What is the Difference Between Random and Chaotic Sequences

In everyday language, people tend to use the words “random” and “chaotic” interchangably. You will recall from the text that chaotic sequences are in fact generated **deterministically** from the **dynamical system**

$$x_{n+1} = f(x_n) \tag{1}$$

where f is a smooth function on $\mathcal{R}^m$.

This implies that if you create two orbits with the identical initial data, x_0, then the orbits are the same. What makes the dynamical system “chaotic” is the fact that orbits arising from initial data which are arbitrarily close grow apart exponentially.

A bounded sequence of values $\{x_i\}_{i=1}^{\infty}$ coming from (1) is chaotic if

1. $\{x_i\}$ is not asymptotically periodic
2. No Lyapunov exponent vanishes
3. The largest Lyapunov exponent is strictly positive

Random (sometimes called **stochastic**) processes are fundamentally different. Two successive realizations of a random process will give different sequences, even if the initial state is the same.

Since a random process is **non-deterministic**, numerical computation of a “Lyapunov exponent” is not well defined. Wolf’s algorithm (applied to a sequence) looks at the closest neighbor $y_0 = x_K$ of a point x_J and uses it as the initial value of another sequence. The average value of the rate of separation

$$\ln\left(\frac{\|x_{K+1} - x_{J+1}\|}{\|x_K - x_J\|}\right) \tag{2}$$

is used as a measure of the Lyapunov exponent.

Consider the case when the x_i are uniformly randomly distributed on $[0, 1]$. If the x_i’s are truly random then the denominator of (2) may be arbitrarily small while the numerator is large. As the number of points increases, the minimal separation distance approaches zero, while the mean separation of two arbitrary points (“correlation length”) remains bounded away from zero. Consequently, the Lyapunov exponent will increase without bound as the number of points approachs infinity.

We can actually estimate the following for a uniform random sequence of values on $[0, 1]$

$$\sum_{i=1}^{N-1} |x_{i+1} - x_i|^2 = \sum_{i=1}^{N-1} |(x_{i+1} - \bar{x}) - (x_i - \bar{x})|^2$$

$$= \sum_{i=1}^{N-1} |x_{i+1} - \bar{x}|^2 + \sum_{i=1}^{N-1} |x_i - \bar{x}|^2 - 2\sum_{i=1}^{N-1} (x_{i+1} - \bar{x})(x_i - \bar{x})$$

$$\approx 2N\sigma^2 - 2N\alpha$$

where α is the autocorrelation coefficient

$$\frac{\sum_{i=1}^{N-1} (x_{i+1} - \bar{x})(x_i - \bar{x})}{\sum_{i=1}^{N-1} |x_i - \bar{x}|^2}$$

and σ^2 is the variance

$$\sigma^2 = \frac{\sum_{i=1}^{N} |x_i - \bar{x}|^2}{N - 1}$$

Therefore the expected value of $|x_{i+1} - x_i|^2$ is approximately $2\sigma^2 - 2\alpha$. Since $\sigma^2 = 1/12$ and $\alpha \approx 1/N$ for uniform random data, the mean length squared is approximately twice the variance (i.e. 1/6).

This can be verified by running the following matlab program for large enough values:

```
function [d1,d2]=clen(n)

d1=0;
d2=0;
for i=1:n
    x = rand;
    y = rand;
    d1 = d1 + abs(x-y);
    d2 = d2 + (x-y)^2;
end;
d1 = d1/n;
d2 = d2/n;
```

Interestingly enough, this program returns $d1 = 1/3$ and $d2 = 1/6$ asymptotically. I can prove the second result but not the first ...