4.1

1. Review of AFP
   a) Rollback operator
   b) State processes
2. Basic design of cfl library: classes IModel and Slice
3. Implementation of financial models with identical state processes. Classes Similar and IRollbackDensity

Rollback (pricing) operator

4.2

$$V_s = \mathcal{R}(V_t, s) \qquad\qquad V_t$$

$$s \longleftarrow t$$

V_t : random variable that defines a payment at t

$V_s = \mathcal{R}(V_t, s)$: AFP (value) of the payment V_t at time s (capital of replication strategy at s)

Question: How to compute $\mathcal{R}$?

Rollback in terms of $\boxed{4.3}$
money-market measure.

$r_t = r_t(\omega)$: short-term interest rate

$B_t = e^{\int_0^t r_u du}$: bank account

$\mathbb{P}^*$: money-market martingale measure

Def: $\mathbb{P}^*$ is such a measure that

$$\frac{X_t}{B_t} = X_t e^{-\int_0^t r_u du}, \quad 0 \le t \le$$

is $\mathbb{P}^*$ - martingale for any wealth process X,

4.4

that is,

$$X_s e^{-\int_0^s r_u du} = \mathbb{E}_s^*\left[X_t e^{-\int_0^t r_u du} \right]$$

$$\Updownarrow$$

$$X_s = \mathbb{E}_s^*\left[X_t e^{-\int_s^t r_u du} \right]$$

Here

$\mathbb{E}_s^*[\cdot]$: operator of conditional expectation under $\mathbb{P}^*$ given information at s.

4.5

Theorem $\forall\, s<t$:

$$\Re(V_t, s) = \mathbb{E}^*_s\left[V_t e^{-\int_s^t r_u\,du}\right]$$

Proof Follows from the definition of $\mathbb{P}^*$ and the fact that

$$\boxed{\text{AFP}} = \boxed{\begin{array}{l}\text{(initial) wealth}\\ \text{of replication}\\ \text{strategy}\end{array}}$$

Remark

computation of $\mathcal{R}(\cdot, s) \Longleftrightarrow$ computation of $\mathbb{E}_s^*[\cdot]$

We need to implement the operator of conditional expectation under a risk-neutral measure!

4.7

Rollback in terms of forward measures.

$B(s,t)$: price at s of zero-coupon bond with face value \$1 and maturity t

$\mathbb{P}^t$: forward martingale measure for maturity t

Def: $\mathbb{P}^t$ is such a measure, that

$$\frac{X_s}{B(s,t)}, \quad 0 \le s \le t,$$

is $\mathbb{P}^t$-martingale for any wealth process X, that is, $\boxed{4.8}$

$$\frac{X_s}{B(s,t)} = \mathbb{E}_s^t [X_t]$$

$$\Updownarrow$$

$$X_s = B(s,t)\, \mathbb{E}_s^t [X_t]$$

$\mathbb{E}_s^t[\cdot]$: operator of conditional expectation under $\mathbb{P}^t$ given information at s

Theorem $\quad \forall s < t:$ 4.9

$$\mathcal{R}(V_t, s) = \mathbb{E}_s^t[V_t]\, B(s,t)$$

Proof

Replication + definition of forward measure.

Question: Why $\mathbb{P}^t$ 4.10

is called forward martingale measure for maturity t?

$\mathbf{F}(s,t)$: forward price

s: current time

t: delivery

Consider long position:

$X_s = 0$: value at s

$X_t = S_t - F(s,t)$: value at t

$\boxed{4.11}$

$$\underset{X_s}{\underbrace{0}} = B(s,t)\, \mathbb{E}_s^t[\underbrace{S_t - F(s,t)}_{X_t}]$$

$$\Updownarrow$$

$$F(s,t) = \mathbb{E}_s^t[S_t]$$

Hence,

$\left(F(s,t)\right)_{0 \le s \le t}$ is $\mathbb{P}^t$-martingale

State processes

Idea: efficient storage for relevent random variables.

