Stochastic Calculus, Courant Institute, Fall 2020
http://www.math.nyu.edu/faculty/goodman/teaching/StochCalc2020/index.html

Week 3

Jonathan Goodman, November, 2020

1 Introduction value functions

There is an important two way relation between diffusion processes and partial differential equations. In one direction, we learn about diffusion processes by solving some associated diffusion equations, which are partial differential equations. In the other direction, we find solutions of diffusion equations by expressing the solution as the expected value of some quantity related to a diffusion process. This allows us to find the value by simulation.

This class explores the relation between diffusion equations and diffusion processes in the special case in which the diffusion process is Brownian motion and the diffusion equation is a variant of the heat equation. Week 4 describes this relationship for general diffusions and diffusion equations. The fundamental ideas are the same, but general diffusions are able to model a bigger range of systems.

In this class, X_t will be Brownian motion. The *cumulative normal* distribution function is

$$\begin{aligned} N(x) &= \Pr(Z < x) \ , \quad Z \sim \mathcal{N}(0,1) \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{z^2}{2}} \, dz \ . \end{aligned}$$

This has the values

$$\begin{aligned} N(x) &\to 0 \text{ as } x \to -\infty \\ N(x) &\to 1 \text{ as } x \to +\infty \\ N(0) &= \frac{1}{2} \ . \end{aligned}$$

2 Deriving backward equations

Let X_t be Brownian motion. A simple value function is

$$f(x,t) = \mathrm{E}[\, V(X_T) \mid X_t = x] \ . \tag{1}$$

This is defined for any $t \le T$. This function satisfies the *backward heat equation*

$$\partial_t f + \frac{1}{2}\partial_x^2 f = 0 \ . \tag{2}$$

These equations are supplemented with *final conditions*

$$f(x,T) = V(x) \ . \tag{3}$$

The *final time* is the largest t where f is defined, which is $t = T$. The final conditions specify f at the final time. The PDE (partial differential equation) (2) may not be obvious, but the final conditions are obvious. If you put $t = T$ in the value function definition (1), you find yourself asking what is the expected value of $V(X_T)$ conditional on $X_T = x$.

We give two derivations of the backward equation eqrefbes. The first one uses specific calculations with the *fundamental solution* (also called *heat kernel* or *Green's function*). People often don't like this derivation because it gives no insight as to why f satisfies the PDE or the significance of the PDE. Another drawback is that you have to know the fundamental solution before you find the PDE. This is possible for Brownian motion and a few other specific examples, but it is not possible for most diffusions. The second derivation may be applied to any diffusion process. We will see in Week 4 that it gives the backward equation in terms of the infinitesimal mean and infinitesimal variance of the diffusion process.

The first derivation starts with an expression of $f(x,t)$ as an integral of the *final data*, V, This integral expresses $f(x,t)$ as the expected value (1) when X_t is Gaussian with mean x and variance $T-t$. This is the conditional distribution of X_T after a time increment of length $T-t$ starting at $X_t = x$. We use y to represent a possible value of X_T, so the conditional density is

$$X_T \sim \mathcal{N}(x, T-t) = G(y,x,T-t) = \frac{1}{\sqrt{2\pi(T-t)}} e^{-\frac{(y-x)^2}{2(T-t)}} \ . \tag{4}$$

Therefore, the expected value (1) is

$$f(x,t) = \mathrm{E}[\, V(X_T) \mid X_t = x] = \int_{-\infty}^{\infty} V(y) \frac{1}{\sqrt{2\pi(T-t)}} e^{-\frac{(y-x)^2}{2(T-t)}} \, dy \ . \tag{5}$$

This integral formula for the value function allows us to derive the PDE (2), and it tells us some things about what kind of function the value function can be.

