RQuantLib: Interfacing QuantLib from R

Dirk Eddelbuettel[1] Khanh Nguyen[2]

[1] Debian Project

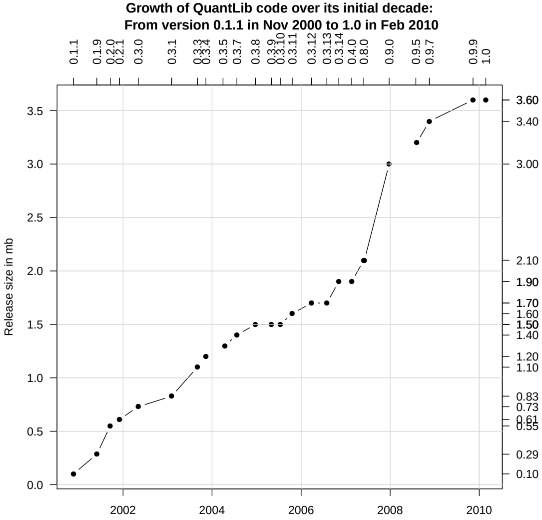
[2] UMASS at Boston

Presentation on 23 July 2010 at *useR! 2010*
National Institute of Standards and Technology (NIST)
Gaithersburg, Maryland, USA

Outline

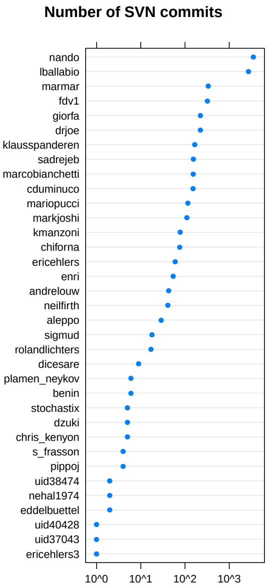
QuantLib releases

Showing the growth of QuantLib over time



- The initial QuantLib release was 0.1.1 in Nov 2000
- The first Debian QuantLib package was prepared in May 2001
- Boost has been a QuantLib requirement since July 2004
- The long awaited QuantLib 1.0.0 release appeared in Feb 2010

A few key points about QuantLib



QuantLib ...

- is a C++ library for financial quantitative analysts and developers.
- was started in 2000 and is hosted on Sourceforge.Net
- is a free software project under a very liberal license allowing for inclusion in commercial projects.
- is primarily the work of Ferdinando Ametrano and Luigi Ballabio.
- is sponsored by the Italian consultancy StatPro which derives consulting income from it.

QuantLib Architecture

How is it put together and how do I use it?

- QuantLib is written in C++ and fairly rigourously designed.
- Luigi Ballabio has draft chapters on the QuantLib design and implementation at `http://sites.google.com/site/luigiballabio/qlbook`.
- QuantLib use the Boost testing framework and employs hundreds of detailed unit tests.
- QuantLib makes extensive use of Swig and bindings for Java, Perl, Python, Ruby, C#, Guile ... exist.
- QuantLibAddin exports a procedural interface to a number of platforms including Excel and Oo Calc.
- Several *manual* (non-SWIG) extension such as **RQuantLib** exist as well.

Key Modules

A rough guide, slight re-arranged from the QuantLib documentation

- Pricing engines (Asian, Barrier, Basket, Cap/Floor, Cliquet, Forward, Quanto, Swaption, Vanilla)
- Finite-differences framework
- Fixed-Income (Short-rate modelling, Term structures)
- Currencies and FX rates
- Financial instruments
- Math tools (Lattice method, Monte Carlo Framework, Stochastic Process)
- Date and time calculations (Calendars, Day Counters)
- Utilities (Numeric types, Design patterns, Output manipulators)
- QuantLib macros (Numeric limits, Debugging)

