

Cálculo aproximado da las integrales definidas de las funciones derivables.
Aladar Peter Santha

Résumé:

Dans cet article on expose le calcul de l'intégral $I = \int_a^b f(x)dx$ d'une fonction $(2m+1)$ -fois derivable sur l'intervalle $[a, b]$. Si les points x_i , $i=1, \dots, 2p-1$ divisent l'intervalle en $2p$ parties de la même longueur $h = (b-a)/(2p)$, alors

$$I = 2h \sum_{i=0}^{p-1} f(x_{2i+1}) + \frac{2h^3}{3!} \sum_{i=1}^p f''(x_{2i+1}) + \dots + \frac{2h^{2m+1}}{(2m+1)!} \sum_{i=1}^p f^{(2m)}(x_{2i+1}) + R$$

, où

$$x \in [a, b] \quad \text{et} \quad |f^{(2m+1)}(x)| < M \quad \Rightarrow \quad |R| \leq \frac{b-a}{2p} \frac{M}{(2m+1)!}$$

On expose aussi la programmation de ces calculs à partir des dérivées calculées d'une manière symbolique ou approchée.

Si f es una función continua en el intervalo $[a, b]$ entonces existen varios métodos numéricos para el cálculo de la integral definida:

$$I = \int_a^b f(x)dx. \quad (1)$$

La mayoría de estos métodos utilizan solamente los valores de la función f que han sido calculados para ciertos puntos del intervalo $[a, b]$. Suponiendo que la función f es $(2m+1)$ - veces derivable en el intervalo $[a, b]$ y considerando la siguiente división del intervalo $[a, b]$,

$$a = x_0 < x_1 < x_2 < \dots < x_{2p} = b$$

$$x_{i+1} - x_i = t = \frac{b-a}{2p} \quad i = 0, 1, 2, \dots, 2p-1$$

, sobre cada uno de los intervalos:

$$[x_{2i-1}, x_{2i}] \quad i = 1, 2, \dots, p$$

, la función se aproximarán por el polinomio Taylor de grado $2m$, asociado a la función f en el punto x_{2i-1} . Así, para $x \in [x_{2i-1}, x_{2i}]$

$$f(x) = \sum_{k=0}^{2m} \frac{f^{(k)}(x_{2i-1})}{k!} (x - x_{2i-1})^k + \frac{f^{(2m+1)}(c_i)}{(2m+1)!} (x - x_{2i-1})^{2m+1}$$

, donde c_i depende de x y está comprendido entre x_{2i-1} y x_{2i} . Por tanto,

$$I_{2i-1} = \int_{x_{2i-2}}^{x_{2i}} f(x)dx = \sum_{k=0}^{2m} \int_{x_{2i-2}}^{x_{2i}} \frac{f^{(k)}(x_{2i-1})}{k!} (x - x_{2i-1})^k dx + R_{2i-1}$$

, donde

$$R_{2i-1} = \int_{x_{2i-2}}^{x_{2i}} \frac{f^{(2m+1)}(c_i)}{(2m+1)!} (x - x_{2i-1})^{2m+1} dx.$$

A continuación,

$$I_{2i-1} = \sum_{k=0}^{2m} \frac{f^{(k)}(x_{2i-1})}{k+1} \left| x - x_{2i-1} \right|^{k+1} + R_{2i-1}.$$

Luego, puesto que

$$\left| x - x_{2i-1} \right|^{k+1} = |x_{2i} - x_{2i-1}|^{k+1} - |x_{2i} - x_{2i-1}|^{k+1} = t^{k+1} - (-t)^{k+1} = \begin{cases} 0 & \text{si } k \text{ es impar} \\ 2t^{k+1} & \text{si } k \text{ es par} \end{cases}$$

, tendremos

$$I_{2i-1} = 2tf(x_{2i-1}) + \frac{2t^3}{3!} f''(x_{2i-1}) + \cdots + \frac{2t^{2m+1}}{(2m+1)!} f^{2m}(x_{2i-1}) + R_{2i-1}$$

, y así $I = I_1 + I_3 + \cdots + I_{2p-1}$, es decir

$$I = 2t \sum_{i=1}^p f(x_{2i-1}) + \frac{2t^3}{3!} \sum_{i=1}^p f''(x_{2i-1}) + \cdots + \frac{2t^{2m+1}}{(2m+1)!} \sum_i^p f^{2m}(x_{2i-1}) \quad (2)$$

, donde

$$R = \sum_{i=1}^p R_{2i-1} \quad (3)$$

Si la función f^{2m+1} está acotada sobre el intervalo $[a, b]$, es decir,

$$|f^{2m+1}| \leq M$$

, para cualquier $x \in [a, b]$, entonces

$$\begin{aligned} |R| &\leq \sum_{i=1}^p |R_{2i-1}| \leq \frac{2M}{(2m+1)!} \sum_{i=1}^p \left| \int_{x_{2i-1}}^{x_{2i}} f^{2m+1}(x) dx \right| \\ &\leq \frac{2M}{(2m+1)!} \sum_{i=1}^p \left| \left[\frac{f^{2m+2}(x)}{2m+2} \right]_{x_{2i-1}}^{x_{2i}} \right| \leq \frac{2M}{(2m+2)!} \sum_{i=1}^p |f^{2m+2}(x)| \\ &\leq \frac{2pM}{(2m+2)!} t^{2m+2} = \frac{M}{(p+1)(2m+2)!} t^{2m+2} \end{aligned} \quad (4)$$

Ejemplo 1: Sea f la función definida por

$$f(x) = \exp(-0.5x^2) \quad (5)$$

, y sean $a = 0$, $b = 2$ y $m = 3$. Las derivadas de orden 2, 4, 6 y 7 de la función f tienen las expresiones siguientes:

$$\begin{aligned} f''(x) &= (x^2 - 1) \exp(-0.5x^2) \\ f^{(4)}(x) &= (x^4 - 6x^2 + 3) \exp(-0.5x^2) \\ f^{(6)}(x) &= (x^6 - 15x^4 + 45x^2 - 15) \exp(-0.5x^2) \\ f^{(7)}(x) &= (-x^7 + 21x^5 - 108x^3 + 111x) \exp(-0.5x^2) \end{aligned}$$

Construyendo la gráfica de la función $y = |f^{(7)}(x)|$ por ordenador, se observa que $M = 38$ es una cota superior de esta función en el intervalo $[0, 2]$ y así para $x \in [0, 2]$ y $2p = 60$, tendremos

$$|R| \leq \frac{38 \cdot 2^8}{60^7 \cdot 8!} = 0.00000000000000861875 \cdots < 10^{-13}$$

