Testing Random Number Generators

Dr. John Mellor-Crummey

Department of Computer Science
Rice University

johnmc@cs.rice.edu

Testing Random Number Generators

Does observed data satisfies a particular distribution?

- **Chi-square test**
- **Kolmogorov-Smirnov test**
- **Serial correlation test**
- **Two-level tests**
- **K-distributivity**
- **Serial test**
- **Spectral test**

Chi-Square Test

- **Designed for testing discrete distributions, large samples**
- **General test: can be used for testing any distribution**
  - —uniform random number generators
  - —random variate generators
- **The statistical test:** $\sum_{k=1}^{n} \frac{(o_i - e_i)^2}{e_i} < \chi^2_{[1-\alpha;k-1]}$?
- **Components**
  - —k is the number of bins in the histogram
  - —o_i is the number of observed values in bin i in the histogram
  - —e_i is the number of expected values in bin i in the histogram
- **The test**
  - —if the sum is less than $\chi^2_{[1-\alpha;k-1]}$, then the hypothesis that the observations come from the specified distribution cannot be rejected at a level of significance α

Chi-Square Constraints

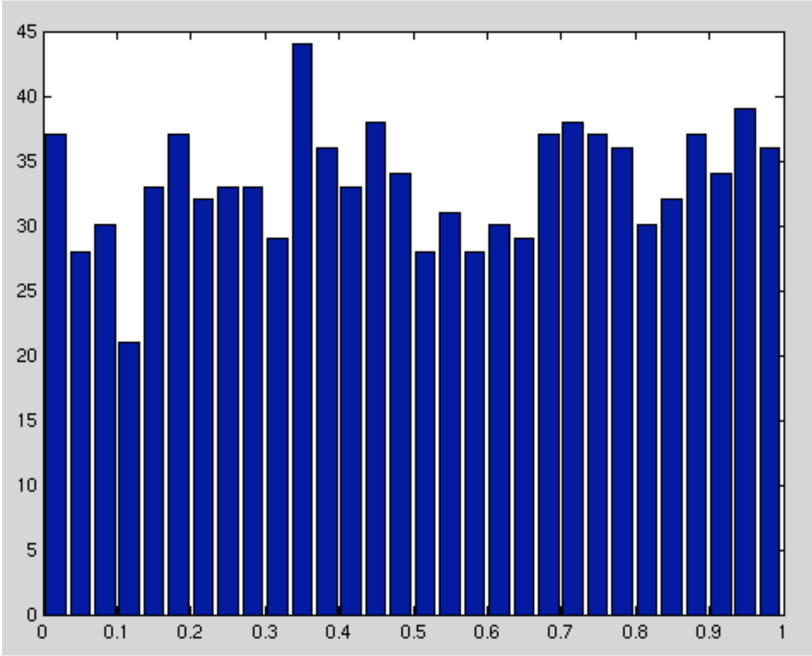
- Data must be a random sample of the population
  - —to which you wish to generalize claims
- Data must be reported in raw frequencies (not percentages)
- Measured variables must be independent
- Values/categories on independent and dependent variables
  - — must be mutually exclusive and exhaustive
- Observed frequencies cannot be too small
- Use Chi-square test only when observations are independent:
  - —no category or response is influenced by another
- Chi-square is an approximate test of the probability of getting the frequencies you've actually observed if the null hypothesis were true
  - —based on the expectation that within any category, sample frequencies are normally distributed about the expected population value
  - —distribution cannot be normal when expected population values are close to zero since frequencies cannot be negative
  - —when expected frequencies are large, there is no problem with the assumption of normal distribution

Chi-Square Test of Matlab's U(0,1)

```
[n,x] = hist(rand(1000,1),30)
bar(x,n)

e = 1000/30.0
sum(power(n-e,2)/e)
```

