The Monte Carlo method is a method that is used to generate elements from a given probability distribution.

We use it to compute integrals.

**Example.** Suppose we wish to compute

\[ I = \int_0^L dx f(x) \]

The MC strategy: rewrite I as an average over a probability distribution.

Define: \( \pi(x) = \frac{1}{L} \) in \([0, L]\)

\( \pi(x) \) is a probability density in \([0, L]\)

(i) \( \pi(x) \geq 0 \)

(ii) \( \int_0^L dx \pi(x) = 1 \)

Then, we rewrite

\[ I = \int_0^L dx f(x) = L \int_0^L \frac{dx}{L} f(x) = L \int_0^L \pi(x) f(x) = L \left< f \right>_\pi \]

The MC method is used to compute \( \left< f \right>_\pi \).
Algorithm to compute \( \langle f \rangle_\pi \)  

Input: \( N \) = number of iterations  

(pseudo code) 

\[
\text{Sum} = 0. \\
\text{REPEAT} \; N \; \text{TIMES} \\
\quad \text{Generate } X \text{ with probability } \pi(x) \\
\quad \text{Sum} = \text{Sum} + f(X) \\
\text{END \ REPEAT} \\
\langle f \rangle_\pi = \frac{1}{N} \; \text{Sum}
\]

The reason why this algorithm works is 
the sample-mean theorem 

\[
\frac{1}{N} \sum_{i=1}^{N} f(X_i) \rightarrow \langle f \rangle_\pi \quad \text{for } N \rightarrow \infty \\
\text{extracted with probability (density) } \pi(x)
\]

THIS RESULT IS TRUE for any PROBABILITY \( \pi(x) \) 

The practical problem: generate numbers \( X_i \) 
with probability \( \pi(x) \) 

Monte Carlo algorithms are meant to solve 
this problem.
THE BASIC ROUTINE

All MC programs include a routine (we call it \( \text{RAN}(\cdot) \)) that generates uniform random numbers in \([0, 1]\).

\( \text{RAN}(\cdot) \) generates numbers \( X_i \) such that

\[
\text{prob}(a < X_i < a + \delta) = \delta \quad \text{with} \quad 0 \leq a < a + \delta \leq 1
\]

The second (VERY IMPORTANT) property of \( \text{RAN}(\cdot) \):

There are NO CORRELATIONS between different random numbers \( X_i, X_j \) \((i \neq j)\)

\[
\text{prob}(a < X_i < a + \delta \text{ AND } b < X_j < b + \delta) = \delta^2
\]

This routine can be used to generate numbers \( X_i \) uniformly distributed in \([0, L]\)

\[ X_i = L \times \text{RAN}(\cdot) \]

This routine can be used to generate an INTEGER number uniformly in \(0, \ldots, L\).

\[ X_i = \text{floor} \left[ \frac{U_i}{L+1} \times \text{RAN}(\cdot) \right] \]

\[ \text{prob } U_i = 1/4 \]

\[ 0 \leq U_i < 4 \]

\[ L = 3 \]

\[ \begin{array}{cccc}
0 & 1 & 2 & 3 & 4 \\
X_i = 0 & X_i = 1 & X_i = 2 & X_i = 3 & \text{all with probability } \frac{1}{4}
\end{array} \]
AN IMPORTANT COMMENT

Most random number generators (RNG) generate uniform random numbers in \([0, 1[\), NOT \([0, 1]\)

\(\text{RAN}( ) = 1\) never occurs

This is important in many contexts.

If \(\text{RAN}( ) = 1\) is generated the algorithm to generate an integer DOES NOT WORK

\[ \text{RAN}( ) = 1 \quad \rightarrow \quad U_i = (L+1) \quad \rightarrow \quad X_i = \text{floor}(L+1) = L+1 \]

\(X_i\) does not belong to 0...L

(The program may crash)