Class SinglePath

Members
➢ protected ▪ vector< double > Values ▪ double StartTime ▪ double EndTime ▪ size_t NbSteps

Methods
▪ SinglePath (double start, double end, size_t nbSteps) ▪ void InsertValue (double val) ▪ double GetState (double time) ▪ vector<double> GetAllValues ( )

Class RandomProcess

Members
➢ protected ▪ RandomGenerator* Generator ▪ vector<SinglePath*> Paths ▪ int Dimension

Methods
▪ RandomProcess (RandomGenerator* Gen, int dim) ▪ virtual void Simulate (double startTime, double endTime, size_t nbSteps) = 0 ▪ SinglePath* GetPath (int dimension = 0)

Class Brownian1D : public RandomProcess

Members	Methods
(no member)	▪ Brownian1D (RandomGenerator* Gen) ▪ void Simulate (double startTime, double endTime, size_t nbSteps)

Class BrownianND : public RandomProcess

Members	Methods
➢ protected ▪ vector < vector<double> >* CorrelationMatrix	▪ BrownianND (RandomGenerator* Gen , int dim, vector < vector<double> >* Corr) ▪ void Simulate (double startTime, double endTime, size_t nbSteps)

Class BlackScholes1D : public RandomProcess

Members
➢ protected ▪ double Spot ▪ double Rate ▪ double Vol

Methods
▪ BlackScholes (RandomGenerator* Gen, double spot, double rate, double vol)

class BSEuler1D : public BlackScholes1D	
Members	Methods
(no member)	▪ BSEuler1D (RandomGenerator* Gen, double spot, double rate, double vol) ▪ void Simulate (double startTime, double endTime, size_t nbSteps)
class BSMilstein1D : public BlackScholes1D	
Members	Methods
(no member)	▪ BSMilstein1D (RandomGenerator* Gen, double spot, double rate, double vol) ▪ void Simulate (double startTime, double endTime, size_t nbSteps)

Class BlackScholes2D : public RandomProcess

Members
➢ protected ▪ double Spot1 , double Spot2 ▪ double Rate1 , double Rate2 ▪ double Vol1 , double Vol2 ▪ double Rho

Methods
▪ BlackScholes2D (RandomGenerator* Gen, double spot1, double spot2, double rate1, double rate2, double vol1, double vol2, double rho)

class BSMilstein2D : public BlackScholes2D	
Members	Methods
(no member)	▪ BSMilstein2D (RandomGenerator* Gen, double spot1, double spot2, double rate1, double rate2, double vol1, double vol2, double rho) ▪ void Simulate (double startTime, double endTime, size_t nbSteps)

Heston Process

The Heston process is defined as follows

$$\begin{cases} dS_t &= \mu S_t + \sqrt{V_t} S_t dW_t^S \\ dV_t &= \kappa(\theta - V_t)\, dt + \sigma \sqrt{V_t} dW_t^V \end{cases}$$

Where

- The two brownians are correlated : $d < W^S, W^V >_t = \rho dt$
- μ is the rate of return of the asset
- θ is the long run variance
- κ is the rate at which V_t reverts to θ
- σ is the volatility of the volatility

Class Heston : public RandomProcess

Members

- protected
- double Spot
- double InitVariance
- double Mu
- double Theta
- double Kappa
- double Sigma
- double Rho

Methods

- Heston (RandomGenerator* Gen,
  double Spot,
  double InitVariance,
  double Mu,
  double Theta,
  double Kappa,
  double Sigma,
  double Rho)

- void Simulate (double startTime, double endTime, size_t nbSteps)