Example (Binomial model)

r: one-step interest rate

$$\begin{tikzcd}
 & uS_n \\
S_n \arrow[ur, "\omega_{n+1}=H"] \arrow[dr, "\omega_{n+1}=T"'] & \\
 & dS_n
\end{tikzcd}$$

n $\qquad\qquad$ $n+1$

u: relative change "up"

d: — " — "down"

4.13

Consider payment

$V_{n+1} = V_{n+1}(\omega_1 \dots \omega_{n+1})$

at time $n+1$

Rollback operator:

$$V_n \xleftarrow{\text{rollback}} V_{n+1}$$

$$V_n = \mathcal{R}(V_{n+1}, n) = \frac{1}{1+r}\left[\tilde{p}\, V_{n+1}(\omega_{n+1}=H) + \tilde{q}\, V_{n+1}(\omega_{n+1}=T)\right]$$

$\tilde{p}$ & $\tilde{q}$: one-step risk-neutral probabilities

"Naive" storage scheme: 4.14

record values of

$V_n = V_n(\omega_1 \ldots \omega_n)$

for any $(\omega_1, \omega_2, \ldots, \omega_n)$

\# of records $= 2^n$

(too big)

However, to price standard options we need to operate with random variables in the form:

$V_n = f_n(S_n)$

where $f_n = f_n(x)$ is a deterministic function $\boxed{4.15}$

# of records for "standard" storage scheme $= n+1$

practical

$(S_n)_{0 \le n \le N}$ is an example of a state process.

Definition A stochastic process $(X_t)_{0\le t\le T}$ is $\boxed{4.16}$ called a state process if $\forall\, s\le t$ and deterministic function

$$f = f(x)$$

$\exists$ deterministic function

$$g = g(x)$$

such that

$$\begin{array}{c} g(X_s) \xleftarrow{\text{rollback}} f(X_t) \\ \| \\ \mathcal{R}(X_t, s) \end{array}$$

Remark For a state [4.17]
process $X=(X_t)_{0\le t\le T}$
denote by

$$\mathcal{X}(X_t)=\left\{ f(X_t) : \begin{array}{l} f \text{ is} \\ \text{determ.} \\ \text{function} \end{array} \right\}$$

the family of random variables determined by (measurable w.r.t) X_t.

Then

(a) for particular time t the family $\mathcal{X}(X_t)$ is closed under any arithmetic and functional operation

(b) for two times $s<t$
and any $\boxed{4.18}$
$\xi \in \mathcal{X}(X_t)$
$(\xi = f(X_t))$
the result of rollback operator between t and s belongs to $\mathcal{X}(X_s)$:

$$\mathfrak{R}(f(X_t), s) = g(X_s)$$

for some deterministic $g = g(x)$.

Recall Slice !!!

4.19

Implementation of a financial model consists of

(a) specification of a state process X

(b) implementation of necessary operations for random variables from

$$\mathcal{X}(X_t) = \{ f(X_t(\omega)) : f = f(x)$$

determ. function

(i) for given time t —
all arithmetic & functional

(ii) between two times $\boxed{4.20}$
$s<t$ — rollback

$$g(X_s) = \mathcal{R}(f(X_t), s)$$

Examples:

$\exp(X_t)$, $I(X_t \geq K)$

Characterisation of state processes as Markov processes. 4.21

Recall that a stochastic process $X = (X_t)_{0 \le t \le T}$ is called Markov process if for any $s < t$ and any $f = f(x)$ $\exists\ g = g(x)$ such that

$$g(X_s) = \mathbb{E}_s[f(X_t)]$$

4.22

Theorem

(i) X is a state process

$\Updownarrow$

(ii) for any time t

(a) $(X_s)_{0 \le s \le t}$ is a Markov process under $\mathbb{P}^t$

(b) discount factor at s with maturity t is determined by (measurable w.r.t.) X_s:

$$B(s,t) = f(X_s)$$

for some $f = f(x)$.

