Financial Computing (with C++) 1

Instructor : Dmitry Kramkou TA : Gerard Brunick Communication: Blackboard Materials: (a) cfl library + Examples (b) becture Motes (c) Optional : Books on (c++ (with STL library) (i) Stroustrup (ii) Josuttis

2 Requirements: 1. Notebook with MSVisual Studie . Net 2. "Basic" Knowledge of relevent areas: a) C++ B) Stochastic Analysis c) Finance (Derivatives), Arbitrage - Free Pricing) numerical Anglysis d) Course WORK! Exam Home WORKS # 4 (best 3) 3 grade 50% 50% Glabs YES Ν

| Results fro years : |) h2 | prei | liou | 5 |
|-------------------------|------|------|------|----|
| YEAR 200 # of stude | ent | = 2 | 33 | |
| # of solved problems | 3 | 2 | 1 | 0 |
| # of students | 3 | 9 | 9 | 12 |

| YEAR 2003 # of students = 55 | | | | | |
|---------------------------------|---|----|----|----|--|
| # of solved problems | | | | | |
| # of students | 3 | 22 | 12 | 16 | |

Goals of the course : A: "Theoretical" - review and expand your knowledge of 1) OOP with C++ 2) Mathematical Finance 3) stochastic Calculus 4) numerical Auglysis B: "Practical" - improve (and test!) your ability to use C++ for financial computations.

5 Method of study: A: "Theoretical" - discuss design and implementation of a C++ library for pricing of derivatives (case study = cfl bibrary) B: "Practical" - use off library for pricing of concrete (quite compli-cated!) derivatives.

Q: What item is mere 4 important for practical implementations? 1. Type of underlying - NO 2. Dependence on - YES history

Standard option for an asset

American put-call

1

Today

P : strike for put option

C : strike for call option (C > P)

 $(t_i)_{1 \leq i \leq N}$: exercise times

A holder of the option can exercise it at any time t_i by selling the stock for the strike of put option P or buying the stock for the strike of call option C. If the holder will not exercise the option, then the option will expire worthless. Exercise Put (Bsellat P)

continue * × 41 ti Exercise Call (buy at C)

Standard option on interest rates

American swaption

 $(t_i)_{1 \le i \le N}$: exercise times

Parameters of underlying swap:

N : notional.

R : fixed rate

 δt : interval of time between the payments given as year fraction.

9

ma: total number of payments

side : this parameter defines the side of the swap contract, i.e. whether one pays "fixed" and receives "float" or otherwise.

A holder of the option can enter into the underlying swap agreement at any exercise time L_{e} . This time will become after that the issue time of the swap.

continue Today (to to swap

9

Barrier option for an asset

Down-and-out call

L : lower barrier

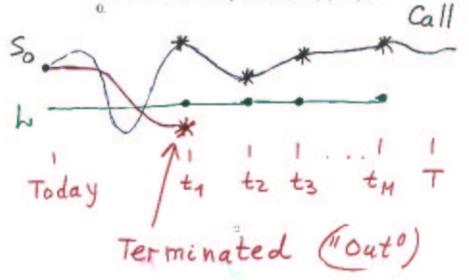
 $(t_i)_{1 \le i \le M}$: barrier times

K : strike

T : maturity $(T > t_M)$.

The payoff of the option at maturity equals the payoff of the standard call option if the spot price was above the barrier for all barrier times t_j . Otherwise, the payoff equals

10



Path dependent option on an asset

Convertible bond

Underlying coupon bond :

N : notional

R : coupon rate

 of t interval of time between the payments given as year fraction.
 11

m : total number of coupon payments.

The coupon times of the bond equal

 $c_i = t_0 + i\delta t, \quad 1 \le i \le m,$

where t_0 is the issue time for the bond. At any of coupon times the holder of the bond receives the coupon payment $NR\delta t$. At the last coupon time he also receives the notional amount N.

