Chapter 4

Variance Reduction Techniques

Introduction.

In this chapter we discuss techniques for improving on the speed and efficiency of a simulation, usually called "variance reduction techniques".

Much of the simulation literature concerns *discrete event simulations* (DES), simulations of systems that are assumed to change instantaneously in response to sudden or discrete events. These are the most common in operations research and examples are simulations of processes such as networks or queues. Simulation models in which the process is characterized by a state, with changes only at discrete time points are DES. In modeling an inventory system, for example, the arrival of a batch of raw materials can be considered as an event which precipitates a sudden change in the state of the system, followed by a demand some discrete time later when the state of the system changes again. A system driven by differential equations in continuous time is an example of a DES because the changes occur continuously in time. One approach to DES is *future event*

simulation which schedules one or more future events at a time, choosing the event in the future event set which has minimum time, updating the state of the system and the clock accordingly, and then repeating this whole procedure. A stock price which moves by discrete amounts may be considered a DES. In fact this approach is often used in valuing American options by Monte Carlo methods with binomial or trinomial trees.

Often we identify one or more *performance measures* by which the system is to be judged, and *parameters* which may be adjusted to improve the system performance. Examples are the delay for an air traffic control system, customer waiting times for a bank teller scheduling system, delays or throughput for computer networks, response times for the location of fire stations or supply depots, etc. Performance measures again are important in engineering examples or in operations research, but less common in finance. They may be used to calibrate a simulation model, however. For example our performance measure might be the average distance between observed option prices on a given stock and prices obtained by simulation from given model parameters. In all cases, the *performance measure* is usually the expected value of a complicated function of many variables, often expressible only by a computer program with some simulated random variables as input. Whether these input random variables are generated by inverse transform, or acceptance-rejection or some other method, they are ultimately a function of uniform[0,1] random variables $U_1, U_2, ...$ These uniform random variables determine such quantities as the normally distributed increments of the logarithm of the stock price. In summary, the simulation is used simply to estimate a multidimensional integral of the form

$$E(g(U_1, ..., U_d)) = \int \int .. \int g(u_1, u_2, ...u_d) du_1 du_2 \dots du_d \tag{4.1}$$

over the unit cube in d dimensions where often d is large.

As an example in finance, suppose that we wish to price a European option on a stock price under the following *stochastic volatility* model.

Example 33 *Suppose the daily asset returns under a risk-neutral distribution is assumed to be a variance mixture of the Normal distribution, by which we mean that the variance itself is random, independent of the normal variable and follows a distribution with moment generating function* $s(s)$*. More specifically assume under the* Q *measure that the stock price at time* $n\Delta t$ *is determined from*

$$S_{(n+1)\Delta t} = S_{n\Delta t}\frac{\exp\{r\Delta t + \sigma_{n+1}Z_{n+1}\}}{m(\frac{1}{2})}$$

where, under the risk-neutral distribution, the positive random variables σ_i^2 *are assumed to have a distribution with moment generating function* $m(s) = E\{\exp(s\sigma_i)\}$*,* Z_i *is standard normal independent of* σ_i^2 *and both* (Z_i, σ_i^2) *are independent of the process up to time* $n\Delta t$*. We wish to determine the price of a European call option with maturity* T*, and strike price* K*.*

It should be noted that the rather strange choice of $m(\frac{1}{2})$ in the denominator above is such that the discounted process is a martingale, since

$$\begin{aligned} E\left[\frac{\exp\{\sigma_{n+1}Z_{n+1}\}}{m(\frac{1}{2})}\right] &= E\{E\left[\frac{\exp\{\sigma_{n+1}Z_{n+1}\}}{m(\frac{1}{2})}|\sigma_{n+1}\right]\} \\ &= E\{\frac{\exp\{\sigma_{n+1}^2/2\}}{m(\frac{1}{2})}\} \\ &= 1. \end{aligned}$$

There are many ways of simulating an option price in the above example, some much more efficient than others. We might, for example, simulate all of the $2n$ random variables $\{\sigma_i, Z_i, i = 1, ..., n = T/\Delta t\}$ and use these to determine the simulated value of S_T, finally averaging the discounted payoff from the option in this simulation, i.e. $e^{-rT}(S_T - K)^+$. The price of this option at time 0 is the average of many such simulations (say we do this a total of N times) discounted to present,

$$\overline{e^{-rT}(S_T - K)^+}$$

where $\overline{x}$ denotes the average of the $x's$ observed over all simulations. This is

a description of a crude and inefficient method of conducting this simulation. Roughly the time required for the simulation is proportional to $2Nn$, the total number of random variables generated. This chapter discusses some of the many improvements possible in problems like this. Since each simulation requires at least $d = 2n$ independent uniform random variables to generate the values $\{\sigma_i, Z_i, i = 1, ..., n\}$ then we are trying to estimate a rather complicated integral of the form 4.1 of high dimension d. In this case, however, we can immediately see some obvious improvements. Notice that we can rewrite S_T in the form

$$S_T = S_0 \frac{\exp\{rT + \sigma Z\}}{m^n(\frac{1}{2})} \tag{4.2}$$

where the random variable $\sigma^2 = \sum_{i=1}^n \sigma_i^2$ has moment generating function $m^n(s)$ and Z is independent standard normal. Obviously, if we can simulate σ directly, we can avoid the computation involved in generating the individual σ_i. Further savings are possible in the light of the Black-Scholes formula which provides the price of a call option when a stock price is given by (4.2) and the volatility parameter σ is non-random. Since the expected return from the call under the risk-neutral distribution can be written, using the Black-Scholes formula,

$$E(e^{-rT}(S_T - K)^+) = E\{E[e^{-rT}(S_T - K)^+|\sigma]\}$$
$$= e^{-rT} E\{S_0 \Phi(\frac{\log(S_0/K) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}) - Ke^{-rT}\Phi(\frac{\log(S_0/K) + (r - \frac{\sigma^2}{2})T}{\sigma\sqrt{T}})\}$$

which is now a one-dimensional integral over the distribution of σ. This can now be evaluated either by a one-dimensional numerical integration or by repeatedly simulating the value of σ and averaging the values of

$$e^{-rT} S_0 \Phi(\frac{\log(S_0/K) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}) - Ke^{-rT}\Phi(\frac{\log(S_0/K) + (r - \frac{\sigma^2}{2})T}{\sigma\sqrt{T}})$$

obtained from these simulations. As a special case we might take the distribution of σ_i^2 to be Gamma$(\alpha\Delta t, \beta)$ with moment generating function

$$m(s) = \frac{1}{(1 - \beta s)^{\alpha\Delta t}}$$

in which case the distribution of σ^2 is Gamma$(\alpha T, \beta)$. This is the so-called "variance-gamma" distribution investigated extensively by ..... and originally suggested as a model for stock prices by .....Alternatively many other wider-tailed alternatives to the normal returns model can be written as a variance mixture of the normal distribution and option prices can be simulated in this way. For example when the variance is generated having the distribution of the *reciprocal of a gamma* random variable, the returns have a student's t distribution. Similarly, the stable distributions and the Laplace distribution all have a representation as a variance mixture of the normal.

The rest of this chapter discusses "variance reduction techniques" such as the one employed above for evaluating integrals like (4.1), beginning with the much simpler case of an integral in one dimension.

Variance reduction for one-dimensional Monte-Carlo Integration.

We wish to evaluate a one-dimensional integral $\int_0^1 f(u)du$, which we will denote by θ using by Monte-Carlo methods. We have seen before that whatever the random variables that are input to our simulation program they are usually generated using uniform[0,1] random variables U so without loss of generality we can assume that the integral is with respect to the uniform[0,1] probability density function, i.e. we wish to estimate

$$\theta = E\{f(U)\} = \int_0^1 f(u)du.$$

One simple approach, called *crude Monte Carlo* is to randomly sample $U_i \sim Uniform[0,1]$ and then average the values of $f(U_i)$ obtain

$$\hat{\theta}_{CR} = \frac{1}{n}\sum_{i=1}^{n} f(U_i).$$

It is easy to see that $E(\hat{\theta}_{CR}) = \theta$ so that this average is an *unbiased estimator* of the integral and the variance of the estimator is

$$var(\hat{\theta}_{CR}) = var(f(U_1))/n.$$

Example 34 *A crude simulation of a call option price under the Black-Scholes model:*

For a simple example that we will use throughout, consider an integral used to price a call option. We saw in Section 3.8 that if a European option has payoff $V(S_T)$ *where* S_T *is the value of the stock at maturity* T*, then the option can be valued at present* $(t = 0)$ *using the discounted future payoff from the option under the risk neutral measure;*

$$e^{-rT}E[V(S_T)] = e^{-rT}E[V(S_0e^X)]$$

where, in the Black-Scholes model, the random variable $X = \ln(S_T/S_0)$ *has a normal distribution with mean* $rT - \sigma^2T/2$ *and variance* σ^2T*. A normally distributed random variable* X *can be generated by inverse transform and so we can assume that* $X = \Phi^{-1}(U; rT - \frac{\sigma^2}{2}T, \sigma^2T)$ *is a function of a uniform*$[0, 1]$ *random variable* U *where* $\Phi^{-1}(U; rT - \frac{\sigma^2}{2}T, \sigma^2T)$ *is the inverse of the normal* $(rT - \sigma^2T/2,\ \sigma^2T)$ *cumulative distribution function. Then the value of the option can be written as an expectation over the distribution of the uniform random variable* U*,*

$$E\{f(U)\} = \int_0^1 f(u)du$$

$$\text{where } f(u) = e^{-rT}V(S_0 \exp\{\Phi^{-1}(U; rT - \frac{\sigma^2}{2}T, \sigma^2T)\})$$

This function is graphed in Figure 4.1 in the case of a simple call option with strike price K*, with payoff at maturity* $V(S_T) = (S_T - K)^+$*, the current stock price* $S_0 = \$10$*, the exercise price* K *is* $\$10$*, the annual interest rate* $r = 5\%$*, the maturity is three months or one quarter of year* $T = 0.25$*, and the annual volatility* $\sigma = 0.20$*.*

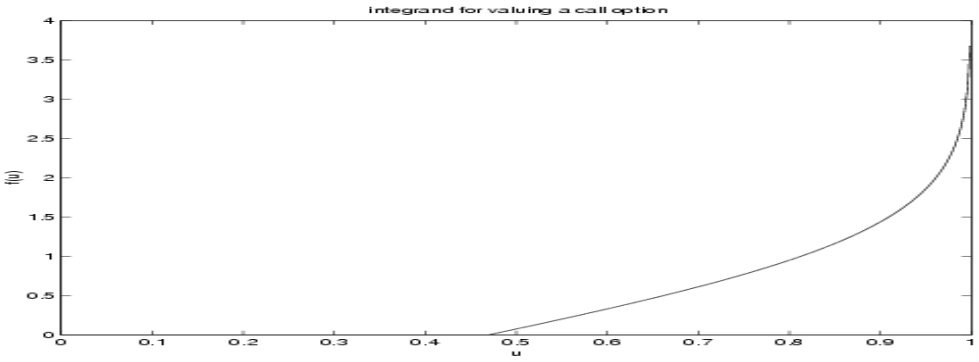

Figure 4.1: The function $f(u)$ whose integral provides the value of a call option

