1 Misc

- $\bar{X} \sim \mathcal{N}(\mathbb{E}[X], \sigma_X^2/n)$; $\widehat{\sigma}_X = \sqrt{\sum(X_i - \bar{X})^2/(n-1)}$; $\widehat{se} = \widehat{\sigma}_X/\sqrt{n}$.

2 Random Number Generation

- **Prob Integral Transform:** The key is that $F_X(X) \sim U(0,1)$; $X = F_X^{-1}(F_X(X)) \overset{\mathrm{d}}{=} F_X^{-1}(U)$. $\mathrm{Exp}(\lambda): F^{-1}(u) = -\frac{1}{\lambda}\log(1-u)$.
- **Aliasing Table:** Generate random states (discrete, finite dist)

```
function MakeTable(pmf)                          // pmf: a list of {'x', 'p'} maps
    n ← len(pmf)                                 // table[i]: (state, p, alias) row
    pmf[:, 'p'] *= (n − 1)
    table ← zeroes((n − 1, 3))
    for i = 0 : n − 1 do
        pmf.sort(key = lambda t:t['p'])
        table[i, 0] ← pmf[0]['x'];   table[i, 2] ← pmf[−1]['x']
        table[i, 1] ← p ← pmf[0]['p']
        pmf[−1]['p'] -= (1 − p);   pmf.popfront()
    return table
function Draw(table, n)                          // draw sample of size n
    U ← uniform(size=n);   V ← (len(table) − 1)*U
    I ← ⌈V⌉;   W ← I − V
    Y ← (W ≤ table[I, 1])
    return table[I, 0].*Y + table[I, 2].*(1 − Y)  // elementwise .*
```

- **Rejection Method:** When F^{-1} hard to obtain. Idea is $f_X(x) = cf_Y(x)\frac{f_X(x)}{cf_Y(x)} = cf_Y(x)g(x)$. We must ensure $\mathrm{supp}(f_X) \subseteq \mathrm{supp}(f_Y)$; and choose c such that $g(x) \in [0,1]$ for all x. **Procedures:** (1) $Y = F_Y^{-1}(U)$. (2) An independent $V \sim U(0,1)$, reject Y if $V \le g(Y) = f_X(Y)/cf_Y(Y)$; otherwise return Y. **Efficiency:** $1/c$ is acceptance rate, we want to minimize it. Choose $\sup f_X(x)/f_Y(x)$.

3 Some Distributions

- **Mixture of Normals:** Generate $U \sim U(0,1)$, $Z \sim \mathcal{N}(0,1)$ independently. Return $X = 0.6Z$ if $U \le 0.82$; $X = 1.98Z$ otherwise. X has approximately variance 1 and heavy tails.
- **Generalized Λ:** Generate by prob integral transform: $F^{-1}(u) = \lambda_1 + \frac{1}{\lambda_2}(u^{\lambda_3} - (1-u^{\lambda_4}))$; $u \in [0,1]$. λ_1 is the center, λ_2 is scale parameter; λ_3, λ_4 fit the 3rd and 4th moment. Symmetric if $\lambda_3 = \lambda_4$.
- **Multivariate Normal:** Simulate multivariate normals $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{(d\times d)})$ by iid normals $\boldsymbol{Z}$. Positive semi-definite $\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^\top$ (eigen-decomp), or $\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^\top$ (Cholesky). $\boldsymbol{X} \overset{\mathrm{d}}{=} \boldsymbol{A}\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{D}^{1/2}\boldsymbol{Z}$. $\boldsymbol{A}$ lower triangular, $\boldsymbol{D}$ diagonal.
- **Multivariate t:** Univariate case $T_\nu \overset{\mathrm{d}}{=} Z/\sqrt{S/\nu}$, where Z is $\mathcal{N}(0,1)$, $S \sim \chi^2(\nu) \sim \Gamma(\frac{\nu}{2}, \frac{1}{2})$; Z, S independent. Note that $\Gamma(\frac{\nu}{2}, \frac{1}{2})$ is sum of $\frac{\nu}{2}$ iid $\mathrm{Exp}(1/2)$'s. ($\Gamma(1,\lambda) = \mathrm{Exp}(\lambda)$; $\Gamma(k,\lambda)$ is sum of k iid $\Gamma(1,\lambda)$'s) Multivariate case: $\boldsymbol{T} \sim t(\nu; \boldsymbol{0}, \boldsymbol{\Sigma}^{(d\times d)})$ (df ν, mean $\boldsymbol{0}$, covariance $\boldsymbol{\Sigma}$, all d components must have same df) **Procedures:** (1) $\boldsymbol{A} \leftarrow \mathtt{cholesky}(\boldsymbol{\Sigma})$. (2) $\boldsymbol{Z} \leftarrow \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$; $\boldsymbol{X} \leftarrow \boldsymbol{A}\boldsymbol{Z}$. (3) $S \leftarrow \chi^2(\nu)$. (4) $\boldsymbol{T} \leftarrow \boldsymbol{X}/\sqrt{S/\nu}$.
- **Copulas:** The key idea is to separate multivariate distribution $\boldsymbol{X}$ into individual marginals $\{F_{X_i}\}$, and the covariance $\boldsymbol{\Sigma}$.

```
function GaussianCopula(Σ^{d×d}, {F_i^{-1}(·)}_{i=1}^d)  // a list of inv cdfs
    A ← cholesky(Σ);   Z ← N(0, I);   Y ← AZ
    U ← normal.cdf(Y/√diag(Σ))                    // U_i = Φ(Y_i/σ_i)
    return {F_i^{-1}(U_i)}                        // X = (F_1^{-1}(U_1), ..., F_d^{-1}(U_d))
```

```
function TCopula(Σ^{d×d}, {F_i^{-1}(·)}_{i=1}^d)         // same A, Z, Y
    ...A, Z, Y...;   S ← ξ²(ν);   T ← Y/√(S/ν)
    U ← t.cdf(T/√diag(Σ), df = ν)                 // U_i = F_{T_ν}(T_i/σ_i)
    return {F_i^{-1}(U_i)}                        // X = (F_1^{-1}(U_1), ..., F_d^{-1}(U_d))
```

