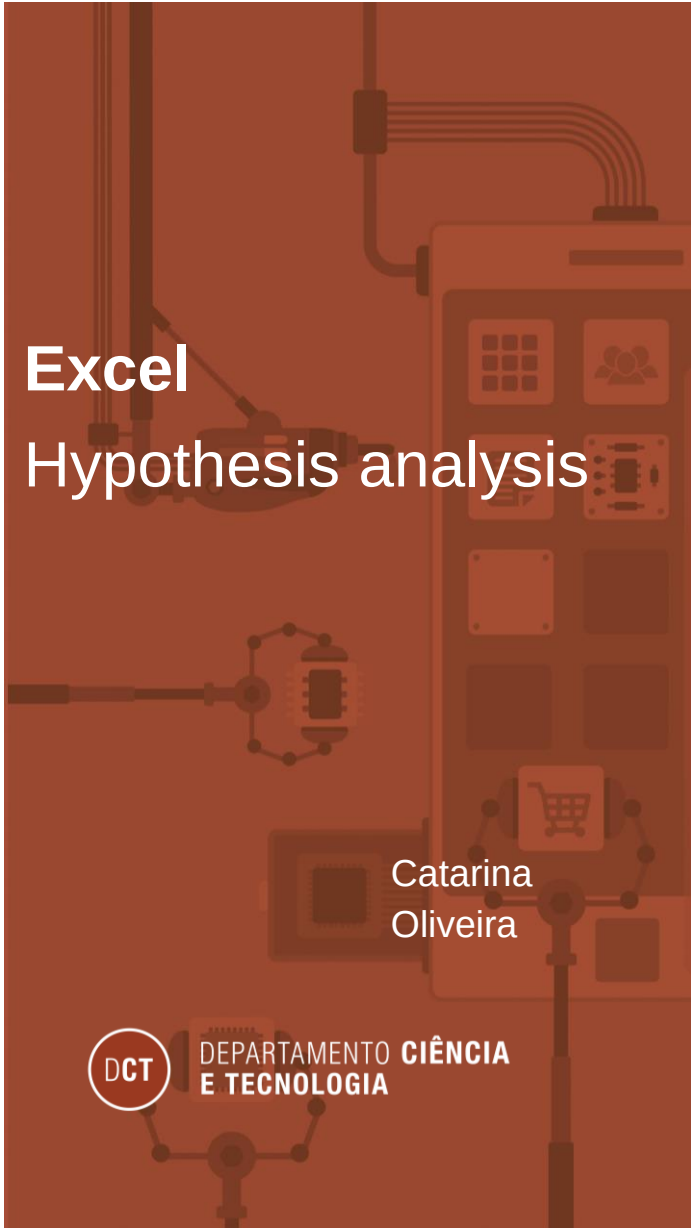



UNIVERSIDADE PORTUCALENSE

# CONTENT

# Hypothesis Analysis

Allow solving problems: Simulation | Forecasting | Linear Programming | …

Based on two main concepts:

- **Input cells**: set of cells where the problem data is inserted/changed. The values must affect the result of at least one result cell through the use of references or formulas
    - Directly: the cell's formula uses a reference to the input cell
    - Indirectly: the cell's formula has a reference to another cell, and this one uses a reference to the input cell
- **Result cells**: set of cells that, using formulas, execute calculations using direct or indirect references to the iput cells

Types:

- **Data table**: allows determining and analysing the values of one or more result cell according to one or two input cells
- **Scenario Manager**: allows determining and analysing the values of several result cells according to several input cells
- **Seek Goal**: allows determining the value of an input cell according to the predicted value of a result cell
- **Solver**: allows determining the value of several input cells according to the predicted value of a result cell, considering a set of constraints the input cells will have to respect

# Data tables

Data tables: allow the simulation and analysis of the impact that a set of values in one or two input cells will have on the results of one or more formulas or result cells

Instead, we can manually replace the cell value and analyse the result

Limited technique: we can only see the impact of one value per change

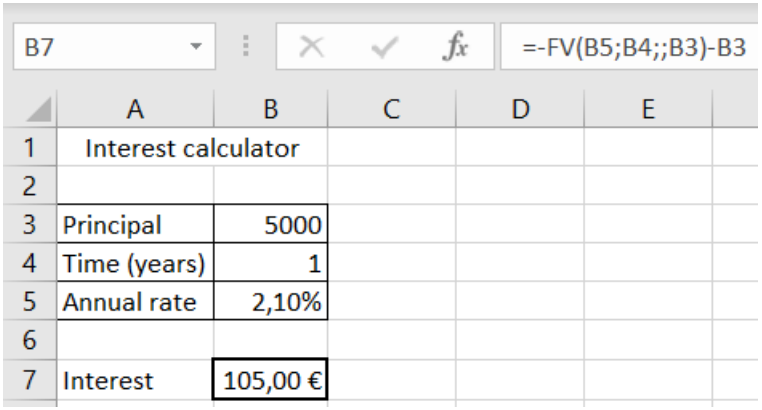
Example: calculate the interest of an investment

Test and determine the impact of different rates, times, principal

Considering variable rates, times and principal, the analysis can be made in two different ways:

- **One variable** simulation tables (individually): evaluate only the impact of:
  - Rate variation
  - Time variation
  - Principal variation
- **Two variable** simulation tables: evaluate the combined impact of:
  - Rate and time variation
  - Time and principal variation
  - Rate and principal variation

# One variable data table




B3, B4 and B5: input cells

B7: result cell

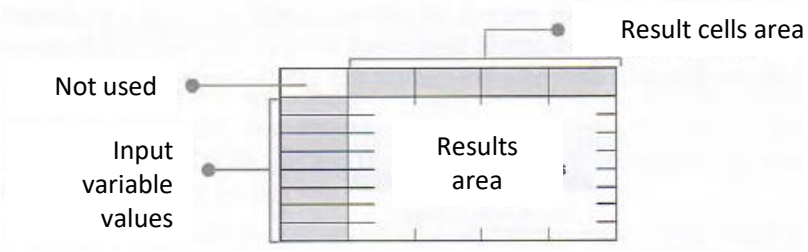
Assuming fixed principal and time, analyse the impact of different interest rates (1,90%, 2,00%, 2,10%, 2,25%, 2,50%) on the final rate value

**Manually**: insert the different values into B5 and memorise the results

**Data tables**: allows us to automatically calculate the several results in a simulation table

# One variable data table

To build a one variable data table, first we need to build a table with following structure:




**First column**, from the second cell: insert the values to test (1,90%, 2,00%, 2,10%, 2,25%, 2,50%)

**First row**, from the second cell: insert reference to the result cells (in this case, it's only B7 → =B7)

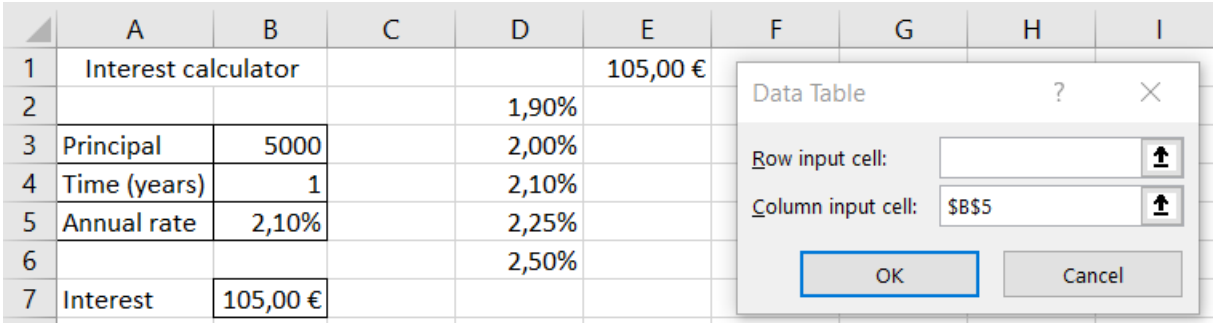
E1 =B7

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Interest calculator | | | | 105,00 € |
| 2 | | | | 1,90% | |
| 3 | Principal | 5000 | | 2,00% | |
| 4 | Time (years) | 1 | | 2,10% | |
| 5 | Annual rate | 2,10% | | 2,25% | |
| 6 | | | | 2,50% | |
| 7 | Interest | 105,00 € | | | |

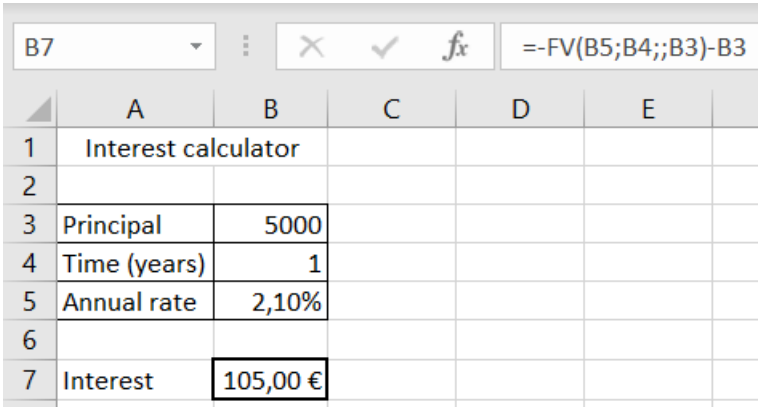
# One variable data table

**How to:**

1. Select the table (in this case D1:E6)
2. Tab “Data” > Group “Forecast” > Command “What-if Analysis” > Option “Data table”
3. Fill the dialog “Data Table”
    1. Insert the reference of the input cell in one of the fields:
        - **Row input cell**: if the values to test are in the simulation table’s first row
        - **Column input cell**: if the values to test are in the simulation table’s first column (in this case, we insert here $B$5 – absolute reference to the cell that contains the rate)



# Two variables data table




B3, B4 and B5: input cells

B7: result cell

Assuming the fixed principal only, analyse:

- Different interest rates (1,90%, 2,00%, 2,10%, 2,25%, 2,50%)
- Different times (1, 2, 3, 4, 5 years)

