Monte Carlo Simulation: A Quick Overview

Monte Carlo simulation uses game theory to estimate the behavior of situations which do not lend themselves to direct computational solutions such as when:

- The associated equations of the model are not directly solvable
- The associated equations of the model are not fully known
- The burden of computation necessary to calculate the results of all possible inputs to the equations of the is beyond the scope of current computer machinery

For more information see:

http://www.google.com/search?sourceid=navclient&ie=UTF-8&rlz=1T4GGIH_enUS286US286&q=What+Is+Monte+Carlo+352740760X_c01.pdf

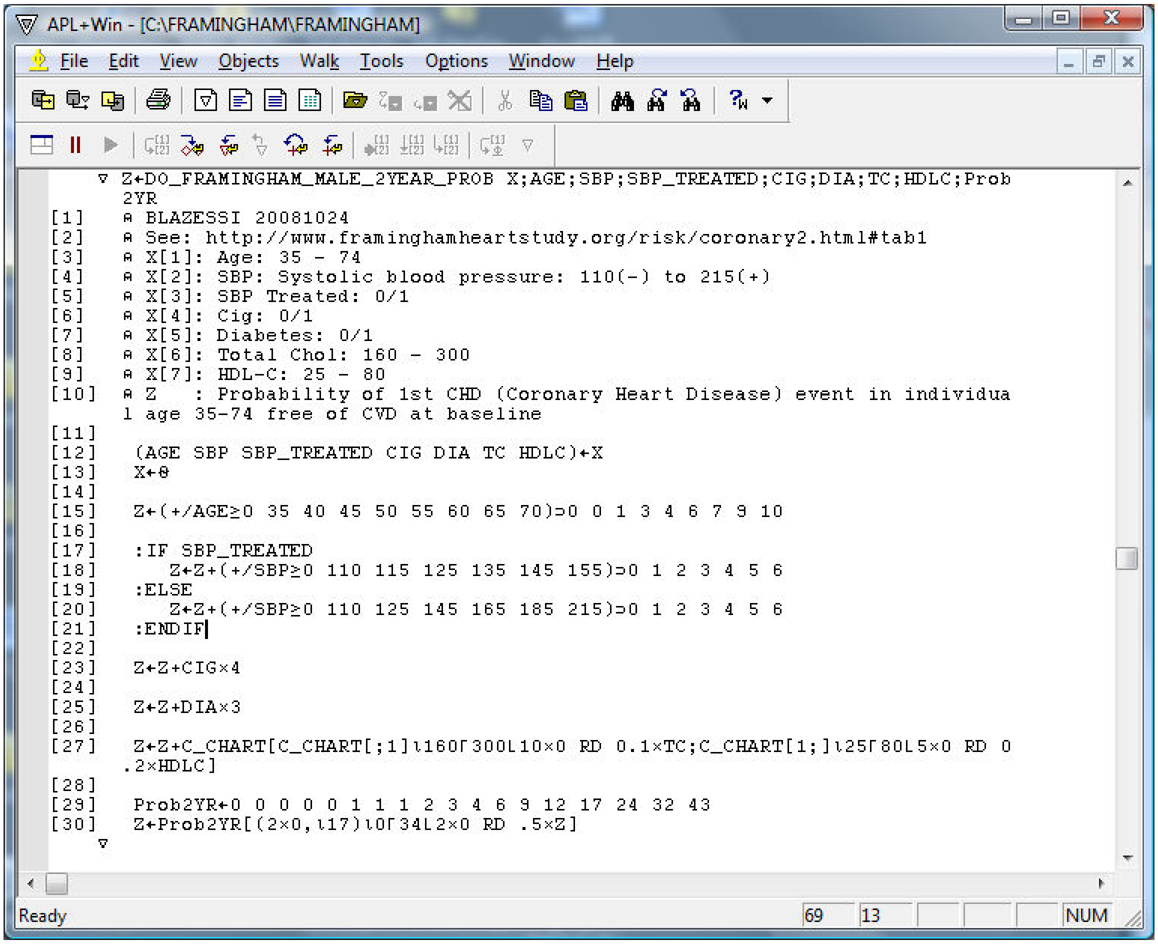
http://www.math.columbia.edu/~ik/FernKarSPT.pdf

