'////////////////////////////////////////////////////////////////////////////////////////////////////

'////////////////////////////////////////////////////////////////////////////////////////////////////

Option Explicit 'Requires that all variables to be declared explicitly.

Option Base 1 'The "Option Base" statement allows to specify 0 or 1 as the

'default first index of arrays.

'////////////////////////////////////////////////////////////////////////////////////////////////////

'////////////////////////////////////////////////////////////////////////////////////////////////////

Private PUB\_NPV\_VAL As Double

Private PUB\_NPV\_VECTOR As Variant

Private PUB\_DATE\_VECTOR As Variant

Private PUB\_DATA\_VECTOR As Variant

Private Const PUB\_EPSILON As Double = 2 ^ 52

'////////////////////////////////////////////////////////////////////////////////////////////////////

'The main purpose of this section is to show how to work with Optimization problems in Excel.

'Of course this speaks about math, statistic and numeric calculus but this is not a math or a

'statistic class. Therefore, we will not cover theorems and demonstrations. You will find, on

'the contrary, many examples that explain, step by step, how to structure an optimization problem

'and reach the result that you need, straight and easy. Tips and tricks for general applications

'in VBA will be covered in class.

'////////////////////////////////////////////////////////////////////////////////////////////////////

'A Must Read:

'////////////////////////////////////////////////////////////////////////////////////////////////////

'http://www.gummy-stuff.org/Newton-Method.htm

'http://www.gummy-stuff.org/IRR2.htm

'http://www.gummy-stuff.org/XIRR-bug.htm

'http://www.gummy-stuff.org/multiple-returns.htm

'http://www.gummy-stuff.org/NPV.htm

'http://www.effas-ebc.org/fileadmin/presentations/Milan\_June\_2010/WMarty\_The\_IRR\_Equation\_and\_its\_multiple\_solutions.pdf

'Optional Readings:

'http://www.gummy-stuff.org/loans.htm

'http://www.gummy-stuff.org/newton\_explain.htm

'http://www.gummy-stuff.org/Rate-of-Return-2.htm

'http://www.gummy-stuff.org/Rate-of-Return-3.htm

'http://www.gummy-stuff.org/Rate-of-Return-4.htm

'////////////////////////////////////////////////////////////////////////////////////////////////////

Sub Run\_Newton()

Dim i As Integer: i = Range("dx").Row - 4

With ActiveSheet.Cells(i, 1)

Union(Range(.Cells(6, 5), .Cells(20, 5)), \_

Range(.Cells(6, 6), .Cells(20, 6))).ClearContents

Range(.Cells(6, 5), .Cells(20, 5)).Formula = "=" + .Cells(2, 3)

Range(.Cells(6, 6), .Cells(20, 6)).Formula = "=" + .Cells(3, 3)

Call Run\_Plot

End With

End Sub

Sub Run\_Plot()

Dim i As Integer: i = Range("dx").Row - 4

With ActiveSheet.Cells(i, 1)

Range(.Cells(8, 9), .Cells(58, 9)).ClearContents

Range(.Cells(8, 9), .Cells(58, 9)).Formula = "=" + .Cells(4, 8)

End With

End Sub

Function NEWTON\_CHARACTERISTICS\_SOLVER\_FUNC(ByVal YFORMULA\_STR As String, \_

ByVal YDFORMULA\_STR As String, \_

Optional ByVal XGUESS\_VAL As Double = 0.2, \_

Optional ByVal NBINS As Long = 20, \_

Optional ByVal OUTPUT As Integer = 1)

Dim i As Long

Dim X\_VAL As Double

Dim Y\_VAL As Double 'f(x)

Dim YD\_VAL As Double 'f'(x)

Dim TEMP\_MATRIX As Variant

On Error GoTo ERROR\_LABEL

ReDim TEMP\_MATRIX(0 To NBINS, 1 To 3)

TEMP\_MATRIX(0, 1) = "x": TEMP\_MATRIX(0, 2) = "f(x)": TEMP\_MATRIX(0, 3) = "f'(x)"

i = 1

TEMP\_MATRIX(i, 1) = XGUESS\_VAL

X\_VAL = TEMP\_MATRIX(i, 1)

