Numerical Differentiator and Integrator

```
Open "numerical_calculus_output.txt" For Output As #1
Screen 12
Dim Shared DEFAULT_NUMERICAL_DERIVATIVE_STEP##
Dim Shared DEFAULT_N_TRAPEZOID%
Dim Shared DEFAULT_N_SIMPSON13%
Dim Shared DEFAULT_N_SIMPSON38%
DEFAULT_NUMERICAL_DERIVATIVE_STEP## = 1E-7
DEFAULT_N_TRAPEZOID% = 100
DEFAULT_N_SIMPSON13% = 100
DEFAULT_N_SIMPSON38% = 51

Dim Shared FUNCTION_NUMBER%     ' Function used all over the program
FUNCTION_NUMBER% = 1

' START: Main Driver code
pl "..............  Numerical Differentiator and Integrator ..........."
main:
lb
FUNCTION_NUMBER% = in_int%(" -> Enter Function Number (as defined in program): ")
mode% = in_int%(" -> Enter mode (0: Differentiation, 1: Integration): ")
Select Case mode%
   Case 0
      main_differentiate
   Case 1
      main_integrate
   Case Else
      pl "ERROR: Invalid mode " + s$(mode%)
End Select
GoTo main
' END: Main Driver code

Sub main_differentiate
   x## = in_float64##(" -> Differentiation Domain Point: ")
   diff_step## = in_float64##(" -> Differentiation Step (0 for default): ")
   If diff_step## = 0 Then
      diff_step## = DEFAULT_NUMERICAL_DERIVATIVE_STEP##
   End If
   lb
   pl " Derivative at " + s$(x##) + " : " + s$(numericalDerivative##(x##, diff_step##))
End Sub

Sub main_integrate
   a## = in_float64##(" -> Integration Lower Limit: ")
   b## = in_float64##(" -> Integration Upper Limit: ")
   integration_intervals% = in_int%(" -> Integration Interval Count (0 for default): ")
   lb
   pl "RESULTS "
   pl "  Trapezoid: " + s$(trapezoid##(a##, b##, integration_intervals%))
   pl "  Simpson 1/3: " + s$(simpson13##(a##, b##, integration_intervals%))
   pl "  Simpson 3/8: " + s$(simpson38##(a##, b##, integration_intervals%))
End Sub

' Function Definitions
Function f1## (x##)
   f1## = x## - (3 ^ 0.5)
End Function

Function f2## (x##)
   f2## = (x## * x## * x##) - (x## * x##) + 1
End Function

Function f3## (x##)
   f3## = (x## * Exp(x##)) - Cos(x##)
End Function
Function f4## (x##)
   f4## = (x## * x## * x##) - Sin(x##) + 4
End Function

Function f5## (x##)
   f5## = (x## * Exp(x##)) - 1
End Function

Function f## (x##) ' Function used all over the program
   Select Case FUNCTION_NUMBER%
      Case 2
         f## = f2##(x##)
      Case 3
         f## = f3##(x##)
      Case 4
         f## = f4##(x##)
      Case 5
         f## = f5##(x##)
      Case Else
         f## = f1##(x##)
   End Select
End Function

' ......... Differentiation ...........
Function numericalDerivative## (x##, h##)
   numericalDerivative## = (f##(x## + h##) - f##(x##)) / h##
End Function

Function numericalDerivativeDef## (x##)
```

```
    numericalDerivativeDef## = numericalDerivative##(x##,
DEFAULT_NUMERICAL_DERIVATIVE_STEP##)
End Function

' ..... Integration .....
Function trapezoid## (a##, b##, n%)
    range## = b## - a##
    If range## = 0 Then
        trapezoid## = 0
        Exit Function
    End If

    If n% < 2 Then
        n% = DEFAULT_N_TRAPEZOID%
    End If

    h## = range## / n%
    res## = f##(a##)
    res## = res## + f##(b##)

    For i% = 1 To n% - 1
        res## = res## + (2 * f##(a## + (i% * h##)))
    Next i%
    trapezoid## = res## * (h## / 2)
End Function

Function simpson13## (a##, b##, n%)
    range## = b## - a##
    If range## = 0 Then
        simpson13## = 0
        Exit Function
    End If

    If n% < 2 Then
        n% = DEFAULT_N_SIMPSON13%
    End If

    If n% Mod 2 <> 0 Then
        n% = n% + 1 ' Must be even
    End If

    h## = range## / n%
    res## = f##(a##)
    res## = res## + f##(b##)

    For i% = 1 To n% - 1
        If (i% Mod 2) = 0 Then
            m% = 2
        Else
            m% = 4
        End If
        res## = res## + (m% * f##(a## + (i% * h##)))
    Next i%
    simpson13## = res## * (h## / 3)
End Function

Function simpson38## (a##, b##, n%)
    range## = b## - a##
    If range## = 0 Then
        simpson38## = 0
        Exit Function
    End If

    If n% < 2 Then
        n% = DEFAULT_N_SIMPSON38%
    End If

    r% = n% Mod 3
    If r% <> 0 Then
        n% = n% + (3 - r%) ' Must be multiple of 3
    End If

    h## = range## / n%
    res## = f##(a##)
    res## = res## + f##(b##)

    For i% = 1 To n% - 1
        If (i% Mod 3) = 0 Then
            m% = 2
        Else
            m% = 3
        End If
        res## = res## + (m% * f##(a## + (i% * h##)))
    Next i%
    simpson38## = res## * h## * (3 / 8)
End Function

' ..... Formatting .....
Sub p (st$) ' Print and log a given string WITHOUT line break
    Print st$;
    Print #1, st$;
End Sub

Sub lb ' Print and log a line break
    Print
    Print #1, ""
End Sub
```

```
Sub pl (st$) ' Print and log given string WITH line break
   Print st$
   Print #1, st$
End Sub

Function s$ (i##)
   s$ = LTrim$(RTrim$(Str$(i##)))
End Function

'.............. INPUT .............
Function in_str$ (caption$)
   p (caption$)
   Input "", v$
   Print #1, v$
   in_str$ = v$
End Function

Function in_int% (caption$)
   p (caption$)
   Input "", v%
   Print #1, s$(v%)
   in_int% = v%
End Function

Function in_float64## (caption$)
   p (caption$)
   Input "", v##
   Print #1, s$(v##)
   in_float64## = v##
End Function
```