4 Path Generation

Suppose we do N periods, $\Delta = T/N$. $i = 1, 2, ..., N$.

- **GBM:** $S_{(i+1)\Delta}|S_{i\Delta} \overset{\mathrm{d}}{=} S_{i\Delta}\exp\{(r - .5\sigma^2)\Delta + \sigma\sqrt{\Delta}Z_{i+1}\}$; or $S_{i\Delta} \overset{\mathrm{d}}{=} S_0\exp\{(r - .5\sigma^2)i\Delta + \sigma\sqrt{\Delta}(Z_1 + ... + Z_i)\}$. **Multidimensional GBM:** $dS_k/S_k = \mu_k dt + \sum_{j=1}^d A_{kj}dW_j(t)$, $\boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{\Sigma}^{d\times d}$. Simulated by driver $\boldsymbol{Z}^{N\times d}$. For $k = 1, ..., d$: $S_{k,(i+1)\Delta}|S_{k,i\Delta} \overset{\mathrm{d}}{=} S_{k,i\Delta}\exp\{(\mu_k - .5\sigma_k^2)\Delta + \sqrt{\Delta}\sum_{j=1}^d A_{kj}Z_{i+1,j}\}$.
- **Vasicek (O-U):** $dr(t) = \alpha(b - r(t))dt + \sigma dW(t)$ is a mean-reversion model. b is long term mean interest rate, α is mean reversion speed. $r(t)|r(s) \overset{\mathrm{d}}{=} e^{-\alpha(t-s)}r(s) + b(1 - e^{-\alpha(t-s)}) + \sigma\sqrt{(1 - e^{-2\alpha(t-s)})Z/2\alpha}$ $r(t)|r(s) \xrightarrow{d} \mathcal{N}(b, \frac{\sigma^2}{2\alpha})$, as $t \to \infty$, has a stationary distribution.
- **CIR:** $dr(t) = \alpha(b - r(t)) + \sigma\sqrt{r(t)}dW(t)$, non-central χ^2 trans density. $\chi^2(\nu, \lambda) \overset{\mathrm{d}}{=} \sum_{i=1}^\nu \mathcal{N}(m_i, 1)$; $\lambda = \sum m_i^2$ non-centrality parameter. $r(t)|r(s) \overset{\mathrm{d}}{=} \frac{\sigma^2(1-e^{-\alpha(t-s)})}{4\alpha}\chi^2(\nu, \lambda)$; $\nu = \frac{4b\alpha}{\sigma^2}$; $\lambda = \frac{4\alpha e^{-\alpha(t-s)}}{\sigma^2(1-e^{-\alpha(t-s)})}r(s)$
- **Hull White:** $dS(t)/S(t) = r(t)dt + \sigma(t)dW(t)$. Option price has black-scholes solution: $\mathtt{bs}(S_0, K, R^*, \Sigma^*, T)$. $R^* = \frac{1}{T}\int_0^T r(t)dt$; $\Sigma^* = \sqrt{\int_0^T \sigma^2(t)dt/T} \leftarrow \sqrt{\sum_1^N \widehat{\sigma}^2(i\Delta)/N}$, can only simulate $\sigma(t)$.
- **Discretization:** Want to approximate SDE: $dS(t) = a(S(t), t)dt + b(S(t), t)dW(t)$.**Euler:** (first order) $\widehat{S}_{(i+1)\Delta}|\widehat{S}_{i\Delta} \overset{\mathrm{d}}{=} \widehat{S}_{i\Delta} + a(\widehat{S}_{i\Delta}, i\Delta)\Delta + b(\widehat{S}_{i\Delta}, i\Delta)\sqrt{\Delta}Z_{i+1}$; assume constant a, b over $[i\Delta, (i+1)\Delta]$; error $\sim \mathcal{O}(\Delta)$. **Milstein:** (quadratic) $\widehat{S}_{(i+1)\Delta}|\widehat{S}_{i\Delta} \overset{\mathrm{d}}{=} \widehat{S}_{i\Delta} + a(\widehat{S}_{i\Delta}, i\Delta)\Delta + b(\widehat{S}_{i\Delta}, i\Delta)\sqrt{\Delta}Z_{i+1} + \frac{1}{2}b(\widehat{S}_{i\Delta}, i\Delta)\frac{\partial b(\widehat{S}_{i\Delta}, i\Delta)}{\partial \widehat{S}_{i\Delta}}(Z_{i+1}^2 - 1)\Delta$; error $\sim \mathcal{O}(\Delta^2)$. **Disc factor:** $D(T) = e^{-\int_0^T r(t)dt} \leftarrow e^{-\Delta\sum_1^N r(i\Delta)}$.