GoSub CALC\_LINE

TEMP\_MATRIX(i, 2) = Y\_VAL

TEMP\_MATRIX(i, 3) = YD\_VAL

For i = 2 To NBINS 'Solve f(x) = 0

TEMP\_MATRIX(i, 1) = TEMP\_MATRIX(i - 1, 1) - TEMP\_MATRIX(i - 1, 2) / TEMP\_MATRIX(i - 1, 3) 'Newton

X\_VAL = TEMP\_MATRIX(i, 1)

GoSub CALC\_LINE

TEMP\_MATRIX(i, 2) = Y\_VAL

TEMP\_MATRIX(i, 3) = YD\_VAL

Next i

Select Case OUTPUT

Case 0

NEWTON\_CHARACTERISTICS\_SOLVER\_FUNC = TEMP\_MATRIX

Case Else 'IRR

NEWTON\_CHARACTERISTICS\_SOLVER\_FUNC = TEMP\_MATRIX(NBINS, 1)

End Select

Exit Function

'--------------------------------------------------------------------------------------------------------------------------

CALC\_LINE:

'--------------------------------------------------------------------------------------------------------------------------

Y\_VAL = Excel.Application.Evaluate(Replace(YFORMULA\_STR, "x", X\_VAL))

'YFORMULA\_STR = 10 \* (1 + X\_VAL) ^ 5 - 5 \* (1 + X\_VAL) ^ 4 - 5 \* (1 + X\_VAL) ^ 3 - 5 \* (1 + X\_VAL) ^ 2 - 5 \* (1 + X\_VAL) - 5

YD\_VAL = Excel.Application.Evaluate(Replace(YDFORMULA\_STR, "x", X\_VAL))

'YDFORMULA\_STR = 50 \* (1 + X\_VAL) ^ 4 - 20 \* (1 + X\_VAL) ^ 3 - 15 \* (1 + X\_VAL) ^ 2 - 10 \* (1 + X\_VAL) - 5

'--------------------------------------------------------------------------------------------------------------------------

Return

'--------------------------------------------------------------------------------------------------------------------------

ERROR\_LABEL:

NEWTON\_CHARACTERISTICS\_SOLVER\_FUNC = Err.number

End Function

Function IRR\_FUNC(ByRef DATA\_RNG As Variant, \_

Optional ByRef DATE\_RNG As Variant, \_

Optional ByVal GUESS\_VAL As Double = 0.1, \_

Optional ByVal nLOOPS As Long = 1000, \_

Optional ByVal tolerance As Double = 10 ^ -10) '0.00000000001

Dim COUNTER As Long

Dim CONVERG\_VAL As Integer

Dim FORMULA\_STR As String

On Error GoTo ERROR\_LABEL

PUB\_DATA\_VECTOR = DATA\_RNG

If UBound(PUB\_DATA\_VECTOR, 1) = 1 Then

PUB\_DATA\_VECTOR = MATRIX\_TRANSPOSE\_FUNC(PUB\_DATA\_VECTOR)

End If

If IsArray(DATE\_RNG) = True Then

PUB\_DATE\_VECTOR = DATE\_RNG

If UBound(PUB\_DATE\_VECTOR, 1) = 1 Then

PUB\_DATE\_VECTOR = MATRIX\_TRANSPOSE\_FUNC(PUB\_DATE\_VECTOR)

End If

If UBound(PUB\_DATE\_VECTOR, 1) <> UBound(PUB\_DATA\_VECTOR, 1) Then: GoTo ERROR\_LABEL

FORMULA\_STR = "CALL\_XIRR\_OBJ\_FUNC"

Else

FORMULA\_STR = "CALL\_IRR\_OBJ\_FUNC"

End If

If GUESS\_VAL <= 0 Then: GUESS\_VAL = 10 ^ -1

'If GUESS\_VAL >= 1 Then: GUESS\_VAL = 1 - tolerance

'IRR\_FUNC = NEWTON\_ZERO\_FUNC(GUESS\_VAL, "CALL\_XIRR\_OBJ\_FUNC", "", CONVERG\_VAL, COUNTER, nLOOPS, tolerance)

'IRR\_FUNC = CALL\_TEST\_ZERO\_FRAME\_FUNC(LOWER\_GUESS\_VAL, UPPER\_GUESS\_VAL, "CALL\_XIRR\_OBJ\_FUNC")

IRR\_FUNC = MULLER\_ZERO\_FUNC(-GUESS\_VAL, GUESS\_VAL, FORMULA\_STR, CONVERG\_VAL, COUNTER, nLOOPS, tolerance)

Exit Function

ERROR\_LABEL:

IRR\_FUNC = Err.number

End Function

'Returns the NPV for a schedule of cash flows that is not necessarily periodic.

