Department of Mathematics at University of California, San Diego

Computational Finance

Differential Equation Techniques [**Lectures 8-10**]

Michael Holst

June 1, 2015

Contents

1 Modeling Financial Options with the Black-Scholes Equation (Lecture 8)

Recall that by an *option*, we mean the right to buy (a *call option*) or sell (a *put option*) one unit of a potentially risky *underlying asset* (typically a stock) within an agreed upon *fixed period* T at an agreed upon *fixed price* K per unit. By *right*, we mean we have the option, but we we do not an *obligation* to carry out the transaction. Since an option (and related financial instruments) are built and priced based on an *underlying asset*, they are refered to as a particular type of (financial) *derivative*. An option is simply a contraction between a seller (or *writer*), and a buyer (or *holder*); the *holder* purchases the option from the *writer*, paying what is called a *premium*. In this way, options are effectively a financial instrument for betting on a stock price either rising or falling. What value should the seller charge for the premium? What premium should a buyer be willing to pay? In this section, we will develop *differential equation models* of financial options to help us answer these questions. An outline of this section is as follows.

In Section 1.1, we will look at the basic ideas behind modeling options with differential equations, based on the Black-Merton-Scholes model of the market. We will derive the Black-Scholes equation using Ito's Lemma, and then examine how the Black-Scholes equation can be used to model standard options as well as exotic options. We will also look at some nonlinear options models that go beyond the basic Black-Scholes equation. Since the Black-Scholes equation cannot be solved in closed form except for the simplest situations, in Section 2 we will develop basic some basic algorithms for solving the Black-Scholes equations numerically using a computer. This will require that we understand a little bit more about the Black-Scholes equations, and about how to develop and use numerical methods for equations of this type. Our focus in Section 2 will be on finite difference methods, including their basic formulation and their properties. Some key concerns are stability, consistency, and convergence, and this will lead us to look at two different types of methods, namely explicit and implicit methods. One of our key concerns will be how to treat boundary conditions; while the Black-Scholes model of European options fit nicely into the standard formulation of finite difference methods, one of the difficult features of using the Black-Scholes model for American options is the appearance of a *free boundary*. We will examine some techniques for handling free boundaries numerically, and look at some of the available software tools such as MATLAB.

Most of our discussion of financial options, and their modeling using differential equations, follows [4, 2]. The material on ordinary and partial differential equations, and numerical methods for their solution, may be found in [1, 3].

1.1 Differential Equation Models of Financial Options

We now develop some basic ideas for modeling financial options using differential equations. Let us first make some definitions for continuing our preliminary discussion above: •1

- ***option:*** right to buy or sell one unit of underlying asset within fixed time at fixed cost
- ***underlying:*** the asset on which the option is built
- ***writer:*** the seller of the option
- ***holder:*** the buyer of the option
- ***premium:*** the cost of the option
- ***T:*** the maturity date of the option
- ***K:*** the strike (or exercise) price
- ***payoff:*** value of option at strike time t
- ***call option:*** right to buy underlying for agreed price K by date T

•1 *WARNING. There are still some typos in the next few sections that I will fix in the next couple of days. -michael holst*

- ***put option:*** right to sell underlying for agreed price K by date T

The contract between the writer and the holder is generally as follows, although some of the details will vary between different types of options, such as American, European, and Asian options.

Required Writer Actions (at any time $0 \leq t \leq T$***):***

- *calls:* Must deliver underlying for strike price K (at any $t \leq T$).
- *puts:* Must buy underlying for strike price K (at any $t \leq T$).

Possible Holder Actions (at any time $0 \leq t$***):***

- Retain option (do nothing).
- Sell option at current market price on market exchange (at any $t < T$).
- Exercise the option (at any $t \leq T$).
- Let the option expire as "worthless" (at any $t \geq T$).

The basic idea is that the writer sells the option, and the premium they recieves compensates the risk they take. The holder buys the option, and the premium they pay is for the chance to exercise the option at the right time to earn back more than the premium. In these notes we will follow the approach in [4, 2] and build mathematical models options primarily from the perspective of the *holder* (the models can also be rewritten so as to be from the perspective of the writer).

The distinctions between American, European, and Asian (and other *exotic* options) are mainly concerning these three features:

1) One underlying asset, or a *basket* of underlying assets.
2) Permitted strike (exercise) times: any $t \in [0, T]$, or only at $t = T$.
3) Instantaneous payoff, or a path-dependent payoff.

The common types of options are then:

- ***Standard (Vanilla) Option:*** one underlying with instantaneous payoff.
- ***Exotic Option:*** an option which is not a standard option.
- ***American Option:*** A standard option that can be exercised at any t, $0 \leq t \leq T$.
- ***European Option:*** A standard option that can be exercised only at $t = T$.
- ***Asian Option:*** An exotic option, with one or a basket of underlyings, where the payoff depends on the average value of the asset during option lifetime.
- ***Barrier Option:*** An exotic option, with one or a basket of underlyings, where the payoff depends on whether stock price hits a prescribed barrier during option lifetime.

We are going to focus mainly on *standard options* in the sense listed above, although all of the techniques we will develop can (and have been) extended to exotic options, and more complex options models.

1.2 Basic Ideas and Variables for Modeling Standard Options

For the moment, let us restrict the discussion to *standard options*, which from our discussion above means: •2

1) One underlying asset.
2) Instantaneous payoff.

At this point we need to introduce some additional notation in order to make things a bit more precise. First, there are four key parameters that play a role in building a differential equation model of an option. Each parameter is real-valued, meaning that it is a single real number for a

•2 *WARNING. There are still some typos in the next few sections that I will fix in the next couple of days. -michael holst*

particular instance of the model. This will be important from a modeling perspective, since it will allow us to consider these parameters as all taking values in the set of real numbers $\mathbb{R}$.

