

Códigos VBA para algoritmo utilizado no cálculo da volatilidade implícita do modelo Corrado-Su (1996)

VBA Vale

a. Funções

```
Function GBlackScholes(CallPutFlag As String, S As Double, X As Double, T As Double, r As Double, b As Double, v As Double) As Double
Dim d1 As Double, d2 As Double
d1 = (Math.Log(S / X) + (b + v ^ 2 / 2) * T) / (v * Math.Sqrt(T))
d2 = d1 - v * Math.Sqrt(T)
If CallPutFlag = "c" Then
    GBlackScholes = S * Math.Exp((b - r) * T) *
WorksheetFunction.NormSDist(d1) - X * Math.Exp(-r * T) *
WorksheetFunction.NormSDist(d2)
ElseIf CallPutFlag = "p" Then
    GBlackScholes = X * Math.Exp(-r * T) * WorksheetFunction.NormSDist(-d2)
- S * Math.Exp((b - r) * T) * WorksheetFunction.NormSDist(-d1)
End If
End Function
```

```
Function SkewKurtCorradoSu(CallPutFlag As String, S As Double, X As Double, T As Double, r As Double, b As Double, v As Double, Skew As Double, Kurt As Double) As Double
Dim Q3 As Double, Q4 As Double
Dim d1 As Double, d2 As Double
Dim CallValue As Double
d1 = (Math.Log(S / X) + (b + v ^ 2 / 2) * T) / (v * Math.Sqrt(T))
d2 = d1 - v * Math.Sqrt(T)
Q4 = 1 / 24 * S * v * Math.Sqrt(T) * ((d1 ^ 2 - 1 - 3 * v * Math.Sqrt(T) * d2) * (1 / Math.Sqrt(2 * 3.14159265) * Math.Exp(-d1 ^ 2 / 2)) + v ^ 3 * T ^ 1.5 *
WorksheetFunction.NormSDist(d1))
Q3 = 1 / 6 * S * v * Math.Sqrt(T) * ((2 * v * Math.Sqrt(T) - d1) * (1 / Math.Sqrt(2 * 3.14159265) * Math.Exp(-d1 ^ 2 / 2)) + v ^ 2 * T *
WorksheetFunction.NormSDist(d1))
CallValue = GBlackScholes("c", S, X, T, r, b, v) + Skew * Q3 + (Kurt - 3) * Q4
If CallPutFlag = "c" Then
    SkewKurtCorradoSu = CallValue
Else
    SkewKurtCorradoSu = CallValue - S * Math.Exp((b - r) * T) + X * Math.Exp(-r * T)
End If
End Function
```

```

Function CSImpliedVolatilityBisection(CallPutFlag As String, S As Double, X
As Double, T As Double, r As Double, b As Double, Skew As Double, Kurt As
Double, cm As Double) As Variant
Dim vLow As Double, vHigh As Double, vi As Double
Dim cLow As Double, cHigh As Double, epsilon As Double
Dim counter As Integer
vLow = 0.005
vHigh = 4
epsilon = 0.001
cLow = SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vLow, Skew, Kurt)
cHigh = SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vHigh, Skew, Kurt)
counter = 0
vi = vLow + (cm - cLow) * (vHigh - vLow) / (cHigh - cLow)
While Abs(cm - SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vi, Skew, Kurt))
> epsilon
    counter = counter + 1
    If counter = 1000 Then
        CSImpliedVolatilityBisection = "NA"
        Exit Function
    End If
    If SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vi, Skew, Kurt) < cm Then
        vLow = vi
    Else
        vHigh = vi
    End If
    cLow = SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vLow, Skew, Kurt)
    cHigh = SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vHigh, Skew, Kurt)
    vi = vLow + (cm - cLow) * (vHigh - vLow) / (cHigh - cLow)
Wend
CSImpliedVolatilityBisection = vi
End Function

