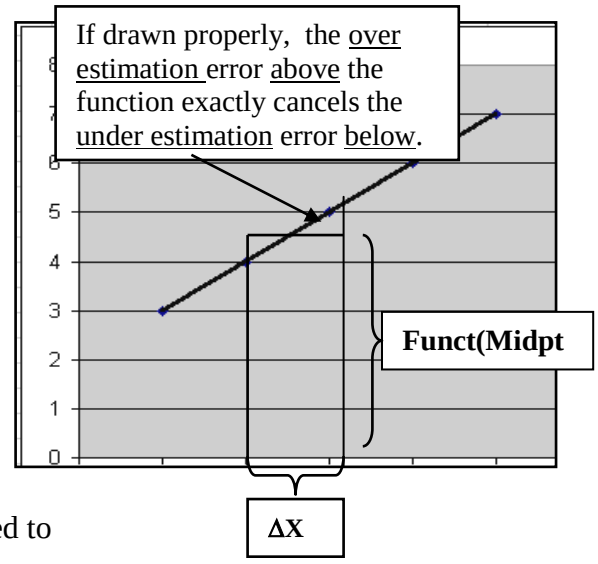
CHAPTER # 8 ITERATIVE TECHNIQUES APPLIED TO NUMERICAL INTEGRATION

This chapter continues the examination of iterative methods. In the most general sense, an iterative method is one which repeats the same steps or sequence of instructions over and over. From this point of view, the loops examined in Chapter 3, and any loop moreover, are all examples of iterative methods. The incremental iteration of Chapter 6 represents more of what engineers and scientists conceive of as a "true" iterative method. As described in Chapter 6 we can generalize the typical applications of incremental iteration into two groups : i) to track the change of state of some system, or, to borrow from kinematics terminology, to track its trajectory by allowing it to move through a series of incremental changes, and ii) to build something up as a sum of a group of contributing components where the whole is equal to the sum of the parts (referred to as a "chop & sum" method in class). In this chapter the second type of iterative analysis will be used to determine electric and gravitational vector field values, and scalar electric potential field values. The analysis is based a knowledge of the theoretical behaviour of the system on an incremental level; if this is not known the analysis cannot be applied.

8.1 Numerical Integration

Integration is simply a means of determining the area under a function by imagining the area to be divided into a series of incremental areas **f(x) Δx** and then summing up all the contributions[1]. The "limits" of the integral (or summation) represent the physical extent of the area[2]. The approximation being made is that over sufficiently small **Δx** the product **f(x) Δx** is exactly the actual area (where **x** is the value at the midpoint of interval **Δx**). This will be true if **Δx** is small enough that the function appears to be *linear* so that the over estimation error is cancelled out by the under estimation error. The value of **f(x)** is thus assumed to be constant at its midpoint value over the interval. This same approximation was made in Chapter 6 where it was assumed that the acceleration was constant which allowed the kinematics equations to be applied. In actual fact, the variation of the acceleration was essentially linear over the small **Δt** which led to errors that cancelled out as here.



Programming the summation of the **f(x) Δx** values is easily done with a running total as shown in **Example # 1** that follows. A second running total is used to keep track of the midpoint values of **x**.

