Programming with Arrays

Larry Caretto
Mechanical Engineering 209

Computer Programming for Mechanical Engineers

April 13, 2017

California State University Northridge

Outline

- Review array basics
  - Notation, declaration and minimum subscripts for 1D and 2D arrays
  - Using arrays in for loops and expressions
  - Arrays to and from worksheet
- Two-dimensional arrays
- Passing arrays to other procedures
- Getting cell data from worksheets to VBA array variables
- Programming assignment four

California State University Northridge 2

Semester Calendar

Date	Assignment	Date	Assignment
April 18	Assignment 4 Due	April 20	Quiz 4
April 25	Assignment 5 Due	April 27	
May 2	Quiz 5	May 4	
May 9	Assignment 6 Due	May 11	Programming Exam
		May 18	Final Exam 12:45 to 2:45 pm

California State University Northridge 3

Review VBA Arrays

Math	VBA
x_1	x(1)
x_2	x(2)
x_3	x(3)
x_4	x(4)
x_5	x(5)
x_6	x(6)

- Math notation, x_i, is replaced by VBA notation x(i)
  - Name for (i) is index or subscript
  - View this one-dimen-sional array as one row or one column of data
  - Must declare maximum array size

California State University Northridge 4

Review Declaring Arrays

- Use Dim with array size
  - Usual statement: Dim x As …
  - Array statement: Dim x(<size>) As …
  - May use <size> as <minimum> To <maximum> to specify minimum and minimum subscript for an individual array
  - <size> as a single number specifies the maximum subscript
    - In this case default <minimum> is normally zero
    - Can change default <minimum> to one with statement: `Option Base 1`

California State University Northridge 5

Review Using Array Data

- We can use individual array elements like any other programming variable
  - E = stress(3)/strain(3)
  - k = 3 : E = stress(k)/strain(k)
  - For k = 1 To N : sum = sum + x(k) : Next k
  - j = 2 : power(j) = current(j) * voltage(j)
  - Dim pressure(1 to 10) as Double
    meanPressure = Application. _
    Worksheetfunction.Average(pressure)

California State University Northridge 6

Review For Loops and Arrays

- A common application of arrays uses for loops to set elements in an array

```
For k = 1 To 10
    x(k) = cos(k*PI/10)
Next k
```

- Can sum all elements in an array

```
sum = x(1)
For k = 2 To NX
    sum = sum + x(k)
Next k
```

California State University Northridge 7

Review For Loops and Arrays II

- Process all data in a data set

```
For k = first To last
  power(k) = amps(k) * volts(k)
Next k
```

- Find mean of computed result

```
sumP = 0
For k = first To last
 sumP = sumP + amps(k) * volts(k)
Next k
meanP = sump / (last – first + 1)
```

California State University Northridge 8

Review Two-dimensional Arrays

- View as two-dimensional table of data
- Row and column subscripts
- Must set both subscripts in Dim statement
  - Dim e(1 to 4,1 To 6) As Double
  - Same default minimum index (0 or 1) as 1D

	I(1)	I(2)	I(3)	I(4)	I(5)	I(6)
V(1)	e(1,1)	e(1,2)	e(1,3)	e(1,4)	e(1,5)	e(1,6)
V(2)	e(2,1)	e(2,2)	e(2,3)	e(2,4)	e(2,5)	e(2,6)
V(3)	e(3,1)	e(3,2)	e(3,3)	e(3,4)	e(3,5)	e(3,6)
V(4)	e(4,1)	e(4,2)	e(4,3)	e(4,4)	e(4,5)	e(4,6)

California State University Northridge 9

Review For Loops/2D Arrays

- Two-way table from two-dimensional arrays and nested for loops
  - Need to define P(k) and T(j) arrays prior to executing this code

```
For k = 1 to 10
  For j = 1 to 20
    rho(i,j) = M*P(k) / (R*T(j))
  Next j
Next k
```

California State University Northridge 10

Review Worksheets and Arrays

- Can replace x = Range("A1:T10").Value by code below (using numbers only)

```
Const NX As Long = 10, NY As Long = 20
Dim row As Long, col as Long
Dim x(1 to NX, 1 To NY)
For row = 1 To NX
   For col = 1 To NY
     Cells(row,col).Value= x(row,col)
   Next col
Next row
```

Note use of Const for array bounds and for loop limits. Changes to NX or NY can be done by changing Const values.

California State University Northridge 11

Adjustable Size Arrays

- Usual statement, Dim x(5) As …, assumes that we know the maximum size of the array when the program is written
- What if this is not the case?
- Dim x(5) As … says three things
  - x is an array
  - The x array has one dimension
  - It's maximum subscript is 5
- When we do not know the size initially, we use Dim() without size!

