3. Stochastic Integration and ODEs

Session 14
Oct. 19, 2018

3.1 Brownian Motion

we had: binomial distribution $b(j,n,p) = \binom{n}{j} p^j (1-p)^{n-j}$

(n trials, j times "up", p = probability for "up")

center and rescale $x = \dfrac{j - \mathbb{E}(j)}{\sqrt{\mathrm{Var}(j)}}$

Central limit theorem (CLT): $\sqrt{\mathrm{Var}(j)}\, b(j,n,p) \xrightarrow{n\to\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$

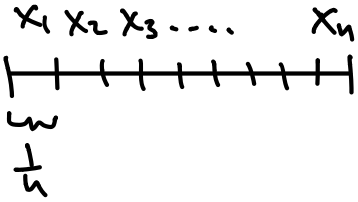
let X be the random variable at $T=1$ distributed according to scaled binomial distribution in the limit $n\to\infty$, i.e., according to normal dist. with mean 0 and variance 1.

X_1, X_2 same process and independent

we have $1 = \mathrm{Var}(X) = \mathrm{Var}(X_1 + X_2) \underset{\substack{\uparrow \\ \text{independence}}}{=} \mathrm{Var}(X_1) + \mathrm{Var}(X_2)$

$\underset{\substack{\nearrow \\ \text{same distribution}}}{=} 2\,\mathrm{Var}(X_1)$

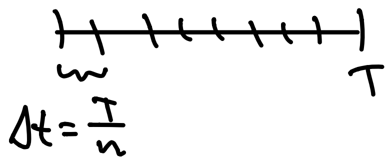
$\Rightarrow X_1$ distributed according to $\frac{1}{\sqrt{2}}\mathcal{N}(0,1)$ or $\mathcal{N}(0, \frac{1}{\sqrt{2}})$.



$$\Rightarrow 1 = Var(X) = n\,Var(X_1)$$

$$\Rightarrow X_i \sim \frac{1}{\sqrt{n}} \mathcal{N}(0,1) \quad \left(\sim \mathcal{N}(0, \tfrac{1}{\sqrt{n}})\right)$$

or, taking T into account:



and $X_i \sim \frac{1}{\sqrt{\Delta t}} \mathcal{N}(0,1)$

this motivates the following rigorous definition:

Def.: A stochastic process $t \mapsto W(t)$ for $t \in [0,\infty)$ is called Brownian Motion (BM) or Wiener process if:

a) $W(0) = 0$

b) each realization is continuous in t

c) for any $0 \le s_1 < s_2 < t_1 < t_2$ the increments

$W(s_2) - W(s_1)$ and $W(t_2) - W(t_1)$ are independent

d) $W(t_2) - W(t_1)$ is distributed like $\sqrt{t_2 - t_1}\,\mathcal{N}(0,1)$ for all $t_1 < t_2$

Note:
- BM is one example of a Markov process, i.e., the future development is independent of current value.
- BM not appropriate for stock price development, since parameters like the mean and the variance are missing, and BM can be negative

Geometric Brownian Motion: $S(t) = S(0) \cdot e^{(\mu - \frac{\sigma^2}{2})t + \sigma W(t)}$

(we check later what its mean and variance are)

Python implementation:

- BM: $W_0 = 0$

  $W_1 = \sqrt{\Delta t} \cdot$ sample from $\mathcal{N}(0,1)$

  $W_2 = W_1 + \sqrt{\Delta t} \cdot$ sample from $\mathcal{N}(0,1)$

  in python: dW = normal(0, 1, size=n) $\cdot \sqrt{\Delta t}$

  W = cumsum(dW) (cumulative sum)

  W = r_[0, W] (add time 0)

  $\left(\begin{array}{l} a = (\ldots), b = (\ldots) \\ r\_[a,b] = (\underbrace{\ldots}_{a}, \underbrace{\ldots}_{b}) \end{array} \right)$