And observe the impact on the final interest value

# Two variables data table

To build a two variables data table, first we need to build a table with the following structure:


**First cell** (first row, first column): formula or reference to the result cell (in this case, B7 → =B7)

**First column**, from the second cell: values to test for one of the input cells (1,90%, 2,00%, 2,10%, 2,25%, 2,50%)

**First row**, from the second cell: values to test to the other input cell (1, 2, 3, 4, 5)

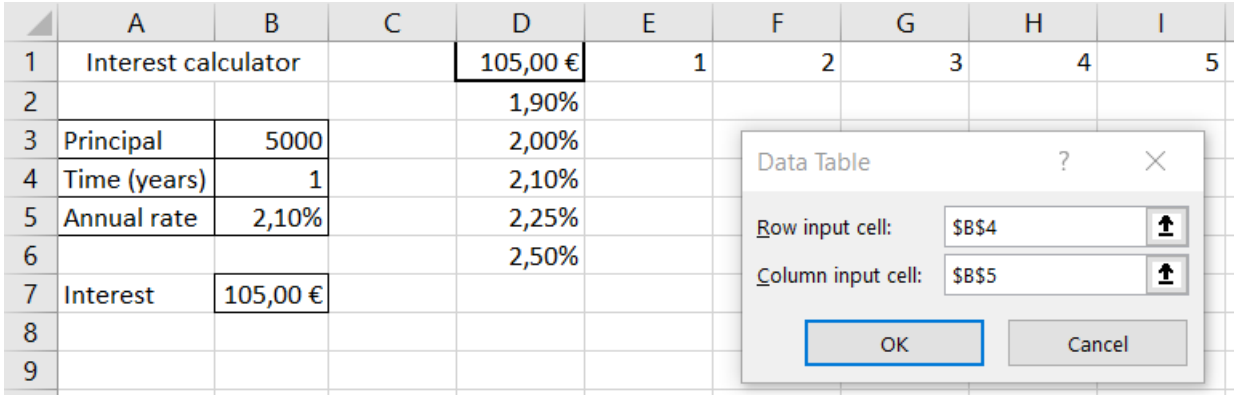
D1 | =B7

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Interest calculator | | | 105,00 € | 1 | 2 | 3 | 4 | 5 |
| 2 | | | | 1,90% | | | | | |
| 3 | Principal | 5000 | | 2,00% | | | | | |
| 4 | Time (years) | 1 | | 2,10% | | | | | |
| 5 | Annual rate | 2,10% | | 2,25% | | | | | |
| 6 | | | | 2,50% | | | | | |
| 7 | Interest | 105,00 € | | | | | | | |

# Two variables data table

**How to:**

1. Select the table (in this case D1:I6)
2. Tab “Data” > Group “Forecast” > Command “What-if Analysis” > Option “Data table”
3. Fill the dialog “Data Table”
    1. Insert the reference of the input cell in one of the fields:
        - **Row input cell**: if the values to test are in the simulation table’s first row (in this case, $B$4 – the cell that contains the time)
        - **Column input cell**: if the values to test are in the simulation table’s first column (in this case, $B$5 – the cell that contains the interest rate)



# Scenario Manager

Allows:

- Define, create and save different sets of values for certain input cells that can be applied to the spreadsheet at any time, to observe the impact of the values on the result cells.
- Automatically generate scenario reports that easily and accurately present the scenarios characteristics and results

Most common application: creation of prediction models (e.g. sales of a company based on several factors)

- Models include scenarios with names like “best scenario”, “worst scenario”, “regular scenario”
- For a set of variables, the predictable values are combined

# Scenario Manager

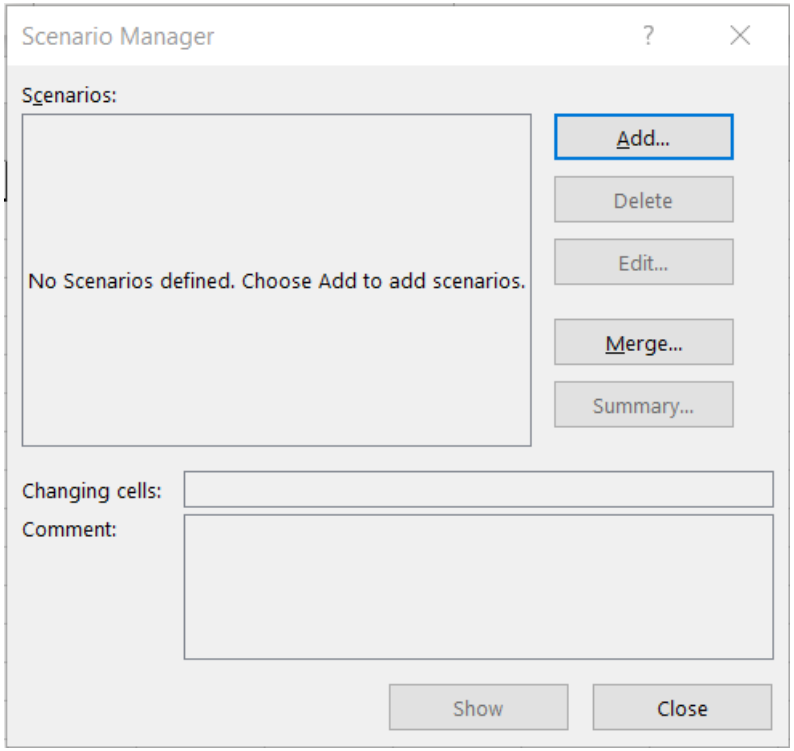
**Preparation** (before building a scenario):