Para calcular la integral, el código Visual-Basic es la siguiente:

```

Public Function f(ByVal x As Double) As Double
     $f = \exp(-0.5 * x * x)$ 
End Function
-----
Public Function f2(ByVal x As Double) As Double
     $f_2(x) = (x^2 - 1) \exp(-0.5 * x * x)$ 
End Function
-----
Public Function f4(ByVal x As Double) As Double
     $f_4(x) = (x^4 - 6 * x^2 + 3) \exp(-0.5 * x * x)$ 
End Function
-----
Public Function f6(ByVal x As Double) As Double
     $f_6(x) = (x^6 - 15 * x^4 + 45 * x^2 - 15) \exp(-0.5 * x * x)$ 
End Function
-----
Public Function MST6(ByVal a As Double, ByVal b As Double, ByVal np As Integer) As String
    Dim i As Integer, s As Double, d As Double, a1 As Double, p As Double
    Dim s0 As Double, s2 As Double, s4 As Double, s6 As Double
    Dim j As Integer, sa As Double, sb As Double, n1 As Integer
    '---- Cálculo de la integral
    For j = 1 To 2
        s0 = 0: s2 = 0: s4 = 0: s6 = 0: s = 0
        If j = 1 Then n1 = np Else n1 = 2 * np
        d = (b - a) / n1: p = n1 / 2: a1 = a + d
        For i = 1 To p
            s0 = s0 + f(a1)
            s2 = s2 + f2(a1)
            s4 = s4 + f4(a1)
            s6 = s6 + f6(a1)
            a1 = a1 + 2 * d
        Next i
        s = 2 * d * s0 + (d ^ 3 / 3) * s2 + (d ^ 5 / 60) * s4 + (d ^ 7 / 2520) * s6
        If j = 1 Then sa = s Else sb = s
    Next j
    MST6 = precision(sa, sb)
End Function
-----
Public Function precision(ByVal t1 As Double, ByVal t2 As Double) As String
    Dim se As Double, sf As Double, sd As Double, sg As Double
    Dim p As Integer, t As Double, k As Integer, i As Integer
    Dim sol As String, rc As String
    p = Len(Str$(Fix(t1))) - 1: rc = Chr$(13) + Chr$(10)
    If t1 = t2 Then
        sol = "I = " + RutinaEdicion(t1)
    Else
        If Fix(t1) <> Fix(t2) Then
            sol = "I = " + RutinaEdicion(t1)
            sol = sol + rc + rc
            sol = sol + "Doblando el número de los nodos," + rc + rc
            sol = sol + "I = " + RutinaEdicion(t2)
        Else
            se = t1: sf = t2: sd = 1: k = 1
            Do
                se = se * 10: sf = sf * 10
                If Fix(se) <> Fix(sf) Then
                    sg = Fix(se)
                End If
            Loop
        End If
    End If
End Function

```

```

        Exit Do
    Else
        If k = 15 - p + 1 Then
            sg = Fix(se)
            Exit Do
        Else
            k = k + 1
        End If
    End If
Loop
For i = 1 To k
    sd = sd * 10
Next i
t = sg / sd
sol = "I = " + RutinaEdicion(t)
End If
End If
precision = sol
End Function
'-----
```

```

Public Function RutinaEdicion(ByVal t As Double) As String
    Dim u As String
    If Abs(t) < 1 Then
        If t < 0 Then u = u + "- 0" Else u = u + "0"
        If t >> 0 Then u = u + Str$(Abs(t))
    Else
        u = u + Str$(t)
    End If
    RutinaEdicion = u
End Function
```

Dividiendo el intervalo $[1, 2]$ en $np = 34$ partes iguales, el código de arriba devuelve el siguiente valor de la integral:

$$I = 1.1962880133 2261 \quad (6)$$

, que tiene todas sus cifras exactas, excepto la última que está redondeada.

Si el cálculo manual de las derivadas es laborioso, para hallar el valor de la integral se pueden utilizar solamente las derivadas de orden 2 y 4 ó solamente la derivada de orden 2, con las funciones siguientes:

```

Public Function MST4(ByVal a As Double, ByVal b As Double, ByVal np As Integer) As String
    Dim i As Integer, s As Double, d As Double, a1 As Double
    Dim s0 As Double, s2 As Double, s4 As Double, p As Double
    Dim j As Integer, sa As Double, sb As Double, n1 As Integer
    ' ----- Cálculo de la integral
    For j = 1 To 2
        If j = 1 Then n1 = np Else n1 = 2 * np
        s0 = 0: s2 = 0: s4 = 0: s = 0
        d = (b - a) / n1: p = n1 / 2: a1 = a + d
        For i = 1 To p
            s0 = s0 + f(a1)
            s2 = s2 + f2(a1)
            s4 = s4 + f4(a1)
            a1 = a1 + 2 * d
        Next i
        s = 2 * d * s0 + (d ^ 3 / 3) * s2 + (d ^ 5 / 60) * s4
        If j = 1 Then sa = s Else sb = s
    Next j
    MST4 = precision(sa, sb)
```

```

End Function
'----- Cálculo de la integral
Public Function MST2(ByVal a As Double, ByVal b As Double, ByVal np As Integer) As String
    Dim i As Integer, s As Double, d As Double, a1 As Double
    Dim s0 As Double, s2 As Double, p As Double
    Dim j As Integer, sa As Double, sb As Double, n1 As Integer
    '----- Cálculo de la integral
    For j = 1 To 2
        If j = 1 Then n1 = np Else n1 = 2 * np
        s0 = 0: s2 = 0: s = 0
        d = (b - a) / n1: p = n1 / 2: a1 = a + d
        For i = 1 To p
            s0 = s0 + f(a1)
            s2 = s2 + f2(a1)
            a1 = a1 + 2 * d
        Next i
        s = 2 * d * s0 + (d ^ 3 / 3) * s2
        If j = 1 Then sa = s Else sb = s
    Next j
    MST2 = precision(sa, sb)
End Function

```

Para alcanzar la misma precisión que en el apartado (6), con la función MST4 se debería dividir el intervalo $[0,2]$ en $np = 148$ partes iguales, y con la función MST2 se debería dividir el mismo intervalo en $np = 1350$ partes iguales.

En el caso cuando el cálculo de las derivadas es complicada y necesita mucho tiempo, se podría utilizar la función siguiente, que emplea el cálculo aproximado de las derivadas de orden superior:

```

Public Function MST4B(ByVal a As Double, ByVal b As Double, ByVal np As Integer) As String
    Dim s0 As Double, s2 As Double, s4 As Double, p As Integer
    Dim i As Integer, s As Double, d As Double, a1 As Double, h As Double
    Dim j As Integer, sa As Double, sb As Double, n1 As Integer
    '----- Cálculo de la integral
    For j = 1 To 2
        If j = 1 Then n1 = np Else n1 = 2 * np
        s0 = 0: s2 = 0: s4 = 0: s = 0
        d = (b - a) / n1: p = n1 / 2: a1 = a + d
        For i = 1 To p
            s0 = s0 + f(a1)
            h = 0.0000001
            s2 = s2 + (f(a1 + 2 * h) - 2 * f(a1 + h) + f(a1)) / (h ^ 2)
            h = 0.0001
            s4 = s4 + (f(a1 + 4 * h) - 4 * f(a1 + 3 * h) + 6 * f(a1 + 2 * h) - 4 * f(a1 + h) + f(a1)) / (h ^ 4)
            a1 = a1 + 2 * d
        Next i
        s = 2 * d * s0 + (d ^ 3 / 3) * s2 + (d ^ 5 / 60) * s4
        If j = 1 Then sa = s Else sb = s
    Next j
    MST4B = precision(sa, sb)
End Function

```

Calculando la integral definida de la función (5) entre los límites $a = 0$ y $b = 2$ con la función MST4B, dividiendo el intervalo en 1000 partes iguales, se obtiene el resultado:

$$I = 1.196288017$$

, donde todas las cifras son exactas, excepto la última.