Options: Fifteen solutions and three different exercises

```
$ EquityOption

Option type = Put
Maturity = May 17th, 1999
Underlying price = 36
Strike = 40
Risk-free interest rate = 6.000000 %
Dividend yield = 0.000000 %
Volatility = 20.000000 %


Method                             European     Bermudan     American
Black-Scholes                      3.844308     N/A          N/A
Barone-Adesi/Whaley                N/A          N/A          4.459628
Bjerksund/Stensland                N/A          N/A          4.453064
Integral                           3.844309     N/A          N/A
Finite differences                 3.844342     4.360807     4.486118
Binomial Jarrow-Rudd               3.844132     4.361174     4.486552
Binomial Cox-Ross-Rubinstein       3.843504     4.360861     4.486415
Additive equiprobabilities         3.836911     4.354455     4.480097
Binomial Trigeorgis                3.843557     4.360909     4.486461
Binomial Tian                      3.844171     4.361176     4.486413
Binomial Leisen-Reimer             3.844308     4.360713     4.486076
Binomial Joshi                     3.844308     4.360713     4.486076
MC (crude)                         3.834522     N/A          N/A
QMC (Sobol)                        3.844613     N/A          N/A
MC (Longstaff Schwartz)            N/A          N/A          4.481675

Run completed in 5 s
```

Errors from discrete hedging (Derman and Kamal)

```
$ DiscreteHedging

Option value: 2.51207

         |          |      P&L |      P&L | Derman&Kamal |      P&L |      P&L
 samples |   trades |     mean | std.dev. |      formula | skewness | kurtosis
-----------------------------------------------------------------------------
   50000 |       21 |   -0.001 |     0.43 |         0.44 |    -0.33 |     1.56
   50000 |       84 |    0.000 |     0.22 |         0.22 |    -0.20 |     1.68

Run completed in 16 s
```

Other examples include `SwapValuation`, `Repo`, `Replication`, `FRA`, `FittedBondCurve`, `Bonds`, `BermudanSwaption`, `CDS`, `ConvertibleBonds`, `CallableBonds` and `MarketModels`.
Also available are `quantlib-benchmark` (running 85 tests) and `quantlib-test-suite` (running 446 tests cases).

Outline

Overview

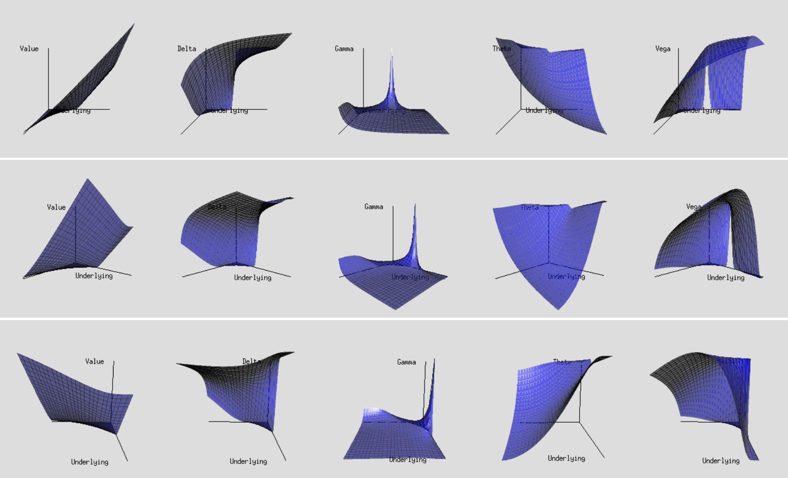
- Initial implementation: Standard equity option pricing:
  - pricers and greeks for European and American options
  - first set of exotics using barrier and binaries
  - also implied volatility calculations where available
- First external contribution: Curves and Swaption pricing.
- Second external contribution (as Google Summer of Code): Fixed Income Functionality (more on this below)
- Other small extensions on date and holiday calculations.

Option Valuation and Greeks
Analytical results where available

```
R> example(EuropeanOption)

ErpnOpR> # simple call with unnamed parameters
ErpnOpR> EuropeanOption("call", 100, 100, 0.01, 0.03, 0.5, 0.4)
Concise summary of valuation for EuropeanOption
   value    delta    gamma     vega    theta      rho   divRho
 11.6365   0.5673   0.0138  27.6336 -11.8390  22.5475 -28.3657

ErpnOpR> # simple call with some explicit parameters, and slightly increased vol:
ErpnOpR> EuropeanOption(type="call", underlying=100, strike=100, dividendYield=0.01,
ErpnOp+ riskFreeRate=0.03, maturity=0.5, volatility=0.5)
Concise summary of valuation for EuropeanOption
   value    delta    gamma     vega    theta      rho   divRho
 14.3927   0.5783   0.0110  27.4848 -14.4673  21.7206 -28.9169
R> example(BinaryOption)

BnryOpR> BinaryOption(binType="asset", type="call", excType="european",
BnryOp+              underlying=100, strike=100, dividendYield=0.02,
BnryOp+              riskFreeRate=0.03, maturity=0.5, volatility=0.4, cashPayoff=10)
Concise summary of valuation for BinaryOption
  value   delta   gamma    vega   theta     rho  divRho
 55.760   1.937   0.006  12.065  -5.090  68.944 -96.824
R> example(BarrierOption)

BrrrOpR> BarrierOption(barrType="downin", type="call", underlying=100,
BrrrOp+  strike=100, dividendYield=0.02, riskFreeRate=0.03,
BrrrOp+  maturity=0.5, volatility=0.4, barrier=90)
Concise summary of valuation for BarrierOption
 value  delta  gamma   vega  theta    rho divRho
 3.738    NaN    NaN    NaN    NaN    NaN    NaN
```