Parameter	Meaning
K	The strike (or exercise) price.
T	The maturity date of the option (in years).
r	Continuously compounded risk-free interest rate (per year).
σ	Annual volatility of the underlying asset (per year).

The parameters r and σ are *market parameters* that affect the price of the option $V(t)$. The parameter r is the risk-free interest rate (per year). For example, if $r = 0.07$, then the interest rate is 7%. The parameter σ is the volatility of the price $S(t)$, which can be defined as the stndard deviation of fluctions in $S(t)$, which is typically divided by $\sqrt{T}$ for scaling purposes. For example, if $\sigma = 0.4$, then the volatility of 40%. We will assume that both r and σ are constant throughout the lifetime of the option, so that $r, \sigma \in \mathbb{R}$ are fixed real numbers for all $t \in [0, T]$. Similarly, for a particular option being modeled, both $K, T \in \mathbb{R}$ are fixed real numbers for all $t \in [0, T]$.

There are three variables that play a role in a differential equation model of an option. The first is the independent real variable t, taking values of time (measured in years) in the range $0 \leq t \leq T$, which we write as $t \in [0, T]$. Next is the value of the underlying asset, denoted as $S(t)$, which is a time-dependent function that represents input to our model of the option. It is a real-value function of its single real independent variable t, and we equivalently will write $S(t)$, or S_t, or simply as S. The third variable is the main dependent variable we wish to develop a model for, namely the value of the option V. This is a real-value function of its *two* variables S and t; we denote this as $V(S, t)$ to emphasize its dependence on both t and $S(t)$. However, it is also a function of the four parameters K, T, r, and σ above, so that in principal we should think about V as a function of all six (the two variables and four parameters), or $V = V(S, t; K, T, r, \sigma)$.

Variable	Meaning
t	The current time $t \in [0, T]$ (in years)
$S(t)$	Spot (current, instantaneous) price of the underlying asset S at time $t \in [0, T]$
$V(S, t)$	Instantaneous value V of the option for asset S at time $t \in [0, T]$

For a *rational investor* (i.e., an investor trying to maximize profit and minimize costs), the value of a standard call option, denoted V_c, at its exercise time of $t \in [0, T]$ is clearly:

$$V_c(S_t, t) = \left\{ \begin{array}{ll} 0, & \text{if } S_t \leq K \text{ (option is worthless)} \\ S_t - K, & \text{if } S_t > K \text{ (option is exercised)} \end{array} \right\}. \tag{1.1}$$

Similarly, for a rational investor, the value of a standard put option, denoted V_p, at its exercise time of $t \in [0, T]$ is clearly:

$$V_p(S_t, t) = \left\{ \begin{array}{ll} 0, & \text{if } S_t \geq K \text{ (option is worthless)} \\ K - S_t, & \text{if } S_t < K \text{ (option is exercised)} \end{array} \right\}. \tag{1.2}$$

These are called the *payoff functions*. We can write the payoff functions more compactly as:

$$V_c(S_t, t) = \max\{S_t - K, 0\}, \qquad \text{(call payoff function)} \tag{1.3}$$

$$V_p(S_t, t) = \max\{K - S_t, 0\}. \qquad \text{(put payoff function)} \tag{1.4}$$

However, it is useful to define the following function:

$$g^+ = \max\{g, 0\}, \qquad g \in \mathbb{R}. \tag{1.5}$$

This allows an even more compact representation:

$$V_c(S_t, t) = (S_t - K)^+, \tag{1.6}$$
$$V_p(S_t, t) = (K - S_t)^+. \tag{1.7}$$

Note that payoff functions of this type are *pointwise*, meaning that their value depends only on the current stock price $S_t = S(t)$ at the exercise time $t \in [0, T]$. Path-dependent payoff functions are used for exotic options, and they cannot be written in this form since their values depend on the value of the stock $S(t)$ at other time points $t \in [0, T]$.

The payoff functions (1.6)–(1.7) are for standard options (one underlying asset with instantaeneous payoff). American options allow for exercising at any $t \in [0, T]$, so that the payoff is then simply the value of these functions evaluated at time t. European options require exercising only at $t = T$, so that their payoff functions are always (1.6)–(1.7) evaluated at $t = T$, or:

$$V_c(S_T, T) = (S_T - K)^+, \tag{1.8}$$
$$V_p(S_T, T) = (K - S_T)^+. \tag{1.9}$$

When there is a need to draw a distinction between European and American options (and payoffs), we will denote then as V^E for European options, and V^A for American options.

A rational investor will only exercise the option when it is profitable, leading to the form of the payoff functions (1.6)–(1.7). Since European options can only be exercised at $t = T$, there is less opportunity to maximize profit/minimize cost when compared to American options. This is a type of *a priori* bound or inequality that relates the two option values, and the option values with the call and stock values. The following *a priori bounds* can be derived for European and American options (V with no superscript applies to both types of options):

$$0 \leq V_c(S_T, T) \leq S_t, \tag{1.10}$$
$$0 \leq V_p(S_T, T) \leq K, \tag{1.11}$$
$$V^E(S_t, t) \leq V^A(S_t, t), \tag{1.12}$$
$$S_t - K \leq V_c^A(S_t, t), \tag{1.13}$$
$$K - S_t \leq V_p^A(S_t, t), \tag{1.14}$$
$$V_p^E(S_t, t) \leq Ke^{-r(T-t)}. \tag{1.15}$$

1.3 The Black-Merton-Scholes Model of Standard Options

The *Black-Merton-Scholes model* of the market involves the following basic assumptions:

Assumption 1.1 (Black-Merton-Scholes Market Model)**.** *The option is a standard European option, and:*

- *There is no arbitrage.*
- *It is a frictionless market.*
- *The underlying asset price obeys a geometric Brownian motion.*
- *The interest rate and volatility r and σ are constant for all $t \in [0, T]$.*

The *no arbitrage* assumption or principle is that aribitrage opportunties are not available (or not allowed). A fictionless market means that there are no transaction fees, that the cost of lending and borrowing money is equal, that all information is available immediately with no delay, and that all assets and credits are available at any time and in any amount. The assumption of constant interest rate and volatility is an idealization, but necessary of this model.