- ensemble of BMs: M BM paths

  in python: dW = normal(0, 1, size=(M, N))

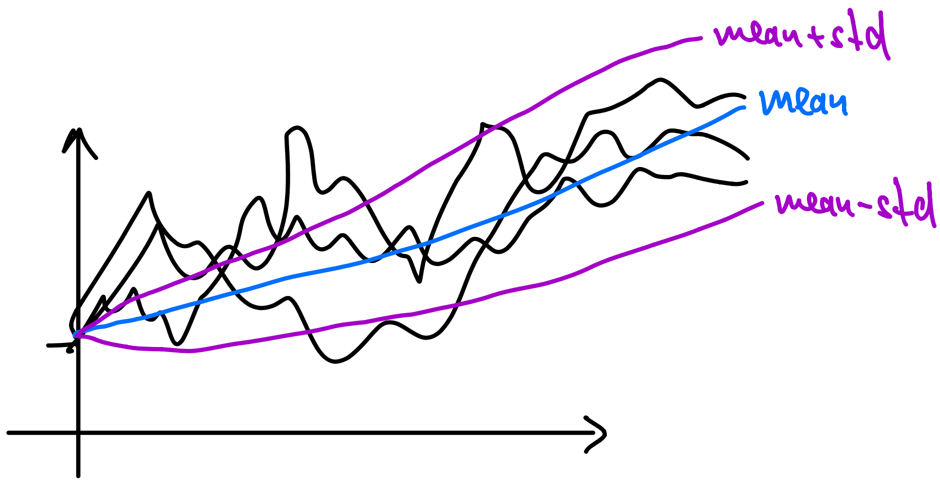
  M → # of samples

  N → # of timesteps

  W = cumsum(dW, axis=1)

  axis=1 → cumulative sum over row entries

  e.g.: mean(W, axis=0), std(W, axis=0) (i.e., over samples)



• compare with binomial tree, using $r=\mu$, $u=\frac{1}{d}$, $u=e^{\sigma\sqrt{\frac{T}{n}}}$,

$$p=p_u=\frac{e^{r\frac{T}{n}}-d}{u-d}$$

python: random_sample((M,N)) gives a matrix of uniform random samples of interval [0,1]

$$A=\text{random\_sample}((M,N))<p \implies \begin{array}{cccc} d & u & u & d \\ \swarrow & \swarrow & \swarrow & \swarrow \end{array} \begin{pmatrix} 0 & 1 & 1 & 0 & \dots \\ \cdot & \cdot & \cdot\cdot & \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

(use $d+(u-d)^A$)

1 with prob. p

• for GBM one could use

$$S(t)=S(0)\,\text{cumprod}\Big(\exp\big(\underbrace{dt}_{=\Delta t=\frac{t}{n}}(\mu-\tfrac{\sigma^2}{2})+\sigma\,dW\big)\Big)$$

• seed(k) for fixed k gives you same realizations

Monte-Carlo method:

random samplings to approximate expectation values

Ex.: binomial tree model for European call options:

$$C = \sum_{j=0}^{n} b(j,n,p) \underbrace{R^{-n} \max(0, S u^j d^{n-j} - X)}_{f(j,n)} = \mathbb{E}(f)$$

Monte-Carlo: m samples $j_1, \dots, j_m$ from $b(j,n,p)$ and

compute $\frac{1}{m} \sum_{k=1}^{m} f(j_k, n) \xrightarrow{m \to \infty} \mathbb{E}(f)$

(law of large numbers)

idea/hope: time efficient method, since $m \ll n$ to yield good results

convergence rate $\Big| \underbrace{\frac{1}{m} \sum_{k=1}^{m} f(j_k, n)}_{C_m} - C \Big| \sim A m^{-\beta}$

loglog plot: $\ln |C_m - C| \sim \ln A - \beta \ln m$

conv. rate

Problem 3: use geom. BM to generate m paths $S_i(t)$ and evaluate payoff