

```
Public Type Curve
    curve_name As String
    overnight_rate As Single
    swap_time(0 To 17) As Integer
    swap_rate(0 To 17) As Single
    zero_curve(0 To 50) As Single
    euribor_rate(0 To 12) As Single
    df_euribor(0 To 12) As Single
    zero_curve_euribor(0 To 12) As Single
    interpolation_method As Integer
End Type
```

```
Public EUR As Curve
Public GBP As Curve
Public USD As Curve
Public JPY As Curve
```

```
Public EURversion As Integer
```

```
Sub update_curve_EUR()
```

```
Dim mat()
Dim b()
Dim zero_rate()
```

```
Dim mat_euribor()
Dim b_euribor()
Dim zero_rate_euribor()
```

```
EUR.curve_name = Worksheets("EUR").Range("B1")
EUR.interpolation_method = Worksheets("EUR").Range("B4")
```

```
EUR.overnight_rate = Worksheets("EUR").Range("C5")
EUR.euribor_rate(0) = Worksheets("EUR").Range("C5") / 100
EUR.euribor_rate(1) = Worksheets("EUR").Range("C5") / 100
EUR.euribor_rate(2) = Worksheets("EUR").Range("C6") / 100
EUR.euribor_rate(3) = Worksheets("EUR").Range("C7") / 100
EUR.euribor_rate(4) = Worksheets("EUR").Range("C8") / 100
EUR.euribor_rate(5) = Worksheets("EUR").Range("C9") / 100
EUR.euribor_rate(6) = Worksheets("EUR").Range("C10") / 100
EUR.euribor_rate(7) = Worksheets("EUR").Range("C11") / 100
EUR.euribor_rate(8) = Worksheets("EUR").Range("C12") / 100
EUR.euribor_rate(9) = Worksheets("EUR").Range("C13") / 100
EUR.euribor_rate(10) = Worksheets("EUR").Range("C14") / 100
EUR.euribor_rate(11) = Worksheets("EUR").Range("C15") / 100
EUR.euribor_rate(12) = Worksheets("EUR").Range("C16") / 100
```

```
For a1 = 1 To 17
```

```

a2 = a1 + 18
EUR.swap_time(a1) = Worksheets("EUR").Range("B" & a2)
EUR.swap_rate(a1) = Worksheets("EUR").Range("C" & a2)
Next a1

EUR.swap_rate(0) = EUR.overnight_rate
EUR.zero_curve(0) = EUR.overnight_rate / 100

ReDim mat_euribor(1 To 12, 1 To 12)
ReDim b_euribor(1 To 12)
ReDim zero_rate_euribor(0 To 12)

For i = 1 To 12
    b_euribor(i) = 1
    For j = 1 To 12
        mat_euribor(i, j) = 0

        If i = j Then mat_euribor(i, j) = EUR.euribor_rate(i) / 12 + 1
        If i > j Then mat_euribor(i, j) = EUR.euribor_rate(i) / 12
        If i < j Then mat_euribor(i, j) = 0

    Next j
Next i

disc_fact_euribor = Application.MMult(Application.MInverse(mat_euribor),
Application.Transpose(b_euribor))

For i = 1 To 12
    EUR.df_euribor(i) = disc_fact_euribor(i, 1)
    ' EUR.zero_curve_euribor(i) = ((1 / disc_fact_euribor(i, 1)) ^ (1 / i) - 1) * 12
    EUR.zero_curve_euribor(i) = ((1 / disc_fact_euribor(i, 1)) ^ (1 / (i / 12)) - 1)

Next i

EUR.df_euribor(0) = 1
EUR.zero_curve_euribor(0) = EUR.overnight_rate / 100

EURversion = EURversion + 1

tmax = 45

ReDim mat(1 To tmax, 1 To tmax)
ReDim b(1 To tmax)
ReDim zero_rate(0 To tmax)

For i = 1 To tmax
    b(i) = 1
    For j = 1 To tmax
        mat(i, j) = 0

```

```

    If i = j Then mat(i, j) = interpola(i) / 100 + 1
    If i > j Then mat(i, j) = interpola(i) / 100
    If i < j Then mat(i, j) = 0

    Next j
Next i

disc_fact = Application.MMult(Application.MInverse(mat),
Application.Transpose(b))

For i = 1 To tmax
    EUR.zero_curve(i) = (1 / disc_fact(i, 1)) ^ (1 / i) - 1
Next i
EUR.zero_curve(0) = EUR.overnight_rate

Worksheets("EUR").Range("C1").Value = "EUR:" & EURversion
Worksheets("INTEREST_RATE_SWAP").Range("C22").Value = "EUR:" &
EURversion
End Sub

Public Function forward(date1, date2)

forward = (Discount(date1) / Discount(date2)) ^ (365 / (date2 - date1)) - 1

End Function

Public Function Discount(date1)

ahora = Worksheets("EUR").Range("B3")
year1 = (date1 - ahora) / 365
tmax = 45

If year1 > 1 Then

    For a1 = 1 To tmax
        If (a1 > year1) Then
            m = a1
            a1 = tmax
        End If
    Next a1

    EUR.zero_curve(0) = EUR.overnight_rate / 100

    If (year1 >= tmax) Then linear_rate = EUR.zero_curve(tmax)

