
§1. Ordinary Differential Equations

Let \( y = (y_1, \ldots, y_n)^T \) be a column vector in \( \mathbb{R}^n \).

Consider the ordinary differential equation,

\[
\frac{dy}{dt} = f(y, t), \quad \text{where } f \in \mathbb{R}.
\]

valued function of \( y \in \mathbb{R}^n \) and \( t \in \mathbb{R} \). We shall think of \( t \) as being the time variable. Now in general (1.1) has a unique solution if we specify the point at some time \( t_0 \), i.e., \( y(t_0) = y_0 \).

Example: Consider the system,

\[
\frac{dy_1}{dt} = y_2, \quad \frac{dy_2}{dt} = 3y_2 - 2y_1.
\]

We wish to find the unique solution of (1.2) with the given initial condition,

\[
y_1(0) = 1, \quad y_2(0) = 0.
\]

We can find an explicit solution by rewriting (1.2) as a scalar ODE,
\[ \frac{d^2 y}{dt^2} = 2 \frac{dy}{dt} - 2y. \]

We find a basis of solutions for (1.4) by substituting 
\[ y(t) = e^{\pm t} \] which is a solution of
\[ \frac{d^2 y}{dt^2} = 2 \frac{dy}{dt} - 2y. \]

(1.5) \( \frac{d^2 y}{dt^2} + 2y = 0 \Rightarrow \lambda = 1, 2 \), so the general solution of (1.4) is given by the formula
\[ y(t) = c_1 e^t + c_2 e^{2t}. \]

We now use the initial conditions (1.3) to uniquely determine \( c_1 \) and \( c_2 \) by

\[ c_1 + c_2 = 0, \quad c_1 + 2c_2 = 0 \Rightarrow c_1 = 2, \quad c_2 = -1, \]

so the unique solution of (1.2), (1.3) is given by

(1.6) \[ y(t) = 2e^t - e^{2t}. \]

In general we cannot produce explicit formulas for the solutions of (1.1). Hence we need to resort to numerical methods. Thus we discretize the time variable in increments \( \Delta t = t_k \). Thus time now takes values \( 0, t_k, 2t_k, t_k, \ldots \). We replace the derivative by a difference which approximates the derivative for small \( \Delta t \). There are several ways to do this,

(a) \text{Forward difference:} \quad \frac{dy}{dt}(t) \approx \frac{y(t + \Delta t) - y(t)}{\Delta t}
Example 2. \[ a(t) = 0, \quad x(t) = \frac{1}{2}a(t) \]

Consider the initial condition \( y(0) = 0, \) \( \dot{y}(0) = 0 . \) Then the solution to the differential equation is

\[ y(t) = \frac{1}{2} a(t) t^2 . \]
Then (1.11) becomes

\[ \frac{y^{m+1} - y^m}{\Delta t} = r y^m \Rightarrow y^{m+1} = \left[ 1 + r \Delta t \right] y^m \]

where we get the formula,

\[ y^m = \left[ 1 + r \Delta t \right] ^m. \]

We show that (1.14) converges to the exact solution of (1.12) as \( \Delta t \) goes to 0. Thus since \( y^m \approx y(\ln \Delta t) \) on putting \( T = m \Delta t \) we see that (1.14) gives

\[ \left[ 1 + \frac{r T}{m} \right]^m \approx y(T). \]

Letting \( \Delta t \to 0 \Rightarrow m \to \infty \) we have

\[ \lim_{m \to \infty} \left[ 1 + \frac{r T}{m} \right]^m = e^{r T}, \]

and

\[ y(T) = e^{r T} \text{ is the exact solution of (1.12)}. \]

If we use (1.6) to obtain a numerical method for solving (1.1) it is called Euler's backward method given by

\[ \frac{y^m - y^{m-1}}{\Delta t} = f(y^m, y^{m-1}). \]

Observe that in order to compute \( y^m \) in terms of \( y^{m-1} \) we need to solve an equation. For this reason the backwards method is called an implicit method. Hence it is in general more difficult to implement than the forward method. It has however important advantages including...
The forward method. To see this let us consider the backward scheme for (1.12) again. Thus

\[ y^m - y^{m-1} = \frac{\Delta t}{C} \Rightarrow y^m = \frac{y^{m-1}}{(1 - \frac{\Delta t}{C})}, \]

where we get the formula,

\[ y^m = \left[ 1 - \frac{\Delta t}{C} \right]^{-1}, \]

which again converges as \( \Delta t \to 0 \) to the exact solution \( y(t) = e^{Ct} \) of (1.12). Suppose now \( C \) is large and negative. Then in order for the forward Euler solution (1.13), (1.14) to begin to approximate the exact solution we must take \( \Delta t \) sufficiently small so that \( 1 + \frac{\Delta t}{C} > 0 \) i.e., \( \Delta t < 2\frac{1}{|C|} \). On the other hand, with backward Euler (1.18), (1.19) there is no critical value of \( \Delta t \) for which the numerical solution starts to behave badly.

In the application to PDE we will be considering a large system of equations, the simplest prototype of which looks like,

\[ \frac{dy}{dt} = -C_1 y_1, \quad \ldots, \quad \frac{dy}{dt} = -C_n y_n, \]

where \( C_1, \ldots, C_n > 0 \) and \( 0 < 2 C_n \Delta t \leq 2 \). Thus the condition of stability for forward Euler is now \( \Delta t < 2/C_n^2 \) which will lead to the time step \( \Delta t \) be very small if \( n \) is large.
We consider the errors in the numerical methods we have considered. Thus let \( T > 0 \) be a given time and \( \Delta t \) satisfy \( M \Delta t = T \) for some integer \( M \). We wish to estimate

\[
E_f(\Delta t) = y^M(\Delta t) - y^T(\Delta t), \quad \text{where } y^T(\Delta t) = \text{the exact solution at time } T. \quad \text{For forward and backward Euler we have } E_f(\Delta t) = \Delta t.
\]

To see this, we regard the exact solution to the differential equation as an approximate solution to the difference equation. Thus, for (1.11), we have

\[
y[(m+1)\Delta t] - y[m\Delta t] = \Delta t \left( y[m\Delta t], m \Delta t \right) + O[(\Delta t)^2].
\]

The error at each step—called the truncation error—is \( O[(\Delta t)^2] \). For \( M = T / \Delta t \) steps to get to \( T \) with \( M \), the error (1.21) should be

\[
E_f(\Delta t) \sim M O[(\Delta t)^2] = O(\Delta t).
\]

Thus a general principle:

**The error in solution = \( O(\Delta t) \) after \( M \) local truncation errors.**
Finally we consider a second order method — the Trapezoid method given by

\[ y^m = y^{m-1} + \frac{\Delta t}{2} \left[ f(y^m, m\Delta t) + f(y^{m-1}, (m-1)\Delta t) \right]. \]

Note that \( f(y, t) = f(t) \) depends only on \( t \). The solution of (1.1) is

\[ y(T) = y_0 + \int_0^T f(t) \, dt, \]

in which case the numerical method (1.24) is just the Trapezoid rule for computing the integral in (1.25). It is well known that for the Trapezoid rule \( E_T(\Delta t) = O(\Delta t^2) \). It is a second order method of integration and so is its generalization (1.24) for solving the equation (1.1). Note that (1.24) is also an implicit method.

For the example (1.12) the Trapezoid method gives

\[ y^m = y^{m-1} + \frac{\Delta t}{2} \left[ y^m + y^{m-1} \right], \]

where

\[ \left[ 1 - \frac{\Delta t}{2} \right] y^m = \left[ 1 + \frac{\Delta t}{2} \right] y^{m-1}, \]

so

\[ y^m = \left[ \frac{1 + \Delta t \sqrt{2}}{1 - \Delta t \sqrt{2}} \right]^m, \]

which evidently converges to the exact solution as \( \Delta t \to 0 \). Observe also that this is always stable if \( \gamma < 0 \) no matter how large \( \Delta t \) is, for

\[ \left| \frac{1 + \Delta t \sqrt{2}}{1 - \Delta t \sqrt{2}} \right| < 1, \quad \gamma < 0. \]
§ 2. Numerical Linear Algebra

We consider methods for numerically solving linear systems,

\[ \mathbf{Ax} = \mathbf{b}, \quad \text{where} \quad \mathbf{A} \text{ is an } n \times n \text{ matrix and } n \text{ is large.} \]

We assume also that \( \mathbf{A} \) is a sparse matrix with \( \mathbf{A} \) having \( O(n) \) non-zero entries, i.e., \( \mathbf{A} \) is a very sparse matrix.

We consider iterative methods as much more efficient ways of solving numerically.

(2.1) Thus, the usual Gaussian elimination procedure.

The basic idea is to rewrite (2.1) as a fixed point problem,

\[ \mathbf{u} = \mathbf{Bu} + \mathbf{c}. \]

If the matrix \( \mathbf{B} \) is small, then the iterative scheme

\[ \mathbf{u}^{k+1} = \mathbf{Bu}^{k} + \mathbf{c}, \quad k = 0, 1, 2, \ldots \]

where \( \mathbf{u}_0 \) is an arbitrary vector will converge on the solution.

We consider a particular iterative scheme known as the Gauss-Seidel algorithm. Thus we write

(2.4) \[ \mathbf{A} = \mathbf{D} - \mathbf{E} - \mathbf{F}, \]

where \( \mathbf{D} \) is the diagonal entries of \( \mathbf{A} \), \( \mathbf{E} \) the entries below the diagonal, and \( \mathbf{F} \) the entries above the diagonal, where

\[ \mathbf{A} = \begin{bmatrix} \mathbf{D} & -\mathbf{F} \\ -\mathbf{E} & \mathbf{D} \end{bmatrix} \]

Thus (2.1) can be rewritten

\[ \mathbf{D} \mathbf{u} = (\mathbf{E} + \mathbf{F}) \mathbf{u} + \mathbf{b}. \]

If \( \mathbf{D} \) is invertible, then we can solve
\[ B = D^{-1} (E + F), \quad c = D^{-1} b, \]

where the iteration scheme is

\[ D u_{k+1} = E u_k + F u_k + b, \]

This iteration is the Jacobi method. We can improve this somewhat by updating the last term on the RHS of (2.6) to yield

\[ D u_{k+1} = E u_k + F u_k + b, \]

where we have

\[ B = (D - E)^{-1} F, \quad c = (D - E)^{-1} b. \]

Note that the iteration matrix \( B \) is easily computable because \( D - E \) is lower triangular and hence easily inverted. The iteration (2.7) is called the Gauss-Seidel method.

This is a variation on the Gauss-Seidel method called the relaxation or SOR (successive over-relaxation) method. In this we introduce a parameter \( \omega \) called the relaxation parameter into the Gauss-Seidel method. To do this we rewrite (2.7) as

\[ (D - \omega E) u = \left[ (1 - \omega) D + \omega F \right] u + \omega b, \]

Thus we obtain an iterative scheme with

\[ B = (D - \omega E)^{-1} \left[ (1 - \omega) D + \omega F \right], \quad c = (D - \omega E)^{-1} b. \]

Evidently \( \omega = 1 \) corresponds to the Gauss-Seidel method. For the scheme to converge one needs \( 0 < \omega < 2 \). It turns out that for certain matrices \( A \), corresponding to discretizations of 2nd
The differential equations (2.11) are not uniquely determined for any \( w > 2 \).

To illustrate the Stüben scheme we consider the problem of solving the differential equation

\[
-2 \frac{d^2 u}{dx^2} = f(x), \quad 0 < x < 1.
\]

The solution of (2.11) is not uniquely determined since the homogeneous equation with \( f = 0 \) has a general solution \( u(x) = c_1 + c_2 x \) with arbitrary constants \( c_1, c_2 \). Hence to get a unique solution to (2.11) we must impose boundary conditions. The simplest such condition is a Dirichlet condition where we specify \( u(0) = a, \quad u(1) = b \), say

\[
(2.12) \quad u(0) = a, \quad u(1) = b.
\]

We can solve numerically (2.11), (2.12) if we first discretize the interval \( [0, 1] \) as \( 0, h, 2h, \ldots \), \( Nh = 2 \). Next we use the formula

\[
(2.13) \quad 2 \frac{u(x) - u(x+h) - u(x-h)}{h^2} = -\frac{d^2 u}{dx^2} + o(h^2),
\]

for any \( C^2 \) function \( u(x) \). Substituting \( u_n = u(nh) \), \( n = 0, 1, 2, \ldots, N \) obtain a discrete version of (2.11) as given by

\[
(2.14) \quad 2 \frac{u_n - u_{n+1} - u_{n-1}}{h^2} = f_n, \quad n = 1, \ldots, N,
\]

where \( f_n = f(nh) \). Setting \( u_0 = a \).
\[ u_{n+1} = 1 \]

After (2.14) is an \( m \times m \) system of equations which is invertible, if we solve

\[ (2.14) \]

exactly then one has that

\[ (2.15) \]

Thus this method of discretization of (2.11), (2.12) is second order accurate.

