Mathos Project

Exploring the computational power for mathematical calculations

By Mathos Project Community

Π

Contents

Mathos Core Library **3**

1 Finance **4**

1.1 Present Value . . . 4
1.1.1 Parameters . . . 4
1.1.2 Example . . . 5
1.2 Net Present Value . . . 5
1.2.1 Parameters . . . 5
1.2.2 Example . . . 5
1.3 Future Value . . . 5
1.3.1 Parameters . . . 6
1.3.2 Example . . . 6
1.4 Future Value Of An Annuity . . . 6
1.4.1 Example . . . 6
1.5 Annuity Payment (Present Value) . . . 7
1.5.1 Parameters . . . 7
1.5.2 Example . . . 7
1.6 Annuity Payment (Future Value) . . . 7
1.6.1 Example . . . 8
1.7 Remaining Balance Of Annuity . . . 8
1.7.1 Parameters . . . 9
1.7.2 Example . . . 9

2 Uncertainty Number **10**

2.1 Basics . . . 10
2.2 More variables . . . 10
2.3 Working with raw data . . . 11

List of Tables

1.1 Parameters of Present Value . . . 4
1.2 Parameters of Net Present Value . . . 5
1.3 Parameters of Future Value . . . 6
1.4 Parameters of Future Value of An Annuity . . . 6
1.5 Parameters of Annuity Payment (Present Value) . . . 7
1.6 Parameters of Annuity Payment (Future Value) . . . 8
1.7 Parameters of Remaining Balance Of Annuity . . . 9

Part

Mathos Core Library

1 Finance

This part of library represents Time Value of Money functions.

Time Value of Money (TVM) is an important concept in financial management. It can be used to compare investment alternatives and to solve problems involving loans, mortgages, leases, savings, and annuities.

TVM is based on the concept that a dollar that you have today is worth more than the promise or expectation that you will receive a dollar in the future. Money that you hold today is worth more because you can invest it and earn interest. After all, you should receive some compensation for foregoing spending. For instance, you can invest your dollar for one year at a 6% annual interest rate and accumulate $1.06 at the end of the year. You can say that the future value of the dollar is $1.06 given a 6% interest rate and a one-year period. It follows that the present value of the $1.06 you expect to receive in one year is only $1.

A key concept of TVM is that a single sum of money or a series of equal, evenly-spaced payments or receipts promised in the future can be converted to an equivalent value today. Conversely, you can determine the value to which a single sum or a series of future payments will grow to at some future date.

1.1 Present Value

Present value, also known as **present discounted value**, is a future amount of money that has been discounted to reflect its current value, as if it existed today. The present value is always less than or equal to the future value because money has interest-earning potential, a characteristic referred to as the time value of money.

```
decimal PresentValue(decimal futureValue, decimal rateOfReturn, int
    numberOfPeriods, bool round = true)
```

Calculates the present value of the *futureValue* for a specified *rateOfReturn* over *numberOfPeriods* periods.

1.1.1 Parameters

Parameter	Description
`decimal futureValue`	Future value (ex 100)
`decimal rateOfReturn`	Rate of return (ex 6 for 6%)
`int numberOfPeriods`	Number of periods
`bool round`	Determines whether the result is rounded to 2 decimal places

Table 1.1: Parameters of Present Value

1.1.2 Example

An individual wishes to determine how much money she would need to put into her money market account to have \$100 one year today if she is earning 5% interest on her account, simple interest.

```
PresentValue(100, 5, 1)
```

When it solve for PV, she would need \$95.24 today in order to reach \$100 one year from now at a rate of 5% simple interest.

1.2 Net Present Value

Net present value (NPV) or **Net present worth (NPW)** of a time series of cash flows, both incoming and outgoing, is defined as the sum of the present values (PVs) of the individual cash flows of the same entity.

```
decimal NetPresentValue(decimal initialInvestment, IList<decimal> cashFlow,
    decimal rateOfReturn, bool round)
```

Calculates the net present value for an investment with multiple case flows over equi-distant time intervals at a given rate of return

1.2.1 Parameters

Parameter	**Description**
`decimal initialInvestment`	Initial investment (ex 10000)
`IList<decimal> cashFlow`	List of expected cash flows from investment (ex 200, 100, 300))
`decimal rateOfReturn`	Expected rate of return (ex 5 for 5%)
`bool round`	Determines whether the result is rounded to 2 decimal places

Table 1.2: Parameters of Net Present Value

1.2.2 Example

An initial investment on plant and machinery of \$8320 thousand is expected to generate cash inflows of \$3411 thousand, \$4070 thousand, \$5824 thousand and \$2065 thousand at the end of first, second, third and fourth year respectively. At the end of the fourth year, the machinery will be sold for \$900 thousand. Calculate the present value of the investment if the discount rate is 18%.

```
decimal[] flow = new decimal[4] { 3411, 4070, 5824, 2065 + 900};
NetPresentValue(8320, flow, 18)
```

Net Present value is $\$2567,67$ thousand.

