Imperial College
London

UNIVERSITY OF LONDON

Course: M3S9/M4S9 (SOLUTIONS)
Setter: Adams
Checker: McCoy
Editor: Coleman
External: Kent
Date: December 7, 2004

BSc and MSci EXAMINATIONS (MATHEMATICS)
MAY–JUNE 2004

M3S9/M4S9 (SOLUTIONS)

STOCHASTIC SIMULATION

Setter's signature	Checker's signature	Editor's signature
.....	.....	.....

1. (a) Congruential generators define a recursion. There are two cases. The *mixed* generator is

seen ⇓

$$X_{n+1} = (aX_n + b) \text{mod}\, m$$

and the multiplicative generator is

$$X_{n+1} = aX_n \text{mod}\, m$$

where a is the *multiplier*, b is the *shift*, m is the *modulus*, and X_0 is the *seed*, and $a, b, X_0 < m$. In either case, we obtain psuedorandom uniform variates U from

$$U_i = \frac{X_i}{m}$$

A multiplicative generator is maximal if it has period $m - 1$. 5

(b) (i) The frequency test compares the number of times each digit was observed with the number of times expected under the null hyopthesis. In this case, we assume uniformity, so the expected values are

seen ⇓

$$E_0 = E_1 = \ldots = E_k = \frac{n}{k+1}$$

and these are compared with observed values $O_0, O_1, \ldots, O_k$.
The test statistic is

$$\sum_{i=0}^{k} \frac{(O_i - E_i)^2}{E_i}$$

and this is compared with a Chi-squared distribution with k degrees of freedom. 4

(ii) On the basis of this output, there is no evidence, at the 5% significance level, to reject the hypothesis that the generator is not uniform.

unseen ⇓

(iii) The Frequency test does not consider the correlation structure of the data. An example of data which would pass the test but which is not consistent with our requirements for randomness is 2

unseen ⇓

`0 0 0 0 0 0 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4 4 ...` 2

(iv) Could consider dividing (a, b) into k bins of equal width, and counting how many variates fall in each bin. Then compare these observed counts with the probabilities implied by the continuous distribution.

unseen ⇓

2

(c) (i) By choosing a random seed on each processor, we run the risk of generating overlapping sequences. unseen ⇓

1

(ii) From an arbitrary seed X_0, this results enables us to start the first processor with X_1, the second processor with X_{n+1}, the third processor with X_{2n+1} and so on. for example, with $M = 12$ and three processors, each processor compute $n = 4$ random numbers as follows unseen ⇓

X0 ->	Processor 1	Processor 2	Processor 3
	X1	X5	X9
	X2	X6	X10
	X3	X7	X11
	X4	X8	X12

Thereby using exactly the same sequence as would be used on a single processor starting with X_0.

A problem with implementing this algorithm could by computing X_n^k for large k. 4

2. (a) Want $X \sim \exp(\lambda)$. So, seen ⇓

$$F_X(X) = 1 - \mathrm{e}^{-\lambda X}$$
$$\text{Setting } U = 1 - \mathrm{e}^{-\lambda X},$$
$$\Rightarrow X = \frac{-1}{\lambda}\log(1-U) \Rightarrow F_X^{-1}(U) = -\lambda^{-1}\log(1-U)$$

So an inversion algorithm for exponential variates is

1. Generate $U = u \sim U(0,1) \Rightarrow 1 - U \sim U(0,1)$
2. Set $X = -\lambda^{-1}\log(U)$. Then $X \sim \exp(\lambda)$

5

(b) The moment generating function for an Exponential distribution is unseen ⇓

$$\begin{aligned} M_X(t) &= \int_0^\infty e^{tx}\lambda e^{-\lambda x}\,dx \\ &= \lambda\int_0^\infty e^{-x(\lambda-t)} \\ &= \lambda^n\left[\frac{e^{-x(\lambda-t)}}{\lambda-t}\right]_0^\infty \\ &= \frac{\lambda^n}{\lambda-t}, \quad \lambda > t \end{aligned}$$