Today $C_1 C_1 C_1 N$ $C_1 C_2 C_m = T$ C = NRSt : couponpayment

Resettable strike : K : initial strike. \underline{K} : lower bound for the strike \overline{K} : upper bound for the strike (r_{\star}) : reset times for the strike. At time r_i the value of the strike is reset to L spot price $S(r_i)$ of the stock if $K \leq S(r_i) \leq$ 15 K. 2. lower bound for the strike \underline{K} if $S(r_i) \leq \underline{K}$. K 3. upper bound for the strike K if $S(r_i) \ge K$. K K 2, Dependency! Path

Exercise option for bond holder :

 (t_i) : exercise times for the bond.

At time t_i the holder of the bond can convert it to $N/K(t_i)$ shares of stock, where

- 1. $K(t_i)$ equals initial strike K if there are no reset times before t_i , i.e. $t_i < r_1$.
- K(t_i) equals the reset value at the largest reset time r_k which is less or equal t_k.

convert mto N/K(ti) shares continue .

Put option for bond holder :

- P: put redemption price, P < 1. At a put redemption time the holder of the bond can sell it back to issuer for put redemption amount NP.
- (p_i) put redemption times, i.e. those times when the bond holder can sell the bond back to issuer for the put redemption amount.

continue P2 P3 . .

Hard call option for bond issuer :

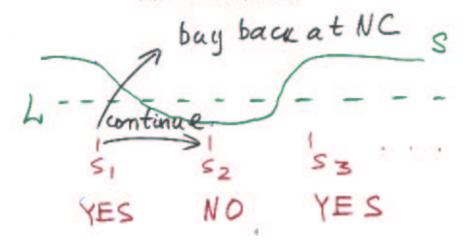
- H: hard call redemption price, H > 1. The issuer can buy the bond back for hard call redemption amount NH at a hard call redemption time.
- (h_{2}) : hard call redemption times.

option for issuer Buy back at NH. (continue h;

Soft call option for bond issuer :

L : soft call lower barrier. The issuer of the bond can buy back the bond for the soft call redemption amount at a soft call redemption time only if the price of the stock at this time is greater than L. 16

- C: soft call redemption price, C > 1. The issuer can buy the bond back for soft call redemption amount NC at a soft call redemption time only if the stock price at this time is greater that the soft call lower barrier.
- (s_i) : soft call redemption times.



Option for bond issuer at maturity :

M : mandatory strike.

At maturity the issuer of the bond has the right to deliver N/M stocks to the bond holder instead of paying the notional N. Option for issuer pay N deliver N/M

(mat

TERE min (N) payoff 10 at maturity

<u>Remark</u> Both models will be implemented in their most general forms.

For example, our Black 19
model will include as parti-
cular cases:
a) Classicar Black& Scholes
model:
$$M = const$$

 $dS_t = S_t (Ndt + OdW_t)$
b) Time dependent version:
Interestrate (M_t) t ≥0
 $dS_t = S_t (N_t dt + O_t dW_t)$
c) FX model
 $M = S_t (N_t dt + O_t dW_t)$
c) FX model
 $M = S_t (N_t dt + O_t dW_t)$

20 FX rate : $dS_{t} = S_{t}((rd_{r}f)dt + 6dW_{t})$ d) Commodity model $dS_{t} = S_{t} \left\{ \left(\phi_{t} - \lambda \ln \xi \right) dt \right\}$ + GdW, 2: mean reversion coefficient KEYIDEA: write the model in terms of forward prices!

F(t,T): forward price

$$T$$
 maturity
time of forward
 $dF(t,T) = F(t,T) A(T) G(t) dW_t$
 $A = (A(T))_{T \ge 0}$ shape"
function
 $A = 1$: stocks + fx
 $A = A = T$: commodities

mean - reversion

effaBlackaData class Reference



Main Page Modules Namespace List Class Hierarchy Alphabetical List Data Structures File List Numespace Members Data Fields

cfl::Black::Data Class Reference

[Black model for a single asset.]

The gammeters of Black model. More ...

Public Methods

 Duta ()

 Duta (const Function &rtDiscount, const Function &rForward, const Function &rVolatility, double dimitedTime)

 Duta (const Function &rtDiscount, const Function &rForward, const Function &rtDiscount, const Function &rforward, double disgna, double discount function &rtDiscount, const Function &rforward, double disgna, double discount (const Function &rforward, double disgna, double discount () const const Function, & discount () const const Function, & forward () const const Function, & shape () const const Function, & shape () const

Detailed Description

This class defines the parameters of the Black model. The set of parameters consists of discount, forward, shape and volatility curves.

