International Journal of Sciences

Research Article Volume 11 – August 2022 (08)

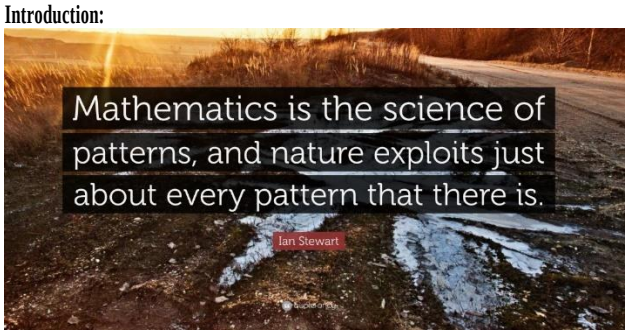
A New Way to See Mathematical Patterns

Maher Ali Rusho

*Dedicated to **Sowrav Mozumder**

Abstract: Pattern is voice of Math -Math is the voice of science-Science is the voice of God. Pattern is the beauty in Mathematics. Even a newborn-baby can also detect patterns in nature. But not all pattern can easily be detected by our brain. We need to generalize this into a new frame of mathematics. So I I have invented a new method called EULER STOCHASTIC PROCESS. To prove these lemma we will use basic euler-identity method. I am grateful to Sowrav Mozumder to give me this wonderful problem. I am going to prove this by the help of my computer program.

Introduction:



Src: **https://quotefancy.com/quote/1507763/Ian-Stewart-Mathematics-is-the- science-of-patterns-and-nature-exploits-just-about-every**

The main theme of this research paper to find this solution of this question:
What will be the next number in this series:
92-16-32-42-74-81-25-41-91-50-54-57-61-42-8-55-
73-96-91-31-42-74-31-67-68-47-96-30-17-32-53...

FROM LOOKing AT fiRST iT SeeMS HARD . BUT if we USe POE MeTHoD oR pRoceSS of eLiMinATion MeTHoD By THe HeLp of goMpUTeR pRogRAM ,we cAn eASiLy SoLVe THiS pROBLeM . LeT'S ASSUMe THe Numbers are evenly distributed in sample space and called this numbers a point topological space , We also assume that this pattern is in poisson distribution . because here the number system can be large enough ,it can be infinity also (assume!) ,but this probability is so little or vice versa . We know from poisson distribution the probabitly id:

P(x)= (lambda^x/x!)*e^-lambda
Here lambda indicates the mean of this stochastic process. We assume that the next number is X ,and write the PDF using varience of X. we know Var(x)=X^2(P(X)-P(X^2).
Then new equation form

P(X)=(np)^var(x)/P(x)-P(x)^2
Write it into this program:
%BPATH1 Brownian path simulation randn('state',100) % set the state of randn T = 1; N = 5 00; dt = T/N; dW = zeros(1,N); % preallocate arrays ... W = zeros(1,N); % for efficiency dW(1) = sqrt(dt)*randn; % first approximation outside the loop ... W(1) = dW(1); % since W(0) = 0 is not allowed for j = 2:N dW(j) = sqrt(dt)*randn; % general increment W(j) = W(j-1) + dW(j); end plot([0:dt:T],[0,W],'r-') % plot W against t xlabel('t','FontSize',16) ylabel('W(t)','FontSize',16,'Rotation',0)plot %BPATH1 Brownian path simulation randn('state',100) % set the state of randn T = 1; N = 5 00; dt = T/N; dW = zeros(1,N); % preallocate arrays ... W = zeros(1,N); % for efficiency dW(1) = sqrt(dt)*randn; % first approximation outside the loop ... W(1) = dW(1); % since W(0) = 0 is not allowed for j = 2:N dW(j) = sqrt(dt)*randn; % general increment W(j) = W(j-1) + dW(j); end plot([0:dt:T],[0,W],'r-') % plot W against t xlabel('t','FontSize',16) ylabel('W(t)','FontSize',16,'Rotation',0) randon output to P9x) finding

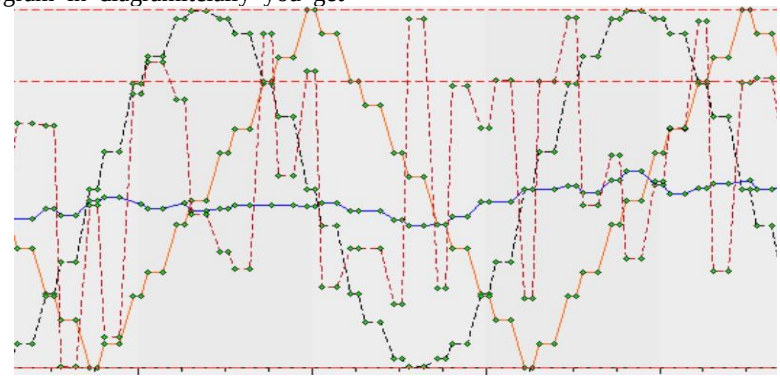
P(x)=(np)^var(x)?p(x)-p(x)2---Labds^np
(redirected to.
probability 1, and conditions 2 and 3 tell us that (2.1) Wj = Wj−1 + dWj , j = 1, 2,...,N, where each dWj is an independent random variable of the form √ δtN(0, 1). The MATLAB M-file bpath1.m in Listing 1 performs one simulation of discretized Brownian motion over [0, 1] with N = 500. Here, the random number generator randn is used—each call to randn produces an independent "pseudorandom" number from the N(0, 1) distribution. In order to make experiments repeatable, MATLAB allows the initial state of the random number generator to be set. We set the state, arbitrarily, to be 100 with the command randn('state',100). Subsequent runs of bpath1.m would then produce the same output. Different simulations can be performed by resetting the state, e.g., to randn('state',200). The numbers from randn are scaled by √δt and used as increments in the for loop that creates the 1-by-N array W. There is a minor inconvenience: MATLAB starts arrays from index 1 and not index 0. Hence, we compute W as W(1),W(2),...,W(N) and then use plot([0:dt:T],[0,W]) in order to include the initial value W(0) = 0 in the picture. Figure 1 shows the result; note that for the purpose of visualization, the discrete data has been joined by straight lines. We will refer to an array W

