equation we need an initial guess which can be obtained from a still-coarser equation. This leads to another recursively defined construct called the Full Multigrid algorithm (FMG).

The FMG-Cycle produces a sequence of approximate solutions corresponding to increasingly finer grids. When using the FMG cycle, two questions naturally arise: How many multigrid cycles should be done on each level? And, how fine should the finest grid be before we accept the solution?

If FMG is being used to solve for $\bar{v}^h$, it is not necessary to solve for $\bar{v}^{2h}$ exactly. Only a good initial point is needed to start a new multigrid cycle on the next finer grid. It is not profitable to cycle once the error is smaller than the relative discretization error; the multigrid cycle

[5]For technical reasons, linear interpolation is not appropriate. Typical Poisson-type PDEs arising in finance require cubic interpolation. Generally, one should use an interpolation scheme of one order larger than the order of the underlying differential operator. Using linear interpolation would introduce spurious high-frequency errors in the fine grid starting point, which would then have to be smoothed. Since smoothing errors on the finest grid is a relatively expensive operation, it should be avoided if possible.

* **Algorithm:** $v^h \leftarrow FMG^h_{h_0}(f^h, \eta, \gamma, s)$

1. If $h = h_0$ set $v^h = 0$ and go to Step 4, else
2. Set $f^{2h} = I^{2h}_h f^h$ and
$$v^{2h} \leftarrow FMG^{2h}_{h_0}(f^{2h}, \eta, \gamma, s).$$
3. Initialize $v^h \leftarrow \Pi^h_{2h} v^{2h}$.
4. Repeat γ times:
$$v^h \leftarrow MG^h_{h_0}(v^h, f^h, \eta, s)$$
and return v^h.

on Ω_{2h} controls only the error $\left\|v^{2h} - \bar{v}^{2h}\right\|$ while we are interested in finding a starting point close to $\bar{v}^h$, which is not necessarily $I^h_{2h}\bar{v}^{2h}$. Since the total error is

$$\left\|I^h_{2h} v^{2h} - \bar{v}^h\right\| \leq \left\|v^{2h} - \bar{v}^{2h}\right\| + \left\|I^h_{2h}\bar{v}^{2h} - \bar{v}^h\right\|$$

there is no point in iterating after

$$\left\|v^{2h} - \bar{v}^{2h}\right\| << \left\|I^h_{2h}\bar{v}^{2h} - \bar{v}^h\right\|.$$

Often, only two or three V-Cycles are necessary to satisfy this condition. Once the finest level has been reached, more cycles may be necessary to reduce the error to a prescribed tolerance.

The approximations $v^{h_0}, \ldots, v^{h_L}$ generated by the FMG-Cycle can be used to determine when h is small enough to control discretization error. If these iterates converge, they will have the Cauchy property, and the convergence rate of the sequence $\left\|I^h_{2h}(v^{h_{L-1}}) - v^{h_L}\right\|$ can be used to determine when the error $\left\|v^{h_L} - \bar{v}^0\right\|$ is within a prescribed tolerance.

2.4 Smoothing Analysis

The two-grid analysis of the example problem suggested the quality of the smoother principally determines the performance of a multigrid algorithm. In particular, given a linear system $L^h v^h = f^h$, the smoothing factor is of interest. For a smoother S, this can be defined

as

$$G(S^h) = \max_{\theta \in \Theta^{\text{High}}} \left| \hat{S}^h(\theta) \right|,$$

where the frequency space is $\Theta = [-\pi, \pi]^d$, the low frequencies are $\Theta^{\text{Low}} = [-\pi/2, \pi/2]^d$, and the high frequencies are $\Theta^{\text{High}} = \Theta \setminus \Theta^{\text{Low}}$. If a total of s pre- and post-smoothing sweeps are done at each level, a properly designed W-Cycle or V-Cycle should reduce the error by about $G(S^h)^s$ independent of the number of levels in the cycle.

For some problems, a scheme such as Gauss-Seidel can be quickly implemented and the smoothing factor assessed numerically. For more involved problems, designing an efficient smoother requires detailed analysis. Ill-posed problems may need reformulation since there may be no efficient smoother. We proceed formally to outline a general strategy for designing smoothers and to derive conditions under which an efficient smoother exists.

To solve $L^h v^h = f^h$ starting with a guess v^h, we seek a correction $\widetilde{v}^h$ that satisfies

$$L^h \widetilde{v}^h = f^h - L^h v^h.$$

A process that uses only the residuals $f^h - L^h v^h$ to compute $\widetilde{v}^h$ is called *efficient* since the residuals are readily computed. Given a matrix D we can define the correction

$$\widetilde{v}^h = D(f^h - L^h v^h) \tag{2.19}$$

If D is close to $(L^h)^{-1}$ for high frequencies, then the method should produce a smoothing correction step. If D is simple to compute, the method is efficient.

The simplest choice of D is a constant times the identity matrix. For many problems, this can define an efficient and effective smoother. To find an appropriate constant, we first note that elliptic equations $Lv = 0$ are often regarded as long-time equilibrium limits of parabolic equations of the form $Lv + (d/dt)v = 0$. As time passes ($t \to -\infty$) the configuration $v(x,t)$ becomes increasingly regular with $(d/dt)v$ approaching zero. An explicit finite difference scheme is a common discretization of this problem.

$$\frac{v^{t-\Delta t} - v^t}{\Delta t} = L^h v^t$$

As time marches, this equation defines an iterative process

$$v^{t-\Delta t} \leftarrow (\underbrace{I + \Delta t L^h}_{S^h_{\Delta t}})v^t,$$

which, for classical problems, converges to a smooth, equilibrium configuration $\bar{v}(x)$ that solves $L\bar{v} = 0$.

A stability analysis prescribes that this iteration respect the property that the total energy in the configuration v does not increase as the process evolves. One way to enforce this is to require that all frequencies are damped at each step, meaning

$$\left|\hat{S}^h_{\Delta t}(\theta)\right| < 1, \text{ for all } \theta \in \Theta.$$

This condition is satisfied when (assuming for now that $\hat{L}^h(\theta)$ does not depend on x)

$$0 < \Delta t \leq \underbrace{\min_{\theta\in\Theta} -2\hat{L}^h(\theta)}_{\Delta t^{\max}}.$$

This condition enforces convergence, but does not guarantee good smoothing, which requires a judicious choice of Δt. Ideally, we choose the time step size to achieve the most favorable smoothing factor. That is, we seek the Δt^* that solves

$$G^* = \min_{0<\Delta t\leq\Delta t^{\max}} \underbrace{\max_{\theta\in\Theta^{\text{High}}} \left|1 + \Delta t\hat{L}^h(\theta)\right|}_{G(S^h_{\Delta t})}.$$

Provided that for $\theta \in \Theta^{\text{High}}$ we have

$$\hat{L}^h(\theta) < 0, \quad \text{and} \tag{2.20}$$

$$\min_{\theta\in\Theta^{\text{High}}} \left|\hat{L}^h(\theta)\right| > 0 \tag{2.21}$$

we choose

$$\Delta t^* = \min\Big(\frac{1}{\min_{\theta\in\Theta^{\text{High}}} \left|\hat{L}^h(\theta)\right|}, \Delta t^{\max}\Big)$$

and expect good smoothing ($G^* << 1$). We also expect a convergent iteration, but since an explicit finite difference scheme with small time steps is being used, convergence may be

slow.

Condition (2.21) says that L^h amplifies high-frequencies. This property, known in the literature as *h-ellipticity*, is essentially necessary and sufficient for the existence of an efficient smoother [57].

In general, the operator L will have a space-dependent symbol $\hat{L}(\theta, x)$. In this case, the above analysis holds if Δt can be chosen small enough so that the relevant inequalities hold uniformly in x. A better smoother can often be constructed by choosing a space-dependent time step. In this case, the matrix D in (2.19) is not a constant times the identity but a diagonal matrix. Generally, putting more work into constructing D results in a better smoother.

2.4.1 Examples and Probabilistic Interpretation

We now take a probabilistic view of the h-ellipticity property and consider the Fokker-Plank equation governing the transition density of Brownian motion with drift.

$$-(d/dt)v = \mu v_x + \frac{\sigma^2}{2} v_{xx}$$

By convention, time runs backward from a given configuration. Two discretizations of this equation are given by

$$\frac{v^{t-\Delta t} - v^t}{\Delta t} = L_k^h v^t, \quad k = 1, 2$$

where L_1^h and L_2^h are given in Section 2.1. From this equation, we write the iteration

$$v^{t-\Delta t} = \big(I + \Delta t L_k^h\big) v^t,$$

which can be written compactly as

$$v_j^{t-\Delta t} = (v_{j-1}^t, v_j^t, v_{j+1}^t) \begin{bmatrix} \frac{\sigma^2 \Delta t}{2h^2} - \frac{\mu \Delta t}{2h} \\ 1 - \frac{\Delta t \sigma^2}{h^2} \\ \frac{\sigma^2 \Delta t}{2h^2} + \frac{\mu \Delta t}{2h} \end{bmatrix} \text{ for the operator } L_2^h, \text{ and}$$

$$v_j^{t-\Delta t} = (v_{j-1}^t, v_j^t, v_{j+1}^t) \begin{bmatrix} \frac{\sigma^2 \Delta t}{2h^2} \\ 1 - \frac{\mu \Delta t}{h} - \frac{\Delta t \sigma^2}{h^2} \\ \frac{\sigma^2 \Delta t}{2h^2} + \frac{\mu \Delta t}{h} \end{bmatrix} \text{ for the operator } L_1^h.$$

According to the dynamic programming view, the weighting vectors on the right side of these equations represent transition probabilities from state j to neighboring states. For these weights to define a valid probability transition matrix, the weights must lie between zero and one and sum to one. This translates into the conditions

$$\begin{aligned} 0 \leq \Delta t \leq \frac{h^2}{\sigma^2}, \quad \sigma^2 \geq h\,|\mu|\,, \qquad \text{for } L_2^h \\ 0 \leq \sigma^2 + 2\mu h \leq \frac{2h^2}{\Delta t}, \quad \sigma^2 \geq -h\mu, \qquad \text{for } L_1^h. \end{aligned}$$

The first set of inequalities identifies the largest Δt that will enforce convergence of the iteration. The second set is related to the h-ellipticity of the operators L_2^h and L_1^h. If they hold, it will be possible to pick a $\Delta t > 0$ that not only enforces convergence, but also defines a good smoother.

Using the symbols of L_2^h and L_1^h given in (2.5)-(2.6) we make the following observations:

- If $\sigma^2 > 0$, the condition $\sigma^2 > h\,|\mu|$ is equivalent to L_2^h being h-elliptic. If $\mu \neq 0$, h-ellipticity is lost on a sufficiently coarse grid (one with large h).

- If $\sigma^2 > 0$ and $\mu > 0$, then L_1^h is h-elliptic no matter how coarse the grid. This stability is bought by low-order accuracy. If $\mu < 0$, then the condition $\sigma^2 \geq -h\mu$ implies h-ellipticity.

- If $\sigma^2 = 0$ and $\mu > 0$, then L_1^h is h-elliptic and a $\Delta t > 0$ can be chosen to build a good smoother.

- If $\sigma^2 = 0$ and $\mu < 0$, then L_1^h is still h-elliptic, but the symbol is positive, which violates condition (2.20). This means that there is no $\Delta t > 0$ that will result in a convergent scheme. The flow of time is reversed in this case, and a negative Δt is required for smoothing. This implies that the condition $\sigma^2 > 0$ imparts an orientation to time flow.

- The two-sided and one-sided approximations to v_x are related by

$$\frac{1}{h}[0, -1, 1] = \frac{1}{2h}[-1, 0, 1] + \frac{h}{2h^2}[1, -2, 1].$$

The second term corresponds to the stencil for the usual approximation for v_{xx}. For $\mu > 0$, we therefore interpret the operator L_1^h as being an adjusted version of L_2^h with the coefficient σ^2 replaced with $\sigma^2 + \mu h$. This can also be seen by inspecting the symbols in (2.5)-(2.6). This extra term corresponds to the addition of artificial viscosity or noise to the system.

A 2D Example

Consider the two-dimensional PDE over $\mathbb{R}^2$, $Lv = 0$, where the operator

$$Lv = v_{xx} + v_{yy}$$

corresponds to the generator of a two-dimensional Brownian motion. We consider two discretizations of this equation.

$$\begin{aligned}
(L_+^h v)_{i,j} &= \frac{1}{h^2}\Big(v_{i,j-1} + v_{i,j+1} + v_{i-1,j} + v_{i+1,j} - 4v_{i,j}\Big) = 0, \text{ and} \\
(L_\times^h v)_{i,j} &= \frac{1}{2h^2}\Big(v_{i+1,j-1} + v_{i-1,j+1} + v_{i-1,j+1} + v_{i+1,j+1} - 4v_{i,j}\Big) = 0, \text{ or} \\
L_+^h &= \frac{1}{h^2}\begin{pmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{pmatrix}, \quad L_\times^h = \frac{1}{2h^2}\begin{pmatrix} 1 & & 1 \\ & -4 & \\ 1 & & 1 \end{pmatrix}.
\end{aligned}$$

These operators are called the discrete five-point Laplacian and the skew Laplacian. We can rewrite these equations in dynamic programming form

$$v = (I + \Delta t L^h)v \quad \text{where} \quad \Delta t = \frac{h^2}{4} \quad \text{and}$$

$$\begin{aligned}
v_{i,j} &= \frac{1}{4}\Big(v_{i,j-1} + v_{i,j+1} + v_{i-1,j} + v_{i+1,j}\Big), \text{ and} \\
v_{i,j} &= \frac{1}{4}\Big(v_{i+1,j-1} + v_{i-1,j+1} + v_{i-1,j+1} + v_{i+1,j+1}\Big), \text{ or} \\
P_+^h &= \frac{1}{4}\begin{pmatrix} & 1 & \\ 1 & 0 & 1 \\ & 1 & \end{pmatrix}, \quad P_\times^h = \frac{1}{4}\begin{pmatrix} 1 & & 1 \\ & 0 & \\ 1 & & 1 \end{pmatrix}
\end{aligned}$$

where P_+^h and $P_\times^h$ represent transition probabilities in stencil notation. The discrete symbols of these generators are

$$\hat{L}_+^h(\theta_1,\theta_2) = -\frac{4}{h^2}\big(\sin^2(\theta_1/2)+\sin^2(\theta_2/2)\big)$$
$$\hat{L}_\times^h(\theta_1,\theta_2) = -\frac{2}{h^2}\big(\sin^2(\theta_1/2)\cos^2(\theta_2/2)+\sin^2(\theta_2/2)\cos^2(\theta_1/2)\big).$$

Clearly, L_+^h is h-elliptic. However, $L_\times^h$ is not h-elliptic because the symbol vanishes at $(\theta_1,\theta_2)=(\pi,\pi)$. This point is subtle since $P_\times^h$ defines valid transition probabilities. The problem is that the domain is not quite connected at the discrete level: the random walk has zero probability of starting at state (x_i,y_j) and ever visiting the neighboring states $(x_i,y_{j\pm1})$ and $(x_{i\pm1},y_j)$.

2.5 Multigrid for Nonlinear Equations

If the operator L^h is linear, the the FAS update

$$L^{2h}v^{2h} = I_h^{2h}(f^h - L^h v^h) + L^{2h}I_h^{2h}v^h,$$
$$v^h \leftarrow v^h + I_{2h}^h(v^{2h} - I_h^{2h}v^h)$$

simplifies, allowing us to solve for the correction rather than for the whole function

$$L^{2h}\widetilde{v}^{2h} = I_h^{2h}(f^h - L^h v^h),$$
$$v^h \leftarrow v^h + I_{2h}^h\widetilde{v}^{2h}.$$

Historically, multigrid was developed in this form to solve linear equations. The FAS was subsequently developed as an extension to handle nonlinear problems [11]. Since this thesis treats mostly nonlinear equations, the algorithms in this chapter were stated in terms of the full FAS.

There are two main approaches for solving nonlinear problems with multigrid. One is to use a scheme such as the FAS that incorporates the nonlinearity directly into coarse grid problems.

The other is to use Newton's method, solving the equation $L^h(v^h) = f^h$ by iterating on v^h:

$$\nabla L^h(v^h)\widetilde{v}^h = f^h - L^h(v^h),$$
$$v^h \leftarrow v^h + \widetilde{v}^h.$$

Multigrid techniques could then be used to solve the linearized equation. If a V-Cycle scheme is used to solve the linear equation, one must choose how many cycles to do per Newton iteration. To solve the equation exactly may require many cycles. If only one or two are used, the quadratic convergence typical of Newton's method does not obtain since the linear equation is not solved exactly.

In many cases, in particular with HJB equations [3, 35], the FAS and the Newton/multigrid method with one V-Cycle per Newton step exhibit nearly identical performance. Akian [3] concluded that using the Newton/multigrid scheme often permitted a more efficient implementation since evaluation of the optimal controls are required once per multigrid cycle and only on the fine grid rather than during smoothing and residual computations on every level. For calibration problems is often more appropriate to use the FAS since the coarse-grid problem defines a valid economic problem rather than a linearized equation that is difficult to interpret. This can be helpful when designing a solver for the problem on the coarsest grid.