The assumption that the underlying asset obeys a *geometric Brownian motion* means that it follows

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \tag{1.16}$$

with constant μ and σ. This is a special (linear) case of an Itŏ motion:

$$dX_t = a(X_t, t)dt + b(X_t, t)dW_t, \tag{1.17}$$

where W_t is a Weiner process (itself a very special case of an Itŏ motion, $dX_t = dW_t$, i.e. $a = 0$ and $b = 1$). A dividend with continuous yield $\delta \geq 0$ can be included into the Brownian motion:

$$dS_t = (\mu - \delta)S_t dt + \sigma S_t dW_t, \tag{1.18}$$

which produces decrease in S at each interval dt by amount $\delta S dt$.

1.4 Derivation of the Black-Scholes Equation Using Ito's Lemma

Imagine that we have a portfolio at time t with α_t shares of an asset with value S_t, and β_t shares of bond B_t. Assume the bound is riskless so that

$$dB_t = rB_t dt. \tag{1.19}$$

At time t, the *wealth process* of the portfolio is:

$$P_t = \alpha_t S_t + \beta_t B_t. \tag{1.20}$$

We wish the portfolio to hedge a European option with value V_t and payoff V_T at maturity $t = T$. Therefore, we aim to construct α_t and β_t so that the portfolio *replicates* the paoff, i.e.

$$P_T = V_T = \text{ payoff }. \tag{1.21}$$

Since a European option cannot be traded before maturity, our portfolio will be assumed to be *closed*, with no investment and no payout before $t = T$. This means our portfolio has the *self-finance* property:

$$dP_T = \alpha_t dS_t + beta_t dB_t. \tag{1.22}$$

Thus, the change in value P_t is due only to changes in the prices of S and B. Note that together with the no arbitrage assumption, (1.21)–(1.22) imply

$$P_t = V_t, \qquad t \in [0, T], \tag{1.23}$$

since both have the same payout at $t = T$. Therefore, the replicating and self-financing portfolio is equivalent to the European option, and duplicates the risk of the option.

Assume now that the value function $P_t = V(S, t)$ is sufficiently smooth (in fact, we need $V \in C^{2,1}$, meaning twice continuously differentiable in S, and ones continuously differentiable in t). This gives us access to *Itŏ's Lemma*, which is basically a stochastic version of the following (deterministic) chain rule:

$$\frac{d}{dt}g(x(t), t) = \frac{\partial g}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial g}{\partial t}.$$

The standard statement of this Lemma is the following.

Lemma 1.2 (Itŏ's Lemma). *Suppose X_t is an Itŏ process, and let $g \in C^{2,1}$. Then $Y_t = g(X_t, t)$ is also an Itŏ process with the same Weiner process W_t, and*

$$dY_t = \left(\frac{\partial g}{\partial x}a + \frac{\partial g}{\partial t} + \frac{1}{2}\frac{\partial^2 g}{\partial x^2}b^2\right) + \frac{\partial g}{\partial x}bdW_t,$$

where the derivatives and cofficients depend on X_t and t.

If we apply Itŏ's Lemma to our geometric Brownian motion (1.18) we obtain

$$dP = \left(\mu S\frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2}\right)dt + \sigma S\frac{\partial V}{\partial S}dW. \tag{1.24}$$

However, if we combine (1.18), (1.19), and (1.22) we obtain an alternate expression:

$$dP = (\alpha\mu S + \beta r B)\,dt + \alpha\sigma S dW. \tag{1.25}$$

The coefficients in these two expressions for dP must match; matching the dW cofficients gives an equation for α:

$$\alpha_t = \frac{\partial V(S_t, t)}{\partial S}. \tag{1.26}$$

Matching the dt coefficients gives a relation for β, where the Weinter term drops out. The term βB can be eliminated using (1.20) and (1.23) via:

$$S\frac{\partial V}{\partial S} + \beta B = V. \tag{1.27}$$

This then gives finally the *Black-Scholes equation*

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} + (r-\delta)S\frac{\partial V}{\partial S} - rV = 0, \qquad 0 < S < \infty, \qquad 0 \le t < T, \tag{1.28}$$

$$V(T) = P_T. \tag{1.29}$$

1.5 Reformulation of Black-Scholes for Numerical Solution

Now that we have the Black-Scholes equation, how do we solve it in order to determine a good pricing for the option, either as a write or a holder? Unfortunately, except for very special simplified models, we cannot solve the Black-Scholes equation (1.28) in "closed form", meaning that we generally cannot write down a functin $V(S,t)$ that solves (1.28)–(1.29). Instead, what we will now do is develop numerical methods, paying attention to how they are designed so that they are both *accurate* and *reliable*.