- Identify the input cells
- Define the values the values those cell will have on the scenario
- We don’t need to think about the result cells yet, because the building of the scenario is just a mechanism to automatically replace on the spreadsheet the values contained in a set of input cells.
    - As the input cells are associated with result cells, by replacing the values in the input cells, Excel will automatically calculate and update the result cells

# Scenario Manager

**How to**

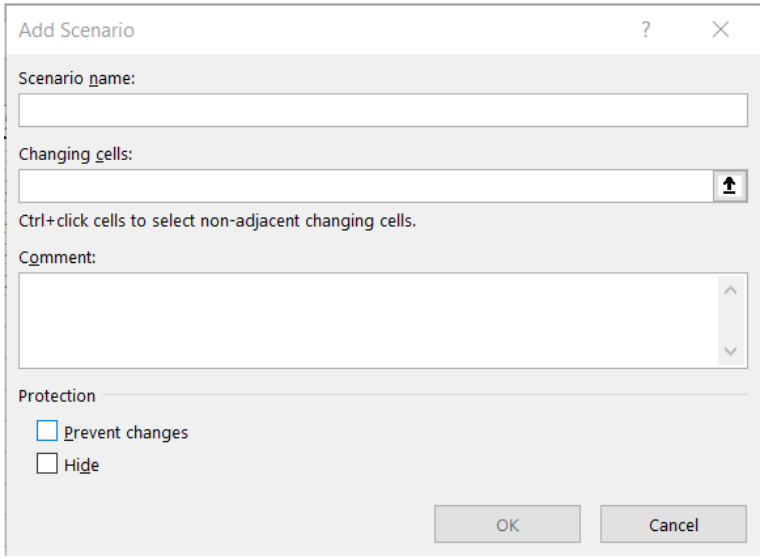
1. Tab “Data” > Group “Forecast” > Command “What-if Analysis” > Option “Scenario Manager”
2. On the dialog “Scenario Manager” Click “Add”



# Scenario Manager

**How to (cont.)**

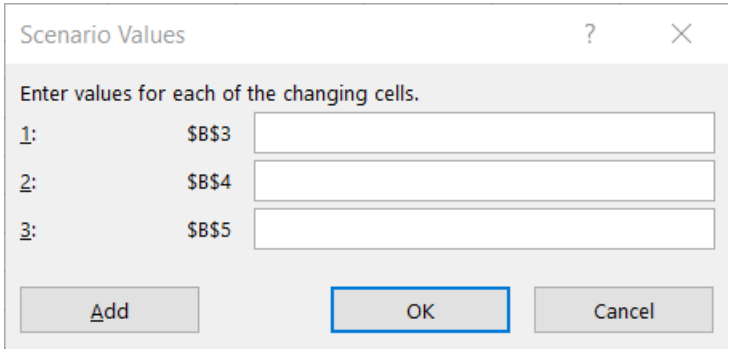
3. On the Add Scenario dialog:
    - **Scenario name**: insert the name to easily identify the scenario
    - **Changing cells**: input cells for which the values of the scenario will be set
    - (optional) **Comment**: additional information
    - (optional) **Protection**: define if the scenario needs to be protected or hidden



# Scenario Manager

**How to (cont.)**

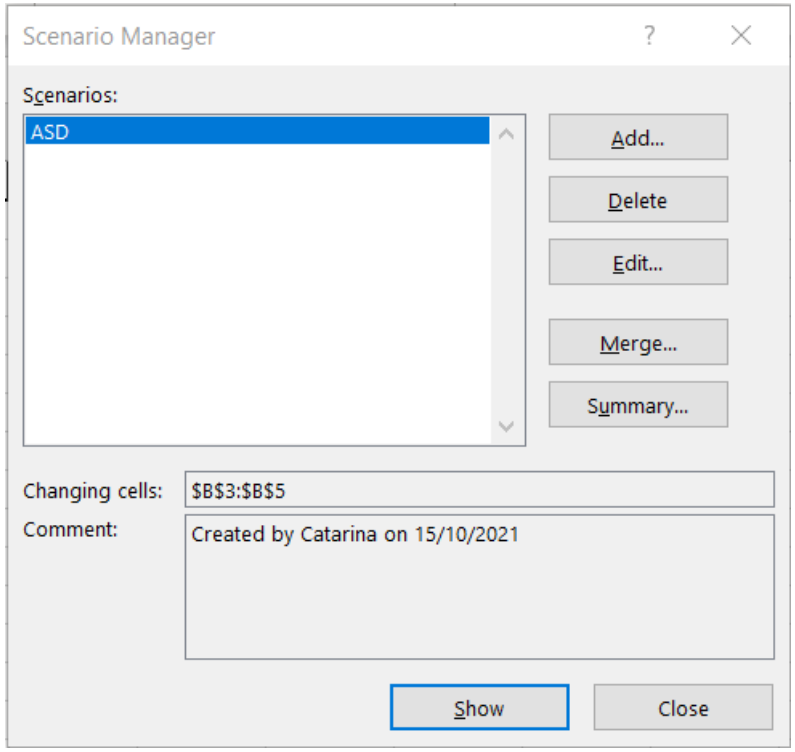
4. On the dialog “Scenario Values”:
    4. For each input cells, insert the value considered for the scenario



