Course Introduction

Larry Caretto
Mechanical Engineering 309
Numerical Analysis of Engineering Systems

January 22, 2014

California State University Northridge

Outline

- First class day items: roll, outline, etc.
- Class goals and learning objectives
- Assessment quiz
- Excel exercise
- VBA exercise
- Review of Excel and VBA
- First assignment

Basic Information

- Larry Caretto, Jacaranda (Engineering) 3314, lcaretto@csun.edu, no phone
  - Questions by email best contact method
- Office hours: MW 6 – 6:45 pm and F 8 – 8:45 am in JD 3314
- http://www.csun.edu/~lcaretto/me309
- Singiresu S. Rao, *Applied Numerical Methods for Engineering and Science*, Pearson Education, 2002
- Excel and MATLAB references on outline

Self Introductions

- Take picture of next student in serpentine order and sign roll in same order
- In same order introduce yourself with following information
  - Name and school attended before CSUN
  - Career goals/technical interests
  - Rate your familiarity with (1) Excel, (2) VBA, and (3) Programming (1=none to 5=expert)
  - Give one interesting fact about yourself

Course Learning Objectives

- Be able to apply various techniques in numerical analysis
- Understand and do calculations about truncation errors and roundoff errors, that can occur in numerical methods.
- Understand and be able to use the basics of matrix analysis
- Be able to use MATLAB functions for solving numerical analysis problems

Course Learning Objectives

- Be able to write programs in Visual Basic that can be used as user defined functions (UDF) or macros for Excel spreadsheets.
- Be able to use MATLAB for symbolic mathematical solutions of algebraic equations, integrals and differential equations.
- Use MATLAB code for m-files.

Numerical Analysis Methods

- Roots of equations
- Simultaneous linear equations
- Numerical integration and differentiation
- Polynomial interpolation
- Regression analysis (statistical curve fitting)
- Numerical solution of ordinary differential equations

California State University Northridge 7

Prerequisite Course

- ME 209 or other introductory programming course
- ME 209 uses the same VBA and Excel techniques expanded in this course
- Students who have no VBA experience should pay particular attention to review
- First exercise, due Monday, February 3, 11:59 pm, is mostly a review (Late Deadline Wednesday, February 5, 11:59 pm)

California State University Northridge 8

Class Operation

- Classes will have lecture material and time to work on exercises/assignments
- Seven programming assignments
- Eight quizzes at start of class on selected Wednesdays (no computer use)
- Midterm exam (March 26)
- Programming Exam (open book, May 7)
- Final Exam (May 12)
  - Quizzes, midterm, and final will be closed book with instructor-supplied information sheet

California State University Northridge 9

Programming Assignments

- Seven programming assignments to be submitted by email with computer file for calculations and Word file for discussion
  - Discussion will be graded on completeness of description, grammar, and appearance
- Two students can choose to submit a common submission for their grade
  - If you do this, make sure that you understand everything in the assignment
- See schedule in course outline

California State University Northridge 10

Assignment Deadlines

- Email submission by deadline must include computer file (Excel or MATLAB) and Word Document
  - Contents of these files will be specified in the assignment
  - MATLAB assignments will have single file
- Two deadlines: first has no late penalty
- Second deadline has 30% late penalty
- Zero score if submitted after second deadline

California State University Northridge 11

Eight In-Class Quizzes

- Held on Wednesdays at start of class
- Thirty minutes, closed book, students use instructor-supplied list of computer commands in MATLAB and VBA
- Based on previous lecture and current programming assignments
- Write code on paper; no implementation on computer required

California State University Northridge 12

Grading

- Wednesday in-class quizzes 20%
- Programming assignments 20%
- Midterm Exam (March 26) 15%
- Programming exam (May 7) 20%
- Final Exam (May 12) 25%
- Plus/minus grading will be used
- Grading criteria/details in course outline
- No make-up quizzes or exams

California State University Northridge 13

See the Course Outline

- Download from course web site
  - http://www.csun.edu/~lcaretto/me309/outline.pdf
- Contains lecture schedule, reading assignments, dates for assignments, quizzes, exams
- Also read information on the following items
  - Class participation and courtesy
  - Collaboration versus plagiarism: students found cheating receive F grade in course
- Students are responsible for changes to outline announced in class

California State University Northridge 14

Galileo Galilei

(1564-1642)

You cannot teach people anything; you can only help them find it within themselves

http://space.about.com/od/astronomyhistory/a/galileoquotes.htm

(Accessed January 16,2013)

California State University Northridge 15

Goals for this Course

- My goal is to help all students find within themselves sufficient knowledge of numerical analysis and programming so that they will all get an A grade in the course
- What is your goal for this course?
- What will we do to help us achieve our mutual goals?