See also: eff:: Black:: Model

Constructor & Destructor Documentation

efinBlackmDatamData() [in]ine]

Default constructor.

cfl::Black::Data::Data: const Function & rDiscount, const Function & rForward, const Function & rVolatility, double divitiolTime

nk:@MSHStore:D:ProjectsiCourse#FCLibrary/DucketLehm://classeff || IBlack || 1D., 10/27/2004

```
Page 1 of 3 23
 off (Block: Data class Reference)
                             >
   Constructs parameters of classical Black model.
    Porameters:
          riblicount A discount curve.
           research A broard error.
restabling A solatility curve.
           dimittalTime Initial time given as year fraction.
 effi:Black::Dalas:Dafaj const Function & rDiscount,
const Function & rForward,
double dSigma,
double dSigma,
                            .
   Constructs parameters of dassical Bluck model with converse volatility.
    Parameters:
          rDiscourt A discourt curve.
          rDanaan - A forward curve,
15oraani - A forward curve,
d<sup>8</sup>igma - A volocillty
dbattedTata loitia, time given as year lasetor.
 eff:: Bluck:: Duta:: Duta: const Function & rDiscount,
const Function & rForward,
const Function & rVolatility,
const Function & rShape,
double dhuitalFime
                            1
Constructs parameters of general Black mode ...
   Parameters:
           eDiscount A discount conver-
eHarward A Revend conve-
           el'idatility A volatility corve.
                          A shape function. This function, defines the shape of movements of the curve of
           eShape.
                          forward prices.
           division?time initial time given as year fraction.
 elledBack:/Data::Data:enst Tunction & +Discount,
                                const Function & Forward,
                                             dSigma,
dLambda,
                                double.
                                double.
                                double
                                                      diminial Time
                            .
```

Constructs parameters of general Black model with stationary valuatility.

rik (@MSITStore:D/Projects/Courses/FCL in any Dock Rehmstelnssef, 1_1Black_1_10., 10/222004

| | s Reference | Page 7 of 3 |
|-------------------------|--|-------------|
| Purameters: | | |
| ribiceant | A discours curve | |
| (Farward | | |
| alSigma | The volutility of agoing the | |
| di.ambila | The mean-reversion opefficient for log of spot price under the rit | sk neumil |
| | TROSESSIERS. | |
| dimitial'i imi | e Initial time given as year fraction. | |
| Member Fur | ection Documentation | |
| coast Function& cf | D::Black::Duta::discount(_) const | |
| Returns: | | |
| The discount | . curve. | |
| coust Function& cf | htBlack::Data::forward() const | |
| Returns: The forward | Ethanes. | |
| | | |
| double effetBlacket | Duta::initialTime() const | |
| Referens: | | |
| The unital ti | une as year traction. | |
| const Function& ef | LaBlackaDataashape() const | |
| | | |
| Remarks: | arve for movements of forward prices. | |
| | arve its introvencing of conversion proces. | |
| The shape c | In Back: Data wolutility() const | |
| The shape c | In:Black::Data::volutility() const | |

Menuration Was 134 26 15 35 35 2664 per (FL 1) .

mini/AMSITSharesDCProjects+Courses#CLibrary-DoubdFlichmedial-sheft 1_18lands_1_10... 10927/2004

Data curves for financial models.



Main Page Modules Namespace List Class Hierarchy Alphabetien/List Data Structures File List Namespace Members, Data Fields

Data curves for financial models.

Functions

Function discount (double d'vield, double d'initial l'inte)
 Function discount (const Function &rVield, double d'initial l'inte)
 Function discount (double dSigna, double d'arrivita, double d'initialTime)
 Function forward (double dSpot, double dCostOfCarry, double d'initialTime)
 Function forward (double dSpot, const Function &rCostOfCarry, double d'initialTime)
 Function forward (double dSpot, double dDividendVield, const Function &rCostOfCarry, double dinitialTime)
 Function forward (double dSpot, double dDividendVield, const Function &rCostOfCarry, double dinitialTime)
 Function assetShape (double dLambda, double d'InitialTime)
 Function brudShape (double dLambda, double d'InitialTime)

Detailed Description

This module contains implementations of different data curves for financial modes such as discount, forward, volatility, etc. .