Si la función es integrable en el intervalo $I = [a, b]$ pero no es derivable en un número finito de puntos

$c_i \in [c_{i-1}, c_i]$ del intervalo I , entonces

$$\int_a^b f(x) dx = \int_a^{c_1} f(x) dx + \cdots + \int_{c_{i-1}}^{c_i} f(x) dx + \cdots + \int_{c_k}^b f(x) dx$$

A continuación se exponen funciones para el cálculo de las integrales definidas de las funciones derivables, que utilizan los cálculos con los enteros y decimales largos, el cálculo de las funciones elementales con precisión grande y el cálculo aproximado de las derivadas de orden superior. Naturalmente esto hará que los cálculos duren más tiempo. Sin embargo, en el futuro se espera una considerable aumento de la velocidad de los ordenadores y entonces estas funciones serán más rápidas y podrían sobrepasar la precisión de los métodos habituales actuales.

```
Public Function MST2C(ByVal ax As String, ByVal bx As String, ByVal np As Integer) As String
    Dim i As Integer, s As String, d As String, a1 As String, x(2) As String
    Dim s0 As String, s2 As String, p As Integer, n As Integer, h As String
    Dim sx As String, pr As Integer, sb As String, nn(2) As String
    Dim j As Integer, sa As String
    n = 7
    ' pr es la precisión en los cálculos con decimales
    '----- Cálculo de la integral
    nn(1) = Mid$(Str$(Abs(np)), 2): pr = 16
    x(1) = nn(1): x(2) = "2": nn(2) = Multiplicar(x(), n)
    For j = 1 To 2
        s0 = "0": s2 = "0": s = "0"
        x(1) = bx: x(2) = ax: x(1) = Restar(x(), n): x(2) = nn(j)
        d = DividirDec(x(), 12, n): x(1) = ax: x(2) = d: a1 = SumarDec(x(), n)
        p = Val(nn(j)) / 2
        For i = 1 To p
            x(1) = s0: x(2) = g(a1, pr): s0 = SumarDec(x(), n)
            x(1) = s2: x(2) = g2(a1): s2 = SumarDec(x(), n)
            x(1) = d: x(2) = "2": x(1) = MultiplicarDec(x(), n): x(2) = a1: a1 = SumarDec(x(), n)
        Next i
        x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = s0
        s = MultiplicarDec(x(), n)
        x(1) = d: x(2) = "3": x(1) = PotenciasDec(x(), n): x(2) = "3"
        x(1) = DividirDec(x(), pr, n): x(2) = s2: x(1) = MultiplicarDec(x(), n): x(2) = s
        s = SumarDec(x(), n)
        If j = 1 Then sa = s Else sb = s
    Next j
    MST2C = PrecisionString(sa, sb)
End Function
```

```
Public Function MST4C(ByVal ax As String, ByVal bx As String, ByVal np As Integer) As String
    Dim s0 As String, s2 As String, s4 As String, p As Integer, x(2) As String
    Dim i As Integer, s As String, d As String, a1 As String, nn(2) As String
    Dim j As Integer, sa As String, sb As String, n As Integer
    Dim pr As Integer
    n = 7
    ' pr es la precisión en los cálculos con decimales
    '----- Cálculo de la integral
    nn(1) = Mid$(Str$(Abs(np)), 2): pr = 16
    x(1) = nn(1): x(2) = "2": nn(2) = Multiplicar(x(), n)
    For j = 1 To 2
        s0 = "0": s2 = "0": s4 = "0": s = "0"
        x(1) = bx: x(2) = ax: x(1) = RestarDec(x(), n): x(2) = nn(j)
        d = DividirDec(x(), pr, n): p = Val(nn(j)) / 2
        x(1) = ax: x(2) = d: a1 = SumarDec(x(), n)
        For i = 1 To p
            x(1) = s0: x(2) = g(a1, pr): s0 = SumarDec(x(), n)
            x(1) = s2: x(2) = g2(a1): s2 = SumarDec(x(), n)
```

```

x(1) = s4: x(2) = g4(a1): s4 = SumarDec(x(), n)
x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = a1: a1 = SumarDec(x(), n)
Next i
x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = s0
s = MultiplicarDec(x(), n)
x(1) = d: x(2) = "3": x(1) = PotenciasDec(x(), n): x(2) = "3"
x(1) = DividirDec(x(), pr, n): x(2) = s2: x(1) = MultiplicarDec(x(), n): x(2) = s
s = SumarDec(x(), n): x(1) = d: x(2) = "5": x(1) = PotenciasDec(x(), n): x(2) = "60"
x(1) = DividirDec(x(), pr, n): x(2) = s4: x(1) = MultiplicarDec(x(), n): x(2) = s
s = SumarDec(x(), n)
If j = 1 Then sa = s Else sb = s
Next j
MST4C = PrecisionString(sa, sb)
End Function
=====

Public Function MST6C(ByVal ax As String, ByVal bx As String, ByVal np As Integer) As String
    Dim s0 As String, s2 As String, s4 As String, s6 As String, x(2) As String
    Dim i As Integer, s As String, d As String, a1 As String, pr As Integer, nn(2) As String
    Dim j As Integer, sa As String, sb As String, n As Integer, p As Integer
    n = 7
    ' pr es la precisión en los cálculos con decimales
    '----- Cálculo de la integral
    nn(1) = Mid$(Str$(Abs(np)), 2): pr = 16
    x(1) = nn(1): x(2) = "2": nn(2) = Multiplicar(x(), n)
    For j = 1 To 2
        s0 = "0": s2 = "0": s4 = "0": s6 = "0": s = "0"
        x(1) = bx: x(2) = ax: x(1) = RestarDec(x(), n): x(2) = nn(j)
        d = DividirDec(x(), pr, n): p = Val(nn(j)) / 2
        x(1) = ax: x(2) = d: a1 = SumarDec(x(), n)
        For i = 1 To p
            x(1) = s0: x(2) = g(a1, pr): s0 = SumarDec(x(), n)
            x(1) = s2: x(2) = g2(a1): s2 = SumarDec(x(), n)
            x(1) = s4: x(2) = g4(a1): s4 = SumarDec(x(), n)
            x(1) = s6: x(2) = g6(a1): s6 = SumarDec(x(), n)
            x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = a1: a1 = SumarDec(x(), n)
        Next i
        x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = s0
        s = MultiplicarDec(x(), n)
        x(1) = d: x(2) = "3": x(1) = PotenciasDec(x(), n): x(2) = "3"
        x(1) = DividirDec(x(), pr, n): x(2) = s2: x(1) = MultiplicarDec(x(), n): x(2) = s
        s = SumarDec(x(), n): x(1) = d: x(2) = "5": x(1) = PotenciasDec(x(), n): x(2) = "60"
        x(1) = DividirDec(x(), pr, n): x(2) = s4: x(1) = MultiplicarDec(x(), n): x(2) = s
        s = SumarDec(x(), n): x(1) = d: x(2) = "7": x(1) = PotenciasDec(x(), n): x(2) = "2520"
        x(1) = DividirDec(x(), pr, n): x(2) = s6: x(1) = MultiplicarDec(x(), n): x(2) = s
        s = SumarDec(x(), n)
        If j = 1 Then sa = s Else sb = s
    Next j
    MST6C = PrecisionString(sa, sb)
End Function
=====

Public Function PrecisionString(ByVal sa As String, ByVal sb As String) As String
    Dim lsa As Integer, i As Integer, x(2) As String, n As Integer, z As String, sol As String
    Dim lsb As Integer, intsa As String, intsb As String, frsa As String, frsb As String
    n = 7: lsa = Len(sa): lsb = Len(sb)
    For i = 1 To lsa
        z = Right$(Left$(sa, i), 1)
        If z = "." Then
            intsa = Left$(sa, i - 1): frsa = Right$(sa, lsa - i)
            Exit For
        End If
    Next i
    For i = 1 To lsb

