


Looping Introduction


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

February 28, 2017



Outline


- Solution to February 23 exercise
- 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



February 23 Exercise Solution

```
Function getGrade(score As Double, mean As Double, _
    standardDeviation As Double) As String
    If score > mean + 1.2 * standardDeviation Then
        getGrade = "A"
    ElseIf score > mean + 0.52 * standardDeviation Then
        getGrade = "B"
    ElseIf score > mean - 0.52 * standardDeviation Then
        getGrade = "C"
    ElseIf score > mean - 1.2 * standardDeviation Then
        getGrade = "D"
    Else
        getGrade = "F"
    End If
End Function
```


See complete
workbook with
solution online



Review Boolean Variables

- Boolean variables have two possible values: TRUE or FALSE
- Can set Boolean variables equal to a relational or logical expression
- Must declare variables as Boolean in a Dim statement



```
Dim converged As Boolean
Dim error As Boolean
Dim metricUnits As Boolean
```



Review Boolean Variables II

- Four examples of using Boolean variables



```
converged = err <= maxErr
error = x < xMin or x > xMax
If leap(thisYear) Then febDays = 29
Function leap(year As Integer) _
    As Boolean
    leap = (year Mod 4 = 0 And _
        year Mod 100 <> 0) _
        Or (year Mod 400 = 0)
End Function
```



Function with Boolean Return Value

```
Function notNumericData(a As Variant, b _
    As Variant, c As Variant) As Boolean
    notNumericData = False
    If Not IsNumeric(a) Then
        notNumericData = True
        MsgBox "a=" & a & " is nonNumeric data"
    End If
    If Not IsNumeric(b) Then
        notNumericData = True
        MsgBox "b=" & b & " is nonNumeric data"
    End If
    If Not IsNumeric(c) Then
        notNumericData = True
        MsgBox "c=" & c & " is nonNumeric data"
    End If
End Function
```

Example of Use:
If notNumericData(a,b,c) Then Exit Function



Looping

- Looping repeats the same set of commands several times
 - This is usually done with different data each time as in the following examples
 - Prepare the grade report for every student in the university
 - Repeat a trial-and-error calculation until the error is less than the minimum error allowed
 - Process the experimental data for every run and compute average values of the results

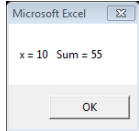
California State University Northridge 7

Conditional Loops

- Conditional loops are repeated until a certain condition changes
- First conditional VBA loop: Do While

```

Sub test()
  Dim x As Long, sum As Long
  x = 0 : sum = 0
  Do While x < 10
    x = x + 1 : sum = sum + x
  Loop
  MsgBox "x = " & x & " Sum = " & sum
End Sub
    
```



What was last value of x in loop? 9 → 10

California State University Northridge

The Do While Loop

- The Do While <condition> loop executes as long as the <condition> is true
- The Do While x < 10 ends only when x ≥ 10 **at the start of the loop**
- In the operation of the loop, a new value of x is tested before the loop starts
- Once the loop starts all statements before the "Loop" statement are completed

– Will see alternative exit procedure later

California State University Northridge 9

Square Root Trial and Error

- Find a, the square root of A, by the following iteration equation
 - $a_{k+1} = \frac{a_k}{2} + \frac{A}{2a_k}$ **a_0 is initial guess; $a_1, a_2, a_3,$ etc. are subsequent values**
 - Do we believe this works? Look at an example of finding $\sqrt{9}$ with initial guess of 1
 - Here A = 9 and $a_0 = 1$
 - $a_1 = \frac{a_0}{2} + \frac{A}{2a_0} = \frac{1}{2} + \frac{9}{2(1)} = 5$
 - $a_2 = \frac{a_1}{2} + \frac{A}{2a_1} = \frac{5}{2} + \frac{9}{2(5)} = \frac{25+9}{10} = \frac{34}{10}$

California State University Northridge 10

Results of Iteration

k	a_k	a_{k+1}
0	1	5
1	5	3.4
2	3.4	3.023529412
3	3.023529412	3.000091554
4	3.000091554	3.000000001396980
5	3.000000001396980	3.000000000000000