Consider now the various iteration schemes applied to solve (2.14). Let \( u_k^n \), \( n = 0, \ldots, N+1 \), be the \( k \) th iteration where \( u_0^n = a \), \( u_{N+1}^n = b \). Then the Jacobi iteration for (2.14) is given by

\[ (2.16) \]

\[ u_{n+1}^{k+1} = \frac{1}{2a} \left[ h^2 f_n + 2 \left( u_n^{k+1} + u_n^k \right) \right]. \]

For the initial \( u_0^n \) at \( k = 0 \) we can set simply \( u_0^n = 0 \). The lyapunov—dahlquist iteration is given by

\[ (2.17) \]

\[ u_{n+1}^{k+1} = \frac{1}{2a} \left[ h^2 f_n + 2 \left( u_n^{k+1} + u_n^k \right) \right]. \]

Note that we can perform the iteration (2.17) by starting with \( n = 1, 2, \ldots \) where the RHS of (2.17) is always predetermined. The lyapunov—dahlquist iteration requires somewhat more computation. We introduce an intermediate step,

\[ (2.18) \]

\[ y_{n+1}^k = \frac{1}{2a} \left[ h^2 f_n + 2 \left( u_{n-1}^{k+1} + u_{n+1}^k \right) \right], \]

so \( y_n^k \) is the lyapunov—dahlquist iteration. Then to compute (2.17) we additionally compute

\[ (2.19) \]

\[ u_{n+1}^{k+1} = u_n^k + \omega \left[ y_{n+1}^k - u_n^k \right]. \]
Another common form of boundary conditions are the Neumann boundary conditions, where we specify the derivatives of \( u(x) \) at \( x = 0, L \),

\[
\begin{align*}
\frac{du}{dx}(0) &= a, \\
\frac{du}{dx}(L) &= b.
\end{align*}
\]

If we set \( a = b = 0 \), then it is clear that \( u(x) \equiv 0 \) satisfies the homogeneous equation \((2.11)\),

where there is no uniqueness for solutions of \((2.11)\) even if a solution exists. We can remove

uniqueness by considering a slight perturbation of \((2.11)\),

\[
-2 \frac{d^2u}{dx^2} + u(x) = f(x), \quad 0 < x < L,
\]

with the boundary conditions \((2.20)\). Differentiating the 2nd derivative as before we get from \((2.22)\),

\[
2 \left[ \frac{u_{n+1} - u_{n-1}}{h^2} \right] + u_n = f_n.
\]

We consider how to impose the boundary conditions \((2.20)\). We shall take the

simple situation where

\[
\begin{align*}
a &= b = f'(0) = f'(1) = 0.
\end{align*}
\]

Thus

\[
\begin{align*}
0 &= \frac{du}{dx}(0) = \frac{d^2u}{dx^2}(0) = \frac{du}{dx}(1) = \frac{d^3u}{dx^3}(1).
\end{align*}
\]

Recall that for the Dirichlet b.c. we had \( n = 1 \), \( n = (2.14) \) is a \( 2 \times 2 \) system.

For the Neumann problem we take \( n = 0, n + 1 \),
\[ \text{i.e., on } (N+2) \times (N+2) \text{ system, where we set} \]
\[ (2.25) \quad u_1 = u_A, \quad u_{N+2} = u_B. \]

The reason is - Almost

\[ (2.26) \quad \frac{2u(0) - 2u(h)}{h^2} = -\frac{d^2u(0)}{dx^2} + O(h^2), \]

in view of (2.24). It follows that the discretization (2.22), (2.25) is second order accurate.

We consider the case of the convergence of the various iteration schemes - Jacobi, Gauss-Seidel and so on. Recall the basic iteration scheme (2.2).

This will converge provided that for some norm

\[ \|B\| < 1, \]

the matrix B satisfies \( \|B\| < 1 \). To see this we consider the error \( e_k \) given by

\[ (2.27) \quad e_k = u_k - u, \quad \text{where } u \text{ is the exact solution of (2.2) and } u_k \text{ is the } k \text{th iteration of (2.3).} \]

Thus

\[ (2.28) \quad e_{k+1} = Be_k, \quad \text{where} \]

\[ (2.29) \quad e_k = e_k^{(0)} \cdot \]

Using the fact that

\[ (2.30) \quad \|e_k\| = \|Be_k^{(0)}\| \leq \|B\|_k \|e_0\| \leq \|B\|_k \|e_0\|, \]

it follows that if \( \|B\|_k < 1 \), then

\[ (2.31) \quad \lim_{k \to \infty} e_k = 0. \]

Thus, for many possible norms for matrices
but a very simple one is as follows:

\[ (2.32) \quad \| B \|_1 = \| B \|_\infty = \max_{i=1}^{\infty} |b_{ij}| \]

If \( B \) is the \( n \times n \) matrix \( B = (b_{ij}) \), we compute this for the Jacobi-Armijo matrix for (2.23)

\[ (2.33) \quad \nu_n = \frac{2 \left[ \nu_{n+1} + \nu_{n-1} \right]}{2 + 1^2} \]

where we have

\[ (2.34) \quad \| B \|_\infty = 2.2 \log (2.2+1^2) < 1. \]

The iteration scheme therefore converges but if \( h \) is small the convergence rate will be slow.

The above considerations enable us to introduce a rule of thumb to guide us in designing an Armijo scheme for solving a differential equation.

**Rule of Thumb:** Let \( B \) be the Jacobi-Armijo matrix for the problem. Then use \( \| B \|_\infty \leq 1 \).

**Rule of Thumb:** Let \( B \) be the Jacobi-Armijo matrix for the problem. Then use \( \| B \|_\infty \leq 1 \).

A matrix of differential equations such that the homogeneous eqns. has the constant function as a solution. Then the term 1 of such one of \( B \) is \( 1 \). Hence we just need that all entries in \( B \) are non-negative.

Next we cannot differential equations on an infinite interval, say \([0, \infty)\). However, solve these equations numerically. The basic approach here
As in Exercise 2.1 we intend to extend the interval [0, a] to something finite, say [0, 1]. We must have to introduce a boundary condition at 0 and this will depend on the behavior of the solution we are looking for at large values. The simplest thing to do is to impose a Dirichlet b.c. at 1 which is consistent with this behavior. Consider for example the equation

\[ \frac{d^2 u}{dx^2} + u(x) = 0, \quad 0 < x < a, \quad u(0) = 2, \]

(2.36)

We can solve this problem exactly. The general solution of (2.35) is

\[ u(x) = c_1 e^x + c_2 e^{-x}, \]

(2.37)

There are clearly an infinite number of solutions to (2.35), (2.36) but the solution becomes unique once we specify the behavior of \( u(x) \) for large values of \( x \). Suppose for example we specify

\[ u(x) \approx \frac{1}{2} e^x, \quad x > 77 \]

(2.38)

Then the unique solution is

\[ u(x) = \frac{1}{2} e^x + \frac{3}{2} e^{-x}. \]

(2.39)

To find this solution numerically we can simply choose \( L \) to be large and set the Dirichlet b.c.

\[ u(L) = \frac{1}{2} e^L. \]

(2.40)

Then the numerical solution \( 1 \) \( (2.35), (2.36) \)

(2.41) will be close to the exact solution \( 1 \) \( (2.39) \) as we take \( L \) large.
$3$ Partial Differential Equations

The partial differential equations which occur in finance are parabolic. We consider the Black–Scholes equation,

\begin{equation}
\frac{\partial C}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + \frac{1}{2} \frac{\partial C}{\partial S} - r C = 0; \quad S > 0, \quad t < T.
\end{equation}

Here \( S \) = stock price, \( \sigma \) = stock volatility, \( r \) = risk free rate. The function \( C = C(S, t) \) is the value of a call option at time \( t \leq T \) with expiration date \( T \) and strike price \( K \).

We solve (3.1) with the terminal condition,

\begin{equation}
C(S, T) = (S - K)^+. 
\end{equation}

The equation (3.1) is derived by showing that Delta hedging of the option is risk free whence the option is equivalent to a hedged position in the stock.

We can transform the problem (3.1), (3.2) into a problem for the heat equation by a series of transformations. First we do our pricing in terms of money at time \( T \). Thus we put

\begin{equation}
F = S e^{-r (T-t)}, \quad \text{future price of stock at time } T.
\end{equation}

\begin{equation}
V = C e^{-r (T-t)} = \text{value of option in terms of money at time } T.
\end{equation}
Then (2.1), (2.2) becomes

(3.4) \[ \frac{\partial \nu}{\partial t} + \frac{1}{2} \sigma^2 F^2 \frac{\partial^2 \nu}{\partial F^2} = 0, \quad t < T, \quad F \neq 0, \]

\[ \nu(t, F, T) = (F - K)^+. \]

Next, we can make the problem dimensionless and rewrite it as:

(3.5) \[ F = \frac{y}{y_0}, \quad \nu = \frac{y}{y_0}, \quad \tau = \frac{1}{2} \sigma^2 (T - t), \]

where (3.4) becomes

(3.6) \[ \frac{\partial \omega}{\partial \tau} = \frac{y_0^2}{y^2} \frac{\partial^2 \omega}{\partial y^2}, \quad 0 < \tau, \quad y > 0, \]

\[ \omega(y, 0) = (y - 1)^+. \]

We can make (3.6) into a constant coefficient equation by going to log variables: \( y = e^x \), where

(3.7) \[ \frac{\partial \omega}{\partial \tau} = \frac{\partial^2 \omega}{\partial x^2} - \frac{\partial \omega}{\partial x}, \quad 0 < \tau, \quad -\infty < x < \infty, \]

\[ \omega(x, 0) = (e^x - 1)^+. \]

Finally, we can remove the 1st derivative in (3.7) by sending

(3.8) \[ u = \omega \exp \left[ \frac{x}{2} - \frac{\tau}{4} \right], \]

where

(3.9) \[ \frac{\partial u}{\partial \tau} = \frac{1}{2} \frac{\partial^2 u}{\partial x^2}, \quad \tau > 0, \quad -\infty < x < \infty, \]

\[ u(x, 0) = (e^{\frac{x}{2}} - e^{-\frac{x}{2}})^+. \]

In order to completely specify the heat equation problem (3.8), (3.9), we need to state the
boundary condition for \( u(x,t) \) as \( t \to \pm \infty \). These can be derived from the b.c.'s for the initial
boundary values equation (2.1). Thus
\[ C(0, t) = 0, \quad C(L, t) = S - K e^{-\lambda t}, \quad t > 0. \]
Note that the second b.c. is exactly \( F - K \) discounted as today. For (2.9), these translate into
\[ u(x, t) = 0, \quad x \to -\infty, \quad u(x, t) = \exp \left[ \frac{A + B}{2} \right], \quad x \to \infty. \]
The condition for \( x \to \infty \) of \( u(x,t) \) is the same as
\[ v(F, t) \sim F, \quad F \to \infty, \quad F = \frac{A - B}{2}. \]
\[ v \sim F = K u T \sim K e^{-\lambda t}, \quad x \to \infty. \]
We can use our methods for solving the
heat equation numerically. These methods consist of
a combination of the ODE methods of §2 and
the discretization techniques of §3. Consider for
simplicity the following problem:
\[ \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < L, \quad t > 0, \]
with initial condition
\[ u(x, 0) = u_0(x), \quad \text{and boundary conditions} \]
which are Dirichlet,
\[ u(0, t) = 0, \quad u(L, t) = 0. \]
We can interpret (2.13) — (2.15) in terms of
heat where \( u(x, t) \) is the temperature of a
bar of length \( L \) at the point \( x \) at time \( t \).
The function \( u_0(x) \) gives the initial temperature
distribution on the bar and the b.c.'s mean the
endpoints of the bar are maintained at fixed
temperatures \( a \) and \( b \).
We can turn (3.13) into an ODE system by using the discretization (2.13) of the second derivative. Thus if we put \( u_n(t) = u_n(h, t) \), \( n = 0, \ldots, N+1 \) we have approximately

\[
\frac{du_n}{dt} = \frac{2u_{n+1} + u_{n-1} - 2u_n}{h^2}, \quad n = 1, \ldots, N,
\]

where we set \( u_0(t) \equiv a \), \( u_{N+1}(t) \equiv b \). This system is second order accurate in space i.e. \( h = O(h^2) \). Now we introduce an ODE solver to give us a numerical method for solving (3.16). We need to be careful however for (3.16) is a stiff system. To see this let us look for a solution of (3.12) of the form

\[
u(x, t) = e^{\lambda t} \psi(x) \]

Taking homogeneous i.e. \( a = b = 0 \) we need from (3.12), (3.15) that

\[
d^2\psi + \lambda \psi = 0, \quad 0 < x < \pi, \quad \psi(0) = \psi(\pi) = 0.
\]

The solutions to (3.18) are the eigenfunctions and eigenvalues given by

\[
\lambda = \pi^2 n^2, \quad \psi_n(x) = \sin n x, \quad n = 1, 2, \ldots \]

Thus the solution to (3.13) - (3.15) with \( a = b = 0 \) and

\[
u_0(x) = \psi_0 \sin \pi x
\]

is given by

\[
u(x, t) = y(t) \sin \pi x,
\]

where \( y(t) \) satisfies the ODE
\[
\frac{dy}{dt} = -\pi^2 n^2 y, \quad y(0) = y_0.
\]

Since \( n \) can be large, the time step \( \Delta t \) we can use in numerically solving (3.22) with a forward method will have to be very small. Hence implicit methods are more appropriate for solving the heat equation numerically.