```

```

z = Right$(Left$(sb, i), 1)
If z = "." Then
    intsb = Left$(sb, i - 1): frsb = Right$(sb, lsb - i)
    Exit For
End If
Next i
If intsa <> intsb Then
    sol = intsa
Else
    For i = 1 To Len(frsa)
        If Right$(Left$(frsa, i), 1) <> Right$(Left$(frsb, i), 1) Then
            sol = intsa + "." + Left$(frsa, i - 1)
            Exit For
        End If
    Next i
End If
PrecisionString = sol
End Function
' =====

Public Function g(ByVal x0 As String, pr As Integer) As String
    Dim u As String, x(2) As String, n As Integer
    n = 7
    x(1) = x0: x(2) = x0: x(1) = MultiplicarDec(x(), n): x(2) = "-0.5"
    u = MultiplicarDec(x(), n)
    g = ExpP(u, pr)
End Function
' =====

Public Function g2(ByVal x0 As String) As String
    Dim h As String, sx As String, x(2) As String
    Dim b(2) As String, n As Integer, pr As Integer
    n = 7: h = "0.00000000001": pr = 2 * (Len(h) - 2) + 16
    b(0) = x0
    x(1) = b(0): x(2) = h: b(1) = SumarDec(x(), n)
    x(1) = b(1): x(2) = h: b(2) = SumarDec(x(), n)
    x(1) = "-2": x(2) = g(b(1), pr): sx = MultiplicarDec(x(), n)
    x(1) = sx: x(2) = g(b(0), pr): sx = SumarDec(x(), n)
    x(1) = sx: x(2) = g(b(2), pr): sx = SumarDec(x(), n)
    x(1) = h: x(2) = "2": x(2) = PotenciasDec(x(), n): x(1) = sx
    g2 = DividirDec(x(), 16, n)
End Function
' =====

Public Function g4(ByVal x0 As String) As String
    Dim h As String, sx As String, x(2) As String
    Dim b(4) As String, n As Integer, i As Integer, pr As Integer
    n = 7: h = "0.0000001": pr = 4 * (Len(h) - 2) + 16
    b(0) = x0
    For i = 1 To 4
        x(1) = b(i - 1): x(2) = h: b(i) = SumarDec(x(), n)
    Next i
    x(1) = g(b(0), pr): x(2) = g(b(4), pr): sx = SumarDec(x(), n)
    x(1) = g(b(1), pr): x(2) = g(b(3), pr): x(1) = SumarDec(x(), n): x(2) = "-4"
    x(1) = MultiplicarDec(x(), n): x(2) = sx: sx = SumarDec(x(), n)
    x(1) = g(b(2), pr): x(2) = "6": x(1) = MultiplicarDec(x(), n)
    x(2) = sx: sx = SumarDec(x(), n): x(1) = h: x(2) = "4"
    x(2) = PotenciasDec(x(), n): x(1) = sx
    g4 = DividirDec(x(), 16, n)
End Function
' =====

Public Function g6(ByVal x0 As String) As String
    Dim h As String, sx As String, x(2) As String
    Dim b(6) As String, n As Integer, i As Integer, pr As Integer
    n = 7: h = "0.00001": pr = 6 * (Len(h) - 2) + 16

```

```

b(0) = x0
For i = 1 To 6
    x(1) = b(i - 1); x(2) = h; b(i) = SumarDec(x(), n)
Next i
x(1) = g(b(0), pr); x(2) = g(b(6), pr); sx = SumarDec(x(), n)
x(1) = g(b(1), pr); x(2) = g(b(5), pr); x(1) = SumarDec(x(), n); x(2) = "-6"
x(1) = MultiplicarDec(x(), n); x(2) = sx; sx = SumarDec(x(), n)
x(1) = g(b(2), pr); x(2) = g(b(4), pr); x(1) = SumarDec(x(), n); x(2) = "15"
x(1) = MultiplicarDec(x(), n); x(2) = sx; sx = SumarDec(x(), n)
x(1) = g(b(3), pr); x(2) = "-20"; x(1) = MultiplicarDec(x(), n)
x(2) = sx; sx = SumarDec(x(), n); x(1) = h; x(2) = "6"
x(2) = PotenciasDec(x(), n); x(1) = sx
g6 = DividirDec(x(), 16, n)
End Function

```

Para que el tiempo de ejecución sea menor, se recomienda que el número de las partes en que se divide el intervalo sea una potencia de 2, aunque podría ser también cualquier número par.

Al efectuar el cálculo de la integral de la función (5) entre los límites $a = 0$ y $b = 2$ con las funciones MST2C, MST4C y MST6C, se obtienen los siguientes valores para la integral:

Nr. Divisiones:	8	16	32	64	128
MST2C	1.1962	1.196288	1.1962880	1.19628801	1.196288013
MST4C	1.19628	1.19628801	1.196288013	1.196288013322	1.19628801332260

Nr. Didivisiones	8	16	32	64
MST6C	1.196288013	1.19628801332	1.196288013322	1.19628801332260

Finalmente, hallando las derivadas manualmente y utilizando el cálculo de las funciones elementales con precisión grande, para calcular el valor de la integral se pueden considerar también las funciones siguientes:

```

Public Function MST2D(ByVal ax As String, ByVal bx As String, ByVal np As Integer) As String
    Dim i As Integer, s As String, d As String, a1 As String, x(2) As String
    Dim s0 As String, s2 As String, p As Integer, n As Integer, j As Integer
    Dim sx As String, pr As Integer, nn(2) As String, sa As String, sb As String
    n = 7; pr = 24; nn(1) = Mid$(Str$(Abs(np)), 2)
    x(1) = nn(1); x(2) = "2"; nn(2) = Multiplicar(x(), n)
    ' pr es la precisión en los cálculos con decimales
    '---- Cálculo de la integral
    For j = 1 To 2
        s0 = "0"; s2 = "0"; s = "0"
        x(1) = bx; x(2) = ax; x(1) = Restar(x(), n); x(2) = nn(j)
        d = DividirDec(x(), 12, n); x(1) = ax; x(2) = d; a1 = SumarDec(x(), n)
        p = Val(nn(j)) / 2
        For i = 1 To p
            x(1) = s0; x(2) = h(a1, pr); s0 = SumarDec(x(), n)
            x(1) = s2; x(2) = h2(a1, pr); s2 = SumarDec(x(), n)
            x(1) = d; x(2) = "2"; x(1) = MultiplicarDec(x(), n); x(2) = a1; a1 = SumarDec(x(), n)
        Next i
        x(1) = "2"; x(2) = d; x(1) = MultiplicarDec(x(), n); x(2) = s0
        s = MultiplicarDec(x(), n)
        x(1) = d; x(2) = "3"; x(1) = PotenciasDec(x(), n); x(2) = "3"
        x(1) = DividirDec(x(), pr, n); x(2) = s2; x(1) = MultiplicarDec(x(), n); x(2) = s
        s = SumarDec(x(), n)
    End Function

