
AIMMS Function Reference - Financial Functions - Securities

This file contains only one chapter of the book. For a free download of the complete book in pdf format, please visit www.aimms.com

Financial Functions - Securities

There are several types of securities, each with its own features and scheduled cash flows. Cash flows can be scheduled at the end of every coupon period or just at the end of the security's life. If we see a security as an investment, its yield can be viewed as the internal rate of return. The cash flows of a security can consist of periodic payments (equal to a certain percentage of the par value), the coupons, and the future value of the security. In general, the general cash flow equation

Securities

$$v_p(1+r)^N + p \sum_{i=1}^N (1+r)^{i-1} + v_f = 0$$

where v_p is the present value, v_f is the future value, N the number of periods, p is a constant periodic payment and r is the constant interest rate, holds. AIMMS provides functions for the most common types of securities like treasury bills and bonds. However, the present value, future value, periodic payments, number of periods and interest rate are different for each security type.

We distinguish three main types of securities:

Security types

- securities with zero coupon periods (discounted securities),
- securities with one coupon period (at maturity), and
- securities with multiple coupon periods

In the case of discounted (or zero coupon) securities such as treasury bills, there are no periodical payments. The only positive cash flow is a fixed redemption at the end of the security's life. Therefore, only the value of this redemption and the investment made for the security determine its yield. In this case, the present value is equal to the price $-P$, the price at which the security is bought at the settlement date, there are 0 periods (so no periodic payments), and the future value at the maturity date is equal to the redemption R . Thus the general cash flow equation reduces to

Discounted securities

$$-P(1+r_y f_{SM}) + R = 0$$

where r_y is the annual yield of the security, and f_{SM} is the difference (in fractions of years) between the settlement and maturity date, computed with respect to the specified day count basis [method](#).

Commonly with discounted securities, the yield is not expressed in terms of the price, but in terms of the fixed redemption. The discount rate is the increase in value per year as a percentage of the redemption. The relationship between the yield r_y and the discount rate r_d is given by

Discount rate

$$1 + r_y f_{SM} = \frac{1}{1 - r_d f_{SM}}$$

which leads to the following equivalent relation between price and redemption

$$-P + R(1 - r_d f_{SM}) = 0$$

A treasury bill is a discounted security with less than one year from settlement until maturity, the number of days in one year is fixed at 360 and redemption is fixed at 100.

Treasury bills

AIMMS supports the following functions for securities with zero coupon periods:

Functions for discounted securities

- `SecurityDiscountedPrice`
- `SecurityDiscountedRedemption`
- `SecurityDiscountedYield`
- `SecurityDiscountedRate`
- `TreasuryBillPrice`
- `TreasuryBillYield`
- `TreasuryBillBondEquivalent`

Securities that only pay interest at maturity can be seen as securities with only one coupon period, where the accrued interest increases linearly in time until it is paid (when the security expires), and the redemption equals the par value of the security. In the general cash flow equation,

One-coupon securities

- the present value

$$v_p = -P - v_{par} r_c f_{IS},$$

where P is the price of the account at settlement and f_{IS} is the difference between the issue and settlement date (in fraction of years) with respect to the specified day count basis `method`, to account for the accrued interest from the issue date until settlement,

- the periodic payment

$$p = v_{par} r_y f_{IM},$$

where r_y is the annual yield and f_{IM} is the difference between the issue and maturity date (in fraction of years) with respect to the specified day count basis `method`, and

- the interest rate

$$r = r_y f_{SM},$$

where f_{SM} is the difference between the settlement and maturity date (in fraction of years) with respect to the specified day count basis `method`.

This results in the following equation for securities with one coupon period:

$$(-P - v_{par}r_c f_{IS})(1 + r_y f_{SM}) + v_{par}r_y f_{IM} + v_{par} = 0$$

AIMMS supports the following functions for securities with one coupon period:

- SecurityMaturityPrice
- SecurityMaturityCouponRate
- SecurityMaturityYield
- SecurityMaturityAccruedInterest

Functions for one-coupon securities

For securities with multiple coupon periods, interest will be accrued linearly during and paid at the end of each coupon period (i.e. at the coupon date). In the general cash flow equation

