

**El algoritmo de Ruffini (Horner) y su generalización para ordenadores.  
Aladar Peter Santha**

**Teorema 1:** Si  $f(X) = a_n X^n + a_{n-1} X^{n-1} + \dots + a_2 X^2 + a_1 X + a_0$  y  $g(X) = X - a$ , son dos polinomios con coeficientes en un anillo conmutativo con elemento neutro, entonces existe un polinomio  $q(X) \in A[X]$  de grado  $n-1$  y  $r \in A$  tales que

$$f(X) = (X - a)q(X) + r \quad (1)$$

Demostración: Suponiendo que

$$\begin{aligned} q(X) &= b_{n-1} X^{n-1} + b_{n-2} X^{n-2} + \dots + b_1 X + b_0 \\ (X - a)q(X) + r &= b_{n-1} X^n + (b_{n-2} - ab_{n-1}) X^{n-1} + (b_{n-3} - ab_{n-2}) X^{n-2} + \dots \\ &\quad \dots + (b_2 - ab_3) X^3 + (b_1 - ab_2) X^2 + (b_0 - ab_1) X + r - ab_0 \end{aligned}$$

, de la igualdad (1) resulta que

$$\begin{aligned} b_{n-1} &= a_n \\ b_{n-2} - ab_{n-1} &= a_{n-1} \Leftrightarrow b_{n-2} = a_{n-1} + ab_{n-1} \\ b_{n-3} - ab_{n-2} &= a_{n-2} \Leftrightarrow b_{n-3} = a_{n-2} + ab_{n-2} \\ &\dots\dots\dots \\ b_2 - ab_3 &= a_3 \Leftrightarrow b_2 = a_3 + ab_3 \\ b_1 - ab_2 &= a_2 \Leftrightarrow b_1 = a_2 + ab_2 \\ b_0 - ab_1 &= a_1 \Leftrightarrow b_0 = a_1 + ab_1 \\ r - ab_0 &= a_0 \Leftrightarrow r = a_0 + ab_0 \end{aligned} \quad (2)$$

Según las igualdades (2), al conocer el coeficiente  $b_i$  del polinomio  $q(X)$ , el coeficiente  $b_{i-1}$  se puede calcular según la fórmula siguiente:

$$b_{i-1} = a_i + ab_i \quad (i = n-1, \dots, 0).$$

Este procedimiento sencillo para el cálculo de los coeficientes del cociente, entre el polinomio  $f(X)$  y el polinomio  $X - a$ , se llama el algoritmo de Ruffini (ó Hörner). El cálculo se puede organizar según el esquema siguiente:

	$a_n$	$a_{n-1}$	$a_{n-2}$	...	$a_2$	$a_1$	$a_0$
$a$		$ab_{n-1}$	$ab_{n-2}$	...	$ab_2$	$ab_1$	$ab_0$
	$b_{n-1} = a_n$	$b_{n-2}$	$b_{n-3}$	...	$b_1$	$b_0$	$r$

Para rellenar la tabla, se completan primero las casillas con los coeficientes del polinomio y la casilla con el valor de  $a$ . Como segundo paso, en la casilla que se encuentra en la columna de  $a_n$  y la tercera fila, se coloca el valor de  $a_n$ . En la tercera etapa, se rellenan progresivamente las casillas de la segunda y la tercera fila. Para esto se multiplica con  $a$  el contenido de la última casilla rellena en la tercera fila y se coloca el producto en la casilla de la segunda fila de la columna siguiente, luego se suma el contenido de ésta última casilla con el contenido de la casilla que se encuentra en la primera fila de la misma columna, colocando la suma en la tercera fila de ésta columna. El proceso se continúa hasta que la tabla quedará rellena.

**Ejemplo 1:** Si  $A = R$ ,  $f(X) = X^4 - 2X^2 + 2X + 5$  y  $a = 2$ , entonces se obtiene la tabla siguiente:

	1	0	-2	3	-5
2		2	4	4	14
	1	2	2	7	11

, de donde resulta que

$$q(X) = X^3 + 2X^2 + 2X + 7 \text{ y } r = 11.$$

Si los coeficientes del polinomio y el valor del parámetro  $a$  son enteros o decimales, la codificación del cálculo del ejemplo 1 es la siguiente:

```
Public Function RuffiniAR(ByRef p() As Double, ByVal a As Double) As String
    Dim q() As Double, g As Integer, i As Integer
    Dim res As String, c() As Double, rc As String
    g = UBound(p()): rc = Chr$(13) + Chr$(10)
    ReDim q(g), c(g - 1)
    q(0) = p(0)
    For i = 0 To g - 1
        q(i + 1) = q(i) * a + p(i + 1)
    Next i
    For i = 0 To g - 1: c(i) = q(i): Next i 'c() es el cociente
    res = "Cociente = " + FormatoPolR(c()) + rc
    RuffiniAR = res + "Resto = " + Str$(q(g)) ' q(g) es el resto
End Function

Public Function FormatoPolR(ByRef qx() As Double) As String
    Dim i As Integer, pol As String
    gx = UBound(qx())
    If Abs(qx(0)) = 1 Then
        If qx(0) = 1 Then
            If gx > 1 Then
                pol = pol + "X^" + Str$(gx)
            Else
                pol = pol + "X"
            End If
        End If
        If qx(0) = -1 Then
            If gx > 1 Then
                pol = pol + "- X^" + Str$(gx)
            Else
                pol = pol + "- X"
            End If
        End If
    Else
        If Abs(qx(0)) > 1 Then
            pol = pol + Str$(qx(0))
        Else
            If qx(0) > 0 Then
                pol = pol + "0" + Mid$(Str$(Abs(qx(0))), 2)
            Else
                pol = pol + "- 0" + Mid$(Str$(Abs(qx(0))), 2)
            End If
        End If
        If gx > 1 Then
            pol = pol + " X^" + Str$(gx)
        Else
            pol = pol + " X"
        End If
    End If
    For i = 1 To gx
        If qx(i) <> 0 Then
            If qx(i) < 0 Then
                pol = pol + " - "
            Else
                pol = pol + " + "
            End If
            If Abs(qx(i)) <> 1 Then
                If Abs(qx(i)) > 1 Then
                    pol = pol + Mid$(Str$(Abs(qx(i))), 2)
                End If
            End If
        End If
    Next i
End Function
```

```

Else
    pol = pol + "0" + Mid$(Str$(Abs(qx(i))), 2)
End If
Else
    If i = gx Then
        pol = pol + Mid$(Str$(1), 2)
    End If
End If
If i < gx Then
    pol = pol + "X"
End If
If i < gx - 1 Then
    pol = pol + "^" + Str$(gx - i)
End If
End If
Next i
VerPolR = pol
End Function

```

**Observación 1:** Si se pone  $X = a$  en la igualdad (1) resulta que  $f \stackrel{\sim}{=} r$ , es decir el valor del polinomio  $f \stackrel{\sim}{=}$  en el punto  $a$  es el resto de la división del polinomio con  $X - a$ . De hecho, el esquema de Ruffini se utiliza más bien para calcular los valores de las funciones de tipo polinomio (este es el caso cuando se aproximan los ceros de un polinomio por el método de la bipartición). La función *RuffiniAR* se puede adaptar a este caso, cuando no interesan los coeficientes del cociente:

```

Public Function ValPolR(ByRef p() As Double, ByVal a As Double) As Double
    Dim i As Integer, gp As Integer, q As Double
    gp = UBound(p()): q = p(gp)
    For i = 0 To gp - 1: q = q * a + p(i + 1): Next i
    ValPolR = q
End Function

```

Si el polinomio o el parámetro  $a$  tienen valores que no son ni enteros ni decimales, entonces considerando valores aproximados para ellos, el código *RuffiniA* devolverá un resultado que es también aproximado. Por ejemplo, si

$$f \stackrel{\sim}{=} X^4 + \sqrt{3}X - 2X^2 + 2X + 5 \text{ y } a = 7/3$$

Entonces considerando las aproximaciones  $\sqrt{3} \approx 1.732050807588$  y  $7/3 \approx 2.333333333333$ , la función *RuffiniA* devolverá el resultado siguiente:

$$q \stackrel{\sim}{=} X^3 + q_1 X^2 + q_2 X + q_3 \text{ y } r \approx 50.4232133...$$

, donde  $q_1 \approx 4.065384140901$ ,  $q_2 \approx 7.48589632876764$  y  $q_3 \approx 4670914337887$ .

La precisión de estos últimos cálculos no es muy grande, teniendo en cuenta que haciendo los cálculos a mano se obtiene  $r = 50.42319...$ .

Si en la resolución de las ecuaciones algebraicas por el método de la bipartición se quieren calcular los ceros con más de 16 decimales después del punto decimal, los valores de la polinomio hay que calcular con el método de Ruffini adaptado a las operaciones con enteros y decimales largos:

```

Public Function ValPolRG(ByRef p() As String, ByVal a As String) As String
    Dim i As Integer, gp As Integer, q As String, x(2) As String, n As Integer
    gp = UBound(p()): q = p(gp): n = 7
    For i = 0 To gp - 1
        x(1) = q: x(2) = a: x(1) = MultiplicarDec(x(1), x(2))
        x(2) = p(i + 1): q = SumarDec(x(1), x(2))
    Next i
    ValPolRG = q
End Function

```

**Ejemplo 2:** Si  $A = C$ ,  $f \stackrel{\sim}{=} C - 3i \stackrel{\sim}{=} X^4 - 2X^2 + C + i \stackrel{\sim}{=} X + 7 - 4i$  y  $a = 3 + i$ , entonces se obtiene la tabla siguiente:

	$2-3i$	$0$	$-2$	$5+i$	$7-4i$
$3+i$		$9-7i$	$34-12i$	$108-4i$	$342+104i$
	$2-3i$	$9-7i$	$32-12i$	$113-3i$	$349+100i$

Así,

$$q(X) = (2-3i)X^3 + (9-7i)X^2 + (34-12i)X + 113-3i \quad \text{y} \quad r = 349+100i$$

Los cálculos se pueden programar para ordenadores de la manera siguiente:

Public Function RuffiniAC(ByRef p1() As Double, ByRef p2() As Double, ByRef a() As Double) As Variant

Dim i As Integer, gx As Integer, coci As String, r As String, rc As String

Dim q() As Double, x() As Double, rt() As Double, ra As String

gx = UBound(p1()): rc = Chr\$(13) + Chr\$(10)

ReDim q(gx, 2), x(2)

q(0, 1) = p1(0): q(0, 2) = p2(0)

For i = 1 To gx

    x(1) = q(i - 1, 1): x(2) = q(i - 1, 2)

    rt() = ProdNC(x(), a())

    q(i, 1) = rt(1) + p1(i): q(i, 2) = rt(2) + p2(i)

Next i

ReDim q1(gx - 1), q2(gx - 1)

For i = 0 To gx - 1

    q1(i) = q(i, 1): q2(i) = q(i, 2)

Next i

coci = FormatoPolinomioComplejo(q1(), q2())

ra = "Cociente: " + coci + rc

r = FormatoNumeroComplejo(q(gx, 1), q(gx, 2))

ra = ra + rc + "Resto de la división = " + r

RuffiniAC = ra

End Function

Public Function ProdNC(ByRef x() As Double, ByRef a() As Double) As Variant

Dim pr() As Double

ReDim pr(2)

pr(1) = x(1) \* a(1) - x(2) \* a(2)

pr(2) = x(1) \* a(2) + a(1) \* x(2)

ProdNC = pr()

End Function

Public Function FormatoPolinomioComplejo(ByRef z1() As Double, ByRef z2() As Double) As String

Dim i As Integer, j As Integer, gx As Integer

Dim cd As String, cm As String

gx = UBound(z1())

For i = 0 To gx

    If z1(i) <> 0 Or z2(i) <> 0 Then

        If i = 0 Then

            If z2(0) = 0 Then

                If Abs(z1(0)) <> 1 Then

                    cm = f2(z1(0))

            Else

                If gx <> 0 Then

                    If z1(0) = -1 Then cm = "-"

                Else

                    If z1(0) = -1 Then cm = Str\$(-1) Else cm = Mid\$(Str\$(1), 2)

                End If

            End If

        Else

            If gx <> 0 Then

                If z1(0) <> 0 Then

                    cm = cm + "(" + f2(z1(0))