Chapter 3

Calibration: The Finite Parameter Case

3.1 The Basic Problem

We now propose a general multigrid scheme for calibrating models based on stochastic control problems. The generic problem is to find a function v over a domain $\Omega \subset \mathbb{R}^d$ that solves an HJB equation, denoted by

$$L(v;\alpha) = f, \tag{3.1a}$$

for a given set of parameters $\alpha \in \mathbb{R}^M$, which are identified by imposing an M-dimensional restriction, denoted by

$$F(v,\alpha) = b. \tag{3.1b}$$

The first equation determines the *value function* associated with the HJB equation, and we call to the second equation the *calibration condition*.

As in the previous chapter, we assume the calibration problem is discretized on a family of grids Ω_{h_n} and denote the discrete problems by

$$L^h(v^h;\alpha^h) = f^h, \tag{3.2a}$$
$$F^h(v^h;\alpha^h) = b^h, \tag{3.2b}$$

which we assume have unique solutions for any right side vectors f^h and b^h.

Problems with similar structure have been treated successfully with multigrid methods. One common example is the Poisson problem with von Neumann boundary conditions:

$$\begin{aligned} v_{xx} + v_{yy} &= f, \text{ on } \Omega \\ \nabla v \cdot \mathbf{n} &= b, \text{ on } \partial\Omega \end{aligned}$$

where $\mathbf{n}$ denotes the unit outward normal vector. For this problem to have a solution, it must satisfy the *compatibility condition* $\int_\Omega f = \int_{\partial\Omega} b$. If a solution does exist, it is not unique and another equation must be imposed. One possible restriction is to specify the value of v at a particular point; another is to impose a global constraint such as $\int_\Omega v = 0$. Multigrid treatment of this equation is outlined in [57].

This structure also arises in the context of distributed control theory where parameters have a global effect on the behavior of a system. Ta'asan [55, 56] proposes a multigrid technique for a class of problems where the system is governed by an elliptic PDE. We use this work as a point of departure to study calibration problems in finance. Section 3.2 outlines a basic multigrid strategy. While the basic technique often works well, problems in finance often require some refinements. We introduce an appropriate modification in Section 3.3 and consider its performance in three detailed examples.

3.1.1 A First Example

We begin by working with a simple model problem defined over the domain $\Omega = [0, 1]$.

$$\alpha \sin(\pi x) + \frac{\sigma^2}{2} v_{xx} = 0, \tag{3.3a}$$

$$v(0) = v(1) = 0,$$

$$v(\frac{1}{2}) = b \tag{3.3b}$$

In probabilistic terms, v defines the expected cumulative reward as

$$v(x) = E\left[\int_0^{T_{\partial\Omega}} \alpha \sin(\pi X_t) dt | X_0 = x\right]$$

where the state process is a scaled Brownian motion $X_t = \sigma W_t$ and the stopping time $T_{\partial\Omega} = \inf\{t \geq 0 \ : \ X_t \notin (0, 1)\}$ is the first time X exits the unit interval. The calibration problem is to determine what scaling of the reward function $\alpha \sin(\pi x)$ will produce an expected cumulative reward of b if the state process starts at $X_0 = 1/2$.

For a given α the solution to the differential equation is $v(x;\alpha) = \frac{2\alpha}{\pi^2\sigma^2}\sin(\pi x)$. Choosing α to enforce the calibration condition gives $\bar{\alpha} = b\pi^2\sigma^2/2$ so that $\bar{v}(x;\bar{\alpha}) = b\sin(\pi x)$.

3.1.2 Direct Solvers

A basic computational strategy for solving problems of the form (3.1) is to regard $v(x;\alpha)$ as a function of α implicitly defined by equation (3.1a). The calibration problem then reduces to solving the equation M-dimensional equation $C(\alpha) = b$, where $C(\alpha) = F(v(\cdot;\alpha);\alpha)$.

If the map C is well-behaved and its gradient computable, an arsenal of techniques are available for solving this equation. If a good starting point is chosen and $\nabla C(\alpha)$ is nonsingular near the solution, then Newton's method, defined by the iteration

$$\nabla C(\alpha^n)(\alpha^{n+1} - \alpha^n) + C(\alpha^n) = b,$$

is usually the preferred solution technique. For this chapter, we make the simplifying assumption that C is well-behaved and Newton's method converges globally. If C is not well-behaved, more sophisticated and delicate algorithms might be used, a good starting point chosen in advance, or the problem reformulated.

For each iteration of Newton's method, both $C(\alpha^n)$ and $\nabla C(\alpha^n)$ need to be evaluated. If α is M-dimensional, computing the gradient involves solving M linearized versions of equation (3.1a). Newton's method typically converges in a small number of iterations that is independent of the number of parameters. Therefore, the cost of solving $C(\alpha) = b$ this way is proportional to $M + 1$ times the cost of solving equation (3.1a).

For some problems, it may not be convenient to compute the gradient of C. For these problems, several *direct methods* are available to solve the equation using only evaluations of $C(\alpha)$. The most popular of these are the quasi-Newton scheme of Broyden-Fletcher-Goldfarb-Shanno (BFGS) and the Nelder-Mead simplex algorithm [49]. Depending on the problem, these methods typically require a number of function evaluations proportional to either M or $M\log(M)$. The computational cost is then this factor times the cost of solving equation (3.1a).

The complexity of C and the efficacy of the solver determine the number of function evaluations required to solve $C(\alpha) = b$. Little can be done to reduce this. But whatever technique is used, the computational burden is proportional to the cost of solving (3.1a) alone. The multigrid technique presented here aims at reducing this cost.

3.2 A Multigrid Strategy

The basic technique for using multigrid to solve the calibration problem (3.2) is captured in the following two-grid iteration:

* **Algorithm:** $(v^h, \alpha) \leftarrow TG(v^h, \alpha)$

1. Solve the equation $L^h(v^h, \alpha) = f^h$ for v^h exactly (holding α fixed).

2. Find (v^{2h}, α') solving the coarse-grid full calibration problem on Ω_{2h},

$$L^{2h}(v^{2h}, \alpha') = f^{2h}, \tag{3.4a}$$
$$F^{2h}(v^{2h}, \alpha') = b^{2h}; \tag{3.4b}$$

   with the FAS right sides defined as

$$f^{2h} = I_h^{2h}(f^h - L^h(v^h, \alpha)) + L^{2h}(I_h^{2h} v^h, \alpha), \tag{3.4c}$$
$$b^{2h} = \quad b^h - F^h(v^h, \alpha) \quad + \quad F^{2h}(I_h^{2h} v^h, \alpha). \tag{3.4d}$$

3. Update the iterates according to

$$v^h \leftarrow \quad v^h + I_{2h}^h(v^{2h} - I_h^{2h} v^h), \tag{3.5a}$$
$$\alpha \leftarrow \quad \alpha'. \tag{3.5b}$$

4. Solve $L^h(v^h, \alpha) = f^h$ for v^h exactly, (holding α fixed) and return (v^h, α).

This algorithm is essentially the two-grid scheme in Section 2.3.6 with an extreme interpretation of smooth error: For grid functions only the zero function is smooth; for the parameter space all vectors are smooth.[1] In this sense, an exact solver is always an effective, if expensive, smoother. On the fine-grid, the residuals of $L^h(v^h, \alpha) = f^h$ are "smoothed" with an exact solver, while α is not updated since the corresponding errors are already "smooth." The crucial feature of this algorithm is that α is only adjusted on the coarse grid, where seaching for α by repeatedly solving for v is considerably cheaper.

[1] We say all vectors are smooth, since all perturbations have only global effects on v. Therefore, no parameter adjustment is needed on fine levels since the variation is reflected in the coarse-grid problem. It could also be said that only the zero vector is "smooth" since the problem cannot be coarsened in the parameter space.

In this setup, v^h is defined implicitly as a function of α. We therefore interpret this two-grid algorithm as an iteration in α alone, with the fine-grid problem solving the equation $C^h(\alpha) = b^h$ and the coarse-grid problem solving the equation $C^{2h}(\alpha'; \alpha) = b^h$ where C^{2h} reflects the FAS adjustments. According to the defect correction principle in Section 2.3.4, the two-grid scheme can be interpreted in terms of the iteration

$$C^{2h}(\alpha^{n+1}) = b^h - C^h(\alpha^n) + C^{2h}(\alpha^n).$$

The FAS is constructed so the fine-grid solution $\bar{\alpha}$ is a fixed point to this iteration. To assure convergence, the iteration must contract. If C^h and C^{2h} are well-behaved, the condition

$$\varrho\Big(I - \big(\nabla C^h(\bar{\alpha})\big)^{-1} \nabla C^{2h}(\bar{\alpha})\Big) << 1, \tag{3.6}$$

enforces the contraction property for α close to $\bar{\alpha}$. This algorithm should be efficient when the contraction factor is small and the coarse-grid problem is much cheaper to solve than the fine-grid problem.

Remark 2. *The basic principle of this algorithm is to defer the search for α to the coarse grid where solving for v is cheap. A solver for $C^{2h}(\alpha) = b$ is still needed. A natural choice for this solver is a quasi-Newton method with a Jacobian updating scheme such as the one devised by Broyden-Fletcher-Goldfarb-Shanno (BFGS). This method is outlined in [49]. One appealing feature of the BFGS scheme is that it can be initialized with an estimate of the Jacobian already computed. If this estimate is accurate and the iteration is started near the solution, performance close to Newton's method can be expected for well-behaved problems. Since each of many V-Cycles calls the direct solver on the coarsest grid, reusing the Jacobian estimate from previous V-Cycles can significantly reduce runtime if the coarse-grid problem has several parameters and starts near the solution.*

The Example Continued

We now explore the performance of the two-grid scheme using a discrete version of the model problem (3.3):

$$\alpha \sin(\pi x_j^h) + (A^h v^h)_j = 0, \text{ with } \quad (A^h v^h)_j = \frac{\sigma^2}{2} \frac{v_{j+1}^h - 2v_j^h + v_{j-1}^h}{h^2}, \tag{3.7a}$$

$$\text{for } j = 1, \ldots, N_h - 1, \text{ and}$$

$$v_0^h = v_{N_h}^h = 0,$$

$$v^h(\frac{1}{2}) = b. \tag{3.7b}$$

The solution v^h for a fixed α is

$$v_j^h = -\frac{\alpha}{\widehat{A}^h(\pi)} \sin(\pi x_j^h), \text{ where} \tag{3.8}$$

$$\widehat{A}^h(\theta) = -\frac{\sigma^2}{h^2}\big((1 - \cos(\theta)\big).$$

Since the two-grid algorithm solves exactly equation (3.7a) at the fine level, the corresponding residuals are eliminated and the coarse-grid problem prescribed by the FAS takes a given v^h and α and seeks v^{2h} and α' that solve

$$\alpha' \sin(\pi x^{2h}) + A^{2h} v^{2h} = \alpha \sin(\pi x^{2h}) + A^{2h} I_h^{2h} v^h, \tag{3.9a}$$

$$\text{for } j = 1, \ldots, N_{2h} - 1, \text{ and}$$

$$v_0^{2h} = v_{N_{2h}}^{2h} = 0,$$

$$v^{2h}(\frac{1}{2}) = b. \tag{3.9b}$$

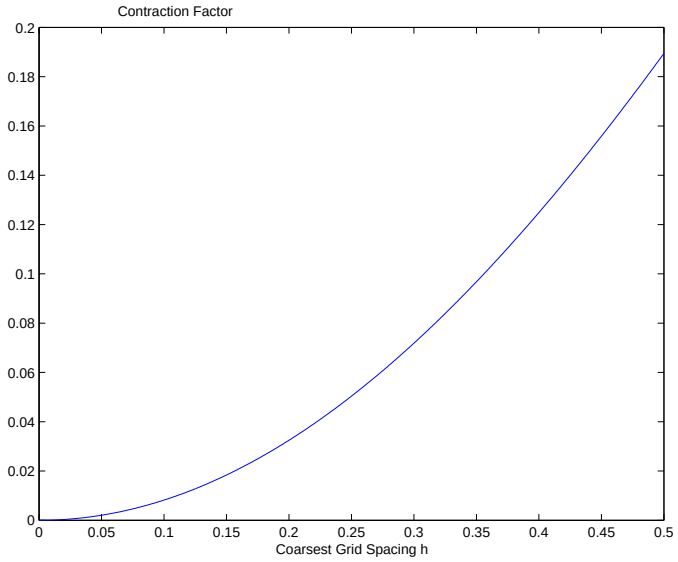
If injection is used to restrict grid functions and residuals, the solution to the coarse-grid problem is

$$\alpha' = \alpha\Big(1 - \frac{\hat{A}^{2h}(2h\pi)}{\hat{A}^h(h\pi)}\Big) - \hat{A}^{2h}(2h\pi), \tag{3.10}$$

$$v^{2h} = b \sin(\pi x^{2h}).$$

We could apply this two-grid scheme with coarse and fine-grid spacing other than $2h$ and h. Equivalently, we could build a V-Cycle from this two-grid scheme. If the fine-grid has spacing $2^l h_0$ for $l \geq 1$ and the coarse-grid has spacing h_0, the coarse-grid correction defines

Figure 3.1: Contraction factors in equation (3.11) in the limit as $l \to \infty$.



a fixed point iteration in α with a contraction rate given by

$$1 - \frac{\hat{A}^{h_0}(h_0\pi)}{\hat{A}^{h_0/2^l}(h_0\pi/2^l)}. \tag{3.11}$$

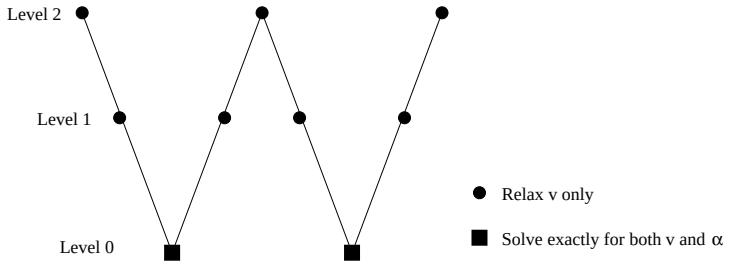
Figure 3.1 shows that when the spacing of the coarsest grid becomes too large, the coarse-grid problem can no longer reasonably mimic the fine-grid problem's variations in α.

This two-grid algorithm can be viewed as a special case of the scheme by Lewis & Nash [37] for optimizing functionals subject to generic PDE constraints. They use the FAS to construct a coarse-grid optimization problem and use an exact solver rather than a smoother to handle the PDE constraints.

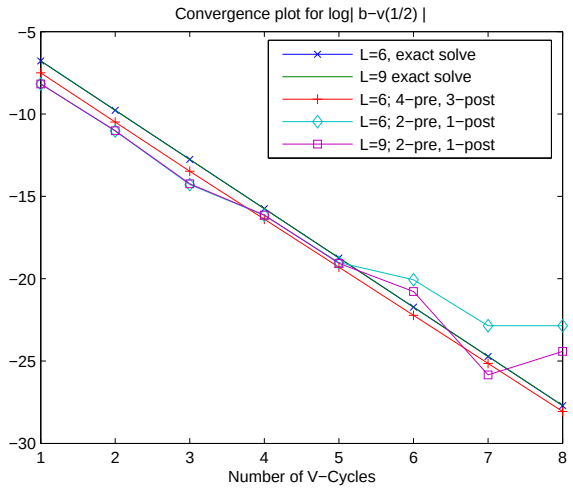
For the example at hand, using an exact solver to smooth errors on the fine-grid is unnecessarily expensive. Ta'asan [7, 55, 56] considers optimzation problems subject to constraints govered by elliptic PDE and applies FAS to the optimality conditions, which form an elliptic system that can be efficiently smoothed.

To build a more efficient V-Cycle, we use a few Gauss-Seidel smoothing sweeps on the fine-grid. This is shown in Figure 3.2. For the example problems, a few smoothing sweeps per level gives performance similar to using an exact solver for the fine-grid equation $L^h(v^h, \alpha) = f^h$. Figure 3.3(b) shows convergence plots for errors in α using V-cycles in various configurations. For all configurations, errors are reduced by more than a factor of twelve in each V-Cycle. Using four smoothing sweeps per level gives identical performance to using an exact solver.

Figure 3.2: V-Cycle performance.



(a) V-cycles for a calibration problem. Parameters are adjusted only on the coarsest grid while the value function is relaxed at every level.



(b) Convergence plots for the residuals of the calibration condition for V-cycles with various smoother configurations and depths.

* **Algorithm:** $(v^h, \alpha) \leftarrow \text{V-Cycle}(L, v^h, \alpha, f^h, b^h)$

1. If $L = 0$, apply a direct solver to the full calibration problem on the coarsest grid, equations (3.2), and return the solutions v^h and α. Else,
2. Apply a smoother to $L^h(v^h, \alpha) = f^h$ (holding α fixed.
3. Compute the FAS coarse-grid right sides, f^{2h} and b^{2h}, according to equations (3.4).
4. Set $(v^{2h}, \alpha') \leftarrow \text{V-Cycle}(L-1, I_h^{2h} v^h, \alpha, f^{2h}, b^{2h})$.
5. Apply the corrections (3.5).
6. Apply a smoother to $L^h(v^h, \alpha') = f^h$ (holding α fixed). Return v^h and α.