Multi-coupon securities

- the number of periods

$$N = \lceil f f_{SM} \rceil,$$

where f is the coupon frequency (number of coupon periods per year), and f_{SM} the difference between settlement and maturity date (in fraction of years) with respect to the specified day count basis **method**,

- the present value

$$v_p = -P - v_{par} \frac{r_c}{f} \frac{f_{PS}}{f_{PN}},$$

where P is the price of the security at settlement, v_{par} the par value of the security, r_c the annual coupon rate, f_{PS} the difference (in fraction of years) between the previous coupon and settlement date, and f_{PN} the difference between the previous and next coupon date, both with respect to the specified day count basis **method**,

- the periodic payment

$$p = v_{par} \frac{r_c}{f}$$

- the interest rate

$$r = \frac{r_y}{f},$$

where r_y is the annual yield.

This results in the following equation for securities with multiple coupon periods:

$$\left(-P - v_{par} \frac{r_c}{f} \frac{f_{PS}}{f_{PN}}\right)^{N-1+\frac{f_{SN}}{f_{PN}}} + \sum_{i=1}^N v_{par} \frac{r_c}{f} \left(1 + \frac{r_y}{f}\right)^{N-i} + R = 0$$

AIMMS supports the following functions for securities with multiple coupon periods:

Functions for multi-coupon securities

- SecurityCouponNumber
- SecurityCouponPreviousDate
- SecurityCouponNextDate
- SecurityCouponDays
- SecurityCouponDaysPreSettlement
- SecurityCouponDaysPostSettlement
- SecurityPeriodicPrice
- SecurityPeriodicRedemption
- SecurityPeriodicCouponRate
- SecurityPeriodicYieldAll
- SecurityPeriodicYield
- SecurityPeriodicAccruedInterest
- SecurityPeriodicDuration
- SecurityPeriodicDurationModified

SecurityDiscountedPrice

The function `SecurityDiscountedPrice` returns the price of a discounted security at settlement date.

```
SecurityDiscountedPrice(
    SettlementDate,      ! (input) scalar string expression
    MaturityDate,        ! (input) scalar string expression
    Redemption,          ! (input) numerical expression
    DiscountRate,        ! (input) numerical expression
    [Basis]              ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be given in a date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

Redemption

The amount repaid at maturity date. *Redemption* must be a positive real number.

DiscountRate

The rate the security's value increases per year as a percentage of the redemption value. *DiscountRate* must be a positive real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityDiscountedPrice` returns the price of the security at settlement date.

Remarks:

- This function can be used in an objective function or constraint and the input parameters *Redemption* and *DiscountRate* can be used as a variable.
- The function `SecurityDiscountedPrice` is similar to the Excel function `PRICEDISC`.

See also:

Day count basis [methods](#). General [equations](#) for discounted securities.

SecurityDiscountedRedemption

The function `SecurityDiscountedRedemption` returns the repayment at maturity date of a discounted security.

```
SecurityDiscountedRedemption(
    SettlementDate,      ! (input) scalar string expression
    MaturityDate,        ! (input) scalar string expression
    Price,                ! (input) numerical expression
    DiscountRate,        ! (input) numerical expression
    [Basis]               ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be given in a date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

Price

The price of the security at settlement date. *Price* must be a positive real number.

DiscountRate

The rate the security's value increases per year as a percentage of the redemption value. *DiscountRate* must be a positive real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityDiscountedRedemption` returns the amount paid at maturity date.

Remarks:

- This function can be used in an objective function or constraint and the input parameters *Price* and *DiscountRate* can be used as a variable.
- The function `SecurityDiscountedRedemption` is similar to the Excel function RECEIVED.

See also:

Day count basis [methods](#). General [equations](#) for discounted securities.

SecurityDiscountedYield

The function `SecurityDiscountedYield` returns the yield of a discounted security at maturity date.

```
SecurityDiscountedYield(
    SettlementDate,      ! (input) scalar string expression
    MaturityDate,       ! (input) scalar string expression
    Price,              ! (input) numerical expression
    Redemption,         ! (input) numerical expression
    [Basis]             ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be given in a date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

Price

The price of the security at settlement date. *Price* must be a positive real number.

Redemption

The amount repaid at maturity date. *Redemption* must be a positive real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityDiscountedYield` returns the annual rate the security's value increases as a percentage of the price.

Remarks:

- This function can be used in an objective function or constraint and the input parameters *Price* and *Redemption* can be used as a variable.
- The function `SecurityDiscountedYield` is similar to the Excel function `YIELDDISC`.

See also:

Day count basis [methods](#). General [equations](#) for discounted securities.

SecurityDiscountedRate

The function `SecurityDiscountedRate` returns the discount rate of a discounted security.