5 Variance Reduction

- **Antithetic Variables:** Use negatively correlated paths: simulating the second half of sample using the negative of $\boldsymbol{U}, \boldsymbol{Z}$'s that drive the first half. $\boldsymbol{U}^A = \boldsymbol{1} - \boldsymbol{U}$, $\boldsymbol{Z}^A = -\boldsymbol{Z}$. Final sample is $V_i \leftarrow (C_i + C_i^A)/2$.
- **Control Variables:** We want to estimate $\mathbb{E}[Y]$ with $\bar{Y}$. We also calculate $\bar{X}$ with the same paths, but $\mathbb{E}[X]$ is **known**. Adjust the estimate to $\widehat{Y} = \bar{Y} + \widehat{a}(\bar{X} - \mathbb{E}[X])$, $\widehat{a} = -\sigma_{XY}/\sigma_X^2 = -\rho_{XY}\sigma_Y/\sigma_X$ is chosen to minimize $\mathbb{V}\mathrm{ar}[\widehat{Y}]$, but can only be estimated itself: $\widehat{\widehat{a}} = -\widehat{\rho}_{XY}\widehat{\sigma}_Y/\widehat{\sigma}_X$. $\widehat{se} = \widehat{\sigma}_Y\sqrt{1 - \widehat{\rho}^2}/\sqrt{n}$. Note that $\mathbb{E}[\widehat{Y}(\widehat{a})] = \mathbb{E}[Y] \neq \mathbb{E}[\widehat{Y}(\widehat{\widehat{a}})]$, the resulting estimator is biased.

5.1 Importance Sampling

$\mathbb{E}[e^{-rT}h(\boldsymbol{X})] = \int e^{-rT}h(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} = \int e^{-rT}h(\boldsymbol{x})\frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{f_{\boldsymbol{Y}}(\boldsymbol{x})}f_{\boldsymbol{Y}}(\boldsymbol{x})d\boldsymbol{x}$, $\mathrm{supp}(hf_{\boldsymbol{X}}) \subseteq \mathrm{supp}(f_{\boldsymbol{Y}})$; absolute continuity. Old payoff is h, new payoff is $hf_{\boldsymbol{X}}/f_{\boldsymbol{Y}}$, new sampling distribution is $f_{\boldsymbol{Y}}$.