If only two pre-smoothing and one post-smoothing sweeps are used, convergence is still favorable, but the rate is not uniform when errors become very small.

3.3 The Need for τ-Expansion

We now consider a more challenging calibration problem that requires a modification of the FAS to achieve acceptable convergence rates. We seek to find a parameter α and a grid function v^h defined over Ω_h (embedded in $[0,1]$) such that

$$\sin(\pi x/2) + \alpha^2\Big[\frac{v^h_{i+1} - v^h_{i-1}}{2h}\Big] + \alpha\Big[\frac{v^h_{i+1} - 2v^h_i + v^h_{i-1}}{h^2}\Big] = 0, \tag{3.12a}$$

$$\text{for } j = 1, \ldots, N_{2h} - 1, \text{ and } \quad v^h_0 = v^h_{N_h} = 0,$$

$$\frac{v^h_{N_h} - v^h_{N_h-1}}{\alpha^2 h} = b. \tag{3.12b}$$

The FAS scheme (3.4) defines right side vectors that adjust for relative discretization in error in v. This scheme neglects to account for the effects on v caused by changes in α done on the coarsest level. To capture these effects, we replace the usual FAS right sides (3.4c,3.4d) with functions that depend on the changes made to α on the coarsest grid.

$$f^{2h}(\alpha') = I^{2h}_h f^h + \underbrace{L^{2h}(I^{2h}_h v^h, \alpha') - I^{2h}_h L^h(v^h, \alpha')}_{\tau^f(\alpha')}$$

$$b^{2h}(\alpha') = \quad b^h + \underbrace{F^{2h}(I^{2h}_h v^h, \alpha') - F^h(v^h, \alpha')}_{\tau^b(\alpha')}$$

Often, it will suffice to introduce only a first-order τ-expansion to the right side.

$$f^{2h}(\alpha') = I^{2h}_h f^h + \nabla_\alpha \tau^f(\alpha)(\alpha' - \alpha)$$

$$b^{2h}(\alpha') = \quad b^h + \nabla_\alpha \tau^b(\alpha^h)(\alpha' - \alpha)$$

In either case, this technique keeps track of how changing α affects the discretization errors of v at various levels. Figure 3.4(a) shows how the coarse-grid calibration condition residual for the easier problem (3.7) varies with α. No τ-expansion is required since the coarse-grid correction essentially shifts the calibration residuals in parallel. This can be seen from equation (3.8). For the harder problem (3.12), the calibration condition residuals on the coarse-grid have more significant variation in α, and τ-expansion is needed to ensure a good

two-grid contraction rate. Indeed, when using the standard FAS (zeroth order τ-expansion), error reduction factors for the two-grid scheme is too small for practical purposes (about 3.6). In contrast, with full and linear τ-expansion, the respective factors are favorable (10.9 and 13.7).

Based on the two-grid scheme, an effective V-cycle or FMG-cycle can be implemented. Figure 3.4(b) compares the V-Cycle convergence of an implementation of the full τ-expansion scheme with that of the standard FAS. When using τ-expansion, the V-Cycle is still effective when only a few smoothing sweeps are used on each fine level.

3.4 An Example from Economics

To illustrate these techniques, we apply them to numerically calibrate a model recently developed by Farhi & Panageas [24] and Dybvig & Liu [23]. The model describes the consumption and investment problem of a utility-maximizing individual who has claim to labor income and an option to retire at any time. This model builds on the work of Duffie et. al. [22] and Koo [34], who incorporate the effects of labor income into the classic model of Merton [42]. Munk [44, 45] uses the techniques of Kushner & Dupuis [35] to solve these models numerically. We build on this work, incorporating the retirement option and parameter calibration into the numerical analysis.

The model posits an investor who trades in a Black-Scholes-Merton financial market with interest rate $r > 0$ and a stock price evolving according to the geometric Brownian Motion

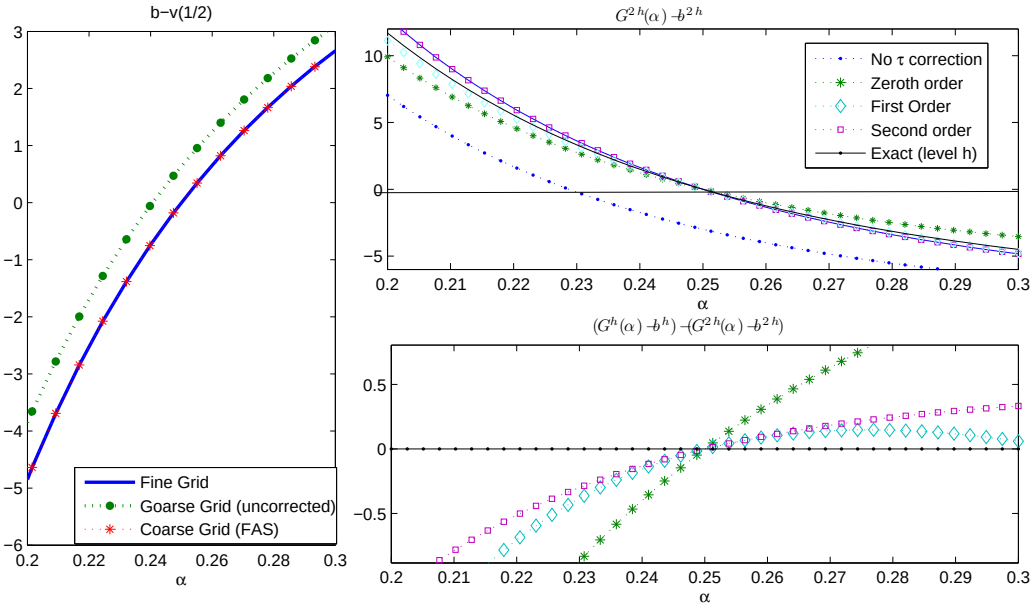
$$\frac{dS_t}{S_t} = (\mu + r)dt + \sigma dW_t.$$

The positive constants μ and σ govern the excess expected rate of return and the volatility of the stock. In this version of the model, stock price variations are the only source of uncertainty.

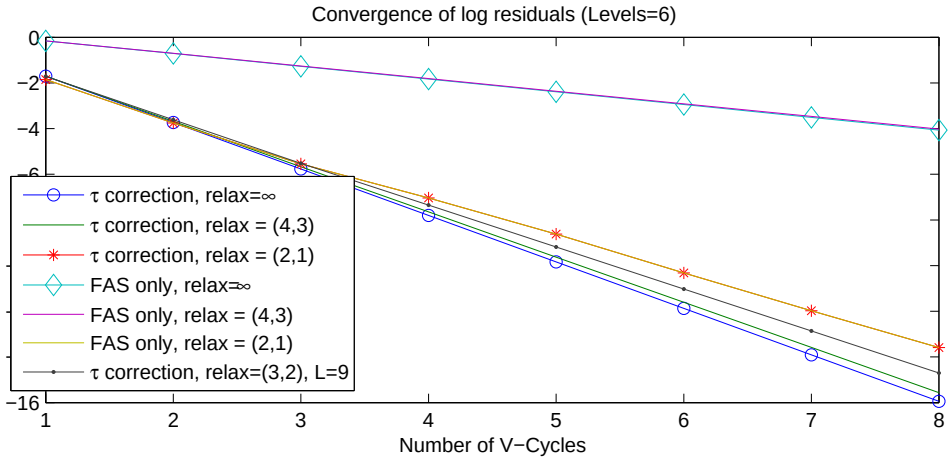
The investor has claim to labor income that grows at rate κ until he chooses to retire. After this time, he receives no wage income but has more leisure time. Let T_{R} denote the stopping time associated with the investor's retirement. Then, the wage process Y_t follows simple geometric growth until retirement.

$$\begin{aligned} \frac{dY_t}{Y_t} &= \kappa dt \text{ on } [0, T_{\text{Retire}}) \\ Y_t &= 0 \quad \text{ on } [T_{\text{Retire}}, \infty) \end{aligned}$$

Figure 3.3: Illustrating τ-expansion.



(a) The left panel shows how the calibration condition varies with α for the easy problem (3.3). The right shows the same for the hard problem (3.12) and the effects of τ-expansion.



(b) Convergence plots for the residual of the calibration condition for various smoother configurations. $N = 7$ grid points were used on the coarsest grid and the V-Cycle is L=6 levels deep. FAS and full τ-correction schemes are considered. Error reductions for plain FAS are less than 2 per V-Cycle. With τ-expansion they are in the range 5–7.

If the investor chooses portfolio and consumption processes p and c, his financial wealth evolves from $X_0 \geq 0$ according to the SDE

$$dX_t = \big(X_t r + p_t \mu + (Y_t - c_t)\big)dt + \sigma p_t dW_t.$$

The agent must choose only portfolio and consumption processes that respect the constraint $X_t \geq 0$ for all t with probability one. This admissibility constraint both excludes the use of doubling strategies and prevents the agent from borrowing against unrealized labor income. We denote by $\mathcal{X}(x, y)$ the set of all retirement times and predictable consumption and portfolio processes associated with an admissible wealth and wage processes (X_t, Y_t) that start at $(X_0, Y_0) = (x, y)$. The investor's problem is then characterized by the value function[2]

$$v(X_0, Y_0) = \operatorname*{ess\,sup}_{c,p,T_{\mathrm{R}} \in \mathcal{X}(x,y)} E\Big[\int_0^{T_{\mathrm{R}}} e^{-\beta t} u(c_t)dt + \quad e^{-\beta T_{\mathrm{R}}} v(\alpha X_{T_{\mathrm{R}}}, 0) | X_0, Y_0\Big], \tag{3.13}$$

where $\beta > 0$ is interpreted as a mortality rate, and the utility function is $u(x) = \frac{x^{1-\gamma}}{1-\gamma}$ for a risk aversion parameter $\gamma > 1$. The parameter $\alpha > 1$ reflects the investor's willingness to substitute leisure time for income.

3.4.1 Characterizing the Value Function

After the agent retires, the income process vanishes. This implies $v(x, 0)$ solves the classical problem of consumption and investment of Merton [42]. The associated value function is $v^{\mathrm{Merton}}(x) = Ax^{1-\gamma}$, where the relevant constant is

$$A = \Big(\frac{\beta - r(1-\gamma)}{\gamma} - \frac{1-\gamma}{2\gamma^2}\frac{\mu^2}{\sigma^2}\Big)^{-\gamma}.$$

The following system of HJB complementary inequalities characterizes the value function:

$$\sup_{c,p} u(c) - \beta v + [rx + \mu p + (y - c)]v_x + \frac{1}{2}\sigma^2 p^2 v_{xx} + \kappa y v_y \leq 0 \tag{3.14a}$$

$$v(x, y) \geq v^{\mathrm{Merton}}(\alpha x), \tag{3.14b}$$

with at least one of the inequalities binding for each $(x, y) \in [0, \infty)^2 - \mathbf{0}$.

[2]Existence, uniqueness, and regularity solutions are established in [24] and [23]. They solve the case we consider here explicitly using duality techniques. With minor modifications our numerical scheme extends to the cases that admit no explicit solutions.

The problem's homothetic structure allows the value function to be factored as $v(x,y) = V(\frac{x}{y})y^{1-\gamma}$ for an appropriate function of one variable, $V(z)$. The relevant state variable is the wealth-to-wage ratio $Z_t = X_t/Y_t$, and the optimal controls satisfy

$$c = y(V_z)^{-1/\gamma}, \quad p = y\frac{-\mu V_z}{\sigma^2 V_{zz}},$$

so that $\hat{c} = c/y$ and $\hat{p} = p/y$ depend only on $z = x/y$ for $z \geq 0$. With these substitutions we derive corresponding complementary HJB inequalities for V:

$$\sup_{\hat{c},\hat{p}} u(\hat{c}) + (\kappa(1-\gamma) - \beta)V + \big((r-\kappa)z + \mu\hat{p} + (1-\hat{c})\big)V_z + \frac{1}{2}\sigma^2\hat{p}^2 V_{zz} \leq 0, \tag{3.15a}$$

$$V(z) \geq v^{\text{Merton}}(\alpha z) \tag{3.15b}$$

with at least one of these inequalities binding for each $z \geq 0$.

This formulation is probably the best point of departure for numerical treatment if one is only interested in solving the retirement problem. For the calibration problem considered in Section 3.4.2, this formulation is not appropriate so we consider an alternative.

The relations (3.15) are the optimality conditions corresponding to the stochastic control problem defined by

$$V(Z_0) = \operatorname*{ess\,sup}_{\hat{c},\hat{p},T_{\text{R}}} \; E\Big[\int_0^{T_{\text{R}}} e^{(\kappa(1-\gamma)-\beta)t}u(\hat{c}_t)dt + e^{-\beta T_{\text{R}}}v^{\text{Merton}}(\alpha Z_{T_{\text{R}}})|Z_0\Big]$$

where the controlled process Z follows the SDE

$$dZ_t = (r-\kappa)Z_t + \mu\hat{p}_t + (1-\hat{c})dt + \sigma\hat{p}dW_t$$

and $Z \geq 0$. The relevant stopping times are characterized by the first time Z_t hits a given level indexed by z^*. If the agent retires when $Z_t = z^*$, then the associated value function satisfies the HJB equation

$$\sup_{\hat{c},\hat{p}} u(\hat{c}) + (\kappa(1-\gamma) - \beta)V + \big((r-\kappa)z + \mu\hat{p} + (1-\hat{c})\big)V_z + \frac{1}{2}\sigma^2\hat{p}^2 V_{zz} = 0 \tag{3.16a}$$

for each $z \in [0, z^*)$, along with the *value matching* condition that states

$$V(z^*) = v^{\text{Merton}}(\alpha z^*). \tag{3.16b}$$

The boundary condition associated with $z = 0$ is implicit in equation (3.16a) since the admissibility condition $X_t \geq 0$ implies the agent holds no stock when $Z_t = 0$. This removes the second order term and we interpret V_z in the one-sided sense. With V defined in terms of z^*, the task is to choose z^* to maximize $V(0)$. The associated optimality condition is

$$\frac{d}{dz}V(z^*) = \frac{d}{dz^*}v^{\text{Merton}}(\alpha z^*), \tag{3.16c}$$

which is called the *smooth pasting* condition[3]. It says that at the instant the agent chooses to retire, he is indifferent between retiring and not.

3.4.2 Calibration

Equations (3.16) define the retirement model for a given set of parameters. Most of the parameters can be estimated from market data or calibrated using a simpler model. One approach to identifying the parameter α is to set up the model so that the investor chooses to retire in about, say, fifty years. Mathematically, the expected retirement time is

$$R(z) = E\left[T_{\text{Retire}} | Z_0 = z\right].$$

The calibration problem consists of choosing α so that $R(0) = \mu_{\text{Retire}}$ for a given target μ_{Retire}. If the controls $(\hat{c}, \hat{p})$ and retirement boundary z^* are given, R satisfies the equation

$$1 + \big((r-\kappa)z + \mu\hat{p} + (1-\hat{c})\big)R_z + \frac{1}{2}\sigma^2\hat{p}^2 R_{zz} = 0 \quad \text{on } [0, z^*) \tag{3.17a}$$

$$R(z^*) = 0. \tag{3.17b}$$

From a computational viewpoint, the full calibration problem consists of choosing α such that:

1. The value function V and the free boundary z^* solve the HJB equation (3.16a)-(3.16b) on $[0, z^*)$ together with the smooth pasting condition (3.16c),
2. The function R satisfies the system (3.17) on $[0, z^*]$, and
3. The calibration condition $C(\alpha) = \mu_{\text{Retire}}$ is satisfied where $C(\alpha) = R(0)$.

[3]The smooth pasting condition is presented informally in [20] and [54]. A rigorous account of the applicability of this condition in a particular context can be found in [13].

Said differently, we want to recover α by inverting the mapping

$$\alpha \mapsto (V, \hat{c}, \hat{p}, z^*) \mapsto R(z) \mapsto C(\alpha) = \mu_{\text{Retire}}$$

where V and R are regarded as implicit functions of α. Having formulated the optimal retirement problem in a form consistent with a generic calibration problem, we apply the techniques developed in this chapter to solve it numerically.