We verify that f satisfies the backward equation (2) by calculating the partial derivatives of the integral representation (5). If you differentiate with respect to x or with respect to t, the derivatives go inside the integral and then onto the fundamental solution (4). That is,

$$\partial_t f(x,t) = \partial_t \int V(y) G(y,x,T-t) \, dy = \int V(y) [\, \partial_t G(y,x,T-t)] \, dy \ ,$$

and

$$\partial_x^2 f(x,t) = \int V(y) \left[\, \partial_x^2 G(y,x,T-t)\right] \, dy \ ,$$

We add these together and get

$$\partial_t f + \frac{1}{2}\partial_x^2 f = \int V(y)\left[\partial_t G(y,x,T-t) + \frac{1}{2}\partial_x^2 G(y,x,T-t)\right]\,dy\;.$$

We can calculate the combination of derivatives in $[\cdots]$ by differentiating the formula (4). Here is the time derivative calculation. You have to do this slowly and carefully to get the signs and the powers of $(T-t)$ right.

$$\begin{aligned}
\partial_t\left[(2\pi(T-t))^{-\frac{1}{2}}\,e^{-\frac{(y-x)^2}{2(T-t)}}\right] &= \left[\frac{1}{\sqrt{2\pi}}\partial_t (T-t)^{-\frac{1}{2}}\right]e^{-\frac{(y-x)^2}{2(T-t)}} + \frac{1}{\sqrt{2\pi(T-t)}}\left[\partial_t e^{-\frac{(y-x)^2}{2(T-t)}}\right] \\
&= \frac{1}{2}\frac{1}{\sqrt{2\pi}}(T-t)^{-\frac{3}{2}}e^{-\frac{(y-x)^2}{2(T-t)}} - \frac{1}{\sqrt{2\pi(T-t)}}\frac{(y-x)^2}{2(T-t)^2}e^{-\frac{(y-x)^2}{2(T-t)}} \\
&= \frac{1}{2}\frac{1}{\sqrt{2\pi}}\left[\frac{1}{(T-t)^{\frac{3}{3}}} - \frac{(y-x)^2}{(T-t)^{\frac{5}{2}}}\right]e^{-\frac{(y-x)^2}{2(T-t)}}\;.
\end{aligned}$$

We have to do two ∂_x calculations, but they are not as tricky. First

$$\begin{aligned}
\partial_x G(y,x,T-t) &= \frac{1}{\sqrt{2\pi(T-t)}}\frac{y-x}{T-t}e^{-\frac{(y-x)^2}{2(T-t)}} \\
&= \frac{1}{\sqrt{2\pi}}\frac{y-x}{(T-t)^{\frac{3}{2}}}e^{-\frac{(y-x)^2}{2(T-t)}}\;.
\end{aligned}$$

Then

$$\partial_x^2 G(y,x,T-t) = \frac{1}{\sqrt{2\pi}}\left[\frac{-1}{(T-t)^{\frac{3}{2}}}e^{-\frac{(y-x)^2}{2(T-t)}} + \frac{(y-x)^2}{(T-t)^{\frac{5}{2}}}e^{-\frac{(y-x)^2}{2(T-t)}}\right]\;.$$

Now, compare the two results and you see that the $(T-t)^{-\frac{3}{2}}$ terms and the $(y-x)^2(T-t)^{-\frac{5}{2}}$ terms cancel in $\partial_t G + \frac{1}{2}\partial_x^2 G$. The calculation

$$\partial_t G(y,x,T-t) + \frac{1}{2}\partial_x^2 G(y,x,T-t) = 0$$

implies that the value function satisfies the backward equation (2). Section 3 gives some uses of the integral formula (5).

The second derivation uses "local information", the infinitesimal mean and variance of Brownian motion. The derivation looks at the terms in Ito's lemma and asks what equation f would have to satisfy to that

$$Y_t = f(X_t, t)$$

is a martingale. Once Y_t is a martingale,

$$\mathrm{E}\big[\,Y_T \mid X_{[0,t]}\big] = Y_t\;.$$

In other notation, this is

$$\mathrm{E}[\,f(X_T) \mid X_t = x] = f(x,t)\;.$$

If $f(x,T)$ has final values $f(x,T) = V(x)$, we have the desired expected value formula (1).