We first consider the forward Euler method applied to the system (3.16). If we put \( u^n = u_n(x, tk) \sim u(nh, mk) \), this gives us

\[
\frac{u^{m+1}_n - u^m_n}{\Delta t} = \frac{u^{m+1}_{n+1} + u^{m+1}_{n-1} - 2u^m_n}{\Delta x^2}, \quad n = 1, \ldots, N,
\]

where we have

\[
u^{m+1}_n = d u^m_n + (1-2d) u^m_{n-1} + 2 u^m_{n+1},
\]

where \( d = \frac{\Delta t}{\Delta x^2} \). Now the stiffness of the system requires us to make \( \Delta t \) small, sufficiently small in fact. Note that all coefficients on the RHS of (3.24) are non-negative. Thus we need \( d \leq \frac{1}{12} \). In that case the iteration (3.24) can be interpreted in terms of a random walk. Thus

\[
u^{m+1}_n = E \left[ u^m_n \right] \quad \text{for a walk started at } y_0, \text{ after } n \text{ steps. The walk steps at } y \text{ with prob. } \frac{1-2d}{d} \text{ and moves to } y \pm 1 \text{ with equal prob. } \frac{d}{2}. \]

Here, since we have
(3.26) \[ \max_{0 \leq n \leq N} |u_n^{m+1}| \leq \max_{0 \leq n \leq N} |u_n^m| \]

where the scheme is stable if \( 2 < 1/2 \). Otherwise it will in general be unstable.

If we apply the backwards Euler method to the ODE system (3.16) we get the fully implicit scheme for solving the heat equation,

\[
\frac{u_n^m - u_n^{m-1}}{k} = \frac{u_{n+1}^m + u_{n-1}^m - 2u_n^m}{h^2},
\]

which is the same as

\[
(1 + 2\beta) u_n^m - 2u_{n+1}^m - 2u_{n-1}^m = u_n^{m-1},
\]

where this is now a linear algebra problem of the form \( Au = b \) which we can solve by the Gaussian-Seidel or for algorithms of §2. In fact (3.28) is a discretization of the differential equation (2.21) with Dirichlet b.c. The smaller \( 2 \) is the faster the iteration algorithm will converge for (3.28). The error in solving (3.27) from the exact problem is \( O(k) \) coming from the time error - and \( O(h^2) \) from the spatial error. We should choose \( \beta \) to be close to \( 1 \) and \( 0(1/k^2) \), have the same order. Since \( 2 = k/h^2 \) we should therefore choose \( \beta = 0(1/k) \). This method is even more computationally intensive than forward Euler. On the other hand we can take \( \beta \) to be very large, say \( \beta = 0(1/k) \) in
which case \( k = h \) and the Actel error in the method is \( O(h) \). The Actel Forward Euler does not allow us this option since we must have \( \alpha = O(1) \). Evidently, if \( \alpha = O(\frac{1}{h}) \) the method is computationally much more efficient than Forward Euler. Although the Actel

error will be larger - \( O(h) \) instead of \( O(h^2) \).

We can with the same amount of computation attain an implicit method for which

the Actel error is \( O(h^2) \). To do this we introduce a 2nd order accurate method for

time - the Crank-Nicholson method. Applying this to

the ODE system (3.16) yields the difference scheme

\[
(3.29) \quad u_n^{m} = u_n^{m-1} + \frac{h^2}{2} \left( \frac{u_{n+1}^m + u_{n-1}^m - 2u_n^m}{h^2} + \frac{u_{n+1}^{m-1} + u_{n-1}^{m-1} - 2u_n^{m-1}}{h^2} \right)
\]

The algorithm (3.29) is called the Crank-

Nicholson algorithm for solving the heat equation.

We can rewrite (3.29) as

\[
(2.30) \quad (1+\delta) u_n^m - \frac{\delta}{2} u_{n+1}^m - \frac{\delta}{2} u_{n-1}^m =

(1-\delta) u_n^{m-1} + \frac{\delta}{2} u_{n+1}^{m-1} + \frac{\delta}{2} u_{n-1}^{m-1}
\]

The equation (2.30) is again of the form \( Au = b \)

and we can solve it via Gaussian-Seidel or SOR

no matter how large \( \delta \) is. Since this scheme

is 2nd order we can by taking \( \alpha = O(\frac{1}{h}) \)
where \( h \) is chosen a solution which is accurate correct to \( O(h^2) \).

In implementing the above scheme for (3.12)-(3.15) are typically chosen \( h = 2^{-5}, 5 \approx 1,2, \ldots \) to each of its value is a systematic increment on the screen. In order to represent the computed function graphically or compare errors we may wish to find the integrated values of \( u(x, t) \) at fixed values of \( x \) independent of \( h \), say \( x = 0, 125, 25, 75, 125 \). To estimate the values at \( x = 1 \), \ldots \) we use interpolation. The simplest form of interpolation method is linear interpolation where the function is approximated by a piecewise linear function. The obvious disadvantage of this is that the interpolated function is not differentiated at the grid points. Here if no grid points are accurate to \( O(h^2) \) this is not a good method of interpolation. We need interpolation which preserves the differentiability, and hence the error of the interpolated function from the true solution is \( O(h^3) \) for all \( x \). One can do this by using polynomial interpolation, cyclic functions, etc.

The numerical methods we have introduced
can be applied directly to the PDE equation (3.1). More generally, let us consider a case where the stock pays a dividend at rate D. Then (3.1) gets modified to

\[
\frac{\partial C}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + \left( r - D \right) S \frac{\partial C}{\partial S} - rC = 0, \quad S > 0, \quad t < T.
\]

The terminal and boundary conditions are given by

\[
C(S, T) = (S - K)^+, \quad \text{for } T = T_t - (T - t),
\]

\[
C(0, T) = 0, \quad C(S, T) \sim SE^{-r(T-t)} - K, \quad S \to \infty.
\]

The b.c. at \( S = 0 \) is obtained by considering what the contract is to deliver a stock \( S \) at time \( T \). The present day price of a stock to be delivered at time \( T \) is \( S e^{-(T-t)} \) since we buy the stock at time \( t \) and collect the dividends in the interval \( [t, T] \) before delivering it at time \( T \). Note that both functions \( S e^{-(T-t)} \) and \( K e^{-(T-t)} \) satisfy the PDE (3.31). Now we choose an interval \([0, S_0] \) on which to implement the numerical scheme. How do we decide how large to make \( S_0 \)? This depends on \( r \) and \( K \). Suppose \( S_0 \) is sufficiently small that we do not expect more than 5% fluctuation in the stock price in \( [0, S_0] \).
interval \([t, T]\). Then we can take \(\sigma \to \infty\) since this time interval is sufficient. Note that the difference \(\Delta r^2\) is zero and \(r^2\) will not significantly distinguish \(\Psi(T) - \Psi) \) and \(\Psi(T)\), where

\[\Psi \Delta \text{c.o.} \quad (3.36) \text{ is valid for } \quad s = s_{\text{max}}. \]

We consider this model of the case interest rate model for bond pricing. In this model, let \(B(s, t)\), \(t \leq s \leq T\) be the price of a bond which has a face value of \(s\) payable at time \(T\) and no coupon. The current short rate of interest \(r\) is the rate of interest \(r(t), t \leq s \leq T\) at all times, as shown in advance. Then

\[B(s, t) = \exp \left[ - \int_t^s r(s) \, ds \right]. \tag{3.34} \]

In general \(r(s)\) is unknown. The next simplest model is to consider \(r(s)\) to be a stochastic process. In the CIR model \(r(s)\) is given by

\[dr = (\alpha - \beta r) \, dt + \sigma \sqrt{r} \, dW(t), \tag{3.35} \]

where \(W\) is the Wiener process. In that case

\[B(s, t) = E \left[ \exp \left( - \int_t^s r(s) \, ds \right) \bigg| r(T) = s \right]. \tag{3.36} \]

Note that (3.34) is the theoretical exact value of the bond since we can hedge with a money market account which produces a zero risk part plus the fact that this does not always hold in practice. So because the short term rate is not
It is easy to see that the solution (3.26) with $\phi(x)$ satisfying (3.25) is

\[
\frac{\partial \phi}{\partial t} + \frac{1}{2} \sigma^2 t \frac{\partial^2 \phi}{\partial x^2} + (a - \frac{1}{2} \sigma^2) \frac{\partial \phi}{\partial x} - r \phi = 0, \quad 0 < t < \infty,
\]

(3.27)

\[
\phi(x, T) = x.
\]

The above model the interest rate $r(t)$ always satisfies $r(t) \geq 0$. It has known the unfortunate property that if $r(t_o) = 0$ for some $t_o$, then $r(t) = 0$ for all $t > t_o$, clearly an undesirable property of actual short rates. One can avoid this by choosing $a$ large enough, in fact $a > \sigma^2/2$. Then $r(t) > 0$ for all $t$ provided $r(t_o) > 0$.

Consider now the problem of solving (3.27) numerically. As before we restrict the interval of $t$ to $0 < t < \infty$. For the $r = 0$ case we could just simply set $r = 0$ in (3.27) to obtain

\[
\frac{\partial \phi}{\partial t} + a \frac{\partial \phi}{\partial x} - \frac{\partial \phi}{\partial x} = 0.
\]

(3.29)

For $r = 0$ case the b.c. is not so clear. The process (3.25) when the mean reverting property is set aside runs for a long time it will settle down around $r = a/2$. If $r - b$ is much smaller than the time to equilibrium then we can ignore the diffusion term in (3.27) to get the b.c. at once as

\[
\frac{\partial \phi}{\partial t} + (a - b \gamma) \frac{\partial \phi}{\partial x} - r \phi = 0.
\]

(3.40)
We can implement these directly or implicitly. For $\tau = 0$ the c.e. (3.39) is

\[ B^m_n - B^m_0 + \frac{m}{h} \frac{B^m_n - B^m_0}{h} = 0. \]

Note that (3.41) is explicit because we are backwards in time with $m < M$ where $M \neq 0$.

Alternatively, we can do a second order approximation for $\partial B_{ij}/\partial t$ given by

\[ \frac{\partial^2 B^m_n}{\partial t^2} = \frac{3}{2h} B^m_0 - 4 B^m_n + B^m_n. \]

By Taylor expansion we have

\[ B(c + h, t) = B + \frac{3}{2} B_{rr} h^2 + \frac{3}{2} B_{rrr} h^3 + O(h^4). \]

Thus leads to

\[ \alpha + h + c = 0, \]

\[ b + 2c = 4, \]

\[ b + 4c = 0, \quad \text{where} \]

\[ \alpha = \frac{-3}{2}, \quad b = 2, \quad c = -\frac{1}{2}. \]
§ 4  American Options  

Options can have early exercise features which means one can exercise the option at any time. These are generally referred to as American options. Let us consider an American put option with strike price \( K \) and expiration date \( T \). The payoff function of the option is \((K-S)^+\). Let \( P(S, t) \) be the price of the option at time \( t < T \). Explicitly we have

\[
(4.1) \quad P(S, T) = (K-S)^+. 
\]

Since we can exercise the option at any time we also have

\[
(4.2) \quad P(S, t) > (K-S)^+, \quad t < T. 
\]

Observe now that the price of an American put must be at least the price of a European put, where \( P(S, t) > 0 \) for all \( t < T \). It follows that we can prove equality in (4.2) only when \( S < K \).  