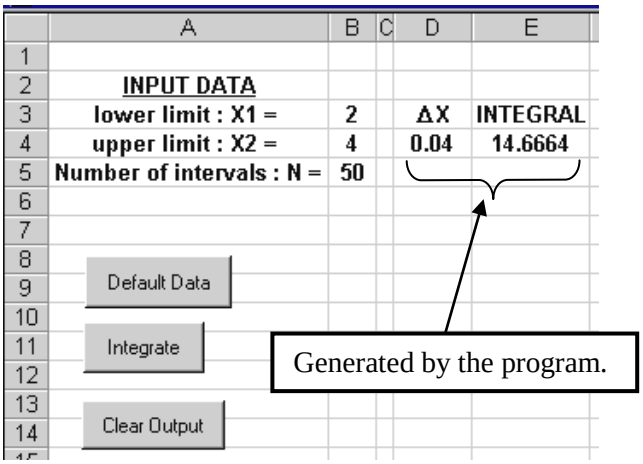
---

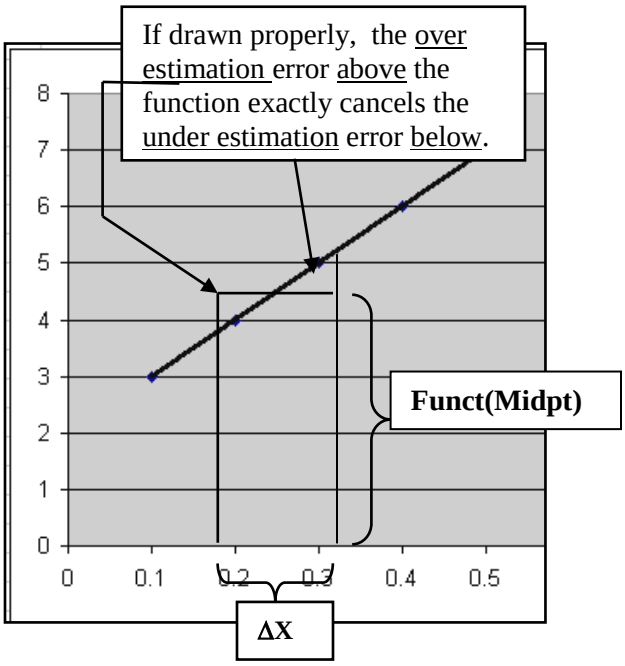
[1] Recall that the symbol for summing up a number of discrete items is the Cyrillic **Σ** which is equivalent to capital S (for "Sum") in our Roman alphabet. An integral sign $\int$, which is the symbol for summing up continuous values, is simply a stretched "S" (again representing "Sum").

[2] Note that the term "limit" has an entirely different connotation here where it describes the physical dimensions, extent, or "limits" of the situation. The other use of the word "limit" is to describe the "limiting value" of something, which may represent its value as something is made small, such as "the limit as Δx approaches 0", or which might represent the value of an expression as a certain value of the variable is approached : the value of **f(x)** as **x** approaches 2.

Example # 1 : Numerical Integration.

This exercise demonstrates how a function can be integrated numerically between two points (limits). The interval between the limits is divided into N **elements** (pieces) each of width **ΔX** ; the area contribution of each element is calculated using the function value at the midpoint of the element so that **dArea = Funct(Midpt) * ΔX** . A loop is used to add up all the incremental area contributions.






```
Option Explicit

Private Sub cmdIntegrate_Click()
Dim N As Integer, I As Integer, Sum As Double, Midpt As Double
Dim X1 As Double, X2 As Double, dArea As Double, deltaX As Double

X1 = Range("B3")
X2 = Range("B4")
N = Range("B5")

deltaX = (X2 - X1) / N
Midpt = X1 + (deltaX / 2)

I = 1
Sum = 0

Do
    dArea = Funct(Midpt) * deltaX
    Sum = Sum + dArea
    Midpt = Midpt + deltaX
    I = I + 1
Loop Until I = N + 1   ' doesn't actually do the (N + 1)-th loop

Range("D4") = deltaX
Range("E4") = Sum

End Sub
```

The 1st interval midpoint occurs is located at **ΔX / 2** to the right of the lower limit

