

Graphic User Interfaces and Validation

CEE3804: Computer Applications for Civil and Environmental Engineers

Objectives

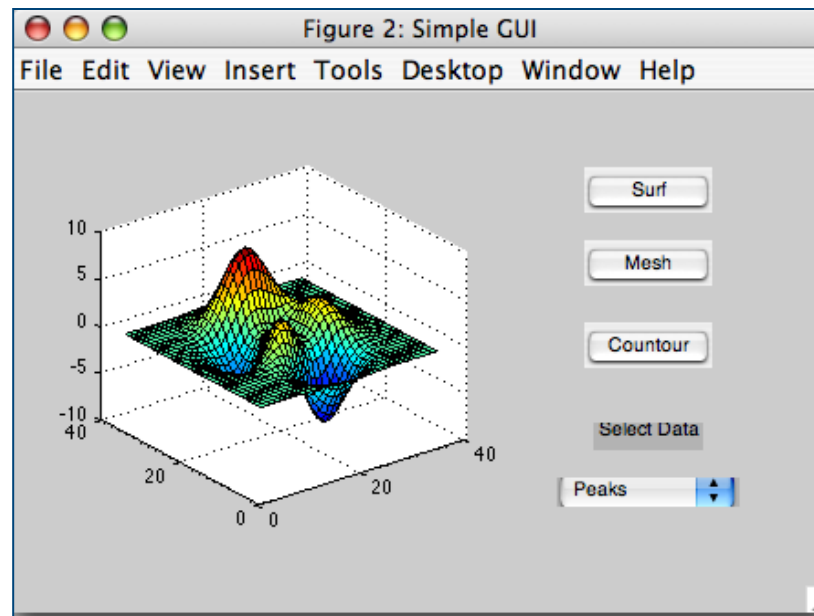
- **Apply good principles of data validation in spreadsheets**
- **Understand the principles of Graphic User Interfaces (GUI)**
- **Provide some examples to understand simple GUIs**

Why Graphic User Interfaces?

- Promotes better use of your programs by non-technical users or by your own colleagues
- Users do not have to be familiar with every line of code in VBA or Matlab to use your program
- GUIs can present only relevant information to a user

Sample GUI

Source: Mathworks



Techniques to Provide Ease of Use

- **In VBA**
 - Color code and data validation/entry
 - Types of controls
 - User Form controls
 - ActiveX controls
- **In Matlab**
 - Command window input queries
 - Use Matlab Graphic User Interface maker

VBA Techniques (Data Validation)

- A regular spreadsheet can direct a user to vary certain number of parameters in the program
- Use color coding and data validation

Runway_capacity.xls

File on the web server

	A	B	C	D	E	F	G
1	Runway Saturation Capacity Estimation						
2	Using the Analytical Model of Harris						
3							
4	Programmer: A. Trani (January 2002)						
5	Amendments:	1	7-Apr-03		Corrected formula to estimate	E(delta j)	
6							
7							
8	Technical Parameters (inputs)				Parameter	Values	
9	Dep-Arrival Separation (nm)				δ	2	
10	Common Approach Length (nm)				γ	6	
11	Standard deviation of Position Delivery Error (s)				σ	18	
12	Probability of Violation				Pv	5	
13	Cumulative Normal at Pv				qv	1.65	
14							
15		TERP A	TERP B	TERP C			
16	ROT (s)	43	48	59	47.7	E(ROT)	
17	Percent Mix	50	30	20	100	Total %	
18	Vapproach (kn)	110	140	145			
19							
20	Minimum Separation Matrix (nm)			Arrivals-Arrivals		Airport Type	

VBA Techniques (Data Validation)

- This program allows certain inputs to the user from a list of inputs (check the Common Approach Length parameter - Gamma)
- This reduces possible data entry errors

	A	B	C	D	E	F	G
1	Runway Saturation Capacity Estimation						
2	Using the Analytical Model of Harris						
3							
4	Programmer: A. Trani (January 2002)						
5	Amendments:	1	7-Apr-03		Corrected formula to estimate	E(delta j)	
6							
7							
8	Technical Parameters (inputs)				Parameter	Values	
9	Dep-Arrival Separation (nm)				δ	2	
10	Common Approach Length (nm)				γ	6	
11	Standard deviation of Position Delivery Error (s)				σ		
12	Probability of Violation				P		
13	Cumulative Normal at P _v				Q		
14							
15		TERP A	TERP B	TERP C			
16	ROT (s)	43	48	59			
17	Percent Mix	50	30	20	100 Total %		
18	V _{approach} (kn)	110	140	145			