Indeed, if \( P(S, t) \geq (K-S)^+ \) then one can hedge the option by a portfolio of stocks i.e. dollars hedge the option, whereas \( P(S,t) \) must satisfy

\[
(4.3) \quad \frac{\partial P}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} + \rho S \frac{\partial P}{\partial S} \frac{\partial r}{\partial S} = 0. 
\]

Consequently, the price of the option is satisfied by (4.3) if \( P > K-S \) and in general,

\[
(4.4) \quad P > K-S. 
\]

Thus we get a picture.
Thus if \( S \) lies below the red line we have the option. The red line itself is given by

\[
\frac{\partial S}{\partial S} = -1, \quad S = S_e(t),
\]

and this curve is called the exercise boundary. This is a so-called free boundary since it is not given a priori but is determined by solving

\[
(4.3), \quad (4.4). \quad \text{One should note that on the free boundary one has that}
\]

\[
\frac{\partial S}{\partial S} = -1, \quad S = S_e(t),
\]

Now (4.1) is evident from (4.5) if we approach \( S_e(t) \) with \( S < S_e(t) \). It is not immediately clear if we approach \( S_e(t) \) with \( S > S_e(t) \). To see this consider the situation with

\[
\lim_{S \to S_e(t) + \delta S} \frac{\partial S}{\partial S} < -1. \quad \text{Then it is clear that for } S > S_e(t) \text{, but close to it, that (4.4) does not hold. On the other hand}
\]

\[
\lim_{S \to S_e(t) + \delta S} \frac{\partial S}{\partial S} > -1.
\]

Recall that for \( p > (K-s)^+ \), we hedge
The option with 0.7% of stock, hence, we have

\[ (4.10) \quad \text{a put option} - (0.7\%) \text{ stock} = \text{cash flow.} \]

Consider next what happens when \( S \) decreases as \( S_2(t) \) and \( (4.9) \) holds. Then the put option consists of

\[ (4.11) \quad \text{a put option} + \Delta (\text{stock}) \text{ with } 2 < 1, \]

if \( S \) decreases from \( S_2(t) \). Then it is clear that the portfolio \( (4.11) \) yields a profit. On the other hand, if \( S \) increases, there is no profit. Thus, it is therefore an advantage where we must rule out \((4.9)\). The graph \( S_2(t) \) as a function of \( t \) looks like the following:

\[ -\text{slope} = -1 \]

\[ -2 \quad \text{no change} \]

\[ 2 \]

We can numerically solve \((4.3), (4.4)\) by simply adapting the methods used for European options. For the American put option problem, \((4.3), (4.4)\), thus \( C_n, \Delta \) for \((4.3)\) yields the equation

\[ (4.12) \quad \frac{P_{n+1}^m - P_n^m}{\Delta t} + \frac{1}{2} \sigma^2 \left( n \Delta S \right)^2 \frac{1}{2} \left[ \frac{P_{n+1}^{m+1} - 2P_n^m + P_{n-1}^m}{(\Delta S)^2} \right] + \gamma \left( n \Delta S \right) \frac{1}{2} \left[ \frac{P_{n+1}^m - 2P_n^m + P_{n-1}^m}{2 \Delta S} \right] - \frac{1}{2} \left[ \frac{P_n^{m+1} + P_n^m}{2 \Delta S} \right] = 0. \]
Here \( M_{AT}=T, \ n=0, \ldots, \ N \) and we are going to work backwards in time. Thus \( p_n \) are given from the

\[ (4.13) \]

\[ p_m = \begin{bmatrix} K - n \Delta \xi \end{bmatrix} \]

given \( p_{m+1} \) we compute \( p_m \) from (4.12) using

(4.12) and the specified boundary conditions \( p_0 \)

If we write (4.12) as

\[ (4.14) \]

\[ p_n = p_{n-1} + p_{m+1} \]

we define iterations for \( k = 0, 1, \ldots \).

Thus in the record part we enforce the conditions (4.14), this scheme will solve the free boundary problem.

§ 5 Path Dependent Options

The payoff on these options depends not only on

the final stock price but also the actual path.

For an American option the payoff depends on the

the stock price at the expiration date. For a

European option the payoff depends on the

average stock price during the lifetime. What these options have in common is that
The pde governing the price of the option has 2 variables \( S_t \) and \( t \). \( S_t \): current stock price, \( t \): current time. \( t = 0 \) is the beginning of the option until current time. For the Black-Scholes option, current stock price \( S_t \) is replaced by mean or most stock price from beginning of the option.

We first consider the continuously sampled American option. Let us suppose the life of the option is \( 0 < t < T \), and the pay off is

\[(5.1) \quad \text{payoff} = \left( e^{-rT} \int_0^T f(S_t) \, ds + s(1) \right),\]

so this is like a perpetual option. To price this we define \( I \) by

\[(5.2) \quad I = \int_0^T f(S_t) \, dt, \quad \text{and let the value of the option be } V(S, I, t), \quad \text{thus implicitly assuming we can value the option as a function of the 3 variables } (S, I, t). \]

Note in the case when

\[(5.3) \quad \text{value of option at time } 0 = 0. \quad V(S, 0, 0).\]

For the American case, we have

\[(5.4) \quad V(S, I, T) = \left[ -\frac{I}{T} \right] + f(S_T).\]

To price the option we construct a portfolio \( I_t \) consisting of

\[(5.5) \quad I_t = \frac{1}{2} \text{ call option} - \frac{\partial V}{\partial S} \text{ (strike)},\]

as usual. Then using the no-arbitrage argument we have

\[(5.6) \quad d I_t = r I_t \, dt, \quad \text{where} \quad d I_t = \left[ \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + S \frac{\partial V}{\partial S} \right] \, dt,\]

since we have
\[ dI = 2 \, dt \quad \text{Here the p.d.e for} \quad \text{the cell is given by} \]

\[ \frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 \frac{\partial^2 V}{\partial s^2} + \frac{1}{2} \frac{\partial^2 V}{\partial t^2} - (\rho t - s) V = 0, \quad 0 < t, \, I < \infty, \quad 0 < t < T. \]

In order to solve (5.9) with terminal condition,

\[ V(0, T, I) = 0, \]

we need to impose boundary conditions. On the other side, if \( I \to 0 \), then it never becomes large again where the value of the option should be 0. We have therefore

\[ V(0, I, I) = 0. \]

As \( I \to 0 \), the b.c. is not obvious. Explicitly,

\[ I = 0 \quad \text{at the beginning of the option.} \]

We can see what the \( I = 0 \) b.c. is by making a variable change in (5.9) to reduce it to a p.d.e in just 2 variables. We put

\[ s = \frac{t}{\sigma^2} \quad \rightarrow \quad V(s, I, t) = \frac{1}{\sigma^2} W(s, t). \]

Thus

\[ \frac{\partial V}{\partial t} = \frac{1}{\sigma^2} \frac{\partial W}{\partial t}, \quad \frac{\partial V}{\partial s} = \frac{1}{2} \frac{\partial^2 W}{\partial s^2} \left( \frac{-s}{\sigma^2} \right). \]

\[ \frac{\partial^2 V}{\partial s^2} = \frac{1}{2} \frac{\partial^2 W}{\partial s^2} \left( \frac{-s}{\sigma^2} \right) + \frac{1}{\sigma^2} \frac{\partial^2 W}{\partial s^2} = \frac{1}{2} \frac{\partial^2 W}{\partial s^2} \left( \frac{-s}{\sigma^2} \right) = \frac{1}{2} \frac{\partial^2 W}{\partial s^2}. \]

\[ \frac{\partial V}{\partial I} = \frac{1}{\sigma^2} \frac{\partial W}{\partial I} = - \frac{1}{\sigma^2} \frac{\partial W}{\partial s} \left( \frac{s}{\sigma^2} \right) = \frac{1}{\sigma^2} \frac{\partial W}{\partial s}. \]

Hence (5.9) becomes

\[ \frac{\partial W}{\partial t} + \frac{1}{2} \sigma^2 \frac{\partial^2 W}{\partial s^2} \left[ \frac{1}{2} s^2 \frac{\partial^2 W}{\partial s^2} \right] + \frac{1}{2} \frac{\partial^2 W}{\partial s^2} \left[ \left( \frac{1}{s^2} - \frac{1}{\sigma^2} \right) \frac{\partial W}{\partial s} \right] = 0, \]
which in the case as
\[ (5.14) \quad \frac{\partial W}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 W}{\partial S^2} + (1 - \frac{\delta}{\alpha}) \frac{\partial W}{\partial S} = 0. \]

The normal condition (5.4) becomes now
\[ (5.15) \quad W(S, T) = \frac{1}{T} (T - S) +. \]
The b.c. (5.10) is given by
\[ (5.16) \quad W(S, 0) = 0. \]
To get the \( S = 0 \) b.c. we just set \( S = 0 \)
in (5.16) to obtain
\[ (5.17) \quad \frac{\partial W}{\partial S} + \frac{\partial W}{\partial t} = 0. \]
Note the payoff in (5.15) is the same as a put option limit (5.14) is not quite a Black-Scholes equation. The b.c. (5.17) is similar to the 0 b.c. for the CIR model.

To solve (5.14) - (5.17) numerically we discretize on a finite interval \([0, 1]\) where we take \( S_{\max} > T \) so the b.c. (5.16) is
\[ (5.18) \quad W(S_{\max}, T) = 0. \]
The explicit discretization of (5.14) is given by
\[ (5.19) \quad \frac{W_{n}^{m+1} - 2W_{n}^{m} + W_{n-1}^{m}}{\Delta S^2} + \frac{1}{2} \sigma^2 (\Delta S)^2 \frac{W_{n+1}^{m-2} - 2W_{n}^{m-2} + W_{n-1}^{m-2}}{\Delta 2 S^2}
+ (1 - \alpha \Delta S) \frac{W_{n+1}^{m-1} - 2W_{n}^{m-1} + W_{n-1}^{m-1}}{2 S^2} = 0, \]
where we work backwards in time as usual. The b.c. (5.17) is implemented as
\[ (5.20) \quad \frac{W_{0}^{m} - W_{0}^{m-1}}{\Delta t} - \frac{3W_{0}^{m} - 4W_{1}^{m} + W_{2}^{m}}{2 \Delta S} = 0. \]
We can also consider directly sampled asian options. Note that the payoff on the call option is

\[
\left[ S(t) - \frac{1}{T+1} \sum_{i=0}^{T} S(t_i) \right]^{+}
\]

Here, the payoff is the excess of the arithmetic stock price over the stock average at the times \( t_i \). The price of this option can be determined by solving a PDE in 3 variables, but we have no restriction to a 2 variable problem. We define

\[
I = \sum_{\substack{t_i \leq t \leq T}} S(t_i)
\]

The value of the option is a function of current stock price \( S \), \( I \) and \( t \). Explicitly, for the American conditions on the value \( V(S, I, t) \) we have

\[
V(S, I, T) = \left[ S - \frac{I}{T+1} \right]^{+}
\]

Otherwise, that is for any \( S \), \( I \) we need to put in jump conditions at \( t = t_i \). So in view of

\[
V(S, I, t_i^-) = V(S, I + I, t_i^+)
\]
Note that although the PDE (5.24) does not involve \( I \) explicitly, the jump conditions (5.25) really make this a 3 variable PDE problem since we need to solve (5.24) for all values of \( I \).

To solve (5.24) numerically we wish on a 2 dim grid.

Thus: \( t = 0, \Delta t, \ldots, M\Delta t = T; \quad l = 0, N + 1, 2N + 1, \ldots, 5N + 1 \).

Let us consider instead of the pay off (5.22) a pert option. Large pay off if given by:

\[
V(l, n, i, t) = \left( K - \frac{i}{l+1} \right)_{+}^2
\]

When

\[ K \text{ is the strike price. In that case as } \ell \to \infty, \quad V \to 0. \]

Now we have

The h.c.

\[
V(0, 0, i, t) = 0,
\]

and for \( l \geq 0 \) the B.S. equation (5.24) will yield

\[
V(l, 0, i, t) = \left( K - \frac{i}{l+1} \right)_{+}^2
\]

The h.c. (5.21), (5.22) is all we need, i.e. no h.c. on \( i \) would required that the PDE not explicitly involve \( I \).

Assume now the sampling points for \( i \) are uniformly spaced grid points:

\[
l = 0, \Delta l, t = m\Delta t, t_2 = 2m\Delta t, \ldots, t_T = m\Delta t = T
\]

i.e. pick \( M \) so that \( T \) divides \( M \) and write \( n = M/\ell \). Then we set

\[
V(l, n\Delta l, j\Delta s, m\Delta t) \quad V(n, j)
\]

\[ 0 \leq m \leq M, \quad 0 \leq n \leq N, \quad 0 \leq j \leq J. \]
As usual we work backwards in time, so we first compute \( V^M \), then \( V^{M-1} \), etc. going from \( V^M \).

To \( V^n \) we can use explicit or implicit methods.

Here we focus on the Bank-Nicholson scheme.

We first for the Assumed condition

\[
V^{n+1}_{n,j} = \left[ K - \frac{j}{t+1} (\Delta S + \Delta S) \right]^+ 
\]

which corresponds to a condition \((5.25)\) i.e.

\[
V^n_{n,j} = V(T, S, J, t) 
\]

Before we have computed \( V^M, V^{M-1}, \ldots, V^{M+1} \). Then we compute \( V^m \) as follows:

(a) \( q \Delta S > (t+1) \Delta S \) then \( V^m \equiv 0 \) for all \( n \) (payoff will be \( q \)).