California State University Northridge 16

How to get your A

- Spend six to ten hours per week outside class on study and course assignments
- Prepare for lecture and be ready to ask questions
  - Read the assigned reading before class
  - Download, print, and review the lecture presentations before class
    - Use presentations instead of writing notes
    - Listen to the lecture and ask questions
    - Write additional notes on presentations

California State University Northridge 17

How to Get your A, Part II

- Study with fellow students and try to answer each other's questions
- Do the programs as well as you can before asking for help
  - Make sure you understand the help you get
- Contact me by email, telephone or office visits to ask questions
- Work with a group of classmates to help each other learn

California State University Northridge 18

What I will do to help

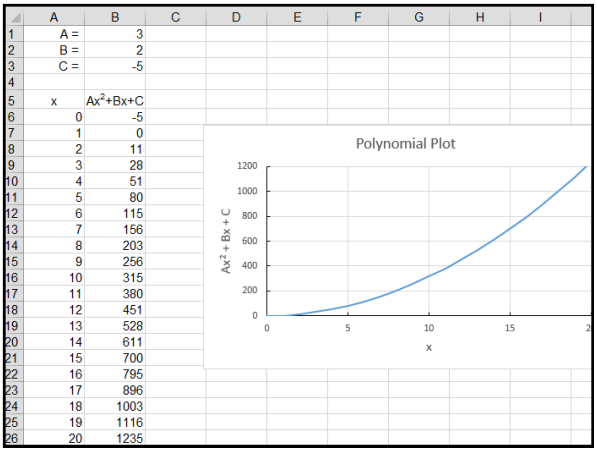
- Arrive at class a few minutes early to answer any questions you may have
- Give lectures that stress application of basics to writing correct programs
- Return programs and exams promptly so that you can learn from your errors
- Be available for questions in my office (visit or telephone, if avaiilable) or email
  - Send entire class emails as appropriate

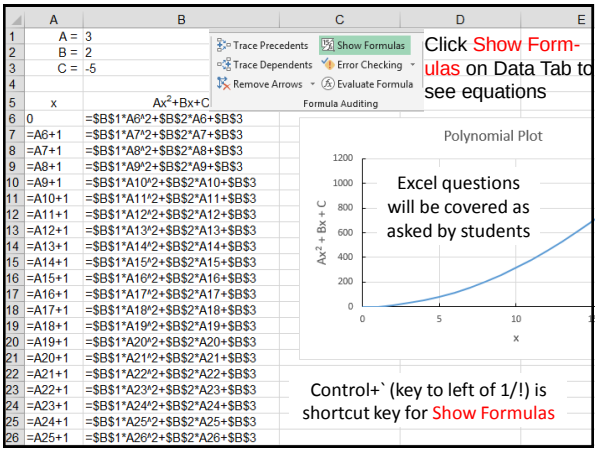
Preliminary Assessment

- Designed to help instruction
- One set of questions on student background
- Second set of questions is ungraded quiz
- Take about 10 minutes for this assessment
- Hand yours in when finished
  - Will call time when most students are done

Excel Exercise

- Compute values of $y = Ax^2 + Bx + C$ for $x = 0$ to $x = 20$ with $\Delta x = 1$
- Enter the values of A, B, and C in single cells so that they can be changed
  - Use A = 3, B = 2, and C = –5
- Plot the results





VBA Exercise

- Write VBA code to compute the volume of a right circular cylinder: $V = \pi R^2 H$
  - Option A: a user defined function (UDF)
  - Option B: Exchange data with worksheet using objects like range("A1").Value
  - Define π as PI = 4 * atn(1) within code
    - Recall that $\tan(45^o) = \tan(\pi/4) = 1$ so that $\tan^{-1}(1) = \pi/4$
    - atn is VBA function for arctangent
  - Test your code with data from and result to worksheet (Is your answer correct?)

