

2411 HW 10

Duncan Wilkie

18 November 2021

1

The corresponding program appears in the Script Files. The performance of this method is at best marginally faster-converging than the Wallis product formula. With 100,000 iterations, the result is barely good to 10^{-2} .

2

The Monte Carlo estimate is

$$I \approx (b - a) \frac{1}{N} \sum_i f(x_i) = 7 \cdot 16 \left(\sqrt{3} + \sqrt{6} + \sqrt{7} + \sqrt{1} + \sqrt{5} + \sqrt{2} + 6 \cdot 10 \right) = 83.391$$

To increase the accuracy of this estimate, simply take more point samples.

3

The program appears in the Script Files.

4

This can be accomplished via the Kolmogorov-Smirnov test. The cumulative distribution function for the uniform distribution on $[0, 1]$ is

$$F(x) = \int_a^x \frac{1}{b-a} dx = \int_0^1 dx = x$$

The experimental CDF of the sample of size n can be calculated by

$$F_n(x) = \frac{\text{number of sample members less than } x}{n}$$

These are combined to compute the K-S statistic

$$D_n = \sup |F(x) - F_n(x)|$$

The null hypothesis, that different distributions are involved, is rejected with significance α when

$$\sqrt{n}D_n > K_\alpha$$

where K_α is calculated from the Kolmogorov CDF

$$P(K \leq x) = \frac{\sqrt{2\pi}}{x} \sum_{k=1}^{\infty} e^{-(2k-1)^2 \pi^2 / (8x^2)}$$

by

$$P(K \leq K_\alpha) = 1 - \alpha$$

To actually test this for the random number generators, one must choose a sample size to test. Given that sample, one must compute the $\sqrt{n}D_n$ value, and compare that to a the Kolmogorov CDF value corresponding to pre-chosen α .

In this case, we choose a sample size of 1000, and $\alpha = 0.05$. The values of $\sqrt{n}D_n$ thus found are

$$\sqrt{n}D_n = \sqrt{1000}(0.047346) = 1.4972$$

and

$$\sqrt{n}D_n = \sqrt{1000}(0.005562) = .17589$$

for `drand48()` and the custom linear congruential generator respectively. The value of K_α that gives $P(k \leq K_\alpha) = 1 - \alpha = 0.95$ is, according to the table published by Smirnov, approximately 1.36 (correct to within .0005 in $1 - \alpha$). The built-in random number generator is therefore able to reject the null hypothesis, and the custom one is not.

To determine if the values are random, we compute the entropy of the random number generators over some number of bins using the proportion of numbers in each over some number of trials as an estimate of the probability, and compare it to the theoretical entropy of a truly random one. If it is within some tolerance, say 10% of the expected value, we say it is random. We take 300 bins evenly spaced between zero and one. Given a uniformly distributed random variable over $[0, 1]$, the probability that a given bin is chosen is $1/300$. The entropy is then

$$H(X) = - \sum_{i=1}^{300} \frac{1}{300} \ln(1/300) = -\ln(300) = 5.704$$

Computing this sum for the output of `drand48()` and the custom linear congruential generator over 10000 trials, one obtains

$$H(X_1) = 5.688$$

and

$$H(X_2) = 5.5235$$

respectively. `drand48()` is clearly closer, and is within 10% of the theoretical value, so it is random. On the other hand, the custom generator is not.

5 Script Files

5.1

```
Script started on Fri 19 Nov 2021 04:45:01 PM CST
tput: unknown terminal "st-256color"
tcsh: No entry for terminal type "st-256color"
tcsh: using dumb terminal settings.
[dwilk14@tigers ~/HW10]$ cat dwilk14_hw10p1.cpp
#include <cstdlib>
#include <fstream>

using namespace std;

double piMC(int n) {
    int inside = 0;
    for (int i = 0; i < n; i++) {
        double x = 2 * drand48() - 1, y = 2 * drand48() - 1;
        if (x*x + y*y <= 1)
            ++inside;
    }

    return 4 * (double)inside / (double)n;
}

int main() {
    ofstream out;
    out.open("output1.txt");
    srand48(12);

    out << "n " << "approx pi" << endl;
    out << "100 " << piMC(100) << endl;
    out << "10000 " << piMC(10000) << endl;
    out << "100000 " << piMC(100000) << endl;

    return 0;
}
[dwilk14@tigers ~/HW10]$ g++ dwilk14_hw10p1.cpp -o dwilk14_hw10p1
[dwilk14@tigers ~/HW10]$ ./dwerk14_hw10p1
[dwilk14@tigers ~/HW10]$ cp dwilk14_hw10p1.txt /home3/kristina/phys2411/.
[dwilk14@tigers ~/HW10]$ exit
exit
```

Script done on Fri 19 Nov 2021 04:45:38 PM CST

5.2

Script started on Fri 19 Nov 2021 04:45:48 PM CST

tput: unknown terminal "st-256color"

tcsh: No entry for terminal type "st-256color"

tcsh: using dumb terminal settings.

[dwilk14@tigers ~/HW10]\$ cat dwilk14_hw10p2.cpp

```
#include <fstream>
```

```
#include <cstdlib>
```

```
#include <cmath>
```

```
using namespace std;
```

```
double moment48(int n, int k) {  
    double sum = 0;  
    for (int i = 0; i < n; i++) {  
        sum += pow(drand48(), k);  
    }  
  
    return sum / n;  
}
```

```
double momentCustom(int n, int k, int last) {  
    double sum = 0;  
    for (int i = 0; i < n; i++) {  
        last = (57 * last + 1) % 256;  
        sum += pow((double)last / 256., k);  
    }  
  
    return sum / n;  
}
```

```
int main() {  
    ofstream out;  
    out.open("output2.txt");  
    srand48(12);  
    int last = 10;
```

```

out << "k n moment" << endl;

for (int k = 1; k < 4; k++) {
    for (int ex = 2; ex < 6; ex++) {
        int n = pow(10, ex);
        out << k << " " << n << " " << momentCustom(n, k, last) << endl;
    }
}

out << "k n moment" << endl;

for (int k = 1; k < 4; k++) {
    for (int ex = 2; ex < 6; ex++) {
        int n = pow(10, ex);
        out << k << " " << n << " " << moment48(n, k) << endl;
    }
}

return 0;

}
[dwilk14@tigers ~/HW10]$ g++ dwilk14_hw10p2.cpp -o dwilk14_hw10p2
[dwilk14@tigers ~/HW10]$ ./dwerk14_hw10p2
[dwilk14@tigers ~/HW10]$ cp dwilk14_hw10p2.txt /home3/kristina/p
phtrol.sit phys2411/
[dwilk14@tigers ~/HW10]$ cp dwilk14_hw10p2.txt /home3/kristina/phys2411/.
[dwilk14@tigers ~/HW10]$ exit
exit

```

Script done on Fri 19 Nov 2021 04:46:53 PM CST