                End If

            If Abs(z2(0)) <> 1 Then

                If z2(0) <> 0 Then

                    cm = cm + " + " + f1(Mid\$(Str\$(z2(0)), 2)) + " i )"

                Else

                    cm = cm + " - " + f1(Mid\$(Str\$(z2(0)), 2)) + " i )"

```

        End If
    Else
        cm = cm + f2(z2(0)) + " i"
    End If
Else
    If z2(0) = 1 Then cm = cm + " + i )" Else cm = cm + " - i )"
End If
Else
    cm = cm + f2(z1(0))
    If Abs(z2(0)) <> 1 Then
        If z2(0) > 0 Then
            cm = cm + " + " + f1(Mid$(Str$(z2(0)), 2)) + " i"
        Else
            cm = cm + " - " + f1(Mid$(Str$(z2(0)), 2)) + " i"
        End If
    Else
        If z2(0) = 1 Then cm = cm + " + i" Else cm = cm + " - i"
    End If
End If
End If
If gx <> 0 Then
    If gx = 1 Then
        cm = cm + " X "
    Else
        cm = cm + " X^" + Mid$(Str$(gx), 2)
    End If
End If
Else
    If Abs(z2(i)) <> 0 Then
        If i < gx Then
            If z1(i) <> 0 Then
                cd = cd + " + (" + f2((z1(i)))
                If Abs(z2(i)) <> 1 Then
                    If z2(i) > 0 Then
                        cd = cd + " + " + f1(Mid$(Str$(z2(i)), 2)) + " i )"
                    Else
                        If z2(i) < 0 Then
                            cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " i )"
                        End If
                    End If
                Else
                    If z2(i) = 1 Then cd = cd + " + i )" Else cd = cd + " - i )"
                End If
            Else
                If Abs(z2(i)) <> 1 Then
                    If z2(i) < 0 Then
                        cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " i"
                    Else
                        If z2(i) > 0 Then
                            cd = cd + " + " + f1(Mid$(Str$(z2(i)), 2)) + " i"
                        End If
                    End If
                Else
                    If z2(i) = 1 Then cd = cd + " + i" Else cd = cd + " - i"
                End If
            End If
        Else
            If z1(i) > 0 Then
                cd = cd + " + " + f1(Mid$(Str$(z1(i)), 2))
            Else
                If z1(i) < 0 Then
                    cd = cd + " - " + f1(Mid$(Str$(z1(i)), 2))
                End If
            End If
        End If
    End If
    If Abs(z2(i)) <> 1 Then
        If z2(i) > 0 Then
            cd = cd + " + " + f1(Mid$(Str$(z2(i)), 2)) + " i"
        Else
            If z2(i) < 0 Then
                cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " i"
            End If
        End If
    Else
        If z2(i) = 1 Then cd = cd + " + i" Else cd = cd + " - i"
    End If
End If
Else
    If z1(i) > 0 Then
        cd = cd + " + " + f1(Mid$(Str$(z1(i)), 2))
    Else
        If z1(i) < 0 Then
            cd = cd + " - " + f1(Mid$(Str$(z1(i)), 2))
        End If
    End If
    If Abs(z2(i)) <> 1 Then
        If z2(i) > 0 Then
            cd = cd + " + " + f1(Mid$(Str$(z2(i)), 2)) + " i"
        Else
            If z2(i) < 0 Then
                cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " i"
            End If
        End If
    Else
        If z2(i) = 1 Then cd = cd + " + i" Else cd = cd + " - i"
    End If
End If
End If

```

```

Else
    If Abs(z1(i)) <> 0 Then
        If Abs(z1(i)) <> 1 Then
            If z1(i) < 0 Then
                cd = cd + " - " + f1(Mid$(Str$(z1(i)), 2))
            Else
                cd = cd + " + " + f1(Mid$(Str$(z1(i)), 2))
            End If
        Else
            If z1(i) = 1 Then cd = cd + " + " Else cd = cd + " - "
        End If
    End If
End If
If gx > 1 Then
    If i < gx - 1 Then
        cd = cd + " X^" + Mid$(Str$(gx - i), 2)
    Else
        If i = gx - 1 Then
            cd = cd + " X "
        End If
    End If
End If
cm = cm + cd: cd = ""
End If
End If
Next i
FormatoPolinomioComplejo = cm
End Function
'-----
Public Function FormatoNumeroComplejo(ByVal pr As Double, ByVal pi As Double) As String
    Dim r As String
    If pr <> 0 Then
        r = r + f2(pr)
    End If
    If pi <> 0 Then
        If Abs(pi) = 1 Then
            If pi = 1 Then r = r + " + " Else r = r + " - "
        Else
            If pi > 0 Then
                If pr <> 0 Then r = r + " + "
                r = r + f2(pi)
            Else
                r = r + " - " + f1(Mid$(Str$(pi), 2))
            End If
        End If
        r = r + " i"
    End If
    If r = "" Then r = "0"
    FormatoNumeroComplejo = r
End Function
'-----
Public Function f1(ByVal x As String) As String
    If Abs(Val(x)) >= 1 Then
        f1 = x
    Else
        If Left$(x, 1) = "." Then f1 = "0" + x Else f1 = x
    End If
End Function
'-----
Public Function f2(ByVal x As Double) As String
    Dim xx As String
    xx = Str$(x)
    If Abs(x) >= 1 Then
        f2 = Str$(x)
    Else
        If Left$(xx, 1) = "-" Then
            If Left$(xx, 2) = "-." Then f2 = "-0" + Right$(xx, Len(xx) - 1) Else f2 = xx
        Else
            If x = 0 Then
                f2 = Str$(0)
            Else
                If Left$(xx, 2) = "." Then f2 = "0" + Right$(xx, Len(xx) - 1) Else f2 = xx
            End If
        End If
    End If
End Function

```



```

Else
  If z1(0) = -1 Then cm = Str$(-1) Else cm = Mid$(Str$(1), 2)
End If
End If
Else
  If gx <> 0 Then
    If z1(0) <> 0 Then cm = cm + "(" + f2(z1(0))
    If Abs(z2(0)) <> 1 Then
      If z1(0) <> 0 Then
        If z2(0) > 0 Then
          cm = cm + " " + f1(Mid$(Str$(z2(0)), 2)) + " e )"
        Else
          cm = cm + " - " + f1(Mid$(Str$(z2(0)), 2)) + " e )"
        End If
      Else
        cm = cm + f2(z2(0)) + " e"
      End If
    Else
      If z2(0) = 1 Then cm = cm + " + e )" Else cm = cm + " - e )"
    End If
  Else
    cm = cm + f2(z1(0))
    If Abs(z2(0)) <> 1 Then
      If z2(0) > 0 Then
        cm = cm + " " + f1(Mid$(Str$(z2(0)), 2)) + " e"
      Else
        cm = cm + " - " + f1(Mid$(Str$(z2(0)), 2)) + " e"
      End If
    Else
      If z2(0) = 1 Then cm = cm + " + e" Else cm = cm + " - e"
    End If
  End If
End If
If gx <> 0 Then
  If gx = 1 Then
    cm = cm + " X "
  Else
    cm = cm + " X^" + Mid$(Str$(gx), 2)
  End If
End If
Else
  If Abs(z2(i)) <> 0 Then
    If i < gx Then
      If z1(i) <> 0 Then
        cd = cd + "(" + f2((z1(i)))
        If Abs(z2(i)) <> 1 Then
          If z2(i) > 0 Then
            cd = cd + " " + f1(Mid$(Str$(z2(i)), 2)) + " e )"
          Else
            If z2(i) < 0 Then
              cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " e )"
            End If
          End If
        Else
          If z2(i) = 1 Then cd = cd + " + e )" Else cd = cd + " - e )"
        End If
      Else
        If Abs(z2(i)) <> 1 Then
          If z2(i) < 0 Then
            cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " e"
          Else
            If z2(i) > 0 Then
              cd = cd + " " + f1(Mid$(Str$(z2(i)), 2)) + " e"
            End If
          End If
        Else
          If z2(i) = 1 Then cd = cd + " + e" Else cd = cd + " - e"
        End If
      End If
    Else
      If z1(i) > 0 Then
        cd = cd + " " + f1(Mid$(Str$(z1(i)), 2))
      Else
        If z1(i) < 0 Then
          cd = cd + " - " + f1(Mid$(Str$(z1(i)), 2))
        End If
      End If
    End If
  End If

```



```

End If
If Abs(z2(i)) <> 1 Then
    If z2(i) > 0 Then
        cd = cd + " " + f1(Mid$(Str$(z2(i)), 2)) + " e"
    Else
        If z2(i) < 0 Then
            cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " e"
        End If
    End If
Else
    If z2(i) = 1 Then cd = cd + " + e" Else cd = cd + " - e"
End If
End If
Else
    If Abs(z1(i)) <> 0 Then
        If Abs(z1(i)) <> 1 Then
            If z1(i) < 0 Then
                cd = cd + " - " + f1(Mid$(Str$(z1(i)), 2))
            Else
                cd = cd + " " + f1(Mid$(Str$(z1(i)), 2))
            End If
        Else
            If z1(i) = 1 Then cd = cd + " + " Else cd = cd + " - "
        End If
    End If
End If
If gx > 1 Then
    If i < gx - 1 Then
        cd = cd + " X^" + Mid$(Str$(gx - i), 2)
    Else
        If i = gx - 1 Then
            cd = cd + " X "
        End If
    End If
End If
cm = cm + cd: cd = ""
End If
End If
Next i
FormatoPolinomioDual = cm
End Function
'-----
Public Function f1(ByVal x As String) As String
    If Abs(Val(x)) >= 1 Then
        f1 = x
    Else
        If Left$(x, 1) = "." Then
            f1 = "0" + x
        Else
            f1 = x
        End If
    End If
End Function
'-----
Public Function f2(ByVal x As Double) As String
    Dim xx As String
    xx = Str$(x)
    If Abs(x) >= 1 Then
        f2 = Str$(x)
    Else
        If Left$(xx, 1) = "-" Then
            If Left$(xx, 2) = "-." Then
                f2 = "-0" + Right$(xx, Len(xx) - 1)
            Else
                f2 = xx
            End If
        Else
            f2 = xx
        End If
    End If
End Function
'-----
Public Function FormatoNumeroDual(ByVal pr As Double, ByVal pi As Double) As String
    Dim r As String
    If pr <> 0 Then
        r = r + f2(pr)
    End If
    If pi <> 0 Then
        If Abs(pi) = 1 Then
            If pi = 1 Then r = r + " + " Else r = r + " - "

```

```

Else
  If pi > 0 Then
    If pr <> 0 Then
      r = r + " + "
    End If
    r = r + f2(pi)
  Else
    r = r + " - " + f1(Mid$(Str$(pi), 2))
  End If
End If
r = r + " e"
End If
If r = "" Then r = "0"
FormatoNumeroDual = r
End Function

```

Si en lugar de  $X - a$ , se desea dividir el polinomio  $f(X)$  con el polinomio  $bX - a$  ( $b \neq 0$ ), entonces de las igualdades:

$$f(X) = \left(X - \frac{a}{b}\right) \cdot q(X) + r$$

$$f(X) = (bX - a) \cdot q_1(X) + r_1$$

$$\left(X - \frac{a}{b}\right) \cdot q(X) + r = (bX - a) \cdot \frac{q(X)}{b} + r$$

, resulta que

$$q_1(X) = \frac{q(X)}{b} \text{ y } r_1 = r$$

**Ejemplo 4:** Para dividir el polinomio del ejemplo 1 con  $3X - 2$  se dividirá primero el polinomio  $f(X)$  con  $X - 2/3$ . Los cálculos se reflejan en la tabla siguiente:

	1	0	-2	3	-5
$\frac{2}{3}$		$\frac{2}{3}$	$\frac{4}{9}$	$-\frac{28}{27}$	$\frac{106}{81}$
	1	$\frac{2}{3}$	$-\frac{14}{9}$	$\frac{53}{27}$	$-\frac{299}{81}$

, de donde resulta que el cociente y el resto en la división con  $3X - 2$  son:

$$q(X) = \frac{1}{3}X^3 + \frac{2}{9}X^2 - \frac{14}{27}X + \frac{53}{81} \text{ y } r = -\frac{299}{81}$$

, respectivamente.