3.4.3 Numerical Treatment

Conditioning is usually improved by changing variables when the state processes grow geometrically, usually by taking logarithms. Since Z is the ratio of X and Y which both grow geometrically, transforming Z is appropriate. But Z can take the value zero with positive probability (the agent starts with zero wealth), so we use a transformation that behaves like $\log(Z)$ for large z and like the identity map for z near zero, defining $w = \psi(z) = \log(z+1)$ so that $z = \psi^{-1}(w) = e^w - 1$, which behaves like z up to first order for z near zero. Substituting the function $U(\log(z+1)) = V(z)$ into equations (3.16) and (3.17) and making the identifications $\hat{q} = \frac{\hat{p}}{1+z}$ and $z = e^w - 1$, we find the function U satisfies the equation

$$\sup_{\hat{c},\hat{q}} \frac{\hat{c}^{1-\gamma}}{1-\gamma} + (\kappa(1-\gamma) - \beta)U + \underbrace{\left(\frac{(r-\kappa)z}{1+z} + \frac{1-\hat{c}}{1+z} + \mu\hat{q} - \frac{\sigma^2\hat{q}^2}{2}\right)}_{m(\hat{c},\hat{q},z)} U_w + \underbrace{\frac{\sigma^2\hat{q}^2}{2}}_{s(\hat{q},z)} U_{ww} = 0,$$

over $[0, w^*)$ and the boundary conditions

$$\begin{aligned} U(w^*) &= v^{\text{Merton}}(\alpha z^*)\big|_{z^*=(e^{w^*}-1)}, \\ U_w(w^*) &= e^w \frac{d}{dz^*} v^{\text{Merton}}(\alpha z^*)\big|_{z^*=(e^{w^*}-1)}. \end{aligned}$$

Similarly, the function $J(w) = R(e^w - 1)$ satisfies

$$\begin{aligned} 1 + m(\hat{c}, \hat{q}, z)J_w + s(\hat{q}, z)J_{ww} &= 0, \\ J(w^*) &= 0. \end{aligned}$$

Finite Difference Discretization

We use a finite difference scheme over a uniform grid spanning $[0, w^*]$. During the iterative process the whole grid expands or contracts to accommodate changes in w^*, which in our

formulation must be a grid point. The grid is expressed as

$$\Omega_h(w^*) = \{w_j = w^* \frac{j}{N} = w^* jh \quad \text{for} \quad j = 0, 1, \ldots, N+1\}.$$

Given a parameter μ_{Retire} and a grid $w_j = w^* jh$, the discrete calibration problem is to find scalars $\alpha > 1$ and $w^* > 0$ so that

1. The grid function U satisfies the discrete HJB equation[4]

$$\max_{\hat{c},\hat{q}} \frac{\hat{c}^{1-\gamma}}{1-\gamma} - (\kappa(1-\gamma) - \beta)U_j + m(\hat{c}, \hat{q}, z_j)\frac{U_{j+1} - U_j}{w^* h} + s(\hat{q}, z_j)\frac{U_{j+1} - 2U_j + U_{j-1}}{(w^* h)^2} = 0, \quad \text{for } j = 1, \ldots, N, \tag{3.18a}$$

$$\sup_{\hat{c},\hat{p}} \frac{\hat{c}^{1-\gamma}}{1-\gamma} - (\kappa(1-\gamma) - \beta)U_0 + m(\hat{c}, \hat{q}, 0)\frac{U_1 - U_0}{w^* h} = 0, \tag{3.18b}$$

$$\frac{U_{N+1} - U_{N-1}}{2w^* h} = e^{w_N} \frac{d}{dz^*} v^{\text{Merton}}(\alpha z^*)\big|_{z^* = (e^{w^*} - 1)}, \tag{3.18c}$$

2. The grid function J satisfies

$$1 + m(\hat{c}, \hat{q}, z_j)\frac{J_{j+1} - J_j}{w^* h} + s(\hat{q}, z_j)\frac{J_{j+1} - 2J_j + J_{j-1}}{(w^* h)^2} = 0, \quad \text{for } j = 1, \ldots, N, \tag{3.19a}$$

$$1 + m(\hat{c}, \hat{q}, 0)\frac{J_1 - J_0}{w^* h} = 0, \tag{3.19b}$$

$$J_N = 0, \tag{3.19c}$$

[4]Even if the continuous problem is well-posed and has well-behaved controls, the same may not be true of the discrete problem. Following Kushner & Dupuis [35], to ensure well-posedness we require a bound on the controls to ensure the HJB equation remains well-posed:

$$0 < \hat{c}_{\min} \leq \hat{c} \leq 1 + B_1(1 + z),$$
$$0 \leq \hat{q} \leq B_2$$

for some positive constants $\hat{c}_{\min}, B_1, B_2$. These constraints do not bind in the final solution but are imposed to enforce the contraction property of iterative schemes. These constants must be chosen by experimentation and economic reasoning. Since controls vary over a compact set, we are justified in replacing the sup operation with the max operation.

3. The calibration condition $J_0 = \mu_{\text{Retire}}$ and the value matching condition

$$U_N = v^{\text{Merton}}(\alpha z^*)\big|_{z^*=(e^{w^*}-1)} \quad \text{hold.}$$

The HJB equation has two boundary conditions associated with w^*. To define U we use the smooth pasting condition. The location of w^* is associated with the value matching condition. For purposes of a multigrid algorithm, w^* can be thought of as another parameter. In this regard, the calibration problem can be thought of as inverting a two-dimensional map:

$$\begin{pmatrix} \alpha \\ w^* \end{pmatrix} \mapsto (U, \hat{c}, \hat{q}) \mapsto J \mapsto \begin{pmatrix} J_0 \\ U_N \end{pmatrix} = \begin{pmatrix} \mu_{\text{Retire}} \\ v^{\text{Merton}}(\alpha z^*(w^*)) \end{pmatrix}.$$

3.4.4 Multigrid Treatment

To smooth errors in U and J, we use a Gauss-Seidel scheme sweeping from $j = N, \ldots, 1$. This is consistent with the one-sided discretization of U_w and that agents in this model tend to get more wealthy over time. A two-sided approximation would result in an operator that is not h-elliptic near $w = 0$ since $q = 0$ there. First U is relaxed and optimal controls are computed; then J is relaxed. For the HJB equation corresponding to U, it is difficult to solve for U_j explicitly in terms of other quantities. Instead, we relax at point w_j by first choosing the controls $\hat{c}$ and $\hat{q}$ that attain the maximum in equation (3.18a) and then solving for U_j with everything else held fixed. This is essentially a local version of Newton's method applied within the Gauss-Seidel iteration.

We transfer U and J to coarser and finer grids by injection and linear interpolation. Residuals are also transfered by injection.

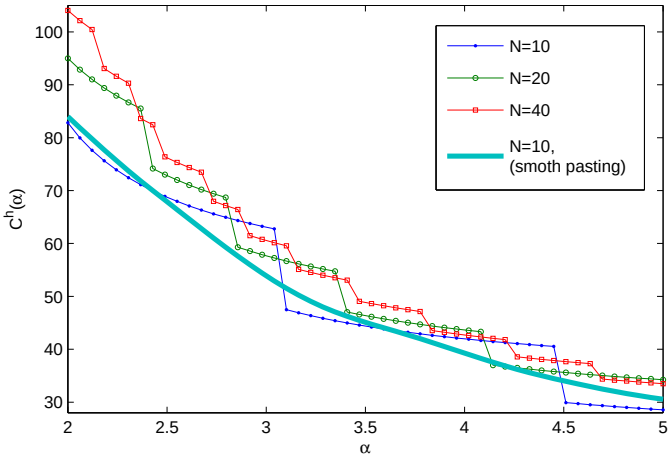
The Need for τ-Expansion

All equations in the system (3.18) have strong nonlinear dependence on w^* or α. The smooth pasting condition has dependence on both. Numerical experiments indicated that calibration with the standard FAS fails unless the coarsest grid is unreasonably fine. We therefore consider a multigrid scheme based on a full τ-expansion in w^* and α.

Finding a starting point

We require a good starting point from which to begin our interative scheme. This is because the smooth pasting formulation of the retirement problem is based on a local optimality

Figure 3.4: The calibration problem is ill-posed when the standard discrete dynamic programming formulation is used. For critical levels of α the stopping boundary moves discontinuously. This causes the expected retirement time to jump as well. The smooth pasting condition ensures the calibration condition varies smoothly in α.



condition and there are spurious solutions. A more robust formulation of the retirement problem is based on the dynamic programming formulation based on equations (3.15). To approximate the retirement problem for a given α, we could write the discrete problem as a dynamic programming equation consistent with the system (3.15):

$$V_j^h = \max\left(v^{\text{Merton}}(\alpha z^*), f_j \Delta t + \sup_{\hat{c},\hat{p}} P^h_{j\to j+(-1,0,1)}(\hat{c},\hat{p})' \begin{pmatrix} v^h_{j+1} \\ v^h_j \\ v^h_{j-1} \end{pmatrix}\right) \tag{3.20}$$

where $j = 0, 1, \ldots$ and $P^h_{j\to j+(-1,0,1)}$ represents the local transition probabilities implicit in the finite difference discretization. The uniqueness theorems of Kushner & Dupuis show this formulation is robust and that Gauss-Seidel relaxation converges from an initial guess of $V^h = 0$.

Remark 3. *While equation* (3.20) *is a reasonable way to solve the retirement problem, it is not appropriate for the calibration problem. Figure 3.4 shows that expected retirement times are not continuous in α when using equation* (3.20) *to define w^*, so the calibration problem is ill-posed at the discrete level. This may not matter for very fine-grids, but our calibration approach adjusts α only on the coarsest grid where these distortions are large. In contrast, the location of the free boundary, and the expected retirement times, vary smoothly in α with formulation based on the smooth pasting condition.*

To find a starting point for the calibration problem, we proceed by choosing an economically reasonable $\alpha > 1$ and solving the equation (3.20) on a coarse-grid. The approximate retirement boundary w^* can be located by the smallest index j such that $U_j = v^{\text{Merton}}(\alpha z^*)$. This point can be used to determine an initial guess for w^*.

Given w^* and α, the smooth pasting formulation can then be used to solve the discrete retirement problem for U^h, w^*, and J^h. At this point, the target retirement time is initialized $\mu_{\text{Retire}} = J_0$.

A continuation method is then used to move μ_{Retire} continouously toward the target level while adjusting α and w^* locally to compensate so that the calibration problem stays solved in the continuation process. This results in a solution to the calibration problem on the coarsest grid. It can also be used as the basis of a coarse-grid solver in a V-Cycle.

3.4.5 Numerical Results

For a numerical experiment, we chose parameters consistent with the study of Dybvig & Liu [23]. The numerical values are listed in Figure 3.5 which shows the value function, consumption and portfolio rules, and expected retirement times. The model was calibrated to $b = 38$ years, which implies $\alpha \approx 3$ depending on the level of discretization.

Each V-Cycle consists of three parts. First, the optimal control functions, $\hat{c}$ and $\hat{q}$, are computed in terms of U and w^*. For the rest of the V-Cycle, the controls are frozen at these values. Then a V-Cycle is performed with α held fixed as U, J, and w^* are updated. This V-Cycle is done so that the value function reflects the changes made in optimal controls. A second V-Cycle is then done where α is adjusted along with U, J, and w^*. Again, w^* and α are adjusted only on the coarsest grid. Figure 3.7 shows the convergence history of several V-Cycles using $L = 5$ levels and three pre- and post- smoothing Gauss-Seidel sweeps for U and J at each level.

The cost of each V-Cycle is about three or four times that of a similar iteration that solves for the value function alone with a fixed parameter α. Indeed, this corresponds to first inner V-Cycle of our iteration. The second inner V-Cycle must relax both U and J, so it incurs roughly twice the cost. Convergence rates for the calibration problem are similar to the textbook multigrid rates observed for the retirement problem. We therefore conclude that the calibration problem does not require much more computation than does solving the model.

Figure 3.5: The solution to the calibration problem with parameters $r = 2\%, \mu = 4\%, \sigma = 15\%, \beta = 1\%$, $\kappa = 0, \gamma = 2, b = 38$ years. The grid spacing in this example is $h = 1.35^{-4}$. The value function is shown with the reward associated with retiring. Smooth-pasting and value-matching conditions can be seen on the right where the wealth-to-wage ratio reaches the retirement threshold. To compare results against the benchmark of Merton's model (shown as a dotted line), consumption and portfolio policies $\hat{c}$ and $\hat{p}$ are normalized by $(Z + 1/(r - \kappa))$, which is a proxy for financial wealth plus the value of labor income. Since the investor retires and cannot short his labor income, his valuation is less than this.

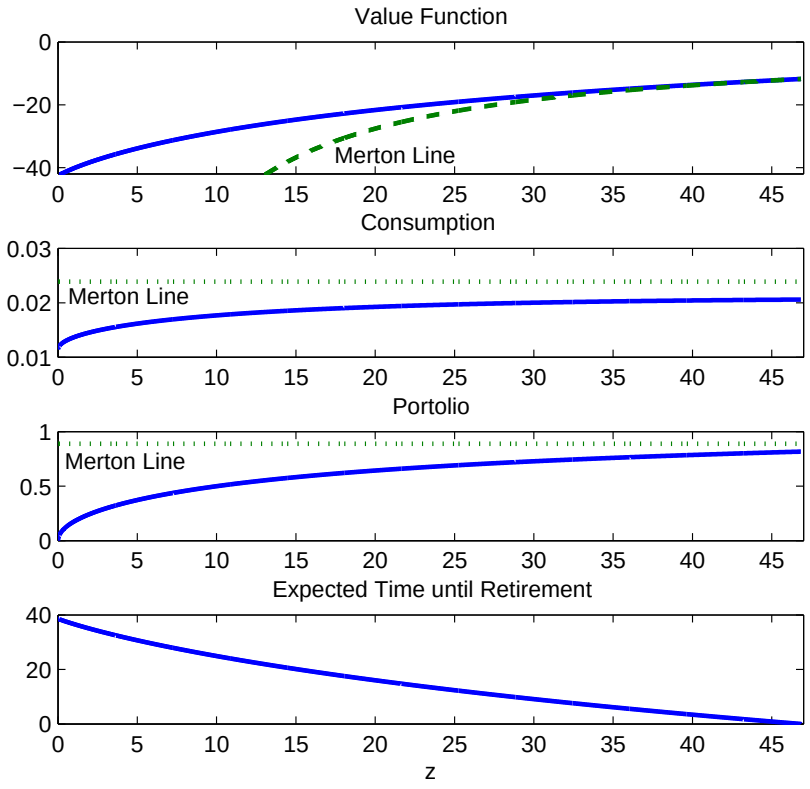


Figure 3.6: The discretization error in calibrated parameter is proportional to h.

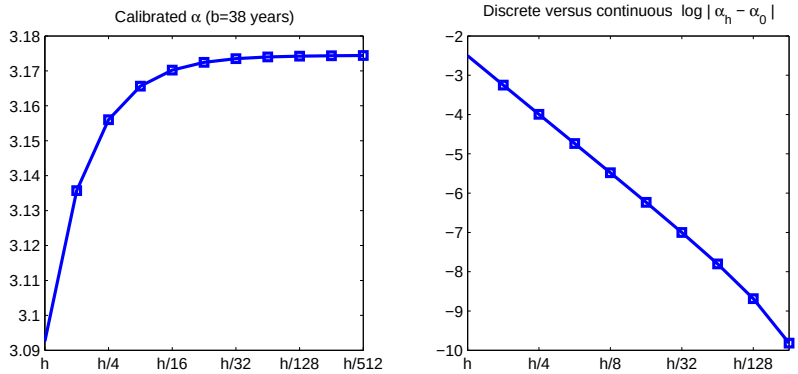
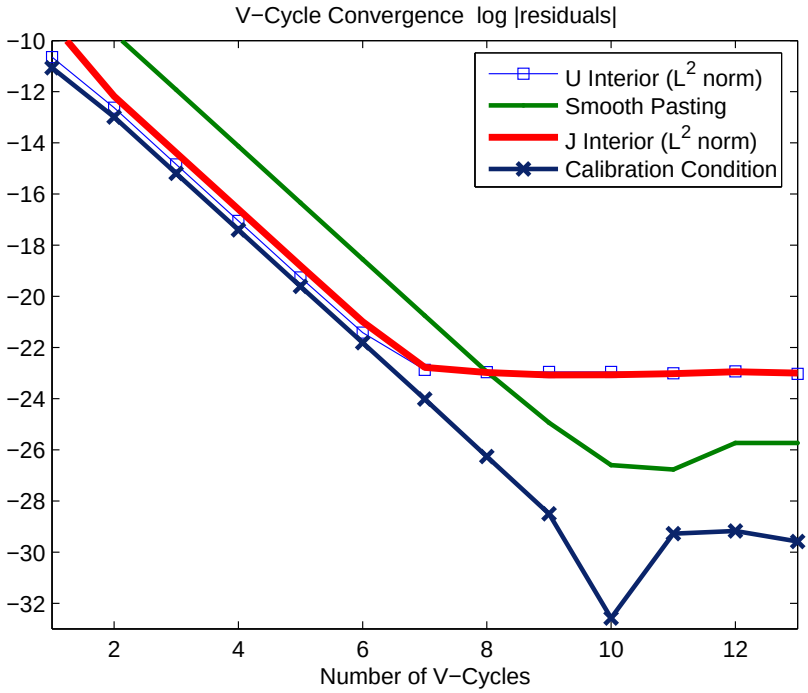


Figure 3.7: A V-Cycle reduces all errors by about 8.6. Here $L = 5$ levels were used with a coarsest grid with $N = 12$ grid-points. The starting point was the cubic interpolant of the solution corresponding to level $L - 1 = 4$. The convergence history of the V-Cycles at all levels is shown in Appendix A.