A simple crude Monte Carlo estimator corresponds to evaluating this function at a large number of randomly selected values of $U_i \sim U[0,1]$ and then averaging the results. For example the following function in Matlab accepts a vector of inputs $u = (U_1, ..., U_n)$ assumed to be Uniform[0,1], outputs the values of $f(U_1), ...f(U_n)$ which can be averaged to give $\hat{\theta}_{CR} = \frac{1}{n}\sum_{i=1}^{n} f(U_i)$.

```
function v=fn(u)
% value of the integrand for a call option with exercise price ex, r=annual interest
rate,
%sigma=annual vol, S0=current stock price.
% u=vector of uniform (0,1) inputs to
%generate normal variates by inverse transform. T=maturity
S0=10 ;K=10;r=.05; sigma=.2 ;T=.25 ; % Values of parameters
ST=S0*exp(norminv(u,r*T-sigma^2*T/2,sigma*sqrt(T)));
%  ST =S_0 exp{Φ^{-1}(U; rT − σ^2/2 T, σ^2 T)} is  stock price at time T
v=exp(-r*T)*max((ST-ex),0);    %  v is the discounted to present payoffs from the
call option
```

and the analogous function in **R,**

```
fn<-function(u,So,strike,r,sigma,T){
```

```
# value of the integrand for a call option with exercise price=strike, r=annual interest rate,
# sigma=annual volatility, So=current stock price, u=uniform (0,1) input to generate normal variates
# by inverse transform. T=time to maturity. For Black-Scholes price, integrate over (0,1).
x<-So*exp(qnorm(u,mean=r*T-sigma^2*T/2,sd=sigma*sqrt(T)))
v<-exp(-r*T)*pmax((x-strike),0)
v}
```

In the case of initial stock price \$10, exercise price=\$10, annual vol=$0.20, r =$ 5%, $T = .25$ (three months), this is run as

u=rand(1,500000); mean(fn(u)) and in **R**, mean(fn(runif(500000),So=10,strike=10,r=.05,sigma=.2,T=

and this provides an approximate value of the option of $\hat{\theta}_{CR} = 0.4620$. The standard error of this estimator, computed using the formula (**??**) below, is around $\sqrt{8.7 \times 10^{-7}}$. We may confirm with the black-scholes formula, again in *Matlab*,

$$[CALL,PUT] = BLSPRICE(10,10,0.05,0.25,0.2,0).$$

The arguments are, in order $(S_0.K, r, T, \sigma, q)$ where the last argument (here $q = 0$) is the annual dividend yield which we assume here to be zero. Provided that no dividends are paid on the stock before the maturity of the option, this is reasonable. This Matlab command provides the result *CALL = 0.4615 and* PUT = 0.3373 indicating that our simulated call option price was reasonably accurate- out by 1 percent or so. The *put option* is an option to sell the stock at the specified price \$10 at the maturity date and is also priced by this same function.

One of the advantages of Monte Carlo methods over numerical techniques is that, because we are using a sample mean, we have a simple estimator of accuracy. In general, when n simulations are conducted, the accuracy is measured

by the standard error of the sample mean. Since

$$var(\hat{\theta}_{CR}) = \frac{var(f(U_1))}{n},$$

the standard error of the sample mean is the standard deviation or

$$SE(\hat{\theta}_{CR}) = \frac{\sigma_f}{\sqrt{n}}. \tag{4.3}$$

where $\sigma_f^2 = var(f(U))$. As usual we estimate σ_f^2 using the sample standard deviation. Since fn(u) provides a whole vector of estimators $(f(U_1), f(U_2), ..., f(U_n))$ then sqrt(var(fn(u))) is the sample estimator of σ_f so the standard error $SE(\hat{\theta}_{CR})$ is given by

```
Sf=sqrt(var(fn(u)));
Sf/sqrt(length(u))
```

giving an estimate 0.6603 of the standard deviation σ_f or standard error $\sigma_f/\sqrt{500000}$ or 0.0009. Of course parameters in statistical problems are usually estimated using an interval estimate or a *confidence interval,* an interval constructed using a method that guarantees capturing the true value of the parameter under similar circumstances with high probability (the confidence coefficient, often taken to be 95%). Formally,

Definition 35 *A 95% confidence interval for a parameter θ is an interval $[L, U]$ with random endpoints L, U such that the probability $P[L \leq \theta \leq U] = 0.95$.*

If we were to repeat the experiment 100 times, say by running 100 more similar independent simulations, and in each case use the results to construct a 95% confidence interval, then this definition implies that roughly 95% of the intervals constructed will contain the true value of the parameter (and of course roughly 5% will not). For an approximately Normal(μ_X, σ_X^2) random variable X, we can use the approximation

$$P[\mu_X - 2\sigma_X \leq X \leq \mu_X + 2\sigma_X] \approx 0.95 \tag{4.4}$$

(i.e. approximately normal variables are within 2 standard deviations of their mean with probability around 95%) to build a simple confidence interval. Strictly, the value $2\sigma_X$ should be replaced by $1.96\sigma_X$ where 1.96 is taken from the Normal distribution tables. The value 2 is very close to correct for a t distribution with 60 degrees of freedom. In any case these confidence intervals which assume approximate normality are typically too short (i.e. contain the true value of the parameter less frequently than stated) for most real data and so a value marginally larger than 1.96 is warranted. Replacing σ_X above by the standard deviation of a sample mean, (4.4) results in the approximately 95% confidence interval

$$\hat{\theta}_{CR} - 2\frac{\sigma_f}{\sqrt{n}} \leq \theta \leq \hat{\theta}_{CR} + 2\frac{\sigma_f}{\sqrt{n}}$$

for the true value θ. With confidence 95%, the true price of the option is within the interval $0.462 \pm 2(0.0009)$. As it happens in this case this interval does capture the true value *0.4615* of the option.

So far Monte Carlo has not told us anything we couldn't obtain from the Black-Scholes formula, but what is we used a distribution other than the normal to generate the returns? This is an easy modification of the above. For example suppose we replace the standard normal by a logistic distribution which, as we have seen, has a density function very similar to the standard normal if we choose $b = 0.625$. Of course the Black-Scholes formula does not apply to a process with logistically distributed returns. We need only replace the standard normal inverse cumulative distribution function by the corresponding inverse for the logistic,

$$F^{-1}(U) = b \ln\left(\frac{U}{1-U}\right)$$

and thus replace the Matlab code, "`norminv(u,T*(r-sigma^2/2),sigma*sqrt(T))''` by `''T*(r-sigma^2/2)+sigma*sqrt(T)*.625*log(u./(1-u))''`. This results in a slight increase in option value (to 0.504) and about a 50% considerable increase in the variance of the estimator.

We will look at the efficiency of various improvements to crude Monte Carlo, and to that end, we record the value of the variance of the estimator based on a single uniform variate in this case;

$$\sigma^2_{crude} = \sigma^2_f = var(f(U)) \approx 0.436.$$

Then the crude Monte Carlo estimator using n function evaluations or n uniform variates has variance approximately $0.436/n$. If I were able to adjust the method so that the variance σ^2_f based on a single evaluation of the function f in the numerator were halved, then I could achieve the same accuracy from a simulation using half the number of function evaluations. For this reason, when we compare two different methods for conducting a simulation, the ratio of variances corresponding to a fixed number of function evaluations can also be interpreted roughly as the ratio of computational effort required for a given predetermined accuracy. We will often compare various new methods of estimating the same function based on variance reduction schemes and quote the efficiency gain over crude Monte-Carlo sampling.

$$\text{Efficiency} = \frac{\text{variance of Crude Monte Carlo Estimator}}{\text{Variance of new estimator}} \tag{4.5}$$

where both numerator a denominator correspond to estimators with the *same number of function evaluations* (since this is usually the more expensive part of the computation). An efficiency of 100 would indicate that the crude Monte Carlo estimator would require 100 times the number of function evaluations to achieve the same variance or standard error of estimator.

Consider a crude estimator obtained from five $U[0,1]$ variates,

$$U_i = 0.1, 0.3, 0.5, 0.6, 0.8, i = 1, ..., 5.$$

The crude Monte Carlo estimator in the case $n = 5$ is displayed in Figure 3.1, the estimator being the sum of the areas of the marked rectangles. Only three of

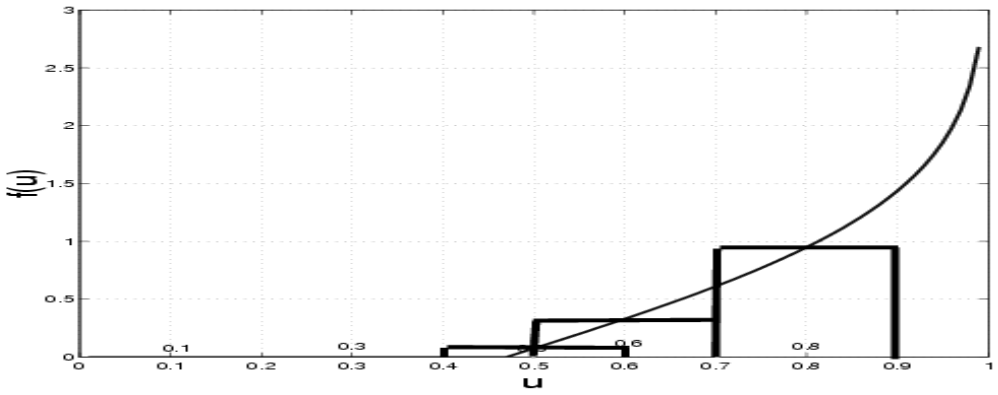

Figure 4.2: Crude Monte Carlo Estimator based on 5 observations $U_i = 0.1, 0.3, 0.5, 0.6, 0.8$

the five points actually contribute to this area since for this particular function

$$f(u) = e^{-rT}(S_0 \exp\{\Phi^{-1}(u; rT - \frac{\sigma^2}{2}T, \sigma^2 T)\} - K)^+ \tag{4.6}$$

and the parameters chosen, $f(0.1) = f(0.3) = 0$. Since these two random numbers contributed 0 and the other three appear to be on average slightly too small, the sum of the area of the rectangles appears to underestimate of the integral. Of course another selection of five uniform random numbers may prove to be even more badly distributed and may result in an under or an overestimate.

There are various ways of improving the efficiency of this estimator, many of which partially emulate numerical integration techniques. First we should note that most numerical integrals, like $\hat{\theta}_{CR}$, are weighted averages of the values of the function at certain points U_i. What if we evaluated the function at non-random points, chosen to attempt reasonable balance between locations where the function is large and small? Numerical integration techniques and quadrature methods choose both points at which we evaluate the function and weights that we attach to these points to provide accurate approximations for polynomials of certain degree. For example, suppose we insist on evaluating the

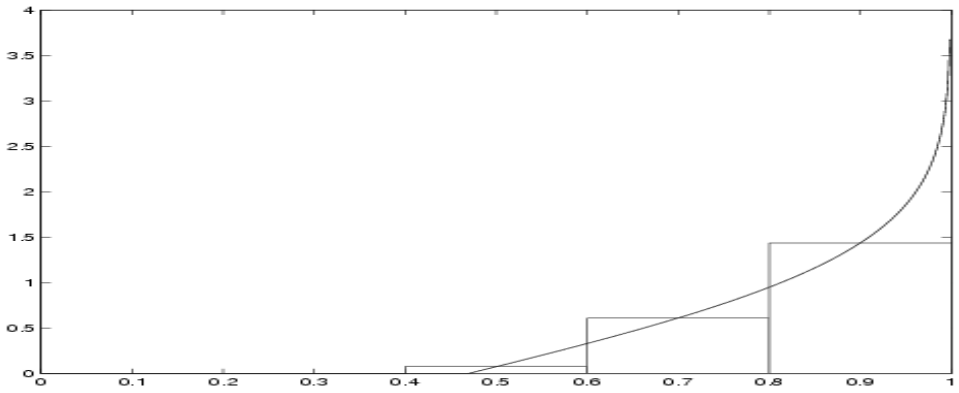

Figure 4.3: Graphical illustration of the trapezoidal rule (4.8)

function at equally spaced points, for example the points $0, 1/n, 2/n, ..., (n-1)/n, 1$. In some sense these points are now "more uniform" than we are likely to obtain from $n+1$ randomly and independently chosen points $U_i, i = 1, 2, ..., n$. The trapezoidal rule corresponds to using such equally spaced points and equal weights (except at the boundary) so that the "estimator" of the integral is

$$\hat{\theta}_{TR} = \frac{1}{2n}\{f(0) + 2f(1/n) + \ldots + 2f(1 - \frac{1}{n}) + f(1)\} \tag{4.7}$$

or the simpler and very similar alternative in our case, with $n = 5$,

$$\hat{\theta}_{TR} = \frac{1}{5}\{f(0.1) + f(0.3) + f(0.5) + f(0.7) + f(0.9)\} \tag{4.8}$$

A reasonable balance between large and small values of the function is almost guaranteed by such a rule, as shown in Figure 4.8 with the observations equally spaced.