VBA Data Validation

- Use the Data validation to limit the entry of numbers into your program

The screenshot shows the Excel Data Validation dialog box with the following settings:

- Settings tab selected
- Validation criteria: Allow: List, Data: between, Source: 1,2,3,4
- Ignore blank and In-cell dropdown are checked
- Apply these changes to all other cells with the same settings is unchecked

The background shows a spreadsheet with a table of parameters and values. The 'Values' column has a dropdown menu with the number 2 selected. The 'Parameter' column lists δ , γ , σ , P_v , and q_v . The 'Values' column lists 2, 0, 18, 5, and 1.65. The cell containing the dropdown menu is highlighted with a blue box and labeled with the number 1. The Data Validation dialog box is labeled with the number 3. The Data Validation icon in the ribbon is labeled with the number 2.

Parameter	Values
δ	2
γ	0
σ	18
P_v	5
q_v	1.65

Why is Data Validation Important?

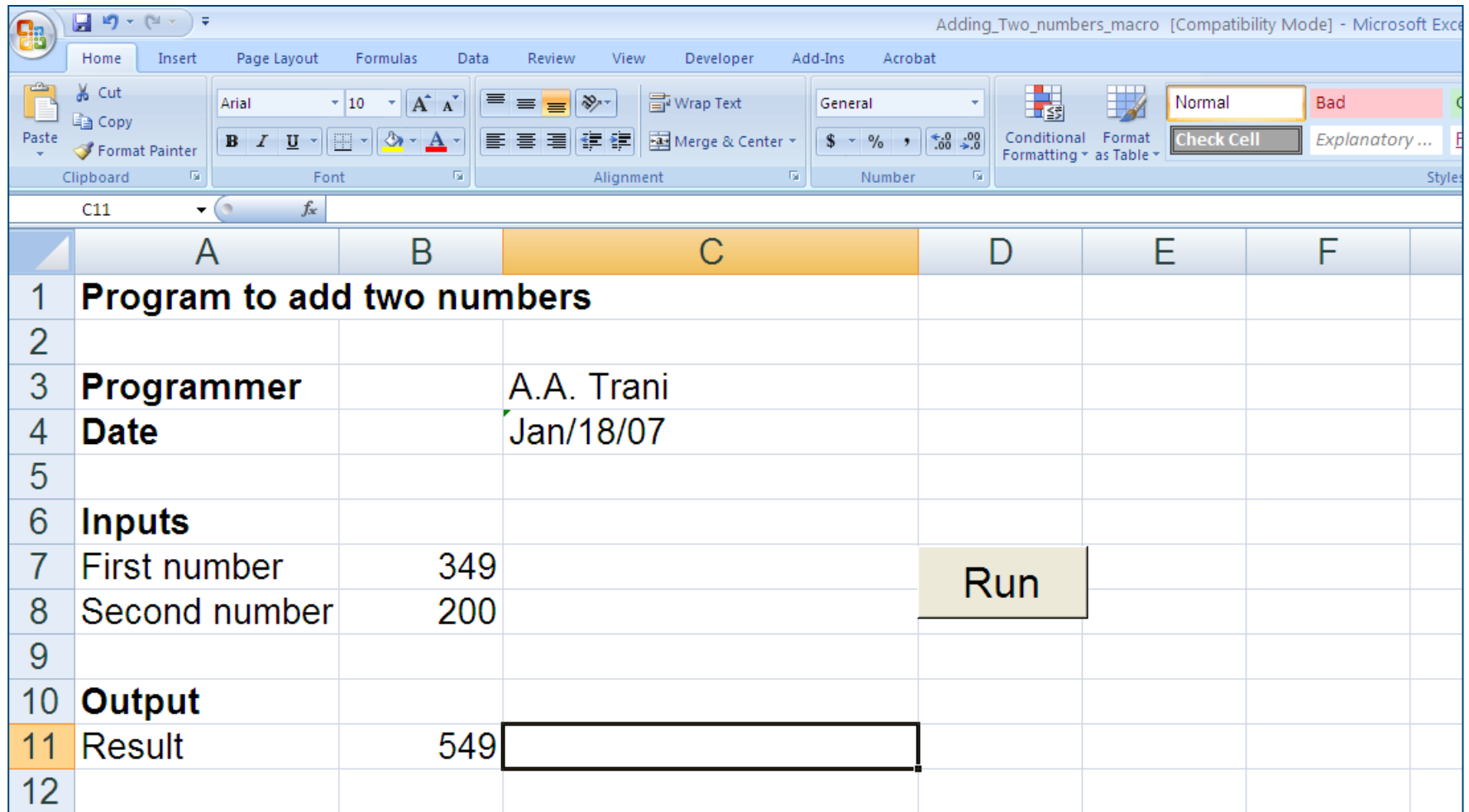
- **Reduces costly (in design and economics) entry data errors**
- **Simplifies the use of the Excel spreadsheet**
- **Make the program more “professional” looking**