Option Valuation and Greeks

The `demo(OptionSurfaces)` provides some animation



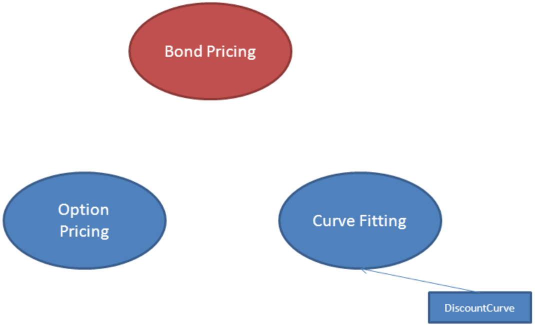
Outline

Fixed Income Development

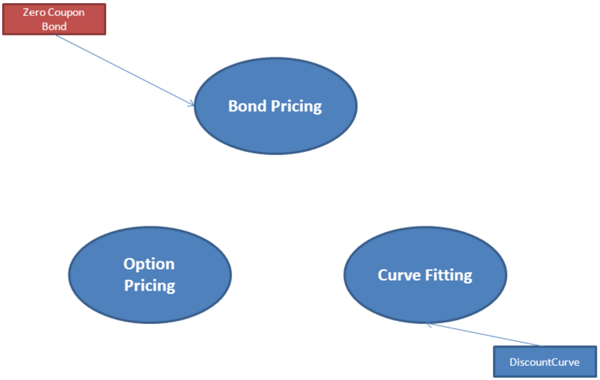
RQuantLib before GSOC 2009...

Fixed Income Development

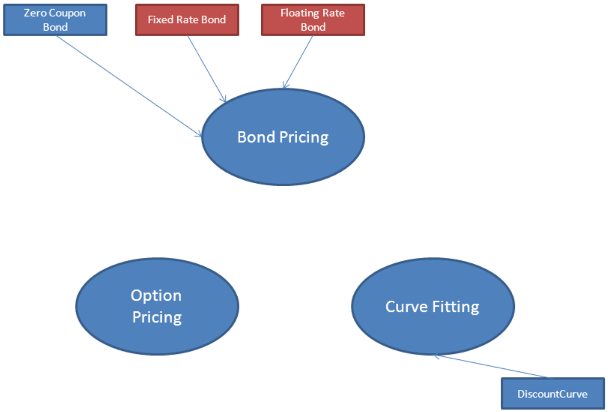
GSOC started. April 2009...



