Uniform pseudo-random number generators

Stochastic simulation methods rely on the possibility of producing (with a computer) a supposedly endless flow of iid (i.e. independent and identically distributed) random variables which are uniformly distributed on $[0, 1]$. The uniform random variables are produced by a so-called random number generator, also called a pseudo-random number generator since in reality anything produced by a computer is deterministic:

**Definition** A uniform pseudo-random number generator is an algorithm which, starting from an initial value $U_0 \in [0, 1]$ and a transformation $D$, produces a sequence $U_0, U_1, \ldots \in [0, 1]$ with $U_{i+1} = D(U_i)$, $i = 0, 1, \ldots$ and such that for all $n$, $(U_1, \ldots, U_n)$ reproduces the behaviour of an iid sample $(V_1, \ldots, V_n)$ of uniform random variables when compared through a usual set of tests.

The following exercises aim at giving an introduction to uniform random number generation; we shall later see how to use this for simulating random variables from standard distributions as well as more complicated distributions. For further details, see e.g.


and the references therein.

**Exercise 1 (Multiplicative congruential generators)**

In many cases a multiplicative congruential generator with parameters $a, m$ is used (more precisely this is often just one ingredient of a more complicated generator) where $a \geq 1$
and $m \geq 2$ are integers. This produces a sequence of integers by the recursion
\begin{align*}
X_0 & \in \{1, \ldots, m - 1\} \\
x_{i+1} &= aX_i \mod m. 
\end{align*}

Here $X_0$ is called the seed. The associated sequence of approximately iid uniform random variables is given by $U_i = X_i / m$. Note that $a$ and $m$ must be chosen such that $X_i \neq 0$ for all $i$ (otherwise it gets stuck as $X_i = 0$ for all sufficiently large $i$). The choice of $a$ and $m$ is of course crucial for the quality of the generator.

1. Show that if $a$ and $m$ have no common prime factor, then $X_i \neq 0$ for $i = 0, 1, \ldots$.
   Hint: Argue that if $X_{i+1} = 0$, then $aX_i = km$ for some integer $k > 0$, and hence we obtain a contradiction.

2. Show that any uniform pseudo-random number generator will repeat itself, i.e.
\[(U_i, U_{i+1}, U_{i+2}, \ldots) = (U_i, \ldots, U_{i+p-1}, U_i, \ldots, U_{i+p-1}, \ldots)\]
for some integers $0 \leq i < p \leq m$; if $p$ is the smallest such integer, it is called the period.

3. Implement in R a multiplicative congruential generator where it is possible to use different values of $a, m$ and $X_0$.
   Hint: Make first a function (see the R manual) for modulus operation (here the R-function `floor` may be helpful), and then a function which takes $a, m, X_0$, and $n$ as input and returns a vector of length $n$ containing $(U_1, \ldots, U_n)$.

4. Generate $U_1, \ldots, U_{1000}$ and show a histogram of these 1000 values using the R function `hist`, setting first $a = 3, m = 31, X_0 = 2$ and next $a = 65539, m = 2^{13}, X_0 = 2^{10}, 2^5, 2$.
   Hint: The command `par(mfrow = c(2, 2))`, which produces a 2x2 array of graphs in a plot window, might be useful.

**Exercise 2 (Evaluating a uniform pseudo-random number generator)**

There exist numerous more or less advanced tests and graphical methods for checking whether a sequence $U_1, \ldots, U_n$ can be considered as effectively being iid uniform random variables. In the sequel we just consider a few simple methods.
1. In general it is hard to test if $U_1, \ldots, U_n$ are identically distributed, since we have only one realisation of each random variable $U_i$. For instance, we can plot the sample path $(1, U_1), \ldots, (n, U_n)$ and study if there appears to be any systematic fluctuation. Let $n = 1000$ and produce such plots in R using the multiplicative congruential generator with first $a = 3, m = 31, X_0 = 2$ and next $a = 65539, m = 2^{13}, X_0 = 2^{10}, 2^5, 2$.

2. Independence can be checked by plotting $X_{j+i}$ against $X_j$ for $j = 1, \ldots, n - i$ where $i$ is an integer such that $1 \leq i < n$ (usually $i$ is not too close to $n$); this is called a lag-$i$ plot, as it can be used for checking dependencies $i$ time steps back. Make lag-1 and lag-2 plots for the sequences $U_1, \ldots, U_n$ considered in the previous question. Hint: `plot(U[1:(n-i)], U[(i+1):n])`

3. To check if $U_1, \ldots, U_n$ are uniformly distributed, a histogram can be produced using the `hist`-function in R. We tried this in Exercise 1.5 above—what do you expect it should look like?

4. Another useful tool is a comparison of the theoretical distribution function with the empirical distribution function by a so-called quantile-quantile (or Q-Q) plot as defined below. Recall first that the $a$-quantile of a (generic) distribution function $F$ is defined by
\[
Q(a) = \inf \{ x \mid F(x) \geq a \}, \quad 0 \leq a \leq 1,
\]
that is the smallest real number $x$ such that $F(x) \geq a$.

a) Discuss what the $a$-quantile is for a continuous random variable and for a discrete random variable.

b) Show that in the case of the uniform distribution,
\[
Q(a) = a, \quad 0 \leq a \leq 1. \tag{3}
\]

c) The empirical distribution function based on a sequence of identically distributed random variables $X_1, \ldots, X_n$ is defined by
\[
\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} 1_{\{X_i \leq x\}}
\]
and the Q-Q plot is the graph
\[
(Q(\hat{F}_n(X_i)), X_i)_{i=1,\ldots,n}.
\]
Argue that because of (3) the points in a Q-Q plot are expected to be close to the identity line if $X_i$ is distributed in accordance with $F$. 

3
5. Implement in R a function \texttt{qqunif}, which produces a Q-Q plot based on uniform random variables \( U_1, \ldots, U_n \) on \([0,1]\).  
   Hint: \emph{For simplicity assume all the} \( U_i \) \emph{are different and use the R functions} \texttt{qunif and sort}.

6. Use \texttt{qqunif(runif(1000))} (we study the function \texttt{runif} in more detail in Exercise 3 below) a number of times to get an idea about what you expect the Q-Q plot should look like when you consider 1000 iid uniformly distributed numbers on the interval \([0,1]\). Compare with Q-Q plots obtained for the sequences \( U_1, \ldots, U_n \) considered above.  
   Hint: \emph{The command} \texttt{abline(0,1)} \emph{superimposes the identity line}.

7. What are the R-functions \texttt{qqnorm} and \texttt{rnorm} doing?

8. Try the command \texttt{qqnorm(rnorm(100))} a number of times and discuss the results.

**Exercise 3 (Uniform pseudo-random number generators in R)**

R uses as default a so-called twisted tausworth generator, which applies by the command \texttt{RNGkind()}. This and other uniform pseudo-random number generators in R are described by the help page for the function \texttt{Random.seed}, where it is also described how the value of the seed can be fixed so that realisations of uniform pseudo-random numbers can be used more than once.

1. Discuss why it could be interesting to reuse uniform pseudo-random numbers.

2. Read the help page for \texttt{runif}. What simulates  
   a) \texttt{runif(100)} and  
   b) \texttt{runif(100,min=1,max=3)}.

3. Test the generator \texttt{runif} by the methods in Exercise 2.