Random Variables and Brownian Motion

Des Higham
Department of Mathematics and Statistics
University of Strathclyde

http://fox.maths.strath.ac.uk/~aas96106/

University of Strathclyde
Science

Lecture Notes:

- Lectures don't overlap completely with the printed notes

Further Reading (see Handout for more details):

- Numerical
  - Cyganowski, Kloeden and Ombach
  - Kloeden & Platen
  - Milstein and Tretyakov
- SDE theory
  - Klebaner
  - Mao

Course Aim: Give an accessible intro. to SDEs and their numerical simulation.

Motivation: SDEs are becoming widely used in science and engineering; notably finance, physics and biology.

> "**It may very well be said that the best way to understand SDEs is to work with their numerical solutions.**"

Salih N. Neftci, in An Introduction to the Mathematics of Financial Derivatives, Academic Press, 2nd Edition, 2000.

Overview of this Lecture: Background Material

- Random variables
- Monte Carlo simulation
- Brownian motion

Continuous Random Variable, $\mathbf{X}$

Probability:

$$\mathbb{P}(a \leq \mathbf{X} \leq b) = \int_a^b f(x)\, dx$$

Expected Value (mean):

$$\mathbb{E}[\mathbf{X}] = \int_{-\infty}^{\infty} x f(x)\, dx$$

Variance:

$$\mathsf{var}(\mathbf{X}) = \mathbb{E}((\mathbf{X} - \mathbb{E}(\mathbf{X}))^2)$$

We can $+$, $-$, $\times$, $\div$ and apply functions to get new random variables: $\sin(\mathbf{X}), e^{\mathbf{X}+\mathbf{Y}^2}, \ldots$

Properties of Random Variables

$$\mathbb{E}[\mathbf{X} + \mathbf{Y}] = \mathbb{E}[\mathbf{X}] + \mathbb{E}[\mathbf{Y}]$$
$$\mathbb{E}[\alpha \mathbf{X}] = \alpha \mathbb{E}[\mathbf{X}]$$
$$\mathsf{var}[\alpha \mathbf{X}] = \alpha^2 \mathsf{var}[\mathbf{X}]$$

$\mathbf{X}$ & $\mathbf{Y}$ are independent $\iff$

$$\mathbb{E}[g(\mathbf{X})h(\mathbf{Y})] = \mathbb{E}[g(\mathbf{X})]\mathbb{E}[h(\mathbf{Y})], \quad \text{for all } g, h : \mathbb{R} \mapsto \mathbb{R}$$

Hence,

$$\mathbf{X} \ \& \ \mathbf{Y} \text{ indep. } \Rightarrow \begin{cases} \mathbb{E}[\mathbf{X}\mathbf{Y}] = \mathbb{E}[\mathbf{X}]\mathbb{E}[\mathbf{Y}] \\ \mathsf{var}[\mathbf{X} + \mathbf{Y}] = \mathsf{var}[\mathbf{X}] + \mathsf{var}[\mathbf{Y}] \end{cases}$$