(b) \( q \Delta S > (t+1) \Delta S \) then proceed as follows:

(c) \( V^m_{n,j} = \left[ K - \frac{j}{t+1} (\Delta S + \Delta S) \right]^+ \) for \( 0 \leq n \leq N \) as follows:

\[
V^n_{n,j} = \frac{V^{n+1}_{n,j} - V^n_{n,j}}{\Delta t} + \frac{1}{2} \Delta S^2 \left( \frac{\Delta S}{\Delta t} \right)^2 \frac{1}{2} \left[ V^{m+1}_{n,j} - 2V^m_{n,j} + V^{m-1}_{n,j} \right] 
\]

(5.32)
\[ + \frac{V_{n+1,\Delta} - 2V_{n,\Delta} + V_{n-1,\Delta}}{2\Delta^2} \left( \frac{V_{n+1,\Delta} - V_{n-1,\Delta}}{2\Delta} \right) \]

\[ + \frac{1}{2} \left[ V_{n,\Delta}^m + V_{n,\Delta}^{m+1} \right] = 0. \]

Solve this system of equations using \( s \Delta^m \).

(c) Suppose we are at a sampling point \( x = n \Delta \), then implement a semi-implicit.

1. Replace \( V_{n,\Delta}^m \) by \( \hat{V}_{n,\Delta}^{m-1} \) if \( n \leq 1 \);

2. Replace \( V_{n,\Delta}^m \) by \( \hat{V}_{n,\Delta}^0 \) if \( n > 1 \).

For implementation: We write (5.35) as

\[ (5.34) \quad V_{n,\Delta}^m = \hat{V}_{n,\Delta}^{m-1} + \rho^n V_{n+1,\Delta} - L_{n,\Delta}. \]

We solve (5.34) to get \( \hat{V}_{n,\Delta}^{m-1} \). Thus set

\[ V_{n,\Delta}^m = \hat{V}_{n,\Delta}^{m-1}. \]

\[ \frac{V_{m,\Delta}^{k+1} - V_{m,\Delta}^k}{\Delta} \]

where we set

\[ V_{m,\Delta}^0 = \hat{V}_{m,\Delta}^0 \]

and

\[ V_{m,\Delta}^k = \hat{V}_{m,\Delta}^{k-1} + \omega \left[ \hat{V}_{m,\Delta}^{k-1} - \hat{V}_{m,\Delta}^{k-2} \right]. \]
Chapter II — Monte Carlo Methods

The basic method is based on the law of averages for independent random variables. Suppose we consider independent tosses of a coin. Then for a large number of tosses, the proportion of heads should converge to 1/2. We can put this more precisely as follows:

Let \( X_1, X_2, \ldots \) be independent random variables, \( X_i = 2 \) with prob. 1/2, \( X_i = 0 \) with prob. 1/2. If we set \( X = \sum X_i \), the win comes 0.

Thus we have

(1.1) \[
\frac{\text{proportion of heads}}{N} \text{ in } N \text{ independent tosses of a coin} = \frac{\sum X_i}{N} = \frac{1}{2} \text{ with prob. } 1.
\]

The law of averages or the strong law of large numbers states that

(1.2) \[
\lim_{N \to \infty} \frac{\sum X_i}{N} = \frac{1}{2} \text{ with prob. 1 } \quad (N \to \infty)
\]

Note: Note that (1.2) is a strong result with probability 1. It is possible for all times to come down tails but the probability of this is 0. Thus

(1.3) \[
P(X_1 = X_2 = \ldots = X_N = 0) = \left(\frac{1}{2}\right)^N, \text{ where}
\]

(1.4) \[
\lim_{N \to \infty} \frac{1}{2^N} = 0.
\]
We can be more precise about this and see that as \( n \) must be used the result is \( F_{1/2} \). The answer to this is a consequence of the central limit theorem:

(1.5) \( 2 \sum_{i=1}^{n} \left[ \frac{x_i \cdot x_i}{n} - \frac{1}{2} \right] \) converges in law to a standard normal variable \( Z \). Thus

(1.6) \( \Phi \left( \frac{1}{\sqrt{2}} \right) \mathcal{N} \left[ -\frac{\bar{Z}^2}{2} \right] \), \( -\infty < Z < \infty \).

Thus

(1.7) \( \lim_{n \to \infty} \Phi \left( \left[ \frac{\bar{Z}^2}{n} \right] \right) = 2 \left[ 1 - \Phi(\bar{Z}) \right] \)

(1.8) \( \Phi(\bar{Z}) = 1 - 1.845 \) is small.

Thus for large \( n \) the chi-square of (1.2) differs by not more than 1.845 by with less than 10% confidence. The previous considerations are applicable for general random variables. Thus let \( \bar{X} \) be a r.v. with finite variance i.e. \( \text{E}[\bar{X}^2] < \infty \). We have

(1.9) \( \mu = \text{E}[\bar{X}] = \text{mean of } \bar{X} \)

(1.10) \( \sigma^2 = \text{Var} \bar{X} = \text{E}[\bar{X}^2] - \text{E} \bar{X}^2 \).
Let \( X_1, X_2, \ldots \) be i.i.d. variables with the distribution of \( X \). Then

\[
X + X + \cdots = E[X] \quad \text{with prob 1.}
\]

(1.11)

\[
\sqrt{n} \left( \frac{X_1 + X_2 + \cdots}{\sqrt{n}} - E[X] \right) \quad \text{converges in law to a standard normal random variable.}
\]

(1.12)

The Monte Carlo method was first used to obtain a good approximation to all kinds of averages. Suppose we have a function \( g : [0, 5] \rightarrow \mathbb{R} \) and we wish to find the approximate value of the integral. We can write this as

\[
\int_0^5 g(x) \, dx = \int_0^5 \left( \frac{1}{5} \right) g(x) \, dx = \frac{1}{5} E[g(Y)]
\]

(1.13)

where \( Y \) is a random variable, uniform on \([0, 1]\). If we can generate efficiently i.i.d. variables \( Y_1, ..., Y_m \) with the uniform distribution then we have

\[
\int_0^5 g(x) \, dx \approx \frac{1}{5} \left( \frac{g(Y_1) + \cdots + g(Y_m)}{m} \right)
\]

(1.14)

with an error which looks like \( O(1/\sqrt{m}) \).

(1.15)

\[
\int_0^5 g(x) \, dx \approx \frac{1}{5} \left( \frac{g(Y_1) + \cdots + g(Y_m)}{m} \right)
\]

where \( Y_1, ..., Y_m \) are uniformly distributed in the interval \( i.e., Y_1, Y_2, \ldots, Y_m = (i-1)/m, \ldots, (n-1)/m \).
The MC method does not seem efficient at all. In fact, the Metropolis algorithm (1.15) has error $= O(1/n)$ — better. A computer uses a slower scheme such as the Vaygert grid rule, and of course, we do not have to generate one. The MC scheme seems into its own when we go to higher dimensions. Thus the error estimate in (1.12) is always $O(1/n)$ even if the integrand is in many dimensions. Thus

$$\int_{(1.16)} f \left( \frac{y}{y_i} \right) dy \approx \frac{1}{n} \left[ g(y_i) \right]$$

with error $O(1/n)$. Note however, the proportionality constant will depend on $d$. That constant is known

$$\int_{(1.19)} f \left( \frac{y}{y_i} \right) dy \approx \frac{1}{n} \left[ g(y_i) \right]$$

with $Y_i$ uniformly distributed in $[0,1]$. Here are some

$$\int_{(1.18)} \left( y_i - y_j \right) \sim \sqrt{1/d}$$

Here error in (1.18) is worse $O(1/n^{1/2})$ which is already worse than MC for $d \geq 2$. Suppose we use with $A_\infty$ as an MC estimator to compute the mean $E[Y]$ of some $Y$.

(a) Generate independent draws of $Y$. Let $A_{\infty}$

say $Y = \Phi(Y)$ and $\Phi$ is uniform. Then

$A_{\infty}$, can be done by generating independent draws of $Y$. 
We expect $\hat{\Sigma}_N$ as an estimate of $\Sigma$, but we also want to report the error estimate in the form of the standard error, which is $\sigma/\sqrt{N}$ when $\sigma$ is the standard deviation of $\Sigma$. We of course do not know $\sigma$ so we estimate $\sigma$ from the data. Thus an unbiased estimate of $\sigma^2$ is

$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left( \frac{1}{N} \sum_{i=1}^{N} x_i^2 - \left( \frac{1}{N} \sum_{i=1}^{N} x_i \right)^2 \right).$$

The standard error for the MC simulation is

$$E_x = \frac{\hat{\sigma}_N}{\sqrt{N}}.$$
\( S_T = S_0 \exp \left[ (r - \frac{\sigma^2}{2}) T + \sigma \sqrt{t} \xi \right] \),
where \( \xi \sim N(0,1) \). Thus, \( S_T \) is the exponential of a normal variable and
\[
E[S_T] = S_0 \exp(\sigma \sqrt{T} \xi).
\]
To obtain an MC estimate of the put price (1.22) we proceed as follows:

1. Generate independent \( N(0,1) \)'s \( \xi^{(i)} \), \( i = 1 \to N \).
2. Compute \( S_T^{(i)} = S_0 \exp \left[ (r - \frac{\sigma^2}{2}) T + \sigma \sqrt{T} \xi^{(i)} \right] \), \( 1 \leq i \leq N \).
3. Compute \( \hat{p}^{(i)} = e^{-rT} (K - S_T^{(i)})^+ \) \( 1 \leq i \leq N \) i.e. discounted payoff.
4. Compute \( \hat{P} = \frac{1}{N} \sum_{i=1}^{N} \hat{p}^{(i)} \) (average MC estimate of put price) as well as the standard error
\[
\hat{s}_N = \sqrt{\frac{1}{N(N-1)} \sum_{i=1}^{N-1} \left( \hat{p}^{(i)} - \hat{P} \right)^2}
\]
5. Repeat \( \hat{P} \), \( \hat{s}_N \) and a convergence diagram i.e. the graph of \( (N, \hat{p}) \), \( 1 \leq N \leq 20 \).

Evidently, the whole MC method requires us to generate random variables with given distribution. As it is unrealistic for a deterministic algorithm to yield random numbers, the best one can do is to generate pseudo-random numbers. The
compute the C. E. R. y using an efficient algorithm when the correlation between distinct events is small. Typically, the computer's basic random number generator produces random numbers which are uniformly distributed in the interval \([0,1]\). Do we get random variables with a different distribution? We may use for example the c.d.f. of the distribution to generate random values of the variable \(Y\) and then values uniform in \([0,1]\). Thus suppose a variable has continuous p.d.f.

\[ f(x) = \begin{cases} \frac{1}{a} & 0 < x < a \\ 0 & \text{else} \end{cases} \]

(1.25)

Then the c.d.f. \(F\) given by

\[ F(x) = \int_{0}^{x} \frac{1}{a} \, dt = \frac{x}{a} \]

(1.26)

is a strictly increasing function \(\mathbb{R} \to (0,1)\). Let \(X_1, X_2, \ldots\) be independent variables uniformly distributed in \((0,1)\). Then we can define variables \(Y_1, Y_2, \ldots\) by

\[ F(Y_i) = X_i, \quad i = 1, 2, \ldots \]

(1.27)

which with p.d.f. \(f\). To see this just note that any \(a \in \mathbb{R}^+\),

\[ P(Y < a) = P(X < \int_{0}^{a} f(t) \, dt) = \int_{0}^{a} f(t) \, dt \quad \text{since } X \text{ is uniform on } (0,1) \text{ where } Y \text{ has p.d.f } f. \]
Of course, to implement this method one needs to use the c.d.f. (1.26) which can be difficult, but now that c.d.f. highly accurate methods exist. An alternative method of generating normal variables which does not involve function inversion is the Box-Muller method. Thus, if \( (U, V) \) are independent uniform random variables on \((0,1)\), then we define \((X, Y)\) by

\[
X = \sqrt{-2 \log U} \cos (2\pi V), \\
Y = \sqrt{-2 \log U} \sin (2\pi V).
\]

One can see that \((X, Y)\) are independent standard normal variables. This is just the well-known change of variables from Cartesian to polar coordinates in dimension 2. Thus if \(X, Y\) are independent standard normal, then the joint pdf is

\[
\exp \left[ -\frac{X^2 + Y^2}{2} \right] \, dx \, dy \quad \text{normalization}
\]


\[
= e^{-\frac{r^2}{2}} \, d\theta \, dr \quad \text{normalization}
\]

Here \( R = (x^2 + y^2)^{1/2} \) and \( \Theta \) is the angle

We use independent \( \Theta \) being uniformly distributed in \([0, 2\pi]\). Hence

\[
e^{-\frac{r^2}{2}} \quad \text{normalization}
\]

\[
es^{\frac{-r^2}{2}} \, d\theta = e^{\frac{-r^2}{2}} d \left[ e^{-\frac{r^2}{2}} \right] = s^{1/2}
\]