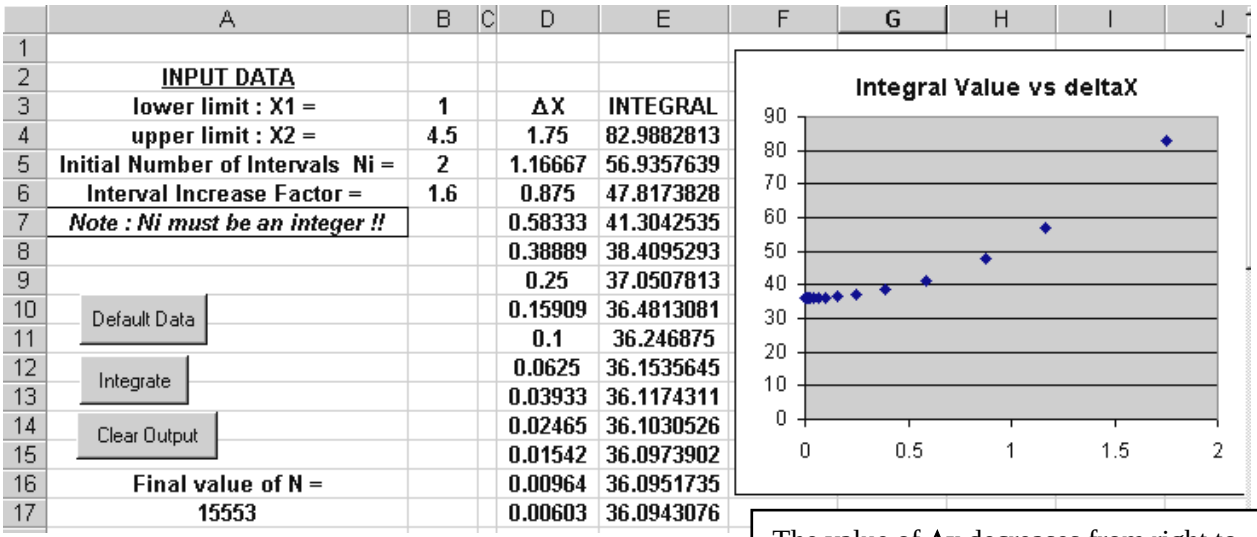
Note the two running totals.

Subsequent interval midpoints are separated by **ΔX** from the previous midpoint.

```
Function Funct(X)
Funct = X ^ 2 - 2 * X + 4
End Function

Private Sub cmdClear_Click()
Range("E4:D4").ClearContents
End Sub

Private Sub cmdDefault_Click()
Range("B3") = 2
Range("B4") = 4
Range("B5") = 50
End Sub
```

Function to be integrated.

Example # 2 : Convergence of Numerical Integrals. The convergence of the integral of Exercise # **1** is displayed graphically in this exercise as the size of $\Delta X \rightarrow 0$.



The value of **Δx** decreases from right to left. The limiting value is on the Y axis.

Modify the code of Exercise # 1as indicated below (new/ modified code is in **bold**).

Option Explicit

Function Funct(X)
Funct = -10 + 20 * X + 30 * X ^ 2 - 10 * X ^ 3
End Function

Private Sub cmdClear_Click()
*Range("E4:D**23**") = 0*
End Sub

```
Private Sub cmdIntegrate_Click()
Dim N As Integer, I As Integer, Sum As Double, Midpt As Double
Dim X1 As Double, X2 As Double, dArea As Double, deltaX As Double
Dim Factor As Double, Ni As Integer, J As Integer, Row As Integer

X1 = Range("B3")
X2 = Range("B4")
Ni = Range("B5")
Factor = Range("B6")
```

Note the **i** .

The number of elements, **N** , is increased by an amount "**Factor**" for 20 successive loops. If the initial value of **Ni** is too large, the value of **N** can easily exceed the maximum integer value of 32,767 . The **If** statement below checks if that will happen before the program is executed.

```
1:

If Int(Ni * Factor ^ 20) > 32700 Then
Factor = InputBox("Enter smaller value of Factor",
                  "Number of intervals too large;  also check Ni")
Range("B6") = Factor
GoTo 1
Else
End If

N = Ni
Row = 4

For J = 1 To 20

    deltaX = (X2 - X1) / N
    Midpt = X1 + (deltaX / 2)
    I = 1
    Sum = 0

      Do
          dArea = Funct(Midpt) * deltaX
          Sum = Sum + dArea
          Midpt = Midpt + deltaX
          I = I + 1
      Loop Until I = N + 1

    Cells(Row, 4) = deltaX
    Range("E" & Row) = Sum

    Row = Row + 1
    N = Int(N * Factor)
Next J

Range("A17") = N

End Sub
```

Function **Int** converts the argument to an integer

Increases **N** by the multiple **Factor**

Since **Factor** may not have an integer value, **Int ( )** is needed to convert **N * Factor** an integer value.

Private Sub cmdDefault_Click()
Range("B3") = **1**
Range("B4") = **4.5**
Range("B5") = **2**
Range("B6") = 1.6
End Sub

Test the **Input Box** operation by setting **Ni = 200** in one trial .