17.900



$$\chi^2_{[.05;29]} = 17.708 \; < \; 17.900 \; < \; \chi^2_{[.10;29]} = 19.768$$

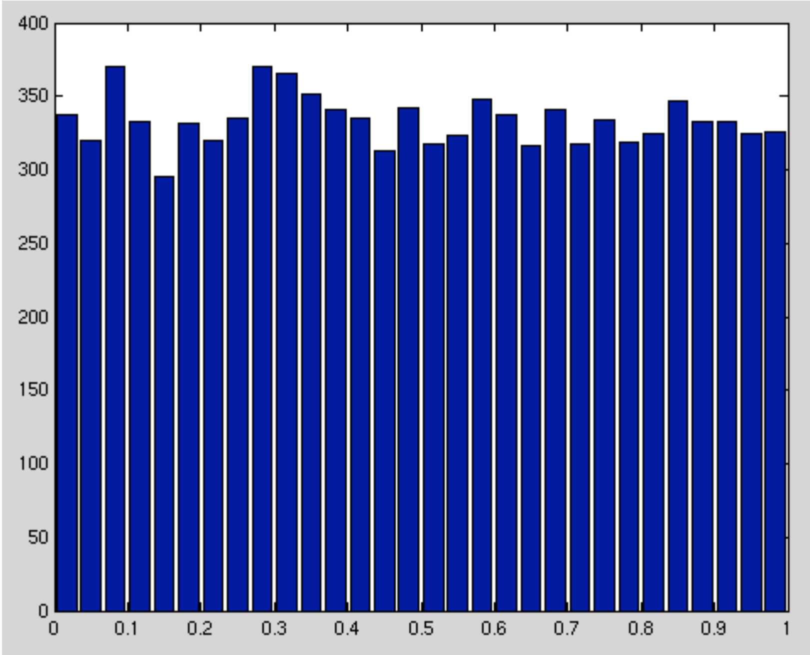
According to the result of the Chi-Square test, we can reject the null hypothesis that Matlab's random number generator generates uniform random numbers with only 5% confidence.

Chi-Square Test of Matlab's U(0,1)

```
[n,x] = hist(rand(10000,1),30)
bar(x,n)

e = 10000/30.0
sum(power(n-e,2)/e)
```

24.71



$$\chi^2_{[.200;29]} = 22.475 < \ 24.71 \ < \ \chi^2_{[.500;29]} = 28.336$$

According to the result of the Chi-Square test, we can reject the null hypothesis that Matlab's random number generator generates uniform random numbers with only 20% confidence.

Kolmogorov-Smirnov Test

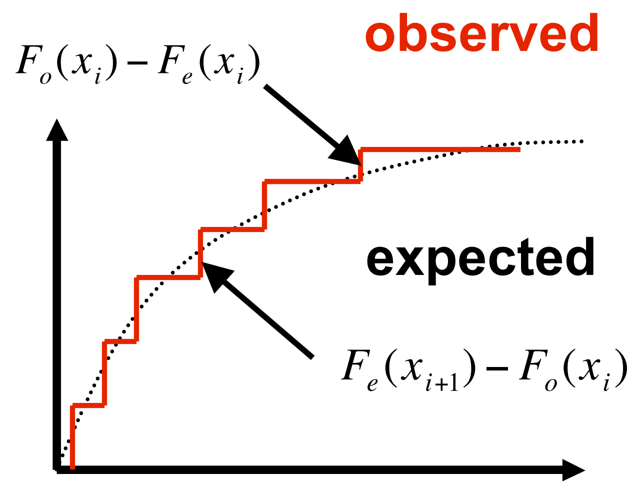
Test if sample of n observations is from a continuous distribution

- **Compare CDF $F_o(x)$ (observed) and CDF $F_e(x)$ (expected)**
  - — **difference between CDF $F_o(x)$ and CDF $F_e(x)$ should be small**
- **Maximum deviations**
  - —**K^+ above expected CDF**

$$K^+ = \sqrt{n} \max_x [F_o(x) - F_e(x)]$$

  - —**K^- below expected CDF**

$$K^- = \sqrt{n} \max_x [F_e(x) - F_o(x)]$$



- **Statistical test**
  - —**if K^+ and K^- are smaller than $K_{[1-\alpha;n]}$, observations said to come from expected distribution with α level of significance**

Kolmogorov-Smirnov Test of U(0,1)