Simpson's rule is to generate equally spaced points and weights that(except for endpoints) alternate $2/3n, 4/3n, 2/3n....$ In the case when n is *even*, the integral is estimated with

$$\hat{\theta}_{SR} = \frac{1}{3n}\{f(0) + 4f(1/n) + 2f(2/n) + \ldots + 4f(\frac{n-1}{n}) + f(1)\}. \tag{4.9}$$

The trapezoidal rule is exact for linear functions and Simpson's rule is exact for quadratic functions.

These one-dimensional numerical integration rules provide some insight into how to achieve lower variance in Monte Carlo integration. It illustrates some options for increasing accuracy over simple random sampling. We may either vary the weights attached to the individual points or vary the points (the U_i) themselves or both. Notice that as long as the U_i individually have distributions that are $Uniform[0,1]$, we can introduce any degree of dependence among them in order to come closer to the equal spacings characteristic of numerical integrals. Even if the U_i are dependent U[0,1], an estimator of the form

$$\frac{1}{n}\sum_{i=1}^{n} f(U_i)$$

will continue to be an unbiased estimator because each of the summands continue to satisfy $E(f(U_i)) = \theta$. Ideally if we introduce dependence among the various U_i and the expected value remains unchanged , we would wish that the variance

$$var(\frac{1}{n}\sum_{i=1}^{n} f(U_i))$$

is reduced over independent uniform. The simplest case of this idea is the use of antithetic random variables.

Antithetic Random Numbers.

Consider first the simple case of $n = 2$ function evaluations at possibly dependent points. Then the estimator is

$$\hat{\theta} = \frac{1}{2}\{f(U_1) + f(U_2)\}$$

with expected value $\theta = \int_0^1 f(u)du$ and variance given by

$$var(\hat{\theta}) = \frac{1}{2}\{var(f(U_1)) + cov[f(U_1), f(U_2)]\}$$

assuming both U_1, U_2 are uniform[0,1]. In the independent case the covariance term disappears and we obtain the variance of the crude Monte-Carlo estimator

$$\frac{1}{2}var(f(U_1)).$$

Notice, however, that if we are able to introduce a *negative covariance*, the resulting variance of $\hat{\theta}$ will be smaller than that of the corresponding crude Monte Carlo estimator, so the question is how to generate this negative covariance. *Suppose for example that* f *is monotone (increasing or decreasing).* Then $f(1-U_1)$ decreases whenever $f(U_1)$ increases, so that substituting $U_2 = 1 - U_1$ has the desired effect and produces a negative covariance(in fact we will show later that we cannot do any better when the function f is monotone). Such a choice of $U_2 = 1 - U_1$ which helps reduce the variability in $f(U_1)$, is termed an *antithetic variate*. In our example, because the function to be integrated is monotone, there is a negative correlation between $f(U_1)$ and $f(1 - U_1)$ and

$$\frac{1}{2}\{var(f(U_1)) + cov[f(U_1), f(U_2)]\} < \frac{1}{2}var(f(U_1)).$$

that is, the variance is decreased over simple random sampling. Of course in practice our sample size is much greater than $n = 2$, but we still enjoy the benefits of this argument if we generate the points in antithetic pairs. For example, to determine the extent of the variance reduction using antithetic random numbers, suppose we generate $500,000$ uniform variates U and use as well the values of $1 - U$ as (for a total of $1,000,000$ function evaluations as before).

F=(fn(u)+fn(1-u))/2;

This results in *mean(F)=0.46186* and *var(F)=0.1121.* The standard error of the estimator is

$$\sqrt{\frac{0.1121}{length(F)}} = \sqrt{}\overline{2.24 \times 107}.$$

Since each of the 500,000 components of F obtains from two function evaluations, the variance should be compared with a crude Monte Carlo estimator with the same number 1000000 function evaluations, $\sigma^2_{crude}/1000000 = 4.35 \times 10^{-7}$. The efficiency gain due to the use of antithetic random numbers is $4.35/2.24$ or about two, so roughly half as many function evaluations using antithetic random numbers provide the same precision as a crude Monte Carlo estimator. There is the additional advantage that only half as many uniform random variables are required. The introduction of antithetic variates has had the same effect on precision as increasing the sample size under crude Monte Carlo by a factor of approximately 2.

We have noted that antithetic random numbers improved the efficiency whenever the function being integrated is monotone in u. What if it is not. For example suppose we use antithetic random numbers to integrate the function $f(u) = u(1-u)$ on the interval $0 < u < 1$? Rather than balance large values with small values and so reduce the variance of the estimator, in this case notice that $f(U)$ and $f(1-U)$ are strongly *positively* correlated, in fact are equal, and so the argument supporting the use of antithetic random numbers for monotone functions will show that in this case they increase the variance over a crude estimator with the same number of function evaluations. Of course this problem can be remedied if we can identify intervals in which the function is monotone, e.g. in this case use antithetic random numbers in the two intervals $[0, \frac{1}{2}]$ and $[\frac{1}{2}, 1]$, so for example we might estimate $\int_0^1 f(u)du$ by an average of terms like

$$\frac{1}{4}\{f(\frac{U_1}{2}) + f(\frac{1-U_1}{2}) + f(\frac{1+U_2}{2}) + f(\frac{2-U_2}{2})\}$$

for independent $U[0,1]$ random variables U_1, U_2.

Stratified Sample.

One of the reasons for the inaccuracy of the crude Monte Carlo estimator in the above example is the large interval, evident in Figure 4.1, in which the function

is zero. Nevertheless, both crude and antithetic Monte Carlo methods sample in that region, this portion of the sample contributing nothing to our integral. Naturally, we would prefer to concentrate our sample in the region where the function is positive, and where the function is more variable, use larger sample sizes. One method designed to achieve this objective is the use of a *stratified sample*. Once again for a simple example we choose $n = 2$ function evaluations, and with $V_1 \sim U[0, a]$ and $V_2 \sim U[a, 1]$ define an estimator

$$\hat{\theta}_{st} = af(V_1) + (1 - a)f(V_2).$$

Note that this is a weighted average of the two function values with weights a and $1 - a$ proportional to the length of the corresponding intervals. It is easy to show once again that the estimator $\hat{\theta}_{st}$ is an unbiased estimator of θ, since

$$\begin{aligned} E(\hat{\theta}_{st}) &= aEf(V_1) + (1 - a)Ef(V_2) \\ &= a\int_0^a f(x)\frac{1}{a}dx + (1 - a)\int_a^1 f(x)\frac{1}{1 - a}dx \\ &= \int_0^1 f(x)dx. \end{aligned}$$

Moreover,

$$var(\hat{\theta}_{st}) = a^2 var[f(V_1)] + (1 - a)^2 var[f(V_2)] + 2a(1 - a)cov[f(V_1), f(V_2)]. \tag{4.10}$$

Even when $V_1, V_2$are independent, so we obtain $var(\hat{\theta}_{st}) = a^2 var[f(V_1)] + (1 - a)^2 var[f(V_2)]$, there may be a dramatic improvement in variance over crude Monte Carlo provided that the variability of f in each of the intervals $[0, a]$ and $[a, 1]$ is substantially less than in the whole interval $[0, 1]$.

Let us return to the call option example above, with f defined by (4.6). Suppose for simplicity we choose independent values of V_1, V_2. In this case

$$var(\hat{\theta}_{st}) = a^2 var[f(V_1)] + (1 - a)^2 var[f(V_2)]. \tag{4.11}$$

For example for $a = .7$, this results in a variance of about 0.046 obtained from the following

```
F=a*fn(a*rand(1,500000))+(1-a)*fn(a+(1-a)*rand(1,500000));
var(F)
```

and the variance of the sample mean of the components of the vector F is var(F)/length(F) or around 9.2×10^{-8}. Since each component of the vector above corresponds to two function evaluations we should compare this with a crude Monte Carlo estimator with $n = 1000000$ having variance $\sigma_f^2 \times 10^{-6} = 4.36 \times 10^{-7}$. This corresponds to an efficiency gain of $43.6/9.2$ or around 5. We can afford to use one fifth the sample size by simply stratifying the sample into two strata. The improvement is somewhat limited by the fact that we are still sampling in a region in which the function is 0 (although now slightly less often).

A general stratified sample estimator is constructed as follows. We subdivide the interval $[0, 1]$ into convenient subintervals $0 = x_0 < x_1 < ...x_k = 1$, and then select n_i random variables uniform on the corresponding interval $V_{ij} \sim U[x_{i-1}, x_i], j = 1, 2, ..., n_i$. Then the estimator of θ is

$$\hat{\theta}_{st} = \sum_{i=1}^{k} (x_i - x_{i-1}) \frac{1}{n_i} \sum_{j=1}^{n_i} f(V_{ij}). \tag{4.12}$$

Once again the weights $(x_i - x_{i-1})$ on the average of the function in the $i'th$ interval are proportional to the lengths of these intervals and the estimator $\hat{\theta}_{st}$ is unbiased;

$$\begin{aligned} E(\hat{\theta}_{st}) &= \sum_{i=1}^{k} (x_i - x_{i-1}) E\{\frac{1}{n_i} \sum_{j=1}^{n_i} f(V_{ij})\} \\ &= \sum_{i=1}^{k} (x_i - x_{i-1}) E f(V_{i1}) \\ &= \sum_{i=1}^{k} (x_i - x_{i-1}) \int_{x_{i-1}}^{x_i} f(x) \frac{1}{x_i - x_{i-1}} dx \\ &= \int_0^1 f(x) dx = \theta. \end{aligned}$$

In the case that all of the V_{ij} are independent, the variance is given by:

$$var(\hat{\theta}_{st}) = \sum_{i=1}^{k}(x_i - x_{i-1})^2 \frac{1}{n_i} var[f(V_{i1})]. \tag{4.13}$$

Once again, if we choose our intervals so that the variation within intervals $var[f(V_{i1})]$ is small, this provides a substantial improvement over crude Monte Carlo. Suppose we wish to choose the sample sizes so as to minimize this variance. Obviously to avoid infinite sample sizes and to keep a ceiling on costs, we need to impose a constraint on the total sample size, say

$$\sum_{i}^{k} n_i = n. \tag{4.14}$$

If we treat the parameters n_i as continuous variables we can use the method of Lagrange multipliers to solve

$$\min_{\{n_i\}} \sum_{i=1}^{k}(x_i - x_{i-1})^2 \frac{1}{n_i} var[f(V_{i1})]$$

subject to constraint (4.14).