The moment generating function for a Gamma distribution is

$$\begin{aligned} M_X(t) &= \int_0^\infty e^{tx}\frac{\lambda^n}{(n-1)!}x^{n-1}e^{-\lambda x}\,dx \\ &= \frac{\lambda}{(n-1)!}\int_0^\infty x^{n-1}e^{-(\lambda-t)x}\,dx \\ \text{But } &\int_0^\infty x^k e^{-\lambda x}\,dx = k!/\lambda^{k+1} \\ &= \frac{\lambda}{(n-1)!}\frac{(n-1)!}{(\lambda-t)^n} = \left[\frac{\lambda}{(\lambda-t)}\right]^n \end{aligned}$$

with $n \geq 1$ and $\lambda > t$.
Now, the moment generating function of a sum of n independent random variables is the product of the n moment generating functions, so for a sum of n exponential variables we have

$$\left[\frac{\lambda}{(\lambda-t)}\right]^n$$

which is the moment generating function of the Gamma(n,λ) distribution. 5

(c) Using the exponential algorithm as a basis, we have

unseen ⇓

1. Generate $U_i = u_i \sim U(0,1), i = 1, 2, \ldots, n$.
2. Set $X = -\lambda^{-1}\log(\prod_{i=1}^{n} U_i)$. Then $X \sim \text{Gamma}(n, \lambda)$

Note that in the second step, we only compute a logarithm once.

2

(d) We have

unseen ⇓

$$f(x) = kx^{\frac{1}{2}}e^{-x} \quad \text{and} \quad g(x) = \frac{2}{3}e^{-\frac{2}{3}x}$$

where $k = \frac{2}{\sqrt{\pi}}$ and $x > 0$.

$$\begin{aligned} \text{Let } y &= \frac{f(x)}{g(x)} = \frac{kx^{\frac{1}{2}}e^{-x}}{\frac{2}{3}e^{-\frac{2}{3}x}} \\ &= \frac{3kx^{\frac{1}{2}}e^{-\frac{x}{3}}}{2} \end{aligned}$$

Maximise y

$$\begin{aligned} \frac{dy}{dx} &= \frac{3kx^{\frac{1}{2}}}{2}\left(-\frac{1}{3}\right)e^{-\frac{x}{3}} + e^{-\frac{x}{3}}\frac{3k}{4}x^{-\frac{1}{2}} = 0 \\ e^{-\frac{x}{3}}\frac{3k}{4}x^{-\frac{1}{2}} &= \frac{kx^{\frac{1}{2}}}{2}e^{-\frac{x}{3}} \end{aligned}$$

so the maximum is obtained at $x = \frac{3}{2}$. Now,

$$M = \sup_x \frac{f(x)}{g(x)} = \frac{3k}{2}\left(\frac{3}{2}\right)^{\frac{1}{2}}e^{-\frac{1}{2}} = \frac{3^{3/2}}{(2\pi e)^{1/2}}$$

Since $k = 2/\sqrt{\pi}$.

A rejection algorithm for generating Gamma$(\frac{3}{2}, 1)$ variates is

1. Generate $U_1 = u_1 \sim U(0,1)$.
2. Set $Y = -\frac{3}{2}\log(U_1)$ (so $Y \sim \text{Exp}(2/3)$ (by inversion))
3. Generate $U_2 = u_2 \sim U(0,1)$.
4. If $U_2 < \sqrt{\frac{2}{3}}e^{\frac{1}{2}}Y^{\frac{1}{2}}e^{-\frac{Y}{3}}$, set X=Y, and $X \sim \text{Gamma}(3/2, 1)$.
5. Otherwise goto 1.