- For uniform random numbers between 0 and 1
  - —expected CDF $F_e(x) = x$
- If x > j-1 observations in a sample of n observations
  - —observed CDF $F_o(x) = j/n$
- To test whether a sample of n random numbers is from U(0,1)
  - —sort n observations in increasing order
  - —let the sorted numbers be $\{x_1, x_2, \ldots, x_n\}$, $x_{n-1} \leq x_n$

$$K^+ = \sqrt{n} \max_j \left(\frac{j}{n} - x_j \right)$$

$$K^- = \sqrt{n} \max_j \left(x_j - \frac{j-1}{n} \right)$$

- Compare resulting K^+, K^- values with those in table
  - —if K^+, K^- values less than K-S table $K_{[1-\alpha;n]}$, observations said to come from the same distribution at α level of significance

K-S Test vs. Chi-Square Test

- K-S test: designed for
  - —small samples
  - —continuous distribution
- Chi-square test: designed for
  - —large samples
  - —discrete distribution
- K-S
  - —based on differences between observed and expected CDF
  - —uses each sample without grouping
  - —exact test, provided parameters of expected distribution known
- Chi-square
  - —based on differences between observed and hypothesized PMF or PDF
  - —requires samples be grouped into small number of cells
  - —approximate test, sensitive to cell size
    - no firm guidelines for choosing appropriate cell sizes

Serial Correlation Test

- **Test if 2 random variables are dependent**
  - —is their covariance non-zero?
    - if so, dependent. <u>converse not true.</u>
- **Given a sequence of random numbers**
  - —autocovariance at lag k, $k \geq 1$

$$R_k = \frac{1}{n-k} \sum_{i=1}^{n-k} \left(U_i - \tfrac{1}{2}\right)\left(U_{i+k} - \tfrac{1}{2}\right)$$

  - —for large n, R_k is normally distributed
    - 0 mean
    - variance of 1/[144(n-k)]
- **Confidence interval for autocovariance** $R_k \pm z_{1-\alpha/2} / \left(12\sqrt{n-k}\right)$
  - —if the interval does not include 0
    - sequence has a significant correlation
- **For k = 0**
  - —R_0 is the variance of the sequence
  - —expected to be 1/12 for IID U(0,1)

Two-level Tests

- **If sample size too small**
  - —previous tests may apply locally but not globally
  - —similarly, global tests may not apply locally
- **Two-level test**
  - —use Chi-square on n samples of size k each
  - —use Chi-square test on set of n Chi-square statistics computed
- **Called: chi-square-on-chi-square test**
- **Similarly: K-S-on-K-S test**
  - —has been used to identify non-random segment of otherwise random sequence