It is easy to show that the optimal choice of sample sizes within intervals are

$$n_i \propto (x_i - x_{i-1})\sqrt{var[f(V_{i1})]}$$

or more precisely that

$$n_i = n\frac{(x_i - x_{i-1})\sqrt{var[f(V_{i1})]}}{\sum_{j=1}^{k}(x_j - x_{j-1})\sqrt{var[f(V_{j1})]}}. \tag{4.15}$$

In practice,of course, this will not necessarily produce an integral value of n_i and so we are forced to round to the nearest integer. For this optimal choice of sample size, the variance is now given by

$$var(\hat{\theta}_{st}) = \frac{1}{n}\{\sum_{j=1}^{k}(x_j - x_{j-1})\sqrt{var[f(V_{j1})]}\}^2$$

The term $\sum_{j=1}^{k}(x_j - x_{j-1})\sqrt{var[f(V_{j1})]}$ is a weighted average of the standard deviation of the function f within the interval (x_{i-1}, x_i) and it is clear that,

at least for a continuous function, these standard deviations can be made small simply by choosing k large with $|x_i - x_{i-1}|$ small. In other words if we ignore the fact that the sample sizes must be integers, at least for a continuous function f, we can achieve arbitrarily small $var(\hat{\theta}_{st})$ using a fixed sample size n simply by stratifying into a very large number of (small) strata. The intervals should be chosen so that the variances $var[f(V_{i1})]$ are small. $n_i \propto (x_i - x_{i-1})\sqrt{var[f(V_{i1})]}$. In summary, *optimal sample sizes are proportional to the lengths of intervals times the standard deviation of function evaluated at a uniform random variable on the interval. For sufficiently small strata we can achieve arbitrarily small variances.* The following function was designed to accept the strata $x_1, x_2, ..., x_k$ and the desired sample size n as input, and then determine optimal sample sizes and the stratified sample estimator as follows:

1. Initially sample sizes of 1000 are chosen from each stratum and these are used to estimate $\sqrt{var[f(V_{i1})]}$
2. Approximately optimal sample sizes n_i are then calculated from (4.15).
3. Samples of size n_i are then taken and the stratified sample estimator (4.12), its variance (4.13) and the sample sizes n_i are output.

```
function [est,v,n]=stratified(x,nsample)
% function for optimal sample size stratified estimator on call option price example
%[est,v,n]=stratified([0 .6 .85 1],100000)  uses three strata (0,.6),(.6 .85),(.85 1)
and total sample size 100000
est=0;
n=[];
m=length(x);
for i=1:m-1                          % the preliminary sample of size 1000
 v= var(callopt2(unifrnd(x(i),x(i+1),1,1000),10,10,.05,.2,.25));
 n=[n (x(i+1)-x(i))*sqrt(v)];
```

```
    end
    n=floor(nsample*n/sum(n));   %calculation of the optimal sample sizes, rounded
down
    v=0;
    for i=1:m-1
     F=callopt2(unifrnd(x(i),x(i+1),1,n(i)),10,10,.05,.2,.25);    %evaluate the function
f at n(i) uniform points in interval
     est=est+(x(i+1)-x(i))*mean(F);
     v=v+var(F)*(x(i+1)-x(i))^2/n(i);
    end
```

A call to *[est,v,n]=stratified([0 .6 .85 1],100000)* for example generates a stratified sample with three strata[0, 0.6], (0.6, 0.85], and (0.8, 1] and outputs the estimate $est = 0.4617$, its variance $v = 3.5 \times 10^{-7}$ and the approximately optimal choice of sample sizes $n = 26855, 31358, 41785$. To compare this with a crude Monte Carlo estimator, note that a total of 99998 function evaluations are used so the efficiency gain is $\sigma_f^2/(99998 \times 3.5 \times 10^{-7}) = 12.8$. Evidently this stratified random sample can account for an improvement in efficiency of about a factor of 13. Of course there is a little setup cost here (a preliminary sample of size 3000) which we have not included in our calculation but the results of that preliminary sample could have been combined with the main sample for a very slight decrease in variance as well). For comparison, the function call

```
[est,v,n]=stratified([.47 .62 .75 .87 .96 1],1000000)
```

uses five strata [.47 .62],[.62 .75], [.75, .87], [.87, .96], [.96, 1] and gives a variance of the estimator of 7.4×10^{-9}. Since a crude sample of the same size has variance around 4.36×10^{-7} the efficiency is about 170. This stratified sample is as good as a crude Monte Carlo estimator with 170 million simulations! By introducing more strata, we can increase this efficiency as much as we wish.

Within a stratified random sample we may also introduce antithetic variates designed to provide negative covariance. For example we may use antithetic

pairs within an interval if we believe that the function is monotone in the interval, or if we believe that the function is increasing across adjacent strata, we can introduce antithetic pairs between two intervals. For example, we may generate $U \sim Uniform[0,1]$ and then sample the point $V_{ij} = x_{i-1} + (x_i - x_{i-1})U$ from the interval (x_{i-1}, x_i) as well as the point $V_{(i+1)j} = x_{i+1} - (x_{i+1} - x_i)U$ from the interval (x_i, x_{i+1}) to obtain antithetic pairs between intervals. For a simple example of this applied to the above call option valuation, consider the estimator based on three strata [0,.47),[0.47 0.84),[0.84 1]. Here we have not bothered to sample to the left of 0.47 since the function is 0 there, so the sample size here is set to 0. Then using antithetic random numbers within each of the two strata [0.47 0.84),[0.84 1], and $U \sim Uniform[0,1]$ we obtain the estimator

$$\hat{\theta}_{str,ant} = \frac{0.37}{2}[f(.47+.37U)+f(.84-.37U)]+\frac{0.16}{2}[f(.84+.16U)+f(1-.16U)]$$

To assess this estimator,
we evaluated, for U a vector of 1000000 uniform,

```
U=rand(1,1000000);
F=.37*.5*(fn(.47+.37*U)+fn(.84-.37*U))+.16*.5*(fn(.84+.16*U)+fn(1-.16*U));
mean(F)                    % gives 0.4615
var(F)/length(F)           %  gives 1.46×10^-9
```

This should be compared with the crude Monte-Carlo estimator having the same number $n = 4\times 10^6$ function evaluations as each of the components of the vector $F : \sigma^2_{crude}/(4\times 10^6) = 1.117\times 10^{-7}$. The gain in efficiency is therefore 1.117/.0146 or approximately 77. The above stratified-antithetic simulation with 1,000,000 input variates and 4,000,000 function evaluations is equivalent to a crude Monte Carlo simulation with sample size 308 million! Variance reduction makes the difference between a simulation that is feasible on a laptop and one that would

require a very long time on a mainframe computer. However on a Pentium IV 2.2GHZ laptop it took approximately 58 seconds to run.

Control Variates.

There are two techniques that permit using knowledge about a function with shape similar to that of f. First, we consider the use of a *control variate*, based on the trivial identity

$$\int f(u)du = \int g(u)du + \int (f(u) - g(u))du. \tag{4.16}$$

for an arbitrary function $g(u)$. Assume that the integral of g is known, so we can substitute its known value for the first term above. The second integral we assume is more difficult and we estimate it by crude Monte Carlo, resulting in estimator

$$\hat{\theta}_{cv} = \int g(u)du + \frac{1}{n}\sum_{i=1}^{n}[f(U_i) - g(U_i)]. \tag{4.17}$$

This estimator is clearly unbiased and has variance

$$\begin{aligned} var(\hat{\theta}_{cv}) &= var\{\frac{1}{n}\sum_{i=1}^{n}[f(U_i) - g(U_i)]\} \\ &= \frac{var[f(U) - g(U)]}{n} \end{aligned}$$

so the variance is reduced over that of crude Monte Carlo estimator having the same sample size n by a factor

$$\frac{var[f(U)]}{var[f(U) - g(U)]} \quad \text{for } U \sim U[0,1]. \tag{4.18}$$

Let us return to the example of pricing a call option. By some experimentation, which could involve a preliminary crude simulation or simply evaluating the function at various points, we discovered that the function

$$g(u) = 6[(u - .47)^+]^2 + (u - .47)^+$$

provided a reasonable approximation to the function $f(u)$. The two functions are compared in Figure 4.4. Moreover, the integral $2 \times 0.53^2 + \frac{1}{2}0.53^3$ of the function $g(.)$ is easy to obtain.

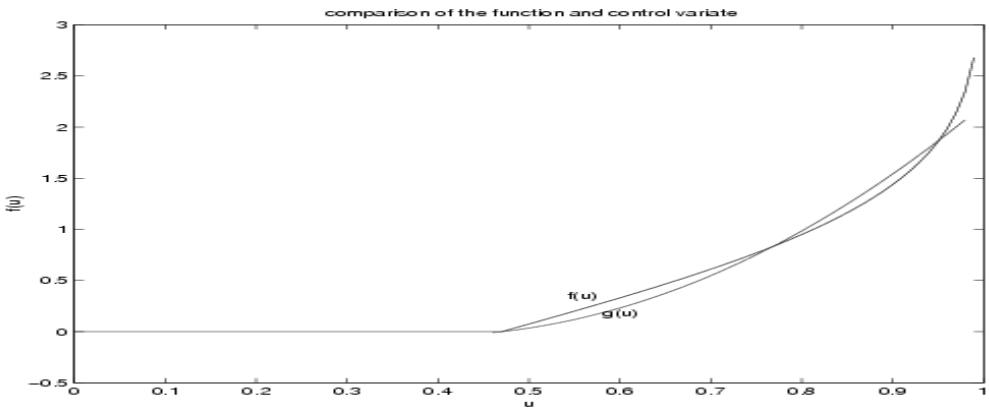

Figure 4.4: Comparison of the function $f(u)$ and the control variate $g(u)$

It is obvious from the figure that since $f(u) - g(u)$ is generally much smaller and less variable than is $f(u)$, $var[f(U) - g(U)] < var(f(U))$. The variance of the crude Monte Carlo estimator is determined by the variability in the function $f(u)$ over its full range. The variance of the control variate estimator is determined by the variance of the difference between the two functions, which in this case is quite small. We used the following matlab functions, the first to generate the function $g(u)$ and the second to determine the efficiency gain of the control variate estimator;

```
function g=GG(u)          % this is the function g(u), a control variate for fn(u)
u=max(0,u-.47);
g=6*u.^2+u;
```

```
function [est,var1,var2]=control(f,g,intg,n)
% run using a statement like [est,var1,var2]=control('fn','GG',intg,n)
%  runs a simulation on the function f using control variate g (both character
```

strings) n times.

% intg is the integral of g % intg=$\int_0^1 g(u)du$

% outputs estimator est and variances var1,var2, variances with and without control variate.

```
U=unifrnd(0,1,1,n);
FN=eval(strcat(f,'(U)'));          % evaluates f(u) for vector u
CN=eval(strcat(g,'(U)'));          % evaluates g(u)
est=intg+mean(FN-CN);
var1=var(FN);
var2=var(FN-CN);
```

Then the call *[est,var1,var2]=control('fn','GG',2*(.53)^3+(.53)^2/2,1000000)* yields the estimate 0.4616 and variance=1.46×10^{-8} for an efficiency gain over crude Monte Carlo of around 30.

This elementary form of control variate suggests using the estimator

$$\int g(u)du + \frac{1}{n}\sum_{i=1}^{n}[f(U_i) - g(U_i)]$$

but it may well be that $g(U)$ is not the best estimator we can imagine for $f(U)$. We can often find a linear function of $g(U)$ which is better by using regression. Since elementary regression yields

$$f(U) - E(f(U)) = \beta(g(U) - E(g(U))) + \epsilon \tag{4.19}$$

where

$$\beta = \frac{cov(f(U), g(U))}{var(g(U))} \tag{4.20}$$

and the errors ϵ have expectation 0, it follows that $E(f(U)) + \epsilon = f(U) - \beta[g(U) - E(g(U))]$ and so$f(U) - \beta[g(U) - E(g(U))]$ is an unbiased estimator of $E(f(U))$. For a sample of N uniform random numbers this becomes

$$\hat{\theta}_{cv} = \beta E(g(U)) + \frac{1}{N}\sum_{i=1}^{N}[f(U_i) - \beta g(U_i)]. \tag{4.21}$$

Moreover this estimator having smallest variance among all linear combinations of $f(U)$and $g(U)$. Note that when $\beta = 1$ (4.21) reduces to the simpler form of the control variate technique (4.17) discussed above. However, the latter is generally better in terms of maximizing efficiency. Of course in practice it is necessary to estimate the covariance and the variances in the definition of β from the simulations themselves by evaluating f and g at many different uniform random variables $U_i, i = 1, 2, ..., N$ and then estimating β using the standard least squares estimator

$$\widehat{\beta} = \frac{N\sum_{i=1}^{N} f(U_i)g(U_i) - \sum_{i=1}^{N} f(U_i)\sum_{i=1}^{N} g(U_i)}{N\sum_{i=1}^{N} g^2(U_i) - (\sum_{i=1}^{N} g(U_i))^2}.$$

Although in theory the substitution of an estimator $\widehat{\beta}$ for the true value β results in a small bias in the estimator, for large numbers of simulations N our estimator $\widehat{\beta}$ is so close to the true value that this bias can be disregarded.

Importance Sampling.

