1. Monte Carlo Simulations of Radioactive Decay

Monte Carlo methods

Monte Carlo methods are widely used in the Sciences, from the integration of multi-dimensional integrals to solving many other problems in chemistry, physics, medicine, biology, or even Dow-Jones forecasting and the traffic flows.

Computational finance is one of the novel fields where Monte Carlo methods have found a new field of applications, with financial engineering as an emerging field. Emerging fields like econophysics are new examples of applications of Monte Carlo methods.

Each field using Monte Carlo methods may apply them in different ways, but in essence *they are using random numbers to examine some problem and approximate its outcome.* As such Monte Carlo methods give us a way to model complex systems that are often extremely hard to investigate with other types of techniques.

What is meant by Monte Carlo Method?

Numerical methods that are known as Monte Carlo methods can be loosely described as statistical simulation methods, where statistical simulation is defined in quite general terms to be any method that utilizes sequences of random numbers to perform the simulation.

The term "Monte Carlo Method" is very general. It is a stochastic technique, meaning that it is based on using random numbers and the statistics of probability to investigate wide range of problems.

Not surprisingly, the term was coined after the casino in the principality of Monte Carlo.Every game in a casino is a game of chance relying on random events such as ball falling into a particular slot on a roulette wheel, being dealt useful cards from a randomly shuffled deck, or the dice falling the right way.

Brief History

The idea of the Monte Carlo(MC) method is a lot older than the computer. The name Monte Carlo is relatively new- it was introduced by von Neumann and Ulam during World War II, as a code word for the secret work at Los Alamos; it was suggested by the gambling casinos at the city of Monte Carlo in Monaco, because of the similarity of statistical simulation to games of chance (Monaco was the center of gambling). The Monte Carlo method was then applied to problems related to the atomic bomb. The idea of using randomness for calculations occurred to Stan Ulam while he was playing a game of cards. He realized that he could calculate the probability of a certain event simply by repeating the game over and over again. From there it was a simple step to realize that the computer could play the games for him. Under the name statistical sampling, the MC method stretches back to times when numerical calculations were performed using pencil and paper. At first, Monte Carlo was a method for estimating integrals which could not be solved by other means. Integrals over poorly-behaved functions and multidimensional integrals were profitable subjects of the MC method. The famous physicist Richard Feynman realized around the time of the Second World War that the time of electronic computing was just around the corner. He created what could be described as a highly pipelined human CPU, by employing a large number of people to use mechanical adding machines in an arithmetic assembly line. A number of crucial calculations to the design of the atomic bomb were performed in this way. It was in the last months of the Second World War when the new ENIAC electronic computer was used for the first time to perform numerical calculations. The technology that went into ENIAC had existed even before but the war had slowed down the construction of the machine. This seems obvious now, but it is actually a subtle question that a physical problem with an exact answer can be approximately solved by studying a suitable random process. Nowadays Monte Carlo has grown to become the most powerful method for solving problems in statistical physics - among many other applications.

The Monte Carlo method took off as a computational tool when computers became available and radical improvements over the "basic" algorithm were found, and the modified algorithm we still use.

So, conclusively in principle, any method that uses random numbers to examine some problem is a Monte Carlo method. The name Monte Carlo is used as a general term for a wide class of stochastic methods. The common factor is that random numbers are used for sampling.

Random Variable

Random numbers as we use them here are numerical approximations to the statistical concept of stochastic variables, sometimes just called random variables. The example we choose is the classic one, the tossing of two dice, its outcome and the corresponding probability. In principle, we could imagine being able to determine exactly the motion of the two dice, and with given initial conditions determine the outcome of the tossing. Alas, we are not capable of pursuing this ideal scheme. However, it does not mean that we do not have certain knowledge of the outcome. This partial knowledge is given by the probability of obtaining a certain number when tossing the dice. To be more precise, the tossing of the dice yields the following possible values

[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

These values are called the *domain*. To this domain we have the corresponding probabilities

[1/36, 2/36, 3/36, 4/36, 5/36, 6/36, 5/36, 4/36, 3/36, 2/36, 1/36].

The numbers in the domain are the outcomes of the physical process tossing the dice.*We cannot tell beforehand whether the outcome is 3 or 5 or any other number in this domain. This defines the randomness of the outcome, or unexpectedness or any other synonymous word which encompasses the uncertitude of the final outcome.* The only thing we can tell beforehand is that say the outcome 2 has a certain probability.
If our favorite hobby is to spend an hour every evening throwing dice and registering the sequence of outcomes, we will note that the numbers in the above domain

[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],

appear in a random order. After 11 throws the results may look like

[10, 8, 6, 3, 6, 9, 11, 8, 12, 4, 5].

Eleven new attempts may results in a totally different sequence of numbers and so forth. Repeating this exercise the next evening, will most likely never give you the same sequences. Thus, we say that the outcome of this hobby of ours is truly random.
Random variables are hence characterized by a domain which contains all possible values that the random value may take. This domain has a corresponding PDF.

We can have another example, consider the radioactive decay of an α-particle from a certain nucleus. Assume that you have at your disposal a Geiger-counter which registers every 10 ms whether an α-particle reaches the counter or not. If we record a hit as 1 and no observation as 0, and repeat this experiment for a long time, the outcome of the experiment is also truly random. We cannot form a specific pattern from the above observations. The only possibility to say something about the outcome is given by the PDF, which in this case is the well-known exponential function $\lambda e^{-\lambda x}$, with λ being proportional to the half-life of the given nucleus which decays.