```

b. Aplicações

```

Sub vol_plan1()

Application.ScreenUpdating = False
Dim v() As Double
For i = 4 To 2832
    datapreg = Plan1.Cells(i, 2)
    dataexe = Plan1.Cells(i, 10)
    'ndias = Plan1.Cells(i, 12) + 5
    For j = 2 To 1940
        datatest = Plan2.Cells(j, 1)
        If datatest = datapreg Then
            linha1 = j
        ElseIf datatest = dataexe Then
            linha2 = j
        End If
    Next
    ReDim v(linha2 - linha1)

```

```

Z = 0
For k = linha1 To linha2
    retorno = Plan2.Cells(k, 3)
    v(Z) = retorno
    Z = Z + 1
Next
desvpad = WorksheetFunction.StDev(v)
volhist = (desvpad) * ((252) ^ (1 / 2))
Plan1.Cells(i, 26) = volhist
Next
Application.ScreenUpdating = True
End Sub
Sub assim_kurt()
Application.ScreenUpdating = False
Dim v() As Double
For i = 4 To 2832
    datapreg = Plan1.Cells(i, 2)
    For j = 2 To 1940
        datateste = Plan2.Cells(j, 1)
        If datateste = datapreg Then
            linha1 = j
        End If
    Next
    ReDim v(42)
    Z = 0
    For k = linha1 - 42 To linha1
        retorno = Plan2.Cells(k, 3)
        v(Z) = retorno
        Z = Z + 1
    Next
    assim = WorksheetFunction.Skew(v)
    curt = WorksheetFunction.Kurt(v)
    Plan1.Cells(i, 27) = assim
    Plan1.Cells(i, 28) = curt
Next
Application.ScreenUpdating = True
End Sub
Sub VolImpCS()
Application.ScreenUpdating = False
For i = 4 To 2832
    If IsNumeric(Plan1.Cells(i, 9)) = True Then
        Plan1.Cells(i, 31) = Funcs.CSImpliedVolatilityBisection(Plan1.Cells(i, 29),
        Plan1.Cells(i, 15), Plan1.Cells(i, 14), Plan1.Cells(i, 30), Plan1.Cells(i, 17),
        Plan1.Cells(i, 17), Plan1.Cells(i, 27), Plan1.Cells(i, 28), Plan1.Cells(i, 9))
        On Error Resume Next
    End If
Next
Application.ScreenUpdating = True
End Sub
Sub calcula_tudo()

```

```

Call Plan1.assim_kurt
Call Plan1.VolImpCS
End Sub

```

VBA Petro

a. Funções

```

Function GBlackScholes(CallPutFlag As String, S As Double, X As Double, T As Double, r As Double, b As Double, v As Double) As Double
Dim d1 As Double, d2 As Double
d1 = (Math.Log(S / X) + (b + v ^ 2 / 2) * T) / (v * Math.Sqrt(T))
d2 = d1 - v * Math.Sqrt(T)
If CallPutFlag = "c" Then
    GBlackScholes = S * Math.Exp((b - r) * T) *
WorksheetFunction.NormSDist(d1) - X * Math.Exp(-r * T) *
WorksheetFunction.NormSDist(d2)
ElseIf CallPutFlag = "p" Then
    GBlackScholes = X * Math.Exp(-r * T) * WorksheetFunction.NormSDist(-d2)
- S * Math.Exp((b - r) * T) * WorksheetFunction.NormSDist(-d1)
End If
End Function

```

```

Function SkewKurtCorradoSu(CallPutFlag As String, S As Double, X As Double, T As Double, r As Double, b As Double, v As Double, Skew As Double, Kurt As Double) As Double
Dim Q3 As Double, Q4 As Double
Dim d1 As Double, d2 As Double
Dim CallValue As Double
d1 = (Math.Log(S / X) + (b + v ^ 2 / 2) * T) / (v * Math.Sqrt(T))
d2 = d1 - v * Math.Sqrt(T)
Q4 = 1 / 24 * S * v * Math.Sqrt(T) * ((d1 ^ 2 - 1 - 3 * v * Math.Sqrt(T) * d2) * (1 / Math.Sqrt(2 * 3.14159265) * Math.Exp(-d1 ^ 2 / 2)) + v ^ 3 * T ^ 1.5 *
WorksheetFunction.NormSDist(d1))
Q3 = 1 / 6 * S * v * Math.Sqrt(T) * ((2 * v * Math.Sqrt(T) - d1) * (1 / Math.Sqrt(2 * 3.14159265) * Math.Exp(-d1 ^ 2 / 2)) + v ^ 2 * T *
WorksheetFunction.NormSDist(d1))
CallValue = GBlackScholes("c", S, X, T, r, b, v) + Skew * Q3 + (Kurt - 3) * Q4
If CallPutFlag = "c" Then
    SkewKurtCorradoSu = CallValue
Else
    SkewKurtCorradoSu = CallValue - S * Math.Exp((b - r) * T) + X * Math.Exp(-r * T)
End If
End Function