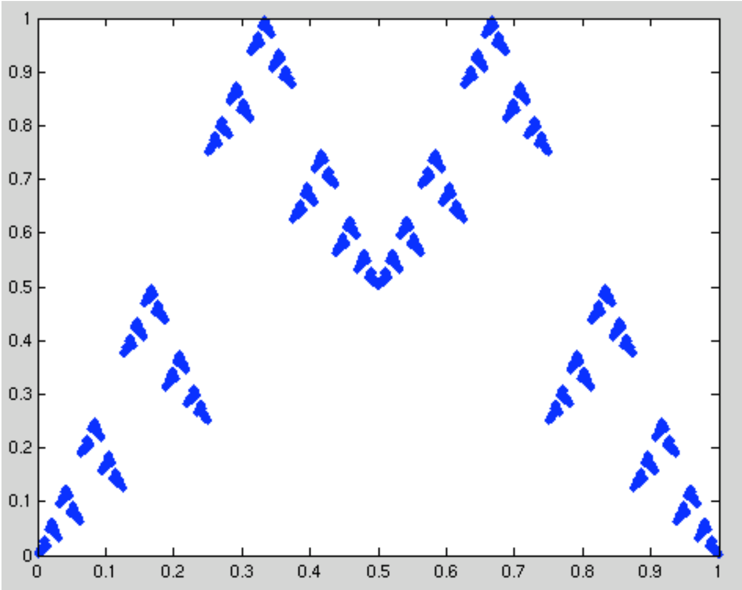
K-Distributivity

AKA k-dimensional uniformity

- Suppose u_n is the n^{th} number in a random sequence U(0,1)
- 1-distributivity
  - given two real numbers a_1 and b_1, $0 < a_1 < b_1 < 1$
  - $P(a_1 \leq u_n < b_1) = b_1 - a_1, \forall\, b_1 > a_1$
- 2-distributivity: generalization in two dimensions
  - given real numbers a_1, b_1, a_2, b_2, such that
    - $0 < a_1 < b_1 < 1$ and $0 < a_2 < b_2 < 1$
  - $P(a_1 \leq u_{n-1} \leq b_1 \text{ and } a_2 \leq u_n \leq b_2) = (b_1 - a_1)(b_2 - a_2), \forall\, b_1 > a_1, b_2 > a_2$
- k-distributivity
  - $P(a_1 \leq u_n \leq b_1 \ldots a_k \leq u_{n+k-1} \leq b_k) = (b_1 - a_1) \ldots (b_k - a_2)$
  - $\forall\, b_i > a_i,\ i=1,2,\ldots k$
- Properties
  - k-distributed sequence is always k-1 distributed (inverse not true)
    - sequence may be uniform in lower dimension, but not higher

2D Visual Check of Overlapping Pairs

- **Example:**
  - Tausworthe $x^{15}+x^{1}+1$ primitive polynomial
  - compute full period sequence of $2^{15}-1$ points
  - plot (x_n,x_{n+1}), n = 1, 2, …, $2^{15}-2$

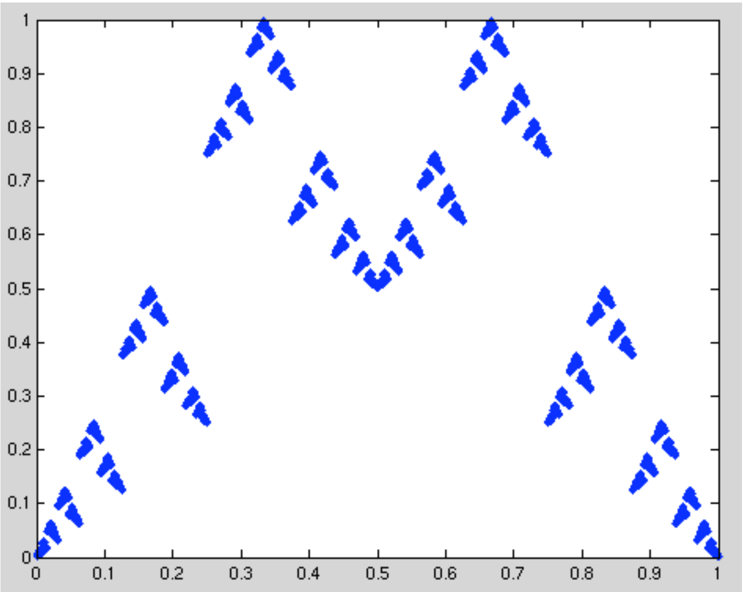


tw2d([15 1 0])

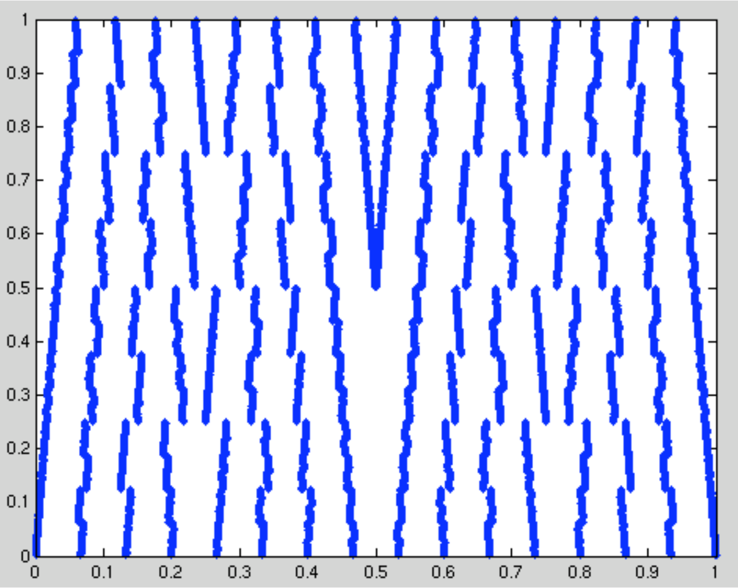
```
function [] = tw2d(polynomial)
p = polynomial(1);
r = ones(1,p);
upper = power(2,p)-1;
x = zeros(upper,1);
for i =1:upper
        n = 0;
        for bits = 1:p
            [b,r] = tausworthe(r,polynomial);
            n = 2 * n + b;
        end
        x(i) = n/upper;
end;
x1 = x(1:(upper-1));
x2 = x(2:upper);
plot(x1,x2,'.');
```