California State University Northridge 12

Adjustable Size Arrays II

- The initial statement Dim x() As … lets the compiler know that x is an array
  - When it sees x used as an array it will not mark it as an error and halt execution
  - When the program learns the size of the array, use a ReDim statement to set the size

```
Dim x() As Double, N As Long
N = InputBox("Enter Size: ")
  ReDim x(1 To N)
```

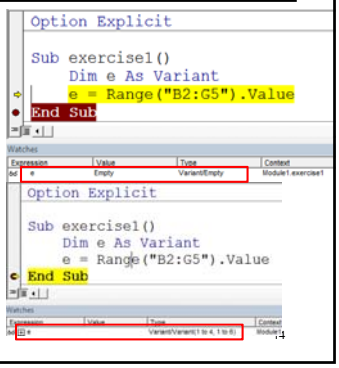
California State University Northridge 13

The Mysterious Variant

- Type Variant Variables, declared as scalars, can be equated to an array
- They then become arrays
- Used for getting cell data to VBA

```
Option Explicit

Sub exercise1()
    Dim e As Variant
    e = Range("B2:G5").Value
End Sub
```

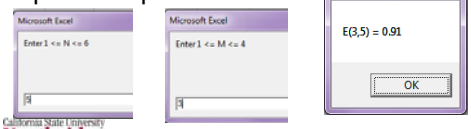


California State University Northridge 14

The Mysterious Variant II

- Can use array created from variant like any other array
- Simple input/ output example

```
Sub exercise1()
    Dim e As Variant, N As Integer, _
                     M As Integer
    e = Range("B2:G5").Value
    N = InputBox("Enter 1 <= N <= 6")
    M = InputBox("Enter 1 <= M <= 4")
    MsgBox "E(" & M & "," & N & _
                    ") = " & e(M, N)
End Sub
```



California State University Northridge 15

One-dimensional Array Exercise

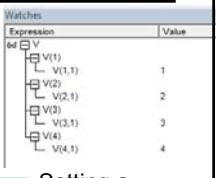
- Write the VBA code to create a current, I, and voltage, V, array from the worksheet data
  - Using only Range.Value (Use type variant.)
  - Using Cells(<row>,<column>).Value

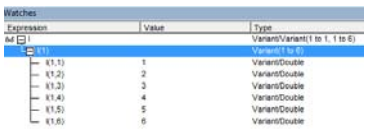
	A	B	C	D	E	F	G
1		I(1)	I(2)	I(3)	I(4)	I(5)	I(6)
2	V(1)	e(1,1)	e(1,2)	e(1,3)	e(1,4)	e(1,5)	e(1,6)
3	V(2)	e(2,1)	e(2,2)	e(2,3)	e(2,4)	e(2,5)	e(2,6)
4	V(3)	e(3,1)	e(3,2)	e(3,3)	e(3,4)	e(3,5)	e(3,6)
5	V(4)	e(4,1)	e(4,2)	e(4,3)	e(4,4)	e(4,5)	e(4,6)

California State University Northridge 16

Exercise Solution for Range

```
Sub exerciseA()
  Dim I As Variant
  Dim V As Variant
  I = Range("B1:G1").Value
  V = Range("A2:A5").Value
End Sub
```



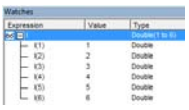


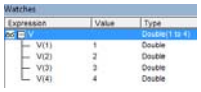
Setting a Variant to a single row or column gives a two-dimensional array

California State University Northridge 17

Exercise Solution for Cells

```
Sub exerciseB()
  Dim I(1 to 6) As Double
  Dim V(1 to 4) As Double
  Dim k As Integer
  For k = 1 To 6
    I(k) = Cells(1,k+1).Value
  Next k
  For k = 1 To 4
    V(k) = Cells(k+1,1).Value
  Next k
End Sub
```





California State University Northridge 18

2D Array to VBA Exercise

- Write the VBA code to create a two-dimensional efficiency array, e, from the worksheet data
- Use Range.Value to get all data for e in a single statement. What is the range for e? B2:G5

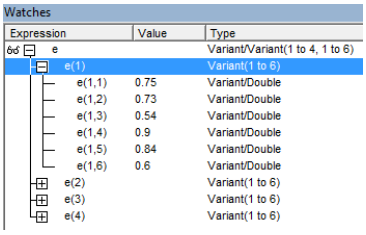
	A	B	C	D	E	F	G
1		I(1)	I(2)	I(3)	I(4)	I(5)	I(6)
2	V(1)	e(1,1)	e(1,2)	e(1,3)	e(1,4)	e(1,5)	e(1,6)
3	V(2)	e(2,1)	e(2,2)	e(2,3)	e(2,4)	e(2,5)	e(2,6)
4	V(3)	e(3,1)	e(3,2)	e(3,3)	e(3,4)	e(3,5)	e(3,6)
5	V(4)	e(4,1)	e(4,2)	e(4,3)	e(4,4)	e(4,5)	e(4,6)

California State University Northridge

Exercise Solution for Range

```
Sub exerciseC()
  Dim e As Variant
  e = Range("B2:G5").Value
End Sub
```

Here e is declared as a scalar variant, but becomes an array when set to a worksheet range



California State University Northridge

Programming Exercise 4

- For table of A/P formula as a function of i and n use a nested for loop over rows and columns
- Use `Cells(row,column).Value _ = i / (1 – (1 + i) ^ (-n)`
- Followed by `Cells(row,column). _ NumberFormat = "0.000000"`
- DO NOT USE Cells.Value = Format(_ i/(1 – (1 + i) ^ (-n), "0.000000")

California State University Northridge

More Exercise Four

- See March 30 notes, slides 12-23 for both A/P and cosine calculations
- For cosine calculation convert input angle to equivalent value between $-\pi/2$ and $\pi/2$ as described in assignment
- Use maximum relative error of 10^{-12} for cosine calculation
- Due 11:59 pm on April 18
- Exercise 5 due 11:59 pm on April 25

California State University Northridge