A second technique that is similar is that of *importance sampling.* Again we depend on having a reasonably simple function g that after muultiplication by some constant, is similar to f. However, rather than attempt to minimize the difference $f(u) - g(u)$ between the two functions, we try and find $g(u)$ such that $f(u)/g(u)$ is nearly a constant. We also require that g is non-negative and can be integrated so that, after rescaling the function, it integrates to one, i.e. it is a probability density function. Assume we can easily generate random variables from the probability density function $g(z)$. The distribution whose probability density function is $g(z), z \in [0, 1]$ is the *importance distribution.* Note that if we generate a random variable Z having the probability density

function $g(z), z \in [0,1]$ then

$$\int f(u)du = \int_0^1 \frac{f(z)}{g(z)} g(z)dz$$
$$= E\left[\frac{f(Z)}{g(Z)}\right]. \tag{4.22}$$

This can therefore be estimated by generating independent random variables Z_i with probability density function $g(z)$ and then setting

$$\hat{\theta}_{im} = \frac{1}{n}\sum_{i=1}^{n} \frac{f(Z_i)}{g(Z_i)}. \tag{4.23}$$

Once again, according to (4.22), this is an unbiased estimator and the variance is

$$var\{\hat{\theta}_{im}\} = \frac{1}{n} var\{\frac{f(Z_1)}{g(Z_1)}\}. \tag{4.24}$$

Returning to our example, we might consider using the same function as before for $g(u)$. However, it is not easy to generate variates from a density proportional to this function g by inverse transform since this would require solving a cubic equation. Instead, let us consider something much simpler, the density function $g(u) = 2(0.53)^{-2}(u - .47)^+$ having cumulative distribution function $G(u) = (0.53)^2\ [(u - .47)^+]^2$ and inverse cumulative distribution function $G^{-1}(u) = 0.47 + 0.53\sqrt{u}$. In this case we generate Z_i using $Z_i = G^{-1}(U_i)$ for $U_i \sim Uniform[0,1]$. The following function simulates an importance sample estimator:

```
function [est,v]=importance(f,g,Ginv,u)
%runs a simulation on the function 'f" using importance density "g"(both character strings) and inverse c.d.f. "Ginverse"
% outputs all estimators (should be averaged) and variance.
% IM is the inverse cf of the importance distribution c.d.f.
% run e.g.
% [est,v]=importance('fn','2*(IM-.47)/(.53)^2;','.47+.53*sqrt(u);',rand(1,1000));
```

IM= eval(Ginv); %=.47+.53*sqrt(u);

%IMdens is the density of the importance sampling distribution at IM

IMdens=eval(g); %2*(IM-.47)/(.53)^2;

FN=eval(strcat(f,'(IM)'));

est=FN./IMdens; % mean(est) prrovides the estimator

v=var(FN./IMdens)/length(IM); % this is the variance of the estimator per simulation

The function was called with *[est,v]=importance('fn','2*(IM-.47)/(.53)^2;','.47+.53*sqrt(u);',rand(1,*

giving an estimate mean(*est)* = *0.4616* with variance 1.28×10^{-8} for an efficiency gain of around 35 over crude Monte Carlo.

Example 36 *(Estimating Quantiles using importance sampling.) Suppose we are able to generate random variables X from a probability density function of the form*

$$f_\theta(x)$$

and we wish to estimate a quantile such as VAR, i.e. estimate x_p such that

$$P_{\theta_0}(X \leq x_p) = p$$

for a certain value θ_0 of the parameter.

As a very simple example suppose S is the sum of 10 independent random variables having the exponential distribution with mean θ, and $f_\theta(x_1, ..., x_{10})$ is the joint probability density function of these 10 observations. Assume $\theta_0 = 1$ and $p = .999$ so that we seek an extreme quantile of the sum, i.e. we want to determine $x_p$such that $P_{\theta_0}(S \leq x_p) = p$. The equation that we wish to solve for x_p is

$$E_{\theta_0}\{I(S \leq x_p)\} = p. \tag{4.25}$$

The crudest estimator of this is obtained by generating a large number of independent observations of S under the parameter value $\theta_0 = 1$ and finding

the p'th quantile, i.e. by defining the empirical c.d.f.. We generate independent random vectors $X_i = (X_{i1}, ..., X_{i10})$ from the probability density $f_{\theta_0}(x_1, ..., x_{10})$ and with $S_i = \sum_{j=1}^{10} X_{ij}$,define

$$\widehat{F}(x) = \frac{1}{n} \sum_{i=1}^{n} I(S_i \leq x). \tag{4.26}$$

Invert it (possibly with interpolation) to estimate the quantile

$$\widehat{x_p} = \widehat{F}^{-1}(p). \tag{4.27}$$

If the true cumulative distribution function is differentiable, the variance of this quantile estimator is asymptotically related to the variance of our estimator of the cumulative distribution function,

$$var(\widehat{x_p}) \simeq \frac{var(\widehat{F}(x_p))}{(F'(x_p))^2},$$

so any variance reduction in the estimator of the c.d.f. us reflected, at least asymptotically, in a variance reduction in the estimator of the quantile. Using importance sampling (4.25) is equivalent to the same technique but with

$$\widehat{F}_I(x) = \frac{1}{n} \sum_{i=1}^{n} W_i I(S_i \leq x) \text{ where} \tag{4.28}$$

$$W_i = \frac{f_{\theta_0}(X_{i1}, ..., X_{i10})}{f_{\theta}(X_{i1}, ..., X_{i10})}$$

Ideally we should choose the value of θ so that the variance of $\widehat{x}_p$ or of

$$W_i I(S_i \leq x_p)$$

is as small as possible. This requires a wise guess or experimentation with various choices of θ. For a given θ we have another choice of empirical cumulative distribution function

$$\widehat{F}_{I2}(x) = \frac{1}{\sum_{i=1}^{n} W_i} \sum_{i=1}^{n} W_i I(S_i \leq x). \tag{4.29}$$

Both of these provide fairly crude estimates of the sample quantiles when observations are weighted and, as one does with the sample median, one could easily interpolate between adjacent values around the value of x_p.

The alternative (4.29) is motivated by the fact that the values W_i appear as weights attached to the observations S_i and it therefore seems reasonable to divide by the sum of the weights. In fact the expected value of the denominator is

$$E_\theta\{\sum_{i=1}^{n} W_i\} = n$$

so the two denominators are similar. In the example where the X_{ij} are independent exponential(1) let us examine the weight on S_i determined by $X_i = (X_{i1}, ..., X_{i10})$,

$$W_i = \frac{f_{\theta_0}(X_{i1}, ..., X_{i10})}{f_\theta(X_{i1}, ..., X_{i10})} = \prod_{j=1}^{10} \frac{\exp(-X_{ij})}{\theta^{-1}\exp(-X_{ij}/\theta)} = \theta^{10}\exp\{-S_i(1-\theta^{-1})\}.$$

The renormalized alternative (4.29) might be necessary for estimating extreme quantiles when the number of simulations is small but only the first provides an completely unbiased estimating function. In our case, using (4.28) with $\theta = 2.5$ we obtained an estimator of $F(x_{0.999})$ with efficiency about 180 times that of a crude Monte Carlo simulation. There is some discussion of various renormalizations of the importance sampling weights in Hesterberg(1995).

Importance Sampling, the Exponential Tilt and the Saddlepoint Approximation

When searching for a convenient importance distribution, particularly if we wish to increase or decrease the frequency of observations in the tails, it is quite common to embed a given density in an exponential family. For example suppose we wish to estimate an integral

$$\int g(x)f(x)dx$$

where $f(x)$ is a probability density function. Suppose $K(s)$ denotes the cumulant generating function (the logarithm of the moment generating function) of the density $f(x)$,i.e. if

$$\exp\{K(s)\} = \int e^{xs} f(x)dx.$$

The cumulant generating function is a useful summary of the moments of a distribution since the mean can be determined as $K'(0)$ and the variance as $K''(0)$. From this single probability density function, we can now produce a whole (exponential) family of densities

$$f_\theta(x) = e^{\theta x - K(\theta)} f(x) \tag{4.30}$$

of which $f(x)$ is a special case corresponding to $\theta = 0$. The density (4.30) is often referred to as an exponential tilt of the original density function and increases the weight in the right tail for $\theta > 0$, decreases it for $\theta < 0$.

This family of densities is closely related to the saddlepoint approximation. If we wish to estimate the value of a probability density function $f(x)$ at a particular point x, then note that this could be obtained from (4.30) if we knew the probability density function $f_\theta(x)$. On the other hand a normal approximation to a density is often reasonable at or around its mode, particularly if we are interested in the density of a sum or an average of independent random variables. The cumulant generating function of the density $f_\theta(x)$ is easily seen to be $K(\theta + s)$ and the mean is therefore $K'(\theta)$. If we choose the parameter $\theta = \theta(x)$ so that

$$K'(\theta) = x \tag{4.31}$$

then the density f_θ has mean x and variance $K''(\theta)$. How do we know for a given value of x there exists a solution to (4.31)? From the properties of cumulant generating functions, $K(t)$ is convex, increasing and $K(0) = 0$.This implies that as t increases, the slope of the cumulant generating function $K'(t)$ is non-decreasing. It therefore approaches a limit $x_{\max}$(finite or infinite) as $t \to \infty$ and as long as we restrict the value of x in (4.31) to the interval $x < x_{\max}$ we can find a solution. The value of the $N(x, K''(\theta))$ at the value x is

$$f_\theta(x) \approx \sqrt{\frac{1}{2\pi K''(\theta)}}$$

and therefore the approximation to the density $f(x)$ is

$$f(x) \approx \sqrt{\frac{1}{2\pi K''(\theta)}} e^{K(\theta)-\theta x}. \tag{4.32}$$

$$\text{where } \theta = \theta(x) \text{ satisfies } K'(\theta) = x.$$

This is the saddlepoint approximation, discovered by Daniels (1954, 1980), and usually applied to the distribution of sums or averages of independent random variables because then the normal approximation is better motivated. Indeed, the saddlepoint approximation to the distribution of the sum of n independent identically distributed random variables is accurate to order $O(n^{-1})$ and if we renormalize it to integrate to one, accuracy to order $O(n^{-3/2})$ is possible, substantially better than the order $O(n^{-1/2})$ of of the usual normal approximation.

Consider, for example, the saddlepoint approximation to the Gamma(α,1) distribution. In this case the cumulant generating function is

$$\begin{aligned} K(t) &= -\alpha \ln(1-t), \\ K'(\theta) &= x \text{ implies } \theta(x) = 1 - \frac{\alpha}{x} \text{ and} \\ K''(\theta) &= \frac{x^2}{\alpha} \end{aligned}$$

Therefore the saddlepoint approximation to the probability density function is

$$\begin{aligned} f(x) &\simeq \sqrt{\frac{\alpha}{2\pi x^2}} \exp\{\alpha \ln(x/\alpha) - x(1 - \frac{\alpha}{x})\} \\ &= \sqrt{\frac{1}{2\pi}} \alpha^{\frac{1}{2}-\alpha} e^{\alpha} x^{\alpha-1} \exp(-x). \end{aligned}$$

This is exactly the gamma density function with Stirling's approximation replacing $\Gamma(\alpha)$ and after renormalization this is exactly the Gamma density function.