2D Visual Check of Overlapping Pairs

- **Example:**
  - —Comparing two Tausworthe polynomials
  - —$x^{15}+x^{1}+1$ primitive polynomial vs. $x^{15}+x^{4}+1$ primitive polynomial compute full period sequence of 2^{15}-1 points
  - —plot (x_n,x_{n+1}), n = 1, 2, …, 2^{15}-2



tw2d([15 1 0])



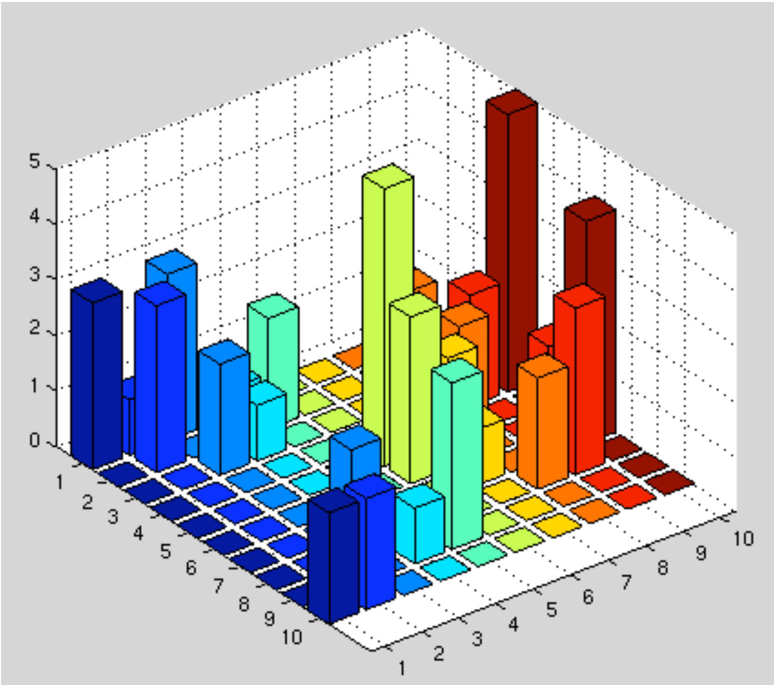
tw2d([15 4 0])

Serial Test

- Check for uniformity in 2D or higher
- In 2D: divide space up into K^2 cells of equal area
- Given n random numbers $\{x_1, x_2, \ldots, x_n\}$ between 0 and 1
- Construct n/2 non-overlapping pairs (x_1,x_2), (x_3,x_4), …, (x_{n-1},x_n)
- Count points in each of K^2 cells
  - —expect $n/(2K^2)$ per cell
- Test deviation of actual counts from expected counts with Chi-square test
  - —DOF=K^2-1
  - —tuples must be non-overlapping, otherwise cell independence for Chi-square is violated
- Can extend this to k dimensions
  - —k-tuples of non-overlapping values

