Imperial College
London

UNIVERSITY OF LONDON

BSc and MSci EXAMINATIONS (MATHEMATICS)
MAY–JUNE 2004

This paper is also taken for the relevant examination for the Associateship.

M3S9/M4S9 Stochastic Simulation

Date: Thursday, 3rd June 2004 Time: 2 pm – 3.30 pm

Credit will be given for all questions attempted but extra credit will be given for complete or nearly complete answers.

There are **four questions** *only, and the exam lasts* 1.5 *hours.*
A project is set which carries the credit of the fifth question.
Calculators may not be used.
Statistical tables will not be available.

1. **(a)** Define recurrence formula for both the mixed and multiplicative congruential pseudorandom number generators. Clearly define all parameters. How are uniform numbers on $[0,1]$ obtained from these generators? What does it mean for a multiplicative generator to have maximal period?

(b) **(i)** Suppose a pseudorandom number generator is claimed to output discrete uniform random variables on the range $0, 1, 2, \ldots, k$, and a sequence of n random numbers is available. Fully describe the 'Frequency Test of Digits' for testing the uniformity of this generator.

(ii) Suppose an Splus function called `freq.test` outputs the value of the test statistic for the 'Frequency Test for Digits'. Consider the following Splus session commands and output, where the object `data` is claimed to be sampled from a discrete uniform distribution on $0, 1, 2, \ldots, 20$.

```
> length(data)
[1] 1000
> freq.test(data)
[1] 10.436
> qchisq(0.95,20)
31.41042
```

What would this output lead you to conclude?

(iii) Give a short example of output from a discrete uniform pseudorandom generator which would not be rejected by the frequency test, but would still not be acceptable as output from a pseudorandom number generator.

(iv) Describe how you could modify the frequency test to deal with continuous pseudorandom numbers distributed uniformly on (a, b).

(c) A distributed memory parallel computer is a collection of p independent processors, each processor having its own memory. We can generate M independent pseudorandom numbers $\{X_1, \ldots, X_M\}$ by generating p independent sequences of length n: $\{X_1^{(j)}, \ldots, X_n^{(j)}\}$, $j = 1, \ldots, p$ where $n = \lfloor (M-1)/p \rfloor + 1$, and taking

$$X_i = X_k^{(j)} \quad i = 1, \ldots, M; \;\; k = (i-1)\text{mod}(n) + 1, \;\; j = \lfloor (i-1)/n \rfloor + 1,$$

where $\lfloor x \rfloor$ is the smallest integer less than or equal to x.

(i) Suggest a reason why starting a congruential generator on each processor with a randomly generated seed may not be a good idea.

(ii) The *leapfrog* method of computing pseudorandom numbers is based on the result that X_{n+k} $(k \geq 1)$ can be calculated as a function of X_n^k . Describe how this result could be used to reproduce $\{X_1, \ldots, X_M\}$ exactly as if it were computed on a single processor. What other problems might occur in implementing this result?

2. **(a)** The random variable X has an Exponential(λ) distribution with density function

$$f_X(x) = \lambda e^{-\lambda x} \qquad x > 0, \lambda > 0$$

Describe the inversion algorithm for generating exponential random variates.

(b) Using moment generating functions, show that a Gamma(n,λ) random variable, with density function

$$f_X(x) = \frac{\lambda^n}{(n-1)!} x^{n-1} e^{-\lambda x} \quad x > 0, \lambda > 0$$

can be expressed as the sum of n independent exponential random variables, each having the Exponential(λ) distribution.

[Recall that $\int_0^\infty x^k e^{-\lambda x}\, dx = k!/\lambda^{k+1}$.]

(c) Modify the algorithm in part **(a)** using the result in part **(b)** to generate Gamma(n, λ) random variables. Pay attention to the efficiency of the algorithm.

(d) Construct a rejection sampling algorithm for generating a random variable, X, from the Gamma$\left(\frac{3}{2}, 1\right)$ density,

$$f_X(x) = k x^{\frac{1}{2}} e^{-x} \qquad x > 0,$$

where $k = \frac{2}{\sqrt{\pi}}$, using an Exponential$\left(\frac{2}{3}\right)$ as the rejection envelope.

3. **(a)** Define the Monte Carlo estimator, $\widehat{\theta}$, of the integral

$$\theta = \int \phi(x) f_X(x)\, \mathrm{d}x,$$

using a sequence $\{X_1, X_2, \ldots, X_n\}$ of independent random variates with density function $f_X(x)$. Show that $\widehat{\theta}$ is unbiased, and that the variance of $\widehat{\theta}$ decreases as n increases.

(b) Describe situations where Monte Carlo integration maybe preferable to quadrature.

(c) Describe how antithetic variates can be used to reduce the variance of the estimator in **(a)**.

(d) The Beta(3,3) distribution has probability density function

$$30x^2(1-x)^2 \qquad 0 \le x \le 1$$

Note that this distribution is symmetric about 0.5. Consider

$$\theta = \int_Z^1 30x^2(1-x)^2\, dx$$

where Z is a constant.

(i) Consider the case of sampling n Beta(3,3) variates and estimating θ as the proportion greater than Z. Write down the ϕ and f_X decomposition, and write down an expression for the variance of the estimator so obtained.

(ii) Propose another possible ϕ and f_X decomposition for estimating θ by Monte Carlo integration.

(e) Consider the integral

$$I = \int_0^\infty 25x^2 \cos(x^2) e^{-25x}\, \mathrm{d}x$$

Suggest an appropriate ϕ and f_X decomposition and hence suggest a Monte Carlo algorithm for computing $\widehat{I}$, the Monte Carlo estimator of I.

4. Describe two of the following topics. You should provide detailed algorithms and take care to define all notation introduced.

(a) Methods for generating Normal random variates.

(b) The Ratio of uniforms method.

(c) The Metropolis-Hastings algorithm.