In many cases the saddlepoint approximation is used to estimate a probability or a moment of a distribution and it is of interest to estimate the error in this approximation. For example suppose that we wish, for some function h, to estimate the error in the saddlepoint approximation to $E(h(S_n))$ where

$S_n = \sum_{i=1}^{n} X_i$ and each random variable X_i has the *non-central chi-squared* distribution with cumulant generating function

$$K(t) = \frac{2\lambda t}{1-2t} - \frac{p}{2}\ln(1-2t).$$

The parameter λ is the *non-centrality parameter* of the distribution and p is the *degrees of freedom.* Notice that the cumulant generating function of the sum takes the same form but with (λ, p) replaced by $(n\lambda, np)$. If $f(x)$ is the actual probability density function of this sum and f_s is the saddlepoint approximation to the same density, then the error in the saddlepoint approximation is

$$\int_0^\infty h(x)f_s(x)dx - \int_0^\infty h(x)f(x)dx = \int_0^\infty h(x)(f_s(x) - f(x))dx. \qquad (4.33)$$

The right hand side of (4.33) can be approximated with a relatively small Monte Carlo sample since the differences appearing in it $f_s(x) - f(x)$ are typically small. In this case, we might use importance sampling and since it is easy to generate from the distribution of S_n, we could simply average simulated values of

$$h(S_n)(\frac{f_s(S_n)}{f(S_n)} - 1).$$

Since it is often computationally expensive to generate random variables whose distribution is a convolution of known densities, it is interesting to ask whether (4.32) makes this any easier. In general, $K(t)$ is convex with $K(0) = 0$ and $\lambda(x) = \theta(x) - \frac{K(\theta(x))}{x}$ is an increasing function of x.This is because, upon differentiation, we obtain

$$\lambda'(x) = \frac{1}{K''(\theta(x))} - 1 + \frac{K(\theta(x))}{x^2} =$$

FINISH

Combining Monte Carlo Estimators.

We have now seen a number of different variance reduction techniques and there are many more possible. With many of these methods such as importance and

stratified sampling are associated parameters which may be chosen in different ways. The variance formula may be used as a basis of choosing a "best" method but these variances and efficiencies must also estimated from the simulation and it is rarely clear *a priori* which sampling procedure and estimator is best. For example if a function f is monotone on $[0,1]$ then an antithetic variate can be introduced with an estimator of the form

$$\hat{\theta}_{a1} = \frac{1}{2}[f(U) + f(1-U)], \quad U \sim U[0,1] \tag{4.34}$$

but if the function is increasing to a maximum somewhere around $\frac{1}{2}$ and then decreasing thereafter we might prefer

$$\hat{\theta}_{a2} = \frac{1}{4}[f(U/2) + f((1-U)/2) + f((1+U)/2) + f(1-U/2)]. \tag{4.35}$$

Notice that any weighted average of these two unbiased estimators of θ would also provide an unbiased estimator of θ. The large number of potential variance reduction techniques is an embarrassment of riches. Which variance reduction methods we should use and how will we know whether it is better than the competitors? Fortunately, the answer is often to use "all of the methods" (within reason of course); that choosing a single method is often neither necessary nor desirable. Rather it is preferable to use a weighted average of the available estimators with the optimal choice of the weights provided by regression.

Suppose in general that we have k estimators or statistics $\widehat{\theta}_i, i = 1, ..k$, all unbiased estimators of the same parameter θ so that $E(\widehat{\theta}_i) = \theta$ for all i . In vector notation, letting $\mathbf{\Theta'} = (\widehat{\theta}_1, ..., \widehat{\theta}_k)$, we write $E(\Theta) = \mathbf{1}\theta$ where $\mathbf{1}$ is the k-dimensional column vector of ones so that $\mathbf{1'} = (1, 1, ..., 1)$. Let us suppose for the moment that we know the variance-covariance matrix V of the vector $\mathbf{\Theta}$, defined by

$$V_{ij} = cov(\widehat{\theta}_i, \widehat{\theta}_j).$$

Theorem 37 *(best linear combinations of estimators)*

The linear combination of the $\widehat{\theta}_i$ which provides an unbiased estimator of θ and

has minimum variance among all linear unbiased estimators is

$$\widehat{\theta}_{blc} = \sum_i b_i \widehat{\theta}_i \tag{4.36}$$

where the vector $\mathbf{b} = (b_1, ..., b_k)'$ *is given by*

$$\mathbf{b} = (\mathbf{1}^t V^{-1} \mathbf{1})^{-1} V^{-1} \mathbf{1}.$$

The variance of the resulting estimator is

$$var(\widehat{\theta}_{blc}) = \mathbf{b}^t V \mathbf{b} = 1/(\mathbf{1}^t V^{-1} \mathbf{1})$$

Proof. The proof is straightforward. It is easy to see that for any linear combination (4.36) the variance of the estimator is

$$\mathbf{b}^t V \mathbf{b}$$

and we wish to minimize this quadratic form as a function of $\mathbf{b}$ subject to the constraint that the coefficients add to one, or that

$$\mathbf{b}'\mathbf{1} = 1.$$

Introducing the Lagrangian, we wish to set the derivatives with respect to the components b_i equal to zero

$$\frac{\partial}{\partial \mathbf{b}} \{\mathbf{b}^t V \mathbf{b} + \lambda(\mathbf{b}'\mathbf{1} - 1)\} = \mathbf{0} \text{ or}$$

$$2V\mathbf{b} + \lambda \mathbf{1} = \mathbf{0}$$

$$\mathbf{b} = \text{constant} \times V^{-1}\mathbf{1}$$

and upon requiring that the coefficients add to one, we discover the value of the constant above is $(\mathbf{1}^t V^{-1} \mathbf{1})^{-1}$. ■

This theorem indicates that the ideal linear combination of estimators has coefficients proportional to the *row sums of the inverse covariance matrix*. Notably, the variance of a particular estimator $\widehat{\theta}_i$ is an ingredient in that sum,

but one of many. In practice, of course, we almost never know the variance-covariance matrix V of a vector of estimators Θ. However, when we do simulation evaluating these estimators using the same uniform input to each, we obtain independent replicated values of Θ. This permits us to estimate the covariance matrix V and since we typically conduct many simulations this estimate can be very accurate. Let us suppose that we have n simulated values of the vectors Θ, and call these $\Theta_1, ..., \Theta_n$. As usual we estimate the covariance matrix V using the sample covariance matrix

$$\widehat{V} = \frac{1}{n-1}\sum_{i=1}^{n}(\Theta_i - \overline{\Theta})(\Theta_i - \overline{\Theta})'$$

where

$$\overline{\Theta} = \frac{1}{n}\sum_{i=1}^{n}\Theta_i.$$

Let us return to the example and attempt to find the best combination of the many estimators we have considered so far. To this end, let

$$\widehat{\theta}_1 = \frac{0.53}{2}[f(.47 + .53u) + f(1 - .53u)] \quad \text{an antithetic estimator,}$$
$$\widehat{\theta}_2 = \frac{0.37}{2}[f(.47 + .37u) + f(.84 - .37u)] + \frac{0.16}{2}[f(.84 + .16u) + f(1 - .16u)],$$
$$\widehat{\theta}_3 = 0.37[f(.47 + .37u)] + 0.16[f(1 - .16u)], \text{ (stratified-antithetic)}$$
$$\widehat{\theta}_4 = \int g(x)dx + [f(u) - g(u)], \quad \text{(control variate)}$$
$$\widehat{\theta}_5 = \hat{\theta}_{im}, \quad \text{the importance sampling estimator (4.23).}$$

Then $\widehat{\theta}_2$, and $\widehat{\theta}_3$ are both stratified-antithetic estimators, $\widehat{\theta}_4$ is a control variate estimator and $\widehat{\theta}_5$ the importance sampling estimator discussed earlier, all obtained from a single input uniform random variate U. In order to determine the optimal linear combination we need to generate simulated values of all 5 estimators using the same uniform random numbers as inputs. We determine the best linear combination of these estimators using

```
function [o,v,b,V]=optimal(U)
% generates optimal linear combination of five estimators and outputs
% average estimator, variance  and weights
%  input U  a row vector of U[0,1] random numbers
T1=(.53/2)*(fn(.47+.53*U)+fn(1-.53*U));
T2=.37*.5*(fn(.47+.37*U)+fn(.84-.37*U))+.16*.5*(fn(.84+.16*U)+fn(1-.16*U));
T3=.37*fn(.47+.37*U)+.16*fn(1-.16*U);
intg=2*(.53)^3+.53^2/2;
T4=intg+fn(U)-GG(U);
T5=importance('fn',U);
X=[T1' T2' T3' T4' T5'];    % matrix whose columns are replications of the same
estimator, a row=5 estimators using same U
mean(X)
V=cov(X);  % this estimates the covariance matrix V
on=ones(5,1);
V1=inv(V);                  % the inverse of the covariance matrix
b=V1*on/(on'*V1*on);   % vector of coefficients of the optimal linear combination
o=mean(X*b);                % vector of the optimal linear combinations
v=1/(on'*V1*on);            % variance of the optimal linear combination based on
a single U
```

One run of this estimator, called with *[o,v,b,V]= optimal(unifrnd(0,1,1,1000000))* yields

$$o = 0.4615$$

$$b' = [-0.5499 \quad 1.4478 \quad 0.1011 \quad 0.0491 \quad -0.0481].$$

The estimate 0.4615 is accurate to at least four decimals which is not surprising since the variance per uniform random number input is $v = 1.13 \times 10^{-5}$. In other words, the variance of the mean based on 1,000,000 uniform input is

1.13×10^{-10}, the standard error is around .00001 so we can expect accuracy to at least 4 decimal places. Note that some of the weights are negative and others are greater than one. Do these negative weights indicate estimators that are worse than useless? The effect of some estimators may be, on subtraction, to render the remaining function more linear and more easily estimated using another method and negative coefficients are quite common in regression generally. The efficiency gain over crude Monte Carlo is an extraordinary 40,000. However since there are 10 function evaluations for each uniform variate input, the efficiency when we adjust for the number of function evaluations is 4,000. This simulation using 1,000,000 uniform random numbers and taking a 63 seconds on a Pentium IV (2.4 GHz) (including the time required to generate all five estimators) is equivalent to *forty billion simulations by crude Monte Carlo, a major task on a supercomputer!*

If we intended to use this simulation method repeatedly, we might well wish to see whether some of the estimators can be omitted without too much loss of information. Since the variance of the optimal estimator is $1/(\mathbf{1}^t V^{-1}\mathbf{1})$, we might use this to attempt to select one of the estimators for deletion. Notice that it is not so much the covariance of the estimators V which enters into Theorem 35 but its inverse $\mathbf{J} = V^{-1}$ which we can consider a type of information matrix by analogy to maximum likelihood theory. For example we could choose to delete the $i'th$ estimator, i.e. delete the $i'th$ row and column of V where i is chosen to have the smallest effect on $1/(\mathbf{1}^t V^{-1}\mathbf{1})$ or its reciprocal $\mathbf{1}^t\mathbf{J}\mathbf{1} = \sum_i \sum_j \mathbf{J}_{ij}$. In particular, if we let $V_{(i)}$ be the matrix V with the $i'th$ row and column deleted and $\mathbf{J}_{(i)} = \mathbf{V}_{(i)}^{-1}$, then we can identify $\mathbf{1}^t\mathbf{J}\mathbf{1} - \mathbf{1}^t\mathbf{J}_{(i)}\mathbf{1}$ as the loss of information when the $i'th$ estimator is deleted. Since not all estimators have the same number of function evaluations, we should adjust this information by $FE(i)$ =number of function evaluations required by the $i'th$ estimator. In other words, if an estimator i is to be deleted, it should be the one corresponding

to

$$\min_i \{\frac{\mathbf{1}^t\mathbf{J}\mathbf{1} - \mathbf{1}^t\mathbf{J}_{(i)}\mathbf{1}}{FE(i)}\}.$$

We should drop this $i'th$ estimator if the minimum is less than the information per function evaluation in the combined estimator, because this means we will increase the information available in our simulation per function evaluation. In the above example with all five estimators included, $\mathbf{1}^t\mathbf{J}\mathbf{1} = 88757$ (with 10 function evaluations per uniform variate) so the information per function evaluation is $8,876$.

i	$\mathbf{1}^t\mathbf{J}\mathbf{1} - \mathbf{1}^t\mathbf{J}_{(i)}\mathbf{1}$	$FE(i)$	$\frac{\mathbf{1}^t\mathbf{J}\mathbf{1}-\mathbf{1}^t\mathbf{J}_{(i)}\mathbf{1}}{FE(i)}$
1	88,048	2	44024
2	87,989	4	21,997
3	28,017	2	14,008
4	55,725	1	55,725
5	32,323	1	32,323

In this case, if we were to eliminate one of the estimators, our choice would likely be number 3 since it contributes the least information per function evaluation. However, since all contribute more than $8,876$ per function evaluation, we should likely retain all five.

