Robbie McKinstry

2/17/16

Introduction to Simulation

Golang and Rand

The Golang standard library originally did not have a fast random number generator for common usage. During Go's infancy, only a cryptographically secure generator was provided in the standard library. As such, generation was intentionally slow.

Before long, Go added a linear congruential generator. However, since linear congruential generators are not ideal for Monte Carlow simulations, the Go core team and the community decided to switch to a different implementation. An algorithms by DP Mitchell and JA, both of Bell Labs, during the time of Plan 9 and Inferno. This is no surprise, since Rob Pike, the creator of Go, spearheaded Plan 9 and Inferno, so he knew both Mitchell and JA well.

Predictions

I expect all tests to pass at the 80% confidence interval. Since the linear congruential generator appears to be "pretty good", it would be surprising if it was no at least pretty good at fooling the tests and wound up appear nonuniform or dependent.

I anticipate the standard library implemention to pass all tests at all confidence intervals. The Go implementations of other utilities has been quiet solid in the past, (see common benchmarking results, and the HTTP/2 package by Brad Fitzpatrick), so I anticipate the algorithm to be both uniform and independent.

I anticipate the Lipschultz generator to fail to appear uniform at 95%, since having a modulus of 1 << 16 means that the range of the numbers produced is smaller than the range produced by a larger mod. As such, I anticipate a shorter period, and so there is potential for dependence. I'm honestly not too certain about my predicion, since 1 << 16 is still very large, despite a fairly large multiplier of 101427.

Lastly, I think the RANDU generator will perform poorly at the independence test as well, since it has a small increment and a small multiplier. As such, it will potentially fail to loop around, and the output values will be of increasing size until passing the modulus.

After inspecting the random numbers, I found them to be sufficiently random, since there appears to me to be an even spread, and the numbers were sufficiently diverse. Additionally, I didn't note any patterns in the output.

Results

Here are the test statitics. The seed used is 122949823, which was chosen since it was prime and sufficiently large.

Kolmogorov

```
Standard Library Test Statistic:        0.072961
Lipschultz Test Statistic:              0.064654
RANDU Test Statistic:                   0.093862
```

Runs

```
Standard Library Test Statistic:        -0.007877
Lipschultz Test Statistic:              -0.005064
RANDU Run Test Statistic:               -0.001688
```

Autocorrelation

```
Standard Library Test Statistic:        0.509099
Lipschultz Test Statistic:              0.391374
RANDU Test Statistic:                   0.095305
```

Chi Squared

```
Standard Library Test Statistic:        3.240000
Lipschultz Test Statistic:              6.134000
RANDU Test Statistic:                   19.376000
```

Significance

Kolmogorov at 80%

Critical Value: 0.107

```
Standard Library Test Statistic:        Insignificant
Lipschultz Test Statistic:              Insignificant
RANDU Test Statistic:                   Insignificant
```

Kolmogorov at 90%

Critical Value: 0.122

```
Standard Library Test Statistic:        Insignificant
Lipschultz Test Statistic:              Insignificant
RANDU Test Statistic:                   Insignificant
```

Kolmogorov at 95%

Critical Value: 0.107

```
Standard Library Test Statistic:        Insignificant
Lipschultz Test Statistic:              Insignificant
RANDU Test Statistic:                   Insignificant
```

Probabilities:

Runs

```
Standard Library Test Statistic:        .500
Lipschultz Test Statistic:              .500
RANDU Run Test Statistic:               .500
```

Autocorrelation

```
Standard Library Test Statistic:        .69146
Lipschultz Test Statistic:              .65173
RANDU Test Statistic:                   .83891
```

Thus, the significance tests are rejected at all confidence intervals, since all values are outside

[.20, .80]. The exception to this is the autocorrelation test, which is significant at the 80% confidence interval.

Discussion

All tests came back insignificant. The only exception was that the RANDU test was significant at the 80% confidence interval. Because the test is only significant at the 80% interval, the results are spurious. It's certainly possible that the test results could have been due to chance. I suspect that the RANDU test came back signficant because of the long modulus and the absence of a `c` value. This should indicate that there's a longer period, and thus more likely to have monotone subsequences.