The first thing we need to do is reformulate (1.28)–(1.29) into a mathematically equivalent form so that it is in a more standard form for use with numerical methods. To this end, we will use the following change of variables:

$$S = Ke^x, \qquad t = T - \frac{2\tau}{\sigma^2}, \qquad q = \frac{2r}{\sigma^2}, \qquad q_\delta = \frac{2(r-\delta)}{\sigma^2}. \tag{1.30}$$

Under this transformation, the value function $V(S,t)$ can be written in terms of the new variables x and τ as:

$$V(S,t) = V\left(Ke^x, T - \frac{2\tau}{\sigma^2}\right) = v(x,\tau), \tag{1.31}$$

where

$$v(x,\tau) = Ke^Q y(x,\tau), \qquad Q = -\frac{1}{2}(q_\delta - 1)x - (\frac{1}{4}(q_\delta - 1)^2 + q)\tau. \tag{1.32}$$

This then gives us finally the following initial value problem for the unknown function $y(x,\tau)$:

$$\frac{\partial y}{\partial \tau} = \frac{\partial^2 y}{\partial x^2}, \qquad -\infty < x < \infty, \qquad 0 \le \tau < \frac{1}{2}\sigma^2 T. \tag{1.33}$$

Some observations:

1) We first solve (1.33) for $y(x,\tau)$, then form $v(x,\tau)$ via (1.32), and then finally recover $V(S,t)$ via (1.31).
2) The "backward" equation for $V(S,t)$ is now a "forward" equation for $y(x,\tau)$.
3) The final time for τ is now $\tau = \frac{1}{2}\sigma^2 T$.
4) The initial conditions for a standard call are:

$$V(S,T) = (S-K)^+ = K\max\{e^x - 1, 0\} = Ke^{-\frac{x}{2}(q_\delta - 1)}y(x,0),$$

with

$$y(x,0) = e^{\frac{x}{2}(q_\delta-1)}\max\{e^x-1,0\} = \begin{cases} e^{\frac{x}{2}(q_\delta-1)}(e^x-1), & x > 0, \\ 0 & x \le 0. \end{cases}$$

Using the identity

$$e^{\frac{x}{2}(q_\delta-1)}(e^x-1), = e^{\frac{x}{2}(q_\delta+1)} - e^{\frac{x}{2}(q_\delta-1)},$$

we have the initial conditions for $y(x,\tau)$:

$$\text{call: } y(x,0) = \max\{e^{\frac{x}{2}(q_\delta+1)} - e^{\frac{x}{2}(q_\delta-1)}, 0\} \tag{1.34}$$

$$\text{put: } y(x,0) = \max\{e^{\frac{x}{2}(q_\delta-1)} - e^{\frac{x}{2}(q_\delta+1)}, 0\}. \tag{1.35}$$

5) Truncating the infinite domain $-\infty < x < +\infty$ is necessary for use of (most) numerical methods; we would like to restrict the domain to a finite interval

$$x_{\min} \le x \le x_{\max}.$$

This can be based (for European options) on *Put-Call Parity*:

$$S + V_p^E - V_c^E = Ke^{-r(T-t)},$$

which can be shown to hold. This gives:

$$V_c(S,t) = S - Ke^{-r(T-t)}, \qquad S \to \infty, \tag{1.36}$$

$$V_p(S,t) = Ke^{-r(T-t)} - S, \qquad S \to 0. \tag{1.37}$$

For European options, one uses as boundary conditions the following:

$$y(x,\tau) = r_1(x,\tau), \qquad x \to -\infty, \tag{1.38}$$

$$y(x,\tau) = r_2(x,\tau), \qquad x \to +\infty, \tag{1.39}$$

where for a call option, one has:

$$r_1(x,\tau) = 0, \tag{1.40}$$

$$r_2(x,\tau) = e^{\frac{1}{2}(q_\delta+1)x + \frac{1}{4}(q_\delta+1)^2\tau}, \tag{1.41}$$

and for a put option, one has:

$$r_1(x,\tau) = e^{\frac{1}{2}(q_\delta-1)x + \frac{1}{4}(q_\delta-1)^2\tau}, \tag{1.42}$$

$$r_2(x,\tau) = 0. \tag{1.43}$$

These are known as *Dirichlet boundary conditions*, and are the last piece of the puzzle we need for developing numerical methods for the Black-Scholes equation.

2 Finite Difference Methods for the Black-Scholes Equations (Lectures 9-10)

Since the Black-Scholes equation cannot be solved in closed form except for the simplest situations, in this section we will develop basic some basic algorithms for solving the Black-Scholes equations numerically using a computer. This will require that we understand a little bit more about the Black-Scholes equations, and about how to develop and use numerical methods for equations of this type. Our focus in this section will be on *finite difference methods*, including their basic formulation and their properties. Some key concerns are stability, consistency, and convergence, and this will lead us to look at two different types of methods, namely explicit and implicit methods. One of our key concerns will be how to treat boundary conditions; while the Black-Scholes model of European options fit nicely into the standard formulation of finite difference methods, one of the difficult features of using the Black-Scholes model for American options is the appearance of a *free boundary*. We will examine some techniques for handling free boundaries numerically, and look at some of the available software tools such as MATLAB.

2.1 Finite Difference Methods: Basic Formulation and Properties

To develop a basic understanding of finite difference methods, let us first examine a simplified version of (1.33) that is time-independent:

$$-\frac{d^2u}{dx^2} = f, \qquad x \in (a,b) \subset \mathbb{R}, \tag{2.1}$$

$$u = u_a, \qquad \text{at } x = a, \tag{2.2}$$

$$u = u_b, \qquad \text{at } x = b. \tag{2.3}$$

When there is only one independent variable (x in this case), it is sometimes convenient to use the simplified notation $u' = \frac{du}{dx}$ and $u'' = \frac{d^2u}{dx^2}$, in which case (2.1) would read:

$$-u'' = f, \qquad x \in (a,b) \subset \mathbb{R}. \tag{2.4}$$

As we noted earlier in the notes, equation (2.1) (or (2.4)), together with the boundary conditions (2.2)–(2.3) is referred to as a *Boundary Value Problem (BVP) in Ordinary Differential Equations (ODE)*. Since the boundary conditions involve the two endpoints of the interval $[a,b]$, this problem is also sometimes called a *Two-Point Boundary Value Problem (in ODE)*.