OUTPUT

EXAMPLE 1: f(x) = x - sqrt(3)

-> Enter Function Number (as defined in program): 1
-> Enter mode (0: Differentiation, 1: Integration): 0
-> Differentiation Domain Point: 5
-> Differentiation Step (0 for default): 0

Derivative at 5 : .9999999999983981

EXAMPLE 2: f(x) = x^3 - x^2 - 1

-> Enter Function Number (as defined in program): 2
-> Enter mode (0: Differentiation, 1: Integration): 1
-> Integration Lower Limit: 0
-> Integration Upper Limit: 10
-> Integration Interval Count (0 for default): 0

RESULTS

Trapezoid: 2176.9
Simpson 1/3: 2176.666666666667
Simpson 3/8: 2176.666666666667

EXAMPLE 3: f(x) = x * exp(x) - 1

-> Enter Function Number (as defined in program): 3
-> Enter mode (0: Differentiation, 1: Integration): 1
-> Integration Lower Limit: -10
-> Integration Upper Limit: 10
-> Integration Interval Count (0 for default): 0

RESULTS

Trapezoid: 199046.2785244964
Simpson 1/3: 198241.8120663023
Simpson 3/8: 198244.5161341935

EXAMPLE 4: f(x) = x ^ 3 - sin(x) + 4

-> Enter Function Number (as defined in program): 4
-> Enter mode (0: Differentiation, 1: Integration): 0
-> Differentiation Domain Point: -100
-> Differentiation Step (0 for default): .000001

Derivative at -100 : 29999.13738148052