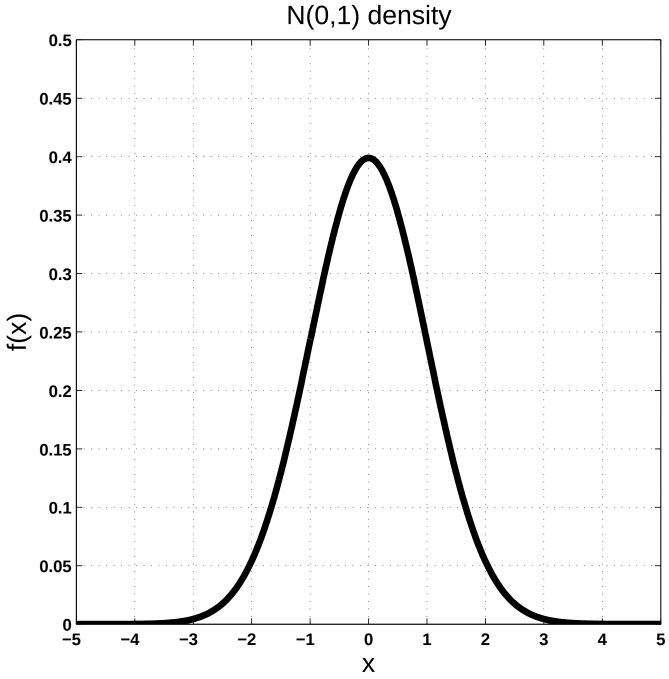
Normal Random Variables

Density

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Mean is μ, variance is σ^2

We write $\mathbf{X} \sim N(\mu, \sigma^2)$



Properties of Normal Random Variables

1. If $\mathbf{X} \sim N(\mu, \sigma^2)$ then $(\mathbf{X} - \mu)/\sigma \sim N(0, 1)$
2. If $\mathbf{Y} \sim N(0, 1)$ then $\sigma\mathbf{Y} + \mu \sim N(\mu, \sigma^2)$
3. If $\mathbf{X} \sim N(\mu_1, \sigma_1^2)$, $\mathbf{Y} \sim N(\mu_2, \sigma_2^2)$ and $\mathbf{X}$ and $\mathbf{Y}$ are independent, then $\mathbf{X} + \mathbf{Y} \sim N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$
4. If $\mathbf{X}$ and $\mathbf{Y}$ are normal random variables then $\mathbf{X}$ and $\mathbf{Y}$ are independent if and only if $\mathbb{E}[\mathbf{X}\mathbf{Y}] = \mathbb{E}[\mathbf{X}]\mathbb{E}[\mathbf{Y}]$

Central Limit Theorem

In words:

"**the sum of a large number of independent random variables (whatever their distribution!) behaves like a normal random variable**"

In maths:
Let $\{\mathbf{X}_i\}_{i\geq 1}$ be i.i.d. with mean a and variance b^2. Then for all $\alpha < \beta$

$$\lim_{M\to\infty} \mathbb{P}\left(\alpha \leq \frac{\sum_{i=1}^{M} \mathbf{X}_i - Ma}{\sqrt{M}b} \leq \beta\right) = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\beta} e^{-\frac{1}{2}x^2}\, dx$$

Pseudo-random Number Generation in MATLAB

`rand` for uniform (0,1), `randn` for N(0,1)

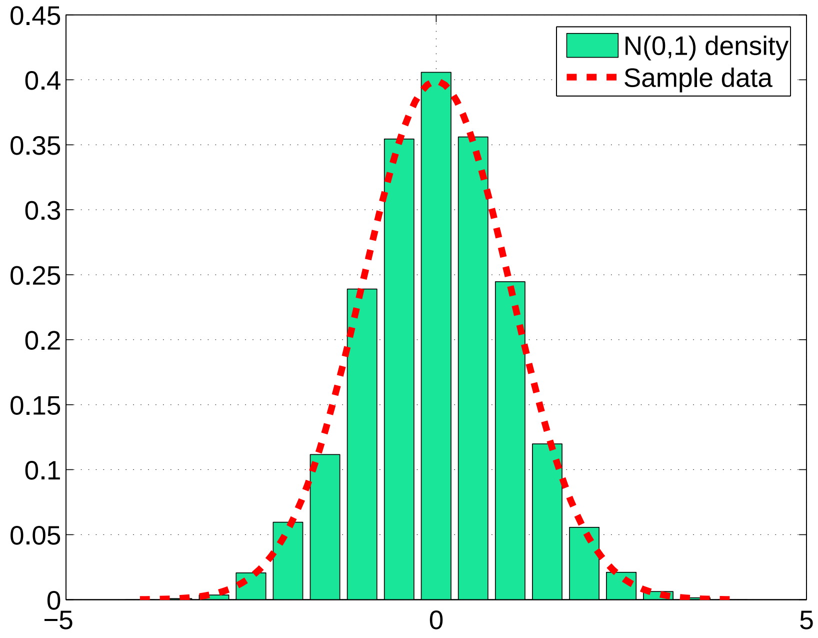
En masse, the samples are designed to have the appropriate statistical properties:

```
>> rand('state',100); randn('state',100)
>> [rand(10,1), randn(10,1)]
ans =
    0.3929    0.9085
    0.6398   -2.2207
    0.7245   -0.2391
    0.6953    0.0687
    0.9058   -2.0202
    0.9429   -0.3641
    0.6350   -0.0813
    0.1500   -1.9797
    0.4741    0.7882
    0.9663    0.7366
```