Example: Serial Test

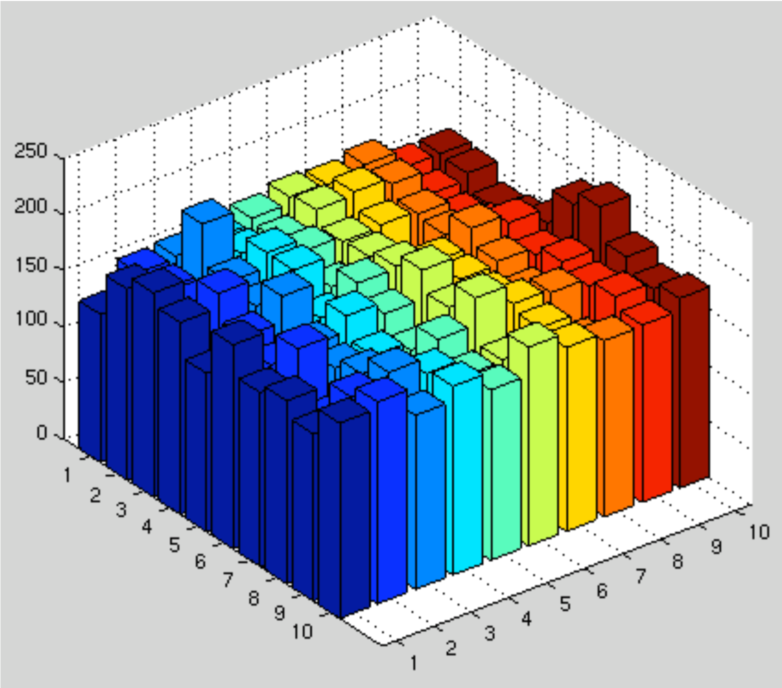
- **Tausworthe polynomial**
  - x^7+x^1+1



```
function [] = tw3d(polynomial)
p = polynomial(1);
r = ones(1,p);
upper = power(2,p)-1;
x = zeros(upper,1);
for i =1:upper
   n = 0;
   for bits = 1:p
     [b,r]= tausworthe(r,polynomial);
     n = 2 * n + b;
   end
   x(i) = n/upper;
end;
top = upper -1 % top is even
x1 = x(1:2:(top-1));
x2 = x(2:2:top);
top2 = top/2;
y = zeros(10,10);
for j = 1:top2
    s1 = min(10,1+floor(x1(j,1)*10));
    s2 = min(10,1+floor(x2(j,1)*10));
    y(s1,s2) = y(s1,s2)+1;
end
bar3(y);
```

Example: Serial Test on Matlab's `rand`

- **Matlab rand**
  - — 2^{15}-1 elements



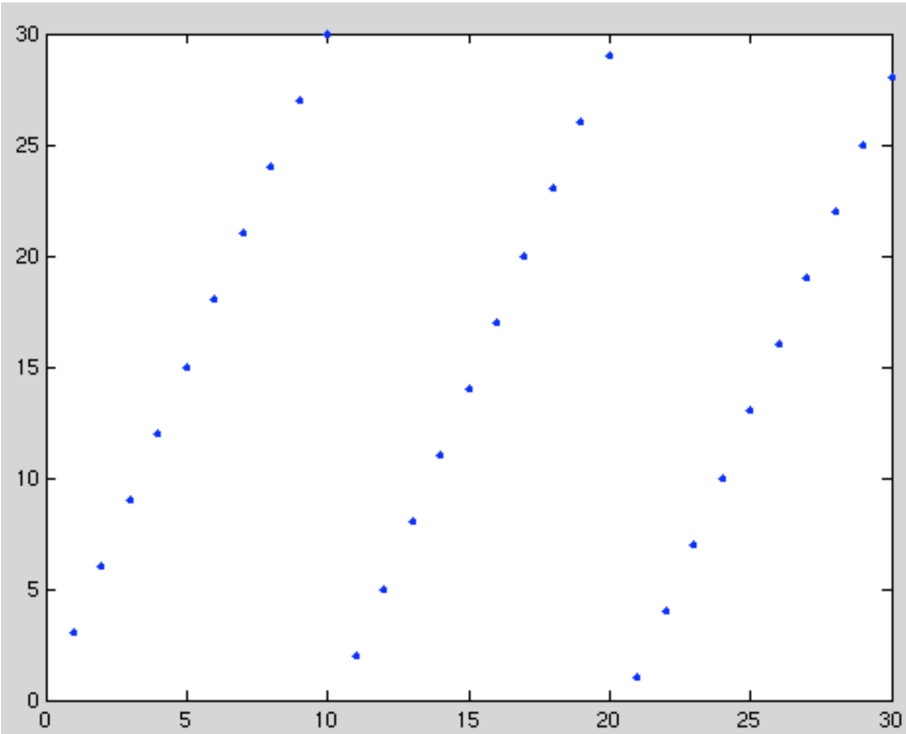

```
function [] = rand3d(p)
upper=power(2,p)-1;
x=rand(upper,1);
top = upper -1 % top is even
x1 = x(1:2:(top-1),1);
x2 = x(2:2:top,1);
top2 = top/2;
y = zeros(10,10);
for j = 1:top2
    s1 =
min(10,1+floor(x1(j,1)*10));
    s2 =
min(10,1+floor(x2(j,1)*10));
    y(s1,s2) = y(s1,s2)+1;
end
bar3(y);
```