1.3 Future Value

Future value is the value of an asset at a specific date. It measures the nominal future sum of money that a given sum of money is "worth" at a specified time in the future assuming a certain interest rate. (For an asset with interest compounded annually.)

```
decimal FutureValue(decimal presentValue, decimal rateOfReturn, int
    numberOfPeriods, bool round = true)
```

Calculates the future value of the *presentValue* for a specified *rateOfReturn* over *numberOfPeriods* periods.

1.3.1 Parameters

Parameter	Description
`decimal presentValue`	Present value (ex 100)
`decimal rateOfReturn`	Rate of return (ex 6 for 6%)
`int numberOfPeriods`	Number of periods
`bool round`	Determines whether the result is rounded to 2 decimal places

Table 1.3: Parameters of Future Value

1.3.2 Example

```
FutureValue(1000, 10, 5)
```

$1000 invested for 5 years at 10%, compounded annually has a future value of $1,610.51.

1.4 Future Value Of An Annuity

Future Value Of An Annuity - the value of a group of payments at a specified date in the future. These payments are known as an annuity, or set of cash flows. The future value of an annuity measures how much you would have in the future given a specified rate of return or discount rate. The future cash flows of the annuity grow at the discount rate, and the higher the discount rate, the higher the future value of the annuity.

```
decimal FutureValueOfAnnuity(decimal periodicPayment, decimal ratePerPeriod,
    int numberOfPeriods, bool round)
```

Calculates the future value of an annuity.

Parameter	Description
`decimal periodicPayment`	Periodic payment amount of annuity
`decimal ratePerPeriod`	Rate per period
`int numberOfPeriods`	Number of periods
`bool round`	Determines whether the result is rounded to 2 decimal places

Table 1.4: Parameters of Future Value of An Annuity

1.4.1 Example

The basis is that providing a lump sum of $5,000 today costs more than providing a cash flow of $1,000 per year for five years. This is because if you provide the lump sum today, you could have invested it and received an additional return.

```
FutureValueOfAnnuity(1000,6,5)
```

Using this example, and assuming a discount rate of 6%, the future value of an annuity that pays $1,000 per year for five years is $5637. This means that if you could get a return on your invested funds of 6% per year, providing an annuity of $1,000 per year would be worth $637 ($5,637 − $5,000) more to the issuer than giving a lump sum.

1.5 Annuity Payment (Present Value)

Annuity Payment (Present Value). The annuity payment formula is used to calculate the periodic payment on an annuity. An annuity is a series of periodic payments that are received at a future date. The present value portion of the formula is the initial payout, with an example being the original payout on an amortized loan.

```
decimal AnnuityPaymentPresentValue(decimal presentValue, decimal ratePerPeriod,
    int numberOfPeriods, bool round)
```

Returns the annuity payment calculated for a present value given the rate and number of periods.

1.5.1 Parameters

Parameter	Description
`decimal presentValue`	Present value of annuity
`decimal rateOfReturn`	Rate of return (ex 6 for 6%)
`int numberOfPeriods`	Number of periods
`bool round`	Determines whether the result is rounded to 2 decimal places

Table 1.5: Parameters of Annuity Payment (Present Value)

1.5.2 Example

You can get a $150,000 home mortgage at 7% annual interest rate for 30 years. Payments are due at the end of each month and interest is compounded monthly. How much will your payments be?

$$\begin{aligned} PVoa &= 150,000 && \text{the loan amount} \\ i &= 0.5833333 && \text{interest per month (7 / 12)} \\ n &= 360 \quad \text{periods} && \text{(12 payments per year for 30 years)} \end{aligned}$$

```
AnnuityPaymentPresentValue(150000,0.5833333m,360)
```

Monthly payments are $997.95

1.6 Annuity Payment (Future Value)

Annuity Payment (Future Value). The annuity payment formula is used to calculate the cash flows of an annuity when future value is known. An annuity is denoted as a series of periodic payments.

The annuity payment formula shown here is specifically used when the future value is known, as opposed to the annuity payment formula used when present value is known. There are not only mathematical differences between calculating an annuity when present value is known and when future value is known, but also differences in the real life application of the formulas.

```
decimal AnnuityPaymentFutureValue(decimal futureValue, decimal ratePerPeriod,
    int numberOfPeriods, bool round)
```

Returns the annuity payment calculated for a future value given the rate and number of periods.