Numerical Results

Equation View

$$a_{k+1} = \frac{a_k}{2} + \frac{A}{2a_k}$$

k	a_k	a_{k+1}
0	1	=B3/2+\$B\$1/(2*B3)
1	5	=B4/2+\$B\$1/(2*B4)
2	3.4	=B5/2+\$B\$1/(2*B5)
3	3.023529412	=B6/2+\$B\$1/(2*B6)
4	3.000091554	=B7/2+\$B\$1/(2*B7)
5	3.000000001396980	=B8/2+\$B\$1/(2*B8)

California State University Northridge 11

Alternative Excel Results

k	a_k	Error
0	1	2.00E+00
1	5	-2.00E+00
2	3.4	-4.00E-01
3	3.0235294117647	-2.35E-02
4	3.0000915541314	-9.16E-05
5	3.0000000013970	-1.40E-09
6	3.0000000000000	

Numerical Results

Equation View

$$a_{k+1} = \frac{a_k}{2} + \frac{A}{2a_k}$$

k	a_k	a_{k+1}
0	1	=B3/2+\$B\$1/(2*B3)
1	5	=B4/2+\$B\$1/(2*B4)
2	3.4	=B5/2+\$B\$1/(2*B5)
3	3.0235294117647	=B6/2+\$B\$1/(2*B6)
4	3.0000915541314	=B7/2+\$B\$1/(2*B7)
5	3.0000000013970	=B8/2+\$B\$1/(2*B8)
6	3.0000000000000	=B9/2+\$B\$1/(2*B9)

California State University Northridge 12

How to do this in VBA

- This is an example of a trial and error loop
- We would want to execute the loop while the error is greater than some allowed error
- The result we seek is $a^2 = A$ so $|a^2 - A|$ is a measure of the absolute error
- How accurate do we want our result to be? That is the allowed error

California State University Northridge 13

Relative or Absolute Error

- A check on the absolute error is given by the condition $|a^2 - A| < \epsilon_{abs}$
- A check on the relative error is given by the condition $|a^2 - A| < \epsilon_{rel} |A|$
 - Relative error is usually more important
 - Command continues the calculations while the error is larger than desired

Do While Abs(a^2-A) > maxRelErr*Abs(A)

– Note use of absolute values to compare magnitudes only!

California State University Northridge 14

Simple Function

```

Function mySqrt(A As Double) As Double
    Const maxRelErr As Double = 1e-12
    mySqrt = 1 'Old and new value
    Do While Abs(mySqrt ^ 2 - A) > maxRelErr * Abs(A)
        mySqrt = mySqrt / 2 + A / (2 * mySqrt)
    Loop
End Function
    
```

$a_{k+1} = \frac{a_k}{2} + \frac{A}{2a_k}$

California State University Northridge 15

Application of mySqrt

N	mySqrt(N)	Err
0	1.1114E-162	1.1E-162
1	1	0
2	1.414213562	0
3	1.732050808	0
4	2	0
5	2.236067977	1.88E-13
6	2.449489743	0
7	2.645751311	0
8	2.828427125	0
9	3	0
10	3.16227766	0

Equation View

D	E	F
10	N	mySqrt(N)
11	0	=mySqrt(D11)
12	1	=mySqrt(D12)
13	2	=mySqrt(D13)
14	3	=mySqrt(D14)
15	4	=mySqrt(D15)
16	5	=mySqrt(D16)
17	6	=mySqrt(D17)
18	7	=mySqrt(D18)
19	8	=mySqrt(D19)
20	9	=mySqrt(D20)
21	10	=mySqrt(D21)

Numerical Results

California State University Northridge

Careful Iterations

- Iteration problems may not converge
- Should be programmed with a maximum iteration limit
- If limit is exceeded an error message is returned
- If function return type is variant, the return may be a numerical result or a string error message

California State University Northridge 17

Careful Function

```

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

Careful Function (continued)