```

```

Function CSImpliedVolatilityBisection(CallPutFlag As String, S As Double, X
As Double, T As Double, r As Double, b As Double, Skew As Double, Kurt As
Double, cm As Double) As Variant
Dim vLow As Double, vHigh As Double, vi As Double
Dim cLow As Double, cHigh As Double, epsilon As Double
Dim counter As Integer
vLow = 0.005
vHigh = 4
epsilon = 0.001
cLow = SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vLow, Skew, Kurt)
cHigh = SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vHigh, Skew, Kurt)
counter = 0
vi = vLow + (cm - cLow) * (vHigh - vLow) / (cHigh - cLow)
While Abs(cm - SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vi, Skew, Kurt))
> epsilon
    counter = counter + 1
    If counter = 1000 Then
        CSImpliedVolatilityBisection = "NA"
        Exit Function
    End If
    If SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vi, Skew, Kurt) < cm Then
        vLow = vi
    Else
        vHigh = vi
    End If
    cLow = SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vLow, Skew, Kurt)
    cHigh = SkewKurtCorradoSu(CallPutFlag, S, X, T, r, b, vHigh, Skew, Kurt)
    vi = vLow + (cm - cLow) * (vHigh - vLow) / (cHigh - cLow)
Wend
CSImpliedVolatilityBisection = vi
End Function

```

b. Aplicações

```

Sub calcula_vol()
Application.ScreenUpdating = False
    Call vol_plan1
    Call vol_plan2
    Call vol_plan3
    Call vol_plan4
Application.ScreenUpdating = True
End Sub

Sub vol_plan1()
Dim v() As Double
For i = 4 To 37
    datapreg = Plan1.Cells(i, 2)
    dataexe = Plan1.Cells(i, 10)
    ndias = Plan1.Cells(i, 12) + 5
    For j = 2 To 2792

```

```

        datatest = Plan6.Cells(j, 1)
        If datatest = datapreg Then
            linha1 = j
        ElseIf datatest = dataexe Then
            linha2 = j
        End If
    Next
    ReDim v(linha2 - linha1)
    Z = 0
    For k = linha1 To linha2
        retorno = Plan6.Cells(k, 3)
        v(Z) = retorno
        Z = Z + 1
    Next
    desvpad = WorksheetFunction.StDev(v)
    volhist = (desvpad) * ((252) ^ (1 / 2))
    Plan1.Cells(i, 21) = volhist
Next
End Sub
Sub vol_plan2()
Dim v() As Double
For i = 4 To 38
    datapreg = Plan2.Cells(i, 2)
    dataexe = Plan2.Cells(i, 10)
    ndias = Plan2.Cells(i, 12) + 5
    For j = 2 To 2792
        datatest = Plan6.Cells(j, 1)
        If datatest = datapreg Then
            linha1 = j
        ElseIf datatest = dataexe Then
            linha2 = j
        End If
    Next
    ReDim v(linha2 - linha1)
    Z = 0
    For k = linha1 To linha2
        retorno = Plan6.Cells(k, 3)
        v(Z) = retorno
        Z = Z + 1
    Next
    desvpad = WorksheetFunction.StDev(v)
    volhist = (desvpad) * ((252) ^ (1 / 2))
    Plan2.Cells(i, 21) = volhist
Next
End Sub
Sub vol_plan3()
Application.ScreenUpdating = False
Dim v() As Double
For i = 4 To 4507
    datapreg = Plan3.Cells(i, 2)