Function NPV\_FUNC(ByVal RATE\_VAL As Double, \_

ByRef DATA\_RNG As Variant, \_

Optional ByRef DATE\_RNG As Variant, \_

Optional ByVal OUTPUT As Integer = 0)

On Error GoTo ERROR\_LABEL

PUB\_DATA\_VECTOR = DATA\_RNG

If UBound(PUB\_DATA\_VECTOR, 1) = 1 Then

PUB\_DATA\_VECTOR = MATRIX\_TRANSPOSE\_FUNC(PUB\_DATA\_VECTOR)

End If

If IsArray(DATE\_RNG) = True Then

PUB\_DATE\_VECTOR = DATE\_RNG

If UBound(PUB\_DATE\_VECTOR, 1) = 1 Then

PUB\_DATE\_VECTOR = MATRIX\_TRANSPOSE\_FUNC(PUB\_DATE\_VECTOR)

End If

If UBound(PUB\_DATE\_VECTOR, 1) <> UBound(PUB\_DATA\_VECTOR, 1) Then: GoTo ERROR\_LABEL

Call CALL\_XIRR\_OBJ\_FUNC(RATE\_VAL)

Else

Call CALL\_IRR\_OBJ\_FUNC(RATE\_VAL)

End If

Select Case OUTPUT

Case 0

NPV\_FUNC = PUB\_NPV\_VAL

Case Else

NPV\_FUNC = PUB\_NPV\_VECTOR

End Select

Exit Function

ERROR\_LABEL:

NPV\_FUNC = Err.number

End Function

'Objective functions for the internal rate of return for a

'schedule of cash flows

Private Function CALL\_IRR\_OBJ\_FUNC(ByVal X\_VAL As Double)

Debug.Print X\_VAL

Dim i As Long

Dim SROW As Long

Dim NROWS As Long

On Error GoTo ERROR\_LABEL

SROW = LBound(PUB\_DATA\_VECTOR, 1): NROWS = UBound(PUB\_DATA\_VECTOR, 1)

ReDim PUB\_NPV\_VECTOR(SROW To NROWS, 1 To 1): PUB\_NPV\_VAL = 0

For i = SROW To NROWS

If PUB\_DATA\_VECTOR(i, 1) <> "" Then

PUB\_NPV\_VECTOR(i, 1) = PUB\_DATA\_VECTOR(i, 1) / (1 + X\_VAL) ^ (i - 1)

PUB\_NPV\_VAL = PUB\_NPV\_VAL + PUB\_NPV\_VECTOR(i, 1)

End If

Next i

CALL\_IRR\_OBJ\_FUNC = PUB\_NPV\_VAL 'Abs(PUB\_NPV\_VAL) ^ 2

Exit Function

ERROR\_LABEL:

CALL\_IRR\_OBJ\_FUNC = PUB\_EPSILON

End Function

Private Function CALL\_XIRR\_OBJ\_FUNC(ByVal X\_VAL As Double)

'not necessarily periodic.

Dim i As Long

Dim SROW As Long

Dim NROWS As Long

On Error GoTo ERROR\_LABEL

SROW = LBound(PUB\_DATE\_VECTOR, 1): NROWS = UBound(PUB\_DATE\_VECTOR, 1)

ReDim PUB\_NPV\_VECTOR(SROW To NROWS, 1 To 2): PUB\_NPV\_VAL = 0

For i = SROW To NROWS

PUB\_NPV\_VECTOR(i, 1) = (PUB\_DATE\_VECTOR(i, 1) - PUB\_DATE\_VECTOR(SROW, 1)) / 365

PUB\_NPV\_VECTOR(i, 2) = PUB\_DATA\_VECTOR(i, 1) / (1 + X\_VAL) ^ PUB\_NPV\_VECTOR(i, 1)

PUB\_NPV\_VAL = PUB\_NPV\_VAL + PUB\_NPV\_VECTOR(i, 2)

Next i

CALL\_XIRR\_OBJ\_FUNC = PUB\_NPV\_VAL 'Abs(PUB\_NPV\_VAL) ^ 2

Exit Function

ERROR\_LABEL:

CALL\_XIRR\_OBJ\_FUNC = PUB\_EPSILON

End Function

'Calculates the values necessary to achieve a specific goal -

'implements the Muller's method

Function MULLER\_ZERO\_FUNC(ByVal LOWER\_VAL As Double, \_

ByVal UPPER\_VAL As Double, \_

ByVal FUNC\_NAME\_STR As String, \_

Optional ByRef CONVERG\_VAL As Integer, \_

Optional ByRef COUNTER As Long, \_

Optional ByVal nLOOPS As Long = 600, \_

Optional ByVal tolerance As Double = 1e-15)

Dim ATEMP\_VAL As Double

Dim BTEMP\_VAL As Double

Dim CTEMP\_VAL As Double

Dim DTEMP\_VAL As Double

Dim TEMP\_MID As Double

Dim TEMP\_MULT As Double

Dim TEMP\_GRAD As Double

Dim TEMP\_FUNC As Double

Dim FIRST\_FUNC As Double

Dim SECOND\_FUNC As Double

On Error GoTo ERROR\_LABEL

'-------------------------------------------------------------------------------

' MULLER implements Muller's method

'

' Parameters:

'

' Input/output, real X, X1, X2.

' On input, three distinct points that start the method.

' On output, X is an approximation to a root of the equation

' which satisfies abs ( F(X) ) < ABSERR, and X1 and X2 are the

' previous estimates.

'-------------------------------------------------------------------------------

CONVERG\_VAL = 0

COUNTER = 0

TEMP\_MID = (LOWER\_VAL + UPPER\_VAL) / 2

SECOND\_FUNC = Excel.Application.Run(FUNC\_NAME\_STR, UPPER\_VAL)

FIRST\_FUNC = Excel.Application.Run(FUNC\_NAME\_STR, LOWER\_VAL)

TEMP\_FUNC = Excel.Application.Run(FUNC\_NAME\_STR, TEMP\_MID)

' Iteration loop:

Do

'

' If the error tolerance is satisfied, then exit.

'

If (Abs(TEMP\_FUNC) <= tolerance) Then

MULLER\_ZERO\_FUNC = TEMP\_MID

Exit Function

End If

COUNTER = COUNTER + 1

If (COUNTER > nLOOPS) Then

CONVERG\_VAL = 2

MULLER\_ZERO\_FUNC = TEMP\_MID

Exit Function

End If

DTEMP\_VAL = (TEMP\_MID - LOWER\_VAL) / (LOWER\_VAL - UPPER\_VAL)

'variabile normalizzata 0 < DTEMP < 1

ATEMP\_VAL = DTEMP\_VAL \* TEMP\_FUNC - DTEMP\_VAL \* (1 + DTEMP\_VAL) \* FIRST\_FUNC + \_

DTEMP\_VAL ^ 2 \* SECOND\_FUNC

BTEMP\_VAL = (2 \* DTEMP\_VAL + 1) \* TEMP\_FUNC - (1 + DTEMP\_VAL) ^ 2 \* FIRST\_FUNC + \_

DTEMP\_VAL ^ 2 \* SECOND\_FUNC

CTEMP\_VAL = (1 + DTEMP\_VAL) \* TEMP\_FUNC

TEMP\_MULT = BTEMP\_VAL ^ 2 - 4 \* ATEMP\_VAL \* CTEMP\_VAL

If TEMP\_MULT < 0 Then TEMP\_MULT = 0

TEMP\_MULT = Sqr(TEMP\_MULT)

If (BTEMP\_VAL < 0) Then: TEMP\_MULT = -TEMP\_MULT

' Set the increment.

'

TEMP\_GRAD = -(TEMP\_MID - LOWER\_VAL) \* 2 \* CTEMP\_VAL / (BTEMP\_VAL + TEMP\_MULT)

'

' Remember current data for next step.

'

UPPER\_VAL = LOWER\_VAL

SECOND\_FUNC = FIRST\_FUNC

LOWER\_VAL = TEMP\_MID

FIRST\_FUNC = TEMP\_FUNC

'

' Update the iterate and function values.

'

TEMP\_MID = TEMP\_MID + TEMP\_GRAD

TEMP\_FUNC = Excel.Application.Run(FUNC\_NAME\_STR, TEMP\_MID)

Loop

Exit Function

ERROR\_LABEL:

MULLER\_ZERO\_FUNC = PUB\_EPSILON

End Function