- **Add comments to the spreadsheet to communicate the steps in the calculations if possible**
- **Adding a comments goes a long way to explain your spreadsheet to others**

VBA Adding Simple Controls

- **Two types of controls in VBA**
- **ActiveX**
 - **Simpler to use**
 - **Good for quick prototyping**
- **Form Controls**
 - **Require more information (code)**
 - **Better looking**
 - **More powerful**

Excel Worksheet (final program)

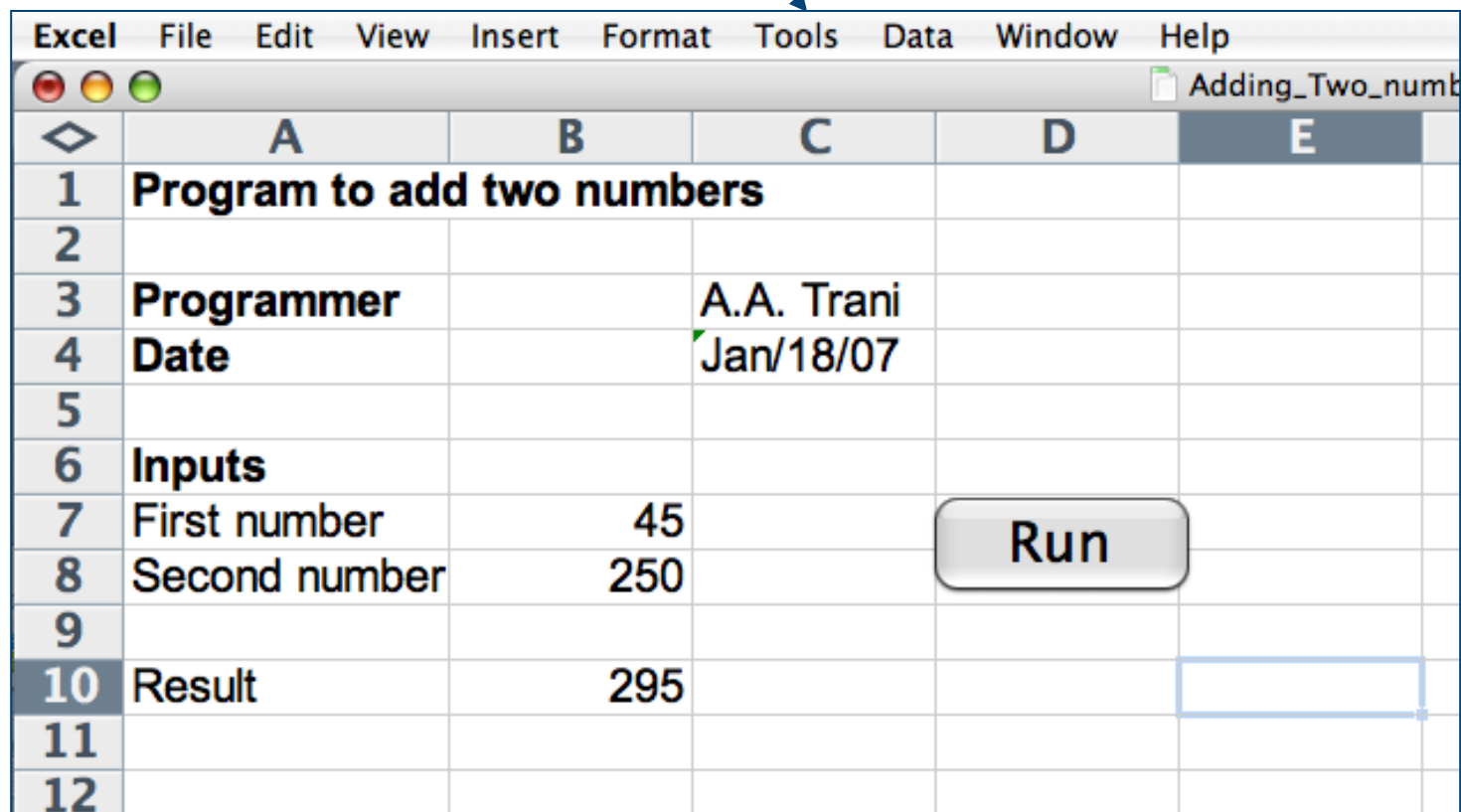


The screenshot shows an Excel worksheet titled "Adding_Two_numbers_macro [Compatibility Mode] - Microsoft Excel". The ribbon includes Home, Insert, Page Layout, Formulas, Data, Review, View, Developer, Add-Ins, and Acrobat. The worksheet has columns A through F and rows 1 through 12. The data is as follows:

	A	B	C	D	E	F
1	Program to add two numbers					
2						
3	Programmer		A.A. Trani			
4	Date		Jan/18/07			
5						
6	Inputs					
7	First number	349				
8	Second number	200				
9						
10	Output					
11	Result	549				
12						

VBA or Macros in Excel

- Go to: Tools/Macro
- Then select VBA



VBA Code

```
Sub Adder ()  
    ' Programmer : Toni Trani  
    ' Date: 09/13/07  
  
    Sheets("Sheet1").Select  
  
    Range("B7").Select  
    a = ActiveCell.Value  
  
    Range("B8").Select  
    b = ActiveCell.Value  
  
    c = a + b  
  
    Range("B11").Select  
    ActiveCell.Value = c  
  
End Sub
```

Name of subroutine

Go to "sheet1"

Select the value of cell B7
and assign it to variable "a"

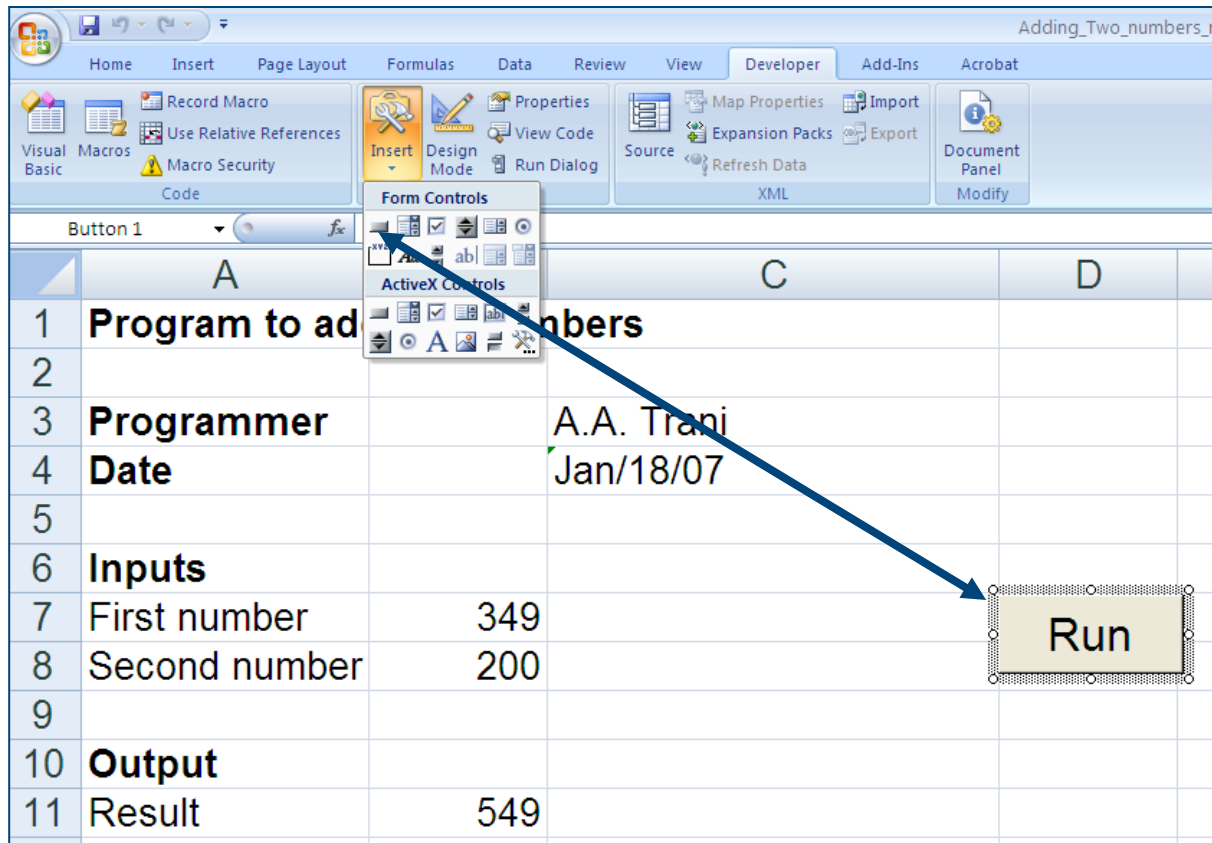
Select the value of cell B8
and assign it to variable