Function Documentation

Function assetShape(double dl.ambda, double dl.atiailTime

) Stationary form for changes in forward price curves. The value of this function at time at given us

year fraction equals

explate the date (dP - disidilation)

Parameters:

dLambda The mean-reversion rate. dlititia/Time The initial time as year function

Function bondShape(double dLambda, double dlaittalTime

Stationary form for changes in yield to maturity and discourt factor curves. The value of this function at time d⁺ given as year fraction equals

mk:@MSITState:D:Project@Coursesil CLibrary/DeckefLehnesigroup__of/Data.html 10:27/2004

```
Page 2 of 4 26
Data curves for financial models.
   ("resp)-dfashdampd" = dfsc.lisl?unettt/dumoon
  Parameters:
        dlambda The mean reversion rate.
        alluitialTime The mitial time as year fraction.
Function discount( const Function & rYield,
                                      dInitial Time
                   double
                  1
  Constructs discount curve. The discount factor for maturity dResson or given as year fraction equals
   septemption (description) = climitiation (
  Parameters:
                    The continuously compounded yield curve.
        "Yield
        abriticalTime The initial time as year fraction.
  Returnst
       The discount curve as function of maturity.
Function discount( double dVield,
double dInitialTime
  Constructs discourt curve. The discourt factor for maturity enstancisty given as year fraction equals
   server-exacted reflecturing - sfor had " reall
  Parameters:
       dYield
                    The constant continuously compounded yield.
       dboitiolTime The initial time as year fraction.
  Returns:
The discount curve as function of matarity.
Function forward( double
                                     dSport,
                  double
                                     dDividend Vield.
                   const Function & Discount,
                  double
                                     dinitial Time
                 )
 Constructs forward curve for a stock. The forward price at maturity discussion as year fraction
 equals
   ofpairs any affisidend/ield/schaturity = dfm = string/sblacount(distantly)
mkti@MSITStote.D.ProjectylCoursesil CLibrary/DectefLehrningroup_cflData.html
                                                                                    18/27/2014
```

Page 3 of 4 24

Parameters: eSpot Use spot price. eDividend/field The dividend yield for the stock eDiscours: Use current discourt, curve. eDistrictTime: The initial time as year fraction.

Data curves for financial models.

Function forwardly double dSpot. const Function & rCorrOfCore, disable dInitialTime

Constructs forward course for an easet. The forward price at maturity 184-101 (y equals

dBest * CKEINCLINDTOnergolDarios hypothytertay = alascuall tab)

Purameters:

diport The spor price. dimarget army The anst-of-corry sets corror. dimarget army The initial time as year facebon.

Function forward(double_dSpot, double_dCostOfCarry, double_dEnitialTime }

Constructs forward convertor an asset. The forward price at manurity (174) and 19 equals

disal f asplitusiels ty obstraty alministration (

Parameters:

dSpox The spot price dCoxOfCorry The cost-of-carry rate. dbottedPrime The initial function year ratelion.

Punction volutility(double d'signor, double d'ambia, double d'attialTime)

Constructs stationary volatility curve. The value of volatility for faile of given as year fraction equals

offges - seril (seriler_classication) distants (inst) - 11312*0_25alls rts* distants

Parameters:

aSigma The short-term volatility diambda The mean-reversion coefficient, distiliation: The initial time as your fraction

nka@MSITNaredD/Projects/Course/FCLDracy/DoctefLehanarginary_of/Data http:////2021/2024

Overview of cfl 28 Library 1 Any model has discrete or finite time structure. (ti)osism of event times as year fractions to: initial time Examples : a) exersise times B) barrier times Event times = All times we need to price given derivative.