Illustration of CLT

10^4 samples of $Z = e^Y$, where Y is a Bernoulli r.v.
Shift and scale for CLT:

```
for k = 1:L
     Z = exp(double(rand(M,1)>(1-p)));
     S(k) = (sum(Z) - M*a)/(b*sqrt(M));
end
```



Monte Carlo Simulation

Suppose we can sample from a random variable $\mathbf{X}$.
Letting $a = \mathbb{E}[\mathbf{X}]$ and $b^2 = \text{var}[\mathbf{X}]$, we want to estimate a.

$$a_M := \frac{1}{M} \sum_{i=1}^{M} \xi_i \qquad \textbf{(sample mean)}$$

$$b_M^2 := \frac{1}{M-1} \sum_{i=1}^{M} (\xi_i - a_M)^2 \qquad \textbf{(sample variance)}$$

Then CLT $\Rightarrow$

$$\left[a_M - 1.96 \frac{b_M}{\sqrt{M}}, a_M + 1.96 \frac{b_M}{\sqrt{M}} \right]$$

is an approximate **95% confidence interval** for a.

Brownian Motion, $\mathbf{W}(t)$, over $0 \le t \le T$

1 $\mathbf{W}(0) = 0$ (with probability 1)

2 For $0 \le s < t \le T$,
$\mathbf{W}(t) - \mathbf{W}(s)$ is $N(0, t-s)$

3 For $0 \le s \le t \le u \le v \le T$,
$\mathbf{W}(v) - \mathbf{W}(u)$ & $\mathbf{W}(t) - \mathbf{W}(s)$ are indep.

Hence,

$$\mathbf{W}(t+\delta t) - \mathbf{W}(t) \;\text{ is }\; N(0, \delta t), \;\; \text{i.e.} \;\; \sqrt{\delta t}\, N(0,1)$$

Discretized Brownian Path

δt = T/L
$\mathbf{W}_0 = 0$
for i = 0 to L-1
    compute a N(0,1) sample ξ_i
    $\mathbf{W}_{i+1} = \mathbf{W}_i + \sqrt{\delta t}\,\xi_i$
end