"b"
Calculate the value of "c"

Select cell B8 and assign it
to variable "c"

Adding the “Run” Button

- Adding the “run” button requires that you insert a control button and then assign a macro to subroutine “adder”



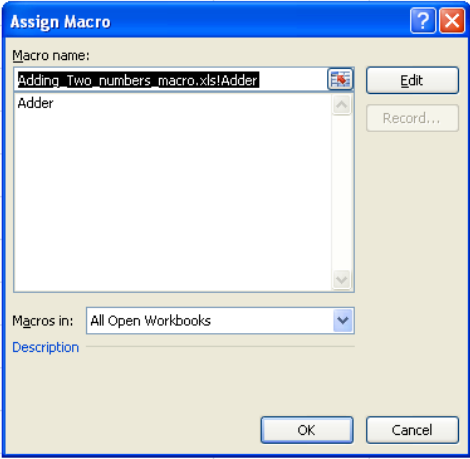
The screenshot shows the Microsoft Excel interface with the Developer tab selected. The ribbon includes options like Record Macro, Use Relative References, Macro Security, Insert, Design Mode, Run Dialog, Properties, View Code, Map Properties, Import, Expansion Packs, Export, Document Panel, and Modify. A 'Form Controls' task pane is open, showing various control options. A blue arrow points from the 'Form Controls' pane to a yellow 'Run' button placed in cell D7 of the spreadsheet. The spreadsheet contains the following data:

	A	B	C	D
1	Program to add		numbers	
2				
3	Programmer		A.A. Trani	
4	Date		Jan/18/07	
5				
6	Inputs			
7	First number	349		Run
8	Second number	200		
9				
10	Output			
11	Result	549		

Assigning the Behavior to the “Run” Button

- Assign the macro “adder” from the list of available macros to button “Run”
- Right-click on the button to assign a behavior

Program to add two numbers		
Programmer	A.A. Trani	
Date	Jan/18/07	
Inputs		
First number	349	Run
Second number	200	
Output		
Result	549	



Previous Example (kicker_with_angleTable_2003.xls)

- Kicker program with “Run” button assigned to a macro

	A	B	C	D	E
2	Kick Calculator		Chapter 5		
3			S. Chapra's Book		
4	Programmer: A. Trani				
5	Date	11/29/2007 10:38			
6	Purpose:	Calculate range and hang time			
7					
8	Parameter	Value	Units		
9	Initial Speed		30 m/s		
10	Initial Angle		50 degrees		
11					
12	Results				
13	Hang Time		4.685 seconds		
14	Range		90.349 meters		

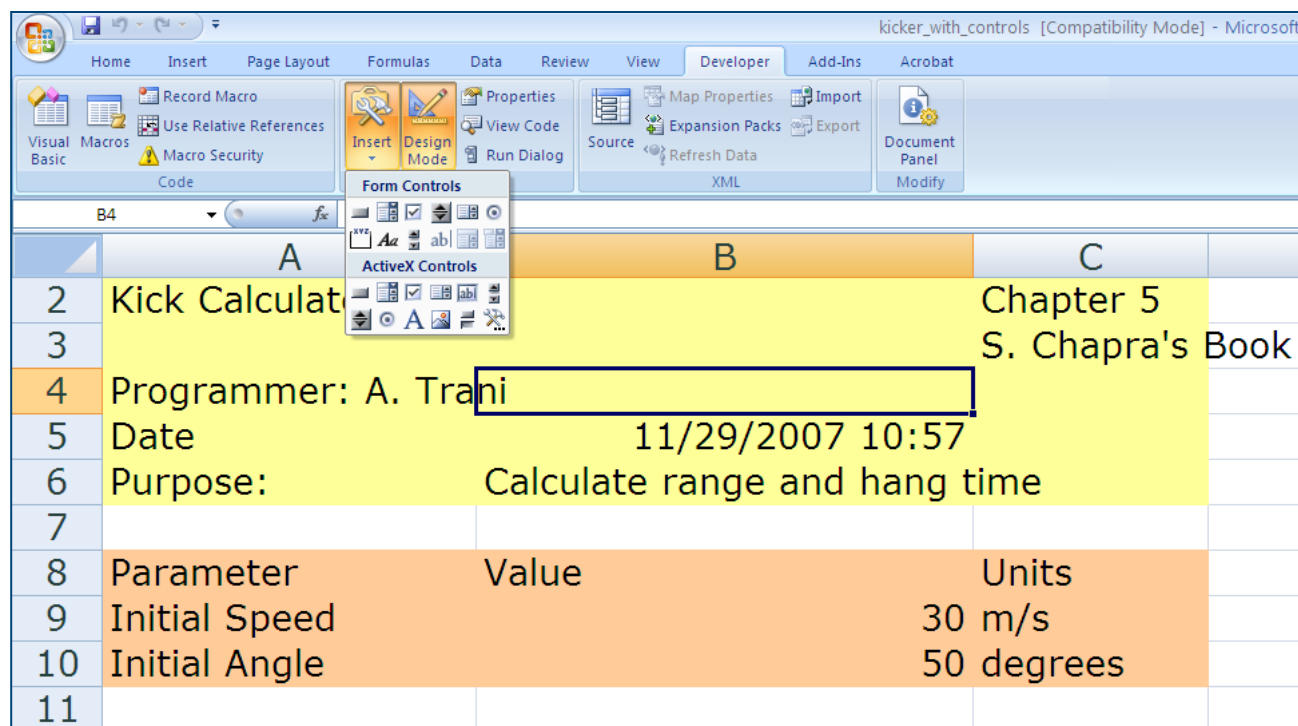
Run "Kick"

