Random Number Generator of STMC and simple Monte Carlo Integration

Bernd Berg

FSU MCMC Course, August 26, September 2, 2008

Random Numbers and Fortran Code

According to Marsaglia and collaborators a list of desirable properties for random number generators is:

(i) *Randomness*. The generator should pass stringent tests for randomness.

(ii) *Long period*.

(iii) *Computational efficiency*.

(iv) *Repeatability*. Initial conditions (seed values) completely determine the resulting sequence of random variables.

(v) *Portability*. Identical sequences of random variables may be produced on a wide variety of computers (for given seed values).

(vi) *Homogeneity*. All subsets of bits of the numbers are random.

Physicists have added a number of their applications as new tests. In particular the exact solution of the $2d$ Ising model is used.

Frequently used is the congruential random number generator. It is based on the recursion

$$I_n = (a\, I_{n-1} + b) \bmod (m) \tag{1}$$

where I_n, a, b and m are integers. Uniformly distributed random numbers are then defined by the real numbers

$$x_n = I_n/m\ . \tag{2}$$

Theorems (see the book by Knuth) state that good choices of the integer constants a, b and m exists, so that the series of random numbers becomes a permutation

$$\pi_0,\ \pi_1,\ \ldots\ \pi_{m-2},\ \pi_{m-1} \tag{3}$$

of $0,\ 1,\ \ldots\ m-2,\ m-1$.

In STMC the random number generator by Marsaglia and collaborators is provided. It has a period 2^{144} and fulfills also the other desirable properties well. It relies on a combination of two generators:

x_n from a lagged Fibonacci series $I_n = I_{n-r} - I_{n-s} \bmod 2^{24}$, $r = 97$, $s = 33$.

y_n from the arithmetic series $I - k$, $I - 2k$, $I - 3k$, $\ldots$, $\bmod [2^{24} - 3]$.

For most applications this generator is a good compromise. Our Fortran code which implements Marsaglia random numbers consists of three subroutines:

`rmaset.f` to set the initial state of the random number generator.

`ranmar.f` which provides one random number per call.

`rmasave.f` to save the final state of the generator.

The subroutine `rmaset.f` continues a saved state or initializes the generator to independent sequences of random numbers defined by distinct pairs of seeds:

$$-1801 \leq \texttt{iseed1} \leq 29527 \quad \text{and} \quad -9373 \leq \texttt{iseed2} \leq 20708 \; . \tag{4}$$

This property makes the generator quite useful for parallel processing.

Table 1: Illustration of a start and a continuations run of the Marsaglia random number generator using the program `mar.f` with the default seeds (a0102_02).

```
RANMAR INITIALIZED.
idat, xr = 1  0.116391063
idat, xr = 2  0.96484679
idat, xr = 3  0.882970393
idat, xr = 4  0.420486867
extra xr =    0.495856345
```

```
MARSAGLIA CONTINUATION.
idat, xr = 1  0.495856345
idat, xr = 2  0.577386141
idat, xr = 3  0.942340136
idat, xr = 4  0.243162394
extra xr =    0.550126791
```

How to get and run the FORTRAN code?

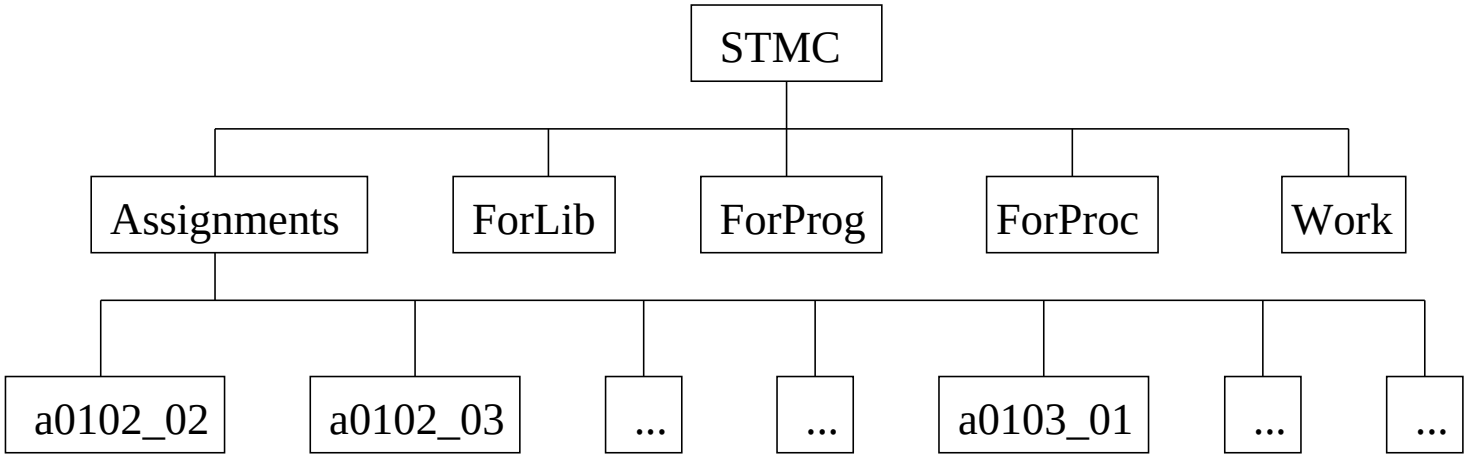

Figure 1: The Fortran routines are provided and prepared to run in the a tree structure of folders depicted in this figure. This tree of directories unfolds from the downloaded file.

To **download** the Fortran code book visit the book website and follow the instructions given there.

The code is provided in the directories `ForLib`, `ForProg` and `ForProc`. `ForLib` contains a library of functions and subroutines which is closed in the sense that no reference to non-standard functions or subroutines outside the library is ever made. Fortran programs are contained in the folder `ForProg` and procedures for interactive use in `ForProc`.

Assignment: Marsaglia random numbers. See coursework website. Understand how to re-start the random number generator as well as how to perform different starts when the continuation data file `ranmar.d` does not exist.

Note: To compile properly, main programs have to be located two levels down from a root directory `STMC`.

The hyperstructure of program dependencies introduced between the levels of the STMC directory tree should be kept intact!

(Unless you really know better.)

Simple Monte Carlo (MC) Integration

Uniformly distributed random number allow one to evaluate integrals of the form

$$\int_{\text{Volume}} d^d x \, f(x_1, \ldots, x_d)$$

where the Volume is defined by constraints. The method is very general, but only efficient as long as the volume constraints and the function f are sufficiently well-behaved (regular).

Examples (classwork and homework):

1. Calculation of π from

$$\pi = \int_{x^2+y^2<1} dx \, dy \ .$$

2. Calculation of the integrals

$$\int_{x^2<\sin(y),y^2<cos(x)} dx\,dy$$

and

$$\int_{x^2<\sin(y),y^2<cos(x)} dx\,dy\,e^r \quad \text{with} \quad r=\sqrt{x^2+y^2}\ .$$