Problem Statement

Write a VBA program to compute the sine of an angle (without using the built-in Sine function).

Use a main program and two user-defined Function subprograms. In the main program, promote user in enter an angle, x, in degrees to use in calculating Sin(x). Use a user-defined Function to calculate the sine function. Use a user-defined Function to calculate a Factorial, N!

Theory

1. The value of sine can be estimated using the following equation

\[
\sin x \approx \sum_{i=1}^{N} (-1)^{i+1} \left[ \frac{x^{2i-1}}{(2i-1)!} \right]
\]

\[
= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots + (-1)^{N+1} \left[ \frac{x^{2N-1}}{(2N-1)!} \right]
\]

2. \(2\pi\) (radians) = 360 (degrees), so 1 radian = \((180/\pi)\) degrees

3. tan \((\pi/4)\) = 1, then \(\pi = 4 \cdot \arctan(1)\)

4. The factorial can be found by multiplying all integers from 1 to N

\[N! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot N\] (N is integer greater than 1)

Assumption

Assume numbers after the last term are small enough to be neglected.

Solution

Private Sub CommandButton1_Click()

'Qiongyu Sun

'sine program

Dim x As Single, sine As Single, y As Single, computed As Single, last As Double
x = InputBox("please enter an angle in degrees")

last = InputBox("please enter a maximum value of the last term in the estimate")

y = x * 4 * Atn(1) / 180

sine = series(y, last)

computed = Sin(y)

MsgBox ("For an angle of " & x & " degrees " & vbCrLf &
& "The estimated sine value " & "is " & Format(sine, "0.000") & vbCrLf &
"Sine value calculated using the VBA built-in function is " & Format(computed, "0.000"))

End Sub

Function series(ByVal a As Single, ByVal max As Double) As Single

Dim term As Double, i As Double, m As Double

series = 0

i = 0

Do

i = i + 1

m = 2 * i - 1

term = (-1) ^ (i + 1) * ((a ^ m) / Fact(m))

series = series + term

If (Abs(term) < max) Then Exit Do

Loop

End Function

Function Fact(ByVal n As Integer) As Double

Dim p As Integer

p = 1

Fact = 1
For p = 1 To n
Fact = Fact * p
Next p
End Function
start

enter x  \[ y = x \times 4 \times \text{Atn}(1) / 180 \]  
Input a maximum value of the
last term in the estimate-last

enter last

\[ y = x \times 4 \times \text{Atn}(1) / 180 \]  
Convert angle in degrees, x, into angle in radians, y.

\[ \text{sine} = \text{series}(y, \text{last}) \]  
Compute sine value using defined function in subprograms

\[ \text{computed} = \sin(x) \]  
compute the sin value using VBA built-in function

output sine and computed

stop
start

series = 0
i = 0

enter a

enter max

\[ i = i + 1 \]
\[ m = 2 \cdot i - 1 \]

\[ \text{term} = (-1)^{i+1} \cdot (a^m) / \text{fact}(m) \]

series = series + term

Is |term| < max

Yes

output series

stop
Discussion and Conclusion

In this problem, we build a function to compute sine value instead of using the VBA built-in function. The equation actually contains infinite terms, but because \( \frac{x^{2i-1}}{(2i-1)!} \) is decreasing as \( i \) value gets larger, the numbers after some value are small enough to be neglected. Thus, we need to set a maximum value of the last term in the estimate, which should be small enough to make the result close enough to the exact sine value. Another thing we need pay attention to is that some number involved in this program contain more digit than “single” can express, then we have to dim them as double. Or else it will cause the error of “overflow”. After calculating the sine value using the VBA built-in function, I compared the two sine values using different angles, and found they are really close. This shows the way we use to estimate is very precise, and the maximum value we choose for the last term is small enough. I find using user-defined Function subprogram can make the main program clearer and brief.

Good job.