We would like to find the unknown function $u(x)$, where $u\colon [a,b] \to \mathbb{R}$, where we are given the interval $[a,b] \subset \mathbb{R}$ with $a < b$, we are given the model function $f(x)$, where $f\colon [a,b] \to \mathbb{R}$, and where we are also given the known boundary values of the function we are after, namely $u(a) = u_a$ and $u(b) = u_b$. If the function $f(x)$ were particularly simple, such as if it were a polynomial or a trigonometric function, then techniques exist that allow for determining $u(x)$ is *closed form*, meaning that we can simply write it down as an explicit function of f and the other input data. However, for the function f which would arise in mathematical models of various physical or other phenomena, we would have no choice but to develop some approximation to the solution u, such as is provided by *numerical methods*. One of the simplest and easiest numerical methods to understand and work with are *finite difference methods*.

Let us now develop a basic finite difference method for (2.1)–(2.3). Let us first recall how we defined the derivative when we first encountered it in the first course on the calculus:

$$\frac{du}{dx} = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h}. \tag{2.5}$$

If we stop short in the limit process and just make sure that h is "small enough" so that we get "close" to the derivative, then we have a *finite difference approximation* to the derivative:

$$\frac{du(x)}{dx} \approx \frac{u(x+h)-u(x)}{h}. \tag{2.6}$$

How good (i.e., accurate) is this approximation? Since we have reviewed a bit about Taylor expansion earlier in the notes, we can actually answer this question precisely. To do so, we simply Taylor expand the term $u(x+h)$ appearing above:

$$\frac{u(x+h)-u(x)}{h} = \frac{[u(x)+u'(x)h+u''(x(\xi))\frac{h^2}{2!}]-u(x)}{h} \tag{2.7}$$

$$= \frac{u'(x)h+u''(x(\xi))\frac{h^2}{2}}{h} \tag{2.8}$$

$$= u'(x)+u''(x(\xi))\frac{h}{2}, \tag{2.9}$$

$$= u'(x)+O(h), \tag{2.10}$$

where we have used the shorthand notation

$$u' = \frac{du}{dx}, \qquad u'' = \frac{d^2u}{dx^2},$$

and where $x(\xi)$ is the point in the interval $(u, u+h)$ that allows the Taylor remainder to appear as the single term involving u'' (see our earlier discussion about Taylor expansion and remainders). Therefore, this finite difference is a *first-order* approximation to u'. Notice that we have also confirmed that in fact it does actually approximate u', since we showed that the difference formula can be written as u' plus an error that goes to zero as h goes to zero. Here is another difference formula known as the *centered difference* approximation to u':

$$\frac{du(x)}{dx} \approx \frac{u(x+h)-u(x-h)}{2h}. \tag{2.11}$$

We can again check the accuracy, this time expanding both terms; let us just examine the numerator first:

$$u(x+h)-u(x-h) = [u(x)+u'(x)h+u''(x)\frac{h^2}{2!}+u'''(\xi_1)\frac{h^3}{3!}] \tag{2.12}$$

$$-[u(x)-u'(x)h+u''(x)\frac{h^2}{2!}-u'''(\xi_2)\frac{h^3}{3!}] \tag{2.13}$$

$$= 2u'(x)h+[u'''(\xi_1)+u'''(\xi_2)]h^3 \tag{2.14}$$

$$= 2u'(x)h+2u'''(\xi)h^3 \tag{2.15}$$

$$= 2u'(x)h+O(h^3), \tag{2.16}$$

where we have used the observation that if $u'''(x)$ is continuous on $[a,b]$, then there exists ξ such that $u(\xi) = [u'''(\xi_1)+u'''(\xi_2)]/2$, allowing us to combine the two Taylor remainder terms into a single term. If we now look at the actual difference formula, we find:

$$\frac{u(x+h)-u(x-h)}{2h} = \frac{2u'(x)h+2u'''(\xi)h^3}{2h} = u'(x)+u'''(\xi)h^2 = u'(x)+O(h^2). \tag{2.17}$$

Therefore, the centered difference (2.11) also approximates u', but it is *second order*: the error term goes to zero as h^2.

By writing down more complex difference formulas, possibly involving more than two points, we can derive higher and higher order approximations to u'. However, our differential equation (2.4) involves the second derivative u''; is their an analogous finite difference formula for a second derivative? Yes, in fact there is a multitude of difference formulas that can be derived for approximations of various derivatives; the one we will find most useful for the second derivative is also a *centered difference* formula, but now for approximating u'':

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = u''(x) + O(h^2). \tag{2.18}$$

To confirm that it has this order of accuracy, it is again a simple Taylor expansion of the two terms $u(x+h)$ and $u(x-h)$ and then a cancellation and combining of terms, as we did above. (Feel free to check this!)