```

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 Command

- An alternative to the Do While ... Loop
 - Starts with the command Do
 - Followed by loop body with all code executed in loop
 - Concludes with Loop Until <condition>
- Loop **continues as long as <condition> is false**
- Loop is exited when <condition> is true
- Redo previous example

Careful Function with Loop Until

```

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)
    Loop Until converged Or _
        iterations > maxIterations
    If Not converged Then mysqrt = _
        "No Convergence"
End Function
    
```

Next slide starts here

Careful Function with Loop Until

```

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
    
```

Count-controlled Loops

- Basic structure of VBA For loop


```
For <counter> = <start>, <finish>
    <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>, _
    <finish> Step <increment>
    <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}

For Loop Increment > 1

```

For k = 1 To 11 Step 3
  MsgBox k
Next k
MsgBox _
  "After exit, k = " & k
    
```

25

nTimes Formula

```

nTimes =
Int ((Finish-Start) / Increment) + 1
=Int ((11-1) / 3) + 1
=Int(3.33 ...) + 1 = 4
    
```

For k = 1 To 11 Step 3
MsgBox k
Next k
MsgBox _
"After exit, k = " & k

- In this example the loop is executed 4 times for k = 1, 4, 7, and 10
- In general, the last value of the loop index is \leq the Finish value of the For command

26

Count Controlled Series Sum

Code below left computes e^x for $x = 1$ using eleven terms in an infinite series

For $x = 1$ eleven terms gives a relative error of 1×10^{-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)
n! = n(n-1)(n-2) ... (3)(2)(1)
    
```

Colon (:) allows multiple statements per line

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

27

Tracing Loops

```

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

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 = $65/24 + 1/120 = 163/60$

28

Evaluation on Worksheet

n	term	sum	term/sum
1	1	1	1
2	0.5	1.5	0.333333
3	0.166667	1.666667	0.092593
4	0.041667	1.708333	0.024375
5	0.008333	1.716667	0.004861
6	0.001389	1.718556	0.000811
7	0.000198	1.718254	1.15E-05
8	2.48E-05	1.718279	1.43E-06
9	2.76E-06	1.718282	1.61E-07
10	2.51E-07	1.718282	1.46E-08
11	2.09E-08	1.718282	1.22E-09
12	1.61E-09	1.718282	9.36E-11
13	1.15E-10	1.718282	6.70E-12
14	7.65E-12	1.718282	4.45E-13
15	4.78E-13	1.718282	2.79E-14

Numerical Values

F	G	H	I
3	n	term	sum
4	1	1	=G4
5	1	1	=H4+G5
6	=F5+1	=G5*x/F6	=H5+G6
7	=F6+1	=G6*x/F7	=H6+G7
8	=F7+1	=G7*x/F8	=H7+G8
9	=F8+1	=G8*x/F9	=H8+G9
10	=F9+1	=G9*x/F10	=H9+G10
11	=F10+1	=G10*x/F11	=H10+G11
12	=F11+1	=G11*x/F12	=H11+G12
13	=F12+1	=G12*x/F13	=H12+G13
14	=F13+1	=G13*x/F14	=H13+G14
15	=F14+1	=G14*x/F15	=H14+G15
16	=F15+1	=G15*x/F16	=H15+G16
17	=F16+1	=G16*x/F17	=H16+G17
18	=F17+1	=G17*x/F18	=H17+G18
19	=F18+1	=G18*x/F19	=H18+G19
20	=F19+1	=G19*x/F20	=H19+G20

Equation View (x is named cell) 29

Careful Function with For Loop

```

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 * mySqrt)
  If Abs(mySqrt2 ^ 2 - A) < _
    maxRelErr * Abs(A) Then _
    Exit Function
    
```

Next slide starts here

Careful Function with For Loop

```
mySqrt = 1 'Only initialization
For iterations = 1 To maxIterations
    mySqrt2 = mySqrt / 2 + A / _
                (2 * mySqrt)
    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
norriage
```