Published at: http://www.ijsciences.com/pub/issue/2022-08/
DOI: 10.18483/ijSci.2609; Online ISSN: 2305-3925; Print ISSN: 2410-4477

Maher Ali Rusho (Correspondence)
+

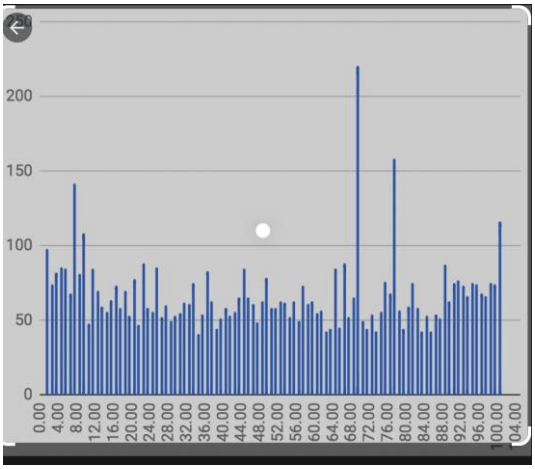
created by the algorithm in bpath1 as a discretized Brownian path. We can perform the same computation more elegantly and efficiently by replacing the for loop with higher level "vectorized" commands, as shown in bpath2.m in Listing 2. Here, we have supplied two arguments to the random number generator: randn(1,N) creates a 1-by-N array of independent N(0, 1) samples. The function cumsum computes the cumulative sum of its argument, so the jth element of the 1- by-N array W is dW(1) + dW(2) + ⋯ + dW(j), as required. Avoiding for loops and thereby computing directly with arrays rather than individual components is the key to writing efficient MATLAB code [3, Chapter 20]. Some of the M- files in this article would be several orders of magnitude slower if written in nonvectorized form. The M-file bpath3.m in Listing 3 produces Figure 2. Here, we evaluate the function u(W(t)) = exp(t + 1 2W(t)) along 1000 discretized Brownian paths. The average of u(W(t)) over these paths is plotted with a solid blue line. Five individual paths are also plotted using a dashed red line. The M- file bpath3.m is vectorized across paths;

The out put of this program in diagramitcially you get

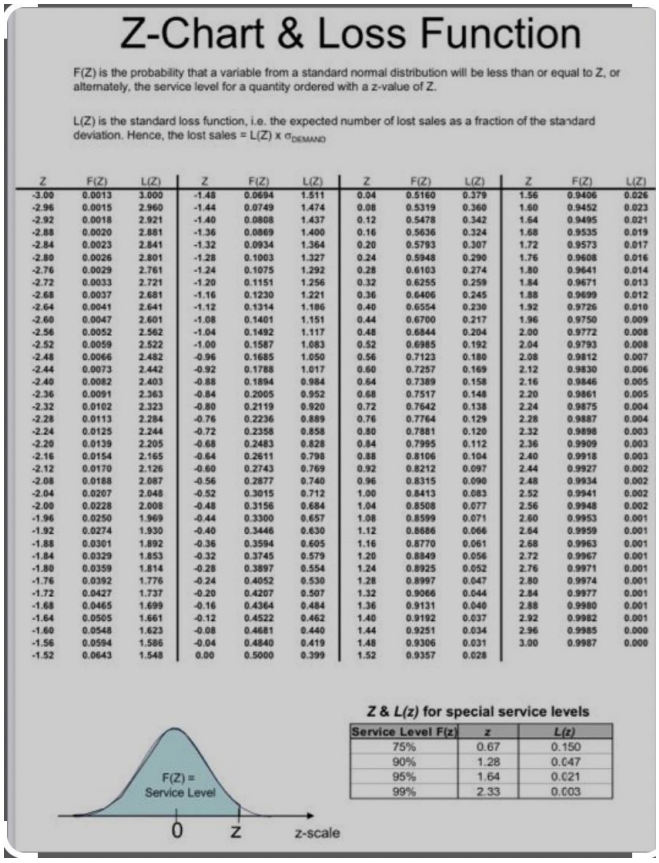


It is seemed to be totally randomized number pattern. But wait for a second If we see small phase iteration we will se there is a some periodic evnt, if we find the probability, we will get the nextpattern.

The vector diagram guaranteed that the number will not exceed 100. And here is the probality series of the next pattern We have to consider here the stochastic probability since it is a randomized series but it has some pattern.
Here is the list below:



We are seeing here the best fit possible is 68, But wait this is not the correct answer, scince it don't start from number 1, it starts number 92, then again iterate the program again and find the chart



We see from the chart the best fit line or probability is 92 which has 56% chance the highest probability to occur.