```

```

If j = 1 Then sa = s Else sb = s
Next j
MST2D = PrecisionString(sa, sb)
End Function
' =====

Public Function MST4D(ByVal ax As String, ByVal bx As String, ByVal np As Integer) As String
Dim s0 As String, s2 As String, s4 As String, p As Integer, x(2) As String
Dim i As Integer, s As String, d As String, a1 As String, pr As Integer
Dim j As Integer, sa As String, sb As String, n As Integer, nn(2) As String
' pr es la precisión en los cálculos con decimales
'---- Cálculo de la integral
n = 7: pr = 24: nn(1) = Mid$(Str$(Abs(np)), 2)
x(1) = nn(1): x(2) = "2": nn(2) = Multiplicar(x(), n)
For j = 1 To 2
    s0 = "0": s2 = "0": s4 = "0": s = "0"
    x(1) = bx: x(2) = ax: x(1) = RestarDec(x(), n): x(2) = nn(j)
    d = DividirDec(x(), pr, n): p = Val(nn(j)) / 2
    x(1) = ax: x(2) = d: a1 = SumarDec(x(), n)
    For i = 1 To p
        x(1) = s0: x(2) = h(a1, pr): s0 = SumarDec(x(), n)
        x(1) = s2: x(2) = h2(a1, pr): s2 = SumarDec(x(), n)
        x(1) = s4: x(2) = h4(a1, pr): s4 = SumarDec(x(), n)
        x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = a1: a1 = SumarDec(x(), n)
    Next i
    x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = s0
    s = MultiplicarDec(x(), n)
    x(1) = d: x(2) = "3": x(1) = PotenciasDec(x(), n): x(2) = "3"
    x(1) = DividirDec(x(), pr, n): x(2) = s2: x(1) = MultiplicarDec(x(), n): x(2) = s
    s = SumarDec(x(), n): x(1) = d: x(2) = "5": x(1) = PotenciasDec(x(), n): x(2) = "60"
    x(1) = DividirDec(x(), pr, n): x(2) = s4: x(1) = MultiplicarDec(x(), n): x(2) = s
    s = SumarDec(x(), n)
    If j = 1 Then sa = s Else sb = s
Next j
MST4D = PrecisionString(sa, sb)
End Function
' =====

Public Function MST6D(ByVal ax As String, ByVal bx As String, ByVal np As Integer) As String
Dim s0 As String, s2 As String, s4 As String, s6 As String, x(2) As String, nn(2) As String
Dim i As Integer, s As String, d As String, a1 As String, pr As Integer
Dim j As Integer, sa As String, sb As String, n As Integer, p As Integer
n = 7: pr = 24 : nn(1) = Mid$(Str$(Abs(np)), 2)
x(1) = nn(1): x(2) = "2": nn(2) = Multiplicar(x(), n)
' pr es la precisión en los cálculos con decimales
'---- Cálculo de la integral
For j = 1 To 2
    s0 = "0": s2 = "0": s4 = "0": s6 = "0": s = "0"
    x(1) = bx: x(2) = ax: x(1) = RestarDec(x(), n): x(2) = nn(j)
    d = DividirDec(x(), pr, n): p = Val(nn(j)) / 2
    x(1) = ax: x(2) = d: a1 = SumarDec(x(), n)
    For i = 1 To p
        x(1) = s0: x(2) = h(a1, pr): s0 = SumarDec(x(), n)
        x(1) = s2: x(2) = h2(a1, pr): s2 = SumarDec(x(), n)
        x(1) = s4: x(2) = h4(a1, pr): s4 = SumarDec(x(), n)
        x(1) = s6: x(2) = h6(a1, pr): s6 = SumarDec(x(), n)
        x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = a1: a1 = SumarDec(x(), n)
    Next i
    x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = s0
    s = MultiplicarDec(x(), n)
    x(1) = d: x(2) = "3": x(1) = PotenciasDec(x(), n): x(2) = "3"
    x(1) = DividirDec(x(), pr, n): x(2) = s2: x(1) = MultiplicarDec(x(), n): x(2) = s
    s = SumarDec(x(), n): x(1) = d: x(2) = "5": x(1) = PotenciasDec(x(), n): x(2) = "60"
    x(1) = DividirDec(x(), pr, n): x(2) = s4: x(1) = MultiplicarDec(x(), n): x(2) = s
    s = SumarDec(x(), n)
    If j = 1 Then sa = s Else sb = s
Next j
MST6D = PrecisionString(sa, sb)
End Function

```

```

s = SumarDec(x(), n): x(1) = d: x(2) = "7": x(1) = PotenciasDec(x(), n): x(2) = "2520"
x(1) = DividirDec(x(), pr, n): x(2) = s6: x(1) = MultiplicarDec(x(), n): x(2) = s
s = SumarDec(x(), n)
If j = 1 Then sa = s Else sb = s
Next j
MST6D = PrecisionString(sa, sb)
End Function
' =====
Public Function h(ByVal x0 As String, pr As Integer) As String
    Dim u As String, x(2) As String, n As Integer
    n = 7
    x(1) = x0: x(2) = x0: x(1) = MultiplicarDec(x(), n): x(2) = "-0.5"
    u = MultiplicarDec(x(), n)
    h = ExpP(u, pr)
End Function
' =====
Public Function h2(ByVal x0 As String, pr As Integer) As String
    Dim x(2) As String, u As String, v As String, n As Integer
    n = 7: x(1) = x0: x(2) = "2"
    x(1) = PotenciasDec(x(), n): x(2) = "1": u = RestarDec(x(), n)
    x(1) = x0: x(2) = x0: x(1) = MultiplicarDec(x(), n): x(2) = "-0.5"
    v = MultiplicarDec(x(), n): x(1) = u: x(2) = ExpP(v, 24)
    h2 = MultiplicarDec(x(), n)
End Function
' =====
Public Function h4(ByVal x0 As String, pr As Integer) As String
    Dim x(2) As String, u As String, v As String, n As Integer
    n = 7: x(1) = x0: x(2) = "4": u = PotenciasDec(x(), n)
    x(1) = x0: x(2) = "2": x(1) = PotenciasDec(x(), n): x(2) = "-6"
    x(1) = MultiplicarDec(x(), n): x(2) = "3": v = SumarDec(x(), n)
    x(1) = u: x(2) = v: u = SumarDec(x(), n)
    x(1) = x0: x(2) = x0: x(1) = MultiplicarDec(x(), n): x(2) = "-0.5"
    v = MultiplicarDec(x(), n): v = ExpP(v, 24)
    x(1) = u: x(2) = v
    h4 = MultiplicarDec(x(), n)
End Function
' =====
Public Function h6(ByVal x0 As String, pr As Integer) As String
    Dim x(2) As String, u As String, v As String, w As String, n As Integer
    n = 7: x(1) = x0: x(2) = "6": u = PotenciasDec(x(), n)
    x(1) = x0: x(2) = "4": x(1) = PotenciasDec(x(), n): x(2) = "-15"
    v = MultiplicarDec(x(), n): x(1) = x0: x(2) = "2": x(1) = PotenciasDec(x(), n): x(2) = "45"
    x(1) = MultiplicarDec(x(), n): x(2) = "-15": w = SumarDec(x(), n)
    x(1) = u: x(2) = v: x(1) = SumarDec(x(), n): x(2) = w: u = SumarDec(x(), n)
    x(1) = x0: x(2) = x0: x(1) = MultiplicarDec(x(), n): x(2) = "-0.5"
    v = MultiplicarDec(x(), n): x(1) = ExpP(v, 24): x(2) = u
    h6 = MultiplicarDec(x(), n)
End Function