**Ejemplo 5:** Para dividir el polinomio del ejemplo 2 con  $(3 - i)X - (2 + 5i)$  se dividirá el polinomio primero entre  $X - \frac{2+5i}{3-i} = X - \left(\frac{1}{10} + \frac{17}{10}i\right)$ , según la tabla siguiente:

	$2+3i$	0	-2	$5+i$	$7-4i$
$\frac{1+17i}{10}$		$\frac{-49+37i}{10}$	$\frac{339-398i}{50}$	$-\frac{28}{27}$	$\frac{106}{81}$
	$2+3i$	$\frac{-49+37i}{10}$	$\frac{239-398i}{50}$	$\frac{53}{27}$	$-\frac{299}{81}$

$$q \searrow \frac{3+11i}{10} X^3 + \frac{92+31i}{10} X^2 + \frac{223-191i}{100} X + \frac{159+53i}{270} \text{ y } r = -\frac{299}{81}$$

Dados el polinomio  $P \searrow$  de grado  $n$  y el número real  $a$ , el esquema de Ruffini es también útil para hallar los coeficientes de un polinomio  $Q \searrow$  de grado  $n$ , tal que  $Q \searrow - a \searrow = P \searrow$ . En efecto, si

$$\begin{aligned} P \searrow &= P_1 \searrow \searrow - a \searrow + r_0 \\ P_1 \searrow &= P_2 \searrow \searrow - a \searrow + r_1 \\ &\dots\dots\dots \\ P_{n-1} \searrow &= P_n \searrow \searrow - a \searrow + r_n \end{aligned}$$

, donde  $\text{grado } P_i \searrow = n - i \quad (i = 1, \dots, n)$

Entonces

$$\begin{aligned} P \searrow &= P_1 \searrow \searrow - a \searrow + r_0 \\ &= P_2 \searrow \searrow - a \searrow + r_1 \searrow - a \searrow + r_0 \\ &= P_3 \searrow \searrow - a \searrow + r_2 \searrow - a \searrow + r_1 \searrow - a \searrow + r_0 \\ &\dots\dots\dots \\ &= r_n \searrow \searrow - a \searrow + r_{n-1} \searrow - a \searrow + \dots + r_1 \searrow - a \searrow + r_0 \end{aligned}$$

, donde  $r_0 = P \searrow, r_1 = P_1 \searrow, r_2 = P_2 \searrow, \dots, r_n = P_n \searrow$ .

Los cálculos anteriores se pueden codificar de la manera siguiente:

```
Public Function RuffiniB(ByRef p() As Double, ByVal a As Double) As String
    Dim i As Integer, j As Integer, g As Integer, r() As Double, q() As Double
    g = UBound(p())
    ReDim r(g), q(g)
    For i = 0 To g
        r(i, 0) = p(i)
    Next i
    For i = 1 To g
        r(0, i) = p(0)
    Next i
    For j = 1 To g-j
        For i = 1 To g - j + 1
            r(i, j) = a * r(i - 1, j) + r(i, j - 1)
        Next i
    Next j
    q(0) = p(0)
    For i = 1 To g
        q(i) = r(i, g - i + 1)
    Next i
    RuffiniB = FormatoPol(q())
End Function
```

Los coeficientes del polinomio buscado serán los números  $q \searrow (i = 0, \dots, g)$ , donde  $g$  es el grado del polinomio  $p()$ .

**Ejemplo 6:** Si  $P \searrow = 2X^3 - 7X^2 + 5X + 9$  y  $a = 3$  entonces según la función anterior,  $Q \searrow = 2X^3 + 11X^2 + 17X + 15$  y así,  $P \searrow = 2 \searrow - 3 \searrow + 11 \searrow - 3 \searrow + 17 \searrow - 3 \searrow + 15$ .

Los cálculos a mano se podrían organizar según la tabla siguiente:

	2,	-7	5	9
3	2	-1	2	15 = $r_0$
3	2	5	17 = $r_1$	
3	2	11 = $r_2$		

	$2 = r_3$			
--	-----------	--	--	--

**Ejemplo 7:** Si  $P(X) = 4 + 3iX^4 + (-7i)X^3 + 4 + 5iX^2 + (-i)X + 4 + 3i$  y  $a = 3 - i$ , entonces

$$Q(X) = 4 + 3iX^4 + (-7 + 21i)X^3 + (401 + 11i)X^2 + (426 - 167i)X + 309 - 282i$$

, y

$$P(X) = 4 + 3i(X - a)^4 + (-7 + 21i)(X - a)^3 + (401 + 11i)(X - a)^2 + (426 - 167i)(X - a) + 309 - 282i$$

El esquema de Ruffini para dividir un polinomio por el polinomio  $X - a$  se puede generalizar para el caso cuando se divide con un polinomio cuyo coeficiente director es 1. En efecto, si se quiere dividir el polinomio

$$A(X) = a_0X^n + a_1X^{n-1} + \dots + a_{n-1}X + a_n \quad (3)$$

, con el polinomio

$$P(X) = X^m + p_1X^{m-1} + \dots + p_{m-1}X + p_m \quad (m \leq n) \quad (4)$$

, sean

$$Q(X) = q_0X^{n-m} + q_1X^{n-m-1} + \dots + q_{n-m-1}X + q_{n-m} \quad (5)$$

$$R(X) = r_0X^{n-m-1} + r_1X^{n-m-2} + \dots + r_{n-m-2}X + r_{n-m-1} \quad (6)$$

, el cociente y el resto de la división, respectivamente. Entonces se cumple la igualdad

$$A(X) = P(X)Q(X) + R(X) \quad (7)$$

A continuación se efectuará el desarrollo de la expresión  $P(X)Q(X) + R(X)$  en los dos casos siguientes:

1) Si  $n = 8$  y  $m = 3$ , el cálculo de la expresión mencionada se puede organizar según la tabla siguiente:

$X^8$	$X^7$	$X^6$	$X^5$	$X^4$	$X^3$	$X^2$	$X^1$	$X^0$
$q_0$	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$			
	$p_1q_0$	$p_1q_1$	$p_1q_2$	$p_1q_3$	$p_1q_4$	$p_1q_5$		
		$p_2q_0$	$p_2q_1$	$p_2q_2$	$p_2q_3$	$p_2q_4$	$p_2q_5$	
			$p_3q_0$	$p_3q_1$	$p_3q_2$	$p_3q_3$	$p_3q_4$	$p_3q_5$
						$r_0$	$r_1$	$r_2$

Así

$$P(X)Q(X) + R(X) = S(X) = s_0X^8 + s_1X^7 + s_2X^6 + s_3X^5 + s_4X^4 + s_5X^3 + s_6X^2 + s_7X + s_8$$

, donde

$$\begin{aligned} s_0 &= q_0 \\ s_1 &= q_1 + p_1q_0 \\ s_2 &= q_2 + p_1q_1 + p_2q_0 \\ s_3 &= q_3 + p_1q_2 + p_2q_1 + p_3q_0 \\ s_4 &= q_4 + p_1q_3 + p_2q_2 + p_3q_1 \end{aligned} \quad (8)$$

$$s_5 = a_5 + p_1q_4 + p_2q_3 + p_3q_2$$

$$s_6 = p_1q_5 + p_2q_4 + p_3q_3 + r_0$$

$$s_7 = p_2q_5 + p_3q_4 + r_1$$

$$s_8 = p_3q_5 + r_2$$

Identificando los polinomios  $A(X)$  y  $S(X)$  resulta que

$$q_0 = a_0$$

$$q_1 = a_1 - p_1q_0 = a_1 - \sum_{j+k=1} p_j \cdot q_k \quad (0 < j \leq m, \quad 0 \leq k < 1)$$

$$q_2 = a_2 - p_1q_1 - p_2q_0 = a_2 - \sum_{j+k=2} p_j \cdot q_k \quad (0 < j \leq m, \quad 0 \leq k < 2)$$

$$q_3 = a_3 - p_1q_2 - p_2q_1 - p_3q_0 = a_3 - \sum_{j+k=3} p_j \cdot q_k \quad (0 < j \leq m, \quad 0 \leq k < 3)$$

$$q_4 = a_4 - p_1q_3 - p_2q_2 - p_3q_1 = a_4 - \sum_{j+k=4} p_j \cdot q_k \quad (0 < j \leq m, \quad k < 4) \quad (9)$$

$$q_5 = a_5 - p_1q_4 - p_2q_3 - p_3q_2 \Leftrightarrow q_{n-n} = a_{n-m} - \sum_{j+k=n-m} p_j \cdot q_k \quad (0 < j \leq m, \quad k < 5)$$

$$r_0 = a_6 - p_1q_5 - p_2q_4 - p_3q_3 = a_{n-m+1} - \sum_{j+k=n-m+1} p_j \cdot q_k \quad (0 < j \leq m, \quad k \leq n-m)$$

$$r_1 = a_7 - p_2q_5 - p_3q_4 = a_{n-m+2} - \sum_{j+k=n-m+2} p_j \cdot q_k \quad (0 < j \leq m, \quad k \leq n-m)$$

$$r_2 = a_8 - p_3q_5 = a_{n-m+3} - \sum_{j+k=n-m+3} p_j \cdot q_k \quad (0 < j \leq m, \quad k \leq n-m)$$

2) Si  $n=8$  y  $m=6$ , el cálculo de la expresión  $P(X)Q(X)+R(X)$  se puede efectuar según la tabla siguiente:

$X^8$	$X^7$	$X^6$	$X^5$	$X^4$	$X^3$	$X^2$	$X^1$	$X^0$
$q_0$	$q_1$	$q_2$						
	$p_1q_0$	$p_1q_1$	$p_1q_2$					
		$p_2q_0$	$p_2q_1$	$p_2q_2$				
			$p_3q_0$	$p_3q_1$	$p_3q_2$			
				$p_4q_0$	$p_4q_1$	$p_4q_2$		
					$p_5q_0$	$p_5q_1$	$p_5q_2$	
						$p_6q_0$	$p_6q_1$	$p_6q_2$
			$r_0$	$r_1$	$r_2$	$r_3$	$r_4$	$r_5$

Así,

$$P(X)Q(X)+R(X)=S(X)=s_0X^8+s_1X^7+s_2X^6+s_3X^5+s_4X^4+s_5X^3+s_6X^2+s_7X+s_8$$

$$s_0 = q_0$$

$$s_1 = q_1 + p_1q_0$$

$$s_2 = q_2 + p_1q_1 + p_2q_0$$

$$s_3 = p_1q_2 + p_2q_1 + p_3q_0 + r_0$$

$$s_4 = p_2q_2 + p_2q_1 + p_3q_0 + r_1$$

$$s_5 = p_3q_2 + p_4q_1 + p_5q_0 + r_2$$

(8')

$$\begin{aligned}
s_6 &= p_4 q_2 + p_5 q_1 + p_6 q_0 + r_3 \\
s_7 &= p_5 q_2 + p_6 q_1 + r_4 \\
s_8 &= p_6 q_2 + r_5
\end{aligned}$$

Luego, Identificando los polinomios  $A_{\leq n-m}$  y  $S_{\leq n-m}$  resulta que

$$\begin{aligned}
q_0 &= a_0 \\
q_1 &= a_1 - p_1 q_0 = a_1 - \sum_{j+k=1} p_j \cdot q_k \quad (0 < j \leq m, \quad 0 \leq k < 1) \\
q_2 &= a_2 - p_1 q_1 - p_2 q_0 = a_2 - \sum_{j+k=2} p_j \cdot q_k \quad (0 < j \leq 3, \quad 0 \leq k < 2) \\
r_0 &= a_3 - p_1 q_2 - p_2 q_1 - p_3 q_0 = a_{n-m+1} - \sum_{j+k=n-m+1} p_j q_k \quad (0 < j \leq m, \quad 0 \leq k \leq n-m) \\
r_1 &= a_4 - p_2 q_2 - p_2 q_1 - p_3 q_0 = a_{n-m+2} - \sum_{j+k=n-m+2} p_j \cdot q_k \quad (0 < j \leq m, \quad 0 \leq k \leq n-m) \\
r_2 &= a_5 - p_3 q_2 - p_4 q_1 - p_5 q_0 = a_{n-m+3} - \sum_{j+k=n-m+3} p_j \cdot q_k \quad (0 < j \leq m, \quad 0 \leq k \leq n-m) \\
r_3 &= a_6 - p_4 q_2 - p_5 q_1 - p_6 q_0 = a_{n-m+4} - \sum_{j+k=n-m+4} p_j \cdot q_k \quad (0 < j \leq m, \quad 0 \leq k \leq n-m) \\
r_4 &= a_7 - p_5 q_2 - p_6 q_1 = a_{n-m+5} - \sum_{j+k=n-m+5} p_j \cdot q_k \quad (0 < j \leq m, \quad 0 \leq k \leq n-m) \\
r_5 &= a_8 - p_6 q_2 = a_{n-m+6} - \sum_{j+k=n-m+6} p_j \cdot q_k \quad (0 < j \leq m, \quad 0 \leq k \leq n-m)
\end{aligned} \tag{9'}$$

Examinando los dos casos anteriores se puede enunciar el teorema siguiente:

**Teorema 2:** Dados los polinomios  $A_{\leq n-m}$  y  $P_{\leq m}$ , las fórmulas que determinan los polinomios  $Q_{\leq n-m}$  y  $R_{\leq n-m}$  y que verifican en la igualdad (7) son las siguientes:

$$q_0 = a_0 \quad y \quad q_i = a_i - \sum_{j+k=i} p_j \cdot q_k, \quad 0 < i \leq n-m, \quad 0 < j \leq m, \quad 0 \leq k < i \tag{10}$$

$$r_i = a_{n-m+i+1} - \sum_{j+k=n-m+i+1} p_j \cdot q_k, \quad 0 \leq i \leq m-1, \quad 0 < j \leq m, \quad 0 \leq k \leq n-m \tag{11}$$

Por supuesto, las fórmulas (10) y (11) no son tan manejables como las fórmulas en el caso de la división entre  $X - a$ , pero son fácilmente programables y el ordenador las ejecuta con tanta facilidad como las personas la regla normal de Ruffini.