# Scenario Manager

**How to (cont.)**

5. Back to the Scenario Manager dialog



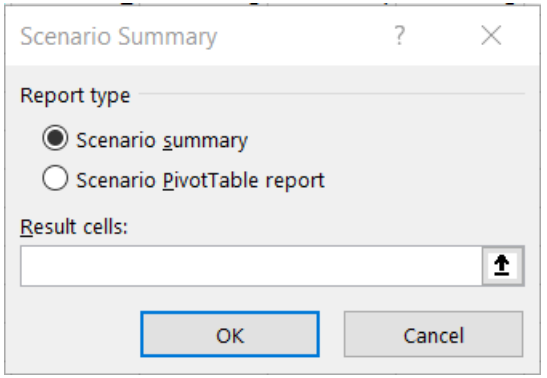

Options:

- **Add** new scenario
- **Delete** an existing scenario
- **Edit** an existing scenario
- **Merge**: join scenarios from other sheets of the same book
- **Summary**: generate a summary report or a pivot table, considering a set of result cells
- **Show**: view the result of applying a scenario directly on the spreadsheet (same as double click on the scenario's name)
- **Close** the scenarios manager

# Scenario Summary

Two types:

- Scenario summary report: automatic generation of a table that presents, for each scenario, the values of the input and result cells
- Scenario PivotTable report: cross the input and result cells' values




Generated in a new Worksheet

**How to:**

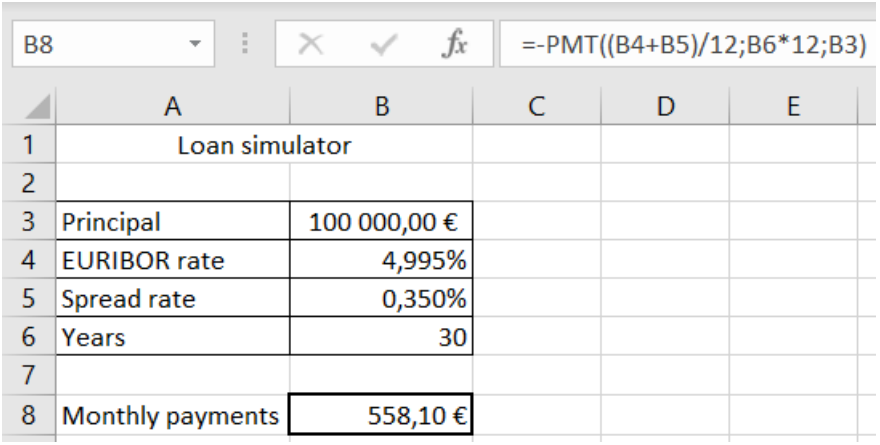
1. On the scenario manager dialog, click "Summary"
2. On the Scenario Summary dialog:
    - Report type: choose the desired type
    - Result cells: by default, Excel presents a list with all the cells depending on the input cells of the scenario. Some can be deleted.

# Goal Seek

Problems where we need to determine the value of an input cell, knowing the result cell's value

Contrary to the data tables and scenarios, we want to determine the input cells' values, and not the result cells' value

Example: loan simulator that calculates the monthly payments of a loan (B8) according to the principal (B3), the interest rates to apply (B4 e B5) and the number of years to pay the loan (B6)



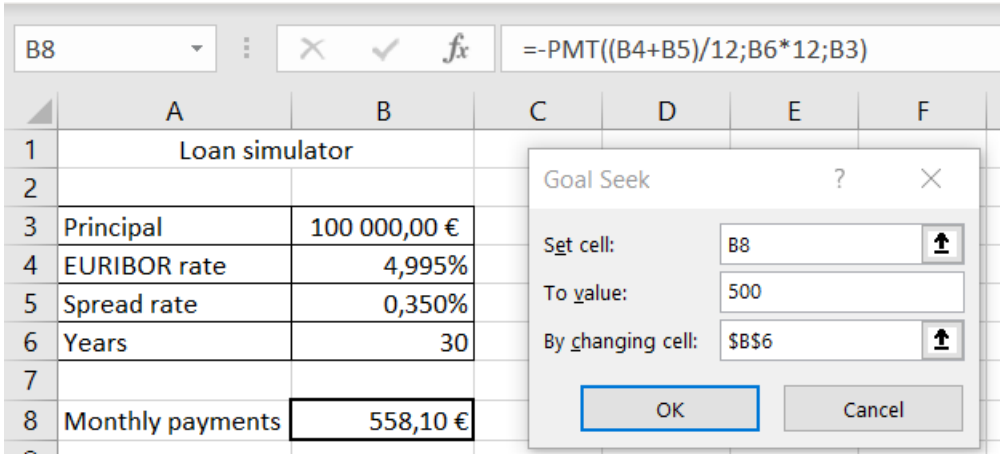

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Loan simulator | | | | |
| 2 | | | | | |
| 3 | Principal | 100 000,00 € | | | |
| 4 | EURIBOR rate | 4,995% | | | |
| 5 | Spread rate | 0,350% | | | |
| 6 | Years | 30 | | | |
| 7 | | | | | |
| 8 | Monthly payments | 558,10 € | | | |