Chapter 4

An Infinite Dimensional Example

Following the developments in the previous chapter, we consider a problem of the form

$$L(v;\alpha) = f$$
$$F(v;\alpha) = b$$

where the first equation determines the value function for a stochastic expectation or control problem. In this chapter, we take a first step toward solving problems where the parameter α is an infinite-dimensional object such as a function. Specifically, we study classical optimality conditions associated with an optimal stopping problem driven by a two-dimensional Brownian motion. While this is not a calibration problem, the optimality conditions take the form of a calibration problem and serve as a laboratory in which we can develop and test multigrid schemes. Techniques from Optimal Shape Design (OSD), Fourier analysis of pseudo-differential operators in particular, are used to design a multigrid solver for these equations.

In the course of the numerical analysis, we discover three properties of the optimal stopping problem. First, a naive discretization of the classical formulation (based on the smooth-pasting condition) results in an badly-conditioned problem. In contrast, the modern formulation of the problem, which consists of a system of variational inequalities that are understood in terms of viscosity solutions, forms a well-conditioned problem. We identify the source of the bad conditioning in the classical formulation and suggest a regularization scheme to remedy the conditioning problem. Our regularization scheme is simply a reformulation and does not distort the solution as would a scheme based on inexact penalties. Finally, by studying the structure of the classical formulation, we are able to identify a probabilistic interpretation of the classical formulation that is difficult to recover from the

modern formulation.

While our study of the optimal stopping problem in its classical form deepens our understanding of this class of problems, the resulting solvers are no faster than the best of those based on the modern variational formulation. Still, developing techniques for solving the stopping problem posed in the classical form helps us develop methods for solving calibration problems that admit no clean alternative formulation. Furthermore, many calibration problems are ill-posed and are badly conditioned like the naive formulation of the stopping problem. In this regard, we expect the methods developed here and borrowed from OSD will be helpful in formulating and solving calibration problems from a variety of financial models based on dynamic programming.

4.1 The Problem

We begin with a standard Brownian Motion in $\mathbb{R}^2$ denoted by (X_t, Y_t), and define the stopping time τ_0 to be the first time X_t hits zero. The *value function* of the optimal stopping problem is defined as

$$v(x,y) = \operatorname*{ess\,sup}_{\tau} E\left[B(Y_{\tau_0})\mathbb{1}_{(\tau_0<\tau)} - \int_0^{\tau_0\wedge\tau} g(X_t,Y_t)dt | X_0 = x, Y_0 = y\right]$$

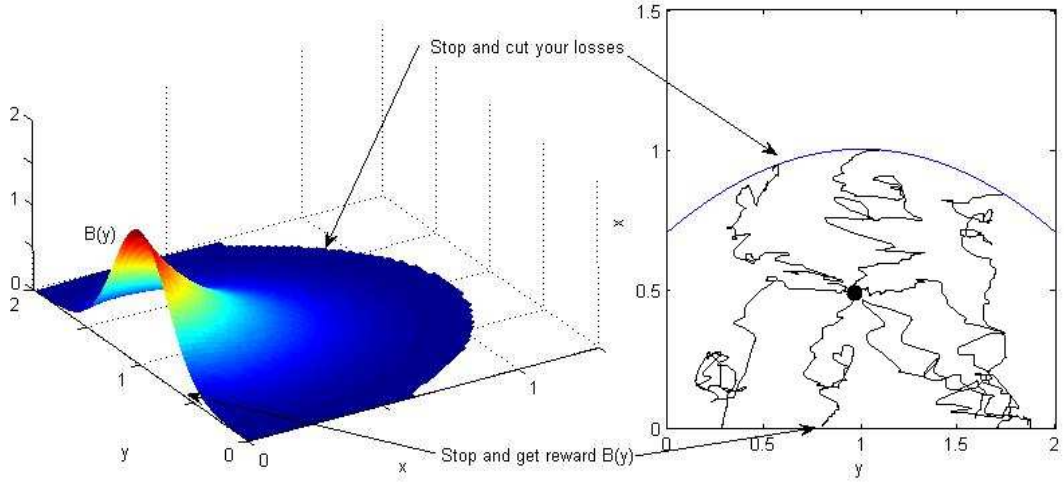
where $B(Y_t) \geq 0$ determines the reward if X_t hits zero before the chosen stopping time, $g(X_t, Y_t) > 0$ measures the local cost per unit time before stopping, and the supremum is taken over all stopping times of the Brownian filtration. This problem strikes a balance between the reward associated with waiting until X hits zero and the costs borne for letting the clock run.

One way to think of this problem is to write it recursively as a dynamic programming equation. A discrete-time, recursive formulation of this equation reads

$$\begin{aligned} v(X_t,Y_t) &= \max\Big(0, E\left[v(X_{t+\Delta t}, Y_{t+\Delta t}) \mid X_t, Y_t\right] - g(X_t,Y_t)\Delta t\Big), \\ v(0,y) &= B(y). \end{aligned}$$

If the Brownian motion is replaced with a symmetric random walk on a uniform lattice with

Figure 4.1: The left panel shows the value function for the optimal stopping problem with constant g. The right panel shows sample paths of the Brownian motion over $\Omega(\alpha)$ that starting from a particular point. If X_t hits zero before you cut your losses, you win a prize $B(Y_t) > 0$.



spacing h and time steps $\Delta t = \frac{1}{2}h^2$, the equation becomes

$$v(x,y) = \max\Big(0, \frac{v(x+h,y) + v(x-h,y) + v(x,y+h) + v(x,y-h)}{4} - g(x,y)\Delta t\Big), \quad (4.1)$$
$$v(0,y) = B(y)$$

where v is defined on lattice points $(x,y) = (ih, jh)$. The modern characterization of the value function is obtained as the limit as $h \to 0$ of this problem. Formally, v is the viscosity solution to the following complementary inequalities:[1]

$$\begin{aligned} \frac{1}{2}\triangle v - g &\leq 0, \\ v &\geq 0, \\ v(\frac{1}{2}\triangle v - g) &= 0, \\ v(0,y) &= B(y). \end{aligned}$$

Equation (4.1) can be rearranged to fit a discrete version of this form. As shown in Figure

[1]Details on the theory of these problems can be found in [25, 33, 35, 50].

4.1, the interface defined by these inequalities partitions the state space into a *continuation region*, where the first inequality holds with equality, and a *stopping region*, where the second inequality binds. On the interface where both inequalities are tight, the incentive to stop and continue are balanced. This is called the *stopping boundary*, and the optimal stopping time is defined as the first time state process hits this boundary.

4.1.1 A Classical Formulation

In the dynamic programming formulation of the optimal stopping problem, the stopping boundary is defined implicitly as an artifact of the value function. In contrast, the classical formulation controls the location of stopping boundary explicitly. If a function $\alpha(y)$ indexes the stopping boundary, the continuation region is defined by

$$\Omega(\alpha) = \{(x, y) : y \in \mathbb{R},\ 0 \leq x \leq \alpha(y)\}.$$

The value function is defined only over the domain $\Omega(\alpha)$, rather than over all of the half-space as with the dynamic programming formulation. The problem is to choose a smooth stopping boundary $\alpha(y)$ and a function v over $\Omega(\alpha)$ that satisfy equations

$$\frac{1}{2}\triangle v = g, \tag{4.2a}$$

$$v(0, y) = B(y), \tag{4.2b}$$

$$v(\alpha(y), y) = 0, \tag{4.2c}$$

$$\nabla v(\alpha(y), y) \cdot \mathbf{n} = 0 \tag{4.2d}$$

where $\mathbf{n}$ denotes the unit outward normal vector. On the stopping boundary, two conditions must hold: one is associated with the value function v and the other with the stopping boundary α. In this way, equations (4.2a)-(4.2d) take the form of a calibration problem. Condition (4.2c), called the *value-matching* condition, follows from the definition of the value function. It simply says that the reward is zero at the stopping time. Equation (4.2d), called the *smooth-pasting* condition, says the location of the free boundary is locally optimal, meaning small perturbations to the boundary have no first order effects on the value function.

Remark 4. *Our representation of the stopping boundary certainly restricts the set of stopping times under consideration. In general, the optimal stopping boundary can not be represented as a function $\alpha(y)$. For certain choices of cost function g, the stopping region may not even be a connected set. In the case that $g > 0$ is constant and $B(y) > 0$, however, the stopping*

boundary can be represented as a function of y.

Even if we were considering a problem with a more complicated stopping boundary, the topology of the optimal continuation region and its basic shape can be determined in a low-resolution discretization. Once the basic structure of the stopping boundary is established, an appropriate parameterization can be chosen, and the methods developed here apply to resolve the stopping boundary at higher accuracy.

4.2 Linearizing the Problem

In practice, it is impractical to perturb the location of a boundary in the process of solving for the value function. To avoid this difficulty, we linearize the problem, working instead over a static domain so that α affects the boundary conditions defining the value function rather than the location of the actual boundary. The resulting linear problem is sometimes called the *small disturbance approximation*.

4.2.1 A Special Case

If B and g are constant functions, then the value function and the location of the free boundary do not depend on y, and the equations (4.2) simplify to

$$\frac{1}{2}v_{xx} = g \tag{4.3a}$$

$$v(0) = B \tag{4.3b}$$

$$v(\alpha) = 0 \tag{4.3c}$$

$$v_x(\alpha) = 0 \tag{4.3d}$$

If v and α solve these equations, then, for any perturbations $\alpha \leftarrow \alpha + \widetilde{\alpha}$ and $v \leftarrow v + \widetilde{v}$, the linearized equations (4.4) have only the trivial solution.

$$\frac{1}{2}\widetilde{v}_{xx} = 0 \tag{4.4a}$$

$$\widetilde{v}(0) = 0 \tag{4.4b}$$

$$\widetilde{v}(\alpha) = 0 \tag{4.4c}$$

$$\widetilde{v}_x(\alpha) = 2g\widetilde{\alpha} \tag{4.4d}$$

Suppose we regard v as being defined implicitly in terms of α as the solution to equations (4.3a,4.3b,4.3c). The problem is then to chose a scalar α to enforce the smooth pasting condition (4.3d). Viewed this way, a perturbation $\widetilde{\alpha}$ induces a variation

$$v \leftarrow v + \widetilde{v} + O(\widetilde{\alpha}^2)$$

where $\widetilde{v}$ satisfies equations (4.4a,4.4b,4.4c). Since these equations admit only the trivial solution, changes in α have no first-order effects on v. In other words, the solution $\widetilde{v}$ does not tell us how to correct the perturbation $\widetilde{\alpha}$.

If instead, v is regarded as an implicit function of α defined by equations (4.3a,4.3b,4.3d), then α should be chosen to solve the value matching condition (4.3c). In this formulation, the perturbation $\widetilde{\alpha}$ induces a first order variation $\widetilde{v}$ that satisfies equations (4.4a,4.4b,4.4d). The solution is $\widetilde{v}(x) = 2g\widetilde{\alpha}x$.

This perturbation affects the residual of the value matching condition (up to first order) by an amount $2g\widetilde{\alpha}\alpha$. We can use this residual to correct for the perturbation $\widetilde{\alpha}$ and drive the system back toward the solution. To do this, we look for $\widetilde{\widetilde{\alpha}} = -\widetilde{\alpha} + O(\|\widetilde{\alpha}\|^2)$ by choosing $\widetilde{\widetilde{\alpha}}$ such that

$$\widetilde{\widetilde{v}}(\alpha) = -\widetilde{v}(\alpha)$$

where $\widetilde{\widetilde{v}}$ solves equations (4.4a,4.4b,4.4d) with $\widetilde{\alpha}$ replaced by $\widetilde{\widetilde{\alpha}}$. Based on this strategy, Newton's method applied to the mapping from α to the value matching condition is defined as follows.

1. Given α_n, solve equations (4.3a,4.3b,4.3d) for the value function $v(x;\alpha)$.
2. Solve the small disturbance approximation, equations (4.4a,4.4b,4.4d), for $\widetilde{v}$ and $\widetilde{\alpha}$.
3. Update $\alpha_{n+1} \leftarrow \alpha_n + \widetilde{\alpha}$ and go to step 1.

Since the equations can be solved in closed form, the iterates can be computed explicitly

$$\alpha_{n+1} = \frac{B + g\alpha_n^2}{2g\alpha_n}.$$

This iteration scheme converges quadratically from a well-chosen starting point.

If numerically solving v and $\widetilde{v}$ in terms of α and $\widetilde{\alpha}$ can be done efficiently, which it can in the one-dimensional case, then this method is a viable computational strategy for locating the stopping boundary.

4.2.2 The General Case

We formulate the two-dimensional problem in an analogous fashion. Given a function $\alpha(y)$ defining the stopping boundary, let the function $v(x, y; \alpha)$ be the solution to equations (4.2a,4.2b,4.2d). We then seek a function $\alpha(y)$ to enforce the value matching condition (4.2c). Starting from a given $\alpha(y)$ a perturbation $\alpha \leftarrow \alpha + \widetilde{\alpha}$ induces a variation

$$v(x, y; \alpha + \widetilde{\alpha}) = v(x, y; \alpha) + \widetilde{v}(x, y) + O(\|\widetilde{\alpha}\|^2)$$

where $\widetilde{v}$ satisfies the equation[2]

$$\frac{1}{2}\triangle\widetilde{v} = 0 \tag{4.5a}$$

$$\widetilde{v}(0, y) = 0 \tag{4.5b}$$

$$\nabla\widetilde{v}(\alpha(y), y) \cdot \mathbf{n} = -\widetilde{\alpha}(y)[v_{xx}, v_{yx}] \cdot \mathbf{n} - \nabla v \cdot \widetilde{\mathbf{n}}. \tag{4.5c}$$

The correction $\widetilde{\alpha}$ should be chosen to enforce the linearized value matching condition

$$\widetilde{v}(\alpha(y), y) = -v(\alpha(y), y; \alpha). \tag{4.5d}$$

Newton's method can be applied to generate a sequence of stopping boundaries:

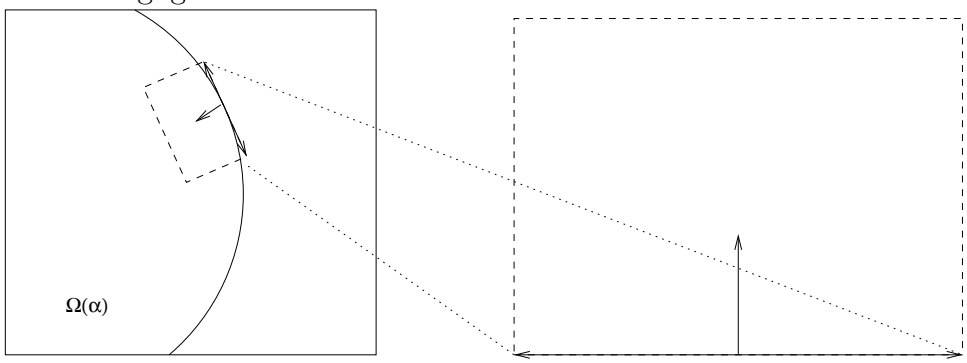
1. Given α, solve equations (4.2a,4.2b,4.2d) for the value function $v(x, y; \alpha)$ on $\Omega(\alpha)$.
2. Solve the small disturbance approximation, equations (4.5a,4.5b,4.5c,4.5d), for $\widetilde{v}$ and $\widetilde{\alpha}$ on $\Omega(\alpha)$.
3. Update $\alpha \leftarrow \alpha + \widetilde{\alpha}$ and go to step 1.

To implement this algorithm, we need a method to solve for both the value function and the associated small disturbance approximation. Solving for the value function is a classical problem in numerical analysis that is well understood. Solving $\widetilde{v}$ for a given $\widetilde{\alpha}$ is similarly

[2]There are several ways to compute the change in the outward unit normal $\widetilde{\mathbf{n}} \propto (1, -\alpha_y(y))$. For numerical purposes, we use first order approximation that moves in the unit tangent direction $\mathbf{t} \propto (\alpha_y(y), 1)$ until it hits the ray generated by the displaced unit normal vector $\mathbf{m} \propto (1, -\alpha_y(y) - \widetilde{\alpha}(y))$. The angle between these vectors is $\theta = \cos^{-1}(\langle \mathbf{n}, \mathbf{m} \rangle)$, and so we use the approximation

$$\widetilde{\mathbf{n}} = \tan(\theta)\mathbf{t}.$$

Figure 4.2: For high frequencies, the effects of far boundary conditions and smooth variation in coefficients are negligible.



straight forward. Solving for $\widetilde{v}$ and $\widetilde{\alpha}$ jointly is much more difficult. We now regard $\widetilde{v}$ as an implicit function of $\widetilde{\alpha}$ and focus on solving for the $\widetilde{\alpha}$ that satisfies the linearized value matching condition. Understanding the mapping from $\widetilde{\alpha}$ to equation (4.5d) is fundamental to the numerical analysis of the problem.