An essential step in Monte Carlo Simulation is to establish the model:

- Results in an expectation value
- Depends on one or more random variables

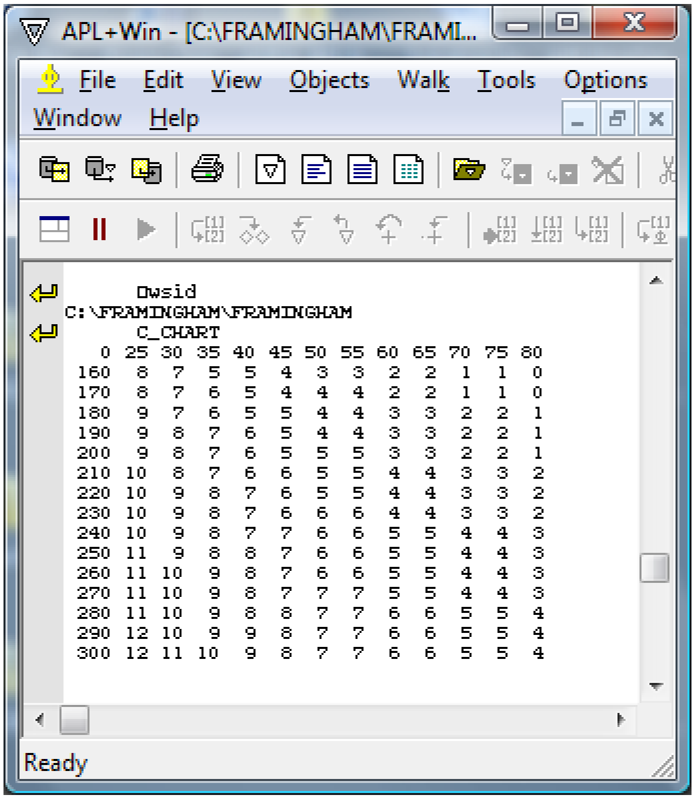
For example the Framingham heart disease study has resulted in several models of the expected occurrence of an heart disease incident within a given time period. For this illustration we use the Framingham model males having a CHD event in the two year period after inception of the evaluation.

The model is composed of an APL function:



```
    ∇ Z←DO_FRAMINGHAM_MALE_2YEAR_PROB X;AGE;SBP;SBP_TREATED;CIG;DIA;TC;HDLC;Prob
      2YR
      ⍝ BLAZESSI 20081024
      ⍝ See: http://www.framinghamheartstudy.org/risk/coronary2.html#tab1
      ⍝ X[1]: Age: 35 - 74
      ⍝ X[2]: SBP: Systolic blood pressure: 110(-) to 215(+)
      ⍝ X[3]: SBP Treated: 0/1
      ⍝ X[4]: Cig: 0/1
      ⍝ X[5]: Diabetes: 0/1
      ⍝ X[6]: Total Chol: 160 - 300
      ⍝ X[7]: HDL-C: 25 - 80
      ⍝ Z   : Probability of 1st CHD (Coronary Heart Disease) event in individua
      l age 35-74 free of CVD at baseline

       (AGE SBP SBP_TREATED CIG DIA TC HDLC)←X
       X←⍬

       Z←(+/AGE≥0 35 40 45 50 55 60 65 70)⊃0 0 1 3 4 6 7 9 10

       :IF SBP_TREATED
          Z←Z+(+/SBP≥0 110 115 125 135 145 155)⊃0 1 2 3 4 5 6
       :ELSE
          Z←Z+(+/SBP≥0 110 125 145 165 185 215)⊃0 1 2 3 4 5 6
       :ENDIF

       Z←Z+CIG×4

       Z←Z+DIA×3

       Z←Z+C_CHART[C_CHART[;1]⍳160⌈300⌊10×0 RD 0.1×TC;C_CHART[1;]⍳25⌈80⌊5×0 RD 0
      .2×HDLC]

       Prob2YR←0 0 0 0 0 1 1 1 2 3 4 6 9 12 17 24 32 43
       Z←Prob2YR[(2×0,⍳17)⍳0⌈34⌊2×0 RD .5×Z]
    ∇
```

The model also includes a reference table based on the Framingham study:


APL+Win - [C:\FRAMINGHAM\FRAMI...
File Edit View Objects Walk Tools Options
Window Help
Ready

```
      ⎕wsid
C:\FRAMINGHAM\FRAMINGHAM
      C_CHART
    0  25  30  35  40  45  50  55  60  65  70  75  80
  160   8   7   5   5   4   3   3   2   2   1   1   0
  170   8   7   6   5   4   4   4   2   2   1   1   0
  180   9   7   6   5   5   4   4   3   3   2   2   1
  190   9   8   7   6   5   4   4   3   3   2   2   1
  200   9   8   7   6   5   5   5   3   3   2   2   1
  210  10   8   7   6   6   5   5   4   4   3   3   2
  220  10   9   8   7   6   5   5   4   4   3   3   2
  230  10   9   8   7   6   6   6   4   4   3   3   2
  240  10   9   8   7   7   6   6   5   5   4   4   3
  250  11   9   8   8   7   6   6   5   5   4   4   3
  260  11  10   9   8   7   6   6   5   5   4   4   3
  270  11  10   9   8   7   7   7   5   5   4   4   3
  280  11  10   9   8   8   7   7   6   6   5   5   4
  290  12  10   9   9   8   7   7   6   6   5   5   4
  300  12  11  10   9   8   7   7   6   6   5   5   4
```

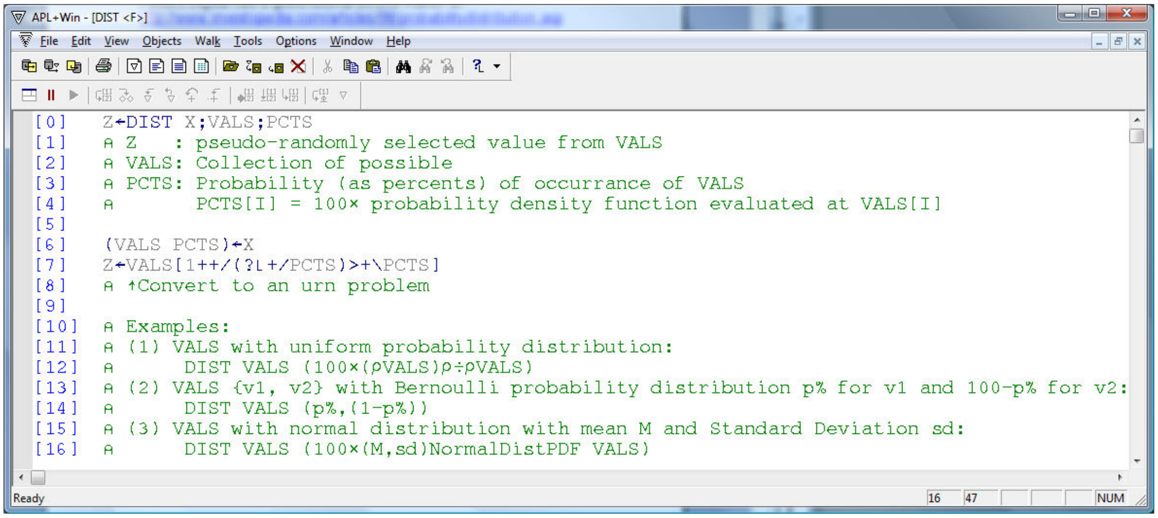
For more information about the Framingham Heart Study see:
http://www.framinghamheartstudy.org/

The assumed distribution(s) of the random variables in the model need to be selected. This is often more art than science and several potential distribution options may need to be examined before a final selection is made.

Forbes Digital has a good tutorial on this matter at:
http://www.investopedia.com/articles/06/probabilitydistribution.asp