```

Las tablas siguientes, dan una idea de las posibilidades de cálculo con las funciones MST2D, MST4D y MS&D.

Función	Nº de las divisiones	Valor de la integral
MST2D	8	1.1962
	16	1.196288
	32	1.1962880
	64	1.19628801
	128	1.196288013

Función	Nº de las divisiones	Valor de la integral
MST4D	8	1.19628
	16	1.19628801
	32	1.196288013
	64	1.196288013322
	128	1.19628801332260

Función	Nº de las divisiones	Valor de la integral
MST6D	8	1.196288013
	16	1.19628801332
	32	1.1962880133226
	64	1.1962880133226082
	128	1.19628801332260820

Si en la función MST6D $pr = 24$ y al intervalo $[0,2]$ se divide en 1024 partes iguales, la integral definida de la función (5) sobre este intervalo es:

$$I = 1.196288013322608202931 \quad (7)$$

, donde hay 21 cifras después del punto decimal (imposible de obtener con los métodos convencionales, que trabajan en la aritmética de punto flotante).

Hasta ahora, para evaluar la precisión del resultado no se ha utilizado la fórmula (4). La precisión se evaluó calculando el valor de la integral para dos divisiones del intervalo, donde la segunda división tenía el doble de puntos que la primera y reteniendo del primer valor de la integral la sección izquierda máxima que figura encaja enteramente en el segundo valor. Por ejemplo, si se calcula el valor de la integral (1) de la función (5) en el intervalo $[0,2]$ (utilizando la función MST4C), dividiendo el intervalo en 8 y luego en 16 partes iguales, se obtienen los valores siguientes para el integral:

$$I_1 = 1.1962878898\dots$$

$$I_2 = 1.1962880114\dots$$

Se observa que la sección izquierda máxima de I_1 que se encuentra enteramente en I_2 , es $1.19628 \approx I$. Se podría proceder de otra manera también, calculando la misma integral solo para 8 divisiones y evaluando una cota superior del error absoluto del error del cálculo utilizando la fórmula (4). El programa siguiente calcula la integral anterior dividiendo el intervalo en 8 partes iguales y determina con la fórmula (4) la cota superior del error absoluto de los cálculos:

```
Public Function MST4CC(ByVal ax As String, ByVal bx As String, ByVal np As Integer) As String
    Dim s0 As String, s2 As String, s4 As String, p As Integer, x(2) As String
    Dim i As Integer, s As String, d As String, a1 As String, u As String
    Dim k As Integer, sa As String, sb As String, n As Integer, u As String
    Dim nps As String, cota As String, v As String, pr As Integer
    Dim dif0 As String, dif1 As String, dif2 As String
    ' pr precisión en los calculos con decimales
    '---- Cálculo de la integral
```

```

n = 7: pr = 16: nps = Mid$(Str$(Abs(np)), 2)
s0 = "0": s2 = "0": s4 = "0": s = "0"
x(1) = bx: x(2) = ax: dif0 = RestarDec(x(), n): x(1) = dif0: x(2) = nps
d = DividirDec(x(), pr, n): p = np / 2
x(1) = ax: x(2) = d: a1 = SumarDec(x(), n)
For i = 1 To p
    x(1) = s0: x(2) = g(a1, pr): s0 = SumarDec(x(), n)
    x(1) = s2: x(2) = g2(a1): s2 = SumarDec(x(), n)
    x(1) = s4: x(2) = g4(a1): s4 = SumarDec(x(), n)
    x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = a1: a1 = SumarDec(x(), n)
Next i
x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = s0
s = MultiplicarDec(x(), n)
x(1) = d: x(2) = "3": x(1) = PotenciasDec(x(), n): x(2) = "3"
x(1) = DividirDec(x(), pr, n): x(2) = s2: x(1) = MultiplicarDec(x(), n): x(2) = s
s = SumarDec(x(), n): x(1) = d: x(2) = "5": x(1) = PotenciasDec(x(), n): x(2) = "60"
x(1) = DividirDec(x(), pr, n): x(2) = s4: x(1) = MultiplicarDec(x(), n): x(2) = s
sa = SumarDec(x(), n)
'Cota superior del valor absoluto de la 5a derivada en el intervalo(ax,bx)
x(1) = bx: x(2) = ax: x(1) = RestarDec(x(), n): x(2) = "16": dif1 = DividirDec(x(), 2, n)
' En las extremidades del intervalo la derivada puede no existir!
x(1) = ax: x(2) = dif1: u = SumarDec(x(), n)
cota = g5(u)
If Left$(cota, 1) = "-" Then cota = Mid$(cota, 2)
For i = 1 To 14
    x(1) = u: x(2) = dif1: u = SumarDec(x(), n)
    v = g5(u)
    If Left$(v, 1) = "-" Then v = Mid$(v, 2)
    x(1) = cota: x(2) = v: dif2 = RestarDec(x(), n)
    If Left$(dif2, 1) = "-" Then cota = v
Next i
'Precisión de la integral
x(1) = dif0: x(2) = "6": x(1) = PotenciasDec(x(), n): x(2) = cota
x(1) = MultiplicarDec(x(), n): x(2) = Factorial(6, n): u = DividirDec(x(), 24, n)
x(1) = np: x(2) = "6": v = PotenciasDec(x(), n): x(1) = u: x(2) = v: u = DividirDec(x(), 24, n)
x(1) = sa: x(2) = u
If Left$(sa, 1) <> "-" Then sb = SumarDec(x(), n) Else sb = RestarDec(x(), n)
k = 1
Do
    If Left$(sa, k) = Left$(sb, k) Then
        k = k + 1
    Else
        s = Left$(sa, k - 1)
    Exit Do
End If
Loop
If s = "" Or Len(s) = 1 Then s = Int(Val(sa))
MST4CC = s
End Function
=====

Public Function g5(ByVal x0 As String) As String
    Dim h As String, sx As String, x(2) As String
    Dim b(5) As String, n As Integer, i As Integer, pr As Integer
    n = 7: h = "0.0000001": pr = 5 * (Len(h) - 2) + 16
    b(0) = x0
    For i = 1 To 5
        x(1) = b(i - 1): x(2) = h: b(i) = SumarDec(x(), n)
    Next i
    x(1) = g(b(5), pr): x(2) = g(b(0), pr): sx = RestarDec(x(), n)
    x(1) = g(b(1), pr): x(2) = g(b(4), pr): x(1) = RestarDec(x(), n): x(2) = "5"
    x(1) = MultiplicarDec(x(), n): x(2) = sx: sx = SumarDec(x(), n)
    x(1) = g(b(3), pr): x(2) = g(b(2), pr): x(1) = RestarDec(x(), n): x(2) = "10"