# Goal Seek

If we want to know the number of years needed so that the monthly payments have the value of 500€

**How to**:

1. Tab “Data” > Group “Forecast” > Command “What-if Analysis” > Option “Goal Seek”
2. On the dialog “Goal Seek”:
    - **Set cell**: reference to the result cell with known value (B8)
    - **To value**: goal value on the result cell (500)
    - **By changing cell**: reference to the cell whose value is to be determined (B6)



# Solver

It allows solving various types of problems in a spreadsheet (eg linear algebra or optimization) involving the maximization, minimization or reaching of a value of a given formula (called Objective Function), which may also be subject to a set of constraints that allow to reduce the number of possible solutions

While Goal Seek allows you to determine only the value of one variable, Solver allows you to determine up to 200 variables, considering a result cell (which represents the Objective Function) and up to 100 restrictions to the problem.

It uses an iterative process, based on mathematical models, which, through several solution attempts, analyzes the results and seeks not only the single solution, but the optimal solution to a problem.

For very complex problems, you can generate multiple solutions that can be saved using different scenarios

# Solver installation

**How to**:

1. Tab “File” > Command “Options”
2. On the dialog Excel Options
    1. Select “Add-ins”
    2. In “Manage” select “Excel Add-ins” and click “Go”
3. On the Add-ins dialog
    1. Select Solver

**Note**:

Sometime, when Solver is active, Excel becomes slower to load. In that case, the add-in should be deactivated for normal use, and activated only when solver is to be used

# Using Solver

**Preparation:**

- Identify the cells that represent the problem variables (they must be empty, as the solver results will be placed there)
- Define and enter the Objective Function formula in a cell, using the cell references that correspond to the variables in that formula
- Place each of the problem's constraints, using one or more cells if necessary. Considering that each of the constraints is defined by an expression with the syntax EXP1 OPERATOR EXP2, the three parts (EXP1, OPERATOR and EXP2) can be placed in three different cells, although it is not always necessary

# Using Solver

Example: Consider the following problem formulation

**Objective**: Minimise the cost

**Variables**: A – Quantity of product A

B – Quantity of product B

**Objective function**: Minimise the expression 10*A+5*B

**Constraints**:

1) 20*A+50*B>=200
2) 50*A+10*B>=150
3) 30*A+30*B>=210
4) A>=0
5) B>=0

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | Variables | | | |
| 3 | | A | | | |
| 4 | | B | | | |
| 5 | | | | | |
| 6 | | Objective function | =10*C3+5*C4 | | |
| 7 | | | | | |
| 8 | | Constraints | | | |
| 9 | | 1) | =20*C3+50*C4 | >= | 200 |
| 10 | | 2) | =50*C3+10*C4 | >= | 150 |
| 11 | | 3) | =30*C3+30*C4 | >= | 210 |
| 12 | | 4) | =C3 | >= | 0 |
| 13 | | 5) | =C4 | >= | 0 |

# Using Solver

**How to**:

1. Tab “Data” > Group “Analysis” > Command “Solver”
2. On the dialog Solver Parameters:
    - **Set objective**: identify the cell that contains the dormula to the problem’s objective function (C6)
    - **To**: define the type of calculation desired for the objective function (Min)
    - **By changing variable cells**: identify the cells used as variables on the objective function (C3:C4)
    - **Subject to the Constraints**: define the problem’s constraints (next slide)
        - **Add** constraint
        - **Change** constraint
        - **Delete** constraint
        - **Reset All**: reset all the parameters inserted
        - **Load/Save**: load or save the solver parameters on a spreadsheet
    - **Make unconstrained variables non-negative**: define that the calculated values for every variable are greater than 0. Similar to creating a >=0 constraint for every variable cell
    - **Select a solving method**: select the optimisation method:
        - **GRG Nonlinear**: uniform non linear problems
        - **Simplex LP**: linear problems
        - **Evolutionary**: non uniform problems
    - **Options**: configure other options associated with the chosen method