Spectral Test

- **Determine how densely k-tuples fill k-dimensional hyperplanes**
- **k-tuples from an LCG fall on a finite number of parallel hyperplanes**
  - —in 2D, pairs of adjacent numbers will lie on finite # lines
  - —in 3D, triples of adjacent numbers will lie on finite # planes
- **Marsaglia (1968)**
  - —shoed that successive k-tuples from an LCG fall on at most $(k!m)^{1/k}$ parallel hyperplanes for LCG with modulus m
- **The test**
  - —determine the maximum distance between adjacent hyperplanes
  - —the greater the distance, the worse the generator

Spectral Test Intuition for LCGs

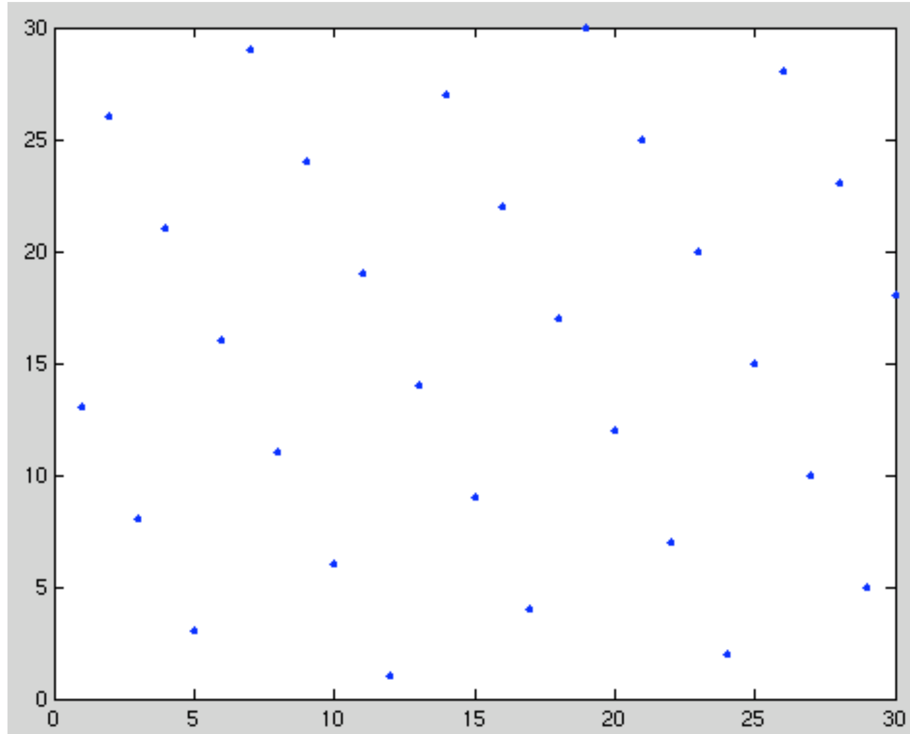
LCG $x_n = 3x_{n-1} \bmod 31$



All numbers lie on lines