```

```

dataexe = Plan3.Cells(i, 10)
ndias = Plan3.Cells(i, 12) + 5
For j = 2 To 2792
    datateste = Plan6.Cells(j, 1)

    If datateste = datapreg Then
        linha1 = j
    ElseIf datateste = dataexe Then
        linha2 = j
    End If
Next
ReDim v(linha2 - linha1)
Z = 0
For k = linha1 To linha2
    retorno = Plan6.Cells(k, 3)
    v(Z) = retorno
    Z = Z + 1
Next
desvpad = WorksheetFunction.StDev(v)
volhist = (desvpad) * ((252) ^ (1 / 2))
Plan3.Cells(i, 21) = volhist
Next
Application.ScreenUpdating = True
End Sub
Sub vol_plan4()
Dim v() As Double
For i = 4 To 322
    datapreg = Plan4.Cells(i, 2)
    dataexe = Plan4.Cells(i, 10)
    ndias = Plan4.Cells(i, 12) + 5
    For j = 2 To 2792
        datateste = Plan6.Cells(j, 1)
        If datateste = datapreg Then
            linha1 = j
        ElseIf datateste = dataexe Then
            linha2 = j
        End If
    Next
    ReDim v(linha2 - linha1)
    Z = 0
    For k = linha1 To linha2
        retorno = Plan6.Cells(k, 3)
        v(Z) = retorno
        Z = Z + 1
    Next
    desvpad = WorksheetFunction.StDev(v)
    volhist = (desvpad) * ((252) ^ (1 / 2))
    Plan4.Cells(i, 21) = volhist
Next
End Sub

```

```

Sub assim_kurt_plan1()
Application.ScreenUpdating = False
Dim v() As Double
For i = 4 To 37
    datapreg = Plan1.Cells(i, 2)
    For j = 2 To 2792
        datatest = Plan6.Cells(j, 1)
        If datatest = datapreg Then
            linha1 = j
        End If
    Next
    ReDim v(42)
    Z = 0
    For k = linha1 - 42 To linha1
        retorno = Plan6.Cells(k, 3)
        v(Z) = retorno
        Z = Z + 1
    Next
    assim = WorksheetFunction.Skew(v)
    curt = WorksheetFunction.Kurt(v)
    Plan1.Cells(i, 22) = assim
    Plan1.Cells(i, 23) = curt
Next
Application.ScreenUpdating = True
End Sub

Sub assim_kurt_plan2()
Application.ScreenUpdating = False
Dim v() As Double
For i = 4 To 38
    datapreg = Plan2.Cells(i, 2)
    For j = 2 To 2792
        datatest = Plan6.Cells(j, 1)
        If datatest = datapreg Then
            linha1 = j
        End If
    Next
    ReDim v(42)
    Z = 0
    For k = linha1 - 42 To linha1
        retorno = Plan6.Cells(k, 3)
        v(Z) = retorno
        Z = Z + 1
    Next
    assim = WorksheetFunction.Skew(v)
    curt = WorksheetFunction.Kurt(v)
    Plan2.Cells(i, 22) = assim
    Plan2.Cells(i, 23) = curt
Next
Application.ScreenUpdating = True
End Sub