```
SecurityDiscountedRate(
    SettlementDate,    ! (input) scalar string expression
    MaturityDate,      ! (input) scalar string expression
    Price,             ! (input) numerical expression
    Redemption,        ! (input) numerical expression
    [Basis]            ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be given in a date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

Price

The price of the security at settlement date. *Price* must be a positive real number.

Redemption

The amount repaid at maturity date. *Redemption* must be a positive real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityDiscountedRate` returns the annual rate the security's value increases as a percentage of the redemption value.

Remarks:

- This function can be used in an objective function or constraint and the input parameters *Price* and *Redemption* can be used as a variable.
- The function `SecurityDiscountedRate` is similar to the Excel function `DISC`.

See also:

Day count basis [methods](#). General [equations](#) for discounted securities.

TreasuryBillPrice

The function `TreasuryBillPrice` returns the price of a Treasury bill at settlement date. A Treasury bill is a discounted security with less than one year from settlement until maturity, the number of days in one year is fixed at 360 and redemption is fixed at 100.

```
TreasuryBillPrice(
    SettlementDate,      ! (input) scalar string expression
    MaturityDate,        ! (input) scalar string expression
    DiscountRate         ! (input) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be given in a date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

DiscountRate

The discount rate of the security as a percentage of the redemption. *DiscountRate* must be a positive real number.

Return value:

The function `TreasuryBillPrice` returns the price of a Treasury bill at settlement date.

Remarks:

- This function can be used in an objective function or constraint and the input parameter *DiscountRate* can be used as a variable.
- The function `TreasuryBillPrice` is similar to the Excel function `TBILLPRICE`.

See also:

General [equations](#) for discounted securities.

TreasuryBillYield

The function `TreasuryBillYield` returns the yield of a Treasury bill at settlement date. A Treasury bill is a discounted security with less than one year from settlement until maturity, the number of days in one year is fixed at 360 and redemption is fixed at 100.

```
TreasuryBillYield(  
    SettlementDate,      ! (input) scalar string expression  
    MaturityDate,       ! (input) scalar string expression  
    Price                ! (input) numerical expression  
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be given in a date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

Price

The price the security is worth at this moment. *Price* must be a positive real number.

Return value:

The function `TreasuryBillYield` returns the annual rate the Treasury bill's value increases as a percentage of the price.

Remarks:

- This function can be used in an objective function or constraint and the input parameter *Price* can be used as a variable.
- The function `TreasuryBillYield` is similar to the Excel function `TBILLYIELD`.

See also:

General [equations](#) for discounted securities.

TreasuryBillBondEquivalent

The function `TreasuryBillBondEquivalent` returns the bond equivalent yield of a treasury bill. A Treasury bill is a discounted security with less than one year from settlement until maturity, the number of days in one year is fixed at 360 and redemption is fixed at 100.

```
TreasuryBillBondEquivalent(
    SettlementDate,    ! (input) scalar string expression
    MaturityDate,      ! (input) scalar string expression
    DiscountRate       ! (input) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be given in a date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

DiscountRate

The discount rate of the security as a percentage of the redemption. *DiscountRate* must be a positive real number.

Return value:

The function `TreasuryBillBondEquivalent` returns the bond equivalent yield of a Treasury bill.

Remarks:

- This function can be used in an objective function or constraint and the input parameter *DiscountRate* can be used as a variable.
- The function `TreasuryBillBondEquivalent` is similar to the Excel function `TBILLEQ`.

See also:

General [equations](#) for discounted securities.

SecurityMaturityPrice

The function `SecurityMaturityPrice` returns the price at settlement date of a security that pays interest at maturity.

