


## Looping Introduction Continues

Larry Caretto  
Mechanical Engineering 209  
**Computer Programming for  
Mechanical Engineers**


March 2, 2017



## Outline

---

- Review Boolean variables
- Introduction to looping
- Conditional loops
  - Do While loop structure
  - Loop Until structure
- Count-controlled loops
  - The VBA For loop
  - General increment and default increment = 1




## The Do While Loop

---

Do While <condition>  
    <loop body: one or more statements>  
Loop

- The Do While <condition> loop executes as long as the <condition> is true
- If <condition> is false when loop starts, no loop statements will be executed
- Loop code should change condition
  - Infinite loop error message ends execution




## The Do ... Loop Until Structure

---

Do  
    <loop body: one or more statements>  
Loop Until <condition>


- The Loop Until <condition> executes as long as the <condition> is false
- Loop statements will be executed once even if <condition> is true initially
- Loop code should change condition
  - Infinite loop error message ends execution



## Count-controlled Loops

---


- Basic structure of VBA For loop  
    For <counter> = <start>, <fi ni sh>  
        <statements>  
    Next <counter>
- <counter> is a long/integer variable that **increases by one** each time through loop
- <start> is the initial value of <counter>
- <last> is the final value of <counter>
- <statements> executed repeatedly may use value of <counter>



## Count-controlled Loops II

---

- Modified structure of VBA For loop  
    For <counter> = <start>, \_  
        <fi ni sh> Step <i ncrement>  
        <statements>  
    Next <counter>
- Here <counter> is changed by <increment> each time through loop
- Loop is executed nTimes times where nTimes = Int((<finish> – <start>) / <increment>) + 1  
    {Not executed if nTimes ≤ 0}



### Do While Example (1/2)

```

Function mySqrt(A As Double) As Variant
    Const maxRelErr As Double = 1E-12
    Const maxIterations As Integer = 100
    Dim converged As Boolean
    Dim iterations As Integer
    mySqrt = 1
    iterations = 0
    converged = False
    Do While Not converged And _
        iterations < maxIterations
        mySqrt = mySqrt / 2 + A / _
            (2 * mySqrt)
        iterations = iterations + 1
        converged = Abs(mySqrt ^ 2 - A) < _

```

Next slide starts here

### Do While Example (2/2)

```

    mySqrt = 1
    iterations = 0
    converged = False
    Do While Not converged And _
        iterations < maxIterations
        mySqrt = mySqrt / 2 + A / _
            (2 * mySqrt)
        iterations = iterations + 1
        converged = Abs(mySqrt ^ 2 - A) < _
            maxRelErr * Abs(A)
    Loop
    If Not converged Then mySqrt = _
        "No Convergence"
End Function

```

### Loop Until Example (1/2)

```

Function mySqrt(A As Double) _
    As Variant
    Const maxRelErr As Double = 1E-12
    Const maxIterations As Integer = 100
    Dim converged As Boolean
    Dim iterations As Integer
    iterations = 0 : mySqrt = 1
    Do
        mySqrt = mySqrt / 2 + A / _
            (2 * mySqrt)
        iterations = iterations + 1
        converged = Abs(mySqrt ^ 2 - A) _
            < maxRelErr * Abs(A)

```

Next slide starts here

### Loop Until Example (2/2)

```

    Do
        mySqrt = mySqrt / 2 + A / _
            (2 * mySqrt)
        iterations = iterations + 1
        converged = Abs(mySqrt ^ 2 - A) _
            < maxRelErr * Abs(A)
    Loop Until converged Or _
        iterations > maxIterations
    If Not converged Then mySqrt = _
        "No Convergence"
End Function