8

seen ⇓

3. (a) The formula for the Monte Carlo estimator is

$$\widehat{\theta} = \frac{1}{n}\sum_{i=1}^{n}\phi(X_i)$$

where $X_1, X_2, \ldots, X_n \overset{\text{ind}}{\sim} f(\cdot)$. [1]

The expected value is

$$\begin{aligned} E_f[\widehat{\theta}] &= E_f\left[\frac{1}{n}\sum_{i=1}^{n}\phi(X_i)\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}E_f[\phi(X_i)] \\ &= \frac{1}{n}nE_f[\phi(X_i)] \\ &= \int f(x)\phi(x)\,\mathrm{d}x = \theta \end{aligned}$$

[2]

and the variance is

$$\begin{aligned} \text{var}(\widehat{\theta}) &= \text{var}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(X_i)\right) \\ &= \frac{1}{n^2}(n\text{var}(\phi(X)) \\ &= \frac{1}{n}\text{var}(\phi(X)) \\ &= \frac{1}{n}E[\{\phi(X) - E(\phi(X))\}^2] \\ &= \frac{1}{n}\int[\phi(x)-\theta]^2 f(x)\,\mathrm{d}x = \frac{c}{n} \end{aligned}$$

So the variance of the estimator decreases as n increases. [2]

(b) Like any computational approximation, Monte Carlo integration methods are only appropriate when analytic solutions are unavailable. Typically, numerical integration methods are faster and more accurate in low dimensions. In higher dimensions numerical integration becomes more difficult, and it is in these situations that Monte Carlo integration methods are most suitable. [2]

unseen ⇓

(c) For two unbiased estimators of θ, $\widehat{\theta}_1$ and $\widehat{\theta}_2$, with variances $\text{var}(\widehat{\theta}_1)$ and $\text{var}(\widehat{\theta}_2)$

seen ⇓

$$E\left[\frac{1}{2}\left(\widehat{\theta}_1 + \widehat{\theta}_2\right)\right] = \theta$$

$$\text{var}\left[\frac{1}{2}\left(\widehat{\theta}_1 + \widehat{\theta}_2\right)\right] = \frac{1}{4}\text{var}(\widehat{\theta}_1) + \frac{1}{2}\text{var}(\widehat{\theta}_2) + \frac{1}{2}\text{cov}(\widehat{\theta}_1, \widehat{\theta}_2)$$

Now suppose

$$\text{var}(\widehat{\theta}_1) = \text{var}(\widehat{\theta}_2)$$

$$\text{var}\left[\frac{1}{2}\left(\widehat{\theta}_1 + \widehat{\theta}_2\right)\right] = \frac{1}{2}\text{var}(\widehat{\theta}_1)\left[1 + \text{corr}(\widehat{\theta}_1, \widehat{\theta}_2)\right]$$

So if we can arrange for corr$(\widehat{\theta}_1, \widehat{\theta}_2)$ to be large and negative, then var $\left[\frac{1}{2}\left(\widehat{\theta}_1 + \widehat{\theta}_2\right)\right] << \text{var}(\widehat{\theta}_1)$. Antithetic variates try to exploit this by taking the average of negatively correlated estimates to get variance reduction.
As a simple example, if we want to evaluate

$$\theta = \int_0^1 \phi(u) f(u)\, du$$

where $f \sim U(0,1)$, and ϕ is monotonic, the the antithetic estimator is

$$\widehat{\theta} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \left(\phi(U_i) + \phi(1 - U_i)\right)$$

4

unseen ⇓

(d) (i) The decomposition here is

$$f(x) = 30x^2(1-x)^2 \qquad \phi(x) = I(x > Z) = \begin{cases} 1 & x > Z, \\ 0 & x \leq Z. \end{cases}$$

Then

$$\widehat{\theta} = \frac{Y}{n}$$

where Y is the number of $X_i > Z$, and $X_i \sim$ Beta$(3,3)$, for $i = 1, 2, \ldots, n$.
Now

$$\begin{aligned} \text{var}(\widehat{\theta}) &= \frac{1}{n^2}\text{var}(Y) \\ &= \frac{1}{n^2} n\theta(1-\theta) \\ &= \frac{1}{n}\theta(1-\theta) \end{aligned}$$

4

(ii) Another simple possibility is to consider both tails, that is, consider $\theta = P(|X|) > Z$.