$x_n = 3x_{n-1} - 31k$, for $k = 0,1,2$

LCG $x_n = 13x_{n-1} \bmod 31$



All numbers lie on lines

$x_n = (-5/2)x_{n-1} - (31/2)k$,

for $k = 0,1,\ldots,5$

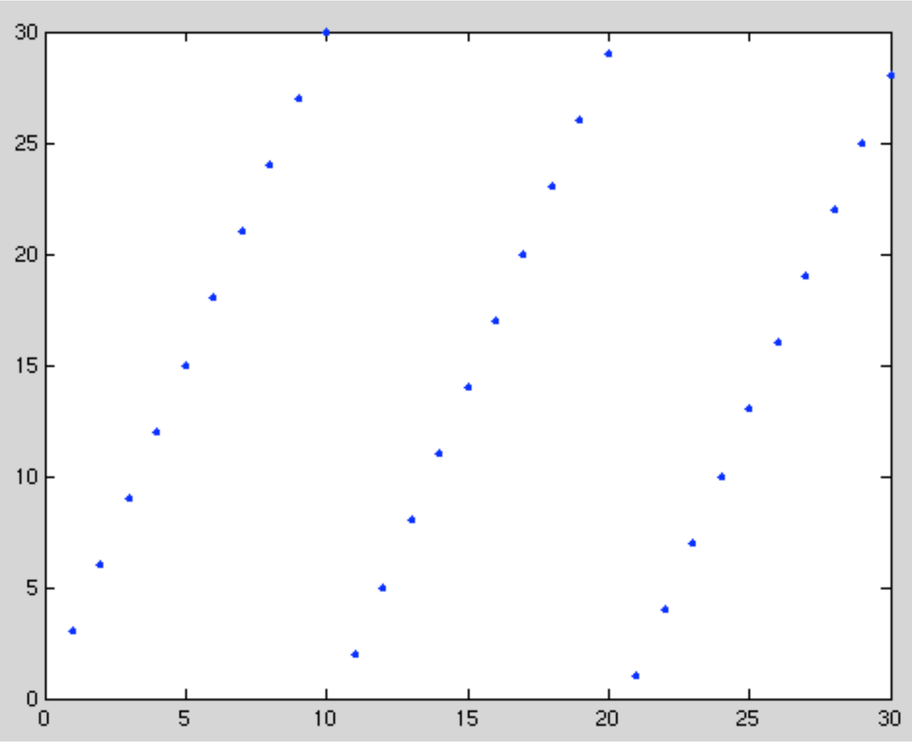
Computing the Spectral Test

- **Distance between two lines**

  $y = ax + c_1$ and $y = ax + c_2$ is given by $\quad | c_2 - c_1 | / \sqrt{1 + a^2}$

Spectral Test for LCGs

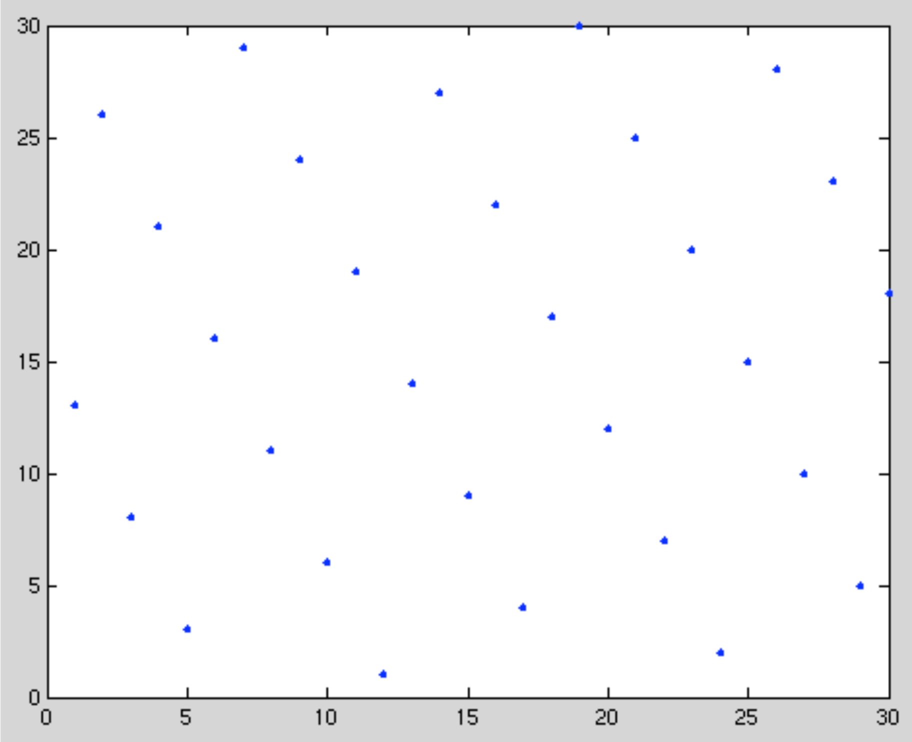
LCG $x_n = 3x_{n-1} \bmod 31$



$x_n = 3x_{n-1} - 31k$, for k = 0,1,2

$$|c_2 - c_1| / \sqrt{1 + a^2} = 31/10 = 9.80$$

LCG $x_n = 13x_{n-1} \bmod 31$



$x_n = (-5/2)x_{n-1} - (31/2)k$,
for k = 0,1,…,5

$$|c_2 - c_1| / \sqrt{1 + a^2} = (31/2) / \sqrt{1 + \left(\frac{5}{2}\right)^2} = 5.76$$