Random Number Generators

A reliable random number generator is critical for the success of a Monte Carlo program. This is particularly important for Monte Carlo simulations which typically involve the use of literally millions of random numbers. If the numbers used are poorly chosen, i.e. if they show non-random behavior over a relatively short interval, the integrity of the Monte Carlo method is severely compromised.

In real life, it is very easy to generate truly random numbers. The lottery machine does this every Saturday night! Indeed, something as simple as drawing numbers of a hat is an excellent way of obtains numbers that are truly random.

In contrast, it is impossible to conceive of an algorithm that results in purely random numbers because by definition an algorithm, and the computer that executes it, is deterministic. That is it is based on well-defined, reproducible concepts.

Pseudo-Random Numbers

A computer is a precise and deterministic machine, and hence, the output of a computer is predictable and not truly *"random".* The word *"random"* represents the output of a truly random physical process such as the decay of radioactive nuclei. In radioactive decay process, we cannot predict which nucleus is going to decay and when. However, we can say that after time t, how many nuclei would have decayed using probabilistic models. Many books refer to computer generated sequences as *"pseudo-random"* numbers.

Numbers generated by any of the standard algorithms are in reality pseudo random numbers, hopefully abiding to the following criteria:

1. They produce a uniform distribution in the interval [0,1],
2. Correlations between random numbers are negligible,
3. Any number in the range is equally likely to be picked,
4. The period before the same sequence of random numbers is repeated is as large as possible and finally,
5. The algorithm should be fast.

Mathematically, for a uniform random number, z, the probability distribution function u(z) is given by

$$u(z) = \begin{cases} \dfrac{1}{b-a}, & a \le z \le b \\ 0, & otherwise \end{cases}$$

A disclaimer again, is however appropriate. It should be fairly obvious that something as deterministic as a computer cannot generate purely random numbers. So it can be said that there is an obvious drawback of MC methods:

There is a statistical error. Sometimes there is a tradeoff between statistical error and systematic error and one needs to find the best compromise.

Radioactive Decay

Radioactive decay is actually a discrete process with each atom having some finite probability of decay. A more physically accurate approach and intuitively accessible approach to radioactive decay is the *Monte Carlo approach*. We will take the approach of testing each atom individually to see if it decays. For a large number of atoms, this approach must be implemented computationally. One of the simplest examples of a physical process for which the Monte Carlo method can be applied is the study of radioactive decay. Here we begin with a sample of N nuclei which decay at a rate λ s^{-1}.The physics of the situation specifies that the rate of decay is given by

$$\frac{dN}{dt} = -\lambda N$$

where the nuclei which decay during the time *dt* can be chosen randomly. Note that this is an example of a physical process which is *random* by nature. The Monte Carlo method is especially well suited for studies of such processes.

The time dependence of the number of undecayed nuclei specifies that the rate of decay is given by

$$N = N_0 e^{-\lambda N}$$

Where N_0 is the initial number of nuclei and λ is the *decay constant* in units of probability per unit time for each nucleus of a given species. For a given nucleus, the probability *P* that a nucleus undergoes radioactive decay in time dt is,

$$P = \lambda dt$$

In the simplest approach, the positions of the nuclei play no role and only the number of undecayed nuclei is monitored. Time is divided into discrete intervals, and each undecayed nucleus is tested for decay. Time is then incremented by one unit and the process is repeated so that the number of undecayed nuclei can be determined as a function of time.

The time discretization must be done intelligently so that a reasonable number of decays occur in each time step. Otherwise the simulation requires too much CPU time to be effective. On the other hand, if the time step is chosen to be too large, then time resolution is lost because too many decays occur during each time step.

The entire process can be repeated many times to obtain a series of independent *measurements,* and the mean value of *N* as well as an error estimate may be determined for each value of time. Note that in this case the measurements are uncorrelated since we begin a new simulation with a different random number sequence each time. The extension of this approach to multiple decay paths is straightforward.

In other words if we know the initial number of nuclei *N*, we can calculate randomly for each nuclei if that nuclei decays or not for some small dt. If it does, we remove it from the sample and recalculate for the next small interval dt. This approach lends itself nicely to a computational implementation.

In this limit we get exponential decay - which we thus see is a good description *on the average* for a large number of particles, but clearly become very inaccurate as the number of

particles gets small. This decay is *spontaneous*, that is, it occurs without (and independent of) external stimulation. While the probability of decay decreases in time as the number of particles decrease, the actual decay is a random process in which the rate for any time period fluctuates, which leads to the typical sound of a *Geiger counter*.

Numerical Approach

In a given time step, each nucleus is given the opportunity to decay with probability *p*. So for a given nucleus, we choose a random number in the range 0 to 1. If the number is less than the probability *p*, then it decays. Here decay simply means that the total number of nuclei is decrease by one. Note that this happens every time step; see the pseudo code below.

Algorithm

1. Input initial number of parent nuclei N_0, and daughter nuclei N_d,
2. Input decay constant λ,
3. Input the number of time steps n,
4. Input time step dt,
5. Loop: t from 0 to n,
6. Loop: i from 0 to N_0 ,
7. Generate a random number R,
8. If R< $\lambda * dt$ then $N_0 = N_0 - 1$ & $N_d = N_d + 1$
9. In case daughter nuclei also decay add steps 5 to 8 for N_d,
10. Output N_0, N_d and N_{gd} (in case daughter nuclei also decay) to a file,
11. Plot N_0, N_d and N_{gd}, each vs time for all t.