4.2.3 Half-space analysis

Assume for now that B and g are constant functions so that the optimal boundary $\alpha(y)$ is constant. A perturbation of the form $\widetilde{\alpha} = e^{\iota k y}$ induces a perturbation $\widetilde{v}$ that solves (4.5a)–(4.5c), which can be computed explicitly. On the stopping boundary, the perturbation corresponding to the value matching condition is

$$\widetilde{v}(\alpha, y) = \underbrace{\frac{2g}{|k|}\big(1 - \frac{e^{\alpha}}{\cosh(k)}\big)}_{\widehat{F}(k)} e^{\iota k y}. \tag{4.6}$$

This exposes a problem with the formulation: Since high-frequency perturbations to α vanish in the value matching condition, the mapping F from $\widetilde{\alpha}$ to $\widetilde{v}(\alpha, y)$ in (4.5d) is not h-elliptic. For numerical purposes, this formulation of the problem is badly conditioned. According to the theory of variational inequalities, see [33] for example, the optimal $\alpha(y)$ is known *a priori* to be at least as smooth as $B(y)$, which is smooth and has essentially no high-frequency components. The value matching condition has trouble differentiating between boundaries that differ only by high frequencies. This suggests that some kind of regularization is necessary in order to choose the correct α from all those that are nearly optimal. There are many ways to do this. One possibility is to consider only low-frequency α that are appropriately smooth. Another is to consider all perturbations but introduce some tie-breaking penalty

for oscillatory perturbations. Both approaches can introduce distortions into the solution. We proceed along different lines.

The value matching condition (4.5d) can be reformulated as

$$\underbrace{\frac{d^2}{dy^2}\big(v(\alpha, y; \alpha) + \widetilde{v}(\alpha, y)\big)}_{G(\widetilde{\alpha})} = 0 \tag{4.7}$$
$$\lim_{y \to \pm\infty} v(\alpha, y; \alpha) + \widetilde{v}(\alpha, y) = 0.$$

It follows that

$$\frac{d^2}{dy^2}\widetilde{v}(\alpha, y) = \underbrace{2g\,|k|\,\big(1 - \frac{e^{\alpha}}{\cosh(k)}\big)}_{\widehat{G}(k)}\, e^{\iota k y}, \tag{4.8}$$

which is characteristic of an h-elliptic operator as defined in Section 2.4 since $\widetilde{G}(k) \approx 2g\,|k|$ for large $|k|$. By taking the second derivative, we have effectively preconditioned the value matching equation in a way that the operator becomes less rather than more singular when considering eigenfunctions of increasingly high frequency. The h-ellipticity of this preconditioned operator is key to designing a multigrid solver.

If B and g are not constant, the preceding analysis fails to hold exactly. It does hold approximately for the high-frequency perturbations responsible for the poor conditioning of the operator $F(\widetilde{\alpha})$ corresponding to the value matching condition.

A small neighborhood around a point $(\alpha(y), y)$ on the stopping boundary induces a new coordinate system spanning a half-space as shown in Figure 4.2. In this local coordinate system, we solve the appropriately transformed version of equations (4.5a,4.5c) on the half-space, ignoring the effects of the far boundary condition (4.5b). For $\widetilde{\alpha}$ of high-frequency, this is justified since these perturbations decay before they interact with the far boundary.[3] Since the Laplacian is invariant to coordinate rotations, the local symbol of the value matching condition derived from the half-space analysis is

$$\widehat{F}(k) \approx \frac{2g(\alpha(y), y)}{|k|}.$$

[3] A detailed analysis of the interaction of elliptic operators and boundary conditions can be found in [1,2].

This is consistent with the symbol (4.6) for large k. This operator can be preconditioned by

$$G(\alpha) = \frac{d^2}{dy^2}\widetilde{v}(\alpha(y), y).$$

to give a symbol $\widehat{G}(k) \approx 2g(\alpha(y), y)|k|$, which approximates the exact symbol (4.8). Since this symbol corresponds to an h-elliptic operator, it can be used to design an effective smoother.[4]

4.3 Discretization

As formulated, the value function is defined over a domain unbounded in the y direction. To solve the problem numerically, a finite domain must be used. A simple approach is to truncate the domain so that y varies in some interval $[0, y^{\max}]$ and impose the von Neumann conditions (4.9c) on the the artificial boundaries. These boundary conditions can be interpreted as replacing the Brownian motion Y by a process that reflects off the artificial boundaries in the inward normal direction.

Given a boundary defined by $\alpha(y)$, the value function $v(x, y; \alpha)$ is defined for $y \in [0, y^{\max}]$ and $x \in [0, \alpha(y)]$ by

$$\frac{1}{2}\triangle v - g = 0, \tag{4.9a}$$

$$v(0, y) = B(y) \tag{4.9b}$$

$$v_y(x, 0) = 0, \quad v_y(x, y^{\max}) = 0 \tag{4.9c}$$

$$\nabla v(\alpha(y), y) \cdot \mathbf{n} = 0. \tag{4.9d}$$

The corresponding small disturbance approximation is

$$\frac{1}{2}\triangle \widetilde{v} = 0 \tag{4.10a}$$

$$\widetilde{v}(0, y) = 0 \tag{4.10b}$$

$$\widetilde{v}_y(x, 0) = 0 \;\; \widetilde{v}_y(x, y^{\max}) = 0 \tag{4.10c}$$

$$\nabla\widetilde{v}(\alpha(y), y) \cdot \mathbf{n} = -[v_{xx}\widetilde{\alpha}, v_{yx}\widetilde{\alpha}] \cdot \mathbf{n} - \nabla v \cdot \widetilde{\mathbf{n}}, \tag{4.10d}$$

[4]If the boundary α has significant curvature it may improve conditioning to re-parameterize α and differentiate with respect to the arc-length of α rather than with respect to y.

where $\widetilde{\alpha}$ is chosen to enforce the linearized value matching condition

$$\widetilde{v}(\alpha(y), y) = -v(\alpha(y), y). \tag{4.10e}$$

4.3.1 Grid Generation

The domain $\Omega(\alpha)$ depends on the function α, which is usually not constant. Choosing a discrete problem to properly reflect this has been the subject of much research in both industry and the academy. In cases where α is constant, the domain is rectangular and it is possible to use a standard finite-difference discretization over a uniformly spaced grid. When α is almost constant, we follow [57, Section 5.7] and use a simple, boundary-fitted grid. The idea is to map the unit square into $\Omega(\alpha)$ and pass the square grid through this transformation to recover the boundary-fitted grid.

Suppose there are $N_x + 1$ grid points on the x axis and $N_y + 1$ points on the y axis. The spacings $h_x = 1/N_x$ and $h_y = 1/N_y$ then define a uniform grid over the unit square. When there is no ambiguity we use h to denote the pair (h_x, h_y). Let $\alpha^h > 0$ be a grid function over the domain $\Omega_{h_y} = \{y_j = y^{\max}(j/N_y) \ : \ j = 0, 1, \ldots, N_y\}$. Then, the boundary-fitted grid is defined by

$$\Omega^h(\alpha^h) = \{(x_{(i,j)}, y_j) = (\alpha^h_j i h_x, j h_y) \ : \ i(j) = 0, 1, \ldots, N_x(N_y)\}.$$

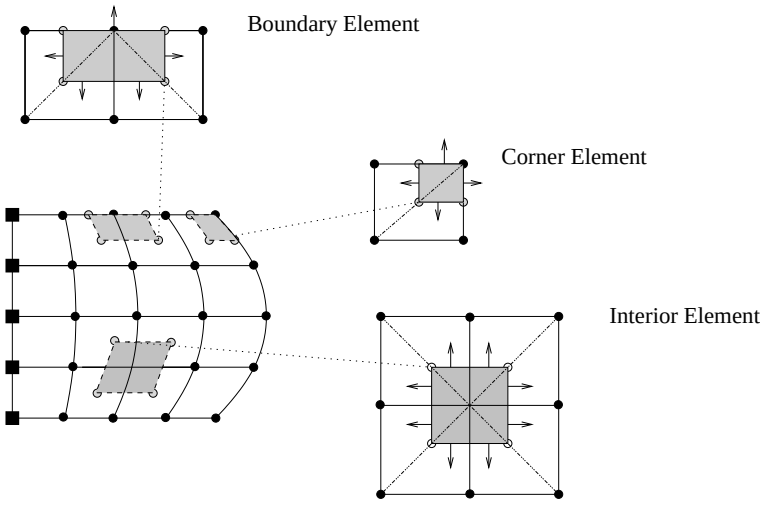
Functions over this grid are defined by their values at grid points. Between grid points, they are defined by bilinear interpolation of the values at neighboring grid points.

4.3.2 A Finite Volume Scheme

As shown in Figure 4.3, the discrete domain is partitioned into interior and boundary elements. Each interior element has four corners. There is much freedom in how to define the location of these corners, and a poor choice leads to a low-accuracy scheme. Formulating high-accuracy discretizations over boundary fitted grids is a challenging task. A detailed analysis of a robust discretization scheme appropriate for Poisson-type and other equations over two-dimensional domains can be found in [48].

This chapter focuses on methods for solving discrete equations associated with a given discretization schemes, not on how to choose an accurate scheme. We therefore use a simple

Figure 4.3: Finite volume discretization on a boundary fitted grid. Shaded regions represent generic elements. Black dots denote grid points. Small circles represent element corners. Small squares locate the grid points defined by Dirichlet conditions. Triangles inside the elements, along with the values of associated grid points, are used to define a gradient on each triangle and outward normal derivatives on the boundary of each element.



yet potentially low-accuracy scheme to locate the corners.

$$c_{i,j} = \frac{v_{i,j} + v_{i+1,j} + v_{i,j+1} + v_{i+1,j+1}}{4}$$
$$i = 0, 1, \ldots, N_x - 1; \quad j = 0, 1, \ldots, N_y - 1.$$

Wherever they are located, these points define elements $E_{i,j}$ that partition the space. The discrete equations associated with the value function are

$$\int_{E_{i,j}} (\frac{1}{2}\triangle v - g) = f_{i,j}, \tag{4.11}$$

and those associated with the small disturbance approximation are

$$\int_{E_{i,j}} \frac{1}{2}\triangle \widetilde{v} = \widetilde{f}_{i,j}.$$

These equations must hold for each element $E_{i,j}$, which is indexed by $i = 1, \ldots, N_x$ and $j = 0, \ldots, N_y$. The arbitrary right side vectors f and $\widetilde{f}$ have been added since this form is prescribed by the FAS coarse-grid problem. They are zero for a problem defined on the

finest grid.

To implement this discretization we use the integration-by-parts formula

$$\int_E \triangle v = \int_{\partial E} \nabla v \cdot \mathbf{n} \tag{4.12}$$

where $\mathbf{n}$ is the unit outward normal vector on the element E. As shown in Figure 4.3, each element is naturally partitioned into triangles. If we assume the function over each triangle is affine, then the three grid points associated with each triangle define a gradient. Residuals in each element can then be computed by applying equation (4.12) and summing over the triangles in that element.

Some edges on the boundary elements are not naturally associated with a triangle, and a gradient cannot be defined as it can for interior elements. For these edges, the von-Neumann boundary conditions (4.9c,4.9d,4.10c,4.10d) define the outward normal derivatives. Elements on the Dirichlet boundary take the values prescribed by equations (4.9b,4.10b) at the associated grid point.

Finally, the linearized value matching condition is given by

$$\widetilde{v}(x_{(N_x,j)}, y_j) = -v(x_{(N_x,j)}, y_j), \quad j = 0, \ldots, N_y.$$

In the case where the grid is regular, this finite-volume scheme reduces to a standard finite-difference[5] scheme based on the five point discrete Laplacian in Section 2.4.1. In this case implementation is considerably simpler, and results in a second-order accurate scheme if the corners of the domain are handled carefully.

4.4 Multigrid Treatment

We now focus on designing a V-cycle scheme to solve for $\widetilde{v}$ and $\widetilde{\alpha}$ in the small disturbance approximation. Assume the value function v in equation (4.11) is solved for exactly in terms of a given boundary α, which may not be optimal.

[5]The finite-volume equations are scaled the area of each element. This scaling must be reflected in a multigrid treatment when residuals are restricted.

4.4.1 Multigrid Components

The Smoother

The most important and distinctive component of this multigrid scheme is the smoother. For a given $\widetilde{\alpha}$, smoothing errors in $\widetilde{v}$ is straightforward. There are many effective point-wise smoothers compatible with this problem. We choose a standard Gauss-Seidel iteration (described in Section 2.3.2) that sweeps through the grid points in lexicographic order: For each $j = 0, 1, \ldots, N_y$ it visits points $i = 1, \ldots, N_x$, updating $\widetilde{v}_{i,j}$ at each point. We denote M_v sweeps of this algorithm by

$$\widetilde{v} \leftarrow \text{GS-LEX}(\widetilde{v}, \widetilde{\alpha}, M_v).$$

It is not tractable to implement such a point-wise smoothing scheme for $\widetilde{\alpha}$ since it is prohibitively expensive to invert the value matching condition one component at a time. As described in Section 2.4, a smoother based on the residuals of the value matching condition can be constructed. Equation (4.6), shows that the mapping between $\widetilde{\alpha}$ and the linearized value-matching condition (4.10e) is not h-elliptic and therefore does not define a smoothing correction. Equation (4.8) prescribes an equivalent value matching condition and provides an h-elliptic operator that does define a smoothing correction. The symbol of this operator at the continuous level is

$$\widehat{G}(k) \approx 2g\,|k|$$

for large $|k|$. It follows that for small h, the discrete symbol is

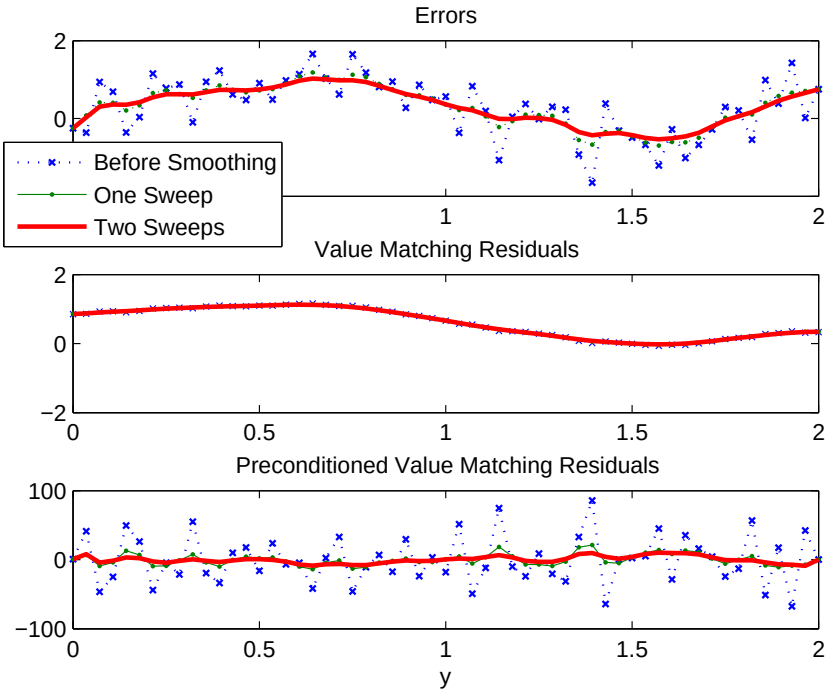
$$\widehat{G}^{h_y}(\theta) \approx \widehat{G}(\theta/h_y).$$

If h_y is too large for this approximation to hold, it is possible to derive the actual discrete symbol or at least a better approximation. For this problem, the approximation suffices.

The residuals of the preconditioned value matching condition defines a search direction, and the symbol gives the magnitude of the step size. The highest frequencies that can be represented on the grid are $|\pi|/h_y$. Therefore, to smooth the highest frequency errors for the equation

$$\frac{d^2}{dy^2}\widetilde{v}(\alpha, y) = -\frac{d^2}{dy^2}v(\alpha, y)$$

Figure 4.4: The effects of smoothing sweeps using $M_v = 3$ and $M_\alpha = 1$. The smoother is both inexpensive and effective. This plot also shows that the value matching residuals do not respond to high-frequency errors but the preconditioned residuals are sensitive.



a smoothing correction is defined by $\widetilde{\alpha} \leftarrow \widetilde{\alpha} + \widetilde{\widetilde{\alpha}}$ where

$$\widetilde{\widetilde{\alpha}} = \xi \frac{\frac{d}{dy^2}\big(v(\alpha, y) - \widetilde{v}(\alpha, y)\big)}{\widehat{G}^{h_y}(\pi)} \tag{4.13}$$

and the step size is $\xi \in (0, 1]$. Choosing ξ must be done experimentally. While any choice defines a smoother, a choice of $\xi = 1$ proves most effective for this problem.

At the discrete level, we replace the second derivative operator with finite difference quotients. The resulting smoother performs well for y safely inside the interval $(0, y^{\max})$, but it may not interact well with the artificial boundary conditions. Indeed, at the corners of the domain where two Neumann conditions hold simultaneously, there is a weak singularity that becomes problematic on fine grids. To overcome this, the smoother does not change the value of the boundary points α_0, α_{N_y} on the fine grid (only on the coarse grid), and the neighboring points α_1, α_{N_y-1} are defined by linear interpolation with their neighbors rather than by the preconditioned value matching residuals.