Esta manera de efectuar la división tiene sus ventajas, sobre todo cuando los coeficientes de los polinomios son enteros o decimales (trabajar con fracciones es más penoso). Si el coeficiente director  $p_0$  del polinomio  $P_{\leq m}$  no fuera 1 entonces habrá que dividir  $P_{\leq m}$  entre  $p_0$  y hacer la división con el polinomio así obtenido. El resto será válido y para obtener el verdadero cociente hay que dividir el cociente obtenido entre  $p_0$ .

**Ejemplo 3:** De acuerdo con lo dicho anteriormente, para hallar el cociente y el resto en la división euclídea del polinomio  $A_{\leq 8}$  entre  $P_{\leq 4}$ , donde

$$\begin{aligned}
A_{\leq 8} &= 2X^8 - 3X^7 + 5X^6 - X^5 + 3X^4 + 4X^3 - 7X^2 + 3X - 4 \\
P_{\leq 4} &= X^4 - 2X^3 + 5X^2 + 3X - 4
\end{aligned}$$

, se obtiene el cociente y el resto siguientes:

$$\begin{aligned}
Q_{\leq 4} &= 2X^4 + X^3 - 3X^2 - 18X - 13 \\
R_{\leq 4} &= 81X^3 + 100X^2 - 30X - 56
\end{aligned}$$

, respectivamente.

Para dividir el polinomio  $A(x)$  entre el polinomio  $P(x)$ , cuyo coeficiente director  $b_0 \neq 1$ , se dividirá el polinomio  $A(x)$  entre el polinomio

$$P_1(x) = \frac{1}{p_0} P(x)$$

, según el esquema general de Ruffini. Suponiendo que,

$$A(x) = P(x)Q(x) + R(x)$$

, de

$$A(x) = \frac{1}{p_0} P(x)Q_1(x) + R_1(x)$$

, resulta que

$$Q(x) = \frac{1}{p_0} Q_1(x) \quad y \quad R(x) = R_1(x)$$

En general, los coeficientes del cociente y del resto serán aproximaciones decimales de los coeficientes reales.

**Ejemplo 4:** Para dividir el polinomio

$$A(x) = 3x^5 - 7x^3 + 4x^2 - 5x + 12$$

, con el polinomio

$$P(x) = 2x^3 - 5x^2 + 7x - 8$$

, se divide primero el polinomio  $A(x)$  entre el polinomio

$$P_1(x) = x^3 - 2.5x^2 + 3.5x - 4$$

, y se obtiene el cociente y el resto

$$Q_1(x) = 3x^2 + 7.5x - 4 \quad y \quad R_1(x) = -7.125x^2 + 20.625x + 17$$

, respectivamente. Entonces el cociente y el resto de la división inicial serán:

$$Q(x) = \frac{1}{2} Q_1(x) = 1.5x^2 + 3.75x - 2 \quad y \quad R(x) = R_1(x)$$

, respectivamente.

**Ejemplo 5:** Para dividir el polinomio

$$A(x) = 3x^5 - 7x^3 + 4x^2 - 5x + 12$$

, entre el polinomio

$$P(x) = 7x^3 - 5x^2 + 7x - 8$$

, hay que dividir el polinomio  $P(x)$  entre 7 y así los coeficientes del polinomio  $P_1(x)$  ya no serán números decimales sino números reales periódicos.

Por tanto, si no se quiere trabajar con fracciones, tendremos que hacer la división de  $A(x)$  con un polinomio  $P_2(x)$  que, al trabajar con bastantes decimales después de la coma, será una buena aproximación del polinomio  $P_1(x)$ . Así el resultado final será también una aproximación del cociente verdadero. Tomando

$$P_2(x) = x^3 - 0.714286x^2 + x - 1.142857$$

, se obtendrá el resultado aproximado siguiente;

$$Q(x) \approx 0.428572x^2 + 0.306123x - 1.209912$$

$$R(x) \approx -0.763851x^2 + 5.918367x + 2.320702$$

Para efectuar la división de esta manera, se pueden utilizar las funciones siguientes:

```
Public Function RuffiniG(ByRef a() As Double, ByRef p() As Double) As Variant
    Dim i As Integer, j As Integer, k As Integer, r() As Double, q() As Double
    Dim r1() As Double, p1() As Double, gr As Integer, sw As Integer
    Dim j1 As Integer, ga As Integer, gp As Integer, gb As Integer
    Dim cxq As String, cxr As String, res(2) As String
    ga = UBound(a()): gp = UBound(p()): gb = ga - gp
```

```

ReDim b(gb), p1(gp)
If p(0) <> 1 Then
    For i = 0 To gp: p1(i) = p(i) / p(0): Next i
Else
    For i = 0 To gp: p1(i) = p(i): Next i
End If
' Cálculo de los coeficientes del cociente.
b(0) = a(0)
For i = 1 To gb
    b(i) = a(i)
    For j = 0 To gp
        For k = 0 To i - 1
            If j + k = i Then
                b(i) = b(i) - p1(j) * b(k)
            End If
        Next k
    Next j
Next i
ReDim q(gb)
If p(0) <> 1 Then
    For i = 0 To gb: q(i) = b(i) / p(0): Next i
Else
    For i = 0 To gb: q(i) = b(i): Next i
End If
cxq = VerPol(q())
' Cálculo de los coeficientes del resto.
ReDim r(gp - 1)
For i = 0 To gp - 1
    r(i) = a(gb + i + 1)
    For j = 1 To gp
        For k = 0 To gb
            If j + k = gb + i + 1 Then
                r(i) = r(i) - p1(j) * b(k)
            End If
        Next k
    Next j
Next i
Next i
gr = UBound(r())
For i = 0 To gr
    If Abs(r(i)) > 0.0000000000001 Then
        sw = 1
    End If
Next i
If sw = 1 Then cxr = VerPol(r()) Else cxr = "0"
res(1) = cxq: res(2) = cxr
RuffiniG = res()
End Function
'-----
Public Function VerPol(ByRef xx() As Double) As String
    Dim i As Integer, gx As Integer, pol As String, x() As Double
    x() = xx(): gx = UBound(x())
    If gx <> 0 Then
        If x(0) <> 0 Then
            If Abs(x(0) - 1) < 10 ^ (-15) Then x(0) = 1
            If Abs(x(0) + 1) < 10 ^ (-15) Then x(0) = -1
            If Abs(x(0)) = 1 Then
                If x(0) = 1 Then
                    If gx > 1 Then
                        pol = pol + "X^" + Str$(gx)
                    Else
                        pol = pol + "X"
                    End If
                End If
            End If
            If x(0) = -1 Then
                If gx > 1 Then
                    pol = pol + "- X^" + Str$(gx)
                Else
                    pol = pol + "- X"
                End If
            End If
        Else
            If Abs(x(0)) > 1 Then
                pol = pol + Str$(x(0))
            Else
                If x(0) > 0 Then
                    pol = pol + "0" + Mid$(Str$(Abs(x(0))), 2)
                End If
            End If
        End If
    End If

```



```

Else
    pol = pol + "- 0" + Mid$(Str$(Abs(x(0))), 2)
End If
End If
If gx > 1 Then
    pol = pol + " X^" + Str$(gx)
Else
    pol = pol + " X"
End If
End If
End If
For i = 1 To gx
    If x(i) <> 0 Then
        If Abs(x(i) - 1) < 10 ^ (-15) Then x(1) = 1
        If Abs(x(i) + 1) < 10 ^ (-15) Then x(i) = -1
        If x(i) < 0 Then
            pol = pol + " - "
        Else
            pol = pol + " + "
        End If
        If Abs(x(i)) <> 1 Then
            If Abs(x(i)) > 1 Then
                pol = pol + Mid$(Str$(Abs(x(i))), 2)
            Else
                pol = pol + "0" + Mid$(Str$(Abs(x(i))), 2)
            End If
        Else
            If i = gx Then
                pol = pol + Mid$(Str$(1), 2)
            End If
        End If
        If i < gx Then
            pol = pol + " X"
        End If
        If i < gx - 1 Then
            pol = pol + "^" + Str$(gx - i)
        End If
    End If
Next i
Else
    If x(0) > 0 Then
        If x(0) < 1 Then
            pol = pol + "0" + Mid$(Str$(x(0)), 2)
        Else
            pol = pol + Mid$(Str$(x(0)), 2)
        End If
    Else
        If x(0) > -1 Then
            pol = pol + "- 0" + Mid$(Str$(Abs(x(0))), 2)
        Else
            pol = pol + "- " + Mid$(Str$(Abs(x(0))), 2)
        End If
    End If
End If
VerPol = pol
End Function

```

**Ejemplo 6:** Según el código anterior, si

$A \curvearrowright = 4X^6 - 7X^5 + 8X^4 + 3X^3 - 12X^2 + 5X + 4$  y  $P \curvearrowright = X^5 - 2X^4 + 3X^3 + 5X^2 - X + 4$ , entonces  $Q \curvearrowright = 4X + 1$  y  $R \curvearrowright = -2X^4 - 20X^3 - 13X^2 - 10X$ .