The first second derivation uses the fact that if you watch the value function change with time, you see a martingale. For our process, Ito's lemma gives

$$df(X_t,,t) = \partial_x f(X_t,t)dX_t + \left[\partial_t f(X_t,t) + \frac{1}{2}\partial_x^2 f(X_t,t)\right] dt$$

If the term in square brackets is zero, $[\cdots] = 0$, then

$$dY_t = df(X_t,t) = \partial_x f(X_t,t)\, dX_t \ .$$

This implies that Y_t is a martingale, because any Ito integral with respect to Brownian motion is a martingale. That's one of the Doob martingale theorems.

$$Y_{T_2} - Y_{T_1} = f(X_{T_2},T_2) - f(X_{T_1},T_1) = \int_{T_1}^{T_2} \partial_x f(X_t,t)dX_t \ .$$

The term in brackets is zero exactly with the backward equation (2) is satisfied.

This derivation may seem mysterious, but it is simple and powerful. We will use derivations like this to find backward equations for other situations.

3 Digital options, smoothing

A *digital* option is one that pays all or nothing depending on some criterion. A digital payout would be

$$V(x) = \begin{cases} 1 \text{ if } x > x_0 \\ 0 \text{ if } x \leq x_0 \end{cases} \ .$$

Corresponding to this is the value function (1). In this case, the value function may be written as a probabability

$$f(x,t) = \Pr(\, X_T > x_0 \mid X_t = x) \ .$$

In fact, the expected value of any $0,1$ function (a function that takes values $V = 0$ or $V = 1$ only) is the probability that the value is 1.

The value function may be expressed in terms of the cumulative normal distribution function. One way to derive the formula uses the fact that, conditional on $X_t = x$, the final position is $X_T \sim \mathcal{N}(x, T-t)$. You can represent such a random variable $Y \sim \mathcal{N}(x, T-t)$ in terms of the standard normal $Z \sim \mathcal{N}(0,1)$ as

$$Y = x + \sqrt{T-t}\, Z \ .$$

This is Gaussian with mean x and variance $T-t$. The condition $X_T > x_0$ has the same probability as $Y > x_0$ (because X_T and Y have the same distribution).

Therefore

$$\begin{aligned} f(x,t) &= \Pr(Y > x_0) \\ &= \Pr\Big(x + \sqrt{T-t}\, Z > x_0 \Big) \\ &= \Pr\left(Z > \frac{x_0 - x}{\sqrt{T-t}} \right) \\ &= 1 - \Pr\left(Z < \frac{x_0 - x}{\sqrt{T-t}} \right) \\ f(x,t) &= 1 - N\left(\frac{x_0 - x}{\sqrt{T-t}} \right) \;. \end{aligned} \tag{6}$$

This has the feature than $f(x,t) \to 0$ as $x \to -\infty$ and $f(x,t) \to 1$ as $x \to \infty$. This is clear from the definition of f, and you can see it in the solution formula. Write

$$z = \frac{x_0 - x}{\sqrt{T-t}} \;.$$

Then $f(x,t) = 1 = N(z)$. For example, we see that $z \to -\infty$ as $x \to \infty$, so $1 - N(z) \to 1 - 1 = 0$. The solution formula (6) implies that for a fixed t, f makes a transition from 0 to 1 as x goes from $-\infty$ to ∞.

The specific formula (6) tells us that the transition from $f \approx 0$ to $f \approx 1$ happens quickly with t is close to T. The "length scale" of the transition is $\sqrt{T-t}$. This means that when x goes from $x_0 - \sqrt{T-t}$ to $x_0 + \sqrt{T-t}$, the value function $f(x,t)$ goes from a value close to zero to a value close to 1. We say that the solution of the backward equation is "smoothing". The sharp discontinuity is the final condition is smoothed into a rapid but smooth transition.