To use this approximation to the derivative throughout the interval $[a, b]$, our first step is to create a finite difference *grid* or *mesh* of points out of the set of independent variable x:

$$a = x_0 < x_1 < x_2 < \ldots < x_n < x_{n+1} = b, \qquad x_i = a + ih, \qquad h = \frac{b-a}{n+1}. \tag{2.19}$$

This is a *uniform mesh* of h points, since the separation between each of the mesh points is exactly $h = (b-a)/n + 1$. If we now use the notation $u_i \approx u(x_i)$, then (2.18) reads

$$u''(x_i) = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + O(h^2). \tag{2.20}$$

If we use this expression to appoximate our differential equation (2.4) at the single point x_i, then we have:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = f_i + O(h^2), \tag{2.21}$$

where we have used the notation $f_i = f(x_i)$. If we drop the $O(h^2)$ error term in (2.21), and instead find u_i satisfying

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = f_i, \tag{2.22}$$

then we no longer finding $u(x_i)$ exactly; instead we are building an approximation:

$$u_i \approx u(x_i).$$

As we have seen above, through Taylor expansion we can characterize the accuracy of this approximation. If we multiply (2.22) through by h^2, and consider finding such a u_i for each point $i = 1, \ldots, n$, then we have a *linear system of equations* for the u_i:

$$-u_{i+1} + 2u_i - u_{i-1} = h^2 f_i, \qquad i = 1, \ldots, n. \tag{2.23}$$

Note that this is a set of n equations in n unknowns u_i, for $i = 1, \ldots, n$. Note that the left-hand side of each equation in (2.23) for $i = 2, \ldots, n-1$ only involves unknowns u_i, where i is between 1 and n. However, the equations for $i = 1$ and $i = n$ are special:

$$-u_2 + 2u_1 - u_0 = h^2 f_1, \tag{2.24}$$

$$-u_{n+1} + 2u_n - u_{n-1} = h^2 f_n. \tag{2.25}$$

In particular, the point u_0 in (2.24) and the point u_{n+1} in (2.25) are actually *known*, and not part of the set of unknowns; they are given by the boundary conditions (2.2)–(2.3). If we use these conditions and move that known information to the right-hand sides, then these two special equations become:

$$-u_2 + 2u_1 = h^2 f_1 + u_a, \tag{2.26}$$
$$2u_n - u_{n-1} = h^2 f_n + u_b. \tag{2.27}$$

All together then we have a system of n equations in n unknowns, which approximates both the differential equation (2.1) and the boundary conditions (2.2)–(2.3):

$$-u_2 + 2u_1 = h^2 f_1 + u_a, \tag{2.28}$$
$$-u_{i+1} + 2u_i - u_{i-1} = h^2 f_i, \qquad i = 2, \ldots, n-1, \tag{2.29}$$
$$2u_n - u_{n-1} = h^2 f_n + u_b. \tag{2.30}$$

We can write this as the following matrix system:

$$AU = F, \tag{2.31}$$

where $A \in \mathbb{R}^{n \times n}$ and $U, F \in \mathbb{R}^n$ are defined as:

$$A = \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & \end{bmatrix}, \qquad U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}, \qquad F = \begin{bmatrix} f_1 + u_a \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n + u_b \end{bmatrix}. \tag{2.32}$$

One can show that the matrix A has some particularly nice properties:

1) It is tri-diagonal; the only non-zero entries are on the main diagonal and the first super- and sub-diagional.
2) It is symmetric: $A_{ij} = A_{ji}$.
3) It is positive definite: $x^T A x > 0$, $\forall x \neq 0$.

We would need to solve this matrix system (using MATLAB's builtin solver, or using another computer algorithm) to produce our approximation in the vector $U \in \mathbb{R}^n$, giving finally

$$u_i \approx u(x_i), \quad i = 1, \ldots, n+1, \tag{2.33}$$

with of course equality at the two end points $i = 0$ and $i = n+1$, since the values of u_0 and u_{n+1} are the known boundary values.

As we discussed earlier in the notes, time-independent problems of this type involving more than one independent variable are called *elliptic partial differential equations*. Elliptic equations represent models of stationary (time-independent) phenomena, such as electrostatics, elastostatics, and so forth. The BVP in ODE we discussed above is the simplest case, and can be viewed as a one-dimensional elliptic equation. The simplest and most common elliptic equation is known as the *Poisson Equation*:

$$-\Delta u = f, \quad x \in \Omega \subset \mathbb{R}^d, \tag{2.34}$$
$$u = 0, \quad x \text{ on } \partial\Omega. \tag{2.35}$$

As we noted earlier in the notes, equation (2.34) together with the boundary conditions (2.35) is referred to as a *Boundary Value Problem (BVP) in Partial Differential Equations (PDE)*. Here, $\Omega \subset \mathbb{R}^d$, known as the *spatial domain*, is the set over which the independent variable $x \in \mathbb{R}^d$ is allowed to range. (For example, Ω might be the sphere of radius one centered at the origin, known as the unit sphere, or it might be a cube.) The set $\partial\Omega$ is the boundary of Ω (e.g. the surface of the unit sphere, or the boundary of a cube). The unknown function u is a real-value function of the d variables $x = [x_1, \ldots, x_d]^T$, so can be viewed as a function of the form $u\colon \Omega \subset \mathbb{R}^d \to \mathbb{R}$. The function f appearing on the right in (2.34) is a *forcing function*, and is a function of the same form as u, so that $f\colon \Omega \subset \mathbb{R}^d \to \mathbb{R}$. The forcing function represents the particular mathematical model, and drives the behavior of the solution u. The first equation (2.34) is the *PDE*, and the second equation (2.35) is the *boundary condition*. The final symbol appearing in (2.34) that we need to define is the *Laplacean operator* ∇; it is just the multi-dimensional analogue of the second derivative appearing in our BVP in ODE (2.4) above:

$$\Delta u = \sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2}. \tag{2.36}$$

If there is only one independent variable, so that $\Omega = (a, b) \subset \mathbb{R}$, then the Laplacean is reduced to simply the second derivative in that one variable:

$$\Delta u = \frac{d^2 u}{dx^2} = u'', \tag{2.37}$$

and the Poisson equation (2.34)–(2.35) is exactly our BVP in ODE above (2.1)–(2.3).