Discretized Brownian Path

δt = T/L

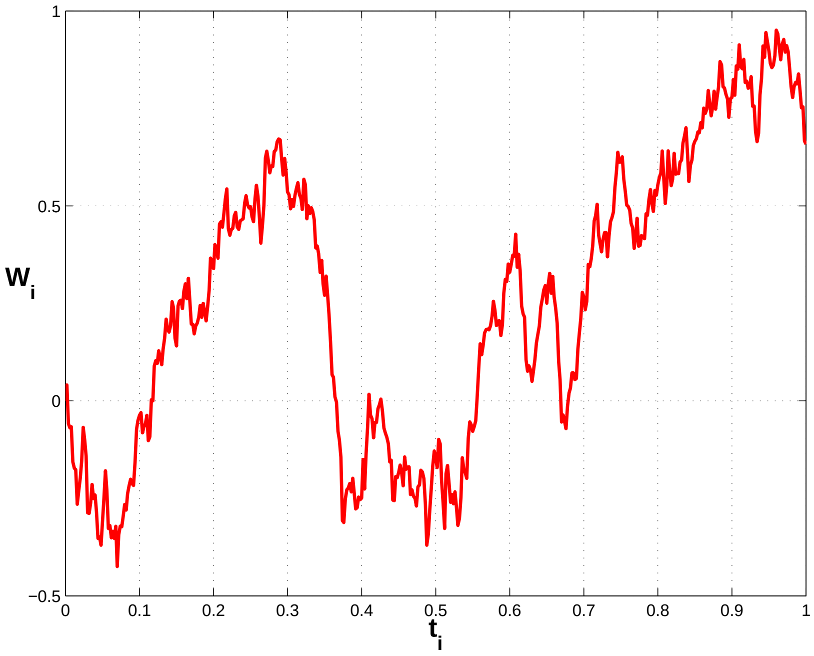
$\mathbf{W}_0 = 0$

for i = 0 to L-1

compute a N(0,1) sample ξ_i

$\mathbf{W}_{i+1} = \mathbf{W}_i + \sqrt{\delta t}\,\xi_i$

end

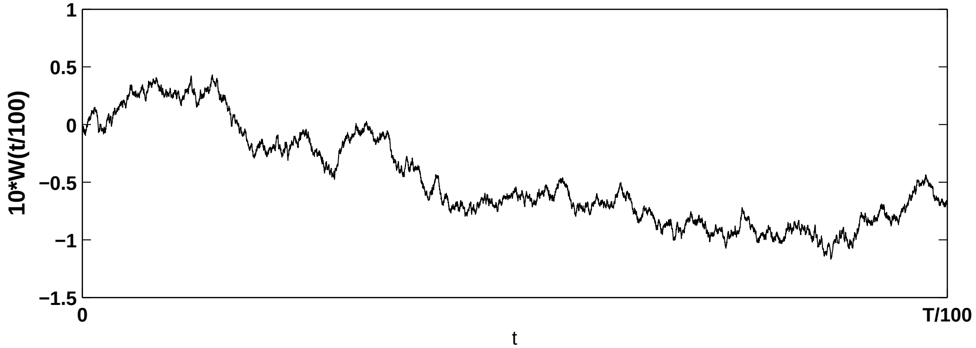


Scaling Property

For any fixed $c > 0$, if $\mathbf{W}(t)$ is an example of Brownian motion, then so is

$$\mathbf{V}(t) := \frac{1}{c}\mathbf{W}(c^2 t)$$

(Simple to check the three defining conditions.)
E.g., $c = 1/10$:



Non-differentiability

Scaling property can be used to show that

$$\mathbb{P}\left(\frac{|\mathbf{W}(1/n^4)|}{1/n^4} > n\right) = \mathbb{P}\left(|\mathbf{W}(1)| > \frac{1}{n}\right)$$

So, with prob. 1, $\mathbf{W}(t)$ is not differentiable at the origin.

In fact $\mathbf{W}(t)$ is **nowhere differentiable**.

Next lecture ...
we will see how to integrate with respect to $\mathbf{W}(t)$.

bpath1.m

```
%BPATH1  Brownian path simulation

clf
randn('state',100)               % set the state of randn
T = 1; N = 500; dt = T/N;
dW = zeros(1,N);                 % preallocate arrays ...
W = zeros(1,N);                  % for efficiency

dW(1) = sqrt(dt)*randn;          % first approx outside loop ...
W(1) = dW(1);                    % since W(0) = 0 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)
```

bpath2.m

```
%BPATH2  Brownian path simulation: vectorized

clf
randn('state',100)             % set the state of randn
T = 1; N = 500; dt = T/N;

dW = sqrt(dt)*randn(1,N);      % increments
W = cumsum(dW);                % cumulative sum

plot([0:dt:T],[0,W],'r-')      % plot W against t
xlabel('t','FontSize',16)
ylabel('W(t)','FontSize',16,'Rotation',0)
```