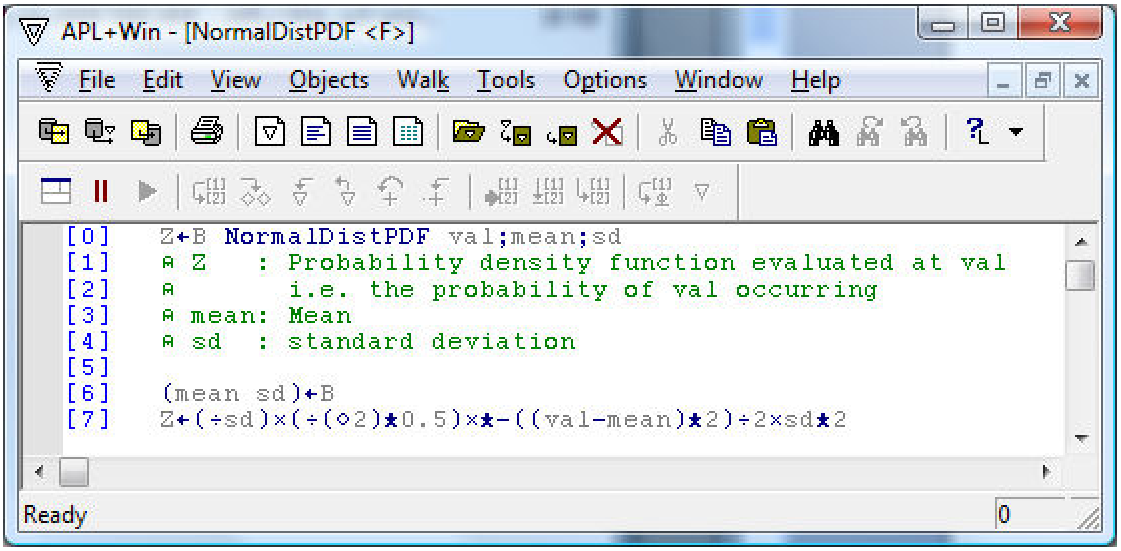
Because digital computation is used to perform the Monte Carlo simulation using a finite set of possible values for each of the inputs, the probability density functions of the distributions of the inputs are used to obtain the probability (as %) of the occurrence of each of the input values.

In this example the DIST function accommodates any distribution by specifying the possible values of the random variables and the associated probabilities of each value (as a percent). The function then selects one of the possible values based upon these associated probabilities. The function uses the uniform distribution (via the APL primitive ? operator) to select values from a theoretical combinatorial ‘urn’ where the number of ‘balls’ with value v (in VALS) is based upon the associated probability density function (as a percent) (in PCTS).



```
Z←DIST X;VALS;PCTS
⍝ Z    : pseudo-randomly selected value from VALS
⍝ VALS: Collection of possible
⍝ PCTS: Probability (as percents) of occurrance of VALS
⍝        PCTS[I] = 100× probability density function evaluated at VALS[I]

(VALS PCTS)←X
Z←VALS[1++/(?⌊+/PCTS)>+\PCTS]
⍝ ↑Convert to an urn problem

⍝ Examples:
⍝ (1) VALS with uniform probability distribution:
⍝       DIST VALS (100×(⍴VALS)⍴÷⍴VALS)
⍝ (2) VALS {v1, v2} with Bernoulli probability distribution p% for v1 and 100-p% for v2:
⍝       DIST VALS (p%,(1-p%))
⍝ (3) VALS with normal distribution with mean M and Standard Deviation sd:
⍝       DIST VALS (100×(M,sd)NormalDistPDF VALS)
```