**Ejemplo 7:** Si  $A \curvearrowright = 4X^6 - 7X^5 + 8X^4 + 3X^3 - 12X^2 + 5X + 4$  y  $P \curvearrowright = 5X^3 + 4X^2 - 3X + 7$ , entonces  $Q \curvearrowright = 0.8X^3 - 2.04X^2 + 3.712X - 4.7136$  y  $R \curvearrowright = 32.2704X^2 - 35.1248X + 36.9952$

**Ejemplo 8:** si  $A \curvearrowright = 7X^4 - 4X^3 + 15X^2 - 3X + 7$  y  $P \curvearrowright = 3X^2 - 11X + 5$ , entonces el ordenador devuelve el resultado siguiente:

$$Q \curvearrowright = 2.3333333333333333X^2 + 7.222222222222222X + 27.5925925925926X^2$$

$$R \curvearrowright = 264.407407407407X - 130.962962963$$

En este caso sencillo, no es difícil ver que

$$Q \left( \frac{7}{3} X^2 + \frac{65}{9} X + \frac{745}{27} \right) y \quad R \left( \frac{7139}{27} X - \frac{3536}{27} \right)$$

Cuando se trata de polinomios con coeficientes complejos en vez de la función *RuffiniGhay* que utilizar la función siguiente, donde  $a_1()$  y  $a_2()$  (  $p01()$  y  $p02()$  ) son las matrices unidimensionales que contienen las partes reales e imaginarias de los coeficientes del dividiendo (divisor), respectivamente.

```
Public Function RuffiniGC(ByRef a1() As Double, ByRef a2() As Double, ByRef p01() As Double, p02() As Double) As Variant
    Dim g1 As Integer, g2 As Integer, gq As Integer, r() As Double, r1() As Double
    Dim i As Integer, j As Integer, j1 As Integer, res(2) As String, r2() As Double
    Dim b() As Double, x(2) As Double, y(2) As Double, p1() As Double, p2() As Double.
    Dim q1() As Double, q2() As Double, b1() As Double, b2() As Double, rr() As Double
    g1 = UBound(a1()): g2 = UBound(p01()): gq = g1 - g2
    ReDim b(gq, 2), b1(gq), b2(gq), r1(g2 - 1), r2(g - 1)
    'División del divisor con su coeficiente director
    If p01(0) = 1 And p02(0) = 0 Then
        p1() = p01(): p2() = p02()
    Else
        ReDim p1(g1), p2(g1)
        p1(0) = 1: p2(0) = 0
        y(1) = p01(0): y(2) = p02(0)
        For k = 1 To g2
            x(1) = p01(k): x(2) = p02(k)
            rr() = DivNC(x(), y())
            p1(k) = rr(1): p2(k) = rr(2)
        Next k
    End If
    'Calculo del cociente
    b(0, 1) = a1(0): b(0, 2) = a2(0)
    For i = 1 To gq
        b(i, 1) = a1(i): b(i, 2) = a2(i)
        For j = 0 To g2
            For k = 0 To i - 1
                If j + k = i Then
                    x(1) = p1(j): x(2) = p2(j): y(1) = b(k, 1): y(2) = b(k, 2)
                    rr() = ProdNC(x(), y())
                    b(i, 1) = b(i, 1) - rr(1): b(i, 2) = b(i, 2) - rr(2)
                End If
            Next k
        Next j
    Next i
    Next i
    For i = 0 To gq
        b1(i) = b(i, 1): b2(i) = b(i, 2)
    Next i
    If p01(0) <> 1 Or p02(0) <> 0 Then
        ReDim q1(gq), q2(gq)
        For i = 0 To gq
            x(1) = b1(i): x(2) = b2(i): y(1) = p01(0): y(2) = p02(0)
            rr() = DivNC(x(), y())
            q1(i) = rr(1): q2(i) = rr(2)
        Next i
    Else
        q1() = b1(): q2() = b2()
    End If
    'Cálculo del resto
    ReDim r(g2 - 1, 2): k = 0
    For i = 0 To g2 - 1
        r(k, 1) = a1(gq + i + 1): r(k, 2) = a2(gq + i + 1)
        j = k: j1 = 0
        Do
            x(1) = p1(j + 1): x(2) = p2(j + 1): y(1) = b1(gq - j1): y(2) = b2(gq - j1)
            rr() = ProdNC(x(), y())
            r(k, 1) = r(k, 1) - rr(1): r(k, 2) = r(k, 2) - rr(2)
            j = j + 1
            j1 = j1 + 1
        Loop While j + 1 <= g2 And gq - j1 >= 0
        k = k + 1
    Next i
    For i = 0 To g2 - 1
        r1(i) = r(i, 1): r2(i) = r(i, 2)
    Next i
    'El cociente en la pantalla
```

```

res(1) = FormatoPolinomioComplejo(q1(), q2())
' El resto en la pantalla
res(2) = FormatoPolinomioComplejo(r1(), r2())
RuffiniGC = res()
End Function
'-----
Public Function ProdNC(ByRef x() As Double, ByRef y() As Double) As Variant
    Dim pr(2) As Double
    pr(1) = x(1) * y(1) - x(2) * y(2)
    pr(2) = x(1) * y(2) + x(2) * y(1)
    ProdNC = pr()
End Function
'-----
Public Function DivNC(ByRef u() As Double, ByRef v() As Double) As Variant
    Dim cmv As Double, co() As Double, x(2) As Double, y(2) As Double, rr() As Double
    ReDim co(2)
    cmv = v(1) * v(1) + v(2) * v(2)
    x(1) = u(1): x(2) = u(2): y(1) = v(1): y(2) = -v(2)
    rr() = ProdNC(x(), y())
    co(1) = rr(1) / cmv: co(2) = rr(2) / cmv
    DivNC = co()
End Function
'-----
Public Function FPolC(ByRef z1() As Double, ByRef z2() As Double) As String
    Dim i As Integer, j As Integer, gx As Integer, pr As Double
    Dim cd As String, cm As String
    gx = UBound(z1()): pr = 0.0000000000000001
    For i = 0 To gx
        If Abs(z1(i) - 1) < pr Then z1(i) = 1
        If Abs(z1(i) + 1) < pr Then z1(i) = -1
        If Abs(z2(i) - 1) < pr Then z2(i) = 1
        If Abs(z2(i) + 1) < pr Then z2(i) = -1
        If Abs(z1(i)) < pr Then z1(i) = 0
        If Abs(z2(i)) < pr Then z2(i) = 0
        If z1(i) <> 0 Or z2(i) <> 0 Then
            If i = 0 Then
                If z2(0) = 0 Then
                    If Abs(z1(0)) <> 1 Then
                        cm = f2(z1(0))
                    Else
                        If gx <> 0 Then
                            If z1(0) = -1 Then
                                cm = "-"
                            End If
                        Else
                            If z1(0) = -1 Then
                                cm = Str$(-1)
                            Else
                                cm = Mid$(Str$(1), 2)
                            End If
                        End If
                    End If
                Else
                    If gx <> 0 Then
                        If z1(0) <> 0 Then
                            cm = cm + "(" + f2(z1(0))
                        End If
                    If Abs(z2(0)) <> 1 Then
                        If z1(0) <> 0 Then
                            If z2(0) > 0 Then
                                cm = cm + " + " + f1(Mid$(Str$(z2(0)), 2)) + " i )"
                            Else
                                cm = cm + " - " + f1(Mid$(Str$(z2(0)), 2)) + " i )"
                            End If
                        Else
                            cm = cm + f2(z2(0)) + " i "
                        End If
                    Else
                        If z2(0) = 1 Then
                            cm = cm + " + i )"
                        Else
                            cm = cm + " - i )"
                        End If
                    End If
                Else
                    cm = cm + f2(z1(0))
                End If
            End If
        End If
    Next i

```

```

If Abs(z2(0)) <> 1 Then
  If z2(0) > 0 Then
    cm = cm + " + " + f1(Mid$(Str$(z2(0)), 2)) + " i"
  Else
    cm = cm + " - " + f1(Mid$(Str$(z2(0)), 2)) + " i"
  End If
Else
  If z2(0) = 1 Then
    cm = cm + "+ i"
  Else
    cm = cm + "- i"
  End If
End If
End If
End If
If gx <> 0 Then
  If gx = 1 Then
    cm = cm + " X "
  Else
    cm = cm + " X^" + Mid$(Str$(gx), 2)
  End If
End If
Else
  If Abs(z2(i)) <> 0 Then
    If i < gx Then
      If z1(i) <> 0 Then
        cd = cd + " + (" + f2((z1(i)))
        If Abs(z2(i)) <> 1 Then
          If z2(i) > 0 Then
            cd = cd + " + " + f1(Mid$(Str$(z2(i)), 2)) + " i)"
          Else
            If z2(i) < 0 Then
              cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " i)"
            End If
          End If
        Else
          If z2(i) = 1 Then
            cd = cd + " + i)"
          Else
            cd = cd + " - i)"
          End If
        End If
      Else
        If Abs(z2(i)) <> 1 Then
          If z2(i) < 0 Then
            cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " i"
          Else
            If z2(i) > 0 Then
              cd = cd + " + " + f1(Mid$(Str$(z2(i)), 2)) + " i"
            End If
          End If
        Else
          If z2(i) = 1 Then
            cd = cd + " + i"
          Else
            cd = cd + " - i"
          End If
        End If
      End If
    Else
      If z1(i) > 0 Then
        cd = cd + " + " + f1(Mid$(Str$(z1(i)), 2))
      Else
        If z1(i) < 0 Then
          cd = cd + " - " + f1(Mid$(Str$(z1(i)), 2))
        End If
      End If
    End If
  If Abs(z2(i)) <> 1 Then
    If z2(i) > 0 Then
      cd = cd + " + " + f1(Mid$(Str$(z2(i)), 2)) + " i"
    Else
      If z2(i) < 0 Then
        cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " i"
      End If
    End If
  End If
Else
  If z1(i) > 0 Then
    cd = cd + " + " + f1(Mid$(Str$(z1(i)), 2))
  Else
    If z1(i) < 0 Then
      cd = cd + " - " + f1(Mid$(Str$(z1(i)), 2))
    End If
  End If
End If

```

```

        If z2(i) = 1 Then
            cd = cd + " + i"
        Else
            cd = cd + " - i"
        End If
    End If
End If
Else
    If Abs(z1(i)) <> 0 Then
        If Abs(z1(i)) <> 1 Then
            If z1(i) < 0 Then
                cd = cd + " - " + f1(Mid$(Str$(z1(i)), 2))
            Else
                cd = cd + " + " + f1(Mid$(Str$(z1(i)), 2))
            End If
        Else
            If z1(i) = 1 Then
                cd = cd + " + "
            Else
                cd = cd + " - "
            End If
        End If
    End If
End If
If gx > 1 Then
    If i < gx - 1 Then
        cd = cd + " X^" + Mid$(Str$(gx - i), 2)
    Else
        If i = gx - 1 Then
            cd = cd + " X "
        End If
    End If
End If
cm = cm + cd: cd = ""
End If
Next i
FPolC = cm
End Function
'-----
Public Function f1(ByVal x As String) As String
    If Abs(Val(x)) >= 1 Then
        f1 = x
    Else
        If Left$(x, 1) = "." Then
            f1 = "0" + x
        Else
            f1 = x
        End If
    End If
End Function
'-----
Public Function f2(ByVal x As Double) As String
    Dim xx As String
    xx = Str$(x)
    If Abs(x) >= 1 Then
        f2 = Str$(x)
    Else
        If Left$(xx, 1) = "-" Then
            If Left$(xx, 2) = "-." Then
                f2 = "-0" + Right$(xx, Len(xx) - 1)
            Else
                f2 = xx
            End If
        Else
            If x = 0 Then
                f2 = Str$(0)
            Else
                If Left$(xx, 2) = "." Then
                    f2 = "0" + Right$(xx, Len(xx) - 1)
                Else
                    f2 = xx
                End If
            End If
        End If
    End If
End Function

```

End Function

### Ejemplo 9: Si

$$A(X) = 1 + 3iX^5 + (-1+i)X^4 + (1+11i)X^3 + (2-5i)X^2 + (-13i)X + 15 - 4i, \text{ y}$$

$$P(X) = X^3 + (-4i)X^2 + (-3+7i)X - 1 - 9i$$

, entonces según el código anterior,

$$Q(X) = 1 + 3iX^2 + (-17+3i)X + 56 - 68i \text{ y } R(X) = (-17+504i)X^2 + (-346-759i)X + 683 + 432i$$

### Ejemplo 10: Si

$$A(X) = 1 + 3iX^5 + (-1+i)X^4 + (1+11i)X^3 + (2-5i)X^2 + (-13i)X + 15 - 4i, \text{ y}$$

$$B(X) = iX^3 + (-3i)X^2 + 6X + 13 - 4i$$

, entonces

$$Q(X) = (-2i)X^2 + (-4+16i)X - 17 + 227i$$

$$R(X) = (-725-1703i)X^2 + (-268-1487i)X - 672 - 3023i$$

### Ejemplo 11: Si

$$A(X) = 1 - 5iX^4 + (-i)X^3 + (-7i)X^2 + (-2i)X + 3 - 8i, \text{ y } P(X) = 2iX^2 + (-6i)X + 7 - i$$