Proof Follows from the formula for rollback operator: $\boxed{4.23}$

$$\mathcal{R}(\cdot\,, s) = B(s,t)\,\mathbb{E}_s^t[\cdot]$$

A model in cfl library

Basic components: $\boxed{4.2}$

(a) state process

$$X = \underbrace{(X^0 \ldots X^{d-1})}_{d\text{-dimensional}}$$

(b) vector of event times

$$t_0 \quad t_1 \quad \ldots \quad t_N$$

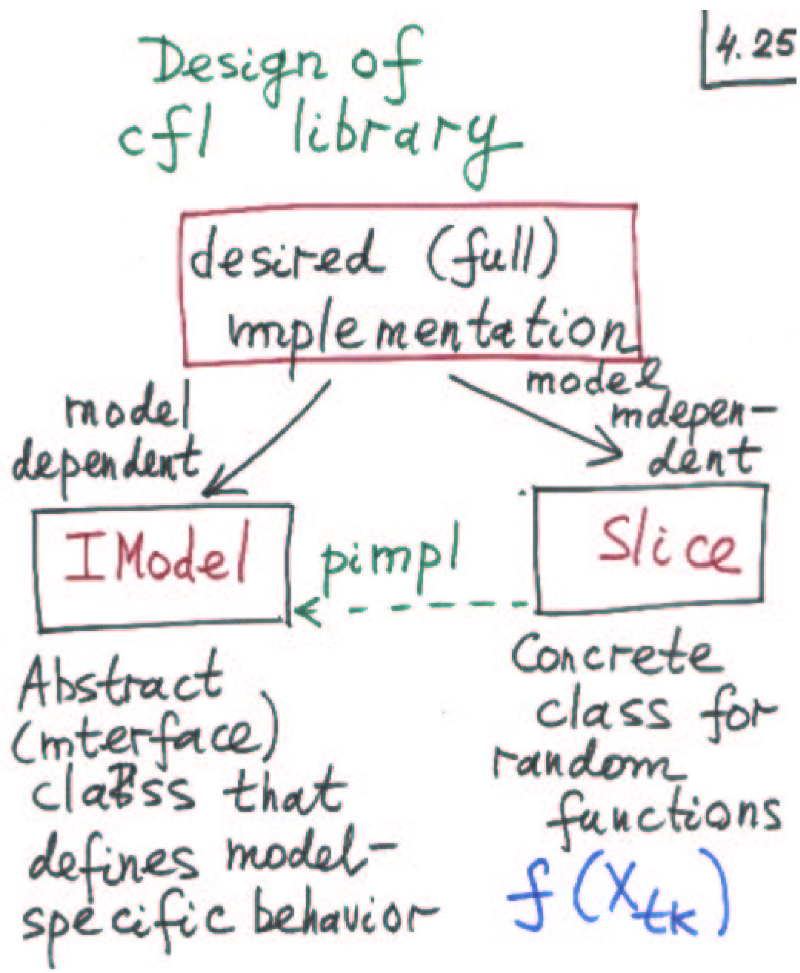
We can use only random variables in the form:

$$f(X_{t_k})$$

deterministic

state process

event time


Design of cfl library
4.25
desired (full) implementation
model dependent
model independent
IModel
pimpl
Slice
Abstract (interface) class that defines model-specific behavior
Concrete class for random functions $f(X_{tk})$

class IModel 4.26

1. Virtual destructor!

2. eventTimes
   sorted vector <double>
   first time = initial time

3. numberOfStates
   returns the number of
   state processes

4. numberOfNodes
   Slice $\longleftrightarrow f(X_{t_k}^{i_1}, \ldots, X_{t_k}^{i_m})$
   $(i_1, \ldots, i_m)$: vector of indexes
   for state processes that
   "support" given random variable.

Returns the number of $\boxed{4.24}$
to double used to represent
Slice object in computer's
memory.

Slice $\longleftrightarrow$ 1 $\boxed{1}$

Slice $\longleftrightarrow$ spot(t_k)

5. origin

   Returns initial value for state process.

6. state

   index of state

   Slice $\longleftrightarrow$ $X^{j}_{t_k}$

   index of event time

7. add Dependence

4.28

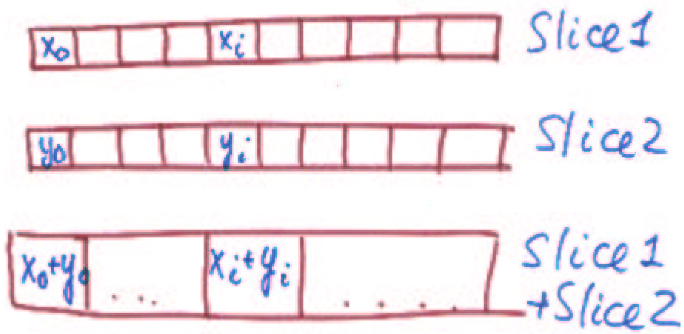
Problem: implement $+$ for two Slices

Slice 1 $\longleftrightarrow f(x^{i_1}, x^{i_2}, \ldots, x^{i_m})$

Slice 2 $\longleftrightarrow g(x^{j_1}, \ldots, x^{j_n})$

Easy case: $(i^1, \ldots, i^m) = (j_1 \cdots j_n)$

Slices are in "agreement"



4.29

Difficult case:

$(i^1, \dots, i^m) \neq (j^1, \dots, j^m)$

Slice 1

Slice2

Storage schemes are different. What to do?