```
SecurityMaturityPrice(
  IssueDate,          ! (input) scalar string expression
  SettlementDate,    ! (input) scalar string expression
  MaturityDate,      ! (input) scalar string expression
  ParValue,          ! (input) numerical expression
  CouponRate,       ! (input) numerical expression
  Yield,            ! (input) numerical expression
  [Basis]           ! (optional) numerical expression
)
```

Arguments:

IssueDate

The date of issue of the security. *IssueDate* must be given in date format.

SettlementDate

The date of settlement of the security. *SettlementDate* must also be in date format and must be a date after *IssueDate*.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

CouponRate

The annual interest rate of the security as a percentage of the par value. *CouponRate* must be a nonnegative real number.

Yield

The yield of the security. *Yield* must be a nonnegative real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityMaturityPrice` returns the price of the security at settlement date.

Remarks:

- This function can be used in an objective function or constraint and the input parameters *ParValue*, *CouponRate*, and *Yield* can be used as a variable.

- The function SecurityMaturityPrice is similar to the Excel function PRICE-MAT.

See also:

Day count basis [methods](#). General [equations](#) for securities with one coupon.

SecurityMaturityCouponRate

The function `SecurityMaturityCouponRate` returns the coupon rate of a security that pays interest at maturity.

```
SecurityMaturityCouponRate(
    IssueDate,          ! (input) scalar string expression
    SettlementDate,    ! (input) scalar string expression
    MaturityDate,      ! (input) scalar string expression
    ParValue,          ! (input) numerical expression
    Price,             ! (input) numerical expression
    Yield,             ! (input) numerical expression
    [Basis]            ! (optional) numerical expression
)
```

Arguments:

IssueDate

The date of issue of the security. *IssueDate* must be given in date format.

SettlementDate

The date of settlement of the security. *SettlementDate* must also be in date format and must be a date after *IssueDate*.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

Price

The price of the security at settlement date. *Price* must be a positive real number.

Yield

The yield of the security. *Yield* must be a nonnegative real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityMaturityCouponRate` returns the annual interest rate of the security as a percentage of the par value.

Remarks:

This function can be used in an objective function or constraint and the input parameters *ParValue*, *Price*, and *Yield* can be used as a variable.

See also:

Day count basis **methods**. General **equations** for securities with one coupon.

SecurityMaturityYield

The function `SecurityMaturityYield` returns the yield of a security that pays interest at maturity.

```
SecurityMaturityYield(
    IssueDate,          ! (input) scalar string expression
    SettlementDate,    ! (input) scalar string expression
    MaturityDate,      ! (input) scalar string expression
    ParValue,          ! (input) numerical expression
    Price,             ! (input) numerical expression
    CouponRate,       ! (input) numerical expression
    [Basis]            ! (optional) numerical expression
)
```

Arguments:

IssueDate

The date of issue of the security. *IssueDate* must be given in date format.

SettlementDate

The date of settlement of the security. *SettlementDate* must also be in date format and must be a date after *IssueDate*.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

Price

The price of the security at settlement date. *Price* must be a positive real number.

CouponRate

The annual interest rate of the security as a percentage of the par value. *CouponRate* must be a nonnegative real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityMaturityYield` returns the annual rate the security's value increases as a percentage of the price.

Remarks:

- This function can be used in an objective function or constraint and the input parameters *ParValue*, *Price*, and *CouponRate* can be used as a variable.
- The function `SecurityMaturityYield` is similar to the Excel function `YIELD-MAT`.

See also:

Day count basis [methods](#). General [equations](#) for securities with one coupon.

SecurityMaturityAccruedInterest

The function `SecurityMaturityAccruedInterest` returns the accrued interest for a security that pays interest at maturity.

```
SecurityMaturityAccruedInterest(
    IssueDate,          ! (input) scalar string expression
    SettlementDate,    ! (input) scalar string expression
    ParValue,          ! (input) numerical expression
    CouponRate,        ! (input) numerical expression
    [Basis]             ! (optional) numerical expression
)
```

Arguments:

IssueDate

The date of issue of the security. *IssueDate* must be given in date format.

SettlementDate

The date of settlement of the security. *SettlementDate* must also be in date format and must be a date after *IssueDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

CouponRate

The annual interest rate of the security as a percentage of the par value. *CouponRate* must be a nonnegative real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityMaturityAccruedInterest` returns the interest accrued from issue date until settlement date.

Remarks:

- This function can be used in an objective function or constraint and the input parameters *CouponRate* and *ParValue* can be used as a variable.
- The function `SecurityMaturityAccruedInterest` is similar to the Excel function `ACCRINTM`.

See also:

Day count basis [methods](#). General [equations](#) for securities with one coupon.

SecurityCouponNumber

The function `SecurityCouponNumber` returns the number of coupons from settlement date and maturity date of a security that pays interest at the end of each coupon period.