4 Quadratic exponential and the ansatz method

Suppose the payout function is a *quadratic exponential*

$$V(x) = e^{-rx^2} \;.$$

This is called "quadratic exponential" rather than "Gaussian" because it is not a probability density. Still, everything related to Brownian motion seems to turn Gaussians into Gaussians. Therefore, we *guess* that the value function has the form

$$f(x,t) = A(t) e^{-s(t)x^2} \;. \tag{7}$$

A mathematical guess like this is called an *ansatz* (German word that means this). You guess the form and then see whether you can find formulas for $A(t)$ and $s(t)$ so that the ansatz (7) satisfies the backward equation (2) and the final condition. The ansatz "method" is to make an ansatz like (7) and then show it works. It's hard to call it a method because it's really just a guess. Experienced people may be led to specific guesses in specific ways, but even for them it's guessing. The final condition is easy, it gives final conditions for A and s, which are

$$A(T) = 1 \ , \quad s(T) = r \ . \tag{8}$$

The ansatz method requires you to put the ansatz (7) into the backward equation (2) and see what this says about A and s. We use a dot for time derivatives, so $\dot{q}(t) = \frac{d}{dt}q(t)$.

$$\partial_t f = \dot{A}(t)e^{-s(t)x^2} - \dot{s}x^2 A(t)e^{-s(t)x^2} \ .$$

Then

$$\partial_x f = -2s(t)xA(t)e^{-s(t)x^2} \ ,$$

and

$$\partial_x^2 f = -2s(t)Ae^{-s(t)x^2} + 4s(t)^2 A(t)x^2 e^{-s(t)x^2} \ .$$

You put this into the backward equation and find

$$\dot{A}(t)e^{-s(t)x^2} - \dot{s}x^2 A(t)e^{-s(t)x^2} + \frac{1}{2}\left[-2s(t)Ae^{-s(t)x^2} + 4s(t)^2 A(t)x^2 e^{-s(t)x^2}\right] = 0 \ .$$

The exponential factor $e^{-s(t)x^2}$ appears in every term and may be cancelled. The rest may be re-arranged to the form

$$x^2\left[-\dot{s}(t)A + 2s(t)^2 A\right] + \left[\dot{A}(t) - s(t)A(t)\right] = 0 \ .$$

The quantities in square brackets are functions of t alone. Therefore, the expression on the left is a quadratic function of x for each fixed t. A polynomial that is equal to zero, as this one is, must have all coefficients equal to zero. This gives two equations

$$\dot{s}(t) = 2s(t)^2 \tag{9}$$

$$\dot{A}(t) = s(t)A(t) \ . \tag{10}$$

It is "easy" to solve these differential equations with the final conditions given. Exercise 1 asks you to do the algebra and interpret the results.

The ansatz method is used in quantitative finance in several places. There are *affine* interest rate models in which the exponent is a linear function of the x variable with a time dependent coefficient and pre-factor.

5 Hitting probabilities and boundary conditions

A *hitting time* is the first time a stochastic process X_t "hits" a specific value or satisfies a given condition. There are hitting time problems in finance that come from contracts with conditions that depend on stochastic market prices. Among these are *knock-out options*, that pay nothing if the price ever exceeds a specified knock-out price.

Suppose X_t is a Brownian motion with $X_0 = x_0$ in the range $a \leq x_0 \leq b$. Suppose you get a payout $V(X_T)$ if $a \leq X_t \leq b$ for all t in the range $0 \leq t \leq T$. Otherwise, you get zero. The value function for this payout satisfies the backward equation (2) if $a < x < b$, but clearly $f = 0$ if $x = a$ or $x = b$. These are *absorbing boundary conditions* (because the Brownian motion is "absorbed" and stopped if it ever touches a boundary point). They are also called *Dirichlet* boundary conditions.

6 Running payouts

A *running payout* is a payout that you get continuously in time rather than just at the final time. A running payout might take the form

$$Y = \int_0^T V(X_t)\, dt \; .$$

A value function approach to this uses a value function that only "sees" the coming reward after time t, not the reward that has arrived (accrued, in financial language) so far. That is

$$f(x,t) = \mathrm{E}\left[\int_t^T V(X_s)\, ds \mid X_t = x \right] \; . \tag{11}$$

An Ito's lemma derivation of a backward equation uses the observation that when time goes from t to $t + dt$, the integral decreases by $V(x)dt$. Therefore

$$\mathrm{E}[\, df(X_t,t)] = -V(X_t)dt \; .$$

The Ito calculation from before (look at the quantity in square braces) implies that

$$\partial_t f(x,t) + \frac{1}{2}\partial_x^2 f(x,t) = -V(x) \; . \tag{12}$$

Of course, the final condition is $f(x,T) = 0$ because the payout stops at the final time T.