Cylinder Volume Solutions

```
Option Explicit
Const PI As Double = 3.14159265358979
Function volCyl(ByVal radius As Double, _
                ByVal height As Double) As Double
    volCyl = PI * radius ^ 2 * height
End Function
Sub getCylVol()
    Dim r As Double
    Dim h As Double
    r = InputBox("Enter Radius: ")
    h = InputBox("Enter Height: ")
    MsgBox "Cylinder Volume = " & volCyl(r, h)
End Sub
```

California State University Northridge 25

```
Sub cylVolFixed()
    Dim radius As Double
    Dim height As Double
    Dim volume As Double
    radius = Range("B1").Value
    height = Range("B2").Value
    volume = PI * radius ^ 2 * height
    Range("B3").Value = volume
End Sub
Sub cylVolFixedShort()
    Dim sht As Worksheet
    Set sht = Worksheets("first")
    sht.Range("B3").Value = PI *sht.Range("B1") _
                .Value ^ 2 * sht.Range("B2").Value
    Set sht = Nothing
End Sub
```

Sub Solution for active sheet

Sub Solution for specific sheet

California State University Northridge 26

Excel Review

- Enter data and formulas into cells
- Formulas refer to other cells in relative and absolute manner
  - Relative references (*e.g.* =A8 in cell A9) copies in relative manner (always refers to cell in same column one row above)
  - Absolute references (*e.g.* =A8 in any cell) always refers to same cell
  - Can have row absolute (A$8) and column absolute ($A8) references

California State University Northridge 27

Notation

- When writing example commands
  - Text to be copied exactly as written is show as plain text: `If Then`
  - Items to be included that can change are written as brief description of the items enclosed in angle brackets: `<condition>`
  - Example: `If <condition> Then`
    - `If x < a Then`
    - `If c = d Than` ← Why is this one invalid?
    - `If x <> b AND y = e Then`

California State University Northridge 28

Functions

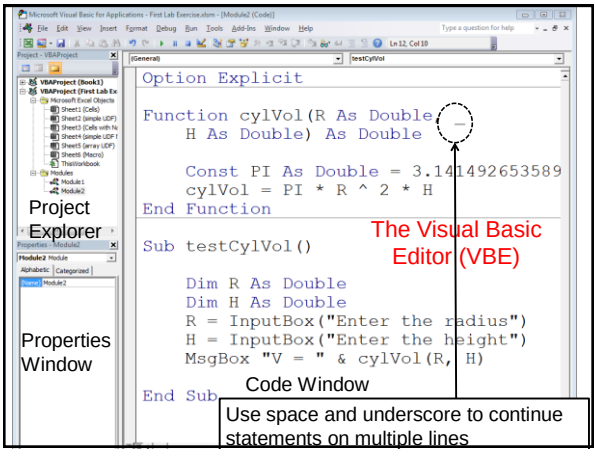
- Excel has over 400 worksheet functions
- With VBA code you can write your own functions to be called from Excel
  - Functions calls from worksheet have the form: =<Function Name>(<Argument List>)
    - *E.g.*: RAND(), SIN(B10), ATAN2(C12, R4), …
- Some VBA functions have different names from Excel functions
  - Square root is SQRT in Excel, SQR in VBA

California State University Northridge 29

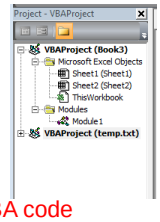
VBA Review

- Use of Visual Basic Editor
- Insert module to write code
- Two kinds of procedures: sub and function
  - Only difference is that function can return a value in the function name
  - Pass information among procedures by order of variables in argument list
    - Two way transfer of information possible in argument list for both function and subs

California State University Northridge 30

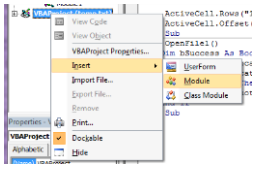


Project Explorer



- Project explorer lists all open workbooks with VBA name and Excel name
- Places to enter code can be hidden or shown by +/-
  - Note VBA name and Excel name for Excel Objects
- Always use modules to enter VBA code
  - Exception is for event-driven code on objects
  - Insert new module if there is not already one shown in the project explorer

Inserting a Module



- Select VBA Editor from the Developer tab or press Alt-F11
- Insert a module by right clicking on file name, selecting Insert | Module
- Code window will appear for new module
  - Can enter several functions and/or subs into one module
  - Can have more than one module in workbook