Common Random Numbers.

We now discuss another variance reduction technique, closely related to antithetic variates called *common random numbers*, used for example whenever we wish to estimate the difference in performance between two systems or any other variable involving a difference such as a slope of a function.

Example 38 *For a simple example suppose we have two estimators $\widehat{\theta}_1, \widehat{\theta}_2$ of the "center" of a symmetric distribution. We would like to know which of these estimators is better in the sense that it has smaller variance when applied to*

a sample from a specific distribution symmetric about its median. If both estimators are unbiased estimators of the median, then the first estimator is better if

$$var(\widehat{\theta}_1) < var(\widehat{\theta}_2)$$

and so we are interested in estimating a quantity like

$$Eh_1(X) - Eh_2(X)$$

where X *is a vector representing a sample from the distribution and* $h_1(X) = \widehat{\theta}_1^2, h_2(X) = \widehat{\theta}_2^2$. *There are at least two ways of estimating these differences;*

1. *Generate samples and hence values of* $h_1(X_i), i = 1, ..., n$ *and* $Eh_2(X_j), j = 1, 2, ..., m$ *independently and use the estimator*

$$\frac{1}{n}\sum_{i=1}^{n} h_1(X_i) - \frac{1}{m}\sum_{j=1}^{m} h_2(X_j).$$

2. *Generate samples and hence values of* $h_1(X_i), h_2(X_i), i = 1, ..., n$ *independently and use the estimator*

$$\frac{1}{n}\sum_{i=1}^{n}(h_1(X_i) - h_2(X_i)).$$

It seems intuitive that the second method is preferable since it removes the variability due to the particular sample from the comparison. This is a common type of problem in which we want to estimate the difference between two expected values. For example we may be considering investing in a new piece of equipment that will speed up processing at one node of a network and we wish to estimate the expected improvement in performance between the new system and the old. In general, suppose that we wish to estimate the difference between two expectations, say

$$Eh_1(X) - Eh_2(Y) \tag{4.37}$$

where the random variable or vector X has cumulative distribution function F_X and Y has cumulative distribution function F_Y. Notice that the variance of a Monte Carlo estimator

$$var[h_1(X) - h_2(Y)] = var[h_1(X)] + var[h_2(Y)] - 2cov\{h_1(X), h_2(Y)\} \quad (4.38)$$

is *small* if we can induce a high degree of *positive correlation* between the generated random variables X and Y. This is precisely the opposite problem that led to antithetic random numbers, where we wished to induce a high degree of negative correlation. The following lemma is due to Hoeffding (1940) and provides a useful bound on the joint cumulative distribution function of two random variables X and Y. Suppose X, Y have cumulative distribution functions $F_X(x)$ and $F_Y(y)$ respectively and joint cumulative distribution function $G(x, y) = P[X \leq x, Y \leq y]$.

Lemma 39 *(a) The joint cumulative distribution function G of (X, Y) always satisfies*

$$(F_X(x) + F_Y(y) - 1)^+ \leq G(x, y) \leq \min(F_X(x), F_Y(y)) \quad (4.39)$$

for all x, y .

(b) Assume that F_X and F_Y are continuous functions. In the case that $X = F_X^{-1}(U)$ and $Y = F_Y^{-1}(U)$ for U uniform on $[0, 1]$, equality is achieved on the right $G(x, y) = \min(F_X(x), F_Y(y))$. In the case that $X = F_X^{-1}(U)$ and $Y = F_Y^{-1}(1 - U)$ there is equality on the left; $(F_X(x) + F_Y(y) - 1)^+ = G(x, y)$.

Proof. *(a) Note that*

$$P[X \leq x, Y \leq y] \leq P[X \leq x] \text{ and similarly}$$
$$\leq P[Y \leq y].$$

This shows that

$$G(x, y) \leq \min(F_X(x), F_Y(y)),$$

verifying the right side of (4.39). Similarly for the left side

$$\begin{aligned} P[X \le x, Y \le y] &= P[X \le x] - P[X \le x, Y > y] \\ &\ge P[X \le x] - P[Y > y] \\ &= F_X(x) - (1 - F_Y(y)) \\ &= (F_X(x) + F_Y(y) - 1). \end{aligned}$$

Since it is also non-negative the left side follows.

For (b) suppose $X = F_X^{-1}(U)$ *and* $Y = F_Y^{-1}(U)$, *then*

$$\begin{aligned} P[X \le x, Y \le y] &= P[F_X^{-1}(U) \le x, F_Y^{-1}(U) \le y] \\ &= P[U \le F_X(x), U \le F_Y(y)] \end{aligned}$$

since $P[X = x] = 0$ *and* $P[Y = y] = 0$.

But

$$P[U \le F_X(x), U \le F_Y(y)] = \min(F_X(x), F_Y(y))$$

verifying the equality on the right of (4.39) for common random numbers. By a similar argument,

$$\begin{aligned} P[F_X^{-1}(U) \le x, F_Y^{-1}(1-U) \le y] &= P[U \le F_X(x), 1 - U \le F_Y(y)] \\ &= P[U \le F_X(x), U \ge 1 - F_Y(y)] \\ &= (F_X(x) - (1 - F_Y(y)))^+ \end{aligned}$$

verifying the equality on the left. ■

The following theorem supports the use of common random numbers to maximize covariance and antithetic random numbers to minimize covariance.

Theorem 40 *(maximum/minimum covariance)*
Suppose h_1 *and* h_2 *are both non-decreasing (or both non-increasing) functions. Subject to the constraint that* X, Y *have cumulative distribution functions* F_X, F_Y *respectively, the covariance*

$$cov[h_1(X), h_2(Y)]$$

is maximized when $Y = F_Y^{-1}(U)$ *and* $X = F_X^{-1}(U)$ *(i.e. for common uniform*$[0,1]$ *random numbers) and is minimized when* $Y = F_Y^{-1}(U)$ *and* $X = F_X^{-1}(1-U)$ *(i.e. for antithetic random numbers).*

Proof. We will sketch a proof of the theorem when the distributions are all continuous and h_1, h_2 are differentiable. Define $G(x,y) = P[X \le x, Y \le y]$. The following representation of covariance is useful: define

$$\begin{aligned} H(x,y) &= P(X > x, Y > y) - P(X > x)P(Y > y) \\ &= G(x,y) - F_X(x)F_Y(y). \end{aligned} \tag{4.40}$$

Notice that, using integration by parts,

$$\begin{aligned} &\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(x,y)h_1'(x)h_2'(y)dxdy \\ &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\partial}{\partial x}H(x,y)h_1(x)h_2'(y)dxdy \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\partial^2}{\partial x \partial y}H(x,y)h_1(x)h_2(y)dxdy \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_1(x)h_2(y)g(x,y)dxdy - \int_{-\infty}^{\infty} h_1(x)f_X(x)dx \int_{-\infty}^{\infty} h_2(y)f_Y(y)dy \\ &= cov(h_1(X), h_2(Y)) \end{aligned} \tag{4.41}$$

where $g(x,y), f_X(x), f_Y(y)$ denote the joint probability density function, the probability density function of X and that of Y respectively. In fact this result holds in general even without the assumption that the distributions are continuous. The covariance between $h_1(X)$ and $h_2(Y)$, for h_1 and h_2 differentiable functions, is

$$cov(h_1(X), h_2(Y)) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(x,y)h_1'(x)h_2'(y)dxdy.$$

The formula shows that to maximize the covariance, if h_1, h_2 are both increasing or both decreasing functions, it is sufficient to maximize $H(x,y)$ for each x, y since $h_1'(x), h_2'(y)$ are both non-negative. Since we are constraining the marginal cumulative distribution functions F_X, F_Y, this is equivalent to maximizing

$G(x, y)$ subject to the constraints

$$\lim_{y\to\infty} G(x, y) = F_X(x)$$

$$\lim_{x\to\infty} G(x, y) = F_Y(y).$$

Lemma 37 shows that the maximum is achieved when common random numbers are used and the minimum achieved when we use antithetic random numbers. ■

We can argue intuitively for the use of common random numbers in the case of a discrete distribution with probability on the points indicated in Figure 4.5. This figure corresponds to a joint distribution with the following probabilities, say

x	0	0.25	0.25	0.75	0.75	1
y	0	0.25	0.75	0.25	0.75	1
$P[X = x, Y = y]$	.1	.2	.2	.1	.2	.2

Suppose we wish to maximize $P[X > x, Y > y]$ subject to the constraint that the probabilities $P[X > x]$ and $P[Y > y]$ are fixed. We have indicated arbitrary fixed values of (x, y) in the figure. Note that if there is any weight attached to the point in the lower right quadrant (labelled "P_2"), some or all of this weight can be reassigned to the point P_3 in the lower left quadrant provided there is an equal movement of weight from the upper left P_4 to the upper right P_1. Such a movement of weight will increase the value of $G(x, y)$ without affecting $P[X \leq x]$ or $P[Y \leq y]$. The weight that we are able to transfer in this example is 0.1, the minimum of the weights on P_4 and P_2 . In general, this continues until there is no weight in one of the off-diagonal quadrants for every choice of (x, y). The resulting distribution in this example is given by

x	0	0.25	0.25	0.75	0.75	1
y	0	0.25	0.75	0.25	0.75	1
$P[X = x, Y = y]$	.1	.3	0	.1	.3	.2

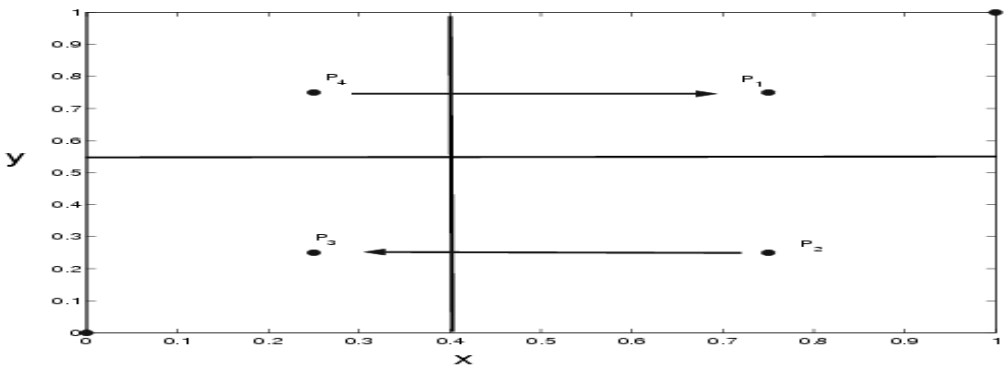

Figure 4.5: Changing weights on points to maximize covariance

and it is easy to see that such a joint distribution can be generated from common random numbers $X = F_X^{-1}(U), Y = F_Y^{-1}(U)$.

Conditioning

We now consider a simple but powerful generalization of control variates. Suppose that we can decompose a random variable T into two components T_1, ε

$$T = T_1 + \varepsilon \tag{4.42}$$

so that T_1, ε are uncorrelated

$$cov(T_1, \varepsilon) = 0.$$

Assume as well that $E(\varepsilon) = 0$. Regression is one method for determining such a decomposition and the error term ε in regression satisfies these conditions. Then T_1 has the same mean as T and it is easy to see that

$$var(T) = var(T_1) + var(\varepsilon)$$

so T_1 as smaller variance than T (unless $\varepsilon = 0$ with probability 1). This means that if we wish to estimate the common mean of T or T_1, the estimator T_1 is preferable, since it has the same mean with smaller variance.