7 Finite difference methods

Finite difference methods are numerical algorithms for solving (approximately) PDEs. They apply to a vast range of PDEs of all types and from all fields. This section describes some finite difference methods for solving the backward equation. The derivation uses the convergence of random walk to Brownian motion (Week 1). The finite difference approximation is the backward equation that the random walk satisfies. There are ways to derive these and other finite difference methods that do not rely on probability.

Consider the value function that satisfies the simple backward equation (2). We often call x the *space variable* and t *time variable*. We consider a random walk approximation to Brownian motion. There is a *space step* Δx and space *grid points* $x_j = j\Delta x$. There is a *time step* Δt and discrete times $t_k = k\Delta t$. The random walk (notation from Week 1) has $X^{\Delta t}_{t_k} = x_j$ for some integer j. In one step, the walk can go left, or not move, or move right. The probabilities to move left, right, or not move are a, c, and b respectively.

$$X^{\Delta t}_{t_{k+1}} = \begin{cases} X^{\Delta t}_{t_k} - \Delta x & \text{with probability } a \\ X^{\Delta t}_{t_k} & \text{with probability } b \\ X^{\Delta t}_{t_k} + \Delta x & \text{with probability } c \end{cases} \tag{13}$$

The discrete value function will be called F. [Be careful when writing by hand to make the continuous value function f look different than the discrete value function F.] Assume that the final time T is one of the discrete times. There is an n with $T = t_n$. We may have to adjust Δt to make this happen. The values of F are

$$F_{kj} = \mathrm{E}\big[\, V\big(X^{\Delta t}_{t_n}\big) \mid X^{\Delta t}_{t_k} = x_j \big] \ . \tag{14}$$

This is like the definition of the continuous value function (1), but applied to the random walk $X^{\Delta t}$ instead of the Brownian motion X.

We have to relate the probabilities a, b, and c to the space step Δx and the time step Δt. The relationship comes from the fact that the random walk increment in one time step should have the mean and variance of the Brownian motion increment over a time Δt, which is Δt. The expected value of the discrete increment should be zero:

$$\mathrm{E}\Big[\, X^{\Delta t}_{t_k} - X^{\Delta t}_{t_{k-1}} \Big] = 0 \ .$$

The possible values of the increment are $\pm\Delta x$ and 0, so we get

$$0 = a\,(-\Delta x) + b\,(0) + c\,(\Delta x) \ .$$

This gives

$$a = c \ .$$

The random walk is symmetric. The variance calculation is similar

$$\Delta t = a\,(\Delta x^2) + b(0) + c(\Delta x^2) = 2a\Delta x^2 \ .$$

This leads to the *CFL* ratio formula

$$a = \frac{1}{2}\,\frac{\Delta t}{\Delta x^2} \ . \tag{15}$$

CFL is for the mathematicians Richard Courant (founder of the Courant Institute), Kurt Friedrichs (one of its first faculty) and Hans Lewy. Their 1928 paper laid the foundations for finite difference solution of PDEs. The fraction is the *CFL ratio*

$$\lambda = \frac{\Delta t}{\Delta x^2} \ . \tag{16}$$

The coefficients have to add up to one because they are probabilities. This leads to a formula for b

$$\begin{aligned}
a + b + c &= 1 \\
\frac{1}{2}\lambda + b + \frac{1}{2}\lambda &= 1 \\
b = 1 - \lambda &= 1 - \frac{\Delta t}{\Delta x^2} \; .
\end{aligned} \tag{17}$$

The fact that $b \geq 0$ implies that

$$1 - \frac{\Delta t}{\Delta x^2} \geq 0 \; .$$

This may be written as

$$\lambda = \frac{\Delta t}{\Delta x^2} \leq 1 \; . \tag{18}$$

This is the famous CFL *stability limit*. People usually want a large CFL number λ so that fewer time steps are required. The formulas (19) with (15) and (17) make sense even if $\lambda > 1$. But the code will "blow up" if you do.