Efficiency: create the vector of event times as small as possible

2. The main class is 130 Slice: represents the value of security at event time. $\begin{array}{c} K: \ cash \ at \ t_2 \\ (S(t_2): \ spotprize \ att_2 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ t_1 & (t_2 \ t_3 \ t_4 \ t_5 \end{array}$ max(S(t2)-K,0): callis payoff at t2

2 types of operations for Slice 1 At given event time ti: all possible arithmetic, functional,...

2. Between 2 event times ti < tj: only rollback Example ti: event time uSlice1: S(ti) spot at ti uSlice2: K cash at ti Slice uSlice3 = max (uSlice1-uSlice2,0.)

u Slice 3: payoff of call max (S(ti) - K, 0.) Wrong code ti: event time uSlice1: spot at ti tj: enother event time uSlice2: spot at tj

Slice u Slice 3 = u Slice 1 + u Slice 2; You want u Slice 3: S(ti) + S(tj) However this is not allowed! Q: How we create needed Slices?

 Basic Slices are created by a model
 Manipulate using

 a) arithmetic k
 functional operations
 for <u>particular time</u>
 b) rollback between
 different event
 times.

 cth::Black::Model class Reference



Main Page Modules <u>Numespace List</u> Case Hierarchy Alphabetical List Data Structures File List Numespace Memory Data Fields

cfl::Black::Model Class Reference

[Black model for a single asset.]

Implementation of Black model. More ...

Public Methods

Model (const Data & Data, const side vector< double > &rEventTimes, double dimerval, double dQuality)

Model (const Data &/Data, const std://ecotor

Model (const Data & Data, const stdrevector* double > &rEventTimes, dquble dimerval, const stdrevector* double > & (Quality)

Model (const Data & Data, const std://vector</double > &rEvenfTimes, double dimerval, const eff; (Brownian &/Drownian)

consi Bata & data () consi

unsigned addState (const PathDependent &rState)

Slice state (unsigned i l'ime, unsigned iState) censt

Slice eash (unsigned iTime, double dAmount) const

Slice discount (unsignal iTime, double diland Maturity) const

Slice spot (unsigned (Time) const

Slice forward (unsigned i lime, double dLorwardMaturity) const

Detailed Description

This class implements Black model for a single asset.

See also: eff::Black::Data

Constructor & Destructor Documentation

efl::Black::Model::Model: Const Data & rDuta, const std::vector= double > & rEventTimes, double diterval, double dQuality }

Constructs an implementation of the model. Use this constructor for priving of standard derivatives.

mk: j@MSITStore:D:Projacts/Control/FCLibrary/Doc/eft.chm://elasseft_1_109uck_1_1... 10/27/2004

Page 2 a 4 36

eil::Bisck::Medel.class Reference

Parameters:

| e Dana | The input data of the model. |
|--------------------|---|
| <i>rEventtimes</i> | The sector of event finors |
| distanced | "the width of the interval of initial values for the main state process. |
| aQvality | The quality of numerical implementation of the model. This parameter defines the model of herween speed and accuracy. |

| eft: Hinck: Mudeb: Model | const side:vector< double > & double double | eData, rEventTimes, dimercal, dimity, dPathDependOnality |
|--------------------------|---|--|
| construction of the | double | dPathDeprodQuality |

Constructs an implementation of the model. Use this constructor for pricing of path dependent derivatives with one additional state process.

Parameters:

| vibata | The input cash of the model. |
|------------------------|---|
| whereast Vicanes | The vector of event times. |
| dimensi | The width of the interval of initial values for the main state process. |
| dQuality | The quality of numerical implementation of the basic state process (arovation motion). |
| All add to moved budde | These alies of convertical in the neutration of additional state means. |

dPatitDependQuality. The quality of numerical implementation of nedicional state proces

eff.::Bineke:Model::Model(const Data & rData,

| const std:://ectors/ double > & | rEventTimes, |
|---------------------------------|-----------------|
| double | dinterval. |
| const std: :vector< double > & | rQuality |
| 1 | |

Caristrates an implementation of the model. Use this constructor for pricing of path dependent derivatives with a number of additional state processes.