Fixed Income Development

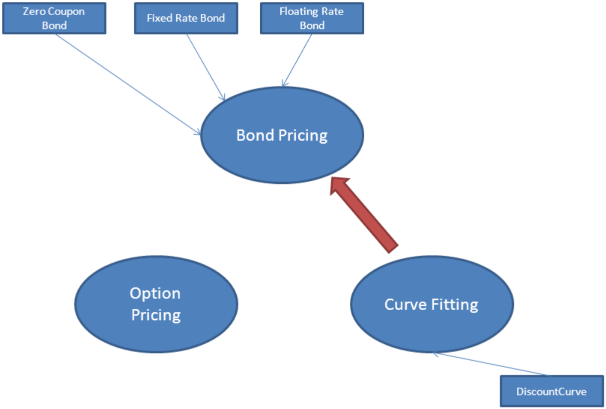


Fixed Income Development

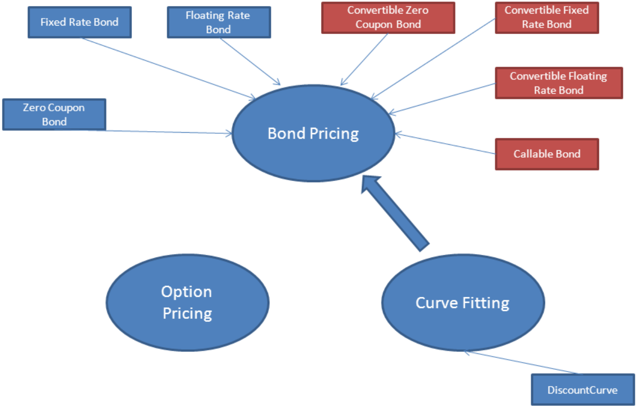


Fixed Income Development

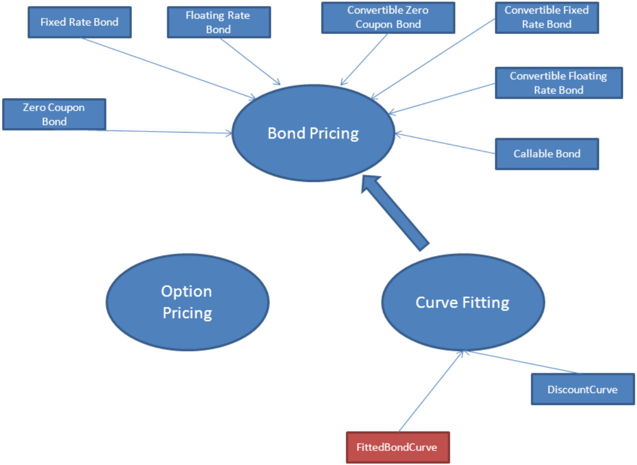
Making curve fitting and bond pricing work together...



Fixed Income Development

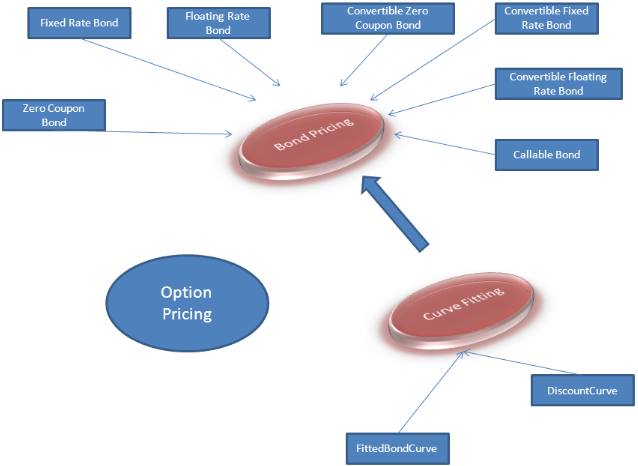


Fixed Income Development



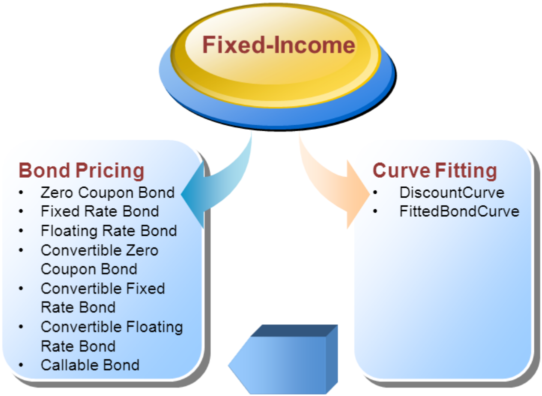
Fixed Income Development

And recently, we have started to add GUIs



Fixed Income Development

In summary



Examples....

- Curve fitting: DiscountCurve
- Curve fitting: FittedBondCurve
- Bond pricing: QuantLib's bond examples
- Bond pricing: Matlab's convertible bond
- Graphical User Interface screenshots

Fixed Income in RQuantLib
Examples: Bond pricing

We construct a bond discounting term structure and then use it to price a zero coupon bond and a fixed rate bond.

