Math 466/666: Programming Project 1

Your work should be presented in the form of a typed report using clear and properly punctuated English. Pencil and paper calculations may be typed or hand written. Where appropriate include full program listings and output. You may work in groups of two or three. If you choose to work in a group, please turn in independently prepared reports that list the other members of your group.

1. This question explores why the mantissas of the floating-point numbers which appear in typical computations follow the reciprocal distribution.

(i) Consider the primary-school multiplication table which shows all products for the numbers 1 through 9. Make a bar-chart depicting how many of the 81 products in that table begin with the digit 1, how many with a 2 and so forth up to 9.

(ii) Write a computer program compute all products of the form

$$d_1 \cdot d_2 \cdot d_3 \qquad \text{where} \qquad d_i \in \{1, 2, \ldots, 9\}$$

and count how many begin with the digit 1, how many with a 2 and so forth up to 9. Also compute the percentages in each category. For reference, the output of your program should look like

```
#     digit      count    percent
          1        218      29.90
          2        137      18.79
          3         94      12.89
          4         81      11.11
          5         46       6.31
          6         43       5.90
          7         37       5.08
          8         37       5.08
          9         36       4.94
```

Please submit your source code—in any language—for this question.

(iii) In the text it is suggested that the mantissas of the numbers which appear in numerical calculations follow the reciprocal distribution given by the density

$$r(x) = \frac{1}{x \log b} \qquad \text{for} \qquad x \in [1/b, 1].$$

Set $b = 10$ and calculate

$$p_d = \int_{d/10}^{(d+1)/10} r(x)dx \qquad \text{for} \qquad d \in \{1, 2, \ldots, 9\}.$$

to find the probability a numbers starts with the digit d under this hypothesis.

(iv) Repeat question (ii) for products of the form

$$\pi_n = d_1 \cdots d_n \qquad \text{for} \qquad n = 4, 5, 6$$

and comment on how the percents computed for different values of n compare with the theoretical probabilities in part (iii).

(v) [Extra credit and for Math 666] Read the sections in our textbook about the reciprocal distribution. State which sections you read and then provide a theoretical explanation why the mantissas of the floating-point numbers which appear in typical computations follow the reciprocal distribution.

2. This question explores how rounding errors accumulate in a sum of numbers. To avoid the difficulties of floating-point arithmetic we consider the simpler case of fixed-point numbers of the form `X.XXXXXXX` where each `X` corresponds to a decimal digit.

(i) Suppose real numbers X_i are chosen randomly according to a uniform distribution in the interval $[0, 1]$. Let X_i^* be the result of rounding X_i to the nearest approximation of the form `X.XXXXXXX`. For definiteness, round so the last digit is even in the case of a tie. Explain why it is reasonable to assume that the resulting rounding errors

$$\varepsilon_i = X_i^* - X_i$$

will be uniformly distributed on the interval $[-0.00000005, 0.00000005]$.

(ii) Statistical simulations can be performed on a computer by specifying a seed and then using a pseudo-random number generator to create a sequence of numbers based on that seed. Let $\mathcal{S}$ be the set of seeds defined as

$$\mathcal{S} = \{\, p(i) : i = 1, \ldots, 20 \,\} \qquad \text{where} \qquad p(x) = x^3 + x + 1.$$

Write a program to compute and print all twenty seeds.

(iii) The following *C* program generates n numbers uniformly distributed on the interval $[-0.00000005, 0.00000005]$ based on the `seed` specified in line 5.

```c
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(){
4     int n=4,seed=992314;
5     srandom(seed);
6     printf("seed %d:",seed);
7     for(int i=0;i<n;i++){
8         double epsilon=0.0000001*random()/RAND_MAX-0.00000005;
9         printf(" %g",epsilon);
10    }
11    printf("\n");
12    return 0;
13 }
```

Modify the above program or write your own to print the first four numbers corresponding to each `seed` $\in \mathcal{S}$. Note that the results may depend on what language, computer and operating systems you choose to use when answering this question. Your report should include source code as well as output.

(iv) The sum of n rounding errors may be simulated by computing

$$E_n(\texttt{seed}) = \sum_{i=1}^{n} \varepsilon_i$$

where ε_i is the sequence of pseudo-random numbers corresponding to `seed` from part (iii) of this question. The root-mean-squared average

$$R_n = \left(\frac{1}{20} \sum_{\texttt{seed} \in \mathcal{S}} \left| E_n(\texttt{seed}) \right|^2 \right)^{1/2}$$

can be used to characterize the expected error after n additions. Write a program to compute R_n for $n = 2^k$ with $k = 2, \ldots, 20$. Include the source code and output.

(v) Show the accumulation of rounding error simulated by your program increases as $\sqrt{n}$ where n is the number of terms in the sum by making a log-log plot of (n, R_n) compared with the function `1e-7*sqrt(x)`.

(vi) [Extra credit and for Math 666] Read the Wikipedia article on the random walk and any other source of information which you find useful. Provide references to what you read and then give a theoretical explanation why the accumulation of rounding errors should grow as $\sqrt{n}$ where n is the number of terms in the sum.