1

(e) Note that this looks like $E[X^2\cos(X^2)]$, where $X \sim$ Exp(25). Thus, an appropriate decomposition may by

unseen ⇓

$$\phi(x) = x^2\cos(x^2) \qquad f(x) = 25e^{-25x}$$

In constructing a Monte Carlo algorithm to estimate I, we require to sample from an exponential distribution, readily achieved by inversion (see previous question).

Monte Carlo Integration Algorithm

1. Obtain $U_i \sim$ Uniform$(0,1)$, independently for $i = 1, 2, \ldots, n$.
2. Set $X_i = \frac{-1}{25}\log_e(U_i)$, for $i = 1, 2, \ldots, n$.
3. Set $T_i = X_i^2\cos(X_i^2)$, for $i = 1, 2, \ldots, n$.
4. Set $I = \frac{1}{n}\sum_{i=1}^{n} T_i$.

4

4. (a) Methods for Generating Normal random deviates

The N(μ, σ) distribution has density

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{-(x-\mu)^2}{2\sigma^2}\right]$$

where $-\infty \leq x, \mu \leq \infty$ and $\sigma > 0$. The standardising transformation

$$Z = \frac{X - \mu}{\sigma}$$

and its inverse allow us restrict attention to generating standard Normal deviates $Z = N(0,1)$. The distribution function of a Normal random variable is not analytically available, and hence the method of inversion cannot be used.

We can consider either general purpose methods for generating from arbitrary distributions, or methods purpose built for the Normal distribution.

First, consider purpose built methods: the Box-Muller method, and the Polar-Marsaglia algorithm.

Box-Muller Algorithm

1. Generate $U_1 \sim U(0,1), \quad U_2 \sim U(0,1)$.
2. Set $R = (-2\log U_2)^{1/2}, \quad A = 2\pi U_1$.
3. Set $X = R\cos A, \quad Y = R\sin A$, then $X, Y \sim$ N$(0,1)$.

A possible disadvantage of this is that computing transcendental functions like sin and cos can be expensive. The Polar-Marsaglia algorithm attempts to avoid this, albeit at the cost of including a rejection step.

Polar-Marsaglia Algorithm

1. Generate $U_i = u_i \sim U(0,1)$ for $i = 1, 2$.
2. Set $V_i = 2u_i - 1$
3. If $S = V_1^2 + V_2^2 \leq 1$
   Let
   $$C = \sqrt{-\frac{2}{S}\log S} \quad \text{set} \quad \begin{matrix} X = CV_1 \\ Y = CV_2 \end{matrix}$$
   Then $X, Y \sim$ N(0,1).
4. Otherwise GOTO 1

Next, we consider general purpose methods. These include ratio-of-uniforms, and the rejection method.

Ratio-of-Uniforms Algorithm

1. Find bounding rectangle for C_h (*i.e* find a, b and c).
2. Generate $(U_1, U_2) \sim U(0,1)$.
3. Set $U = aU_1$, $V = b + (c-b)U_2$.
4. if $U \leq \sqrt{h\left(\frac{V}{U}\right)}$, set $X = \frac{V}{U}$, otherwise GOTO 1

Alternatively, consider the general rejection algorithm

Rejection Algorithm

1. Generate $Y = y \sim g(.)$.
2. Generate $U = u \sim U(0,1)$.
3. If $u \leq \frac{f(y)}{Mg(y)}$ set $X = y$.
4. Otherwise GOTO 1.

Where

$$M = \sup_x \frac{f(x)}{g(x)}$$

To generate Normal deviates, we might use a Cauchy envelope. In either case, we might consider pre-test squeezes to reduce the number of times we have to compute expensive tests.

Splus uses ratio-of-uniforms to generate Normal deviates. Ripley (1987) recommends either the Polar algorithm, or a squeezed version of ratio of uniforms.