Thus the variable \( U = \exp \left[ -\frac{r^2}{2} \right] \) is uniformly distributed in \([0, 1]\). The problem is that this method is rather slow. As the next basic problem 1, how to efficiently find
The integrals
\[
\int_{0,1}^{1} f(y) dy = \frac{1}{N} \sum_{i=1}^{N} f(y_i)
\]

are chosen randomly. The error is \(O(1/N)\). If the \(y_i\) are chosen
uniformly, the error is \(O(11/N)\) which is worse
Then random \(Y_1 > 2\). It is possible to choose
the \(y_i\) in such a way that
the error is almost \(O(11/N)\) — this latter
procedure. These numbers are called low-discrepancy
numbers. It would seem from this that one
should always use low-discrepancy numbers when
estimating high-dimensional integrals. Life is not
a single known. The low-discrepancy numbers
have worked better than random only for
intermediate dimensions \(d < 15\) while for large values
of \(d\) pseudo-random numbers give superior convergence.
An example of a low-discrepancy sequence is the
so-called Halton numbers. To generate \(N\) points in
the interval \([0,1]\) we write

\[(1.32)\]
\[n = \sum_{k=1}^{k-1} a_k 2^{k-1}, \quad 0 \leq a_k < 2, \]
\[\xi(n) = \sum_{k=1}^{k-1} a_k 2^{-k}, \]

Thus

\[1 = 0.101_2 \Rightarrow \xi(1) = 1/2, \]
\[2 = 0.10_2 \Rightarrow \xi(2) = 1/4, \]
\[3 = 0.11_2 \Rightarrow \xi(3) = 1/2 + 1/4 = 3/4, \]
\[4 = 0.010_2 \Rightarrow \xi(4) = 1/8, \]
\[5 = 0.101_2 \Rightarrow \xi(5) = 1/8 + 1/8 = 5/8. \]
To generate Halton numbers in 2 dimensions we pick 2 different primes say 2, 3. Then with

\begin{align}
(2.35) \quad & v = \sum_{k=1}^{\infty} \frac{2}{k} \left( \frac{2}{k-1} \right) \left( \frac{3}{k} \right) \\
& u = \sum_{k=1}^{\infty} \frac{2}{k} \left( \frac{2}{k-1} \right) \left( \frac{3}{k} \right)
\end{align}

we find that

\begin{align}
(2.36) \quad & f_v = \sum_{k=1}^{\infty} \frac{2}{k} \left( \frac{2}{k-1} \right) \left( \frac{3}{k} \right) \\
& f_u = \sum_{k=1}^{\infty} \frac{2}{k} \left( \frac{2}{k-1} \right) \left( \frac{3}{k} \right)
\end{align}

Thus \((f_v, f_u)\) are the Halton numbers in \([0, 1]^2\). For higher dimensions we proceed similarly just picking the representation of the integers \(v\) as different primes. In other cases \((f_v, f_u)\) are as in the dimension 2 case. For

\[(2.37) \quad f_v = \sum_{k=1}^{\infty} \frac{2}{k} \left( \frac{2}{k-1} \right) \left( \frac{3}{k} \right)
\]

we have

\begin{align}
& 1 = 000 \quad \Rightarrow \quad f_v = \frac{1}{13} \\
& 2 = 002 \quad \Rightarrow \quad f_v = \frac{2}{12} \\
& 3 = 010 \quad \Rightarrow \quad f_v = \frac{3}{12} \\
& 4 = 011 \quad \Rightarrow \quad f_v = \frac{1}{12} + \frac{1}{3} = \frac{4}{12} \\
& 5 = 012 \quad \Rightarrow \quad f_v = \frac{1}{12} + \frac{2}{3} = \frac{7}{12}
\end{align}

\section{2 Numerical Solution of Stochastic Diff Eqns}

Here we wish to numerically solve the DE,

\begin{align}
(2.1) \quad & dx_t = a(x_t, t) \, dt + b(x_t, t) \, dW_t \\
& x_0 = x
\end{align}

The most straightforward approach to this is to use a stochastic version of Euler’s method. Thus we discretize time 0, \(\Delta t, 2 \Delta t, \ldots, M \Delta t = T\).

Write \(X^n = X(m \Delta t)\). Then the Euler method applied to (2.1) gives the recursion,

\begin{align}
& X^{n+1} = X^n + a(X^n, t_n) \Delta t + b(X^n, t_n) \, dW_n
\end{align}
\[(2.2) \quad X_t = x + \int_0^t a(x, s) \, ds + \int_0^t b(x, s) \, dW_s\]

which gives

\[(2.3) \quad X_{t+\Delta t} = X_t + a(X_t, t) \Delta t + b(X_t, t) \Delta W_t + \frac{1}{2} \left( b(X_t, t) \right)^2 \Delta t \left[ \frac{\sigma^2}{2} - \frac{1}{2} \right].\]

To see why (2.3) is an improvement over (2.2) let us consider what may be said of the Euler method of (2.2) consists of the approximation:

\[(2.4) \quad X(t + \Delta t) = X(t) + a(X(t), t) \Delta t + \frac{1}{2} \left( b(X(t), t) \right)^2 \Delta t.

This is the SD of the variable in is \[\left\{b(X_t, t) \sqrt{\Delta t} = 0(\Delta t)\right\} \text{ The point about the Milstein scheme is that it gives the correct values of the mean and SD of }\]

\[(2.5) \quad X(t + \Delta t) - X(t) = \int_t^{t+\Delta t} a(x, s) \, ds + \int_t^{t+\Delta t} b(x, s) \, dW_s.\]

We have now
(2.6) \[ \int_t^{t+\Delta t} b(x, s) \, dw_s = \int_t^{t+\Delta t} b(x_t, t) \, dw_t + \Delta b(x(s), t) \frac{\partial b(x(s), t)}{\partial x} \]
\[ + \int_t^{t+\Delta t} d(x(s), s) \, dw_s + \int_t^{t+\Delta t} b(x_t, t) \, dw_t + \text{H.O.} \]
\[ \text{We have now} \quad \text{that}\]
\[ (2.7) \int_{t-\Delta t}^{t+\Delta t} dw_s, \, dw_t = \int_0^{\Delta t} \left[ \omega_s - \omega_t \right]^2 \, dw_s \]
\[ = \frac{\Delta t}{2} \left[ \omega_{t+\Delta t} - \omega_t \right]^2 \]

by Itô's formula. Thus we have
\[ (2.8) \int_t^{t+\Delta t} b(x, s) \, dw_s = b(x_t, t) \left[ \omega_{t+\Delta t} - \omega_t \right]^2 \]
\[ + \frac{1}{2} b(x_t, t) \frac{\partial b(x_t, t)}{\partial x} \left[ \omega_{t+\Delta t} - \omega_t \right]^2 \]
\[ + \text{H.O.} \]

Get the main scheme. Thus we have
\[ \text{so correct to } O(\Delta t^2) \text{ as well as the mean.} \]

Example: Consider the stochastic equation for
\[ \frac{dX_t}{dt} = \delta + \sigma \, \circ \, dW_t \]
Then we have
\[ (2.9a) \int_t^{t+\Delta t} \circ \, dW_t = \text{Stoch.} \left[ \left( \frac{\sigma^2}{2} \right) \Delta t + \sqrt{\Delta t} \mathcal{N} \right] \]
where \( \mathcal{N} \sim N(0,1) \). After expanding this we get
\[(2.11)\]
\[S_{t+1} - S_t = 2\sqrt{\frac{\sigma^2}{\tau}}\Delta t + \sigma \sqrt{\Delta t} \xi \]
\[= S_t \Delta t + S_t \sigma \sqrt{\Delta t} + \frac{1}{2} \left( S_t - S_{t-1} \right) \Delta t + \xi \Delta t \]
\[\approx \text{H.O.}\]

The methods when these give required convergence are
still less used to implement in high dimensions, and hence are not generally used. About the model,
\[(2.12)\]
\[S_{m+1} = S_m \left( 1 + r \Delta t + \sigma \sqrt{\Delta t} \xi_m \right) .\]

Eventually if \(S_m\) is large and negative enough we can have \(S_{m+1} < 0\). Thus is a unique value since we know that \(S > 0\) for all \(t \) with prob 1.

If this happens in our scheme we set \(S_{m+1} = S_m + \Delta t \frac{1}{2} \). We have already noted that for the
simulation of just one path we need to generate
new normal variables \(S_0, S_1, \ldots, S_{m-1}, \mathcal{N} \). To estimate an expectation we must generate \(N\) of
paths and average we will need \(N \cdot \mathcal{N}\) \(\approx 6 \left( \frac{115}{N} \right) \) in the MC estimate. Not

Now there are various ways of modifying the basic MC
method so as to give faster convergence. These
methods come under the heading "variance reduction"
techniques which we should discuss in depth later.

One such method uses another random variable \(\mathcal{N}\). Thus
\[\text{Simulation 1: } S_0 \xrightarrow{\mathcal{N}} S_1, \ldots, S_{m-1}, \quad \text{Simulation 2: } S_0 \xrightarrow{\mathcal{N}} S_1, \ldots, S_{m-1}. \quad \text{Simulation 3: } S_0 \xrightarrow{\mathcal{N}} S_1, \ldots, S_{m-1}. \quad \text{Simulation 4: } \vdots \]

where \(\mathcal{N}\) is a new set of

variables \(\xi_0, \xi_1, \ldots, \xi_{m-1} \).
anti-Natali 0 in - 1 + 1_{-m} \text{ de.}

Finally, we consider higher dimensional cases, in particular a model for a stock under stochastic volatility. We consider a binary option under stochastic volatility. The stock follows the Heston model:

\begin{equation}
\frac{dS_t}{S_t} = \sigma_t \, dW_t + \sqrt{\nu_t} \, dB_t,
\end{equation}

where

\begin{equation}
\begin{align*}
\frac{d\nu_t}{\nu_t} &= [\kappa (\theta - \nu_t) + \nu_t^2 - \rho \sqrt{\nu_t}] \, dt + \rho \sqrt{\nu_t} \, dB_t \\
\text{where} \quad \nu_t, \quad W_t, \quad B_t \quad \text{are correlated BM's}.
\end{align*}
\end{equation}

\begin{equation}
E \left[ dW_t \, dW_t \right] = \sigma^2 \, dt \quad \text{and} \quad 0 < \sigma^2 < 1.
\end{equation}

Note that (2.14) is equivalent to

\begin{equation}
dW_t = \rho \, dZ_t + \sqrt{1 - \rho^2} \, dB_t,
\end{equation}

where \(Z_t\) and \(W_t\) are independent BM's. We also note that \(\nu_t > 0\) and \(\kappa \theta > \sigma^2 / 2\) are necessary. We also note that \(\gamma_t = 0\) with probability 0. \(\gamma_t < \kappa^2 / 2\) otherwise. The process \(\gamma_t\) will have finite positive time to hit \(0\) with finite probability. Let us assume the payoff on the option as:

\begin{equation}
\begin{cases}
\gamma_T > \delta_0 \quad t = 0, \quad 90 \text{ days} \\
0 \quad \text{otherwise}.
\end{cases}
\end{equation}

To solve this, we discretize \(T = 60, 12, 90\) days. \(\delta = \frac{1}{90} \text{ days} = \gamma_{90} \text{ days}\) \(m = 0, 1, 90, \text{ Applying the finite difference to the system (2.13), we have:}

\begin{equation}
\gamma_{m+1} = \gamma_m + [\kappa \theta - (\kappa + \lambda) \gamma_m] \Delta t + \rho \sqrt{\gamma_m} \Delta B_m,
\end{equation}

\(\gamma_{m+1} = \gamma_m + [\kappa \theta - (\kappa + \lambda) \gamma_m] \Delta t + \rho \sqrt{\gamma_m} \Delta B_m, \quad m = 0, 1, 90.
\)
\[ S^{n+1} = S^n \left[ 1 + \frac{\Delta t}{\beta} \int_{t_n}^{t_{n+1}} (p \frac{S^n + \sqrt{1 - \rho^2} \gamma^n}{S^n}) \mathrm{d}t \right], \]
where \( \beta \) and \( \gamma \) are \( \mathcal{N}(0, 1) \) distributed. To solve this system we need to know \( S_0 \) and \( \gamma_0 \). The value of \( S_0 \) is of course not but we would need to estimate \( \gamma_0 \) in some way. Then to do the MC simulation we proceed as follows:

(c) Generate matrices of independent \( \mathcal{N}(0, 1) \) 's
\[ \tilde{S} = (\tilde{S}^{(m)})_{m=0}^{M}, \quad \tilde{\gamma} = (\tilde{\gamma}^{(m)})_{m=0}^{M}, \quad m = 0, 1, \ldots, M-1, \quad n = 0, \ldots, N. \]

(b) Compute path \( S, \gamma \) as in (2.17), \( 0 \leq n \leq M. \)

(c) Compute discounted payoff,
\[ \tilde{V}^{(n)} = \begin{cases} e^{-r \Delta t} \left( \tilde{S}^{(M-50)} + \min_m \tilde{S}^{(m)} \right) > 20 & \text{if } \min_m \tilde{S}^{(m)} \leq 20, \\ 0 & \text{if } \min_m \tilde{S}^{(m)} > 20. \end{cases} \]

(d) Compute averages \( \tilde{V}^{(n)} = \frac{1}{n} \sum_{i=1}^{n} \tilde{V}^{(i)} \) and the standard error \( \tilde{E}_n. \)

(e) Report \( \tilde{V}_n, \tilde{E}_n \) and converage diagram.