<u>Solution</u> Clearly,

Slice1 + Slice2 will depend on state processes with indexes

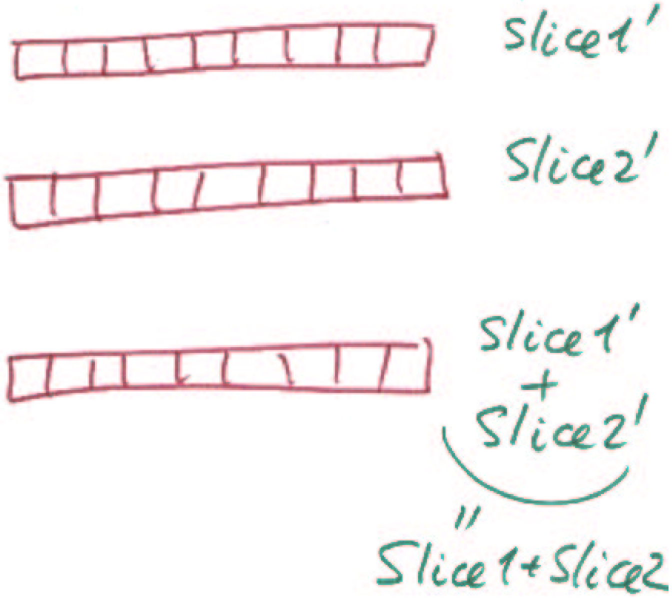
$(k_1, \dots k_e) = (i_1 \dots i_m) \cup (j_1 \dots j_n)$

We then "add dependence" to Slice 1 and Slice 2 or change their storage schemes to that of $\boxed{4.30}$
Slice 1 + Slice 2.

$$\begin{tikzcd}
\text{Slice 1} \arrow[r, "\text{add Dependence}"] \arrow[d, leftrightarrow] & \text{Slice 1}' \arrow[d, leftrightarrow] \\
f(x^{i_1} \ldots x^{i_m}) \arrow[r, phantom, "="] & f'(x^{k_1} \ldots x^{k_l}) \\
\text{Slice 2} \arrow[r, "\text{add Dependence}"] \arrow[d, leftrightarrow] & \text{Slice 2}' \arrow[d, leftrightarrow] \\
g(x^{j_1} \ldots x^{j_n}) & g'(x^{k_1} \ldots x^{k_l})
\end{tikzcd}$$

$\boxed{4.31}$

We are back to "easy" case



4.32

8. rollback

start rSlice $\Longleftrightarrow f(x_{t_e})$

end rSlice $\Longleftrightarrow g(x_{t_k})$

$$g(x_{t_k}) = R(f(x_{t_e}), t_k)$$

$$g(x_{t_k}) \xleftarrow{\text{rollback}} f(x_{t_e})$$

indicator

rSlice $\Longleftrightarrow f(X_{t_k})$

dBarrier $\Longleftrightarrow K$

returns $I\left(f(x_k) > k\right)$

-2	-1	0	1	2	3	rSlice

dBarrier = 0.5

0	0	0	1	1	1	"naive" indicator

0	0	1/4	3/4	1	1	"smart" indicator

10 interpolate $\boxed{4.34}$

Slice $\longleftrightarrow f(X^{i_1}_{t_k}, \dots, X^{i_m}_{t_k})$

discretization

$\longleftarrow$

interpolation

$\longrightarrow$

class Slice

4.35

Slice $\longleftrightarrow f(X_{t_k}^{i_1} \dots X_{t_k}^{i_m})$

Components:
(private members)

1. array of values
   (disretization of $f = f(x^{i_1} \dots x^{i_m})$)
2. vector of dependences
   $(i_1 \dots i_m)$
3. (index of) event time
   t_k
4. pimpl of IModel

4.36

Some functions

(i) 1. rollback

2. indicator

3. interpolate

Great help from STL:

array of values $\longleftrightarrow$ std::valarray

Implementation of financial models with identical state processes ("similar" models) 4.37

Consider a stochastic process

$X = (X_t)_{0 \le t \le T}$

on a filtered probability space

$(\Omega, (\mathcal{F}_t)_{0 \le t \le T}, \mathbb{P})$

Consider two financial models A and B such $\boxed{4.38}$ that they have the same maturity and for both models

X is a state process

$\mathbb{P}$ is the forward martingale measure for maturity T

We call the models A and B "similar".