Declaring Variables

- Use Option Explicit statement at top of module to force variable declaration
  - Provides check on errors due to misspelled variables names (strong typing)
  - Submissions that do not do this will have automatic penalty of 10% of grade
  - Can set VBE options to do this for you
    - In VBE go to **Tools** | **Options** and check box next to **Require Variable Declaration**

Variable Data Types

Data Type	Memory Size	Range
Byte	1 byte	0 to 255
Boolean	2 bytes	TRUE (non-zero) or FALSE (0)
Integer	2 bytes	-32,768 to 32,767
Long (integer)	4 bytes	-2,147,483,648 to 2,147,483,647
Currency (scaled integer)	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Single	4 bytes	(+/-) 1.401298E-45 to 3.402823E38
Double	8 bytes	(+/-) 4.94065645841247E-324 to 1.79769313486232E308
String (fixed-length)	string length (bytes)	1 to ~65,400 characters
String (variable-length)	10 bytes + length	0 to about 2 billion characters
Date	8 bytes	time of day + dates between 1/1/100 to 12/31/9999

Shaded types are most common for engineering VBA

Variant Data Type in VBA

- Same as type for Excel cell – treats any kind of variable
- Requires 16 bytes to keep information on actual type of variable as well as its value (more bytes for longer strings)
- Default if variable is not assigned a specific type in a Dim statement
- Use only for special tasks in VBA –other data types are more efficient

Declaring Variables

- Use the following syntax

```
Dim x As Double
Dim k As Long
Dim s As String, d as Date
Dim y as Double, z As Double
```

- Do ***not*** use the following syntax

```
Dim x, y As Double
```

- In statement above x has default data type of Variant

Declaring Variables II

- Dim statements are usually placed at the start of a procedure
  - Can be placed anywhere before the first use of the variable
- Variables in procedure argument lists and functions are typed in headers

```
Sub test (x as double, y as double)
Function mine (d as Date) as Boolean
```


Variable Names

- Must start with letter and can only use letters, numbers and underscore (_)
- Use meaningful variable names
  - chamberPressure, mainStruttStress
    - Use capital letters to set of different words
- This is more work at start, but makes program easier to read and understand
  - Ease is especially important for other who use your program or for yourself when you use a program after some period of time

Arithmetic Operators

- Operators in order of precedence:
  - Exponentiation ^
  - Unary minus – (*E.g.* –x)
  - Multiply/Divide * /
  - Integer Division \
  - Remainder Mod (*E.g.* 7 Mod 4 = 3)
  - Addition/Subtraction + –
- Use parentheses to overrule normal rules of precedence
  - Compare x / y + z *vs* x / (y + z)

Exercise

- Write the following equations in VBA

$$r = \frac{3x^3 + 6x - 7}{4\sqrt{x} - 8}$$

- r = (3*x^3 + 6*x – 7) /(4*sqr(x) – 8)

$$p = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

- p = exp(-x*x/(2*sigma^2)) / sqr(2 * pi * _ sigma^2)

Symbolic Constants

- Useful way to program items that are constant or not expected to change often
- Syntax:

```
Const PI as Double = 3.14159265358979
```

- A const cannot be assigned a value in any other statement
- Use for mathematical and physical constants and infrequently changed items such as an overhead rate

Relational Operators

- Program logic requires choices based on expressions that are true or false
- Relational operators compare other variables and have true or false results
- The operators are <, <=, =, >, >=, <>
  - Usual definitions with <> as not equal
    - Literally <> is less than or greater than
  - 7 < 7 is false
  - 7 <= 7 is true

California State University Northridge 43

Equality Comparisons

- Real data types (single and double) may have slight errors due to round-off after calculations
  - If x = 1.000000000000 and y = 1.000000000001 the two are practically equal, but x = y is false
  - Test abs(x – y) <= eps * (abs(x) + abs(y)) where eps is a small number (≈ 1e-14 for type Double)
    - Consider x = y if expression is true

California State University Northridge 44

Logical Operators

- Combine true/false values
- Operators are Not, And, Or
- **x And y** is true if both x and y are true
- **x Or y** is true if either x or y is true
- **Not x** is true if x is false and is false if x is true (Not is like a unary –)
- Is the following true or false?