(b) Ratio of uniforms

If (U,V) is a uniformly distributed pair in the unit disc, the ratio U/V from the polar algorithm has the distribution of the ratio of two independent Normal variates. Kinderman and Monahan (1977) considered whether other distributions could be sampled as U/V uniform for (U,V) uniform over some set.

h - function with $h(\cdot) \geq 0$ and $\int h < \infty$. Consider the region in (U,V) space defined by:

$$C_h = \left\{ (U,V) \mid 0 \leq U \leq \sqrt{h\left(\frac{V}{U}\right)} \right\}$$

We show that C_h has finite area, and that if (U,V) are uniform on C_h, then $X = \frac{V}{U}$ has p.d.f. $\frac{h}{\int h}$.

Proof

$$\text{Area}(C_h) = \int\int_{C_h} \mathrm{d}u\,\mathrm{d}v \qquad \left((u,v) \to \left(u, x = \frac{v}{u}\right)\right)$$

So,

$$v = xu \;\Rightarrow\; \frac{\mathrm{d}v}{\mathrm{d}x} = u \;\Rightarrow\; \mathrm{d}v = u\,\mathrm{d}x,$$

and

$$0 \leq u \leq \sqrt{h\left(\frac{v}{u}\right)} \;\Rightarrow\; 0 \leq u \leq \sqrt{h(x)}$$

Giving,

$$\text{Area}(C_h) = \int \left[\int_0^{\sqrt{h(x)}} u \mathrm{d}u \right] \mathrm{d}x = \boxed{\int \tfrac{1}{2} h(x) \mathrm{d}x} < \infty$$

$$\begin{aligned}
f_{U,V}(u,v) &= \begin{cases} \frac{1}{\text{Area}(C_h)} & (u,v) \in C_h \\ 0 & \text{otherwise} \end{cases} \\
f_{U,X}(u,x) &= f_{U,V}(u,ux).u \qquad (u = |J|) \\
&= \frac{u}{\text{Area}(C_h)} \\
f_X(x) &= \int_u f_{U,X}(u,x)\mathrm{d}u = \int_0^{\sqrt{h(x)}} \frac{u}{\text{Area}(C_h)} \mathrm{d}u \\
&= \frac{\frac{1}{2}h(x)}{\frac{1}{2}\int h(x)\mathrm{d}x} = \frac{h}{\int h} \qquad \text{as required}
\end{aligned}$$

Note: only need h up to proportionality.

We need to generate numbers within C_h, one way of doing this is to bound the C_h region within a rectangle.

$$C_h = \left\{ (U,V) \mid 0 \le U \le \sqrt{h\left(\frac{V}{U}\right)} \right\}$$

$\underline{a}$ $0 \le U \le \sup_x \sqrt{h(x)} = a$.

$\underline{b,c}$ $x = \frac{V}{U}$, $\frac{V}{x} \le \sqrt{h(x)}$.

$x \le 0 \quad V \ge x\sqrt{h(x)} \Rightarrow V \ge \inf_{x\le 0} x\sqrt{h(x)} = b.$

$x \ge 0 \quad V \le x\sqrt{h(x)} \Rightarrow V \le \sup_{x\ge 0} x\sqrt{h(x)} = c.$

So, set

$$\boxed{a = \sup_x \sqrt{h(x)} \quad b = \inf_{x\le 0} x\sqrt{h(x)} \quad c = \sup_{x\ge 0} x\sqrt{h(x)}}$$

Such a rectangle will always exist provided $h(x)$ and $x^2 h(x)$ are bounded in the domain of x.

Algorithm

1. Find bounding rectangle for C_h (*i.e* find a, b and c).
2. Generate $(U_1, U_2) \sim U(0,1)$.
3. Set $U = aU_1$, $V = b + (c-b)U_2$.
4. if $U \le \sqrt{h\left(\frac{V}{U}\right)}$, set $X = \frac{V}{U}$, otherwise GOTO 1

Can compute the probability of accepting a proposed point

$$\begin{aligned} \text{Probability of accepting an } X &= \frac{\text{Area}(C_h)}{\text{Area of bounding rectangle}} \\ &= \frac{\frac{1}{2}\int h(x)\,\mathrm{d}x}{a(c-b)} \end{aligned}$$

The method uses a membership test $(U,V) \in C_h$ that can be slow to evaluate. These tests will often be clearly passed or failed and a simple approximation to C_h should suffice. We find sets $C_i \subset C_h \subset C_o$ with $(u,v) \subset C_i$ and $(u,v) \ni C_o$ being easy top determine. Such procedures are called *pre-testing* or *squeezing*.