```

California State University  
Northridge

10

### For Loop Example (1/2)

```

Function mySqrt2(A As Double) As _
    Variant
    Const maxRelErr As Double = 1E-12
    Const maxIterations As Integer = 100
    Dim converged As Boolean
    Dim iteration As Integer
    mySqrt = 1 'Only initialization
    For iterations = 1 To maxIterations
        mySqrt2 = mySqrt2 / 2 + A / _
            (2 * mySqrt2)
        If Abs(mySqrt2 ^ 2 - A) < _
            maxRelErr * Abs(A) Then _
            Exit Function

```

Next slide starts here

### For Loop Example (2/2)

```

    mySqrt = 1 'Only initialization
    For iterations = 1 To maxIterations
        mySqrt2 = mySqrt2 / 2 + A / _
            (2 * mySqrt2)
        If Abs(mySqrt2 ^ 2 - A) < _
            maxRelErr * Abs(A) Then _
            Exit Function
    Next iteration
    'maximum iterations exceeded
    'if we get here
    mySqrt2 = "No Convergence"
End Function

```

Northridge

### Count Controlled Series Sum

- Code below left computes  $e^x$  for  $x = 1$  using ten terms in series
- For  $x = 1$  ten terms gives a relative error of  $1 \times 10^{-8}$

```

x = 1 : term = 1 : sum = term
For n = 1 to 10
    term = term * x / n
    sum = sum + term
Next n
relErr = abs(sum/exp(x) - 1)
    
```

Colon (:) allows multiple statements per line

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

California State University Northridge 13

### Tracing Loops

```

x = 1 : term = 1 : sum = term
For n = 1 to 10
    term = term * x / n
    sum = sum + term
Next n
    
```

Exercise: What is sum for n = 6?  
 term = 1/720  
 sum = 1957/720

Initialization:  $x = 1$  term = 1 sum = term = 1  
 n=1: term =  $1 * 1 / 1 = 1$ ; sum =  $1 + 1 = 2$   
 n=2: term =  $1 * 1 / 2 = 1/2$ ; sum =  $2 + 1/2 = 5/2$   
 n=3: term =  $1/2 * 1 / 3 = 1/6$ ; sum =  $5/2 + 1/6 = 16/6$   
 n=4: term =  $1/6 * 1 / 4 = 1/24$ ; sum =  $16/6 + 1/24 = 65/24$   
 n=5: term =  $1/24 * 1/5 = 1/120$ ; sum =  $65/24 + 1/120 = 163/60$

California State University Northridge 14

### Evaluation on Worksheet

n	x	term	sum	term/sum
1	1	1	1	1
2	0.5	2.5	0.2	
3	0.166667	2.666667	0.0625	
4	1/24	2.708333	0.015385	
5	0.008333	2.716667	0.003067	
6	0.001389	2.718056	0.000511	
7	0.000198	2.718254	7.3E-05	
8	2.48E-05	2.718279	9.12E-06	
9	2.76E-06	2.718282	1.01E-06	
10	2.76E-07	2.718282	1.01E-07	
11	2.51E-08	2.718282	9.22E-09	
12	2.09E-09	2.718282	7.68E-10	
13	1.61E-10	2.718282	5.91E-11	
14	1.15E-11	2.718282	4.22E-12	
15	7.65E-13	2.718282	2.81E-13	
16	4.78E-14	2.718282	1.76E-14	

Numerical Values

F	G	H	I
n	term	sum	term/sum
1	1	=G4	=G4/H4
2	=F5+1	=G5*x/F6	=H4+G6
3	=F6+1	=G6*x/F7	=H5+G6
4	=F7+1	=G7*x/F8	=H7+G8
5	=F8+1	=G8*x/F9	=H8+G9
6	=F9+1	=G9*x/F10	=H9+G10
7	=F10+1	=G10*x/F11	=H10+G11
8	=F11+1	=G11*x/F12	=H11+G12
9	=F12+1	=G12*x/F13	=H12+G13
10	=F13+1	=G13*x/F14	=H13+G14
11	=F14+1	=G14*x/F15	=H14+G15
12	=F15+1	=G15*x/F16	=H15+G16
13	=F16+1	=G16*x/F17	=H16+G17
14	=F17+1	=G17*x/F18	=H17+G18
15	=F18+1	=G18*x/F19	=H18+G19
16	=F19+1	=G19*x/F20	=H19+G20