, entonces

$$Q(X) = (-2.5-i)X^2 + (-5.5-10.75i)X + 16.125 - 56.25i$$

$$R(X) = (-10.125+445.75i)X - 53.625 + 401.875i$$

La regla general de Ruffini se puede aplicar también a los polinomios con coeficientes duales excepto en el caso cuando el coeficiente director del divisor es un divisor de cero (dual puro). En este caso el código es la siguiente:

```
Public Function RuffiniGD(ByRef a1() As Double, ByRef a2() As Double, ByRef p01() As Double, p02() As Double) As Variant
    Dim g1 As Integer, g2 As Integer, gq As Integer, r() As Double, r1() As Double
    Dim i As Integer, j As Integer, j1 As Integer, res(2) As String, r2() As Double
    Dim b() As Double, x(2) As Double, y(2) As Double, p1() As Double, p2() As Double
    Dim q1() As Double, q2() As Double, b1() As Double, b2() As Double, rr() As Double
    g1 = UBound(a1()): g2 = UBound(a2()): gq = g1 - g2
    ReDim b(gq, 2), b1(gq), b2(gq), r1(g2 - 1), r2(g - 1)
    If p01(1) = 0 Then
        MsgBox "¡La división es imposible!"
        Exit Function
    End If
    'División del divisor con su coeficiente director
    If p01(0) = 1 And p02(0) = 0 Then
        p1() = p01(): p2() = p02()
    Else
        ReDim p1(g1), p2(g1)
        p1(0) = 1: p2(0) = 0
        y(1) = p01(0): y(2) = p02(0)
        For k = 1 To g2
            x(1) = p01(k): x(2) = p02(k)
            rr() = DivND(x(), y()) 'Cuando es posible
            p1(k) = rr(1): p2(k) = rr(2)
        Next k
    End If
    'Calculo del cociente
    b(0, 1) = a1(0): b(0, 2) = a2(0)
    For i = 1 To gq
        b(i, 1) = a1(i): b(i, 2) = a2(i)
        For j = 0 To g2
            For k = 0 To i - 1
                If j + k = i Then
                    x(1) = p1(j): x(2) = p2(j): y(1) = b(k, 1): y(2) = b(k, 2)
                    rr() = ProdND(x(), y())
                    b(i, 1) = b(i, 1) - rr(1): b(i, 2) = b(i, 2) - rr(2)
                End If
            Next k
        Next j
    Next i
    Next i
```

```

For i = 0 To gq
    b1(i) = b(i, 1): b2(i) = b(i, 2)
Next i
If p01(0) <> 1 Or p02(0) <> 0 Then
    ReDim q1(gq), q2(gq)
    For i = 0 To gq
        x(1) = b1(i): x(2) = b2(i): y(1) = p01(0): y(2) = p02(0)
        rr() = DivND(x(), y()) ' Cuando es posible
        q1(i) = rr(1): q2(i) = rr(2)
    Next i
Else
    q1() = b1(): q2() = b2()
End If
' Cálculo del resto
ReDim r(g2 - 1, 2): k = 0
For i = 0 To g2 - 1
    r(k, 1) = a1(gq + i + 1): r(k, 2) = a2(gq + i + 1)
    j = k: j1 = 0
    Do
        x(1) = p1(j + 1): x(2) = p2(j + 1): y(1) = b1(gq - j1): y(2) = b2(gq - j1)
        rr() = ProdND(x(), y())
        r(k, 1) = r(k, 1) - rr(1): r(k, 2) = r(k, 2) - rr(2)
        j = j + 1
        j1 = j1 + 1
    Loop While j + 1 <= g2 And gq - j1 >= 0
    k = k + 1
Next i
For i = 0 To g2 - 1
    r1(i) = r(i, 1): r2(i) = r(i, 2)
Next i
' El ociente en la pantalla
res(1) = FormatoPolinomioDual(q1(), q2())
' El resto en la pantalla
res(2) = FormatoPolinomioDual(r1(), r2())
RuffiniGD = res()
End Function
'-----
Public Function DivND(ByRef u() As Double, ByRef v() As Double) As Variant
    Dim cmv As Double, co(2) As Double, x(2) As Double, y(2) As Double, rr() As Double
    If v(1) = 0 Then
        MsgBox "¡La división es imposible!"
        End
    End If
    cmv = v(1) * v(1)
    x(1) = u(1): x(2) = u(2): y(1) = v(1): y(2) = -v(2)
    rr() = ProdND(x(), y())
    co(1) = rr(1) / cmv: co(2) = rr(2) / cmv
    DivND = co()
End Function
'-----
Public Function FPolD(ByRef z1() As Double, ByRef z2() As Double) As String
    Dim i As Integer, j As Integer, gx As Integer, pr As Double
    Dim cd As String, cm As String
    gx = UBound(z1()): pr = 0.0000000000000001
    For i = 0 To gx
        If Abs(z1(i) - 1) < pr Then z1(i) = 1
        If Abs(z1(i) + 1) < pr Then z1(i) = -1
        If Abs(z2(i) - 1) < pr Then z2(i) = 1
        If Abs(z2(i) + 1) < pr Then z2(i) = -1
        If Abs(z1(i)) < pr Then z1(i) = 0
        If Abs(z2(i)) < pr Then z2(i) = 0
        If z1(i) <> 0 Or z2(i) <> 0 Then
            If i = 0 Then
                If z2(0) = 0 Then
                    If Abs(z1(0)) <> 1 Then
                        cm = f2(z1(0))
                    Else
                        If gx <> 0 Then
                            If z1(0) = -1 Then
                                cm = "-"
                            End If
                        Else
                            If z1(0) = -1 Then
                                cm = Str$(-1)
                            Else
                                cm = Mid$(Str$(1), 2)
                            End If
                        End If
                    End If
                Else
                    If Abs(z1(0)) <> 1 Then
                        cm = f2(z1(0))
                    Else
                        If gx <> 0 Then
                            If z1(0) = -1 Then
                                cm = Str$(-1)
                            Else
                                cm = Mid$(Str$(1), 2)
                            End If
                        Else
                            If z1(0) = -1 Then
                                cm = Str$(-1)
                            Else
                                cm = Mid$(Str$(1), 2)
                            End If
                        End If
                    End If
                End If
            End If
        End If
    Next i
    FPolD = cm
End Function

```

```

    End If
  End If
End If
Else
  If gx <> 0 Then
    If z1(0) <> 0 Then
      cm = cm + "(" + f2(z1(0))
    End If
    If Abs(z2(0)) <> 1 Then
      If z1(0) <> 0 Then
        If z2(0) > 0 Then
          cm = cm + " + " + f1(Mid$(Str$(z2(0)), 2)) + " e )"
        Else
          cm = cm + " - " + f1(Mid$(Str$(z2(0)), 2)) + " e )"
        End If
      Else
        cm = cm + f2(z2(0)) + " e"
      End If
    Else
      If z2(0) = 1 Then
        cm = cm + " + e )"
      Else
        cm = cm + " - e )"
      End If
    End If
  Else
    cm = cm + f2(z1(0))
    If Abs(z2(0)) <> 1 Then
      If z2(0) > 0 Then
        cm = cm + " + " + f1(Mid$(Str$(z2(0)), 2)) + " e"
      Else
        cm = cm + " - " + f1(Mid$(Str$(z2(0)), 2)) + " e"
      End If
    Else
      If z2(0) = 1 Then
        cm = cm + "+ e"
      Else
        cm = cm + "- e"
      End If
    End If
  End If
End If
If gx <> 0 Then
  If gx = 1 Then
    cm = cm + " X "
  Else
    cm = cm + " X^" + Mid$(Str$(gx), 2)
  End If
End If
Else
  If Abs(z2(i)) <> 0 Then
    If i < gx Then
      If z1(i) <> 0 Then
        cd = cd + "(" + f2((z1(i)))
      If Abs(z2(i)) <> 1 Then
        If z2(i) > 0 Then
          cd = cd + " + " + f1(Mid$(Str$(z2(i)), 2)) + " e )"
        Else
          If z2(i) < 0 Then
            cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " e )"
          End If
        End If
      Else
        If z2(i) = 1 Then
          cd = cd + " + e )"
        Else
          cd = cd + " - e )"
        End If
      End If
    Else
      If Abs(z2(i)) <> 1 Then
        If z2(i) < 0 Then
          cd = cd + " - " + f1(Mid$(Str$(z2(i)), 2)) + " e"
        Else
          If z2(i) > 0 Then
            cd = cd + " + " + f1(Mid$(Str$(z2(i)), 2)) + " e"
          End If
        End If
      End If
    End If
  End If

```



```

        End If
    End If
Else
    If z2(i) = 1 Then
        cd = cd + " + e"
    Else
        cd = cd + " - e"
    End If
End If
End If
Else
    If z1(i) > 0 Then
        cd = cd + " + " + fl(Mid$(Str$(z1(i)), 2))
    Else
        If z1(i) < 0 Then
            cd = cd + " - " + fl(Mid$(Str$(z1(i)), 2))
        End If
    End If
    If Abs(z2(i)) <> 1 Then
        If z2(i) > 0 Then
            cd = cd + " + " + fl(Mid$(Str$(z2(i)), 2)) + " e"
        Else
            If z2(i) < 0 Then
                cd = cd + " - " + fl(Mid$(Str$(z2(i)), 2)) + " e"
            End If
        End If
    Else
        If z2(i) = 1 Then
            cd = cd + " + e"
        Else
            cd = cd + " - e"
        End If
    End If
End If
Else
    If Abs(z1(i)) <> 0 Then
        If Abs(z1(i)) <> 1 Then
            If z1(i) < 0 Then
                cd = cd + " - " + fl(Mid$(Str$(z1(i)), 2))
            Else
                cd = cd + " + " + fl(Mid$(Str$(z1(i)), 2))
            End If
        Else
            If z1(i) = 1 Then
                cd = cd + " + "
            Else
                cd = cd + " - "
            End If
        End If
    End If
End If
If gx > 1 Then
    If i < gx - 1 Then
        cd = cd + " X^" + Mid$(Str$(gx - i), 2)
    Else
        If i = gx - 1 Then
            cd = cd + " X "
        End If
    End If
End If
cm = cm + cd: cd = ""
End If
Next i
FPolD = cm
End Function

```

### Ejemplo12 : Si

$$A(X) = 4 + 7\varepsilon X^5 + 3 + 4\varepsilon X^3 + 1 + 5\varepsilon X^2 + 3\varepsilon X - 5 + 7\varepsilon \quad y \quad P(X) = X^2 + 9\varepsilon X - 3 + 4\varepsilon$$

, entonces

$$Q(X) = 4 + 7\varepsilon X^3 + 8 - 50\varepsilon X^2 + 25 + 181\varepsilon X - 75 - 700\varepsilon \quad y \quad R(X) = 25 + 2521\varepsilon X - 230 - 1793\varepsilon$$

### Ejemplo13 : Si $A(X) = 6 - 4\varepsilon X^4 + 6 + 2\varepsilon X^3 - 5X + \varepsilon$ y $P(X) = 4X^2 + 6 - 3\varepsilon X + 12 - 5\varepsilon$

, entonces

$$Q(X) = 0.75 - 0.25X^2 + (-0.25 + 1.5625X - 1.75 - 1.625X) \text{ y } R(X) = (-2 - 12.25X + 21 + 11.75X)$$