All the input data and dates are taken from the bond pricing example shipped with QuantLib.

```
#we start with date parameters
fixingDays <- 3
settlementDays <- 3
settlementDate <- as.Date('2008-09-18')
todaysDate <- settlementDate - fixingDays
```

Fixed Income in RQuantLib

Examples: Bond pricing

```
#set up bond discounting term structure
lengths <- c(5, 6, 7, 16, 48)
coupons <- c(0.02375, 0.04625, 0.03125,
             0.04000, 0.04500)
marketQuotes <- c(100.390625, 106.21875,
                  100.59375, 101.6875, 102.140625)
dateparams <- list(settlementDays=settlementDays,
                   period=2, dayCounter="ActualActual",
                   businessDayConvention ="Unadjusted")
curveparams <- list(method="ExponentialSplinesFitting",
                    origDate=todaysDate)
bondDsctTsr <- FittedBondCurve(curveparams, lengths,
                               coupons, marketQuotes,
                               dateparams)
```

Fixed Income in RQuantLib

Examples: Bond pricing

```
#Set up a Zero-Coupon Bond
zc.bond.param <- list(
              maturityDate=as.Date('2013-08-15'),
              issueDate=as.Date('2003-08-15'),
              redemption=116.92)
zc.bond.dateparam <- list(
              refDate=todaysDate,
              settlementDays=settlementDays,
              businessDayConvention='Following')
#Call the pricing function
ZeroCouponBond(zc.bond.param,
               bondDsctTsr,
               zc.bond.dateparam)
```

Fixed Income in RQuantLib

Examples: Bond pricing

```
#Set up a Fixed-Coupon Bond
fixed.bond.param <- list(
                    maturityDate=as.Date('2017-05-15'),
                    issueDate=as.Date('2007-05-15'),
                    redemption=100,
                    effectiveDate=as.Date('2007-05-15'))
fixed.bond.dateparam <- list(
                    settlementDays=settlementDays,
                    dayCounter='ActualActual',
                    period='Semiannual',
                    businessDayConvention='Unadjusted',
                    terminationDateConvention='Unadjusted',
                    dateGeneration='Backward',
                    endOfMonth=0)
fixed.bond.coupon <- c(0.045)
#Call the pricing function
FixedRateBond(fixed.bond.param, fixed.bond.coupon,
              bondDsctTsr, fixed.bond.dateparam)
```

Fixed Income in RQuantLib

Examples: Convertible Bond from Matlab's Fixed Income Toolbox

Perform a spread effect analysis of a 4%-coupon convertible bond callable at 110 at the end of the second year, maturing at par in 5 years, with yield to maturity of 5% and spread (of YTM versus 5-year treasury) of 0, 50, 100, and 150 basis points. The underlying stock pays no dividend.

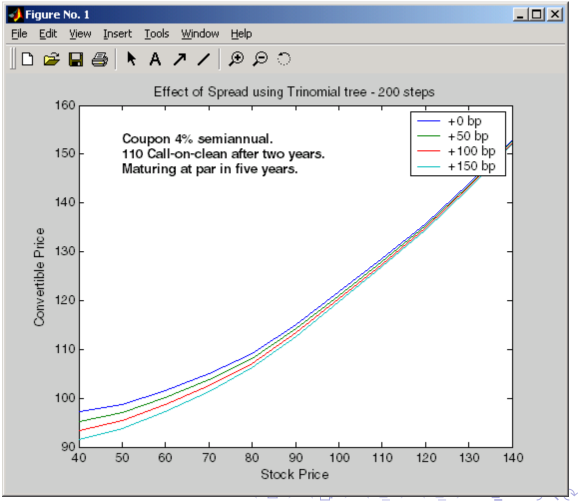
```
RiskFreeRate = 0.05; Sigma        = 0.3;
ConvRatio    = 1;    NumSteps     = 200;
IssueDate    = datenum('2-Jan-2002');
Settle       = datenum('2-Jan-2002');
Maturity     = datenum('2-Jan-2007');
CouponRate   = 0.04;    Period       = 2; Basis        = 1; EndMonthRule = 1;
DividendType = 0; DividendInfo = [];
CallInfo     = [datenum('2-Jan-2004'), 110];
CallType     = 1; TreeType     = 1;
% Nested loop accross prices and static spread dimensions to compute convertible prices.
for j = 0:0.005:0.015;
StaticSpread = j;
      for i = 0:10:100
          Price = 40+i;
          [CbMatrix, UndMatrix, DebtMatrix, EqtyMatrix] =  cbprice(RiskFreeRate, StaticSpread, Sigma, Price, ConvRatio, NumSteps, IssueDate, Settle, Maturity, CouponRate, Period, Basis, EndMonthRule, DividendType, DividendInfo, CallType, CallInfo, TreeType);
           convprice(i/10+1,j*200+1) =  CbMatrix(1,1);
           stock(i/10+1,j*200+1)     =  Price;
        end
end
```