§ 2 Correlation

Two random variables are considered to be correlated if knowing the values of one is useful in predicting the other. The correlation of two random variables is measured. If \( X, Y \) are two random variables, then the covariance of \( X, Y \) is

\[ \text{Cov} [X, Y] = E \left\{ (X - E[X]) (Y - E[Y]) \right\} \]
\[ E[XY] - E[X]E[Y]. \]

Evidently the variance is related to covariance by

\[ \text{Var}[X] = \text{Cov}[X,X]. \]

We can also define the coefficient of correlation \( p \) by

\[ p = \frac{\text{Cov}[X,Y]}{\sqrt{\text{Var}[X]\text{Var}[Y]}}. \]

Note that by the Schwartz inequality

\[ \text{Cov}[X,Y] \leq \sqrt{\text{Var}[X]\text{Var}[Y]}. \]

Here \(-1 \leq p \leq 1\). If \( p > 0 \) the random variables are said to be positively correlated. If \( p < 0 \) they are said to be negatively correlated.

In what follows let us assume that \( X \) and \( Y \) have a bivariate normal distribution.

Then

\[ \text{Cov}[X,Y] = -E[(X-E[X])(Y-E[Y])]. \]

This leads immediately to

\[ \text{Var}[X] = E[(X-E[X])^2]. \]

\[ \text{Var}[Y] = E[(Y-E[Y])^2]. \]

\[ \text{Var}[X] = E[(X-E[X])^2] = p^2 \text{Var}[X] + \sigma_X^2 \]

\[ \text{Var}[Y] = E[(Y-E[Y])^2] = p^2 \text{Var}[Y] + \sigma_Y^2 \]

\[ p^2 \text{Var}[X] = \sigma_X^2 \]

\[ p^2 \text{Var}[Y] = \sigma_Y^2 \]

Thus \( p^2 = \frac{\sigma_X^2}{\sigma_Y^2} \).

Also

\[ \text{Cov}[X,Y] = \frac{\sigma_X \sigma_Y R}{\sqrt{\text{Var}[X]\text{Var}[Y]}}. \]

\[ \text{Cov}[X,Y] = \frac{\sigma_X \sigma_Y R}{\sqrt{\text{Var}[X]\text{Var}[Y]}} = p \sigma_X \sigma_Y. \]

\[ p \sigma_X \sigma_Y = \sigma_X \sigma_Y \rho. \]

\[ p = \rho. \]

\[ p = \frac{\sigma_X \sigma_Y R}{\sigma_X \sigma_Y}. \]

\[ p = \frac{R}{\sqrt{\text{Var}[X]\text{Var}[Y]}}. \]

\[ p = \frac{R}{\sigma_X \sigma_Y}. \]
Here, (3.8) \( p \) = well of cash, for \( W(1+\sigma_1 \cdot W(T)^{1/2}) \), let us consider now how to price a basket option on an option which depends on the price of news of some stock. We shall assume that

\[
\begin{align*}
\frac{dS_1}{S_1} &= r dt + \sigma_1 \, dW_{1,t} + \sigma_2 \, dW_{2,t}, \\
\frac{dS_2}{S_2} &= r dt + \sigma_2 \, dW_{1,t} + \sigma_2 \, dW_{2,t}
\end{align*}
\]

where \( W_{1,t} \) and \( W_{2,t} \) are independent R.W.'s. We solve (3.9) to obtain

\[
\begin{align*}
S_{1,T} &= S_{1,0} \exp \left[ \left( r - \frac{1}{2} \sigma_1^2 \right) T + \sigma_1 \sqrt{T} \xi_1 + \frac{1}{2} \sigma_1^2 T \right] \\
S_{2,T} &= S_{2,0} \exp \left[ \left( r - \frac{1}{2} \sigma_2^2 \right) T + \sigma_2 \sqrt{T} \xi_2 + \frac{1}{2} \sigma_2^2 T \right]
\end{align*}
\]

where \( \xi_1, \xi_2 \) are indep. \( N(0,1) \) r.v's. Note that formula (3.10) can be derived by observing that \( S_{1,T} \) are exponentials of normal variables and

\[
E[S_{1,T}] = E[S_{2,T}] = e^{rT}.
\]

We could not make the value of an option depend on the arithmetic mean of the geometric mean, i.e., \( \left( S_{1,T} + S_{2,T} \right) / 2 \) as the geometric mean.
(a) Let \( Z \sim \mathcal{N}(0,1) \) be \( N \) independent standard normal variables.
\( Z_1, Z_2, ..., Z_N \).

(b) Compute the terminal stock price
\[
Z_{1,t} = \sum_{i=1}^{N} \exp \left( \left( r - \frac{1}{2} \sigma_i^2 \right) T + \sigma_i \sqrt{T} F_{1+i} \right)
\]
\[
Z_{2,t} = \sum_{i=1}^{N} \exp \left( \left( r - \frac{1}{2} \sigma_i^2 \right) T + \sigma_i \sqrt{T} F_{1+i} \right)
\]

(c) Compute the discounted payoff,
\[
Y_i = \exp \left( -rT \right) \left( Z_{1,t} + Z_{2,t} - K \right).
\]

(d) Compute \( \hat{Y}_n = \frac{1}{n} \sum_{i=1}^{n} Y_i \) and
\[
\hat{e}_n = \frac{1}{n \hat{Y}_n} \sum_{i=1}^{n} (Y_i - \hat{Y}_n)^2
\]

(e) Repeat \( \hat{Y}_n, \hat{e}_n \) and a convergence diagram.

Thus will give us the value of the call option with strike \( K \) on the arithmetic mean basket of \( N \) stock \( \text{strikes} \). Certainly one can generalize the foregoing to baskets of many \( \text{strikes} \). Thus in case of stocks
\[ s_i \]  

**Eqs. 3.12**  

\[
\frac{d s_i}{d t} = \gamma s_i + \sigma_i \frac{d W_i}{d t}
\]

\[
\frac{d s_i}{d t} = \alpha s_i + \sigma_i \frac{d W_i}{d t}
\]

He \( W_i, t, \ldots, W_i, t \) are correlated \( R \) - i.i.d.

**Eqs. 3.13**  

[\[ E \left[ d W_i + d W_j \right] = \delta_{ij} dt \text{, where} \]

\[ \delta_{ij} = 1, \quad 1 \leq i, j \leq d, \]

\[ -1 \leq \delta_{ij} = \delta_{ji} \leq 1, \quad 1 \leq i, j \leq d \]

\[ R \] is the correlation matrix,

**Eqs. 3.14**  

\[
R = \begin{bmatrix} \rho_{11} & \ldots & \rho_{1d} \\ \vdots & \ddots & \vdots \\ \rho_{d1} & \ldots & \rho_{dd} \end{bmatrix}
\]

where \( \rho_{ij} \) is the correlation between \( W_i \) and \( W_j \), \( 1 \leq i, j \leq d \).

\[ R \] is a symmetric, positive definite matrix.

Note: Positive definite means

**Eqs. 3.15**  

\[
\sum_{i=1}^{d} \rho_{ii} x_i^2 > 0 \quad \text{for all vectors} \quad x_i \end{bmatrix} \text{ and } \begin{bmatrix} x_{1i} \quad x_{2i} \end{bmatrix}, x \in \mathbb{R}^d.
\]

Thus, follows from (3.12) since

**Eqs. 3.16**  

\[
\sum_{i=1}^{d} s_i \left( d W_i(t) + 3 \right)^2
\]

Hence \( R \) can be factored as

**Eqs. 3.17**  

\[ R = A A^T \]  

On shall note that any matrix of the form (3.17) is symmetric positive definite. The factorization (3.18) is not necessarily
\[ M_{23} \]

\[ M_{11} = A \epsilon_{11} + A \epsilon_{12} \]

\[ \epsilon_{ij} = (A A^T)^{i-j} \]

\[ \frac{d^2}{dt^2} \epsilon_{ij} \]

\[ (\epsilon_{ij})^T = \epsilon_{ji} \]

\[ (3.11) \]

\[ 2 \epsilon_{ij} = 2 \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.20) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.21) \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ (3.12) \]

\[ 2 \epsilon_{ij} = 2 \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (3.11) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.20) \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ (3.12) \]

\[ 2 \epsilon_{ij} = 2 \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.21) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (3.11) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.20) \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ (3.12) \]

\[ 2 \epsilon_{ij} = 2 \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.21) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (3.11) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.20) \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ (3.12) \]

\[ 2 \epsilon_{ij} = 2 \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.21) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (3.11) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.20) \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ (3.12) \]

\[ 2 \epsilon_{ij} = 2 \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.21) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (3.11) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.20) \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ (3.12) \]

\[ 2 \epsilon_{ij} = 2 \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.21) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (3.11) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.20) \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ (3.12) \]

\[ 2 \epsilon_{ij} = 2 \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.21) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (3.11) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.20) \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ A = \frac{A}{2} + \frac{A}{2} \]

\[ (3.12) \]

\[ 2 \epsilon_{ij} = 2 \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]

\[ (2.21) \]

\[ \frac{d}{dt} \epsilon_{ij} = \frac{d}{dt} \epsilon_{ij} \]

\[ \epsilon_{ij} = \epsilon_{ij} \]
(3.22) \[ S_{i, T} = S_{i, 0} \exp \left[ \left( r - \frac{1}{2} \sum_{j=1}^{d} \sigma_{i,j}^2 \right) T + \sum_{j=1}^{d} \sigma_{i,j} \xi_j \right] \]

where \( S_{i, T} \) are \( i.i.d. \) \( N(0, 1) \) variables. Now we can do an MC simulation. As input the values of \( \theta \) and \( \sigma \) as we did before.

\[ S_i \]

### Variance Reduction

We first consider the use of anti-parallel variables as already mentioned. Suppose we use a function \( g : \mathbb{R} \to \mathbb{R} \) and we wish to estimate \( \mathbb{E}[g(X)] \) where \( X \) is \( N(0, 1) \) by MC. Thus we run \( n \) simulations with \( \xi_1, \ldots, \xi_n \) \( N(0, 1) \) variables and then we put

\[ (4.1) \quad \mathbb{E}[g(X)] \sim \frac{1}{n} \left\{ g(\xi^{(1)}) + g(\xi^{(2)}) \right\} \]

This is the standard MC method. The role of convergence of the RHS is the LHS is determined by the variance of the RHS, which is

\[ (4.2) \quad \text{Var} \left[ \frac{1}{n} \left\{ g(\xi^{(1)}) + g(\xi^{(2)}) \right\} \right] = \frac{1}{n^2} \text{Var} \left[ g(\xi) \right], \]

where we use anti-parallel variables to estimate \((4.1)\). Then we have

\[ (4.3) \quad \mathbb{E}[g(X)] \sim \frac{1}{2n} \left\{ g(\xi^{(1)}) + g(-\xi^{(1)}) + g(\xi^{(2)}) + g(-\xi^{(2)}) \right\} \]

The variance of the RHS here is given by
We define
\[
\begin{align*}
\Var \left[ \psi (\xi) - \psi (-\xi) \right] &= \Var \left[ \psi (\xi) + \psi (-\xi) \right] \\
&= \frac{1}{4N} \Var \left[ \psi (\xi) + \psi (-\xi) \right],
\end{align*}
\]

where the use of anti-hermite variables gives a reduction in variance.

This is the same as
\[
\begin{align*}
\Var \left[ \psi (\xi) + \psi (-\xi) \right] &= 2 \Var \left[ \psi (\xi) \right],
\end{align*}
\]

which is evidently the same as
\[
\begin{align*}
\Var \left[ \psi (\xi) \psi (\xi - \xi) \right] - \Var \left[ \psi (\xi) \right]^2 < 0.
\end{align*}
\]

Note that in the case of a univariate put option, for example, we have \( \psi (\xi) \) given by
\[
\begin{align*}
\psi (\xi) = \begin{cases} K - S_0 \exp \left[ \left( \frac{\xi}{2} - \frac{1}{4} \right) T + \frac{1}{8} \xi^2 \right] & \xi > 0, \\
0 & \xi = 0,
\end{cases}
\end{align*}
\]

which is not even a monotonic function of \( \xi \).