```

```

x(1) = MultiplicarDec(x(), n): x(2) = sx: sx = SumarDec(x(), n)
x(1) = h: x(2) = "5" ' x(2) = sx: sx = SumarDec(x(), n):
x(2) = PotenciasDec(x(), n): x(1) = sx
g5 = DividirDec(x(), 16, n)
End Function

```

Utilizando la función MST4CC se obtiene que $I_1 = 1.1962878898 \dots |R| \leq 0.000001481\$61\dots$

Puesto que sumando a I_1 el valor máximo de $|R|$ las primeras 4 decimales de I_1 no quedan afectadas, se puede considerar que el valor aproximado de la integral es $I = 1.1962$ (Si I_1 fuera negativo se debería restar de I_1 el valor máximo de $|R|$ y ver la parte no afectada de I_1). Se observa que la segunda manera de trabajar conduce a un decimal exacto menos en el resultado, aunque el cálculo ha necesitado un poco menos de tiempo. Si la cota del valor absoluto de la 5^a derivada es inmejorable, la pérdida de un decimal exacto en el resultado se debe a que la estimación del error absoluto del cálculo por la fórmula (4) es aceptable pero probablemente mejorable.

Tal como se ha escrito la función MST4CC, a continuación se presenta las funciones MST6CC y MST6DD:

```

Public Function MST6CC(ByVal ax As String, ByVal bx As String, ByVal np As Integer) As String
    Dim s0 As String, s2 As String, s4 As String, s6 As String, x(2) As String
    Dim i As Integer, s As String, d As String, a1 As String, p As Integer
    Dim j As Integer, sa As String, sb As String, n As Integer, k As Integer
    Dim nps As String, cota As String, u As String, v As String, pr As Integer
    Dim dif0 As String, dif1 As String, dif2 As String
    ' pr precisión en los calculos con decimales
    '---- Cálculo de la integral
    n = 7: pr = 24: nps = Mid$(Str$(Abs(np)), 2)
    s0 = "0": s2 = "0": s4 = "0": s6 = "0": s = "0"
    If Left$(nps, 1) = " " Then nps = Mid$(nps, 2)
    x(1) = bx: x(2) = ax: dif0 = RestarDec(x(), n): x(1) = dif0: x(2) = nps
    d = DividirDec(x(), pr, n): p = np / 2
    x(1) = ax: x(2) = d: a1 = SumarDec(x(), n)
    For i = 1 To p
        x(1) = s0: x(2) = g(a1, pr): s0 = SumarDec(x(), n)
        x(1) = s2: x(2) = g2(a1): s2 = SumarDec(x(), n)
        x(1) = s4: x(2) = g4(a1): s4 = SumarDec(x(), n)
        x(1) = s6: x(2) = g6(a1): s6 = SumarDec(x(), n)
        x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = a1: a1 = SumarDec(x(), n)
    Next i
    x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = s0
    s = MultiplicarDec(x(), n)
    x(1) = d: x(2) = "3": x(1) = PotenciasDec(x(), n): x(2) = "3"
    x(1) = DividirDec(x(), pr, n): x(2) = s2: x(1) = MultiplicarDec(x(), n): x(2) = s
    s = SumarDec(x(), n): x(1) = d: x(2) = "5": x(1) = PotenciasDec(x(), n): x(2) = "60"
    x(1) = DividirDec(x(), pr, n): x(2) = s4: x(1) = MultiplicarDec(x(), n): x(2) = s
    s = SumarDec(x(), n): x(1) = d: x(2) = "7": x(1) = PotenciasDec(x(), n): x(2) = "2520"
    x(1) = DividirDec(x(), pr, n): x(2) = s6: x(1) = MultiplicarDec(x(), n): x(2) = s
    sa = SumarDec(x(), n)
    'Cota superior del valor absoluto de la 7a derivada en el intervalo(ax,bx)
    x(1) = bx: x(2) = ax: x(1) = RestarDec(x(), n): x(2) = "16": dif1 = DividirDec(x(), 2, n)
    ' ¡En las extremidades del intervalo la derivada puede no existir!
    x(1) = ax: x(2) = dif1: u = SumarDec(x(), n)
    cota = g7(u)
    If Left$(cota, 1) = "-" Then cota = Mid$(cota, 2)
    For i = 1 To 14

```

```

x(1) = u: x(2) = dif1: u = SumarDec(x(), n)
v = g7(u)
If Left$(v, 1) = "-" Then v = Mid$(v, 2)
x(1) = cota: x(2) = v: dif2 = RestarDec(x(), n)
If Left$(dif2, 1) = "-" Then cota = v
Next i
'Precisión de la integral
x(1) = dif0: x(2) = "8": x(1) = PotenciasDec(x(), n): x(2) = cota
x(1) = MultiplicarDec(x(), n): x(2) = Factorial(8, n): u = DividirDec(x(), 24, n)
x(1) = np: x(2) = "7": v = PotenciasDec(x(), n): x(1) = u: x(2) = v: u = DividirDec(x(), 24, n)
x(1) = sa: x(2) = u
If Left$(sa, 1) <> "-" Then sb = SumarDec(x(), n) Else sb = RestarDec(x(), n)
k = 1
Do
    If Left$(sa, k) = Left$(sb, k) Then
        k = k + 1
    Else
        s = Left$(sa, k - 1)
        Exit Do
    End If
Loop
If s = "" Or Len(s) = 1 Then s = Int(Val(sa))
MST6CC = s
End Function
'=====

Public Function g7(ByVal x0 As String) As String
    Dim h As String, sx As String, x(2) As String
    Dim b(7) As String, n As Integer, i As Integer, pr As Integer
    n = 7: h = "0.0000001": pr = 5 * (Len(h) - 2) + 16
    b(0) = x0
    For i = 1 To 7
        x(1) = b(i - 1): x(2) = h: b(i) = SumarDec(x(), n)
    Next i
    x(1) = g(b(7), pr): x(2) = g(b(0), pr): sx = RestarDec(x(), n)
    x(1) = g(b(1), pr): x(2) = g(b(6), pr): x(1) = RestarDec(x(), n): x(2) = "7"
    x(1) = MultiplicarDec(x(), n): x(2) = sx: sx = SumarDec(x(), n)
    x(1) = g(b(5), pr): x(2) = g(b(2), pr): x(1) = RestarDec(x(), n): x(2) = "21"
    x(1) = MultiplicarDec(x(), n): x(2) = sx: sx = SumarDec(x(), n)
    x(1) = g(b(3), pr): x(2) = g(b(4), pr): x(1) = RestarDec(x(), n): x(2) = "35"
    x(1) = MultiplicarDec(x(), n): x(2) = sx: sx = SumarDec(x(), n)
    x(1) = h: x(2) = "7" ' x(2) = sx: sx = SumarDec(x(), n):
    x(2) = PotenciasDec(x(), n): x(1) = sx
    g7 = DividirDec(x(), 16, n)
End Function