The code `FiniteDifference.py` uses these formulas. The number of grid points in space, n, is specified, along with the length of the interval, L. This determines the space step Δx. The CFL ratio λ is used to find Δt. This time step is then adjusted down slightly so that T is an integer number of time steps from $t = 0$, which is $T = n_t \Delta t$. Most of the work of the code is the time step calculation (19).

The discrete value function satisfies a discrete recursion relation relation. The expected values $F_{k-1,j}$ may be computed from the values F_{kj} using the fact that if $X^{\Delta t}_{t_{k-1}} = x_j$, then $X^{\Delta t}_{t_k}$ is one of the values $x_j - \Delta t = x_{j-1}$ or x_j or $x_j + \Delta t = x_{j+1}$, and the probabilities are a, b, and c. The calculations we're about to do simplify because $X^{\Delta t}$ is a Markov process (definition in Week 1). This implies that, for example, that if $X^{\Delta t}$ steps from x_{j-1} to x_j, then the expected value going forward from x_j doesn't depend on the fact that it came from x_{j-1}. In formulas, this is

$$\mathrm{E}\Big[\, V\big(X^{\Delta t}_{t_n}\big) \mid X^{\Delta t}_{t_{k-1}} = x_j \text{ and } X^{\Delta t}_{t_k} = x_{j-1} \Big] = \mathrm{E}\big[\, V\big(X^{\Delta t}_{t_n}\big) \mid X^{\Delta t}_{t_k} = x_j \big]$$

The conditional expectation calculation using these ideas is

$$\begin{aligned}
F_{k-1,j} &= \mathrm{E}\Big[\, V\big(X^{\Delta t}_{t_n}\big) \mid X^{\Delta t}_{t_{k-1}} = x_j \Big] \\
&= a\,\mathrm{E}\Big[\, V\big(X^{\Delta t}_{t_n}\big) \mid X^{\Delta t}_{t_{k-1}} = x_{j-1} \Big] \\
&\quad + b\,\mathrm{E}\Big[\, V\big(X^{\Delta t}_{t_n}\big) \mid X^{\Delta t}_{t_{k-1}} = x_j \Big] \\
&\quad + c\,\mathrm{E}\Big[\, V\big(X^{\Delta t}_{t_n}\big) \mid X^{\Delta t}_{t_{k-1}} = x_{j+1} \Big] \\
F_{k-1,j} &= a\,F_{k,j-1} + b\,F_{k,j} + c\,F_{k,j+1} \; .
\end{aligned} \tag{19}$$

This calculation starts with given final values $F_{n_t,j} = V(x_j)$. Then it loops over k doing *time steps* going backwards from n_t. Each time step is a loop over j. The boundary conditions in the code are that $f(0,t) = f(L,t) = 0$. This translates into $F_{k,0} = 0$ and $F_{k,n+1} = 0$. That leaves n "interior" grid points $x_1, \cdots, x_n$, which are separated by Δx. Therefore,

$$\Delta x = \frac{L}{n+1} \; .$$

The calculations (19) are done for $j = 1, \cdots, n$. In principle you don't have to store the boundary values because they are known and don't have to be computed. Storing them makes the code simpler. You can do the formula (19) for every j value without writing special code for the end cases $j = 1$ and $j = n$. Values used in this way are *ghost values*.

8 Exercises

1. Carry out the ansatz analysis of Section 4

   (a) Solve the differential equation (9). *Hint.* It may be written

   $$\frac{ds}{s^2} = 2dt \; .$$

   The integral of the left side is $-\frac{1}{s} + C$. The integral of the right side is $2t + C$. The constant is determined by the final condition $s(T) = r$. If you want the Wikipedia solution, it might help to know this is an example of a *Riccati* equation.

   (b) Solve the differential equation (10) and use the final condition to find a formula for the *prefactor* $A(t)$.