A finite difference approximation of the multi-dimensional case is developed exactly as for the one-dimensional case; for example, if there are now two independent variables x and y, and the domain $\Omega = [a, b] \times [c, d] \subset \mathbb{R}^2$, then we would begin by putting down a discrete mesh of points in both independent variables:

$$a = x_0 < x_1 < x_2 < \ldots < x_n < x_{n+1} = b, \qquad x_i = a + ih_x, \qquad h_x = \frac{b-a}{n_x + 1}, \tag{2.38}$$

$$c = y_0 < y_1 < y_2 < \ldots < y_n < y_{n+1} = d, \qquad y_i = a + ih_y, \qquad h_y = \frac{d-c}{n_y + 1}, \tag{2.39}$$

where n_x is the number of points in the x variable, and $n_y$is the the number of points in the y variable, giving the mesh spacings h_x and h_y above. A *uniform mesh* would take $h_x = h_y = h$. Difference approximations are developed in the same way; for example, a centered difference approximation to Δu in the case of two independent variables is:

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} + O(h^2), \tag{2.40}$$

where we have used the natural extension to our earlier notation:

$$u_{i,j} = u(x_i, y_j). \tag{2.41}$$

The accuracy of the approximation (and verification that it does actually approximate Δu) proceeds exactly as above; one performs Taylor expansion in each variable independently:

$$u(x_i + h, y_j) = u(x_i, y_j) + \frac{\partial u(x_i, y_j)}{\partial x} h + \cdots \tag{2.42}$$

$$u(x_i, y_j + h) = u(x_i, y_j) + \frac{\partial u(x_i, y_j)}{\partial y} h + \cdots \tag{2.43}$$

and then simplifies as before. If we again drop the $O(h^2)$ error term in (2.40) and use it as a basis for building a system of equations for an approximation

$$u_{i,j} \approx u(x_i, y_j), \tag{2.44}$$

then we again produce a linear system of n equations in n unknowns:

$$-\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} - \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} = f_{i,j}, \qquad i,j = 1,\ldots,n, \tag{2.45}$$

or more simply:

$$4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1} = h^2 f_{i,j}, \qquad i,j = 1,\ldots,n. \tag{2.46}$$

This is again a linear system of the form (2.31). Unlike the one-dimensional case, there are now various options for ordering the unknowns in the vector U; what is known as the *natural ordering* varies i from 1 to n first, then increment j, then again vary i from 1 to n, until all unknowns $u_{i,j}$ are written as a single array U, which also determines the ordering of the right hand side vector F:

$$U = \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{n,1} \\ --- \\ u_{1,2} \\ u_{2,2} \\ \vdots \\ u_{n,2} \\ --- \\ \vdots \\ --- \\ u_{1,n} \\ u_{2,n} \\ \vdots \\ u_{n,n} \end{bmatrix}, \qquad F = \begin{bmatrix} f_{1,1} \\ f_{2,1} \\ \vdots \\ f_{n,1} \\ --- \\ f_{1,2} \\ f_{2,2} \\ \vdots \\ f_{n,2} \\ --- \\ \vdots \\ --- \\ f_{1,n} \\ f_{2,n} \\ \vdots \\ f_{n,n} \end{bmatrix}. \tag{2.47}$$

This then fixes the order of the equations appearing in the matrix A. The boundary conditions are incorporated into F in exactly the same way as in the one-dimensional case. While the matrix A produced in this was does not have the simple tri-diagonal structure as in the one-dimensional case, it does have a *block* tri-diagonal structure. One solves the linear system (2.31) as before, using e.g. the builtin MATLAB solver, or some other algorithm.

2.2 Explicit and Implicit Methods for Parabolic Equations

So far, we have developed finite difference methods for the BVP in ODE (2.1)–(2.3) and also the BVP in PDE (2.34)–(2.35). However the Black-Scholes problem took the form of a Parabolic partial differential equation, with one independent variable representing space (or the underlying asset price S), and a second independent variable representing time (the lifetime of the option, taken to be in the range $[0, T]$). The final form of the Black-Scholes problem we derived earlier,

together with the boundary conditions on a finite domain and an initial condition in the case of European options, was:

$$u_t = u_{xx}, \qquad x \in (a,b), \quad t \in (0,T]. \tag{2.48}$$

$$u(0,x) = u_0(x), \qquad x \in (a,b), \quad t = 0, \tag{2.49}$$

$$u(t,x) = u_D(t,x), \qquad x = a \text{ or } x = b, \quad t \in (0,T], \tag{2.50}$$

where again we use the shorthand notation

$$u = u(t,x), \qquad u_t = \frac{\partial u(t,x)}{\partial t}, \qquad u_{xx} = \frac{\partial^2 u(t,x)}{\partial x^2}. \tag{2.51}$$

The first equation (2.48) is the PDE, which is just the Black-Scholes equation after the change of variables outlined earlier. The second equation (2.49) is the initial condition that we obtained for European options (by reversing the time variable and taking the final condition as an initial condition), and the third equation (2.50) is the boundary condition we obtained for the Black-Scholes equation by approximating the unbounded interval $-\infty < x < \infty$ by a bounded interval $a \le x \le b$.