The smoothing step in equation (4.13) assumes that $\widetilde{v}$ has been solved exactly for a given

$\widetilde{\alpha}$. It is not necessary to capture all of the variation in $\widetilde{v}$ induced by $\widetilde{\alpha}$. For smoothing, only high frequencies are relevant, and these can be resolved with only a few Gauss-Seidel relaxation sweeps of $\widetilde{v}$. Furthermore, since high-frequency variations in boundary data decay quickly into the interior of the domain, it is only necessary to relax points near the stopping boundary. These considerations can drastically affect the efficiency of the smoother. While the general structure of the smoother was done through a formal analysis, striking an efficient balance between good smoothing factors and total work done in a smoothing sweep must be done experimentally.

Figure 4.4 shows how the following smoother acts on a random configuration with both high and low-frequency components:

* **Algorithm:** $(\widetilde{v}, \widetilde{\alpha}) \leftarrow \text{Smooth}(\widetilde{v}, \widetilde{\alpha}, M_\alpha, M_v)$

1. Repeat M_α times:
   (a) Smooth $\widetilde{v} \leftarrow \text{GS-LEX}(\widetilde{v}, \widetilde{\alpha}, M_v)$.
   (b) Update $\widetilde{\alpha}$ according to (4.13).
2. Smooth $\widetilde{v} \leftarrow \text{GS-LEX}(\widetilde{v}, \widetilde{\alpha}, M_v)$.

Grid Transfer Operators

Transferring grid functions $\widetilde{v}$ and $\widetilde{\alpha}$ is straightforward: Simple injection is used for the restriction operator and bilinear interpolation for prolongation.

With von Neumann boundary conditions, restricting the residuals is more subtle. In [57, Section 5.6.2] it is shown that using injection near von Neumann boundaries can result in improper scaling in the coarse grid problem. As we saw in Figure 2.3.6, even if the scaling problems are fixed, the convergence rate of a multigrid cycle is improved substantially when a higher-order restriction operator is used. A conventional choice for restricting residuals for Poisson-type equations is the restriction operator I_h^{2h} that satisfies

$$\int_{E^{2h}} w^h = \int_{E^{2h}} I_h^{2h} w^h \tag{4.14}$$

for each fine grid function w^h and each coarse grid element E^{2h}. The left hand integral in this formula is computed through the trapezoid rule and the right integral by the midpoint rule. When the grid is uniform and rectangular, this definition produces the full-weighting

operator, written in stencil form as

$$I_h^{2h} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Similarly, stencils for restricting residuals boundary and corner elements are

$$\frac{1}{16}\begin{bmatrix} 2 & 4 & 2 \\ 2 & 4 & 2 \end{bmatrix}, \quad \frac{1}{16}\begin{bmatrix} 2 & 2 \\ 4 & 4 \\ 2 & 2 \end{bmatrix}, \text{ and } \frac{1}{16}\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}.$$

For the linearized value matching condition, residuals are restricted by the one-dimensional full-weighting operator, given in stencil form by $[1, 2, 1]/4$. End points are simply injected.[6]

The Coarse Grid Problem

Suppose the problem on a given grid $\Omega_h(\alpha^h)$ is to find grid functions $\widetilde{v}^h$ and $\widetilde{\alpha}^h$ that satisfy the equations

$$\begin{aligned} \left(L^h(\widetilde{v}^h, \widetilde{\alpha}^h)\right)_{i,j} &= \widetilde{f}^h_{i,j} \\ \widetilde{v}^h_{(N_x^h, j)} &= -v^h_{(N_x^h, j)} \end{aligned}$$

where the operator L^h is defined as

$$\left(L^h(\widetilde{v}^h, \widetilde{\alpha}^h)\right)_{i,j} = \int_{E^h_{i,j}} \frac{1}{2} \triangle \widetilde{v}^h$$

[6]Because residuals are injected at the boundaries $j = 0, N_y$ the total contribution from the residual at neighboring points $j = 1, N_y - 1$ is one half rather than one. This practice loses some information in the coarsening process since it is inconsistent with equation (4.14). A potentially better choice for restriction operators at the boundaries is defined by the stencils $[2, 2, 1]/4$ and $[1, 2, 2]/4$. For the problem at hand, this detail does not seem to improve performance of the V-Cycle. However, the artifacts of residual transfer near the boundaries on the coarsest grid can be seen in the convergence plots in Figure 4.6(b).

with the boundary conditions (4.10b,4.10c,4.10d) appropriately incorporated into the operator. Then, the coarse grid problem is to find grid functions $\widetilde{v}^{2h}$ and $\widetilde{\alpha}^{2h}$ that satisfy

$$\begin{aligned}\left(L^{2h}(\widetilde{v}^{2h},\widetilde{\alpha}^{2h})\right)_{i,j} &= \widetilde{f}^{2h}_{i,j}\\ \widetilde{v}^{2h}_{(N^h_x,j)} &= -v^{2h}_{(N^{2h}_x,j)}\end{aligned}$$

where the right side vector is defined by the Full Approximation Scheme

$$\widetilde{f}^{2h}_{i,j} = \frac{\left|E^{2h}_{i,j}\right|}{\left|E^{h}_{(i,j)'}\right|} I^{2h}_h\left(\widetilde{f}^h - L^h(\widetilde{v}^h,\widetilde{\alpha}^h)\right)_{i,j} + \left(L^{2h}(I^{2h}_h\widetilde{v}^h, I^{2h}_h\widetilde{\alpha}^h)\right)_{i,j}. \tag{4.15}$$

The subscripts $(i,j)'$ denote the fine grid points corresponding to the index (i,j) on the coarse grid. Scaling the residuals in this manner is necessary because the residuals represent integrals of grid functions rather than grid functions themselves. Note that the value matching condition does not require a right side correction.

Exact Solver on Coarsest Grid

To close the V-cycle iteration, we require a robust, exact solver for the small disturbance approximation on the coarsest grid. We write

$$(\widetilde{v}^h,\widetilde{\alpha}^h) \leftarrow \text{ExactSolve}(\widetilde{f}^h).$$

On the coarsest grid, it is desirable to use only three to five grid points per dimension so that $\widetilde{v}^h$ can be efficiently solved for a given of $\widetilde{\alpha}^h$ using dense matrix techniques available in any linear algebra package. This solver establishes a bijective map between $\widetilde{\alpha}^h$ and the residuals of the value matching condition. To find the zero of this map, a direct solver[7] that requires only evaluations of the value matching residuals can be used.

4.4.2 The V-Cycle

These multigrid components can now be assembled into a standard V-cycle algorithm.

[7] See Remark 2 in the previous chapter.

Table 4.1: Estimated error reduction factors for V-Cycles with various smoother configurations and depths. Textbook error reductions for V-cycles are in the range of 8–12.

M_α	M_v	ξ	Level 2	3	4	6	9
1	1	1	8.44	7.78	7.54	6.64	6.76
1	2	1	8.75	8.19	8.04	7.99	8.12
1	∞	1	9.19	8.46	8.61	8.71	n/a
2	1	1	13.37	11.50	10.55	9.77	9.12
2	2	1	13.07	12.46	12.21	11.71	11.33
2	∞	1	13.67	13.28	13.27	12.95	n/a
3	1	1	17.09	16.77	15.04	13.69	12.79
1	1	2/3	4.69	4.04	4.09	4.29	4.17
1	3	2/3	5.13	3.98	4.06	4.06	4.05
2	1	2/3	9.32	9.59	8.97	8.46	8.58
2	2	2/3	10.01	9.74	9.66	9.53	9.38
2	4	2/3	10.60	9.88	9.98	9.89	9.84
3	2	2/3	12.93	12.65	12.63	12.21	11.91

* **Algorithm:** $(\widetilde{v}^h, \widetilde{\alpha}^h) \leftarrow \text{V-Cycle}(L, \widetilde{v}^h, \widetilde{\alpha}^h, \widetilde{f}^h, M_\alpha, M_v)$

1. If $L = 0$ set $(\widetilde{v}^h, \widetilde{\alpha}^h) \leftarrow \text{ExactSolve}(\widetilde{f}^h)$ and return.
2. Otherwise, set $(\widetilde{v}^h, \widetilde{\alpha}^h) \leftarrow \text{Smooth}(\widetilde{v}^h, \widetilde{\alpha}^h, M_\alpha, M_v)$.
3. Compute the coarse-grid right side $\widetilde{f}^{2h}$ according to equation (4.15).
4. Solve the coarse-grid problem

$$(\widetilde{v}^{2h}, \widetilde{\alpha}^{2h}) \leftarrow \text{V-Cycle}(L-1, \widetilde{v}^{2h}, \widetilde{\alpha}^{2h}, \widetilde{f}^{2h}, M_\alpha, M_v)$$

5. Apply the update $(\widetilde{v}^h, \widetilde{\alpha}^h) \leftarrow (\widetilde{v}^h, \widetilde{\alpha}^h) + I_{2h}^h\Big((\widetilde{v}^{2h}, \widetilde{\alpha}^{2h}) + I_h^{2h}(\widetilde{v}^{2h}, \widetilde{\alpha}^{2h})\Big)$
6. Apply post-smoothing $(\widetilde{v}^h, \widetilde{\alpha}^h) \leftarrow \text{Smooth}(\widetilde{v}^h, \widetilde{\alpha}^h, M_\alpha, M_v)$ and return.

Performance of V-cycles

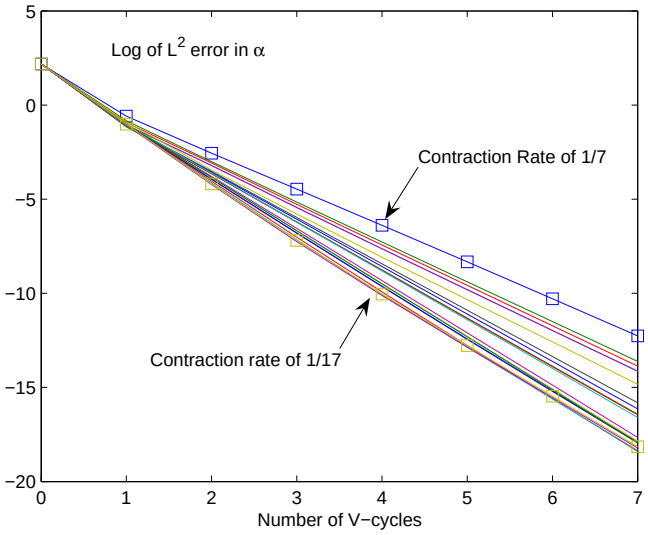
For a first set of numerical experiments, we work with constant B and g and linearize around the exact solution (v^h, α^h) to the problem on the grid $\Omega_h(\alpha^h)$. This choice implies a rectangular domain and simplifies implementation issues considerably. To explore the performance of the V-Cycle, we start with an initial perturbation $\widetilde{\alpha}$ and track the convergence

back to $\widetilde{\alpha} = 0$.

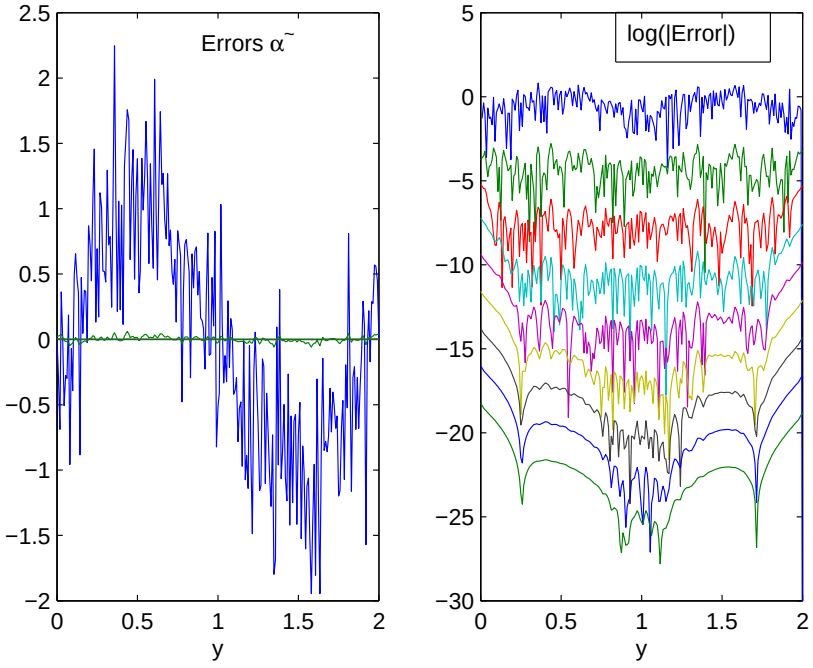
Table 4.1 and Figures 4.6(a) and 4.6(b) show the performance of the V-Cycle starting from a random configuration with both high and low frequencies. Taking a full-size smoothing step ($\xi = 1$) results in a textbook error reduction factor for all but the cheapest smoothing scheme where $(M_\alpha, M_v) = (1, 1)$. The step size for this smoother is particularly well chosen since it exploits the rectangular shape of the domain. For more general boundary configurations, we might not expect the smoother to perform this well. To test this, we chose a step size of $\xi = 2/3$. The rates are more modest and, possibly, more realistic. In this case, a more expensive smoother with $M_\alpha = 2$ bolsters the method enough to give textbook multigrid performance.

Figure 4.6 shows a possible FMG algorithm for solving the stopping problem to a given level of accuracy.

Figure 4.5: Convergence of V-Cycles.

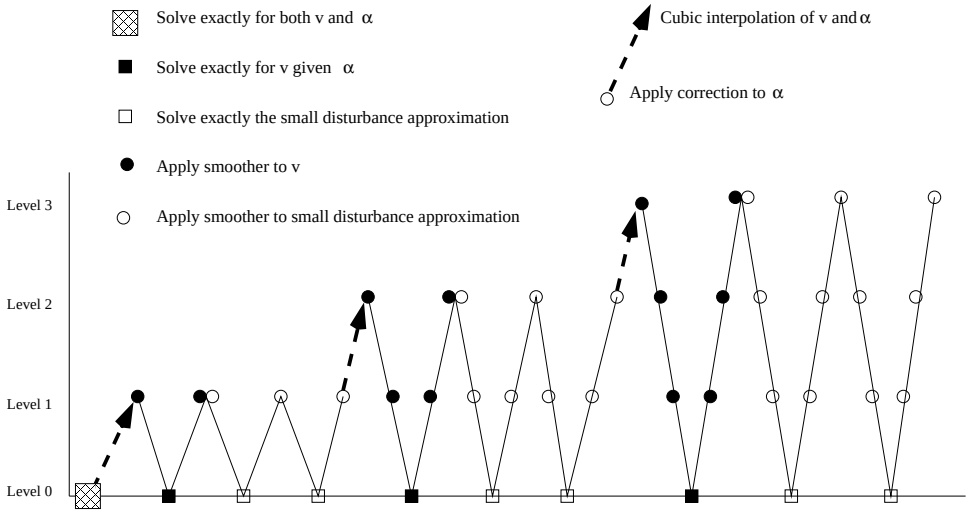


(a) Convergence plots for V-cycles with six levels and various smoother configurations. In all cases the convergence rate settles in immediately after the first V-cycle.



(b) Shown is the convergence of $\widetilde{\alpha}$ starting from a random configuration. Several V-Cycles were run with $L = 5$ levels. Relaxation on each level consists of two iterations of smoothing $\widetilde{\alpha}$ using three Gauss-Seidel sweeps of $\widetilde{v}$ per iteration. The coarsest grid has spacing $h = 1/5$ in both directions. Parameter values are $B = g = 1$. Error reduction factors of the discrete L^2 norm of $\widetilde{\alpha}$ are about 9.2. Artifacts of the artificial boundary conditions can be seen as convergence progresses.

Figure 4.6: A Full Multigrid cycle with embedded Newton-style iteration.



* **Algorithm:** $(v^h, \alpha^h) \leftarrow \text{FMG-Cycle}(L, M_\alpha, M_v, M_{VC}, \widetilde{M}_{VC})$

1. If $L = 0$, solve the coarse-grid and return $(\widetilde{v}^h, \widetilde{\alpha}^h) = \text{ExactSolve}(0)$.
2. Otherwise set $(v^{2h}, \alpha^{2h}) = \text{FMG-Cycle}(L-1, M_\alpha, M_v, M_{VC}, \widetilde{M}_{VC})$.
3. Apply cubic interpolation to α^{2h} to get α^h. Generate grid $\Omega^h(\alpha^h)$ and let v^h be the cubic interpolation of v^{2h} to the new grid.
4. Repeat M_{VC} times: $v^h \leftarrow \text{V-Cycle}(L, v^h, 0, M_v)$
5. Initialize $(\widetilde{v}^h, \widetilde{\alpha}^h) = 0$ and repeat $\widetilde{M}_{VC}$ times:
   $(\widetilde{v}^h, \widetilde{\alpha}^h, M_\alpha, M_v) \leftarrow \text{V-Cycle}(L, \widetilde{v}^h, \widetilde{\alpha}^h, 0, M_\alpha, M_v)$
6. Update $v^h \leftarrow v^h + \widetilde{v}^h$ and $\alpha^h \leftarrow \alpha^h + \widetilde{\alpha}$ and return.