8.2 Incremental Iterative Approximations of the Electric and Gravitational Fields

The field of some distribution of mass or charge can be determined by imagining the distribution to be comprised of a large number of small pieces (referred to as elements). Provided they are small enough, that is as **Δm** or **Δq** approach zero, the elements behave as point masses and point charges so that the well known expressions for the fields can be applied :

$$\Delta g = G\,\Delta m/r^2 \qquad\qquad \Delta E = k\,\Delta q/r^2$$

The point masses and point charges are assumed to be located at the center of a particular element. This process is sometimes referred to as **discretization** since the original continuous distribution has been replaced by a large number of discrete point masses. ["discrete" means separate, distinct, and unconnected.] Since the both the gravitational and electric fields are vectors the X and Y components of the magnitudes of **Δg** and **ΔE** must be taken using the sine and cosine functions. [As with the theoretical analysis it is important to clearly define the **θ = 0** origin in the problem as this determines which trig function gives which component.]

The potential due to a continuous distribution of charge or mass can be found in the same way using the point mass expressions **k Δq/r** and **– GΔm/r** ; however, since potential is a scalar there is no need to take components.

Computations of fields and potentials will be left to assignments and quizzes, but in the meantime you should review the field and potential integrations studied in your Electricity & Magnetism course. The two examples given below apply the chop & sum approach to finding the moment of inertia of a thin rod.

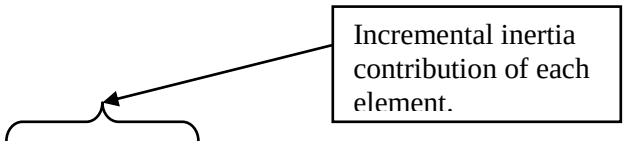
Application # 22 a) : Chop & Sum Calculation of Moment of Inertia

A thin rod extending along the X axis from X_1 to X_2 is divided into N elements. The moment of inertia contribution of each element is calculated as **X^2 Δmass** (the contribution of a point mass), and a loop is used to keep a running total that gives moment of inertia.

	A	B	C	D	E
1					
2	INPUT DATA			WIDTH OF	MOMENT
3	left end : X1 =	0		ELEMENT	OF INERTIA
4	right end : X2 =	2		0.1	39.975
5	Number of intervals : N =	20			
6	Mass of rod (kg) =	30			
7					
8					
9	Default Data				
10					
11	Calculate Inertia				
12					
13					
14	Clear Output				

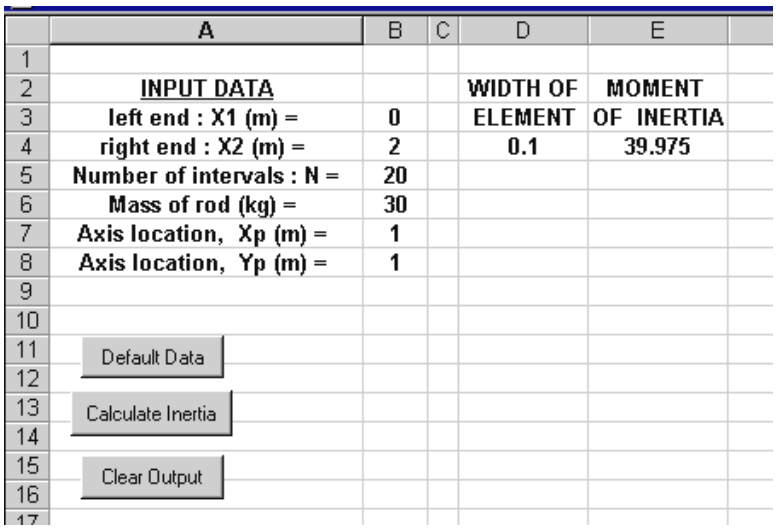
[Much of the code is identical to **Example # 1**; but watch out for slight differences if you've copied **Example # 1**. It might be faster to simply type in all the code.]

```
Option Explicit

Private Sub cmdClear_Click()
Range("E4:D4").ClearContents
End Sub

Private Sub cmdDefault_Click()
Range("B3") = 0
Range("B4") = 2
Range("B5") = 20
Range("B6") = 30
End Sub

Private Sub cmdInertia_Click()
Dim N As Integer, I As Integer, Sum As Double, Midpt As Double
Dim X1 As Double, X2 As Double, dArea As Double, deltaX As Double
Dim Mass As Double, deltaM As Double, X As Double

X1 = Range("B3")
X2 = Range("B4")
N = Range("B5")
Mass = Range("B6")


deltaM = Mass / N
deltaX = (X2 - X1) / N
Midpt = X1 + (deltaX / 2)

I = 1
Sum = 0

Do
      X = Midpt
      Sum = Sum + X ^ 2 * deltaM
      Midpt = Midpt + deltaX
      I = I + 1
Loop Until I = N + 1

Range("D4") = deltaX
Range("E4") = Sum

End Sub
```