Equation View (x is named cell)

California State University Northridge 15

### The Cells Command

- Way to address cell locations by row number and column number
  - Columns A, B, C, are columns 1, 2, 3, etc.
  - Columns Z, AA, AB: columns 26, 27, 28
  - ZZ and AAA are columns 702 and 703
  - Column XFD is column 16384
- Use: Cells(<row>, <col>).Value
  - Note that row comes before column
  - What is conventional name of Cells(5,3)?

California State University Northridge C5 16

### Nested For Loops

- The simplest example of nested for loops is two loops forming a table

```

For row = firstRow to lastRow
    For col =firstCol to lastCol
        <code for computations for each row-col combination>
    Next col
Next row
    
```

California State University Northridge 17

### Example – What is Output

```

Sub testCells()
    Dim row As Long
    Dim col As Long
    For row = 5 To 10
        For col = 1 To 7
            Cells(row, col).Value = row + col
        Next col
    Next row
End Sub
    
```

Output will be in rows 5 through 10 and columns A through G

California State University Northridge 18

### Result of Code

	A1	B2	C3	D4	E5	F6	G7
1							
2							
3							
4							
5	6	7	8	9	10	11	12
6	7	8	9	10	11	12	13
7	8	9	10	11	12	13	14
8	9	10	11	12	13	14	15
9	10	11	12	13	14	15	16
10	11	12	13	14	15	16	17

For row = 5 To 10  
 For col = 1 To 7  
 Cells(row, col).Value = row + col

California State University Northridge 19

```
Sub simpleTwoWayTable()
    Dim velocity As Long, mass As Long
    Dim row As Long, col As Long

    row = 1
    For mass = 10 To 100 Step 10
        col = 1
        For velocity = 5 To 25 Step 5
            Cells(row, col).Value = mass * _
                velocity ^ 2 / 2

            col = col + 1
        Next velocity
        row = row + 1
    Next mass
End Sub
```

Computes table of kinetic energy =  $mV^2/2$

California State University Northridge 20

```
Sub simpleTwoWayTable()
    Dim velocity As Long, mass As Long
    Dim row As Long, col As Long
    row = 1
    For mass = 10 To 100 Step 10
        col = 1
        For velocity = 5 To 25 Step 5
            Cells(row, col).Value = _
                mass * velocity ^ 2 / 2
            col = col + 1
        Next velocity
        row = row + 1
    Next mass
End Sub
```

EXERCISE: Modify code to have mass = 2,4,6,8,10 and velocity = 1,2,3,4,5,6

California State University Northridge 21

### Result of Two-Way Table Code

	A	B	C	D	E
1	125	500	1125	2000	3125
2	250	1000	2250	4000	6250
3	375	1500	3375	6000	9375
4	500	2000	4500	8000	12500
5	625	2500	5625	10000	15625
6	750	3000	6750	12000	18750
7	875	3500	7875	14000	21875
8	1000	4000	9000	16000	25000
9	1125	4500	10125	18000	28125
10	1250	5000	11250	20000	31250

Need to provide headers and title

California State University Northridge 22

### Work on Program Assignment 3

- Macro that uses If-ElseIf structure to solve quadratic equation  $ax^2 + bx + c = 0$
- Checks for and reports errors due to non-numeric data and special cases  $a = b = c = 0$ ,  $a = b = 0$  and  $c \neq 0$
- Gets solution for linear case ( $a = 0$  and  $b \neq 0$ ), duplicate real root ( $b^2 = 4ac$ ), distinct real roots and complex roots ( $b^2 < 4ac$ )

California State University Northridge 23