```

```
    If (year1 < tmax) Then linear_rate = EUR.zero_curve(m - 1) +  
    (EUR.zero_curve(m) - EUR.zero_curve(m - 1)) * ((year1 - (m - 1)))
```

```
End If
```

```
If year1 <= 1 Then
```

```
    month1 = year1 * 12
```

```
    For a1 = 1 To 12
```

```
        If (a1 > month1) Then
```

```
            m = a1
```

```
            a1 = 12
```

```
        End If
```

```
    Next a1
```

```
    linear_rate = EUR.zero_curve_euribor(m - 1) + (EUR.zero_curve_euribor(m) -  
    EUR.zero_curve_euribor(m - 1)) * ((month1 - (m - 1)))
```

```
End If
```

```
' Discount = 1 / (1 + linear_rate*360/365) ^ year1
```

```
Discount = 1 / (1 + linear_rate) ^ (year1)
```

```
End Function
```

```
Public Function curve_data(option1)
```

```
Dim mat_euribor()
```

```
Dim b_euribor()
```

```
Dim zero_rate_euribor()
```

```
ReDim mat_euribor(1 To 12, 1 To 12)
```

```
ReDim b_euribor(1 To 12)
```

```
ReDim zero_rate_euribor(1 To 12)
```

```
For i = 1 To 12
```

```
    b_euribor(i) = 1
```

```
    For j = 1 To 12
```

```
        mat_euribor(i, j) = 0
```

```
' 30/360 Convention
```

```
    If i = j Then mat_euribor(i, j) = EUR.euribor_rate(i) / 12 + 1
```

```
    If i > j Then mat_euribor(i, j) = EUR.euribor_rate(i) / 12
```

```
    If i < j Then mat_euribor(i, j) = 0
```

```

    Next j
Next i

disc_fact = Application.MMult(Application.MInverse(mat_euribor),
Application.Transpose(b_euribor))

```

```
' 30/360 Convention
```

```

For i = 1 To 12
    zero_rate_euribor(i) = ((1 / disc_fact(i, 1)) ^ (1 / i) - 1) * 12
Next i

```

```

If option1 = 0 Then curve_data = EUR.euribor_rate
If option1 = 1 Then curve_data = mat_euribor
If option1 = 2 Then curve_data = disc_fact
If option1 = 3 Then curve_data = zero_rate_euribor
If option1 = 4 Then curve_data = EUR.df_euribor
If option1 = 5 Then curve_data = EUR.zero_curve_euribor
If option1 = 6 Then curve_data = EUR.swap_rate

```

```
End Function
```

```
Public Function interpola(year1)
```

```

For a1 = 1 To 17
    If (EUR.swap_time(a1) > year1) Then
        m = a1
        a1 = 17
    End If
Next a1

```

```
If (year1 >= EUR.swap_time(17)) Then m = 17
```

```
EUR.swap_rate(0) = EUR.overnight_rate
```

```
If (year1 >= EUR.swap_time(17)) Then linear_rate = EUR.swap_rate(17)
```

```

If (year1 < EUR.swap_time(17)) Then linear_rate = EUR.swap_rate(m - 1) +
(EUR.swap_rate(m) - EUR.swap_rate(m - 1)) * ((year1 - EUR.swap_time(m - 1)) /
(EUR.swap_time(m) - EUR.swap_time(m - 1)))

```

```
interpola = linear_rate
```

```
End Function
```

```
Public Function interpola_euribor(month1)
```

```
For a1 = 1 To 12
  If (a1 > month1) Then
    m = a1
    a1 = 12
  End If
Next a1

If (month1 >= 12) Then m = 12

EUR.euribor_rate(0) = EUR.overnight_rate

If (month1 >= 12) Then linear_rate = EUR.euribor_rate(12)

If (month1 < 12) Then linear_rate = EUR.euribor_rate(m - 1) +
(EUR.euribor_rate(m) - EUR.euribor_rate(m - 1)) * ((month1 - (m - 1)))

interpolated_euribor = linear_rate

End Function
```