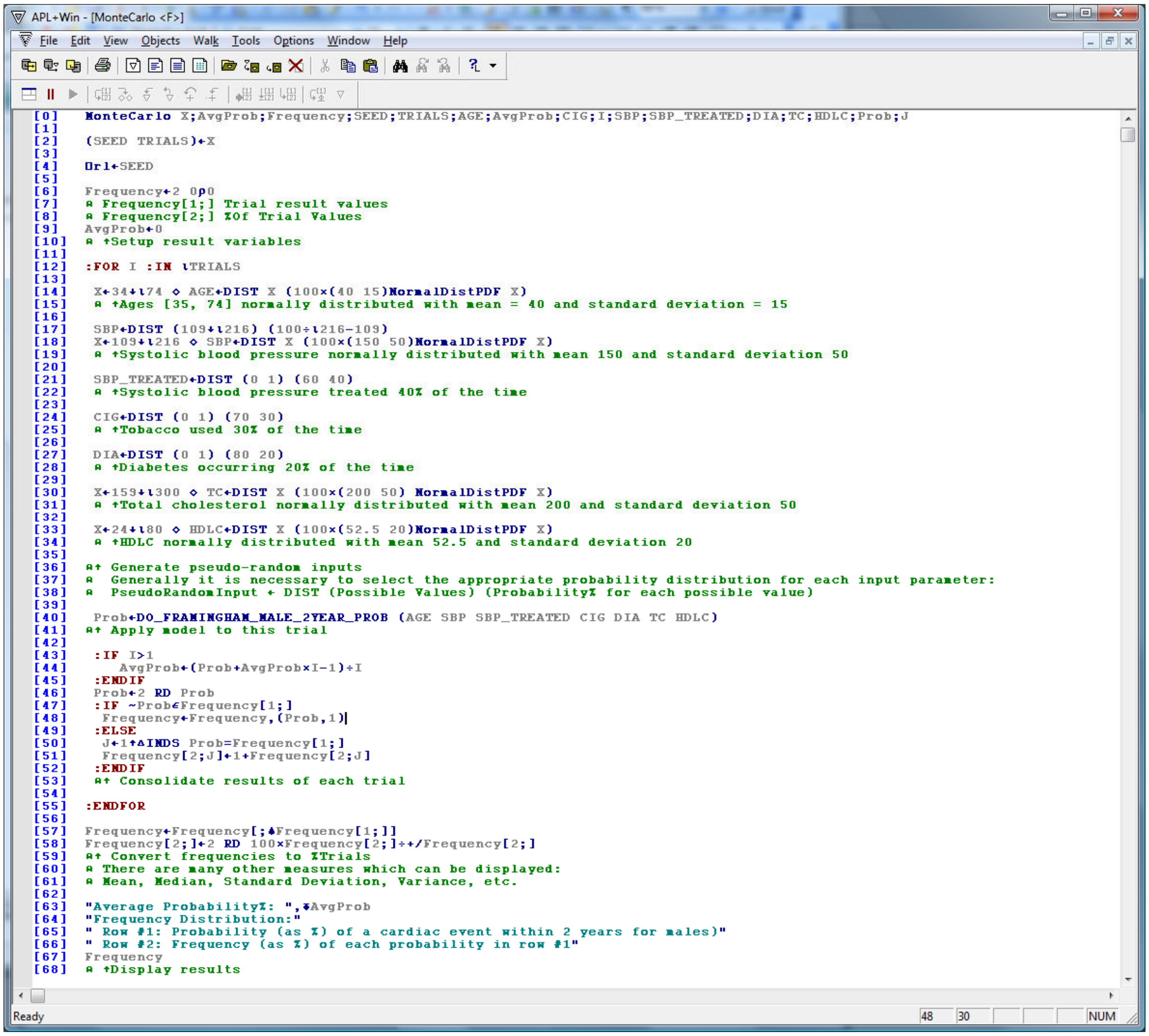
For example the probability density function of the ‘normal’ distribution can be computed using the following APL function:



```
Z←B NormalDistPDF val;mean;sd
⍝ Z   : Probability density function evaluated at val
⍝       i.e. the probability of val occurring
⍝ mean: Mean
⍝ sd  : standard deviation

(mean sd)←B
Z←(÷sd)×(÷(○2)*0.5)×*-((val-mean)*2)÷2×sd*2
```

The simulation control function, “MonteCarlo”, in this example:

- Sequentially generates the pseudo random input values based on the assumed distributions of the inputs to the model
- Calls the model function for each of these trials using the generated input values
- Receives the expected value from each of the model trials
- Accumulates the expected values in a result data set
- Computes the desired measures associated with the result data set. In this case:
    - The average probability of a cardiac event within 2 years
    - The frequency distribution of the probabilities of cardiac event within two years (for males)



```
MonteCarlo X;AvgProb;Frequency;SEED;TRIALS;AGE;AvgProb;CIG;I;SBP;SBP_TREATED;DIA;TC;HDLC;Prob;J

(SEED TRIALS)←X

⎕rl←SEED

Frequency←2 0⍴0
⍝ Frequency[1;] Trial result values
⍝ Frequency[2;] %Of Trial Values
AvgProb←0
⍝ ↑Setup result variables

:FOR I :IN ⍳TRIALS

 X←34+⍳74 ⋄ AGE←DIST X (100×(40 15)NormalDistPDF X)
 ⍝ ↑Ages [35, 74] normally distributed with mean = 40 and standard deviation = 15

 SBP←DIST (109+⍳216) (100÷⍳216-109)
 X←109+⍳216 ⋄ SBP←DIST X (100×(150 50)NormalDistPDF X)
 ⍝ ↑Systolic blood pressure normally distributed with mean 150 and standard deviation 50

 SBP_TREATED←DIST (0 1) (60 40)
 ⍝ ↑Systolic blood pressure treated 40% of the time

 CIG←DIST (0 1) (70 30)
 ⍝ ↑Tobacco used 30% of the time

 DIA←DIST (0 1) (80 20)
 ⍝ ↑Diabetes occurring 20% of the time

 X←159+⍳300 ⋄ TC←DIST X (100×(200 50) NormalDistPDF X)
 ⍝ ↑Total cholesterol normally distributed with mean 200 and standard deviation 50

 X←24+⍳80 ⋄ HDLC←DIST X (100×(52.5 20)NormalDistPDF X)
 ⍝ ↑HDLC normally distributed with mean 52.5 and standard deviation 20

⍝↑ Generate pseudo-random inputs
⍝  Generally it is necessary to select the appropriate probability distribution for each input parameter:
⍝  PseudoRandomInput ← DIST (Possible Values) (Probability% for each possible value)

 Prob←DO_FRAMINGHAM_MALE_2YEAR_PROB (AGE SBP SBP_TREATED CIG DIA TC HDLC)
⍝↑ Apply model to this trial

 :IF I>1
    AvgProb←(Prob+AvgProb×I-1)÷I
 :ENDIF
 Prob←2 RD Prob
 :IF ~Prob∊Frequency[1;]
  Frequency←Frequency,(Prob,1)
 :ELSE
  J←1↑∆INDS Prob=Frequency[1;]
  Frequency[2;J]←1+Frequency[2;J]
 :ENDIF
 ⍝↑ Consolidate results of each trial

:ENDFOR

Frequency←Frequency[;⍋Frequency[1;]]
Frequency[2;]←2 RD 100×Frequency[2;]÷+/Frequency[2;]
⍝↑ Convert frequencies to %Trials
⍝ There are many other measures which can be displayed:
⍝ Mean, Median, Standard Deviation, Variance, etc.

"Average Probability%: ",⍕AvgProb
"Frequency Distribution:"
" Row #1: Probability (as %) of a cardiac event within 2 years for males)"
" Row #2: Frequency (as %) of each probability in row #1"
Frequency
⍝ ↑Display results
```

The inputs for the model have been selected as follows:

- Ages: Range [35, 74] normally distributed with mean 40 and standard deviation 15
- Systolic blood pressure: Range [110, 216] normally distributed with mean 150 and standard deviation 50
- SBP treated for 40% of the population
- Tobacco used by 30% of population
- Diabetes occurs in 20% of the population
- Total Cholesterol: Range [160, 300] normally distributed with mean 200 and standard deviation 50
- HDLC: Range[25, 80] normally distributed with mean 52.5 and standard deviation 20

Using the above MonteCarlo function, for a certain ‘random seed’ and 100,000 trials, the simulation resulted in a ‘Frequency Distribution” as follows:

Row #1 is the Probability (as %) of a cardiac event within 2 years for males and Row #2 is the Frequency (as %) of each probability in row #1.

Row #1:	0	2	3	4	6	9	12	17	24	32
Row #2:	10.52	46	15.77	11.88	7.74	4.54	2.26	0.92	0.28	0.08

So for this example 46% of the 100,000 trials resulted in a 2% probability of a cardiac event within two years for males.

With a bit of effort the results could be displayed in a more cosmetic manner and additional measures could be generated. Typically the results are compared to the various expected frequency distributions, for example using ‘least squares’ analysis, to determine which theoretical probability distribution most closely matches the simulation results. Ultimately the model results are compared to physical trials for additional validation of the model.

Performance of Monte Carlo simulation can often be problematic. Several approaches are commonly used to improve performance:

- Employ more computing power using multi-threading and multi-processors. This technology permits the trials to be carried out asynchronously so that the generation of trials can be performed continuously without waiting for the results. In this case it is necessary for the simulation control function to receive the results by subscribing to an event which is triggered by the completion of a specific trial.

  APL+Win can employ this technology when used with APL WebServices either on a multi-processor workstation or on servers.