Suppose We Add a New Control

- Lets add a slider control to set the values of initial angle and initial speed in this program
- Procedure:
- Go to the “Developer” tab in Excel
- Go to the “View” toolbox (then forms) in older versions of Excel

Adding a New Control (Form Control)

- Check all controls available and add a simple scrollbar control to the “kicker” spreadsheet



The screenshot shows the Microsoft Excel interface with the Developer tab selected. The Form Controls task pane is open, displaying various control options. The spreadsheet content is as follows:

	A	B	C
2	Kick Calculat		Chapter 5
3			S. Chapra's Book
4	Programmer: A. Trani		
5	Date	11/29/2007 10:57	
6	Purpose:	Calculate range and hang time	
7			
8	Parameter	Value	Units
9	Initial Speed		30 m/s
10	Initial Angle		50 degrees
11			

VBA Control Example

- The Control is linked to a cell in your spreadsheet

The screenshot displays an Excel spreadsheet titled 'kicker_with_controls [Compatibility Mode] - Microsoft Excel'. The spreadsheet is organized into sections: a title section (rows 2-4), a date and purpose section (rows 5-6), a parameter table (rows 8-10), a results section (rows 12-14), and a data table (rows 16-22). A 'Run "Kick"' button is located in cell E6. A scrollbar control is positioned over cell C9, and the 'Format Object' dialog box is open, showing the 'Control' tab with various parameters.