Se sabe que los ceros enteros de un polinomio real se encuentran entre los divisores del término libre. Luego, el numerador de un cero fraccionario es divisor del término libre y el denominador es divisor del coeficiente director. El código necesario para calcular los ceros enteros y fraccionarios de un polinomio real se basa también en la regla de Ruffini y es la siguiente:

```
Public Function CERuf(ByRef p0() As Double) As String
    Dim i As Integer, res As String, rc As String, gp0 As Integer
    Dim gp As Integer, cr() As Double, j As Integer, p() As Double
    Dim Era As Double, c As Double, c0 As Double, pc0 As Double
    rc = Chr$(13) + Chr$(10)
    gp0 = UBound(p0())
    If p0(gp0) = 0 Then
        res = "0, ": i = 1
        Do
            If p0(gp0 - i) = 0 Then
                i = i + 1
            Else
                Exit Do
            End If
        Loop
        gp = gp0 - i
        ReDim p(gp)
        For j = 0 To gp: p(j) = p0(j): Next j
    Else
        p() = p0(): gp = gp0
    End If
    cr() = CotasCerosPR2(p())
    If p(gp) = 0 Then res = "0, "
    For i = Int(cr(2) - 1) To Int(cr(1) + 1)
        If i <> 0 Then
            c = p(gp) / i
            If c = Int(c) Then
                c = ValPolR(p(), i)
                If c = 0 Then
                    res = res + Str$(i) + ", "
                End If
            End If
        End If
    Next i
    For j = 2 To Abs(p(0))
        c0 = p(0) / j
        If c0 = Int(c0) Then
            For i = Int(cr(2) - 1) To Int(cr(1) + 1)
                If i <> 0 Then
                    c0 = p(gp) / i
                    If c0 = Int(c0) Then
                        If MaxComDiv2(i, j) = 1 Then
                            c0 = i / j
                            pc0 = ValPolR(p(), c0)
                            Era = Errpa(p(), c0)
                            If pc0 = 0 Or Abs(pc0) < Era Then
                                res = res + Str$(i) + "/" + Str$(j) + ", "
                            End If
                        End If
                    End If
                End If
            Next i
        End If
    Next j
    If Right$(res, 2) = ", " Then res = Left$(res, Len(res) - 2)
    If res = "" Then res = "¡No hay ceros enteros ni racionales!"
    CERuf = res
End Function
'-----
Public Function Errpa(ByRef p() As Double, ByVal a As Double) As Double
    Dim i As Integer, er As Double, ie As Double, gx As Integer
    Dim pd() As Double, epa As Double, rr As Double
```

```

gx = UBound(p()): ie = 0.0000000000000001
ReDim pd(gx - 1), ed(gx - 1)
'----- Polinomio derivado
For i = 0 To gx - 1: pd(i) = p(i) * (gx - i): Next i
'----- Valor absoluto de los coeficientes de pd()
For i = 0 To gx - 1: pd(i) = Abs(p(i)): Next i
'----- Cota superior del error absoluto de pa
er = ValPolR(pd(), Abs(a))
Errpa = er * ie
End Function
'-----
Public Function MaxComDiv2(ByVal a As Long, ByVal b As Long) As Long
Dim ax As Long, bx As Long, x As Long, qx As Long, rx As Long
ax = Abs(a): bx = Abs(b)
If ax < bx Then
x = ax: ax = bx: bx = x
End If
Do
rx = ax Mod bx
If rx = 0 Then Exit Do
ax = bx: bx = rx
Loop
MaxComDiv2 = bx
End Function
'-----
Public Function CotasCerosPR2(ByRef p() As Double) As Variant
'MÉTODO ÁNÓNIMO
Dim a As Double, b As Double, gp As Integer, x(2) As Double
gp = UBound(p())
a = Abs(p(1))
For i = 2 To gp
If Abs(p(i)) > a Then
a = Abs(p(i))
End If
Next i
b = Abs(p(0))
For i = 1 To gp - 1
If Abs(p(i)) > b Then
b = Abs(p(i))
End If
Next i
x(1) = 1 + a / Abs(p(0)): x(2) = -x(1)
' x(1) Cota superior ceros positivos
' x(2) Cota inferior ceros negativos
x(1) = (Int(x(1) * 100) + 1) / 100
x(2) = (Int(x(2) * 100) - 1) / 100
If x(2) < 0 Then x(2) = 0
CotasCerosPR2 = x()
End Function

```

**Ejemplo 14:** Si se considera el polinomio

$$P(X) = 24X^5 + 54X^4 + 5X^3 + 135X^2 - 119X + 21$$

, el código anterior devuelve en la variable *res* los ceros enteros y fraccionarios siguientes: -3, 1/2 y 1/4.

Para calcular los ceros (que son enteros de Gauss) de un polinomio cuyos coeficientes son enteros de Gauss se puede utilizar el código siguiente:

```

Public Function CEGRuf(ByRef p10() As Double, ByRef p20() As Double) As String
Dim i As Long, j As Long, res As String, rc As String, gp0 As Integer
Dim gp As Integer, cr() As Double, c0 As Double, c As Double, mo As Double
Dim a(2) As Double, val() As Double, cc(2) As Double, p1() As Double, p2() As Double
rc = Chr$(13) + Chr$(10): gp0 = UBound(p10())
If p10(gp0) = 0 And p20(gp0) = 0 Then
res = "0, ": i = 1
Do
If p10(gp0 - i) = 0 And p20(gp0 - i) = 0 Then
i = i + 1
Else
Exit Do
End If
Loop

```

```

    gp = gp0 - i
    ReDim p(gp)
    For j = 0 To gp: p1(j) = p10(j): p2(j) = p20(j): Next j
Else
    p1() = p10(): p2() = p20(): gp = gp0
End If
cr() = CotasCerosPC2(p1(), p2())
cc(1) = Int(cr(1)) + 1
cc(2) = Int(cr(2)) - 1
If cc(2) < 0 Then cc(2) = 0
For i = -cc(1) To cc(1)
    For j = -cc(1) To cc(1)
        a(1) = i: a(2) = j
        mo = Sqr(a(1) * a(1) + a(2) * a(2))
        If mo < cc(1) And mo > cc(2) Then
            val = ValPolC(p1(), p2(), a())
            If val(1) = 0 And val(2) = 0 Then
                res = res + FormatoNumeroComplejo(a(1), a(2)) + " , "
            End If
        End If
    Next j
Next i
Next i
If Right$(res, 2) = ", " Then res = Left$(res, Len(res) - 2)
If res = "" Then res = "No hay ceros que sean enteros de Gauss"
CEGRuf = res
End Function
'-----
Public Function ValPolC(ByRef p1() As Double, ByRef p2() As Double, ByRef a() As Double) As Variant
    Dim i As Integer, gx As Integer, coci As String, r As String, rc As String
    Dim q() As Double, x() As Double, rt() As Double, ra As String, resto(2) As Double
    gx = UBound(p1()): rc = Chr$(13) + Chr$(10)
    ReDim q(gx, 2), x(2)
    q(0, 1) = p1(0): q(0, 2) = p2(0)
    For i = 1 To gx
        x(1) = q(i - 1, 1): x(2) = q(i - 1, 2)
        rt() = ProdNC(x(), a())
        q(i, 1) = rt(1) + p1(i): q(i, 2) = rt(2) + p2(i)
    Next i
    ReDim q1(gx - 1), q2(gx - 1)
    For i = 0 To gx - 1
        q1(i) = q(i, 1): q2(i) = q(i, 2)
    Next i
    resto(1) = q(gx, 1): resto(2) = q(gx, 2)
    ValPolC = resto()
End Function
'-----
Public Function CotasCerosPC2(ByRef p1() As Double, p2() As Double) As Variant
    ' TRANSFORMACIONES DEL POLINOMIO
    Dim i As Integer, z As Integer, e As Integer, gq As Integer
    Dim a As Double, b As Double, r(2) As Double, md() As Double
    gp = UBound(p1())
    ReDim md(gp) As Double
    For i = 0 To gp
        md(i) = Sqr(p1(i) * p1(i) + p2(i) * p2(i))
    Next i
    ' Método Anónimo
    ' r(1) cota superior de los módulos de los ceros
    ' r(2) cota inferior de los módulos de los ceros
    For i = 0 To gp
        a = md(1)
        For i = 2 To gp
            If md(i) > a Then a = md(i)
        Next i
        b = md(0)
        For i = 1 To gp - 1
            If md(i) > b Then b = md(i)
        Next i
        r(1) = 1 + a / md(0)
        r(2) = md(gp) / (b + md(gp))
        r(1) = (Int(r(1) * 100) + 1) / 100
        r(2) = (Int(r(2) * 100) - 1) / 100
        If r(2) < 0 Then r(2) = 0
    Next i
    CotasCerosPC2 = r()
End Function
'-----
Public Function ProdNC(ByRef x() As Double, ByRef a() As Double) As Variant

```

```

Dim pr() As Double
ReDim pr(2)
pr(1) = x(1) * a(1) - x(2) * a(2)
pr(2) = x(1) * a(2) + a(1) * x(2)
ProdNC = pr()
End Function

```

**Ejemplo 15:** Dado el polinomio

$$P(\overline{Z}) = Z^4 + (-1+8i)\overline{Z}^3 + (-16-7i)\overline{Z}^2 + (-1+8i)\overline{Z} - 17 - 7i$$

Según el código anterior resulta que sus ceros enteros de Gauss son:  $i$ ,  $-i$ ,  $-1-5i$  y  $2-3i$ .

En el caso de los polinomios con coeficientes duales enteros la búsqueda de los ceros enteros duales se hace de la misma manera que la búsqueda de los ceros enteros de Gauss de los polinomios con coeficientes enteros de Gauss y el código para estos cálculos es muy parecido:

```

Public Function CEDRuf(ByRef p1() As Double, ByRef p2() As Double, Radio As Double) As String
    Dim i As Long, j As Long, res As String, rc As String, gp0 As Integer, r As Integer
    Dim gp As Integer, cr() As Double, c0 As Double, c As Double, mo As Double
    Dim a(2) As Double, val() As Double, cc(2) As Double, p1() As Double, p2() As Double
    rc = Chr$(13) + Chr$(10): gp0 = UBound(p1())
    r = Abs(Radio): r = Int(r)
    If p1(gp0) = 0 And p2(gp0) = 0 Then
        res = "0, ": i = 1
        Do
            If p1(gp0 - i) = 0 And p2(gp0 - i) = 0 Then
                i = i + 1
            Else
                Exit Do
            End If
        Loop
        gp = gp0 - i
        ReDim p(gp)
        For j = 0 To gp: p1(j) = p1(j): p2(j) = p2(j): Next j
    Else
        p1() = p1(): p2() = p2(): gp = gp0
    End If
    For i = -r To r
        For j = -r To r
            a(1) = i: a(2) = j
            mo = Sqr(a(1) * a(1) + a(2) * a(2))
            If mo < Radio Then
                val() = ValPolD(p1(), p2(), a())
                If val(1) = 0 And val(2) = 0 Then
                    res = res + FormatoNumeroDual(a(1), a(2)) + ", "
                End If
            End If
        Next j
    Next i
    If Right$(res, 2) = ", " Then res = Left$(res, Len(res) - 2)
    If res = "" Then
        res = "No hay ceros que sean enteros duales de módulo <= "
        res = res + Str$(Radio)
    End If
    CEDRuf = res
End Function

```

```

'-----
Public Function ValPolD(ByRef p1() As Double, ByRef p2() As Double, ByRef a() As Double) As Variant
    Dim i As Integer, gx As Integer, coci As String, r As String, rc As String
    Dim q() As Double, x() As Double, rt() As Double, ra As String, resto(2) As Double
    gx = UBound(p1()): rc = Chr$(13) + Chr$(10)
    ReDim q(gx, 2), x(2)
    q(0, 1) = p1(0): q(0, 2) = p2(0)
    For i = 1 To gx
        x(1) = q(i - 1, 1): x(2) = q(i - 1, 2)
        rt() = ProdND(x(), a())
        q(i, 1) = rt(1) + p1(i): q(i, 2) = rt(2) + p2(i)
    Next i
    ReDim q1(gx - 1), q2(gx - 1)
    For i = 0 To gx - 1
        q1(i) = q(i, 1): q2(i) = q(i, 2)
    Next i

```

```

Next i
resto(1) = q(gx, 1): resto(2) = q(gx, 2)
ValPolD = resto()
End Function
'-----
Public Function ProdND(ByRef x() As Double, ByRef a() As Double) As Variant
Dim pr() As Double
ReDim pr(2)
pr(1) = x(1) * a(1)
pr(2) = x(1) * a(2) + a(1) * x(2)
ProdND = pr()
End Function

```

**Ejemplo 16:** Si se considera el polinomio dual

$$P(\mathbb{X}) = X^4 - 6\epsilon X^3 + (-53 - 32\epsilon)X^2 + (-108 - 54\epsilon)X + 160 + 92\epsilon$$

, y se buscan los ceros duales enteros de módulo menor o igual a 25, el código anterior devuelve los ceros duales siguientes:

$$-5 + 4\epsilon, -4 - 3\epsilon, \text{ y } 8 + 5\epsilon.$$

**Ejemplo 17:** Si  $P(\mathbb{X}) = X^4 + (-10 + 5\epsilon)X^3 + (-2 - 44\epsilon)X^2 + (-32 + 112\epsilon)X - 64\epsilon$ , el código anterior devuelve las siguientes cero duales enteros de módulo menor o igual a 25:

$$-2\epsilon, 2 - 3\epsilon, 4 + k\epsilon \quad k \in \mathbb{Z} \text{ y } |k| \leq 24$$

Si el polinomio

$$P(X) = a_0 X^n + a_1 X^{n-1} + \cdots + a_{n-1} X + a_n \quad (12)$$

, tiene coeficientes reales y  $A$  es una matriz cuadrada de orden  $m$  con elementos reales, entonces el valor del polinomio  $P$  para la matriz cuadrada  $A$  se define por la igualdad:

$$P(A) = a_0 A^n + a_1 A^{n-1} + \cdots + a_{n-1} A + a_n I$$

, donde  $I$  es la matriz cuadrada unidad de orden  $m$ .

