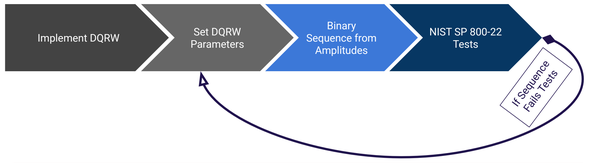
SUNY Polytechnic Institute

Javier Liu, Dr. Chen-Fu Chiang

Simulation of Coin Based Discrete Quantum Random Walk Pseudo-Random Number Generator

Goal Statement

To utilize the asymptotic, chaotic probability amplitudes of **Discrete Coin Based Quantum Random Walks (DQRW)** to produce binary sequences that pass the **NIST SP 800-22 Randomness Statistical Test Suite**



Introduction to Classical Random Walks

The Coin Based Classical Random Walk has the following algorithm:

1. On a number line, start at position x = 0
2. Flip a coin and track the result
3. Move right from position x if coin is heads, otherwise move left.
4. Repeat step 2, with new position x

Figure 1) Example of Classical Walk

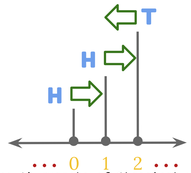


Figure 2) Example of Classical Walk

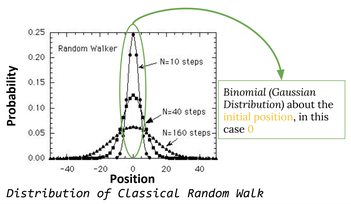


Figure 3) Distribution of Classical Random Walk

Introduction to Quantum Random Walks

The **Discrete Coin Based Quantum Random Walk** is the quantum implementation of the Classical Walk, but uses **qubits** and **superposition**.

Quantum system with two *subspaces*: $|x\rangle$ and $|c\rangle$, where x is the initial position (in this case 0) and c is the coin state, -1 or 1.

1. Two unitary operators
   - $\hat{C}$, **coin operator** which acts upon the *coin subspace*
   - $\hat{S}\ |x, 0\rangle = |x - 1, 0\rangle$ and $\hat{S}\ |x, 1\rangle = |x + 1, 1\rangle$, **shift operator** which looks at the *coin subspace* and acts upon the *position subspace* accordingly.
2. If $\hat{C}$ holds a 0, shift left. If $\hat{C}$ holds a 1, shift right.
3. Repeat step 2 until the walk is finished.
4. Measure the resulting position.

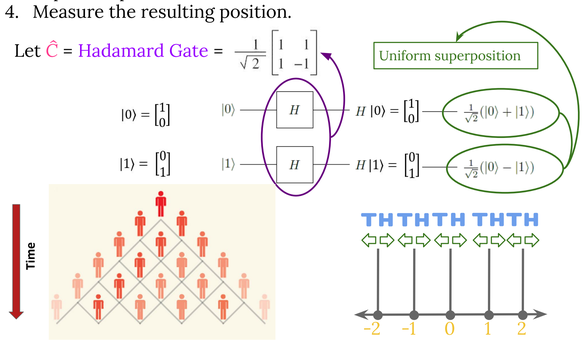


Figure 4) Example of Quantum Walk

Figure 5) Example of Quantum Walk

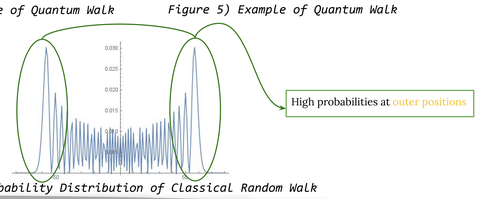


Figure 6) Probability Distribution of Classical Random Walk

NIST SP 800-22: Statistical Test Suite for PRNG

NIST SP 800-22 is a statistical test suite used to run on a random number generator (RNG) in order to see whether the data generated is *truly* "random." based on a *P-value*.

Frequency Tests

A. **Monobits Test:** Examines the proportion of 0s and 1s in the bitstream. If *P-value* is inadequate, either there were too many 1s or too many 0s.
B. **Block Test:** Examines the proportion of 0s and 1s in a m-bit block. Performs **Monobits Test** upon the block.

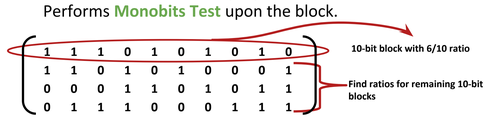


Figure 7) Example of Frequency Tests

Longest Run of Ones Test: Examines the length of the longest consecutive ones in the bitstream and compares it to that of a truly random sequence.

1 0 0 1 0 1 1 1 1 1 1 1 0 0 0

Figure 9) Example of Longest Run of Ones Test

Discrete Fourier Transform (Spectral) Test: Examines the number of peak heights in the Discrete Fourier Transform of the sequence and compares it to that of the Discrete Fourier Transform of a truly random sequence to check for periodic patterns.