- **Change Mean:** Sample shifted normals $X \sim \mathcal{N}(m, 1) \overset{\mathrm{d}}{=} Z + m$; RN derivative is $f_Z/f_X = \exp(-mx + .5m^2)$; only generate final state for non-path-dependent options, $S_T = S_0\exp((r - .5\sigma^2)T + \sigma\sqrt{T}(Z + m))$; new discounted payoff is $e^{-rT}h(S_T)\exp(-m(Z + m) + .5m^2)$.
- **Enforcing Moneyness:** Call itm: $Z > \frac{\log(K/S_0) - (r - .5\sigma^2)T}{\sigma\sqrt{T}} =: L$ Generate X: $f_X(x) = \frac{\phi(x)}{(1-\Phi(L))}, x \ge L$; 0 otherwise. This is truncated normal, cdf $F_X(x) = \frac{\Phi(x) - \Phi(L)}{1 - \Phi(L)}, x \ge L$; 0 otherwise. Simulated by $X \overset{\mathrm{d}}{=} \Phi^{-1}\{U(1 - \Phi(L)) + \Phi(L)\}$. RN derivative is $f_Z/f_X = 1 - \Phi(L)$; hence new discounted payoff is $e^{-rT}h(S_T)(1 - \Phi(L))$.
- **Multiple Change of Measure:** For path-dep derivatives, we can't only sample the final S_T, have to do entire path $(S_\Delta, ..., S_{N\Delta})$ with driver $(Z_1, ..., Z_N)$. If apply change of measure to every single Z_i, the RN derivarive is a product $\prod_{i=1}^N f_{Z_i}(x)/f_{X_i}(x)$. E.g. $X_i \sim \mathcal{N}(m_i, 1)$ indep., RN derivative is $\prod_{i=1}^N \phi(x)/\phi(x - m_i)$.
- **Capriotti's Method:** We want to change sampling distribution from $Z \sim \mathcal{P}_0$ to parametric family $X \sim \mathcal{P}_{\boldsymbol{\theta}}$. (e.g., from $\mathcal{N}(0,1)$ to $\mathcal{N}(\mu, s^2)$), and want to find best $\boldsymbol{\theta}$ to reduce variance. We want to estimate $\mathbb{E}_0[h(\boldsymbol{Z})] = \mathbb{E}_{\boldsymbol{\theta}}[h(\boldsymbol{X})W_{\boldsymbol{\theta}}(\boldsymbol{X})]$; RN derivative $W_{\boldsymbol{\theta}} = d\mathcal{P}_0/d\mathcal{P}_{\boldsymbol{\theta}}$. MC estimator with M samples is $\frac{1}{M}\sum_{i=1}^M h(\boldsymbol{X}_i)W_{\boldsymbol{\theta}}(\boldsymbol{X}_i)$. Min variance $\iff$ Min second moment $\mathbb{E}_{\boldsymbol{\theta}}[h^2(\boldsymbol{X})W_{\boldsymbol{\theta}}^2(\boldsymbol{X})] = \mathbb{E}_0[h^2(\boldsymbol{Z})W_{\boldsymbol{\theta}}(\boldsymbol{Z})] \leftarrow \frac{1}{M}\sum_{i=1}^M h^2(\boldsymbol{Z}_i)W_{\boldsymbol{\theta}}(\boldsymbol{Z}_i)$, can be regarded as RSS of regression $0 = y = h(\boldsymbol{Z})\sqrt{W_{\boldsymbol{\theta}}(\boldsymbol{Z})} + \epsilon$. Find best $\boldsymbol{\theta}$ by fitting regression. For $\mathcal{N}(\theta, 1)$, $W_\theta(x) = \exp(-\theta x + .5\theta^2)$. For $\mathcal{N}(\mu, s^2)$, $W_\theta(x) = s\cdot\exp\{-.5(x^2 - (\frac{x-\mu}{s})^2)\}$.

```
function CapriottiMC(S_0, K, T, σ, r, N, M, n)
    Δ ← T/N;   Z ← N(0, 1, size = M)               // Fit with M ≪ n paths
    function Residual(params)                      // params = (μ, s)
        S_T ← S_0 exp((r − .5σ²)T + σ√T Z)
        W ← params[1] · exp{−.5(Z.² − ((Z−params[0])/params[1]).²)}
        return (S_T − K)⁺ .* √W
    μ*, s* ← least.squares(Residual, θ_0);         // method='lm'
    X ← N(μ*, s*², size = n)                       // n samples
    S_T ← S_0 exp((r − .5σ²)T + σ√T X)
    W ← s* · exp{−.5(X.² − ((X−μ*)/s*).²)}
    C ← e^{−rT}(S_T − K)⁺ .* W
    mean ← sum(C)/n;   se ← √(sum((C − mean)²)/(n(n − 1)))
    return C, mean, se
```