  VisualAPL can inherently initiate any number of independent processing threads and subscribe to events which indicate the completion of these threads.

- Move the accumulation of trial results to a mechanism which can receive the values independently of the initiation of the trial by the simulation control function. This methodology is often used in conjunction with multi-threading as described above.

  This mechanism can be as simple as having each trial save the results to a separate memory location.

  Recently the possibility of using ‘video memory’ has become available and this approach bears additional study.

- Improve the measures used to determine the sensitivity of the model to the inputs. If this sensitivity can be quantified the number of trials can be reduced by limiting the range of the random inputs to the model which represent the regions of greatest sensitivity.

  The June 22 2003 World Conference on Risk included papers on this topic which can be obtained from the authors. See http://www.ramas.com/ipbrussels.htm for details.

How many trials to perform in a Monte Carlo Simulation can sometimes be an issue. Several methods have been employed including:

- Observing the frequency distribution and standard deviation of the results and stopping the simulation when these ‘stabilize’.

- Comparing the resulting distribution to an expected distribution.

- Accumulating the results by summing the trials and assuming that the Kolmogorov Central Limit applies to the model and stopping when the disparity between the summed results and the Gaussian distribution is sufficiently small.

- Where possible create a model which can employ ‘low-discrepancy’ sequences for the inputs. In this case bounds on the ‘error’ between actual and simulated results have been established. See http://en.wikipedia.org/wiki/Low-discrepancy_sequence for more information on this method. The ‘quasi-Monte Carlo methods’ are also related to this concept and have been studied extensively by Giray Okten, see: http://www.math.fsu.edu/People/faculty.php?id=592.

The quality of pseudo-random number generation in a Monte Carlo Simulation is sometimes important. Computer-generated pseudo-random numbers are not truly random and essentially represent a finite, though quite long, sequence of fixed values.

- For example in APL+Win the ? and []rl features are based on the underlying Microsoft Win32 pseudo-random number generator. Analogously the Microsoft .Net pseudo-random number generator is the basis of the VisualAPL ? and []rl features.

- Another approach to random number generation is to prepare a large sequence of pseudo-random numbers which satisfy certain properties of non-clustering. See Le Cuyer’s methodology as an example:

http://www.iro.umontreal.ca/~lecuyer/myftp/papers/wsc01rng.pdf

- Genuine random numbers can be generated using electro-optical methods. Such devices are available commercially and depend on the measurement of physical properties for example by using:

    - A half-silvered mirror and an individual photon detector
    - Measurement of random noise observed when a photo-sensitive cell receives no light input but generates an image which is entirely noise

  For more details see:
  www.**random**.org
  http://www.idquantique.com/news/files/quantis_usb.pdf
  http://www.randomnumbers.info/content/Generating.htm

Reverse Monte Carlo Simulation has applications where the desired probability distribution of the model is known and it is desired to determine the input space which will yield that distribution. Most applications of this technique have been related to basic physics research, but it has potential to improve the design of employee benefit plans, COLI plans and other structures which are constrained by government regulations. For more information see: http://en.wikipedia.org/wiki/Reverse_Monte_Carlo and http://web.me.com/dove_family/martin/publications/EMU_2002_RMC.pdf.

At the same time (c.a. 1953) that Monte Carlo Simulation was used for practical purposes, another form of Monte Carlo Simulation was developed which has been called ‘Sequential Importance Sampling’. It has potential for simulation of situations with significant restraints, such as tax regulations, capital or resource limits, etc. The method is based upon set theory and counting. For more information see: http://www.math.uio.no/avdc/kurs/STK4050/h05/forelesninger/sis.pdf

As with any abstract model applied to ‘real world’ situations, a thorough understanding of the implicit and explicit assumptions is of utmost significance when interpreting the results of a simulation.

In addition the effect of low probability ‘outlying’ outcomes must also be considered. The consideration of these extreme outcomes is the subject of intense study. A an author who discusses this issue colloquially is

Taleb, Nassim Nicholas *The Black Swan: The Impact of the Highly Improbable*
http://www.amazon.com/s/ref=nb_ss?url=search-alias%3Dstripbooks&field-keywords=Taleb+The+Black+Swan

Please note that the material presented in this document is provided without warranty of any kind. Inadvertent omissions of proper attribution of the work of others may have also occurred.