```
SecurityCouponNumber(  
    SettlementDate,    ! (input) scalar string expression  
    MaturityDate,      ! (input) scalar string expression  
    Frequency,         ! (input) numerical expression  
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

Return value:

The function `SecurityCouponNumber` returns the number of coupon payments from the settlement date until the maturity date.

Remarks:

The function `SecurityCouponNumber` is similar to the Excel function `COUPNUM`.

See also:

Day count basis [methods](#). General [equations](#) for securities with multiple coupons.

SecurityCouponPreviousDate

The function `SecurityCouponPreviousDate` returns the last coupon-date previous to settlement date of a security that pays interest at the end of each coupon period.

```
SecurityCouponPreviousDate(  
    SettlementDate,    ! (input) scalar string expression  
    MaturityDate,      ! (input) scalar string expression  
    Frequency          ! (input) numerical expression  
    PreviousDate       ! (output) string parameter  
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

PreviousDate

The date on which the coupon period, in which the settlement date falls, starts and on which the previous coupon period ends.

Remarks:

The function `SecurityCouponPreviousDate` is similar to the Excel function `COUPPCD`.

See also:

General [equations](#) for securities with multiple coupons.

SecurityCouponNextDate

The function SecurityCouponNextDate returns the first coupon-date next to settlement date of a security that pays interest at the end of each coupon period.

```
SecurityCouponNextDate(  
    SettlementDate,      ! (input) scalar string expression  
    MaturityDate,       ! (input) scalar string expression  
    Frequency            ! (input) numerical expression  
    NextDate            ! (output) string parameter  
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

NextDate

The date on which the coupon period ends and on which the next coupon period starts.

Remarks:

The function SecurityCouponNextDate is similar to the Excel function COUP-NCD.

See also:

General [equations](#) for securities with multiple coupons.

SecurityCouponDays

The function `SecurityCouponDays` returns the number of days of the coupon period in which settlement date falls. In other words the number of days from the last coupon-date previous to settlement date until the first coupon-date next to settlement date of a security that pays interest at the end of each coupon period.

```
SecurityCouponDays(
    SettlementDate,      ! (input) scalar string expression
    MaturityDate,       ! (input) scalar string expression
    Frequency,          ! (input) numerical expression
    [Basis]              ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityCouponDays` returns the number of days of the coupon period in which the settlement date falls.

Remarks:

The function `SecurityCouponDays` is similar to the Excel function `COUPDAYS`.

See also:

Day count basis [methods](#). General [equations](#) for securities with multiple coupons.

SecurityCouponDaysPreSettlement

The function `SecurityCouponDaysPreSettlement` returns the number of days from the last coupon-date previous to settlement date until settlement date of a security that pays interest at the end of each coupon period.

```
SecurityCouponDaysPreSettlement(
    SettlementDate,      ! (input) scalar string expression
    MaturityDate,       ! (input) scalar string expression
    Frequency,          ! (input) numerical expression
    [Basis]              ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityCouponDaysPreSettlement` returns the number of days from the previous coupon-date until the settlement date, using the specified day-count basis.

Remarks:

The function `SecurityCouponDaysPreSettlement` is similar to the Excel function `COUPDAYBS`.

See also:

Day count basis [methods](#). General [equations](#) for securities with multiple coupons.

SecurityCouponDaysPostSettlement

The function `SecurityCouponDaysPostSettlement` returns the number of days from the first coupon-date next to settlement date until settlement date of a security that pays interest at the end of each coupon period.

```
SecurityCouponDaysPostSettlement(
    SettlementDate,      ! (input) scalar string expression
    MaturityDate,       ! (input) scalar string expression
    Frequency,          ! (input) numerical expression
    [Basis]              ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityCouponDaysPostSettlement` returns the number of days from the first coupon-date next to settlement date until settlement date.

Remarks:

The function `SecurityCouponDaysPostSettlement` is similar to the Excel function `COUPDAYSNC`.

See also:

Day count basis [methods](#). General [equations](#) for securities with multiple coupons.

SecurityPeriodicPrice

The function `SecurityPeriodicPrice` returns the price at settlement date of a security that pays interest at the end of each coupon period.