One special case is variance reduction by *conditioning*. For the standard definition and properties of conditional expectation see the appendix. One common definition of $E[X|Y]$ is the unique (with probability one) function $g(y)$ of Y which minimizes $E\{X - g(Y)\}^2$. This definition only applies to random variables X which have finite variance and so this definition requires some modification when $E(X^2) = \infty$, but we will assume here that all random variables, say X, Y, Z have finite variances. We can define conditional covariance using conditional expectation as

$$cov(X, Y|Z) = E[XY|Z] - E[X|Z]E[Y|Z]$$

and conditional variance:

$$var(X|Z) = E(X^2|Z) - (E[X|Z])^2.$$

The variance reduction through conditioning is justified by the following well-known result:

Theorem 41 *(a)* $E(X) = E\{E[X|Y]\}$
(b) $cov(X, Y) = E\{cov(X, Y|Z)\} + cov\{E[X|Z], E[Y|Z]\}$
(c) $var(X) = E\{var(X|Z)\} + var\{E[X|Z]\}$

This Theorem is used as follows. Suppose we are considering a candidate estimator $\hat{\theta}$, an unbiased estimator of θ. We also have an arbitrary random variable Z which is somehow related to $\hat{\theta}$. Suppose that we have chosen Z carefully so that we are able to calculate the conditional expectation $T_1 = E[\hat{\theta}|Z]$. Then by part (a) of the above Theorem, T_1 is also an unbiased estimator of θ. Define

$$\varepsilon = \hat{\theta} - T_1.$$

By part (c),

$$var(\hat{\theta}) = var(T_1) + var(\varepsilon)$$

and $var(T_1) = var(\hat{\theta}) - var(\varepsilon) < var(\hat{\theta})$. In other words, for any variable Z, $E[\hat{\theta}|Z]$ has the same expectation as does $\hat{\theta}$ but smaller variance and the decrease in variance is largest if Z and $\hat{\theta}$ are nearly independent, because in this case $E[\hat{\theta}|Z]$ is close to a constant and its variance close to zero. In general the search for an appropriate Z so as to reducing the variance of an estimator by conditioning requires searching for a random variable Z such that:

1. the conditional expectation $E[\hat{\theta}|Z]$ with the original estimator is computable

2. $var(E[\hat{\theta}|Z])$ is substantially smaller than $var(\hat{\theta})$.

Example 42 *(hit or miss)*

Suppose we wish to estimate the area under a certain graph $f(x)$ by the hit and miss method. A crude method would involve determining a multiple c of a probability density function $g(x)$ which dominates $f(x)$ so that $cg(x) \geq f(x)$ for all x. We can generate points (X, Y) at random and uniformly distributed under the graph of $cg(x)$ by generating X by inverse transform $X = G^{-1}(U_1)$ where $G(x)$ is the cumulative distribution function corresponding to density g and then generating Y from the Uniform$[0, cg(X)]$ distribution, say $Y = cg(X)U_2$. An example, with $g(x) = 2x, 0 < x < 1$ and $c = 1/4$ is given in Figure 4.6.

The hit and miss estimator of the area under the graph of f obtains by generating such random points (X, Y) and counting the proportion that fall under the graph of g, i.e. for which $Y \leq f(X)$. This proportion estimates the probability

$$P[Y \leq f(X)] = \frac{\text{area under } f(x)}{\text{area under } cg(x)}$$
$$= \frac{\text{area under } f(x)}{c}$$

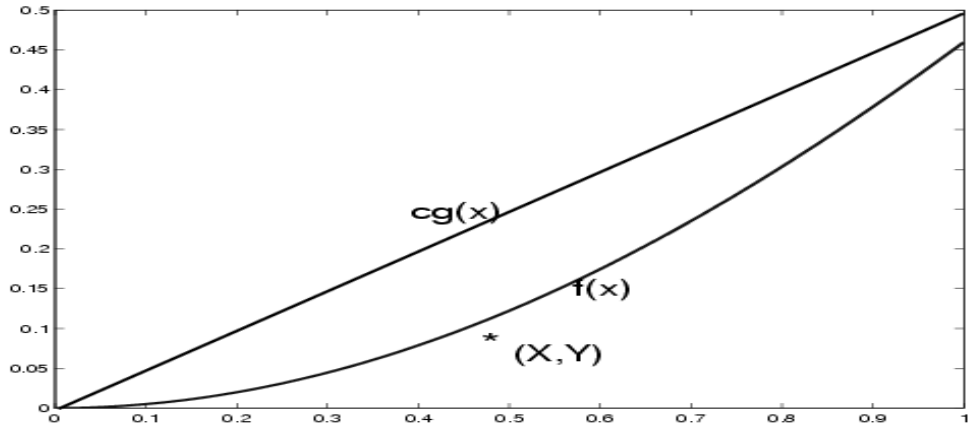

Figure 4.6: Example of the Hit and Miss Method

since $g(x)$ is a probability density function. Notice that if we define

$$W = \begin{cases} c & \text{if} \quad Y \leq f(X) \\ 0 & \text{if} \quad Y > f(X) \end{cases}$$

then

$$\begin{aligned} E(W) &= c \times \frac{\text{area under } f(x)}{\text{area under } cg(x)} \\ &= \text{area under } f(x) \end{aligned}$$

so W is an unbiased estimator of the parameter that we wish to estimate. We might therefore estimate the area under $f(x)$ using a Monte Carlo estimator $\hat{\theta}_{HM} = \frac{1}{n}\sum_{i=1}^{n} W_i$ based on independent values of W_i.This is the "hit-or-miss" estimator. However, in this case it is easy to find a random variable Z such that the conditional expectation $E(Z|W)$ can be determined in closed form. In fact we can choose $Z = X$, we obtain

$$E[W|X] = \frac{f(X)}{g(X)}.$$

This is therefore an unbiased estimator of the same parameter and it has smaller variance than does W. For a sample of size n we should replace the crude estimator $\hat{\theta}_{cr}$ by the estimator

$$\begin{aligned}\hat{\theta}_{Cond} &= \frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)} \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)}{2X_i}\end{aligned}$$

with X_i generated from $X = G^{-1}(U_i) = \sqrt{U_i}, i = 1, 2, ..., n$ and $U_i \sim$ Uniform[0,1]. In this case, the conditional expectation results in a familiar form for the estimator $\hat{\theta}_{Cond}$. This is simply an importance sampling estimator with $g(x)$ the importance distribution. However, this derivation shows that the estimator $\hat{\theta}_{Cond}$ has smaller variance than $\hat{\theta}_{HM}$.

Problems

1. Use both crude and antithetic random numbers to integrate the function

$$\int_0^1 \frac{e^u - 1}{e - 1} du.$$

   What is the efficiency gain attributed to the use of antithetic random numbers?

2. How large a sample size would I need, using antithetic and crude Monte Carlo, in order to estimate the above integral, correct to four decimal places, with probability at least 95%?

3. Under what conditions on f does the use of antithetic random numbers completely correct for the variability of the Monte-Carlo estimator? i.e. When is $var(f(U) + f(1 - U)) = 0$?

4. Show that if we use antithetic random numbers to generate two normal random variables X_1, X_2, having mean $rT - \sigma^2 T/2$ and variance $\sigma^2 T$,

this is equivalent to setting $X_2 = 2(rT - \sigma^2 T/2) - X_1$. In other words, it is not necessary to use the inverse transform method to generate normal random variables in order to permit the use of antithetic random numbers.

5. Show that the variance of a weighted average

$$var(\alpha X + (1-\alpha)W)$$

is minimized over α when

$$\alpha = \frac{var(W) - cov(X,W)}{var(W) + var(X) - 2cov(X,W)}$$

Determine the resulting minimum variance. What if the random variables X, W are independent?

6. Use a stratified random sample to integrate the function

$$\int_0^1 \frac{e^u - 1}{e-1} du.$$

What do you recommend for intervals (two or three) and sample sizes? What is the efficiency gain?

7. Use a combination of stratified random sampling and an antithetic random number in the form

$$\frac{1}{2}[f(U/2) + f(1 - U/2)]$$

to integrate the function

$$\int_0^1 \frac{e^u - 1}{e-1} du.$$

What is the efficiency gain?

8. In the case $f(x) = \frac{e^x - 1}{e-1}$, use $g(x) = x$ as a control variate to integrate over [0,1]. Show that the variance is reduced by a factor of approximately 60. Is there much additional improvement if we use a more general quadratic function of x?

9. In the case $f(x) = \frac{e^x-1}{e-1}$, consider using $g(x) = x$ as a control variate to integrate over [0,1]. Note that regression of $f(U)$ on $g(U)$ yields $f(U)-E(f(U)) = \beta[g(U)-Eg(U)]+\varepsilon$ where the error term ε has mean 0 and is uncorrelated with $g(U)$ and $\beta = cov(f(U), g(U))/var(g(U)$. Therefore, taking expectations on both sides and reorganising the terms, $E(f(U)) = f(U) - \beta[g(U) - E(g(U))]$. The Monte-Carlo estimator

$$\frac{1}{n}\sum_{i=1}^{n}\{f(U_i) - \beta[g(U_i) - E(g(U_i))]\}$$

is an improved control variate estimator, equivalent to the one discussed above in the case $\beta = 1$. Determine how much better this estimator is than the basic control variate case $\beta = 1$ by performing simulations. Show that the variance is reduced by a factor of approximately 60. Is there much additional improvement if we use a more general quadratic function of x?

10. A call option pays an amount $V(S) = 1/(1 + \exp(S(T) - k))$ at time T for some predetermined price k. Discuss what you would use for a control variate and conduct a simulation to determine how it performs, assuming geometric Brownian motion for the stock price, interest rate 5%, annual volatility 20% and various initial stock prices, values of k and T.

11. It has been suggested that stocks are not log-normally distributed but the distribution can be well approximated by replacing the normal distribution by a student t distribution. Suppose that the daily returns X_i are independent with probability density function $f(x) = c(1+(x/b)^2)^{-2}$ (the re-scaled student distribution with 3 degrees of freedom). We wish to estimate a weekly $Var_{.95}$, a value e^v such that $P[\sum_{i=1}^{5} X_i < v] = 0.95$. If we wish to do this by simulation, suggest an appropriate method involving importance sampling. Implement and estimate the variance reduction.

12. Suppose, for example, I have three different simulation estimators Y_1, Y_2, Y_3

whose means depend on two unknown parameters θ_1, θ_2. In particular, suppose Y_1, Y_2, Y_3, are unbiased estimators of $\theta_1, \theta_1 + \theta_2, \theta_2$ respectively. Let us assume for the moment that $var(Y_i) = 1, cov(Y_i, Y_j) = -1/2$. I want to estimate the parameter θ_1. Should I use only the estimator Y_1which is the unbiased estimator of θ_1, or some linear combination of Y_1, Y_2, Y_3? Compare the number of simulations necessary for a certain degree of accuracy.

13. Consider the *systematic sample* estimator based on the trapezoidal rule:

$$\hat{\theta} = \frac{1}{n} \sum_{i=0}^{n-1} f(V + i/n), V \sim U[0, \frac{1}{n}]$$

Discuss the bias and variance of this estimator. In the case $f(x) = x^2$, how does it compare with other estimators such as crude Monte Carlo and antithetic random numbers requiring n function evaluations. Are there any disadvantages to its use?

14. In the case $f(x) = \frac{e^x - 1}{e - 1}$, use $g(x) = x$ as a control variate to integrate over [0,1]. Find the optimal linear combination using estimators (4.34) and (4.35), an importance sampling estimator and the control variate estimator above. What is the efficiency gain over crude Monte-Carlo?

15. The *rho* of an option is the derivative of the option price with respect to the interest rate parameter r. What is the value of ρ for a call option with $S_0 = 10$, strike=10, $r = 0.05, T = .25$ and $\sigma = .2$? Use a simulation to estimate this slope and determine the variance of your estimator. Try using (i) independent simulations at two points and (ii) common random numbers. What can you say about the variances of your estimators?

16. For any random variables X, Y, prove that $P(X \leq x, Y \leq y) - P(X \leq x)P(Y \leq y) = P(X > x, Y > y) - P(X > x)P(Y > y)$ for all x, y.

17. Show that the Jacobian of the transformation used in the proof of Theorem 23; $(x, m) \to (x, y)$ where $y = \exp(-(2m-x)^2/2)$ is given by $\frac{\sqrt{1}}{2y\sqrt{-2\ln(y)}}$.