Our goal is then to build a finite difference method for solving this Initial Boundary Value Problem (IBVP) in PDE. If we put down a mesh of points in the x and t directions,

$$x_i = a + ih, \qquad i = 0, \ldots, n+1, \qquad h = \frac{b-a}{n+1}, \tag{2.52}$$

$$t_j = 0 + jk, \qquad j = 0, \ldots, m+1, \qquad k = \frac{T-0}{m+1}, \tag{2.53}$$

we can use the simple notation:

$$u_i^j = u(t_j, x_i), \qquad i = 0, \ldots, n+1, \qquad j = 0, \ldots, m+1, \tag{2.54}$$

where we use the superscript in the special case of the time variable. (Note that this notation easily allows for more than one spatial variable, which we would handle with additional subscripts.) Our approach will be to use the first-order difference approximation we developed earlier (2.6) to approximate u_t, and then to use the second order difference approximation we saw earlier (2.18) to approximate u_{xx}. Putting these two approximations together for the left- and right-hand sides of (2.48) result in:

$$\frac{u_i^{j+1} - u_i^j}{k} = \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2}, \qquad i = 1, \ldots, n, \qquad j = 1, \ldots, m. \tag{2.55}$$

The value at the new time point

$$u_i^{j+1} \approx u(t_{j+1}, x_i) \tag{2.56}$$

is then determined by placing all of the other quantities on the right-hand-side, giving what is known as the *Forward Euler Method*:

$$u_i^{j+1} = u_i^j + \frac{k}{h^2}[u_{i+1}^j - 2u_i^j + u_{i-1}^j], \qquad i = 1, \ldots, n, \qquad j = 1, \ldots, m. \tag{2.57}$$

Notice however that we could have chosen to evaluate the terms on the right at the new time point (2.56), which gives:

$$\frac{u_i^{j+1} - u_i^j}{k} = \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{h^2}, \qquad i = 1, \ldots, n, \qquad j = 1, \ldots, m. \tag{2.58}$$

Determining the u_i^{j+1} now involves solving a system of equations, which after rearranging looks like:

$$u_i^{j+1} - \frac{k}{h^2}[u_{i+1}^j - 2u_i^j + u_{i-1}^j] = u_i^j, \qquad i = 1, \ldots, n, \qquad j = 1, \ldots, m. \tag{2.59}$$

This is known as the *Backward Euler Method*. If we are willing to solve equations, then in fact we could consider some type of average of the right-hand side at the two time points; this gives a whole collection of methods known as *theta methods*:

$$\frac{u_i^{j+1} - u_i^j}{k} = \theta \left[\frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{h^2} \right] + (1-\theta) \left[\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2} \right], \tag{2.60}$$

where $\theta \in [0, 1]$. If $\theta = 0$ then we get the Forward Euler Method, whereas if $\theta = 1$ then we get the Backward Euler Method. A very important case is $\theta = \frac{1}{2}$, which gives the *Crank-Nicholson Method*:

$$\frac{u_i^{j+1} - u_i^j}{k} = \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{2h^2} + \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{2h^2}. \tag{2.61}$$

Placing the unknowns on the left and the known quantities on the right, we obtain:

$$u_i^{j+1} - \frac{k}{2h^2}[u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}] = u_i^j - \frac{k}{2h^2}[u_{i+1}^j - 2u_i^j + u_{i-1}^j]. \tag{2.62}$$

Note that the matrix A that arose earlier has again appeared. In particular, if we define A and U^k as follows:

$$A = \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & \end{bmatrix}, \qquad U^k = \begin{bmatrix} u_1^k \\ u_2^k \\ \vdots \\ u_{n-1}^k \\ u_n^k \end{bmatrix}, \tag{2.63}$$

then the Forward Euler, Backward Euler, and Crank-Nicholson methods can be written in a simple matrix form as follows:

$$U^{k+1} = \left[I + \frac{k}{h^2} A \right] U^k, \qquad \text{(Forward Euler)} \tag{2.64}$$

$$\left[I - \frac{k}{h^2} A \right] U^{k+1} = U^k, \qquad \text{(Backward Euler)} \tag{2.65}$$

$$\left[I - \frac{k}{2h^2} A \right] U^{k+1} = \left[I + \frac{k}{2h^2} A \right] U^k, \qquad \text{(Crank-Nicholson)} \ . \tag{2.66}$$

For Forward Euler, we just do a simple matrix-vector multiplication and a vector addition to produce the new time approximation U^{k+1} from the previous time approximation U^k. In the case of both Backward Euler and Crank-Nicholson, we must solve a linear system of equations involving the matrix

$$B = \left[I - \frac{k}{\beta h^2} A \right], \tag{2.67}$$

with $\beta = 1$ for Backward Euler, or $\beta = 2$ for Crank-Nicholson, and with the two different right-hand sides as indicated above.

2.3 Stability, Consistency, Convergence

(Some notes on stability, consistency, and convergence of the three main methods derived above will be added here.)

2.4 American Options as a Free Boundary Problem

Our discussions above have been for problems that have fixed boundary conditions and a fixed initial condition. As we have seen, this is the appropriate setting for European options. However, we are also interested in the case of American options, which give rise to what are known as *free boundary problems*.

(We will make some additional comments here about handling free boundary problems.)

2.5 Software Packages for using Finite Difference Methods

(We will make some comments here about standard software packages for solving IBVP in PDE, primarily Parabolic equations.)

References

[1] W. E. Boyce and R. C. DiPrima. *Elementary Differential Equations and Boundary Value Problems*. John Wiley & Sons, Inc., New York, NY, fourth edition, 1977.

[2] R. Seydel. *Tools for Computational Finance*. Springer-Verlag, New York, NY, fifth edition, 2012.

[3] I. Stakgold and M. Holst. *Green's Functions and Boundary Value Problems*. John Wiley & Sons, Inc., New York, NY, third edition, 2011.

[4] P. Wilmott, S. Howison, and J. Dewynne. *The Mathematics of Financial Derivatives*. Cambridge University Press, New York, NY, 2009.