```
SecurityPeriodicPrice(
    SettlementDate,      ! (input) scalar string expression
    MaturityDate,       ! (input) scalar string expression
    ParValue,           ! (input) numerical expression
    Redemption,         ! (input) numerical expression
    Frequency,          ! (input) numerical expression
    CouponRate,        ! (input) numerical expression
    Yield,              ! (input) numerical expression
    [Basis]             ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

Redemption

The amount repaid for the security at the maturity date. *Redemption* must be a positive real number.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

CouponRate

The annual interest rate of the security as a percentage of the par value. *CouponRate* must be a nonnegative real number.

Yield

The yield of the security. *Yield* must be a nonnegative real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityPeriodicPrice` returns the price of the security at settlement date.

Remarks:

- This function can be used in an objective function or constraint and the input parameters *ParValue*, *Redemption*, *CouponRate*, and *Yield* can be used as a variable.
- The function `SecurityPeriodicPrice` is similar to the Excel function `PRICE`.

See also:

Day count basis [methods](#). General [equations](#) for securities with multiple coupons.

SecurityPeriodicRedemption

The function `SecurityPeriodicRedemption` returns the repayment at maturity date of a security that pays interest at the end of each coupon period.

```
SecurityPeriodicRedemption(
    SettlementDate,    ! (input) scalar string expression
    MaturityDate,      ! (input) scalar string expression
    ParValue,          ! (input) numerical expression
    Price,             ! (input) numerical expression
    Frequency,         ! (input) numerical expression
    CouponRate,       ! (input) numerical expression
    Yield,             ! (input) numerical expression
    [Basis]            ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

Price

The price of the security at settlement date. *Price* must be a positive real number.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

CouponRate

The annual interest rate of the security as a percentage of the par value. *CouponRate* must be a nonnegative real number.

Yield

The yield of the security. *Yield* must be a nonnegative real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityPeriodicRedemption` returns the amount repaid for the security at the maturity date.

Remarks:

This function can be used in an objective function or constraint and the input parameters *ParValue*, *Price*, *CouponRate*, and *Yield* can be used as a variable.

See also:

Day count basis [methods](#). General [equations](#) for securities with multiple coupons.

SecurityPeriodicCouponRate

The function `SecurityPeriodicCouponRate` returns the coupon rate of a security that pays interest at the end of each coupon period.

```
SecurityPeriodicCouponRate(
    SettlementDate,    ! (input) scalar string expression
    MaturityDate,      ! (input) scalar string expression
    ParValue,          ! (input) numerical expression
    Price,              ! (input) numerical expression
    Redemption,        ! (input) numerical expression
    Frequency,         ! (input) numerical expression
    Yield,             ! (input) numerical expression
    [Basis]            ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

Price

The price of the security at settlement date. *Price* must be a positive real number.

Redemption

The amount repaid for the security at the maturity date. *Redemption* must be a positive real number.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

Yield

The yield of the security. *Yield* must be a nonnegative real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityPeriodicCouponRate` returns the interest rate per year of the security as a percentage of the par value.

Remarks:

This function can be used in an objective function or constraint and the input parameters *ParValue*, *Price*, *Redemption*, and *Yield* can be used as a variable.

See also:

Day count basis **methods**. General **equations** for securities with multiple coupons.