5.2 Conditional Monte Carlo

$\mathbb{V}\mathrm{ar}[X] = \mathbb{E}[\mathbb{V}\mathrm{ar}[X|Y]] + \mathbb{V}\mathrm{ar}[\mathbb{E}[X|Y]] > \mathbb{V}\mathrm{ar}[\mathbb{E}[X|Y]]$. As we proceed along a path, assign closed-form price as soon as there is one conditional on the current state (e.g. knocked in becomes vanilla).

```
function ConditionalKnockIn(S_0, K, H, T, m, σ, r, N, n)
    Δ ← T/N;   C ← zeros(n);   X ← N(m, 1, size = n)
    S ← gbm.paths(X, S_0, σ, r, T, N)              // with importance sampling
    for i, S_i in enumerate(S) do
        τ_i ← find.first(S_i, cond=lambda s : s < H, default= − 1)
        C[i] ← bs(S_i[τ_i], K, r, σ, T − τ_iΔ) if τ_i ≥ 0 else 0
        C[i] *= Π_1^{τ_i} exp(−mX[i] + .5m²)       // RN derivative
    return C, mean(C), se(C)
```

5.3 Stratification

The lowest driver of all randomness is $\boldsymbol{U}$, so $C_i = g(U_i)$ with some function g. $\mathbb{E}[g(U)] = \int_{[0,1]} g(u)du$. Break unit interval into B bins, $N_B = n/B$ iid uniform samples from each bin $[\frac{i-1}{B}, \frac{i}{B})$. Variance are reduced, the more bins the better.

- **One Dimension:** simulate $\boldsymbol{U}^{B\times N_B}$, each row is a bin. Get prices $C_{ij} = g(U_{ij})$. Bin average $\bar{C}_i = \sum_{j=1}^{N_B} C_{ij}/N_B$. Final estimate $\widehat{C}_{strat} = \frac{1}{n}\sum_{i=1}^{B}\sum_{j=1}^{N_B} C_{ij}$; $\widehat{se}(\widehat{C}_{strat}) = \frac{1}{\sqrt{n}}\sqrt{\frac{1}{B}\sum_{i=1}^{B}\widehat{\sigma}_i^2}$; $\widehat{\sigma}_i^2 = \mathbb{V}\mathrm{ar}[C_{ij}]$ is the sample variance of each bin $\boldsymbol{C}_i$.
- **Multidimensional:** option may rely on d assets. $\mathbb{E}[g(U_1, ..., U_d)] = \int_{[0,1]^d} g(u_1, ..., u_d)du_1...du_d$ (d-folds integral). We can stratify to d-hypercube. 2-assets example: $B = B_1B_2$, $n = B_1B_2N_B$. $\widehat{C}_{strat} = \frac{1}{n}\sum_1^{B_1}\sum_1^{B_2}\sum_{j=1}^{N_B} C_{i_1i_2j}$; $\widehat{se}(\widehat{C}_{strat}) = \frac{1}{\sqrt{n}}\sqrt{\frac{1}{B}\sum_1^{B_1}\sum_1^{B_2}\widehat{\sigma}_{i_1i_2}^2}$. Also for path-dep options, N periods will incur N-dimensions.

```
function Stratification2D(S0, K, σ, r, T, B1, B2, N_B, N_steps)
    bins1 ← linspace(0, 1, B1+1);   bins2 ← linspace(0, 1, B2+1)
    C ← zeros((B1, B2));   Σ ← zeros((B1, B2));   n ← B1 B2 N_B
    for i1 = 1 : B1 do
        for i2 = 1 : B2 do
            U1 ← uniform(bins1[i1], bins1[i1 + 1], size=N_B)
            U2 ← uniform(bins2[i2], bins2[i2 + 1], size=N_B)
            Z1, Z2 ← Φ^{-1}(U1), Φ^{-1}(U2)
            S1, S2 ← gbm.paths((Z1, Z2), S0, σ, r, T, N_steps)
            sample ← e^{-rT} H(S1, S2)
            C[i1, i2] ← mean(sample);   Σ[i2, i2] ← var(sample)
    return mean(C), sqrt(sum(Σ)/(n B1 B2))
function Projection(S0, K, ν^{d×1}, σ, r, T, B, N_B, N_steps)
    bins ← linspace(0, 1, B + 1);   n ← B N_B
    C ← zeros(B);   Σ ← zeros(B);
    for i = 1 : B do
        U ← uniform(bins[i], bins[i + 1], size=N_B)
        X ← Φ^{-1}(U).reshape(N_B, 1)
        Z ← N(0, I − νν^⊤, size=N_B)^{N_B×d} + Xν^⊤
        S1, ..., Sd ← gbm.paths(Z, S0, σ, r, T, N_steps)
        sample ← e^{-rT} H(S1, ..., Sd)
        C[i] ← mean(sample);   Σ[i] ← var(sample)
    return C, mean(C), sqrt(sum(Σ)/(nB))
```