4.39

Since X is a state process we have

$$\underbrace{d^A(s,t)}_{\text{discount factor for model } A} = f_{s,t}(X_s) .$$

deterministic function

$$\underbrace{d^B(s,t)}_{\text{discount factor for model } B} = g_{s,t}(X_s)$$

deterministic function

Denote $\boxed{4.40}$

$$Z_t = \frac{d^A(t,T)}{d^B(t,T)} = \frac{f_{t,T}(X_t)}{g_{t,T}(X)}$$

$$:= h_{t,T}(X_t)$$

We have

$$\mathcal{R}^B(\varphi(X_t), s) = \frac{1}{Z_s} *$$

$$\mathcal{R}^A(\varphi(X_t) Z_t, s) =$$

$$= \frac{1}{h_{s,T}(X_s)} \mathcal{R}^A(\varphi(X_t) h_{t,T}(X_t), t)$$

If we have an implementation of model A it is very easy to implement model B. 4.41

Random variable:

$$Z_t = \frac{d^A(t,T)}{d^B(t,T)}$$ is called the density of rollback operator for model B w.r.t. model A.

14.42

In cfl library this methodology for the implementation of "similar" models is realised through classes

Similar $\xrightarrow{\text{pimpl}}$ IRollbackDen-sity class IRollbackDensity

Pure ~~vir~~ abstract class.

$\boxed{4.43}$

1. at

returns

$$Z_{t_k} = \frac{d\mathcal{R}^B_{t_k}}{d\mathcal{R}^A_{t_k}}$$

the density of new model w.r.t. old model

$$Z_{t_k} \longleftrightarrow f(X_{t_k}) \quad (\leftrightarrow \text{Slice})$$

that is, 4.4

$$X_s e^{-\int_0^s r_u du} = \mathbb{E}_s^*\left[X_t e^{-\int_0^t r_u du}\right]$$

$$\Updownarrow$$

$$X_s = \mathbb{E}_s^*\left[X_t e^{-\int_s^t r_u du}\right]$$

Here

$\mathbb{E}_s^*[\cdot]$: operator of conditional expectation under $\mathbb{P}^*$ given information at s.

4.45

Key example :

$$X_t = \int_0^t \sigma_u \, dB_u$$

$\sigma = (\sigma_t)_{0 \le t \le T}$: deterministic function

$B = (B_t)_{0 \le t \le T}$: standard Brownian motion

This process is a state process for <u>many</u> models:

(a) Extended Black

(b) Hull-White

(c) Black-Karachinski

(d) BDT

⋮

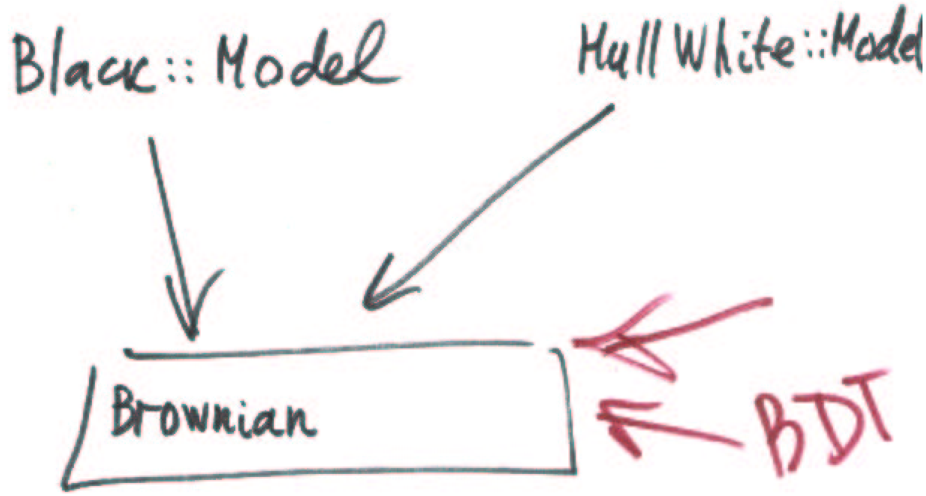
In cfl library an [4.46]
"artificial"
Brownian model
has been implemented
where

$$\mathcal{R}(\cdot, s) = \mathbb{E}_s[\cdot]$$

(interest rate $\equiv 0$)

Then this model was
used to implement
Black and Hull White

4.47



GREAT FOR TESTING!