SecurityPeriodicYieldAll

The procedure `SecurityPeriodicYieldAll` returns the yield(s) of a security that pays interest at the end of each coupon period.

```
SecurityPeriodicYieldAll(
  SettlementDate,      ! (input) scalar string expression
  MaturityDate,       ! (input) scalar string expression
  ParValue,           ! (input) numerical expression
  Price,              ! (input) numerical expression
  Redemption,         ! (input) numerical expression
  Frequency,          ! (input) numerical expression
  CouponRate,        ! (input) numerical expression
  Yield,              ! (output) one-dimensional numerical expression
  NumberSolutions,   ! (output) numerical expression
  [Basis,]            ! (optional) numerical expression
  [Mode]              ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

Price

The price of the security at settlement date. *Price* must be a positive real number.

Redemption

The amount repaid for the security at the maturity date. *Redemption* must be a positive real number.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

CouponRate

The annual interest rate of the security as a percentage of the par value. *CouponRate* must be a nonnegative real number.

Yield

The yield of the security. *Yield* must be a nonnegative real number.

Yield

There is not always a unique solution for yield. Dependent on *Mode* one solution or all the solutions will be given.

NumberSolutions

The number of solutions found. If *Mode* = 0 *NumberSolutions* will always be 1.

Basis

The day-count basis method to be used. The default is 1.

Mode

Indicates whether all the solutions need to be found or just one. *Mode* = 0: the search for solutions stops after one solution is found. *Mode* = 1: the search for solutions continues till all solutions are found.

Remarks:

- When you want to use this procedure in an objective function or constraint you have to use `SecurityPeriodicYield`.
- The function `SecurityPeriodicYieldAll` is similar to the Excel function `YIELD`.

See also:

Day count basis [methods](#). General [equations](#) for securities with multiple coupons.

SecurityPeriodicYield

The function `SecurityPeriodicYield` returns the yield of a security that pays interest at the end of each coupon period. This function uses the procedure `SecurityPeriodicYieldAll` to determine all possible yields and returns the yield that is within the specified bounds.

```
SecurityPeriodicYield(
    SettlementDate,      ! (input) scalar string expression
    MaturityDate,       ! (input) scalar string expression
    ParValue,           ! (input) numerical expression
    Price,              ! (input) numerical expression
    Redemption,         ! (input) numerical expression
    Frequency,          ! (input) numerical expression
    CouponRate,        ! (input) numerical expression
    [Basis,]            ! (optional) numerical expression
    [LowerBound,]      ! (optional) numerical expression
    [UpperBound,]      ! (optional) numerical expression
    [Error]             ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

Price

The price of the security at settlement date. *Price* must be a positive real number.

Redemption

The amount repaid for the security at the maturity date. *Redemption* must be a positive real number.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

CouponRate

The annual interest rate of the security as a percentage of the par value. *CouponRate* must be a nonnegative real number.

Basis

The day-count basis method to be used. The default is 1.

LowerBound

Indicates a minimum for the yield to be accepted by this function. The default is -1 .

UpperBound

Indicates a maximum for the yield to be accepted by this function. The default is 5 .

Error

Indicates whether AIMMS should give an error if multiple solutions are found that satisfy the bounds. *Error* = 0: if multiple solutions are found, return the solution with the smallest absolute value. *Error* = 1: if multiple solutions are found, return an error message. The default is 0.

Return value:

The function `SecurityPeriodicYield` returns the yield of a security that pays interest at the end of each coupon period.

Remarks:

- This function can be used in an objective function or constraint and the input parameters *ParValue*, *Price*, *Redemption*, and *CouponRate* can be used as a variable.
- The function `SecurityPeriodicYield` is similar to the Excel function `YIELD`.

See also:

Day count basis [methods](#). General [equations](#) for securities with multiple coupons.

SecurityPeriodicAccruedInterest

The function `SecurityPeriodicAccruedInterest` returns the accrued interest from the begin of the coupon period until the settlement date for a security that pays interest at the end of each coupon period.

```
SecurityPeriodicAccruedInterest(
    SettlementDate,      ! (input) scalar string expression
    MaturityDate,       ! (input) scalar string expression
    ParValue,           ! (input) numerical expression
    Frequency,          ! (input) numerical expression
    CouponRate,        ! (input) numerical expression
    [Basis]              ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

CouponRate

The annual interest rate of the security as a percentage of the par value. *CouponRate* must be a nonnegative real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityPeriodicAccruedInterest` returns the interest accrued from the begin of the coupon period until settlement date.

Remarks:

This function can be used in an objective function or constraint and the input parameters *ParValue* and *CouponRate* can be used as a variable.

See also:

Day count basis **methods**. General **equations** for securities with multiple coupons.

SecurityPeriodicDuration

The function `SecurityPeriodicDuration` returns the Macauley duration of a security that pays interest at the end of each coupon period. Duration is defined as the weighted average of time it takes to receive a positive cash flow. The present values of the cash flows are used as weights. The duration can be used as a measure of a bond price's response to changes in yield.

