Financial Toolbox Release Notes

These Release Notes summarize the changes introduced in the latest version of the Financial Toolbox. The following Version 2.5 topics are discussed in these Release Notes:

If you are updating your Financial Toolbox software from a version prior to Version 2.4.1, refer to

Printing the Release Notes

If you would like to print the Release Notes, you can link to a PDF version.

Contents

4 Financial Toolbox 2.3 Release Notes

1

Financial Toolbox 2.5 Release Notes

New and Changed Features

Version 2.5 introduces a set of financial statistical computation routines that compute values, such as mean and covariance, when there are missing data elements within a larger data set. These routines implement the Expectation Conditional Maximization (ECM) algorithm with various options that depend upon the percentage of missing at random (MAR) data within the data set. The table below lists the functions that implement the ECM algorithm in the Financial Toolbox.

New Statistical Functions

The following ECM functions have been added at this release.

Expectation Conditional Maximization

`ecmnfish`	Fisher information matrix
`ecmnhess`	Hessian of negative log-likelihood function
`ecmninit`	Initial mean and covariance
`ecmnmle`	Mean and covariance of incomplete multivariate normal data
`ecmnobj`	Negative log-likelihood function
`ecmnstd`	Standard errors for mean and covariance of incomplete data

Known Software and Documentation Problems

At publication time, the Financial Toolbox had no significant known open bugs. The Bug Reports interface on the MathWorks Web site will be updated if any important bugs become known to the MathWorks after the publication of these release notes.

2

Financial Toolbox 2.4.1 Release Notes

New and Changed Features

Functions that accept an optional `Basis` argument now allow bases 0 - 7. The table below provides definitions for these bases values.

Basis Value	Meaning	Description
0 (default)	actual/actual	Actual days held over actual days in coupon period. Denominator is 365 in most years and 366 in a leap yer.
1	30/360 (SIA)	Each month contains 30 days; a year contains 360 days. Payments are adjusted for bonds that pay coupons on the last day of February.
2	actual/360	Actual days held over 360.
3	actual/365	Actual days held over 365, even in leap years.
4	30/360 PSA (Public Securities Association)	Each month contains 30 days; a year contains 360 days. If the last date of the period is the last day of February, the month is extended to 30 days.
5	30/360 ISDA (International Swap Dealers Association)	Variant of 30/360 with slight differences for calculating number of days in a month.
6	30/360 European	Variant of 30/360 used primarily in Europe.
7	actual/365 Japanese	All years contain 365 days. Leap days are ignored.

New Function

The following function has been added at this release.

`daysadd`	Date away from a starting date for any day-count basis.

3

Financial Toolbox 2.4 Release Notes

New Features

This section summarizes the new features and enhancements introduced in the Financial Toolbox Version 2.4.

New Portfolio Management Capabilities

The new portfolio management capabilites in Version 2.4 enable you to compute active returns and the tracking error efficient frontier.

Suppose you wish to identify an efficient set of portfolios that minimize the variance of the difference in returns with respect to a given target portfolio, subject to a given expected excess return. The mean and standard deviation of this excess return are often called the *active return* and *active risk*, respectively. Active risk is sometimes referred to as the tracking error. Since the objective is to track a given target portfolio as closely as possible, the resulting set of portfolios is sometimes referred to as the tracking error efficient frontier.

Two new functions have been added to support these capabilities:

`abs2active`	Convert constraints from absolute format to active format
`active2abs`	Convert constraints from active format to absolute format

Modified Functions

Portfolio Analysis Functions

The functions `ret2tick` and `tick2ret` have been modified slightly. Each function can now accept an additional input argument, `Method`, which allows you to specify a character string indicating the method used to convert between asset returns and prices.

Functions for Pricing and Analyzing Derivatives

The Black's model and Black-Scholes' model functions have been rewritten and clarified.

Black's Model

For the `blkprice` function, the first argument has been renamed `Price` and has been defined as the price of a futures contract.

The `blkimpv` function now allows the additional optional input argument `Class` that allows you to specify the option type (put or call).

Black-Scholes Model

With the exception of `blsimpv`, all Black-Scholes functions have a standard set of input arguments.