   (c) Is $s(t)$ an increasing or decreasing function of t? What does this say about the "width" of the payout and the width of the value function? Intuitively, why should one be wider than the other?

   (d) Is $A(t)$ increasing or decreasing? Why should the maximum of $f(x,t)$ for $t < T$ be larger/smaller (you pick) than the maximum of V?

   (e) Show that

   $$\frac{d}{dt} \int_{-\infty}^{\infty} f(x,t)\, dx = 0 \; .$$

   Are your formulas for s and A consistent with this? Could you derive the formula for A from this identity and the formula for s?

2. Suppose $x > 0$ and $t < T$. Define the *survival probability* starting from x between times t and T to be

   $$f(x,t,T) = \Pr(\, X_s > 0 \;\text{ for all } s \in [t,T]) \; .$$

This is the probability that the Brownian motion does not hit $x = 0$ at any time between t and T. We put in the dependence on the final time T to enable the calculations below. Consider the seemingly different problem with payout $V(x) = 1$ if $x > 0$ and $V(x) = -1$ if $x \leq 0$. The corresponding value function is

$$g(x, t, T) = \mathrm{E}[\, V(X_T) \mid X_t = x] \ .$$

This is defined for any x and $t \leq T$.

(a) Show that $g(-x, t, T) = g(x, t, T)$ for all x and $t < T$. Show that this implies that $g(0, t) = 0$ if $t < T$.

(b) Show that $g(x, T, T) = f(x, T, T)$ if $x > 0$. Assuming that the solution to the problem f satisfies is unique, show that $g(x, t, T) = f(x, t, T)$ if $x > 0$. (The finite difference approximation suggests that the solution is unique because it gives an algorithm for computing it. A course on PDE typically has a real mathematical proof.

(c) Define the hitting time to be the first time the Brownian motion touches the boundary, $x = 0$:

$$\tau = \min\{\, s \mid X_s = 0 \} \ .$$

Let $u(s)$ be the PDF of τ. Show that, conditional on $X_t = x$,

$$u(T) = -\partial_T f(x, t, T) \ .$$

(d) Find a formula for g in terms of the cumulative normal. This is similar to the formula in Section 3.

(e) Find a formula for $u(T)$. We used this formula in Exercise 6 of Week 1. This exercise fulfills the promise made there.

3. Download and run the posted code `FiniteDifference.py`. Check that the resulting plot matches the posted plot. Modify the code to compute the expected running payout function (11) with payout $V(x) = e^{-r(x-\frac{L}{2})^2}$. Choose $L = 10$, $r = 1$, and $T = 2$. Plot a series of computations in the same figure, as `FiniteDifference.py` does, to see how many grid points are needed to get an accurate solution. Explore the grid spacing needed for accurate solution when r is larger – in a qualitative way (larger r needs more/fewer/a lot more/a lot fewer points. Part of this exercise is to derive a finite difference method for the backward equation (12). You can do this by making a random walk approximation to the process and a corresponding finite sum approximation to the running payout.

4. Consider the digital option of Section refsec:d. This exercise asks you to *replicate* the option payout using a given initial *endowment* (amount of money) and a trading strategy on Brownian motion. The trading strategy

is a function $a(x,t)$. The initial endowment is a number g. The Brownian motion starts at $X_0 = 0$.

$$U_T = g + \int_0^T a(X_t, t) dX_t \ .$$

This random variable replicates the payout if $U_T = V(X_T)$. Find a way to replicate the digital payout in this way. *Hint.* Use Ito's lemma, the value function from Section 3, and the formula

$$\partial_x N(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \ .$$

5. Write a simulation code to verify the trading strategy of Exercise 4. Much of the code can be taken from Week 2. Choose a time step Δt and make the proper Ito approximation to the Ito integral of Exercise 4. Estimate the mean square replication error, which is

$$\mathrm{E}\Big[\left(U_T - V(X_T) \right)^2 \Big] \ .$$

This should decrease to zero as $\Delta t \to 0$. You need to make many paths to estimate the expected value accurately.