4.5 A Probabilistic Interpretation

A prominent operator in our numerical analysis is defined by

$$Ae^{\iota ky} = \underbrace{|k|}_{\widehat{A}(k)} e^{\iota ky}.$$

Taking A to be the generator of a Markov process, we solve the Fokker-Plank equation

$$\begin{aligned} p_t &= Ap \\ p(t, y; y') &= \delta_{y'}(y) \end{aligned}$$

to find the transition density is

$$\begin{aligned} p(t, y; y') = \int_{\mathbb{R}} \hat{p}(\theta, t) e^{\iota\theta(y-y')} d\theta, \\ \text{where } \hat{p}(\theta, t) \propto e^{-|\theta|t}. \end{aligned}$$

In the space variables, the transition density defines a *Cauchy process*:

$$\begin{aligned} \text{Prob}(Y_t \in E | Y_0 = y) = \int_E p(t, y; y') dy' \\ \text{where } p(y, y', t) \propto \frac{t}{(y-y')^2 + t^2}. \end{aligned}$$

What stochastic process in the context of the optimal stopping problem has the same transition probabilities?

Suppose the optimal stopping boundary is (α, y) for constant α. Let L_t^X be the cumulative *local time* that X has spent at this boundary. Formally, the local time is the random function

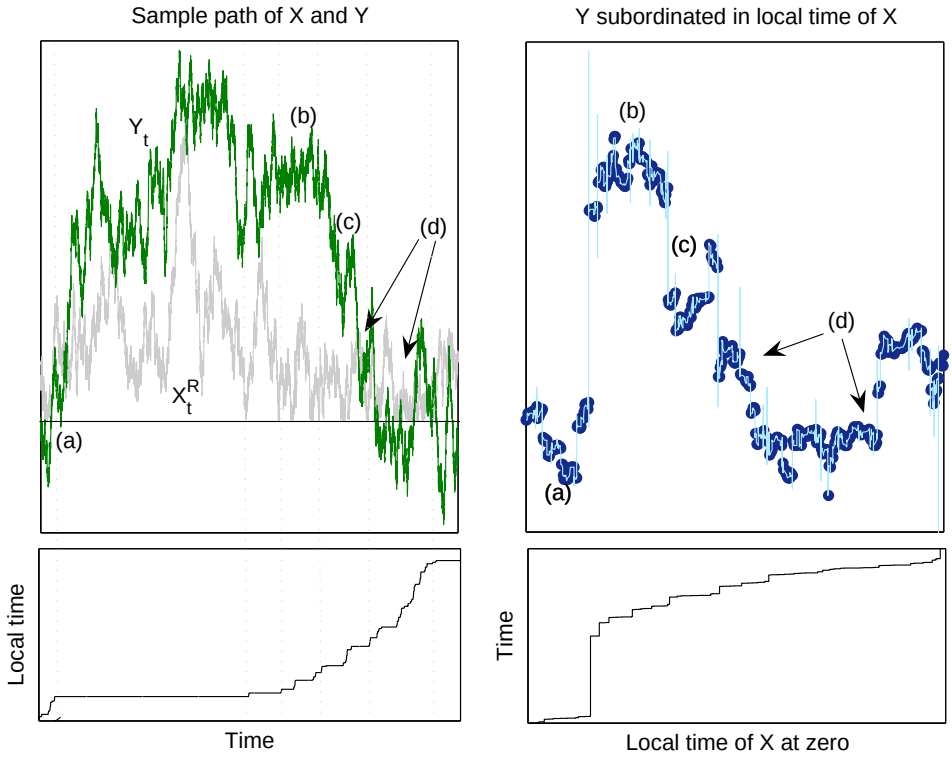
$$L_t^X = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \int_0^t \mathbb{1}(|X_s - \alpha| < \epsilon) ds.$$

Now define the reflection of the process X off the stopping boundary:

$$X_t^R = \begin{cases} X_t & \text{for } X_t \leq \alpha \\ \alpha - (X_t - \alpha) & \text{for } X_t > \alpha. \end{cases}$$

It is straightforward to show that the local time of X_t and X_t^R at α have the same transition

Figure 4.7: Sample paths of X^R and Y plotted in real time and Y subordinated in the local time of X at $\alpha = 0$. Vertical lines indicate where Y has visited during excursions X takes away from zero.



law. Let us now define a process that is Y subordinated in the local time of X^R. That is, the process

$$Z_s = Y_{(L^{X^R})^{-1}(s)}$$

is the Brownian Motion Y where the unit of time is measured not by the physical clock but by the amount of local time accumulated by X^R up to given physical time. A simple conditioning argument shows that the transition probabilities of Z_s are given by $p(s, z; z')$, so Z_s is a Cauchy process. We can relate these two constructions of a Cauchy process by considering the smooth pasting condition

$$v_x(\alpha, y) = 0$$

in relation to the Poisson equation

$$\frac{1}{2}\triangle v = g.$$

The probabilistic interpretation of the smooth pasting condition is that X_t is replaced with X_t^R, which reflects off the boundary in local time. If X is replaced with X^R, the transition density $q(t, x, y; x', y')$ for (X^R, Y) solves the Fokker-Plank equation

$$\begin{aligned} \frac{1}{2}\triangle q &= q_t \\ q_x(t, \alpha, y) &= 0. \end{aligned}$$

If the smooth pasting condition is considered a natural boundary condition for the value function, we can interpret it together with the value matching condition: At the stopping boundary, we are indifferent to stopping and getting zero and continuing if the state process reflects off the boundary in local time. The value matching condition $v(\alpha(y), y) = 0$ is entirely local: it does not depend on nearby points. By taking second derivatives, we effectively introduced a coupling.

To illustrate, suppose the process (X_t^R, Y_t) is on the stopping boundary $(\alpha(y), y)$. The process may then take an excursion away from this boundary and hit the reward boundary after time τ_0 or return to the stopping boundary after time τ_α. In this way, we can reduce the two-dimensional dynamic programming equation to the stopping boundary. Assuming constant g, the dynamic programming equation can be written

$$U(Y_t) = E\left[-g\Delta\tau + \mathbb{1}(\tau_0 < \tau_\alpha)B(Y_{\tau_0}) + \mathbb{1}(\tau_0 \geq \tau_\alpha)U(Y_{\tau_\alpha}) \;\middle|\; X_t = \alpha(Y_t), Y_t\right]$$

where $U(y) = v(\alpha(y), y)$ and $\Delta\tau$ is the amount of physical time the excursion takes, which is indicated by the inverse local time. The value matching condition $v(\alpha(y), y) = 0$ can then be interpreted as follows: The stopping boundary α makes zero the only solution of the above dynamic program. By using Itô's formula to transform the dynamic programming integral equation into an pseudo-differential equation, we find that the relevant operator corresponds to our preconditioned value matching condition where the operator generates a Lévy process.

At the discrete level, with a right side added to the smooth pasting condition, we have

$$\frac{v(\alpha, y) - v(\alpha - h, y)}{h} = f \implies v(\alpha, y) = v(\alpha - h, y) + hf.$$

The interpretation is that when X hits the boundary α it goes to state $\alpha - h$ with probability

one, and a reward of f dollars per unit local time is collected. In contrast to the interior equations where $\Delta t = h^2/2$, the local time step is $\Delta t^{\text{loc}} = h$. This can be seen in the rate at which information decays over time. We saw that $\hat{p}(t,\theta) \propto e^{-|\theta|t}$. In contrast, the transition density for a Brownian motion would have $\hat{p}^{BM}(t,\theta) \propto e^{-|\theta|^2 t}$. This implies that Brownian Motion damps high frequencies faster than does a Cauchy process.

Appendix A

Convergence History of Residuals

A printout of the residuals at all levels can be a valuable diagnostic tool. The following table gives the convergence history on each level of the first three V-Cycles shown in Figure 3.7. The patterns displayed here are characteristic of textbook multigrid performance.

Notice that at level $L = 0$ the coarsest equations are solved down to machine zero. At the finest level $L = 4$, the difference between the end of one V-Cycle and the beginning of the next is the effect of one relaxation sweep.

Level	Interior U	Smooth Pasting	Interior J	Calibration Condition
4	1.18e-5	-1.04e-4	2.20e-5	8.21e-3
3	1.15e-5	-1.07e-4	2.17e-5	8.21e-3
2	1.13e-5	-1.13e-4	2.20e-5	8.21e-3
1	1.08e-5	-1.26e-4	2.27e-5	8.22e-3
0	3.10e-14	5.32e-15	1.33e-14	-2.13e-14
1	8.59e-6	-1.26e-4	1.33e-5	-7.46e-6
2	1.13e-5	-1.93e-4	1.78e-5	-9.45e-6
3	1.28e-5	-2.27e-4	2.e-5	-9.83e-6
4	1.34e-5	2.44e-4	2.08e-5	-9.86e-6
4	1.52e-6	1.31e-5	2.18e-6	-8.66e-4
3	1.48e-6	1.34e-5	2.14e-6	-8.66e-4
2	1.45e-6	- 1.42e-5	2.16e-6	-8.66e-4
1	1.40e-6	1.58e-5	2.23e-6	-8.67e-4
0	2.06e-14	1.77e-15	2.25e-14	-4.26e-14
1	9.02e-7	-1.33e-5	1.40e-6	7.87e-7
2	1.19e-6	-2.03e-5	1.87e-6	9.98e-7
3	1.34e-6	-2.39e-5	2.11e-6	1.03e-6
4	1.40e-6	-2.57e-5	2.19e-6	1.04e-6
4	1.59e-7	-1.38e-6	2.30e-7	9.11e-5
3	1.55e-7	-1.41e-6	2.26e-7	9.11e-5
2	1.52e-7	-1.50e-6	2.28e-7	9.12e-5
1	1.47e-7	-1.67e-6	2.35e-7	9.13e-5
0	2.61e-14	-2.48e-14	2.94e-14	-1.13e-13
1	9.50e-8	-1.40e-6	1.47e-7	-8.29e-8
2	1.25e-7	2.14e-6	1.97e-7	-1.05e-7
3	1.41e-7	-2.52e-6	2.22e-7	-1.09e-7
4	1.47e-7	2.71e-6	2.31e-7	-1.09e-7

Bibliography

[1] S. Agmon, A. Douglis, and L. Nirenberg. Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions. I. *Comm. Pure Appl. Math*, 12:623–727, 1959.

[2] S. Agmon, A. Douglis, and L. Nirenberg. Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions. II. *Comm. Pure Appl. Math*, 17:35–92, 1964.

[3] M. Akian. *Méthodes multigrilles en contrôle stochastique*. PhD thesis, Université Paris IX-Dauphine, 1990.

[4] M. Akian, J. L. Menaldi, and A. Sulem. On an investment-consumption model with transaction costs. *SIAM. J. Control and Optimization*, 34,1:329–364, January 1996.

[5] E. Arian. On the coupling of aerodynamic and structural design. *Journal of Computational Physics*, 135,1:83–96, 1997.

[6] E. Arian and S. Ta'asan. Multigrid one-shot methods for optimal control problems: Infinite dimensional control. *ICASE Report*, No. 94-52, 1994.

[7] E. Arian and S. Ta'asan. Shape optimization in one-shot. In J. Borggaard, editor, *Optimal Design and Control*, pages 53–74. Birkhauser Verlag, 1995.

[8] E. Arian and S. Ta'asan. Analysis of the hessian for aerodynamics optimization: Inviscid flow. *ICASE Report*, 96-28, 1996.

[9] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[10] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–654, 1973.

[11] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math.Comput.*, 31:330–390, 1977.

[12] A. Brandt and C. W. Cryer. Multigrid algorithms for the solutions of linear complementarity problems arising from free boundary problems. *SIAM J. Sci. Stat. Comput.*, 4,4:655–683, 1983.

[13] K. A. Brekke and B. Oksendal. The high contact principle as a sufficiency condition for optimal stopping. In D. Lund and B. Oksendal, editors, *Stochastic Models and Option Values*, pages 187–208. North-Holland, 1991.

[14] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, 2000.

[15] J. Y. Campbell, A. W. Lo, and A. C. MacKinlay. *The Econometrics of Financial Markets*. Princeton University Press, 1997.

[16] N. Clarke and K. Parrott. Multigrid for American option pricing with stochastic volatility. *Applied Mathematical Finance*, 6:177–195, 1999.

[17] J. H. Cochrane. *Asset Pricing*. Princeton University Press, 2001.

[18] R. Cont and P. Tankov. *Financial Modelling with Jump Processes*. Chapman & Hall/CRC, 2004.

[19] R. Cont and P. Tankov. Nonparametric calibration of jump-diffusion option pricing models. *Journal of Computational Finance*, 7, 3:1–49, 2004.

[20] A. K. Dixit. *The Art of Smooth Pasting*. Harwood Academic Publishers, 1993.

[21] A. K. Dixit and R. S. Pindyck. *Investment Under Uncertainty*. Princeton University Press, 1994.

[22] D. Duffie, W. Fleming, H. Soner, and T. Zariphopoulou. Hedging in incomplete markets with HARA utility. *Journal of Economic Dynamics and Control*, 21:753–782, 1997.

[23] P. H. Dybvig and H. Liu. Lifetime consumption and investment: Retirement and constrained borrowing. AFA 2005 Philadelphia Meetings Paper, December, 2004.

[24] E. Farhi and S. Panageas. Saving and investing for early retirement: A theoretical analysis. Working Paper, January 2005.

[25] W. H. Fleming and M. Soner. *Controlled Markov Processes and Viscosity Solutions*. Springer, 1993.

[26] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins University Press; 3 edition, 1996.

[27] L. P. Hansen. Large sample properties of generalized method of moment estimators. *Econometrica*, 50:1029–1054, 1982.

[28] J. Hull. *Options, Futures and Other Derivative Securities*. Prentice Hall, 2000.

[29] K. L. Judd. Computational economics and economic theory: Substitutes or complements? *Journal of Economic Dynamics and Control*, 21, 6:907–942, 1997.

[30] K. L. Judd. *Numerical Methods in Economics*. The MIT Press, 1998.

[31] I. Karatzas and S. E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer-Verlag, 1991.

[32] I. Karatzas and S. E. Shreve. *Methods of Mathematical Finance*. Springer-Verlag, 1998.

[33] D. Kinderlehrer and G. Stampacchia. *An Introduction to Variational Inequalities and Their Applications*. Classics in Applied Mathematics, 31. SIAM, 2000.

[34] H. K. Koo. Consumption and portfolio selection with labor income: a continuous time approach. *Mathematical Finance*, 8(1):49–69, January 1998.

[35] H. J. Kushner and P. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer, 2001.

[36] F. Kydland and E. C. Prescott. The computational experiment: An econometric tool. *Journal of Economic Perspectives*, 10, 1:69–85, 1996.

[37] R. M. Lewis and S. G. Nash. A multigrid approach to the optimization of systems governed by differential equations. Paper AIAA-2000-4890, 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, 2000.

[38] L. Ljungqvist and T. J. Sargent. *Recursive Macroeconomic Theory : Second Edition*. The MIT Press, 2004.

[39] R. E. Lucas. Asset prices in an exchange economy. *Econometrica*, 46:1426–1446, 1978.

[40] R. E. Lucas. Methods and problems in business cycle theory. *Journal of Money, Credit, and Banking*, 12,4:696–715, 1980.

[41] R. Mehra and E. Prescott. The equity premium: A puzzle. *Journal of Monetary Economics*, 22:145–161, 1985.

[42] R. C. Merton. Lifetime portfolio selection under uncertainty: the continuous time case. *The Review of Economic Studies*, 51:247–257, 1969.

[43] R. C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4:141–183, 1973.

[44] C. Munk. Optimal consumption/investment policies with undiversifiable income risk and liquidity constraints. *Journal of Economic Dynamics and Control*, 24,9:1315–1343, 2000.

[45] C. Munk. The markov chain approximation approach for numerical solution of stochastic control problems: Experiences from Merton's problem. *Applied Mathematics and Computation*, v136 n1:47–77, 2003.

[46] M. Musiela and M. Rutkowski. *Martingale Methods in Financial Modeling*. Springer, 1997.

[47] R. A. Nicolaides. On multiple grid and related techniques for solving discrete elliptic systems. *J. Comp. Phys.*, 19:418–431, 1975.

[48] R. A. Nicolaides. Direct discretization of planar div-curl problems. *SIAM J. Numer. Anal.*, 29(1):32–56, 1992.

[49] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

[50] B. Oksendal and A. Sulem. *Applied Stochastic Control of Jump Diffusions*. Springer, 2005.

[51] O. Pironneaux. *Optimal Shape Design for Elliptic Systems*. Springer, 1983.

[52] J. Rust. Numerical dynamic programming in economics. In H. M. Amman, D. A. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*, pages 417–488. North-Holland, 1996.

[53] J. Rust. Using randomization to break the curse of dimensionality. *Econometrica*, 65(3):487–516, 1997.

[54] S. Sodal. A simplified exposition of smooth pasting. *Economics Letters*, 58:217–223, 1998.

[55] S. Ta'asan. One-shot methods for optimal control of distributed parameter systems I: Finite dimensional control. *ICASE Report*, No. 91-2, 1991.

[56] S. Ta'asan. Fast solvers for MDO problems. In N. M. Alexandrov, editor, *Multidisciplinary Design Optimization, state of the art*, pages 53–74. SIAM, 1997.

[57] U. Trottenberg, A. Schuller, and C. W. Oosterlee. *Multigrid*. Elsevier, 2000.