- **Unequal Bins:** partition $\mathbb{R} = \cup_{i=1}^{B} A_i$. Want to estimate $\mathbb{E}[Y] = \mathbb{E}[g(U)]$; let $p_i = \mathbb{P}(Y \in A_i), \sigma_i^2 = \mathbb{V}\mathrm{ar}[Y|Y \in A_i]$. Sample $n_i = nq_i$ from the i-th bin, $\widehat{Y}_{strat} = \sum p_i \frac{1}{n_i}\sum_{j=1}^{n_i} Y_{ij} = \frac{1}{n}\sum_{i=1}^{B}\frac{p_i}{q_i}\sum_{j=1}^{n_i} Y_{ij}$. $\mathbb{V}\mathrm{ar}[\widehat{Y}_{strat}] = \frac{1}{n}\sum_{i=1}^{B}\frac{(p_i\sigma_i)^2}{q_i}$. Find q_i to minimize variance: $q_i^* = p_i\sigma_i/\sum_{j=1}^{B} p_j\sigma_j$.
- **Projection:** Suppose the d-dimensional driver is $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$; we stratify linear combination $X = \boldsymbol{\nu}^\top\boldsymbol{Z}, \|\boldsymbol{\nu}\| = 1$ to make X std.normal. Simulate $X = \Phi^{-1}(U)$, then get $\boldsymbol{Z}$ from conditional distribution $(\boldsymbol{Z}|X = x) \sim \mathcal{N}(x\boldsymbol{\nu}, \boldsymbol{I}_d - \boldsymbol{\nu}\boldsymbol{\nu}^\top)$. The bin of $\boldsymbol{Z}$ corresponds to that of X. Obtain $\widehat{C}_{strat}$ by the d-dim formula with driver $\boldsymbol{Z}$.

6 Simulating Greeks

- **Finite Difference:** $\widehat{\Delta} = (\widehat{C}(S_0 + h) - \widehat{C}(S_0))/h$. Bias$[\widehat{\Delta}] \sim \mathcal{O}(h)$ (taylor expansion), can be reduced by centering $\widehat{\Delta} = (\widehat{C}(S_0 + h) - \widehat{C}(S_0 - h))/2h$, Bias $\sim \mathcal{O}(h^2)$. Called **Resimulation:** use same driver $\boldsymbol{Z}$ to produce both $\widehat{C}(S_0 + h)$ and $\widehat{C}(S_0)$. $\mathbb{V}\mathrm{ar}[\widehat{\Delta}] = \frac{K}{nh^2}$, cross term will reduce the constant K.
- **Pathwise Differentiation:** $\Delta = \frac{\partial}{\partial S_0}\widetilde{\mathbb{E}}[e^{-rT}h(S_T)]$ can interchange expectation and derivative under some conditions. For Eu call: $\Delta = \widetilde{\mathbb{E}}[e^{-rT}\mathbb{1}_{\{S_T\geq K\}}\frac{S_T}{S_0}]$.
Vega $= \widetilde{\mathbb{E}}[e^{-rT}\mathbb{1}_{\{S_T\geq K\}}\frac{S_T}{\sigma}(\log\frac{S_T}{S_0} - (r - d + .5\sigma^2)T)]$; d: dividend.
- **Log-likelihood Method:** Still want to interchange expectation and differential, but this time apply diff to the transition density $f_{S_T|S_0}(s)$: $\Delta = \int_0^\infty e^{-rT}h(S_T)\frac{\partial}{\partial S_0}f_{S_T|S_0}(s)ds = \int_0^\infty e^{-rT}h(S_T)\frac{\partial}{\partial S_0}\log(f_{S_T|S_0}(s))f_{S_T|S_0}(s)ds = \widetilde{\mathbb{E}}[e^{-rT}h(S_T)\frac{\partial}{\partial S_0}\log(f_{S_T|S_0}(S_T))]$, i.e. multiply payoff by the partial derivative of log likelihood. GBM transition density $f_{S_T|S_0}(x) = N'(d_-(x))/(x\sigma\sqrt{T})$; $d_-(x) = [\log(x/S_0) - (r - .5\sigma^2)T]/(\sigma\sqrt{T})$; $\log f_{S_T|S_0}(x) = -\frac{1}{2}d_-(x)^2 - \log(x\sigma\sqrt{T})$. Under GBM:
$\Delta = \widetilde{\mathbb{E}}[e^{-rT}h(S_T)\frac{1}{S_0\sigma^2T}(\log(\frac{S_T}{S_0}) - (r - .5\sigma^2)T)]$;
Vega $= \widetilde{\mathbb{E}}[e^{-rT}h(S_T)[-\sqrt{T}d_-(S_T)(1 - \frac{d_-(S_T)}{\sigma\sqrt{T}}) - \frac{1}{\sigma}]]$