```

Nº de las divisiones:	MST4CC	MST6CC
128	1.1962880133226	1.19628801332260

```

Public Function MST6DD(ByVal ax As String, ByVal bx As String, ByVal np As Integer) As String
    Dim s0 As String, s2 As String, s4 As String, s6 As String, x(2) As String
    Dim i As Integer, s As String, d As String, a1 As String, pr As Integer
    Dim sa As String, sb As String, n As Integer, p As Integer, nps As String
    Dim dif0 As String, dif1 As String, dif2 As String
    Dim u As String, v As String, cota As String, k As Integer
    n = 7: pr = 24: nps = Mid$(Str$(Abs(np)), 2)
    If Left$(nps, 1) = " " Then nps = Mid$(nps, 2)
    'pr es la precisión en los calculos con decimales
    '---- Cálculo de la integral
    s0 = "0": s2 = "0": s4 = "0": s6 = "0": s = "0"
    x(1) = bx: x(2) = ax: dif0 = RestarDec(x(), n): x(1) = dif0: x(2) = nps

```

```

d = DividirDec(x(), pr, n): p = np / 2
x(1) = ax: x(2) = d: a1 = SumarDec(x(), n)
For i = 1 To p
    x(1) = s0: x(2) = h(a1, pr): s0 = SumarDec(x(), n)
    x(1) = s2: x(2) = h2(a1, pr): s2 = SumarDec(x(), n)
    x(1) = s4: x(2) = h4(a1, pr): s4 = SumarDec(x(), n)
    x(1) = s6: x(2) = h6(a1, pr): s6 = SumarDec(x(), n)
    x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = a1: a1 = SumarDec(x(), n)
Next i
x(1) = "2": x(2) = d: x(1) = MultiplicarDec(x(), n): x(2) = s0
s = MultiplicarDec(x(), n)
x(1) = d: x(2) = "3": x(1) = PotenciasDec(x(), n): x(2) = "3"
x(1) = DividirDec(x(), pr, n): x(2) = s2: x(1) = MultiplicarDec(x(), n): x(2) = s
s = SumarDec(x(), n): x(1) = d: x(2) = "5": x(1) = PotenciasDec(x(), n): x(2) = "60"
x(1) = DividirDec(x(), pr, n): x(2) = s4: x(1) = MultiplicarDec(x(), n): x(2) = s
s = SumarDec(x(), n): x(1) = d: x(2) = "7": x(1) = PotenciasDec(x(), n): x(2) = "2520"
x(1) = DividirDec(x(), pr, n): x(2) = s6: x(1) = MultiplicarDec(x(), n): x(2) = s
sa = SumarDec(x(), n)
'Cota superior del valor absoluto de la 7a derivada en el intervalo(ax,bx)
x(1) = bx: x(2) = ax: x(1) = RestarDec(x(), n): x(2) = "16": dif1 = DividirDec(x(), 2, n)
' ¡En las extremidades del intervalo la derivada puede no existir!
x(1) = ax: x(2) = dif1: u = SumarDec(x(), n)
cota = h7(u, 36)
If Left$(cota, 1) = "-" Then cota = Mid$(cota, 2)
For i = 1 To 14
    x(1) = u: x(2) = dif1: u = SumarDec(x(), n)
    v = h7(u, 24)
    If Left$(v, 1) = "-" Then v = Mid$(v, 2)
    x(1) = cota: x(2) = v: dif2 = RestarDec(x(), n)
    If Left$(dif2, 1) = "-" Then cota = v
Next i
'Precisión de la integral
x(1) = dif0: x(2) = "8": x(1) = PotenciasDec(x(), n): x(2) = cota
x(1) = MultiplicarDec(x(), n): x(2) = Factorial(8, n): u = DividirDec(x(), 24, n)
x(1) = np: x(2) = "7": v = PotenciasDec(x(), n): x(1) = u: x(2) = v: u = DividirDec(x(), 24, n)
x(1) = sa: x(2) = u
If Left$(sa, 1) <> "-" Then sb = SumarDec(x(), n) Else sb = RestarDec(x(), n)
k = 1
Do
    If Left$(sa, k) = Left$(sb, k) Then
        k = k + 1
    Else
        s = Left$(sa, k - 1)
        Exit Do
    End If
Loop
If s = "" Or Len(s) = 1 Then s = Int(Val(sa))
MST6DD = PrecisionString(sa, sb)
End Function
=====

Public Function h7(ByVal x0 As String, pr As Integer) As String
    Dim x(2) As String, u As String, v As String, n As Integer
    n = 7: x(1) = x0: x(2) = "7": u = PotenciasDec(x(), n)
    If Left$(u, 1) = "-" Then u = Mid$(u, 2) Else u = u + "_"
    x(1) = x0: x(2) = "5": x(1) = PotenciasDec(x(), n): x(2) = "21"
    v = MultiplicarDec(x(), n): x(1) = x0: x(2) = "3": x(1) = PotenciasDec(x(), n): x(2) = "-108"
    x(1) = MultiplicarDec(x(), n): x(2) = v: v = SumarDec(x(), n): x(1) = u: x(2) = v: u = SumarDec(x(), n)
    x(1) = x0: x(2) = "111": x(1) = MultiplicarDec(x(), n): x(2) = u: u = SumarDec(x(), n)
    x(1) = x0: x(2) = "x0": x(1) = MultiplicarDec(x(), n): x(2) = "-0.5"
    v = MultiplicarDec(x(), n): x(1) = ExpP(v, 24): x(2) = u
    h7 = MultiplicarDec(x(), n)
End Function

```

Nº de las divisiones:	MST6DD
128	1.196288013322608
256	1.19628801332260820
512	1.1962880133226082029
1024	1.196288013322608202931

Las funciones MST4CC y MST6CC son lentas, sobre todo si se pretende más precisión (el intervalo se divide en muchas partes). En los superordenadores del futuro seguramente van a parecer más rápidas.

La función MST6DD es más rápida y permite obtener resultados precisión grande, sin embargo para utilizarla hay que calcular las derivadas hasta el orden 7 (de manera manual),, que a veces no cuesta tanto.

Advertencia: Para utilizar las funciones MST2C, MST4C, MST6C, Mst6CC, MST2D, MST4D, MST6D y MST6DD es necesario cargar en un módulo las funciones para operar con enteros y decimales largos y las funciones elementales de precisión grande [1]-[3]

Bibliografía:

- [1] A. Peter Santha, Cálculos con números enteros largos en ordenadores, Monografías.com
- [2] A. Peter Santha, Cálculos con números decimales largos en ordenadores, Monografías.com
- [3] A. Peter Santha, Cálculo de los valores de las funciones elementales con precisión grande, Monografías.com
- [4] A. Peter Santha, Cálculo aproximado de las derivadas y derivadas parciales de orden superior, Monografías.com
- [5] B. DÉMIDOVITCH et I. MARON, ÉLÉMENTS DE CALCUL NUMÉRIQUE, ÉDITIONS MIR, MOCOU, 1973