```



```

Sub assim_kurt_plan3()
Application.ScreenUpdating = False
Dim v() As Double
For i = 4 To 4507
    datapreg = Plan3.Cells(i, 2)
    For j = 2 To 2792
        datatest = Plan6.Cells(j, 1)
        If datatest = datapreg Then
            linha1 = j
        End If
    Next
    ReDim v(42)
    Z = 0
    For k = linha1 - 42 To linha1
        retorno = Plan6.Cells(k, 3)
        v(Z) = retorno
        Z = Z + 1
    Next
    assim = WorksheetFunction.Skew(v)
    curt = WorksheetFunction.Kurt(v)
    Plan3.Cells(i, 22) = assim
    Plan3.Cells(i, 23) = curt
Next
Application.ScreenUpdating = True
End Sub

Sub assim_kurt_plan4()
Application.ScreenUpdating = False
Dim v() As Double
For i = 4 To 322
    datapreg = Plan4.Cells(i, 2)
    For j = 2 To 2792
        datatest = Plan6.Cells(j, 1)
        If datatest = datapreg Then
            linha1 = j
        End If
    Next
    ReDim v(42)
    Z = 0
    For k = linha1 - 42 To linha1
        retorno = Plan6.Cells(k, 3)
        v(Z) = retorno
        Z = Z + 1
    Next
    assim = WorksheetFunction.Skew(v)
    curt = WorksheetFunction.Kurt(v)
    Plan4.Cells(i, 22) = assim
    Plan4.Cells(i, 23) = curt
Next
Application.ScreenUpdating = True
End Sub

```

```

Sub VolImpCS_plan1()
Application.ScreenUpdating = False
For i = 4 To 37
    If IsNumeric(Plan1.Cells(i, 9)) = True Then
        Plan1.Cells(i, 26) = Funcs.CSImpliedVolatilityBisection(Plan1.Cells(i, 24),
Plan1.Cells(i, 15), Plan1.Cells(i, 14), Plan1.Cells(i, 25), Plan1.Cells(i, 17),
Plan1.Cells(i, 17), Plan1.Cells(i, 22), Plan1.Cells(i, 23), Plan1.Cells(i, 9))
        On Error Resume Next
    End If
Next
Application.ScreenUpdating = True
End Sub
Sub VolImpCS_plan2()
Application.ScreenUpdating = False
For i = 4 To 38
    If IsNumeric(Plan2.Cells(i, 9)) = True Then
        Plan2.Cells(i, 26) = Funcs.CSImpliedVolatilityBisection(Plan2.Cells(i,
24), Plan2.Cells(i, 15), Plan2.Cells(i, 14), Plan2.Cells(i, 25), Plan2.Cells(i, 17),
Plan2.Cells(i, 17), Plan2.Cells(i, 22), Plan2.Cells(i, 23), Plan2.Cells(i, 9))
        On Error Resume Next
    End If
Next
Application.ScreenUpdating = True
End Sub
Sub VolImpCS_plan3()
Application.ScreenUpdating = False
For i = 4 To 4507
    If IsNumeric(Plan3.Cells(i, 9)) = True Then
        Plan3.Cells(i, 26) = Funcs.CSImpliedVolatilityBisection(Plan3.Cells(i, 24),
Plan3.Cells(i, 15), Plan3.Cells(i, 14), Plan3.Cells(i, 25), Plan3.Cells(i, 17),
Plan3.Cells(i, 17), Plan3.Cells(i, 22), Plan3.Cells(i, 23), Plan3.Cells(i, 9))
        On Error Resume Next
    End If
Next
Application.ScreenUpdating = True
End Sub
Sub VolImpCS_plan4()
Application.ScreenUpdating = False
For i = 4 To 322
    If IsNumeric(Plan4.Cells(i, 9)) = True Then
        Plan4.Cells(i, 26) = Funcs.CSImpliedVolatilityBisection(Plan4.Cells(i, 24),
Plan4.Cells(i, 15), Plan4.Cells(i, 14), Plan4.Cells(i, 25), Plan4.Cells(i, 17),
Plan4.Cells(i, 17), Plan4.Cells(i, 22), Plan4.Cells(i, 23), Plan4.Cells(i, 9))
        On Error Resume Next
    End If
Next
Application.ScreenUpdating = True
End Sub
Sub calcula_tudo()
Call Plan1.assim_kurt_plan1

```

```
Call Plan1.assim_kurt_plan2  
Call Plan1.assim_kurt_plan3  
Call Plan1.assim_kurt_plan4  
Call Plan1.VolImpCS_plan1  
Call Plan1.VolImpCS_plan2  
Call Plan1.VolImpCS_plan3  
Call Plan1.VolImpCS_plan4  
End Sub
```