(6 < 3) Or ((12 > 2) And (Not (6<>6)))

(6 < 3 Or 12 > 2) And Not 6<>6

Precedence
Relational
Not
And
Or

California State University Northridge 45

More On Precedence

- Arithmetic operators come first with precedence shown previously
- String concatenation (&) comes next
- Relational operators follow with all operators having equal precedence
- Logical operators in order: Not, And, Or
- Operators of equal precedence are treated from left to right in order
- Parentheses always evaluated first

California State University Northridge 46

Exercise

- A year is a leap year if
  - It is evenly divisible by 4 **and** not evenly divisible by 100
  - **Or** it is evenly divisible by 400
- Y is evenly divisible by 4 if Y Mod 4 = 0
- Write an expression that is true if a variable, "year", is a leap year

((year Mod 4 = 0) and not(year mod 100 = 0)) or(year mod 400 = 0)

year Mod 4 = 0 and not year mod 100 = 0 or year mod 400 = 0

California State University Northridge 47

Variable Scope, Persistence

- Scope is where a variable is defined
- Persistence is how long it is defined
- Variables declared in a procedure have procedure scope and persistence
  - They are defined only for that procedure
  - They are reinitialized each time the procedure is called
  - They are called local variables
  - Use Static instead of Dim for persistence in a procedure between calls

California State University Northridge 48

Example of `Static`

```
Sub test()
  Dim count as Long
   count = count + 1
End Sub
```

Each time sub test is called the variable count is set to zero

```
Sub test2()
  Static count as Long
   count = count + 1
End Sub
```

Each time sub test2 is called the variable count starts with its value from the previous call

California State University Northridge 49

Scope and Persistence II

- Variables declared in declarations section of module (before first procedure) exist for all procedures in module
- For a workbook with multiple modules
  - Module variables in declarations section with Dim or Private have scope of module
  - Scope of global variables set in declarations section as Public is all modules
  - Limit use of module and global variables to provide code that is easier to understand

California State University Northridge 50

Another Example

- k and x are module variables (Dim and Private are equivalent)
- d is a global variable that could be see by routines in another module (not shown here)
- Note use of MsgBox for output shown below

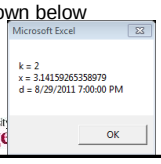
Note use of " _" for multiline statements

```
Option Explicit
Dim k as Long
Private x as Double
Public d as Date
Sub First()
   Call Second()
   MsgBox "k = " & k & _
     vbCrLf & "x = " & x & _
     vbCrLf & "d = " & d
End Sub
Sub Second()
   k = 2
   x = 4 * atn(1)
   d = #8/29/2011 7 pm#
End Sub
```

String constant

& is string concate-nation operator

vbCrLf moves output to new line



California State University Northridge 51

Type Conversion

- Last slide showed example of type conversion done automatically
  - **"k = " & k** converted numeric k to string
- When mixed types are used VBA tries to determine the conversion desired by the user and does it
- You can use a set of functions that do the conversions: CDbl(var), CLng(var), CStr(var), CInt(var), CBool(var)

California State University Northridge 52

First Exercise

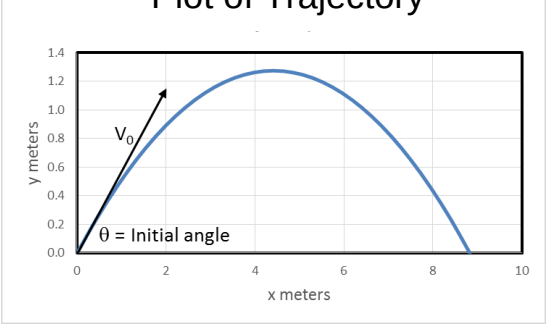
- Download from course web site
  http://www.csun.edu/~lcaretto/me309/pa1.pdf
- Compute x(t) and y(t) from t = 0 to t = t_{max} for a trajectory for given V_0 and θ

$$y = V_0 t sin\theta - \frac{gt^2}{2} \qquad x = V_0 t cos\theta \qquad t_{max} = \frac{2V_0 sin\theta}{g}$$

- Six different approaches using combinations of "A1" cell references, range names, user-defined functions, array function and macro

California State University Northridge 53

Plot of Trajectory



California State University Northridge 54

Example

- Shows results with cell formulas
- Uses SI units
- Use RADIANS function to get sine and cosine
- Time step = (max time) / (number of time steps)