Fixed Income in RQuantLib

Examples: Convertible Bond from Matlab's Fixed Income Toolbox

Source: http://www.mathworks.com/access/helpdesk/help/toolbox/finfixed/cbprice.html

```
plot(stock, convprice);
legend({'+0 bp'; '+50 bp'; '+100 bp'; '+150 bp'});
title ('Effect of Spread using Trinomial tree - 200 steps')
xlabel('Stock Price');
ylabel('Convertible Price');
text(50, 150, ['Coupon 4% semiannual.', sprintf('\n'), ...
    '110 Call-on-clean after two years.' sprintf('\n'), ...
    'Maturing at par in five years.'],'fontweight','Bold')
```



Fixed Income in RQuantLib

Examples: Convertible Bond from Matlab's Fixed Income Toolbox

Doing it in R using RQuantLib....

```
#set up a flat risk free curve
params <- list(tradeDate=as.Date("2002-01-02"), settleDate=as.Date("2002-01-02"),
               interpWhat="discount", interpHow="loglinear")
RiskFreeRate <- DiscountCurve(params, list(flat=0.05),times)
#parameters of the convertible bond
ConvRatio <- 1
issueDate <- as.Date("2002-01-02")
settleDate <- as.Date("2002-01-02")
maturityDate <- as.Date("2007-01-02")
dividendYield <- DiscountCurve(params, list(flat=0.01),times)
dividendSchedule <- data.frame(Type=character(0), Amount=numeric(0),
                               Rate=numeric(0), Date=as.Date(character(0)))
callabilitySchedule <- data.frame(Price=110, Type=0, Date=as.Date("2004-01-02"))
coupon <- 0.04
dateparams <- list(settlementDays=3, period="Semiannual", todayDate=issueDate)
bondparams <- list(exercise="eu", faceAmount=100,
                   divSch=dividendSchedule,
                   callSch=callabilitySchedule,
                   redemption=100,
                   creditSpread=0.005,
                   conversionRatio=ConvRatio,
                   issueDate=issueDate,
                   maturityDate=maturityDate)
```

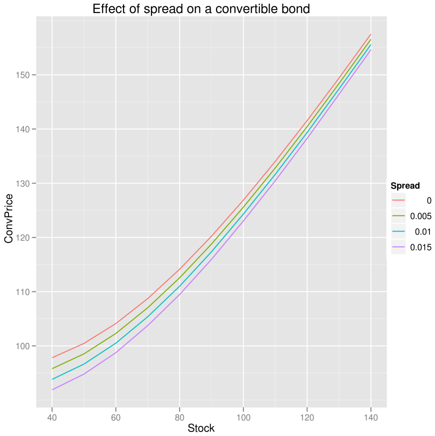
Fixed Income in RQuantLib

Examples: Convertible Bond from Matlab's Fixed Income Toolbox

```
#arguments to construct a BlackScholes process and set up the binomial pricing process
#engine for this bond.
Sigma <- 0.3
process <- list(underlying=40, divYield=dividendYield,
                rff=RiskFreeRate, volatility=Sigma)
#loop through underlying price and spread to produce similar analysis to Matlab
ret <- data.frame()
for (s in c(0, 0.005, 0.010, 0.015)){
   x <- c()
   y <- c()
   i <- 1
   for (p in seq(0, 100, by = 10)) {
      process$underlying <- 40+p
      bondparams$creditSpread <- s
      t <- ConvertibleFixedCouponBond(bondparams,
                                      coupon,
                                      process,
                                      dateparams)
      x[i] <- p + 40
      y[i] <- t$cleanPrice
      i <- i + 1
   }
   z <- rep(s, 11)
   ret <- rbind(ret, data.frame(Stock=x,ConvPrice=y,z))
}
```

Fixed Income in RQuantLib
Examples: Convertible Bond from Matlab's Fixed Income Toolbox