Parameters:

| (Data | The input data of the model. |
|-------------|---|
| (EventTimes | The vector of event times. |
| divier ed | The width of the interval of initial values for the main state process. |
| rQcab@ | The element with index i of the vector $nQuality defines the quality of numerical implementation of the state process with index i .$ |

| cfl::Bluck::Model::Model(| const Data & | eData, |
|---------------------------|-------------------------------|--------------|
| | const and receives double > & | "EventTimes, |
| | double. | dimerval, |
| | const eff: Brownian & | *Brownian |
| | | |

The basic constructor of the model.

aika@MSITStere/DrPonjectsiConses/TCLibrary/DacketLehrto/classeff || Black || || || 10:22/2004

Page 3 of 4 34

eilt: Black.: Model class Keference

Parameters: rDesc Discription data of the model rDasc Discription data of the model rdiscription rdiscription The vestion of event items. rdiscription rdiscription The vestion of event items. rdiscription rdiscription The vestion of items. rdiscription rdiscription A collectate to an object of type effectivened. This object is used to implement if more as

Member Function Documentation

unsigned off:: Black:: Madel::addState(const PathDependent & rState) Adds to the model succlusivation state process. Parameters: eState. The description of a pith dependent process which becomes additional store process in the model. Heturna: The index of the additional state process. Slice eff::Black:Modelmeash(unsigned (Time, double d'imount) const Returns the representation of the constant value deformant at the event time with index define const Buta& eff: :Risek: :Model::data() const A reference to the input data of the model. Slice off: Black: Model: discount(unsigned *iTime*, double *dBondMoneity*) const Returns the discount factor at the event time with index (Line for maturity dBandhfameity, Slice off: Black:: Model: forward(unsigned Time, double dForward/Maturity

I const Returns the forward price of the asset at the event have with index iTime for naturity off-on-condisionarity.

Silice effectBlacks: Model: sepuil, anxigned (Time) const Returns the spot price of the asset at the event time with index (Tame

Slice effe:Blaste: Modelestore(unsigned /Time,

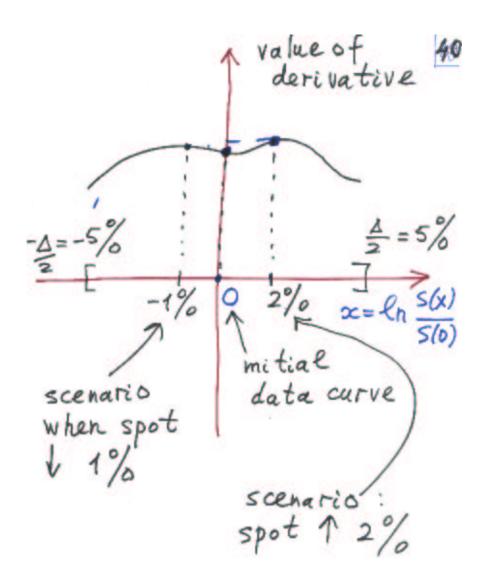
mk;@MSTTStore:DrProjects/Courses/PCLineary/Docie LahmatchusterT_1_1Black_1_1... 10/27/2004

| e THWard Phashi alawa Redmenar | Page-1 at 4 | 38 |
|--|-------------|----|
| unsignal Shate | | |
| Returns incorporation of the same process with index degrees, the event one with | inks Our | |
| The documentation for this class was generated time the following if e- | | |
| * Black/Multh | | |
| | | |
| - included a Mercine 3 18 48 18 200 (Ar CPL 1) (CONTRACTOR - Arts | | |

$mk(\delta D(S), some D(Property Councerbert here (0, a) of characterises. (1) Hatak _1 _ ... = 0.252.09$

Output for Black model
We compute the
value of the option
as the function of
relative change in the
spot price : perturbed

$$\infty = -\ln \frac{S(x)}{S(0)}$$
 mitial
Output : price
 $V_0 = (V_0(x)) - \frac{A}{2} \le x \le \frac{A}{2}$
A: interval for relative
changes



Risk para meters:
delta

$$\Delta = \frac{\partial}{\partial x} V_0(x) |_{x=0}$$

 $\Gamma = 10^{-2} \frac{\partial^2 V(x)}{\partial x^2} |_{x=0}$
 $1_0^{-2} gamma$