Application # 22 b) : Chop & Sum Calculation of Moment of Inertia (About Off Axis Point)

In **Application # 22 a)** the moment of inertia of the rod was calculated for an axis through the origin. This application allows for the axis to be located at any point (X_P , Y_P); the rod is still assumed to lie along the positive X axis (Test/check whether the program would allow the left end of the rod to be located on the negative X axis. Consider what changes might be needed.)



Save **Application 22 a)** under a new name, and modify the code as follows :

```
Option Explicit

Private Sub cmdClear_Click()
Range("E4:D4").ClearContents
End Sub

Private Sub cmdDefault_Click()
Range("B3") = 0
Range("B4") = 2
Range("B5") = 20
Range("B6") = 30
Range("B7") = 1
Range("B8") = 1
End Sub

Function Hypot(X, Y, Xp, Yp)
Hypot = Sqr((X - Xp) ^ 2 + (Y - Yp) ^ 2)
End Function
```

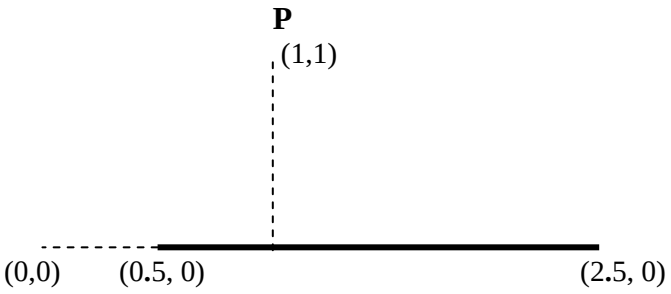
Hypot ( ) calculates the distance from the midpoint of each element to the point where the inertia is being calculated (Xp , Yp) .

```
Private Sub cmdInertia_Click()
Dim N As Integer, I As Integer, Sum As Double, Midpt As Double
Dim X1 As Double, X2 As Double, dArea As Double, deltaX As Double
Dim Mass As Double, deltaM As Double, X As Double
Dim Xp As Double, Yp As Double, r As Double

X1 = Range("B3")
X2 = Range("B4")
N = Range("B5")
Mass = Range("B6")
Xp = Range("B7")
Yp = Range("B8")

deltaM = Mass / N
deltaX = (X2 - X1) / N
Midpt = X1 + (deltaX / 2)

I = 1
Sum = 0

Do
      X = Midpt
      r = Hypot(X, 0, Xp, Yp)
      Sum = Sum + r ^ 2 * deltaM
      Midpt = Midpt + deltaX
      I = I + 1
Loop Until I = N + 1

Range("D4") = deltaX
Range("E4") = Sum

End Sub
```

As the rod lies along the X axis all the elements are located on the X axis (so that Y = 0) .

Problems [Chapter 8]

1. Use incremental iteration to find Ex at point **P** for the line of charge shown below. A total charge of 10 uC is distributed uniformly over the rod (length L = 2 meters). Refer to your class notes from Physics NYB Electricity & Magnetism, and problem 35, pg 736, Serway, and Application # 15 b) pg 7-7 of the class notes of this course for background theory and information.



   Derive the theoretical value for Ex an write it out the value for this situation on the worksheet.

2. Find an interesting** function in the Table of Integrals in your Calculus book and evaluate its integral between limits of your choice. Use incremental iteration to integrate the function and compare with the theoretical answer. Be certain to check the list of built-in Math Functions and Derived Functions when making your choice of function.
   (**one which you'd normally have great difficulty integrating; not a simple function that you've already encountered.)

Add off-axis electric potential problem from assignment April, 2005