MTH 4600 Homework 1, Due Feb 13
A 3-Dimensional Statistical RNG Test

---

1. I want to get to know your computer. Please compile and execute the program `SomeTests.cpp`. It may take a few seconds to run. Take a screen shot of the output and report it to me. Please use the computer that you will generally be using to work the homework. You may report the results for more that one computer.

2. In class we saw that the Multiply-With-Carry RNG did not pass the "permutation test", indicating that eight dimensional vectors of the form $(U_1, U_2, \ldots, U_8)$ are not properly distributed. Here we do a similar test in three dimensions.

Random vectors of the form $\vec{U} = (U_1, U_2, U_3)$, where the U_i are iid uniformly distributed on the unit interval $(0, 1)$, will be uniformly distributed on the 3-dimensional unit cube:

$$\vec{U} \;\sim\; \text{Uniform}\,\{(x, y, z) : 0 < x < 1,\; 0 < y < 1,\; 0 < z < 1\}. \tag{$*$}$$

<u>*The Test*</u>. Split the unit interval into 40 disjoint subintervals of equal length $\delta = \frac{1}{40} = 0.025$ and let $I_k = (k\delta, (k+1)\delta]$ denote the k^{th} subinterval for $0 \le k < 40$. Now, for $i = 1$, 2, and 3, let $M_i = k$, where $U_i \in I_k$, so M_i indicates which of the 40 subintervals the number U_i falls in. Then M_1, M_2, and M_3 should be independent and uniformly distributed on $\{0, 1, 2, \ldots, 39\}$, so each of the $40^3 = 64000$ possibilities for (M_1, M_2, M_3) should be equally likely. Now put $M = M_1 \cdot 40^2 + M_2 \cdot 40 + M_3$, so

$$0 \;\le\; M \;\le\; 39 \cdot 40^2 + 39 \cdot 40 + 39 \;=\; 63999 \;=\; 40^3 - 1.$$

In this fashion the ordered triplets (M_1, M_2, M_3) get mapped 1-1 and onto the numbers 0 through 63999†. Assuming $(*)$, each of the 64000 possible values of M will be equally likely.

Initialize counters X_m to zero for $0 \le m \le 63999$. Generate a random vector (U_1, U_2, U_3) and compute the corresponding number M. Increment the counter X_M by 1. Repeat this n times for n very large, at least 10 million. At this point each X_m is the number of times that $M = m$ among the n simulations. If $(*)$ holds, each X_m should be Binomial$\,(n, p)$, where $p = \frac{1}{64000}$. Let $\mu = np$ and $\sigma = \sqrt{np(1-p)}$. If n is very large, the numbers $Z_m = (X_m - \mu)/\sigma$ should be approximately Normal$\,(0, 1)$ by the Central Limit Theorem.

<u>*Instructions*</u>. Execute this test first for Multiply-With-Carry (the function `MWCUniform`) and then for the Mersenne Twister (`MTUniform`). I have posted "shell" code (`Dim3NormalTest.cpp`) for you to start from. You should take n to be quite large. Try 10 million, 50 million, and then 100 million (this takes about 20 seconds on my computer). Run the 64000 Z_m statistics through the `NormalHistogram` function found in `Functions.h`. This produces the file `HistogramTxtData.txt`, which will show you the results. (If you have access to TeX software, you can "TeX" the file `Histogram.tex` to graphically view the results — show me this if you have it.) Report the results for the two random number generators with the three values of n.

† We are familiar with this in base 10. If each $M_i \in \{0, 1, 2, \ldots, 9\}$, then taking $M = M_1 \cdot 10^2 + M_2 \cdot 10 + M_3$ maps the $10^3 = 1000$ possibilities for (M_1, M_2, M_3) 1-1 and onto the numbers 0 through 999. For example, the triple $(4, 8, 3)$ gets mapped to the number 483.

3. I have included the Mersenne Twister's temper function `Temper()` in the code `Dim3NormalTest.cpp` (at the bottom). Modify the MWC RNG by replacing the line

```
return ((N + 0.5) / 4294967296.0);
```

with

```
return ((Temper(N) + 0.5) / 4294967296.0);
```

and re-do problem 2.

Please submit a hard copy of your code for problem 2, which should be generously commented. For problems 2 and 3, write a description of your observations (including the tables and/or graphs) and conclusions (does ($*$) hold?) — typewritten, and one submission per group. Submit your work as a pdf file through BlackBoard.