7 BDZ (Lookback/Barrier Options)

The Euler scheme knows only the gird $\{S_{i\Delta}\}$, but knows nothing about intervals in between: $S_t \in (S_{(i+1)\Delta}, S_{i\Delta})$. Max/min on the grid is a biased estimator to the max/min along the whole path.

- **Brownian Bridge:** Suppose $dS_t = rS_tdt + \sigma(t, S_t)dW_t$. Model $S_t \in (S_{(i+1)\Delta}, S_{i\Delta})$ as $S_t = S_{i\Delta} + \sigma(i\Delta, S_{i\Delta})B_t$, $t \in [i\Delta, (i+1)\Delta]$. B_t is a brownian bridge process, i.e. BM with fixed endpoints.
$B_t \overset{\mathrm{d}}{=} W_t|\{W_{i\Delta} = 0, W_{(i+1)\Delta} = (S_{(i+1)\Delta} - S_{i\Delta})/\sigma(i\Delta, S_{i\Delta}) =: b\}$
- **Max/Min Brownian Bridge:** We care about max/min of S_t, it suffices to find $\max_{[i\Delta,(i+1)\Delta]} B_t \overset{\mathrm{d}}{=} \max_{[i\Delta,(i+1)\Delta]} W_t|\{W_{i\Delta} = 0, W_{(i+1)\Delta} = b\} \overset{\mathrm{d}}{=} \max_{[0,\Delta]} W_t|\{W_\Delta = b\}$. Clearly $\mathbb{P}(\max_{[0,\Delta]} W_t > x|W_\Delta = b) = 1$ if $x \leq \max(0, b)$. Otherwise Bayes formula: $\mathbb{P}(\max_{[0,\Delta]} W_t > x|W_\Delta = b) = \frac{\mathbb{P}(\{\max_{[0,\Delta]} W_t > x\}\cap\{W_\Delta = b\})}{\mathbb{P}(W_\Delta = b)} = \frac{\mathbb{P}(\{\max_{[0,\Delta]} W_t > x\}\cap\{W_\Delta = 2x - b\})}{\mathbb{P}(W_\Delta = b)}$ (reflection principle) $= \frac{\mathbb{P}(W_\Delta = 2x - b)}{\mathbb{P}(W_\Delta = b)}$ (since $x > b$) $= \exp(-2x(x - b)/\Delta)$. $F_{\max B_t}(x) = 1 - e^{-2x(x-b)/\Delta}$. To simulate $M = \max B_t$ and $m = \min B_t$, use
$M \overset{\mathrm{d}}{=} (b + \sqrt{b^2 - 2\Delta\log(1 - U)})/2$, $m \overset{\mathrm{d}}{=} (b - \sqrt{b^2 - 2\Delta\log(1 - U)})/2$.

```
function BDZPath(S0, K, T, σ, r, N)                  // Generate one path
    S ← zeros(N);  M ← zeros(N);  Δ ← T/N;  last ← S0
    for j = 1 : N do
        S[j] ← last * (1 + rΔ + σ√Δ * normal(0, 1))
        b ← (S[j] − last)/(σ * last)
        M[j] ← last + σ * last * 1/2 [b + sqrt(b^2 − 2Δ log(uniform(0, 1)))]
    return S, M
function LookbackCall(S0, K, T, σ, r, N, n) // pay (max S − K)^+
    C ← zeros(n)
    for i = 1 : n do
        S, M ← BDZPath(S0, K, T, σ, r, N)
        C[i] ← e^{-rT}(max(M) − K)^+
    return C, mean(C), std(C)/√n
```