(c) Metropolis hastings algorithm

Suppose, we want to obtain samples from some distribution $\underline{\pi}$, which has a large number of states and so cannot be sampled from directly.

Set up an irreducible, aperiodic Markov Chain (i.e. define a transition matrix P), with stationary distribution $\underline{\pi}$. Then run this chain until it settles down to $\underline{\pi}$ – states will then be generated with the correct probabilities.

We construct P in the following way:
Choose any symmetric transition matrix Q, with elements q_{ij}.
Suppose at state i, select state j with probability q_{ij}.
Move to state j with probability $\min\left\{1, \frac{\pi_j}{\pi_i}\right\}$, otherwise stay at state i. So, if $\pi_j < \pi_i$, we will not always move.
This defines P in the following way:

$$\begin{aligned} p_{ij} &= \min\left\{1, \frac{\pi_j}{\pi_i}\right\} q_{ij}, \qquad i \neq j \\ p_{ii} &= \underbrace{q_{ii}}_{\substack{\text{prob. choose} \\ \text{state } i}} + \sum_{j \neq i} \underbrace{\max\left\{0, 1 - \frac{\pi_j}{\pi_i}\right\}}_{\substack{\text{prob. not moving} \\ \text{to state } j}} q_{ij} \end{aligned}$$

P defined in this way is certainly a transition matrix (as Q is), but will running the chain for a long time give us $\underline{\pi}$ as the limiting distribution?
For this, we need to show that P is irrecucible, aperiodic and that $\underline{\pi} = \underline{\pi}P$.
To show $\underline{\pi} = \underline{\pi}P$, if suffices to show $\pi_i p_{ij} = \pi_j p_{ji}$. As,

$$\begin{aligned} \Rightarrow \sum_j \pi_i p_{ij} &= \sum_j \pi_j p_{ji} \\ \Rightarrow \pi_i \sum_j p_{ij} &= (\underline{\pi}P)_i \\ \Rightarrow \pi_i &= (\underline{\pi}P)_i \quad \text{(rows of } P \text{ sum to 1)} \\ \Rightarrow \underline{\pi} &= \underline{\pi}P \end{aligned}$$

So, for the P associated with Metropolis, we have

$$\begin{aligned} \pi_i p_{ij} &= \pi_i \min\left\{1, \frac{\pi_j}{\pi_i}\right\} q_{ij} \\ &= \min\left\{\pi_i, \pi_j\right\} q_{ij} \\ &= \min\left\{\pi_i, \pi_j\right\} q_{ji} \quad \text{since } Q \text{ symmetric} \\ &= \pi_j \min\left\{1, \frac{\pi_i}{\pi_j}\right\} q_{ji} \\ &= \pi_j p_{ji} \\ \Rightarrow \underline{\pi} &= \underline{\pi}P \end{aligned}$$

If we can now show that the chain is irreducible and aperiodic, $\underline{\pi}$ will be the unique stationary distribution.

Irreducible?
If Q is irreducible, then so is P (eg $p_{ij} > 0$ iff $q_{ij} > 0$), so choose an irreducible Q.
Aperiodic?
If we can show $p_{ii} > 0$, then state i is aperiodic, and thus all states are aperiodic (as there is only one class).

$$p_{ii} = q_{ii} + \sum_{j \neq i} \max \left\{ 0, 1 - \frac{\pi_j}{\pi_i} \right\} q_{ij}$$

so, $p_{ii} > q_{ii}$ – easy condition to impose.
Basically, choose a Q which ensures irreducibility and aperiodicity.