La matriz  $P(A)$  podría ser calculada hallando primero las potencias  $A^2, \dots, A^n$  de la matriz  $A$  y sumando luego las matrices

$$a_0 A^n, a_1 A^{n-1}, \dots, a_{n-1} A, a_n I$$

Efectuando los cálculos de ésta manera, el volumen de los cálculos es bastante elevado. Sin embargo, observando que

$$P(A) = (\cdots((a_0 A + a_1 I)A + a_2 I)A + a_3 I) + \cdots + a_{n-1} I)A + a_n I \quad (13)$$

, habrá muchísimo menos cálculo. Es evidente que este valor de  $P(A)$  es igual al resto de la división del polinomio

$$P'(\mathbb{X}) = a_0 I \mathbb{X}^n + \cdots + a_{n-1} I \mathbb{X} + a_n I \quad (14)$$

, con coeficientes matriciales, entre  $X - A$ . En efecto,

	$a_0 I$	$a_1 I$	...	$a_k I$	$a_{k+1} I$	...	$a_{n-1} I$	$a_n I$
$A$		$a_0 A$	...			...		
	$q_0 = a_0 I$	$q_1 = a_0 A + a_1 I$	...	$q_k$	$q_{k+1}$	...	$q_{n-1}$	$R$

, donde

$$q_k = (\cdots((a_0 A + a_1 I)A + a_2 I)A + a_3 I) + \cdots + a_{k-1} I)A + a_k I$$

$$q_{k+1} = q_k A + a_{k+1} I = (\cdots((a_0 A + a_1 I)A + a_2 I)A + a_3 I) + \cdots + a_k I)A + a_{k+1} I$$

$$R = q_{n-1} A + a_n I = (\cdots((a_0 A + a_1 I)A + a_2 I)A + a_3 I) + \cdots + a_{n-1} I)A + a_n I = P(\mathbb{X})$$

Utilizando la fórmula (13) para calcular el valor de  $P(A)$ , el código es la siguiente:

```

Public Function VMP(ByRef p() As Double, ByRef m() As Double) As Variant
Dim i As Integer, j As Integer, k As Integer
Dim n As Integer, g As Integer

```

```

g = UBound(p()): n = UBound(m())
ReDim a(n, n)
a() = MultNum(p(), m())
For k = 1 To g
    For i = 1 To n
        a(i, i) = a(i, i) + p(k)
    Next i
    If k <> g Then
        a() = MultMatCuadradas(a(), m())
    End If
Next k
VMP = a()
End Function
'-----
Public Function MultMatCuadradas(ByRef a() As Double, ByRef b() As Double) As Variant
    Dim i As Integer, j As Integer, k As Integer
    Dim n As Integer, z() As Double
    n = UBound(a())
    ReDim z(n, n)
    For i = 1 To n
        For j = 1 To n
            z(i, j) = 0
            For k = 1 To n
                z(i, j) = z(i, j) + a(i, k) * b(k, j)
            Next k
        Next j
    Next i
    MultMatCuadradas = z()
End Function
'-----
Public Function MultNum(ByVal nu As Double, ByRef d() As Double) As Variant
    Dim i As Integer, j As Integer, n As Integer, z() As Double
    n = UBound(d())
    ReDim z(n, n)
    For i = 1 To n
        For j = 1 To n
            z(i, j) = nu * d(i, j)
        Next j
    Next i
    MultNum = z()
End Function
'-----
Public Function VerMatriz(ByRef x() As Double) As Variant
    Dim i As Integer, j As Integer, n As Integer, mt As String
    n = UBound(x())
    For i = 1 To n
        mt = mt + " Fila" + Str$(i) + ": "
        For j = 1 To n
            mt = mt + Str$(x(i, j))
            If j < n Then
                mt = mt + " ; "
            End If
        Next j
        mt = mt + rc + " " + rc
    Next i
    VerMatriz = mt
End Function

```

Si el polinomio (12) y la matriz cuadrada  $A$  son complejos, para calcular el valor del polinomio para la matriz  $A$  se procede de manera análoga al caso real. En este caso el código es la siguiente:

```

Public Function VMPC(ByRef a0() As Double, ByRef p() As Double) As Variant
    Dim i As Integer, j As Integer, k As Integer, u As Integer
    Dim n As Integer, g As Integer
    n = UBound(a0(), 1): g = UBound(p(), 1)
    ReDim a(n, n, 2)

```

```

' Esquema de Riffini
For i = 1 To n
  For j = 1 To n
    a(i, j, 1) = p(0, 1) * a0(i, j, 1) - p(0, 2) * a0(i, j, 2)
    a(i, j, 2) = p(0, 1) * a0(i, j, 2) + p(0, 2) * a0(i, j, 1)
  Next j
Next i
For k = 1 To g
  For i = 1 To n
    a(i, i, 1) = a(i, i, 1) + p(k, 1)
    a(i, i, 2) = a(i, i, 2) + p(k, 2)
  Next i
  If k <> g Then
    a() = MultMatCuadradasC(a(), a0())
  End If
Next k
VMPC = a()
End Function
' -----
Public Function MultMatCuadradasC(ByRef a() As Double, ByRef b() As Double) As Variant
  Dim i As Integer, j As Integer, k As Integer, n As Integer, c() As Double
  n = UBound(a(), 1)
  ReDim c(n, n, 2)
  For i = 1 To n
    For j = 1 To n
      c(i, j, 1) = 0: c(i, j, 2) = 0
      For k = 1 To n
        c(i, j, 1) = c(i, j, 1) + a(i, k, 1) * b(k, j, 1) - a(i, k, 2) * b(k, j, 2)
        c(i, j, 2) = c(i, j, 2) + a(i, k, 1) * b(k, j, 2) + a(i, k, 2) * b(k, j, 1)
      Next k
    Next j
  Next i
  MultMatCuadradasC = c()
End Function
' -----
Public Function VerMatrizC(ByRef x() As Double) As String
  Dim i As Integer, j As Integer, r As String, rc As String
  n = UBound(x()): rc = Chr$(13) + Chr$(10)
  For i = 1 To n
    r = r + " Fila" + Str$(i) + ": "
    For j = 1 To n
      r = r + FormatoComplejo(x(i, j, 1), x(i, j, 2))
      If j < n Then
        r = r + "; "
      End If
    Next j
    r = r + rc
  Next i
  VerMatrizC = r
End Function
' -----
Public Function FormatoComplejo(pr As Double, pi As Double)
  'Escritura de un número complejo en una caja de texto.
  nc = ""
  If pr <> 0 Then
    If Abs(pr) >= 1 Then
      nc = nc + Str$(pr)
    Else
      If pr > 0 Then
        nc = nc + "0" + Str$(pr)
      Else
        nc = nc + "-0" + Mid$(Str$(pr), 2)
      End If
    End If
  End If
  If pi <> 0 Then
    If Abs(pi) = 1 Then
      If pi = 1 Then
        If pr <> 0 Then nc = nc + " + "
      Else
        nc = nc + " - "
      End If
    Else
      If pi > 0 Then
        If pr <> 0 Then

```



```

        nc = nc + " + "
    End If
    If pi > 1 Then
        nc = nc + Str$(pi)
    Else
        nc = nc + "0" + Str$(pi)
    End If
Else
    If pi < -1 Then
        nc = nc + " - " + Mid$(Str$(pi), 2)
    Else
        nc = nc + " - 0" + Mid$(Str$(pi), 2)
    End If
End If
End If
nc = nc + "i"
End If
If nc = "" Then nc = "0"
FormatoComplejo = nc
End Function

```

**Ejemplo 18:** Si  $P(X) = (1+i)X^3 + (2+3i)X^2 + (1-i)X + 3+7i$  y  $A = \begin{pmatrix} -1+i & 4-3i \\ 2+3i & 1+5i \end{pmatrix}$ ,

entonces  $P(A) = \begin{pmatrix} 270+223i & -238+141i \\ -92-177i & -356-9i \end{pmatrix}$

En el caso del cálculo del valor de un polinomio dual para una matriz cuadrada dual, (efectuando pequeñas modificaciones en el código anterior) se pueden utilizar las funciones siguientes:

```

Public Function VPMD(ByRef a0() As Double, ByRef p() As Double) As Variant
    Dim i As Integer, j As Integer, k As Integer, u As Integer
    Dim n As Integer, g As Integer
    n = UBound(a0(), 1): g = UBound(p(), 1)
    ReDim a(n, n, 2)
    ' Esquema de Riffini
    For i = 1 To n
        For j = 1 To n
            a(i, j, 1) = p(0, 1) * a0(i, j, 1) - p(0, 2) * a0(i, j, 2)
            a(i, j, 2) = p(0, 1) * a0(i, j, 2) + p(0, 2) * a0(i, j, 1)
        Next j
    Next i
    For k = 1 To g
        For i = 1 To n
            a(i, i, 1) = a(i, i, 1) + p(k, 1)
            a(i, i, 2) = a(i, i, 2) + p(k, 2)
        Next i
        If k <> g Then
            a() = ProdMD(a(), a0())
        End If
    Next k
    VPMD = a()
End Function
'-----
Public Function ProdMD(ByRef a() As Double, ByRef b() As Double) As Variant
    Dim i As Integer, j As Integer, k As Integer, n As Integer, c() As Double
    n = UBound(a(), 1)
    ReDim c(n, n, 2)
    For i = 1 To n
        For j = 1 To n
            c(i, j, 1) = 0: c(i, j, 2) = 0
            For k = 1 To n
                c(i, j, 1) = c(i, j, 1) + a(i, k, 1) * b(k, j, 1)
                c(i, j, 2) = c(i, j, 2) + a(i, k, 1) * b(k, j, 2) + a(i, k, 2) * b(k, j,
1)
            Next k
        Next j
    Next i
    ProdMD = c()
End Function
'-----

```

```

Public Function VerMatrizResultado(ByRef x() As Double) As String
    Dim i As Integer, j As Integer, r As String, rc As String
    n = UBound(x()): rc = Chr$(13) + Chr$(10)
    For i = 1 To n
        r = r + " Fila" + Str$(i) + ":  "
        For j = 1 To n
            r = r + FormatoDual(x(i, j, 1), x(i, j, 2))
            If j < n Then
                r = r + " ; "
            End If
        Next j
        r = r + rc
    Next i
    VerMatrizResultado = r
End Function
'-----
Public Function FormatoDual(pr As Double, pi As Double) As String
    'Escritura de un número complejo en una caja de texto.
    nc = ""
    If pr <> 0 Then
        If Abs(pr) >= 1 Then
            nc = nc + Str$(pr)
        Else
            If pr > 0 Then
                nc = nc + "0" + Str$(pr)
            Else
                nc = nc + "-0" + Mid$(Str$(pr), 2)
            End If
        End If
    End If
    If pi <> 0 Then
        If Abs(pi) = 1 Then
            If pi = 1 Then
                If pr <> 0 Then nc = nc + " + "
            Else
                nc = nc + " - "
            End If
        Else
            If pi > 0 Then
                If pr <> 0 Then
                    nc = nc + " + "
                End If
                If pi > 1 Then
                    nc = nc + Str$(pi)
                Else
                    nc = nc + "0" + Str$(pi)
                End If
            Else
                If pi < -1 Then
                    nc = nc + " - " + Mid$(Str$(pi), 2)
                Else
                    nc = nc + " - 0" + Mid$(Str$(pi), 2)
                End If
            End If
        End If
        nc = nc + " e"
    End If
    If nc = "" Then nc = "0"
    FormatoDual = nc
End Function

```

**Ejemplo 19:** Si  $P(X) = 2X^3 - 3X^2 + 5X - 7$  y  $A = \begin{pmatrix} -1+\varepsilon & 4-3\varepsilon \\ 2+3\varepsilon & 1+5\varepsilon \end{pmatrix}$ , entonces

$$P(A) = \begin{pmatrix} -57+99\varepsilon & 92-61\varepsilon \\ 46+73\varepsilon & -71+135\varepsilon \end{pmatrix}.$$

Naturalmente, en este ejemplo los coeficientes del polinomio son reales pero se han introducido como números duales con la parte dual nula.

### Bibliografia:

- [1] Ion D. Ion, Nicolae Radu, Algebră, Editura Didactică și Pedagogică, București, 1970.
- [2] A.I. Froda, Algebră superioară, Editura Didactică și Pedagogică, București, 1958.
- [3] Aladar Peter Santha, Acotación de los ceros de un polinomio, Monografias.com, 2012
- [4] Aladar Peter Santha, Cálculos con números enteros largos en ordenadores, Monografias.com, 2012.