Parameter	Value	Units
Initial Speed	30	m/s
Initial Angle	50	degrees

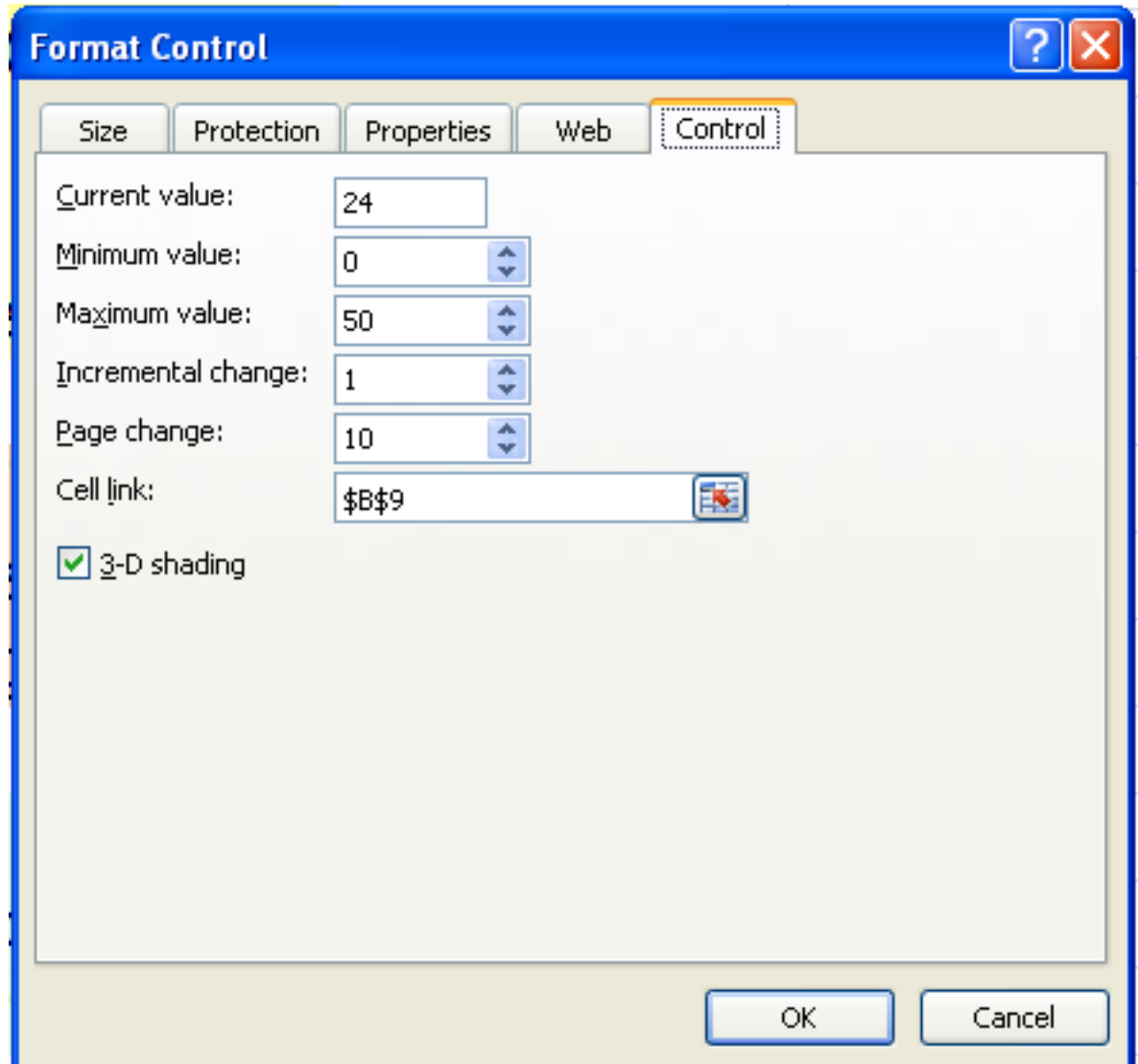
Angle (degrees)	Hangtime (seconds)	Distance (m)
0	0.00	00.00
5	0.53	15.93
10	1.06	31.38
15	1.58	45.87
20	2.09	58.97
25	2.58	70.28

Control (scrollbar)

Control Parameters

VBA Controls : Specify the Parameters of the Control

- Enter the values of the control
- Link the control to a cell (or cells) in your spreadsheet



Add Other Controls to Ease Use of the Spreadsheet

- Add text controls to specify the range of your control to the user
- In this case I added 3 text controls as labels

The screenshot shows an Excel spreadsheet with the following content:

	A	B	C	D	E
2	Kick Calculator		Chapter 5		
3			S. Chapra's Book		
4	Programmer: A. Trani				
5	Date	11/29/2007 11:41			
6	Purpose:	Calculate range and hang time			Run "Kick"
7					
8	Parameter	Value	Units		
9	Initial Speed		24 m/s	<input type="text" value="24"/>	
10	Initial Angle		50 degrees	<input type="text" value="50"/>	
11				0.0	50.0
12	Results			Speed (m/s)	

Adding a New Worksheet as GUI (Graphic User Interface)

- Lets add a new worksheet that acts as Graphic User Interface
- Lets add information about the problem
- Lets add a new control (a combo box) to control the initial angle (a_i)
- The values of the controls should be stated in the GUI interface
- Lets add a new worksheet that contains the values of the combo box for initial angle

Final GUI for Kicker Problem

1 **The Kicker Problem**

2

3 This problem calculates the hang time and the distance
4 traveled by a ball hit with an initial
5 speed (v_i) (in m/s) and initial angle (α_i) (in degrees)
6

7 In the interface below, select the
8 values of (v_i) and (α_i) wanted

9 **Inputs to the Problem**

10 Selected Speed (m/s)

11 0.0 50.0

12 Speed (m/s)

13

14 22

15 Initial Angle (degrees)

16 **Outputs to the Problem**

17 Hang Time 2.062 seconds

18 Range 51.621 meters

19

20

21

22

23

24

GUI For Controls Interface Values for Initial Angle

Added Worksheet

(to show
controls)

Original Worksheet

Added Worksheet

(to store values
of initial angle)

Scrollbar Control for GUI

The Kicker Problem

This problem calculates the hang time and the distance traveled by a ball hit with an initial speed (v_i) (in m/s) and initial angle (α_i) (in degrees)