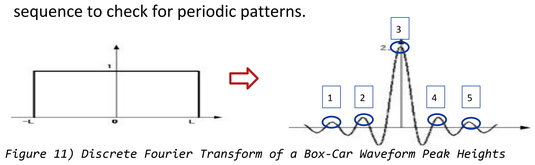


Figure 11) Discrete Fourier Transform of a Box-Car Waveform Peak Heights

Maurer's "Universal Statistical" Test: Examines the number of bits between recurring patterns to check if the sequence can be significantly compressed without loss of data. If *P-value* is inadequate, it implies the sequence is significantly compressible.

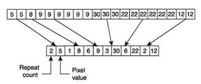


Figure 13) Example of Compression Algorithm: Run Length Encoding

Serial Test: Examines the frequency of all possible (m),(m-1), and (m-2)-bit, patterns in the sequence and compares it to that of a truly random sequence. If *P-value* is inadequate, the sequence's patterns are not uniform.

1 0 0 1 0 1 0 1 0 1 1 1 0 0 0

(m) = 3 ⟹ m_{000}= 1,m_{001}= 1,m_{010}= 3,m_{011}= 1,m_{100}= 2,m_{101}= 3,m_{110}= 1,m_{111}= 1
(m-1) = 2 ⟹ m-1_{00}= 2 ,m-1_{01}= 4 ,m-1_{10}= 5 ,m-1_{11}= 2
(m-2) = 1 ⟹ m-2_0= 8, m-2_1= 7

Figure 15) Example of Serial Test

Approximate Entropy Test: Examines the frequency of all possible (m) and (m+1)-bit patterns in the sequence and compares it to that of a truly random sequence. Similar to the Serial Test in its purpose.

1 0 1 1 0 0 0 1 0 0 1 1 0 0 1

(m+1) = 3 ⟹ m_{000}= 1,m_{001}= 3,m_{010}= 1,m_{011}= 2,m_{100}= 3,m_{101}= 1,m_{110}= 2,m_{111}= 0
(m) = 2 ⟹ m-1_{00}= 4 ,m-1_{01}= 4 ,m-1_{10}= 4 ,m-1_{11}= 2

Figure 16) Example of Approximate Entropy Test

Cumulative Sum (Cusum) Test: Examines the maximum excursion from 0 of the binary sequence and compares it to that of a truly random sequence by converting bits 0, 1 into digits -1, 1 (respectively), and finding the partial sums and cumulative sums. If *P-value* is inadequate, the sequence has too many 1s or 0s towards the end of the sequence, or the 1s and 0s are mixed too evenly.

0 1 0 0 1 1 1 0

$S_1 = X_1$ ⟹ $S_1 = (-1)$
$S_2 = X_1 + X_2$ ⟹ $S_2 = (-1) + 1 = 0$
... ⟹ $S_3 = (-1) + 1 + (-1) = -1$
... ⟹ $S_4 = (-1) + 1 + (-1) + (-1) = -2$
$S_k = X_1 + X_2 + \ldots + X_k$ ⟹ $S_5 = (-1) + 1 + (-1) + (-1) + 1 = -1$
... ⟹ $S_6 = (-1) + 1 + (-1) + (-1) + 1 + 1 = 0$
... ⟹ $S_7 = (-1) + 1 + (-1) + (-1) + 1 + 1 + 1 = 1$
$S_n = X_1 + X_2 + \ldots + X_k + \ldots + X_n$ ⟹ $S_8 = (-1) + 1 + (-1) + (-1) + 1 + 1 + 1 + (-1) = 0$

Maximum excursion of 2

Figure 17) Example of Cusum Test

Runs Test: Examines the number of "runs" in the bitstream and compares it to that of a truly random sequence. If *P-value* is inadequate, it implies the oscillation between 1s and 0s was either too fast or too slow.

1 0 0 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0

Number of "runs": 1 2 3 4 5 6 7 8 9 10

Figure 8) Example of Runs Test

Binary Matrix Rank Test: Examines the ranks of disjoint submatrices of the bitstream compares it to that of a truly random sequence to check for linear dependency within the sequence.

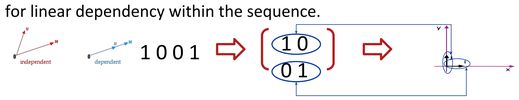


Figure 10) Example of Linearly Independent 2x2 Matrix

Template Matching Tests

A. **Non-overlapping Template Matching Test:** Examines the occurrences of a specified m-bit string in a m-bit window in the sequence. If pattern is found, set the window to the bit after the pattern. Otherwise, set it to the next bit in the sequence.
B. **Overlapping Template Matching Test:** Performs **Non-overlapping Template Matching Test** except when a pattern is found, set the window to the next bit in the sequence.