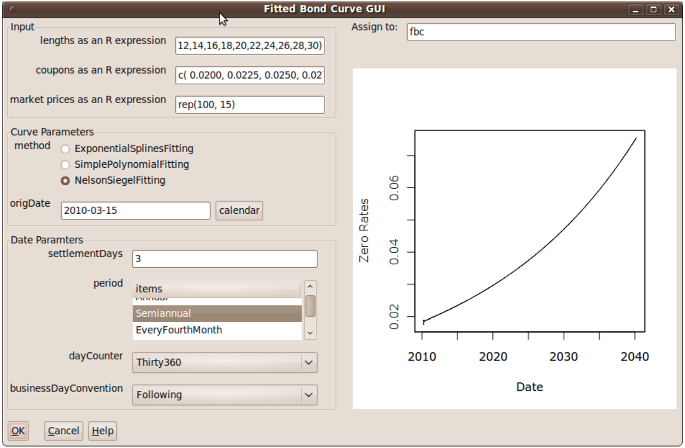
```
#plot the result
>library(ggplot2)
>p <- ggplot(ret, aes(Stock,ConvPrice, colour=factor(z)))
>p + geom_line() + scale_colour_discrete("Spread")
+ opts(title='Effect of spread on a convertible bond'
```



Fixed Income in RQuantLib

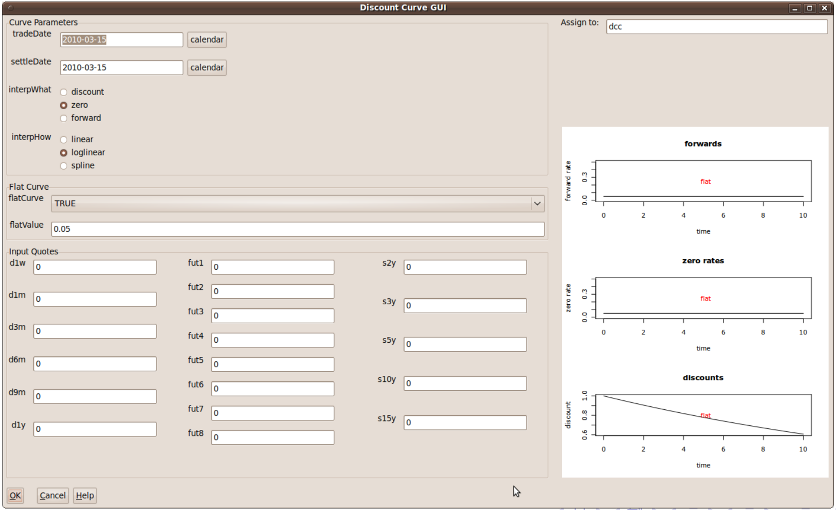
Graphical User Interface: Fitted Curve

RQuantLibGUI provides a graphical user interface via the 'traitr' package by John Verzani.



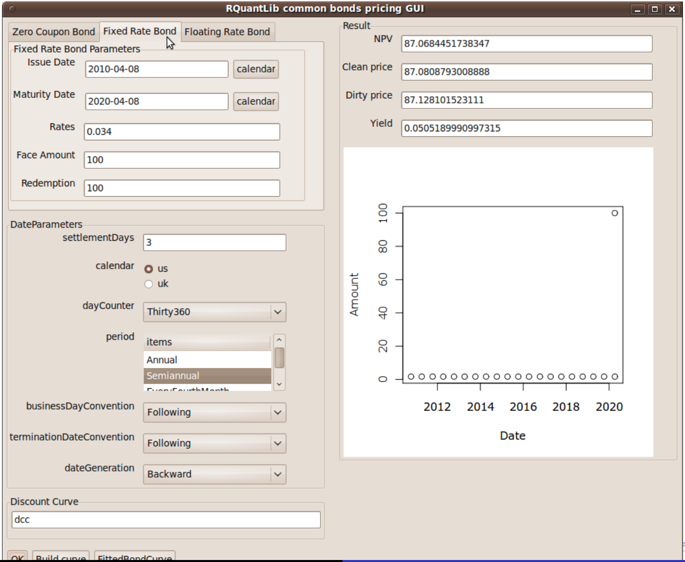
Fixed Income in RQuantLib

Graphical User Interface: Discount Curve



Fixed Income in RQuantLib

Graphical User Interface: Bonds



Outline

Summary and Outlook

- QuantLib represents a decade of work leading to the recent 1.0 release.
- RQuantLib (still) exposes only a subset of the available functionality.
- The conversion to the new Rcpp API (just completed, release pending) should make additions easier.
- Next steps we are thinking about
  - Expanding the GUIs to the option pricers
  - And of course adding more products and QuantLib features
- We welcome feedback as well as contributions – just register at the R-Forge project site.
- Thank you!