In the interface below, select the values of (v_i) and (α_i) wanted

Inputs to the Problem

Selected Speed (m/s): 27

Speed (m/s): 0.0 to 50.0

Initial Angle (degrees): 22

Run "Kick"

Outputs to the Problem

Hang Time	2.062	seconds
Range	51.621	meters

Added Scrollbar Control
(to show the initial speed)

GUI For Controls: Interface, Values for Initial Angle

Combo Box Control for GUI

The Kicker Problem

This problem calculates the hang time and the distance traveled by a ball hit with an initial speed (v_i) (in m/s) and initial angle (α_i) (in degrees)

In the interface below, select the values of (v_i) and (α_i) wanted

Inputs to the Problem

Speed (m/s) Selected Speed (m/s)

Initial Angle (degrees)

Outputs to the Problem

Hang Time 2.062 seconds
Range 51.621 meters

Run "Kick"

Added Combo Box Control (to show the initial angle)

GUI For Controls Interface Values for Initial Angle

Combo Box Connection for GUI

The screenshot shows a MATLAB GUI titled "The Kicker Problem" with the following components:

- Section 1: The Kicker Problem** (rows 1-8):
 - Row 1: **The Kicker Problem**
 - Row 2: (empty)
 - Row 3: This problem calculates the hang time and the distance
 - Row 4: traveled by a ball hit with an initial
 - Row 5: speed (v_i) (in m/s) and initial angle (a_i) (in degrees)
 - Row 6: (empty)
 - Row 7: In the interface below, select the
 - Row 8: values of (v_i) and (a_i) wanted
- Section 2: Inputs to the Problem** (rows 9-15):
 - Row 9: **Inputs to the Problem**
 - Row 10: A slider control for speed, with a value of 27 displayed.
 - Row 11: 0.0 50.0 Selected Speed (m/s)
 - Row 12: Speed (m/s)
 - Row 13: (empty)
 - Row 14: A dropdown menu for initial angle, with 22 selected.
 - Row 15: Initial Angle (degrees)
- Section 3: Outputs to the Problem** (rows 16-18):
 - Row 16: **Outputs to the Problem**
 - Row 17: Hang Time 2.062 seconds
 - Row 18: Range 51.621 meters
- Section 4: Run Button** (row 17): A button labeled "Run 'Kick'" is positioned between the input and output sections.

Connect Combo Box with worksheet (Values for Initial Angle)

The screenshot shows a worksheet with the following data:

	A	B	C	D
1	Initial angle values (degrees)			
2	1			
3	2			
4	3			
5	4			
6	5			
7	6			
8	7			
9	8			
10	9			
11	10			
12	11			
13	12			
14	13			
15	14			
16	15			
17	16			
18	17			
19	18			
20	19			
21	20			
22	21			
23	22			
24	23			
25	24			
26	25			
27	26			
28	27			
29	28			
30	29			
31	30			
32	31			
33	32			
34	33			
35	34			
36	35			
37	36			
38	37			
39	38			
40	39			
41	40			
42	41			
43	42			
44	43			
45	44			
46	45			
47	46			
48	47			
49	48			
50	49			
51	50			
52				
53				
54				
55				

Display Outcomes in the GUI

The Kicker Problem

This problem calculates the hang time and the distance traveled by a ball hit with an initial speed (v_i) (in m/s) and initial angle (a_i) (in degrees)

In the interface below, select the values of (v_i) and (a_i) wanted

Inputs to the Problem

Speed (m/s): 0.0 to 50.0, Selected Speed (m/s): 27

Initial Angle (degrees): 22

Run "Kick"

Outputs to the Problem

Hang Time	2.062	seconds
Range	51.621	meters

Model Output

GUI Resources

- **AskTog — Essays on good design and a list of First Principles for good user interface design. The author, Tognazzini, is a well-respected user interface designer. <http://www.asktog.com/basics/firstPrinciples.html>**
- **Galitz, Wilbert, O., Essential Guide to User Interface Design. Wiley, New York, NY, 2002.**
- **GUI Design Handbook — A detailed guide to the use of GUI controls. http://www.fast-consulting.com/GUI%20Design%20Handbook/GDH_FRNTMTR.htm.**

GUI Resources (More)

- **Usability Glossary** — An extensive glossary of terms related to GUI design, usability, and related topics. <http://www.usabilityfirst.com/glossary/main.cgi>.
- **UsabilityNet** — Covers design principles, user-centered design, and other usability and design-related topics. http://www.usabilitynet.org/management/b_design.htm