8 Longstaff-Schwartz (American Options)

- **Data Structures:** $cashflow, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\tau}, pathin, \boldsymbol{C}$.

```
function LaguerrePoly(S_t, X)             // S_t: all paths at t = iΔ
    X[:, 1] ← exp(−S_t./2);   X[:, 2] ← X[:, 1]. * (1 − S_t)
    X[:, 3] ← X[:, 1]. * (1 − 2S_t + .5S_t.^2);   X[:, 0] ← 1
    return X
function LongstaffSchwartz(S0, K, T, σ, r, N, n)
    S ← gbm.paths(X, S0, σ, r, T, N)               // n paths, N periods
    cashflow ← zeros((N, n));   τ ← (N − 1) * ones(n)
    X, y ← zeros((n, 4)), zeros((n, 1))
    C ← zeros(n);   pathin ← ones(n, type=bool)
    for i = 1 : N do
        pathin ← (K − S[i, :] > 0)   // y[j] = pathin[j] * cf[τ[j], j]
        y ← pathin * cashflow[τ, range(n)]
        X ← apply.along.column(operator*,
                LaguerrePoly(S[i, :]/K, X), pathin)
                           // If pathin[j] = 0, the j-th row X[j, :] = 0
        ŷ ← X(X^⊤X)^{-1}X^⊤y
        exec ← (ŷ < (K − S[i, :])^+)                // Early exercise at i
        τ[exec] ← i
    C ← max(e^{-rτΔ} · cashflow[τ, range(n)], (K − S0)^+)
    return C, mean(C), std(C)/√n
```

9 Credit Default Swap

- **Defaultable Bond:** face 1\$ maturity T bond with default rate π; it recovers R when default. Price of bond is $e^{-rT}(\pi R + (1 - \pi))$, let X be time to default, $\pi = \mathbb{P}(X \leq T)$, if X modeled as $\mathrm{Exp}(\lambda)$, $\pi = 1 - e^{\lambda T}$. **Credit spread** is the defaultable bond risk premium: $e^{-st} = \pi R + (1 - \pi)$. So $\lambda = -\frac{1}{T}\log((e^{-sT} - R)/(1 - R)) \approx s/(1 - R)$.
- **CDS Pricing:** there are d defaultable bonds. k-th to default swap (kTD, $k = 1, ..., d$) pays $1 - R$ if k or more than k bonds default. p_k: prob of no less than k bonds default, Price of kTD is $e^{-rT}(1 - R)p_k$. $\widehat{se} = e^{-rt}(1 - R)\sqrt{\widehat{p}_k(1 - \widehat{p}_k)/n}$. $\widehat{p}_k$ simulated with default times $\boldsymbol{X} \sim \mathrm{Copulas}(\{F_{X_i}^{-1}(\cdot)\}, \boldsymbol{\Sigma})$ with given marginals and covariance matrix.

10 Credit Rating Transition Model

- **Settings:** The rating state process $\{X(t)\}$. τ is time between transitions. $\boldsymbol{P} = \{p_{ij}\}$ instantaneous transition matrix. $p_{ij} = \mathbb{P}(X(\tau + dt) = j|X(\tau) = i)$. $\boldsymbol{\Pi}(T) = \{\pi_{ij}(T)\}$ final state probabilities after time T; $\pi_{ij} = \mathbb{P}(X(T) = j|X(0) = i)$. If $\tau \sim \mathrm{Exp}(\lambda)$, $X(t)$ is a Markov chain, we have $\boldsymbol{\Pi}(T) = \exp(\boldsymbol{Q}T)$, $\boldsymbol{Q}(\boldsymbol{P}, \lambda)$ is transition rate matrix. If not exponential (semi-Markov case), no closed form solution for $\boldsymbol{\Pi}(T)$.

```
function RatingTrans(P, T, M_τ, n, d)                  // M_τ is τ's sampler
    Π ← zeros((d, d)) // table[i] is sampler for X(τ + dt)|X(τ) = i
    tables ← [MakeTable(zip(range(d), P[i])) for i in range(d)]
    for x0 in range(d) do                       // simulate x0-th row of Π
        for i = 1 : n do                                  // draw n samples
            t, x ← 0, x0
            while t ≤ T do
                τ ← M_τ(x);   t += τ
                if t > T then break while   // transition exceeds T
                x ← Draw(tables[x])                  // sample next state
            Π[x0, x] += 1             // while ends, x is the final state
    Π ./= n;   se ← sqrt(Pi(1 − Pi)/n)
    return Π, se
```

Author: Ze Yang (zey@andrew.cmu.edu)