# Defining Constraints

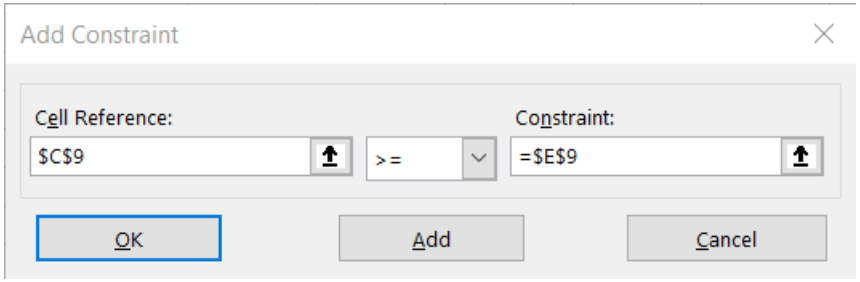
**How to:**

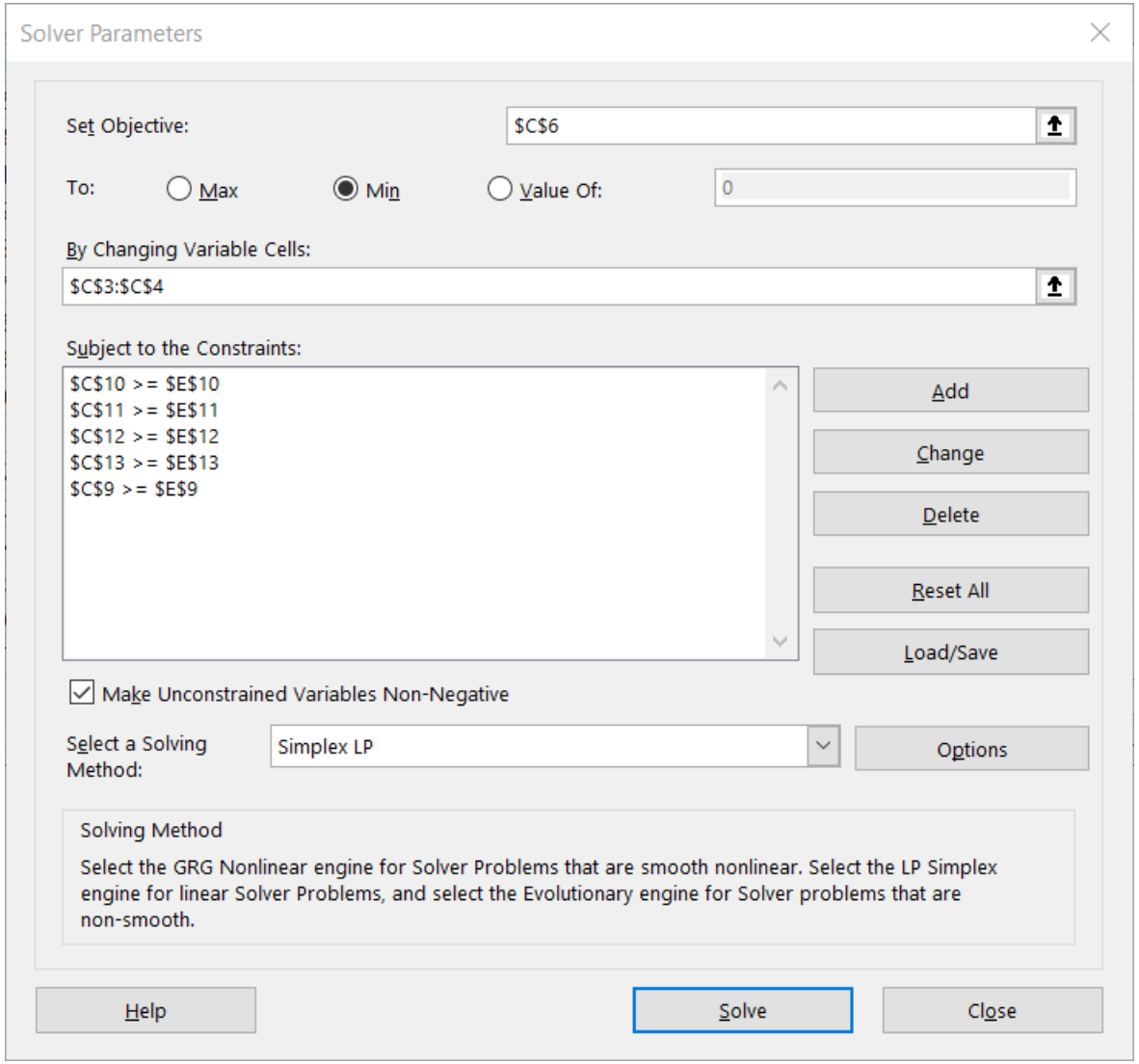
1. On the dialog Solver Parameters, on the constraints section, click Add
2. On the dialog Add Constraint
    - **Cell Reference**: identify the cell with the left part of the constraint

      Example, for the first constraint: C9
    - **Operator list**: allows choosing the operator to use In the constraint

      Example, for the first constraint: >=
    - **Constraint**: identify the cell with the right part of the constraint

      Example, for the first constraint: E9
    - **OK**: finish creating constraints
    - **Add**: save the current constraint and allows to add another



