

Some Coin-Tossing Monte Carlo Experiments*

Jeffrey D. Oldham

2000 Feb 01

In class today, we discussed using Monte Carlo simulations to mimic coin tosses.

Fix to Known Error released 2000Feb02. See Section 5.

1 Number of Heads

The first program just tossed a fair coin a given number of times and returned the number of heads seen. Although we expect fifty percent of the tosses to be heads, there is a wide variation in the actual answers, even for large number of trials.

When running the code specify the desired number of tosses on the command-line argument, e.g., `./coinToss 100` for 100 tosses.

2 Number of Crossings

A fair coin is tossed a given number of times. $+1$ is added for each tail and -1 is added for each head. The question is how many times does the cumulative sum cross from negative to positive xor vice versa? Just equalling zero does not matter.

Here is my code to simulate the coin tosses. After specifying the the number of tosses on the command line, it prints out the number of crossings. Please check the code for correctness: I have not found any errors, but there may still be some.

*©2000 Jeffrey D. Oldham (oldham@cs.trinity.edu). All rights reserved. This document may not be redistributed in any form without the express permission of the author.

3 Helper Program and Files

Monte Carlo simulation involves repeatedly running lots of trials. I wrote a short program to repeatedly run another program, printing out that program's integer output. For example, `./repeater 10 ./coinCrossings 10000` runs `./coinCrossings 10000` ten times, printing one integer for each execution.

The random number generator code (header and implementation) are needed by `coinCrossings..`

4 Compiling the Programs

An easy way to create all the programs is to copy this Makefile and type `make`.

I tested the programs on my Linux computer, which is very similar to the computer science Linux computers. I make no guarantees about compiling under Windows, but please send me indications of how my code is not portable.

5 Known Errors (Fixed 2000Feb02)

2000Feb01, Jeffrey wrote:

The command to set the random seed in `coinCrossings.cc` sometimes yields the same value for different iterations occurring close in time. I do not understand why. Suggestions for fixing the problem are solicited.

The expression to seed the random number generator in `coinCrossings.cc` and `coinToss.cc` was effectively

```
time-in-seconds ^ (process-id << 15 + process-id).
```

Since each trial requires much less than one second, just using the time in seconds will not work. Since most operating systems number each process with a identification number, the second operand is supposed to add more randomness. Process IDs are frequently only 16-bits long so we shift the process ID and then add it again. I got the operator precedence wrong. Here is the corrected expression:

```
time-in-seconds ^ ((process-id << 15) + process-id).
```