`Price`	Current price of the underlying asset.
`Strike`	Exercise price of the option.
`Rate`	Annualized, continuously compounded risk-free rate of return over the life of the option, expressed as a positive decimal number.
`Time`	Time to expiration of the option, expressed in years.

`Volatility`	Annualized asset price volatility (annualized standard deviation of the continuously compounded asset return), expressed as a positive decimal number.
`Yield`	(Optional) Annualized, continuously compounded yield of the underlying asset over the life of the option, expressed as a decimal number. (Default = 0.) For example, for options written on stock indices, `Yield` could represent the dividend yield. For currency options, `Yield` could be the foreign risk-free interest rate.

The syntax for `blsimpv` is now

```
Volatility = blsimpv(Price, Strike, Rate, Time, Value, Limit, ...
Yield, Tolerance, Class)
```

where `Price`, `Strike`, `Rate`, and `Time` are as above. New arguments `Value`, `Limit`, and `Yield` are generalizations of their previous counterparts `Call`, `MaxIterations`, and `DividendRate`.

`Value`	Price of a European option from which the implied volatility of the underlying asset is derived.
`Limit`	(Optional) Positive scalar representing the upper bound of the implied volatility search interval. If `Limit` is empty or unspecified, the default = 10, or 1000% per annum.
`Yield`	(Optional) Annualized, continuously compounded yield of the underlying asset over the life of the option, expressed as a decimal number. (Default = 0.) For example, for options written on stock indices, `Yield` could represent the dividend yield. For currency options, `Yield` could be the foreign risk-free interest rate.

The additional optional input argument `Class` allows you to specify the option type (put or call).

Day-Count Basis Argument

Functions that accept an optional `Basis` argument allow bases 0 - 3 only. Bases 4 - 7 are not allowed.

Clarification of Existing Documentation

Rewritten Function Descriptions

The descriptions of the `amortize` and `portsim` functions have been rewritten to clarify how they work. There are no changes to the operation of the functions themselves.

4

Financial Toolbox 2.3 Release Notes

New Features

Version 2.3 of the Financial Toolbox primarily provides support for Version 1.0 of the Fixed-Income Toolbox.

New Functions

Various categories of functions in the toolbox have been augmented with new functions needed by the Fixed-Income Toolbox. These functions are also useful in other contexts.

Date and Time Components

`thirdwednesday`	Third Wednesday of the month

Date Conversion

`dec2thirtytwo`	Decimal quotation to thirty-second
`thirtytwo2dec`	Thirty-second quotation to decimal

Financial Dates

`days360e`	Days between dates based on 360-day year (European compliant).
`days360isda`	Days between dates based on 360-day year (ISDA compliant).
`days360psa`	Days between dates based on 360-day year (PSA compliant).

Revised Functions

Several financial date functions have an additional argument, `weekend`, which allows you to specify which dates of the week constitute the weekend using your local calendar. These functions effectively internationalize the business day functions within the toolbox.

Financial Date Functions with weekend Argument

`busdate`	Next or previous business day.
`fbusdate`	First business date of month.
`isbusday`	True for dates that are business days.
`lbusdate`	Last business date of month.

Additional Bases for Functions

The basis of a bond refers to the basis or day-count convention for a bond. Basis is normally expressed as a fraction in which the numerator determines the number of days between two dates, and the denominator determines the number of days in the year. Prior releases of this toolbox supported four varieties of basis computation:

Basis Argument	Interpretation
0	actual/actual
1	30/360
2	actual/360
3	actual/365

For functions involved with Fixed-Income Toolbox computations, the set of supported bases has been augmented with four additional bases:

Basis Argument	Interpretation
4	30/360 (Public Securities Association [PSA] compliant)
5	30/360 (International Swap Dealers Association [ISDA] compliant)
6	30/360 (European) compliant
7	actual/365 (Japanese) compliant

Upgrading From an Earlier Release

Below is an upgrade issue involved in upgrading to the Financial Toolbox 2.3 from the Financial Toolbox 2.1 (which was released prior to Release 13) .

Obsolete BDT Functions

The functions `bdtbond` and `bdttrans` are obsolete, and their descriptions have been removed from the documentation. These functions have been replaced by `BDT` functions in the Financial Derivatives Toolbox. For compatibility purposes, the obsolete functions remain in the product. Type `help function_name` at the MATLAB command line for a description.