	A	B	C	D
1	Initial velocity	10	m/s	
2	Initial angle	45	degrees	
3	g	9.80665	m/s^2	
4	Number of time intervals	20		
5	Maximum time	1.442096	s	
6	Time step	0.072105	s	
7				
8				
9		Time (s)	x (m)	y (m)
10		0	0	0
11		0.072105	0.509858	0.484365
12		0.14421	1.019716	0.917745
13		0.216314	1.529574	1.300138
14		0.288419	2.039432	1.631546
15		0.360524	2.549291	1.911968
16		0.432629	3.059149	2.141404
17		0.504734	3.569007	2.319854
18		0.576839	4.078865	2.447319
19		0.648943	4.588723	2.523798
20		0.721048	5.098581	2.549291
21		0.793153	5.608439	2.523798
22		0.865258	6.118297	2.447319
23		0.937363	6.628155	2.319854
24		1.009468	7.138013	2.141404
25		1.081572	7.647872	1.911968
26		1.153677	8.15773	1.631546
27		1.225782	8.667588	1.300138
28		1.297887	9.177446	0.917745
29		1.369992	9.687304	0.484365
30		1.442096	10.19716	0

California State University Northridge

Getting Started with UDFs

```
Option Explicit
Const g As Double = 9.80665    'Acceleration of gravity in m/s^2
Const degreesToRadians As Double = 3.14159265358979 / 180

Function maxTime(ByVal v0 As Double, ByVal theta As Double) _
   as Double
   'Computes time (in s)  for object to return to elevation y = 0
   'v0 is initial velocity in m/s and theta is initial angle in degrees
   maxTime = 2 * v0 * Sin(theta * degreesToRadians) / g
End Function
```

$$y = V_0 t sin\theta - \frac{gt^2}{2} \qquad x = V_0 t cos\theta \qquad t_{max} = \frac{2V_0 sin\theta}{g}$$

California State University Northridge

Write UDFs for y(t) and x(t)

Cell Names

- Can assign meaningful names to cells
- Names are used in formulas and show in name box when cell is selected
- Example below shows change with names

Worksheet Using Names

Density =Pressure/(Gas_Constant*Ten

	A	B	C
1		Ideal Gas Equation for Air	
2	Pressure	100	kPa
3	Temperature	300	K
4	Gas Constant	0.287	kJ/kg·K
5	Density	=Pressure/(Gas_Constant*Temperature)	kg/m^3

Without Names

B5 =B

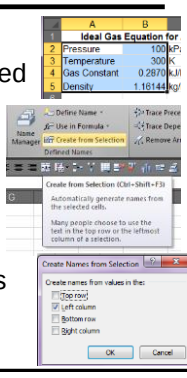
	A	B	C
1		Ideal Gas Equation for Air	
2	Pressure	100	kPa
3	Temperature	300	K
4	Gas Constant	0.287	kJ/kg·K
5	Density	=B2/(B4*B3)	kg/m^3

California State University Northridge

Default cell names are absolute

Getting Cell Names

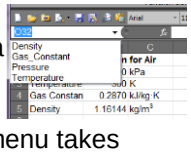
- Select cells with names adjacent to cells to be named
- Choose **Create from Selection** in **Defined Names** group of **Formula** tab
- In resulting dialog box make sure name location is correct and click **OK**



California State University Northridge

Name Box

- The name box, to the left of the formula bar, shows the currently selected cell
- When range names are defined, this box becomes a pulldown menu that shows all the defined names
- Clicking on a name in this menu takes you to the named cell
- Range names may be used as names for multiple cells



California State University Northridge

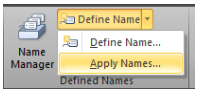
Name Manager

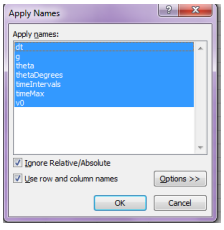
- Main tool for managing range names
  - Choose **Name Manager** in **Defined Names** group of **Formula** tab
  - Create, delete or edit names in Manager
- By default names have scope of entire workbook
- Name manager allows you to make name scope one worksheet
  - More instructions on first exercise

California State University Northridge

Names to Existing Formulas

- Click the down arrow next to **Define Names** and select **Apply Names** from the resulting submenu
- Select names in Apply names dialog and click OK
  - Can leave settings for checkboxes at bottom of dialog as set by Excel





California State University Northridge