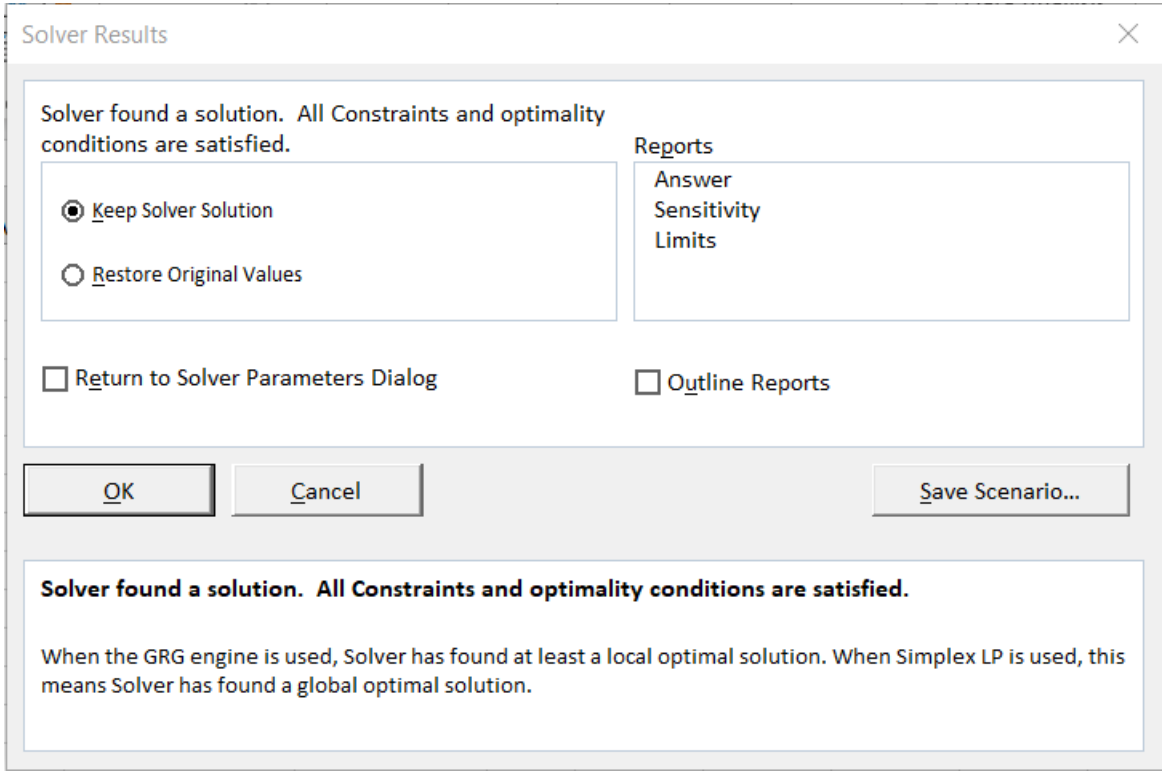

Solver Parameters
Set Objective:
$C$6
To:
Max
Min
Value Of:
0
By Changing Variable Cells:
$C$3:$C$4
Subject to the Constraints:
$C$10 >= $E$10
$C$11 >= $E$11
$C$12 >= $E$12
$C$13 >= $E$13
$C$9 >= $E$9
Add
Change
Delete
Reset All
Load/Save
Make Unconstrained Variables Non-Negative
Select a Solving Method:
Simplex LP
Options
Solving Method
Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.
Help
Solve
Close

# Solver results

On the dialog Solver Results:

- **Keep Solver Solution**: inserts the results into the variables' cells
- **Restore original values**: reset the original values on the variables' cells
- **Reports**: allows the creation of three types of report:
  - ***Answer***: displays information about the target cell (Objective Function) and variable cells with original and final values, as well as restrictions and information about them, namely, their status
  - ***Sensitivity***: displays information about the solution's sensitivity to small changes to the Objective Function formula or constraints. This report is not generated for models with integer restrictions. For nonlinear models, the report provides values for reduced gradients and Lacrange multipliers. For linear models, the report includes reduced costs, target coefficient with allowed to increase and decrease
  - ***Limits***: displays the Objective Function cell and the variable cells with their values, lower and upper limits and target values. This report is not generated for models with integer constraints. The lower bound is the smallest value the adjustable cell can take, keeping all other adjustable cells fixed and still satisfying constraints. The upper limit is the highest value.
- **Return to solver parameters dialog**
- **Outline reports**
- **Save scenario**: create a scenario from the results obtained

# Configure Result



# Result

Example: Consider the following problem formulation

**Objective**: Minimise the cost

**Variables**: A – Quantity of product A

B – Quantity of product B

**Objective function**: Minimise the expression 10*A+5*B

**Constraints**:

1) 20*A+50*B>=200
2) 50*A+10*B>=150
3) 30*A+30*B>=210
4) A>=0
5) B>=0

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | Variables | | | |
| 3 | | A | 2 | | |
| 4 | | B | 5 | | |
| 5 | | | | | |
| 6 | | Objective function | 45 | | |
| 7 | | | | | |
| 8 | | Constraints | | | |
| 9 | | 1) | 290 | >= | 200 |
| 10 | | 2) | 150 | >= | 150 |
| 11 | | 3) | 210 | >= | 210 |
| 12 | | 4) | 2 | >= | 0 |
| 13 | | 5) | 5 | >= | 0 |

**Result:**

We should produce 2 units of product A and 5 of product B with a total cost of 45

UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.