```
SecurityPeriodicDuration(
    SettlementDate,      ! (input) scalar string expression
    MaturityDate,       ! (input) scalar string expression
    ParValue,           ! (input) numerical expression
    Redemption,         ! (input) numerical expression
    Frequency,          ! (input) numerical expression
    CouponRate,        ! (input) numerical expression
    Yield,              ! (input) numerical expression
    [Basis]             ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

Redemption

The amount repaid for the security at the maturity date. *Redemption* must be a positive real number.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

CouponRate

The annual interest rate of the security as a percentage of the par value. *CouponRate* must be a nonnegative real number.

Yield

The yield of the security. *Yield* must be a nonnegative real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityPeriodicDuration` returns the Macauley duration of a security that pays interest at the end of each coupon period. Duration is defined as the weighted average of the time it takes to receive a positive cash flow.

Equation:

The Macauley duration D is computed through the equation

$$D = \frac{\left(N - 1 + \frac{f_{SN}}{f_{PN}}\right) \frac{R}{\left(1 + \frac{r_y}{f}\right)^{N-1 + \frac{f_{SN}}{f_{PN}}}} + \sum_{i=1}^N \left(i - 1 + \frac{f_{SN}}{f_{PN}}\right) \frac{v_{par} \frac{r_c}{f}}{\left(1 + \frac{r_y}{f}\right)^{i-1 + \frac{f_{SN}}{f_{PN}}}}}{\frac{R}{\left(1 + \frac{r_y}{f}\right)^{N-1 + \frac{f_{SN}}{f_{PN}}}} + \sum_{i=1}^N \frac{v_{par} \frac{r_c}{f}}{\left(1 + \frac{r_y}{f}\right)^{i-1 + \frac{f_{SN}}{f_{PN}}}}}$$

where all other variables have the same interpretation as in the general [equations](#) for securities with multiple coupons.

Remarks:

- This function can be used in an objective function or constraint and the input parameters *ParValue*, *Redemption*, *CouponRate*, and *Yield* can be used as a variable.
- The function `SecurityPeriodicDuration` is similar to the Excel function `DURATION`.

See also:

Day count basis [methods](#). General [equations](#) for securities with multiple coupons.

SecurityPeriodicDurationModified

The function `SecurityPeriodicDurationModified` returns the modified Macauley duration of a security that pays interest at the end of each coupon period.

```
SecurityPeriodicDurationModified(
  SettlementDate,      ! (input) scalar string expression
  MaturityDate,       ! (input) scalar string expression
  ParValue,           ! (input) numerical expression
  Redemption,         ! (input) numerical expression
  Frequency,          ! (input) numerical expression
  CouponRate,        ! (input) numerical expression
  Yield,              ! (input) numerical expression
  [Basis]             ! (optional) numerical expression
)
```

Arguments:

SettlementDate

The date of settlement of the security. *SettlementDate* must be in date format.

MaturityDate

The date of maturity of the security. *MaturityDate* must also be in date format and must be a date after *SettlementDate*.

ParValue

The starting value of the security at issue date. *ParValue* must be a positive real number.

Redemption

The amount repaid for the security at the maturity date. *Redemption* must be a positive real number.

Frequency

The number of coupon payments in one year. *Frequency* must be 1 (annual), 2 (semi-annual) or 4 (quarterly).

CouponRate

The annual interest rate of the security as a percentage of the par value. *CouponRate* must be a nonnegative real number.

Yield

The yield of the security. *Yield* must be a nonnegative real number.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `SecurityPeriodicDurationModified` returns the modified Macauley duration of a security that pays interest at the end of each coupon period.

Equation:

The modified duration D_{mod} is computed through the equation

$$D_{mod} = \frac{D}{1 + \frac{r_y}{f}}$$

where D is the Macauley duration.

Remarks:

- This function can be used in an objective function or constraint and the input parameters *ParValue*, *Redemption*, *CouponRate*, and *Yield* can be used as a variable.
- The function `SecurityPeriodicDurationModified` is similar to the Excel function `MDURATION`.

See also:

The function `SecurityPeriodicDuration`. Day count basis [methods](#). General [equations](#) for securities with multiple coupons.