Parameter	Description
`decimal futureValue`	Future value of annuity
`decimal ratePerPeriod`	Rate of period (ex 6 for 6%)
`int numberOfPeriods`	Number of periods
`bool round`	Determines whether the result is rounded to 2 decimal places

Table 1.6: Parameters of Annuity Payment (Future Value)

1.6.1 Example

In 10 years, you will need $50,000 to pay for college tuition. Your savings account pays 5% interest compounded monthly. How much should you save each month to reach your goal?

$$\begin{aligned} FVoa &= 50,000, && \text{the future savings goal} \\ i &= 0.4167 && \text{interest per month (5 / 12)} \\ n = 120 &\quad \text{periods} && \text{(12 payments per year for 10 years)} \end{aligned}$$

```
AnnuityPaymentFutureValue(50000,0.4167m,120)
```

Monthly payments are $321.99

1.7 Remaining Balance Of Annuity

Remaining Balance Of Annuity. The formula for the remaining balance on a loan can be used to calculate the remaining balance at a given time(time n), whether at a future date or at present. The remaining balance on a loan formula is only used for a loan that is amortized, meaning that the portion of interest and principal applied to each payment is predetermined.

```
decimal RemainingBalanceOfAnnuity(decimal originalValue, decimal payment,
    decimal ratePerPeriod, int numberOfPeriods, bool round)
```

Returns the remaining balance of an annuity given the original value, payment amount, rate per period and number of periods paid.

1.7.1 Parameters

Parameter	Description
`decimal originalValue`	Original value of annuity
`decimal payment`	Payment amount
`decimal ratePerPeriod`	Rate of period (ex 6 for 6%)
`int numberOfPeriods`	Number of periods
`bool round`	Determines whether the result is rounded to 2 decimal places

Table 1.7: Parameters of Remaining Balance Of Annuity

1.7.2 Example

Your home mortgage is \$100000 at 7% annual interest rate. Your monthly payments are \$1000. How much are you owe a bank after 5 years payments?

payment = \$12000 (annual payment $\$1000 * 12$ months)

```
RemainingBalanceOfAnnuity(100000,12000,7,5)
```

Remaining Balance is $\$71246, 30$

2 Uncertainty Number

Mathos Core Library Uncertain Number class has undergone some changes, one of them is being able to work with custom functions. In this article, we are to focus on this new feature mainly.

The ability to use custom functions is split up into four methods with the name *Custom-Function* but with different input parameters and different modifiers. A list of this function with different method signatures is shown below:

```
public UncertainNumber CustomFunction(Func<decimal, decimal> function)

public static UncertainNumber CustomFunction(Func<decimal[],decimal> function,
    params UncertainNumber[] points )

public static UncertainNumber[] CustomFunction(Func<decimal,decimal> function,
    string TSV)

public static UncertainNumber[] CustomFunction(Func<decimal[],decimal> function
    , string TSV)
```

2.1 Basics

As you can see, all of them take a function that can be customized in any way using for instance a lambda expression. Let's look at an example of the first function in the list. Say we have declared an Uncertain Number with a value and an uncertainty and want to perform a mathematical calculation with the value, and at the same time keep its uncertainty. As an example, the function $y = 5a^2$ is going to be used.

```
// remember to include Mathos.Statistics.Uncertainty;

UncertainNumber x = new UncertainNumber(32.7M, 2);
x = x.CustomFunction(a => 5 * a * a);

x = x.AutoFormat();

Console.WriteLine(x);
```

This will return 5300 ± 700. Notice that *AutoFormat* makes the value and the uncertainty look in the way they should when you display this data in a rapport.

2.2 More variables

The other three options in the list are static, therefore, in contrast to the first method (described in Basics), you can access them without creating a new instance of the UncertainNumber class.

Let's look at the second method. That will allow us to have more than one variable inside the function. For this example, we are to use three variables.

```
UncertainNumber result = UncertainNumber.CustomFunction(a => a[0] * a[1],
new UncertainNumber(2, 0.1M), new UncertainNumber(3, 0.1M));
result = result.AutoFormat();

Console.WriteLine(result);
```

This will write 6.0 ± 0.5 on the screen. The function that was used in the calculation is $f(x, y) = xy$. Notice that `a[0]` is the first variable and corresponds to the first element of the UncertainNumber array, which in this case is 2.0 ± 0.1.

2.3 Working with raw data

The last two functions allow you to include a TSV (tab separated values) string into the method. This means that you can copy a table from for example Microsoft Excel and get the uncertainties calculated for you. The last function allows you to use more than one variable, while the second last is restricted to one variable only. Both these functions are static, and so have to be included directly through UncertainNumber class (you don't need a new instance of that class). Note that these two functions return an array of UncertainNumber.