Template: 1011

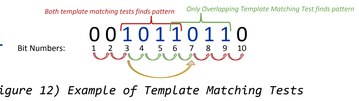


Figure 12) Example of Template Matching Tests

Linear Complexity Test: Examines the length of the linear feedback shift register (LFSR) of the sequence to see whether the sequence is complex enough to that of a truly random sequence. Inadequate *P-values* implies the sequence has a short LFSR.

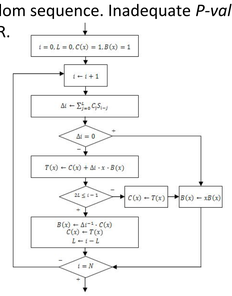
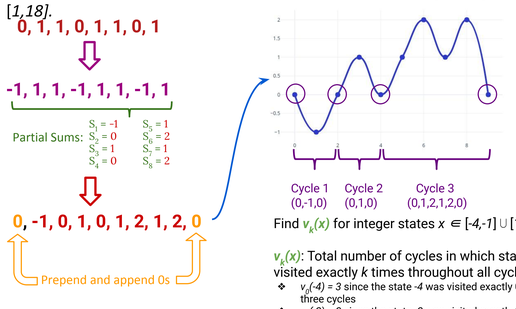


Figure 14) Berlekamp-Massey Algorithm Used to Find Sequence LFSR

Random Excursions and Variant Tests: Examines the number of "cycles" with a certain amount of occurences of a particular state and comparing it to that of a truly random sequence. If *P-value* is inadequate, the sequence's distribution of a particular state across all "cycles." The normal Random Excursions Test examines integer states *[-4,-1]* ∪ *[1,4]*, while the variant Random Excursions Test examines states *[-18,-1]* ∪ *[1,18]*.



Find $v_k(x)$ for integer states $x \in [-4,-1] \cup [1,4]$

$v_k(x)$: Total number of cycles in which state *x* is visited exactly *k* times throughout all cycles

- $v_0(-4) = 3$ since the state -4 was visited exactly 0 times in all three cycles
- $v_0(-2) = 3$ since the state -2 was visited exactly 0 times in all three cycles
- ...
- $v_2(2) = 1$ since the state 2 was visited exactly 2 times in only one cycle
- $v_0(4) = 3$ since the state 4 was visited exactly 0 times in all three cycles

Figure 18) Example of Random Excursions and Random Excursions Variant Tests

Results

P-value method for hypothesis testing

Null Hypothesis, H_0: Binary bitstream is a random sequence

Level of significance $\propto = 0.01$

- If sequence was accepted as random, there's an expected 1 sequence in 100 sequences to be rejected as non-random.
- If *P-value* ≥ 0.01, it indicates the sequence is random with 99% confidence
- If *P-value* ≤ 0.01, it indicates the sequence is non-random with 99% confidence

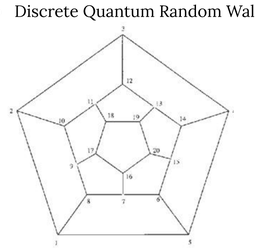
NIST SP800-22 Randomness Tests for the DQRWs-based PRNG.

Test Name	*P-value*	Result
Frequency	0.350485	SUCCESS
BlockFrequency	0.122325	SUCCESS
CumulativeSums (forward)	0.739918	SUCCESS
CumulativeSums (reverse)	0.534146	SUCCESS
Runs	0.739918	SUCCESS
LongestRun	0.350485	SUCCESS
Rank	0.066882	SUCCESS
FFT	0.911413	SUCCESS
NonOverlappingTemplate	0.424142	SUCCESS
OverlappingTemplate	0.534146	SUCCESS
Universal	0.600000	SUCCESS
Serial (Test 1)	0.834308	SUCCESS
Serial (Test 2)	0.637119	SUCCESS
LinearComplexity	0.911413	SUCCESS
Random Excursions	----------	----------
Random Excursions Variant	----------	----------
Approximate Entropy	0.000000	FAIL

Figure 19) Table of Results

Future Endeavors

- Approximate Entropy Test
- Random Excursions Test
- Random Excursions Variant Test
- Discrete Quantum Random Walk along higher dimensional structures



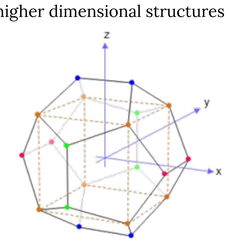
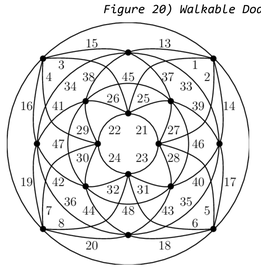


Figure 20) Walkable Dodecahedron Structure



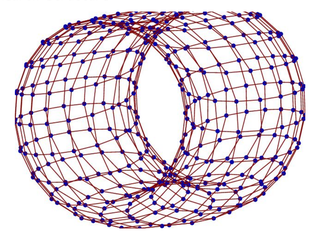


Figure 21) Walkable Toroidal Structure