Another technique for variance reduction is the use of control variates. The idea is as follows: Suppose we wish to calculate the expectation of a function \( \Phi (\xi) \) of the \( \psi (\xi) \) variables. Suppose in addition that we can evaluate a function \( g : \mathbb{R} \rightarrow \mathbb{R} \) and we can estimate \( g \hat{\psi} = \mathbb{E}[g \psi (\xi)] \) accurately. We modify the MC method as follows:

Estimating \( \mathbb{E}[g \psi (\xi)] \) by using an unbiased estimate of \( g \hat{\psi} \). Thus \( \mathbb{E}[\xi_1^{(1)} \psi (\xi_1^{(1)}) | \xi_1^{(1)}] = 0 \), we are interested
(4.10) \[ \mathbb{E}[g(l)] = \frac{1}{2} \sum_{i=0}^2 \mathbb{E}\{g^{(i)}\} + \beta \mathbb{E}\{g^{(2)}\} \]

where \( \beta \) is suitably chosen. It is easy to see that the variance of the RHS is minimized when we put

(4.11) \[ \beta = \text{cov} \left[ \mathbb{E}[g(l)], g(0) \right] / \text{var} \left[ g(0) \right]. \]

We can now not know the exact value of \( \beta \), but it can be estimated from the same simulation as we are using to estimate \( \mathbb{E}[g(l)] \).

An example of this is found in estimating the value of an American option. Suppose we have \( 0 \leq t_0 < t_1 < \ldots < t_n \leq T \) and the option is a put with payoff

(4.12) \[ \text{option payoff} = \left[ X - \frac{1}{T+1} \sum_{i=0}^n S_{t_i} \right]^+ \]

Our control variable is the same kind of option but replacing the \( n \)th node value mean by the geometric mean

(4.13) \[ \text{control variable} = \left[ X - \left( \frac{1}{T+1} \sum_{i=0}^n S_{t_i} \right) \right]^+ \]

since the \( n \)th node mean is geometric it follows that

(4.14) \[ \text{control variable} > \text{option payoff} \]

The advantage of the control variable (4.13)

\[ \text{is that the geometric mean is the expected } \]

\[ \text{value of a random variable}. \]

\[ \text{Here the mean of } (4.13) \]

\[ \text{is computed as just by a } \text{BI formula}. \]

\[ \text{We could of course estimate } \beta \text{ but the simplest is } A \]
contribution to the MC simulation, and hence note that
of the MC simulations will decrease relative to
the mean of a large factor.

Example: Consider an asset of the money put in
the Black-Scholes model. The parameters are

\[ K = 50, \ T = 1 \ \text{year}, \ \delta = 0.2, \ \sigma = 20\%, \ r = 10. \]

Then we have

\[ I_T = 80 \exp \left[ 0.2 \sigma T \right], \quad \xi \sim N(0,1), \]

Thus

\[ P \left( I_T < K \right) = P \left( \xi < \frac{K - 50}{20 \sigma \sqrt{T}} \right) = P \left( \xi < -2.25 \right) \approx 0.025, \]

so only 2.5% of paths end up in the money,
where the relative standard error of the estimate
is high. We have that

\[ \sigma_T = \sigma \left( \frac{K - 80 \exp \left[ (r - \delta)T + 0.5 \sigma^2 T \right]}{80} \right) \]

We put \( \xi = \chi - 2 \) and choose \( \chi = 0 \)

so that

\[ K = 80 \exp \left[ (r - \delta)T - 0.5 \sigma^2 T \right], \]

\( \eta = 0 \) corresponds to the option being in
the money. \( \eta = 0 \) corresponds to the option being
in the money. As in our case, this means \( \delta = 2.25. \)

Now we do our MC estimate with

\[ \gamma^{(n)} = \exp \left[ -\left( K - 80 \exp \left\{ (r - \delta)T \right\} \right) \right. \]

\[ + \left. 0.5 \left( (\gamma^{(n)} - 2) \right)^2 \right\} \exp \left[ 2 \gamma^{(n)} - \frac{1}{2} 2^2 \right] \]

Then our MC estimate \( \hat{\gamma} \) as usual

\[ \hat{\gamma} = \frac{1}{n} \sum_{i=1}^{n} \gamma^{(i)}, \quad 1 \leq \gamma \leq N, \]
Another variance reduction technique is called **importance sampling**. The previous technique of importance sampling was geared towards estimating \( E[g(\xi)] \) when \( g(\xi) \) was a rapidly varying function and concentrated around some particular value, say \( \xi = \xi_0 \). Here we are in an opposite situation where we assume \( g(\xi) \) is slowly varying and the values of \( g(\xi) \) for a large range of \( \xi \) contribute to the expectation. In this region the standard MC estimate can be divided into the region of \( \xi \) space into \( n \) regions of equal probability. Thus we pick \( n \) points at random \( \xi_1, \ldots, \xi_n \) in this region and divide the expectation

\[
E[g(\xi)] = \sum_{i=1}^{n} \frac{g(\xi_i)}{p(\xi_i)}
\]

(4.23) \( P(\xi_i) \), \( k = 1, \ldots, n \)

Thus, when \( \sum_{i=1}^{n} \frac{1}{p(\xi_i)} = 1 \), we do not have a variable \( \xi \) distributed in such a way that \( \xi_i \) is

\[
E[g(\xi)] = \sum_{i=1}^{n} \frac{g(\xi_i)}{p(\xi_i)}
\]

(4.24) \( P(\xi_i) \), \( k = 1, \ldots, n \)

Thus the variance can be reduced by

\[
E[g(\xi)] = \sum_{i=1}^{n} \frac{g(\xi_i)}{p(\xi_i)}
\]

(4.25) \( M \) \( (P(\xi_i) \text{ a random var}) \), \( k = 1, \ldots, n \)

Evidently

\[
E[g(\xi)] = \sum_{i=1}^{n} \frac{g(\xi_i)}{p(\xi_i)}
\]

(4.26) \( P(\xi_i) \), \( k = 1, \ldots, n \)
Thus we can do an MC simulation by choosing independently \( x_i^{(n)} \), \( i = 1, \ldots, M \), \( n = 1, \ldots, N \), where \( x_i^{(n)} \) have the distribution of \( x_i \).

The MC simulation gives

\[
E \left[ g(x) \right] \approx \frac{1}{MN} \sum_{n=1}^{N} \sum_{i=1}^{M} g(x_i^{(n)}).
\]

This will be an effective method if the variation of \( g \) within each region \([a_i, a_{i+1}]\) is small, but the variation between different regions is quite large. We need to ask finally how we would generate random variables with the distribution of the \( x_i \).

This is quite easy if we can do the inversion of the cdf of the normal variable as mentioned before.

To choose a uniform variable \( U \sim U(0, 1) \), we define \( x_i \) in terms of \( U \\sim U(0, 1) \) by the equation

\[
F(x_i) = \frac{i-1}{M}, \quad i = 1, \ldots, M.
\]

Next we can use the method of moment matching. Consider the standard MC method for estimating \( E \left[ g(x) \right] \). Thus we pick \( N \) independent \( N(0,1) \) variables \( x_i^{(n)} \), \( n = 1, \ldots, N \) and put

\[
E \left[ g(x) \right] \approx \frac{1}{N} \sum_{n=1}^{N} g(x_i^{(n)}).
\]

From the sample \( x_i^{(1)}, \ldots, x_i^{(N)} \) of \( N \) independent normal variables we unbiased estimates of the mean and variance of \( x \) as

\[
\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i^{(n)}, \quad \hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left( x_i^{(n)} - \hat{\mu} \right)^2.
\]
Thus we have that

\[ (4.31) \quad E[\hat{x}] = 0, \quad E[\hat{x}^2] = 1. \]

The idea of moment matching is to modify

the simple \pi\ in \text{(4.31)}\ holds in the

sample mean and variance as given by \text{(4.32)}. Thus

let match the \text{r}-\text{th} moment, we define a new

variable \( \hat{z} \), \( n = 1, \ldots, \text{r} \), by

\[ (4.32) \quad \hat{z}(n) = \frac{1}{\sqrt{n}} \left[ \hat{t} - \hat{\mu} \right], \quad n = 1, \ldots, \text{r}, \]

and we use the variables \( \hat{z}(n) \) to do the MC simulation. The degree to which this is an

improvement is unclear.

Finally, we discuss the Brownian Bridge

process. The Brownian Bridge process on the

interval \([0, T]\) is the process \( W_t, 0 \leq t \leq T \)

with conditioning not only on \( W_0 = 0 \) but

also \( \text{at the last time} \ W_T = a \). It is

Thus

a Brownian process normally and hence can

be simulated using MC with sampling

variables. The advantage of this is that

often times it is the final value which is

most important in determining the final

value of an asset. Using Brownian Bridge means we can

use importance sampling for example on \( W_T \)

and then simulate. The part of the path with

normal variables as well. The basic idea of

the Brownian Bridge is as follows: The

complete path with \( W_0 = 0, W_T = a \) is

in the linear path.
Brownian bridge. The RB is a random process with 300 men organised on the fibre path. The key fact is that for $t^1 < t < t^2$, the distribution of $W_t$ conditioned on $W_{t^1}$ and $W_{t^2}$ is normal with mean

$$ (4.23) \quad \text{mean} = \frac{t - t^1}{t^2 - t^1} W_{t^1} + \frac{t - t^2}{t^2 - t^1} W_{t^2}. $$

(4.44) variance $= \frac{(t - t^2)(t - t^1)}{(t^2 - t^1)}.$

Note: The mean is just a linear function of $t$ from $(t^1, W_{t^1})$ to $(t^2, W_{t^2})$. The fact that $W_t$ is normal with variance requires some work. Observe however what The formula (4.44) agrees with our intuition.

(4.45) variance of $W_t$ conditioned on $W_{t^1}, W_{t^2} < \text{variance of } W_t \text{ conditioned on } W_{t^1}$, which is also zero as

$$ (4.46) \quad \text{variance} = \frac{(t - t^2)(t - t^1)}{(t^2 - t^1)} < \frac{(t^2 - t)}{(t^2 - t^1)}, $$

i.e. $t > t^1$. Now RB is particularly easy to construct to any degree of accuracy by introducing new auxiliary variables at a time. Suppose we want to construct $W_{t^1}, 0 < t < t^2$, with $W_0$ fixed. Then choose $M = 2^k$.

(4.47) $W_{12} = \frac{1}{2} \left[ W_0 + W_{T^2} + \left( \frac{t^2}{4} \right) \bar{X} \right].$
We can then generate \( W_T \) from \( W^{1/2} \) and \( W^{1/4} \) from \( W^{1/2} \), \( W_T \) in a similar way, i.e.,

\[
W_{3T/4} = \frac{1}{2} \left[ W_{T/4} + W_{T/2} \right] + \left( \frac{T/2}{4} \right)^{1/2} \xi_2 + \xi_3
\]

\[
W_{T/4} = \frac{1}{2} \left[ W_{T/8} + W_{T/2} + \left( T/8 \right)^{1/2} \xi_3 \right].
\]

Example: An investor wants to support the optimal way with a 4-period period and the payoffs

\[
\text{payoffs} = \left[ \frac{1}{5} \left( S_0 + S_1 + S_2 + S_3 + S_4 \right) \right].
\]

Clearly, \( S_4 \) has a weighting of \( 1/5 \) in the payoffs whereas \( S_0, S_1, S_2, S_3 \) each have a weighting just \( 1/5 \). Thus, it makes sense to make sure to have a good simulation of \( S_4 \) as an exponential of a normal variable. We had \( \mu_4 \) and \( \sigma_4 \),

\[
S_m = \exp \left[ (\mu_4 - \sigma_4^2) m + \sigma_4 W_m \right],
\]

and we generate the paths via \( \xi_k \) i.e.,

(a) For \( N \) paths generate \( N \) independent \( \xi_k \),

\[
\xi_m, 1 \leq n \leq M, \quad m = 1, 2, 3, 4,
\]

(b) Compute paths \( W_k \) and \( S_k \) follows:

\[
W_0 = 0, \quad W_k = \xi_k \xi_{k-1},
\]

\[
W_2 = \frac{1}{2} \left( W_0 + W_4 \right) + \sqrt{\frac{2}{4}} \xi_2
\]

\[
W_1 = \frac{1}{2} \left( W_0 + W_2 \right) + \sqrt{\frac{2}{4}} \xi_1
\]

\[
W_3 = \frac{1}{2} \left( W_1 + W_4 \right) + \sqrt{\frac{2}{4}} \xi_3
\]

Then forced with the MC simulation as usual.