AC 2007-397: USE OF SPREADSHEETS WITH SCALED GRAPHICS TO TEACH STRUCTURAL ENGINEERING

Jorge Tito-Izquierdo, University of Houston-Downtown

Alberto Gomez-Rivas, University of Houston-Downtown

Alberto Gomez-Rivas is Professor of Structural Analysis and Chair of Engineering Technology. Dr. Gomez-Rivas received Ph.D. degrees from the University of Texas, Austin, Texas, in Civil Engineering and from Rice University, Houston, Texas, in Economics. He received the Ingeniero Civil degree, with Honors, from the Universidad Javeriana in Bogotá, Colombia. He also served as Chief of Colombia's Department of Transportation Highway Bridge Division

Use of Spreadsheets with Scaled Graphics to Teach Structural Engineering

Abstract

Engineering is a profession where graphical presentation is very important in understanding and verifying results. Geometric proportions, spacing, and other features can be clearly perceived if a scaled graph is displayed together with the calculations, and thus, the engineering student can make better decisions about the final design. Overall, these spreadsheets with graphical capabilities help the learning process.

The use of Excel spreadsheets in engineering education and professional practice is frequent because this tool is versatile and powerful. However, a deficiency of spreadsheets is the lack of graphic representation. This may be solved by using the programming tools of Visual Basic for Applications® (VBA) that is included in the package of Microsoft Office®. This paper presents a method to include scaled graphs into a spreadsheet, which completes the engineering calculations and helps in the final decision to accept or modify a design. These graphs are also useful in the drafting process, because the graphs can be easily transferred into any computer assisted drafting (CAD) program. Several actual class examples are included.

Introduction

Spreadsheets are frequently used as helpful tools during the structural design process, especially when the final outcome involves assumptions that require verification including design of beams, columns, channel sections, retaining walls, and footings, between others. Engineering is a profession where graphical presentation is very important in understanding and verifying results. Also, geometric proportions, spacing, and other features can be clearly perceived if a scaled graph is displayed on the spreadsheet, and thus, the engineer can make better decisions about the final design.

The use of Microsoft Excel® spreadsheets in engineering education and professional practice is frequent because this tool is versatile and powerful. However, a deficiency of spreadsheets is the lack of geometric presentation graphics and therefore, understanding of the results. This may be solved by using the programming tools of Visual Basic for Applications (VBA) that is included with the standard package of Microsoft Office®[1].

This paper outlines one method to include scaled graphs into a spreadsheet; this graph presentation completes the engineering calculations and helps in the final decision to accept or modify a design. These graphs are also useful in the drafting process, because the graphs can be copy-pasted into any computer assisted drafting (CAD) program. Several actual class examples are included.

Prior to running the VBA program, the drawing data is listed and organized on the spreadsheet. VBA can then be accessed using a command button. The program follows a routine, consisting of reading the geometric data and desired scale, the location where the drawing will be displayed, application of a suitable scale factor to the geometric data, drawing the geometry, and applying any special effects such as color or lines.

Spreadsheets with graphic capabilities improve the learning curve of the students, principally for topics involving design. Furthermore, these spreadsheets have a component of trial and error, since modification of input parameters and obtaining a final modified result is simple, quick, and straightforward. Thus, the use of Excel spreadsheets with subsequent VBA processing has been found by the authors to be an effective and useful tool in facilitating student learning in structural design courses.

Spreadsheet Organization

The spreadsheet may be organized according to the necessity of the problem; generally it is organized as input, partial calculations, results and graphics.

The students are encouraged to prepare a template in order to organize the presentation just once. This template may be used for different purposes. The template provides the typical information required for professional calculation sheets. A good printout, with the corresponding references, sketches and commentaries is a requisite for future use of the spreadsheet.

Sketches, where the variables are described, still may complement the input process as shown in Figure 1. This sketch is done using the drawing tools of excel, without scale. The sketch may be improved using the VBA tools "text box" or "label", which can be linked to the data input in the corresponding cell. Figure 2 describes a method to include the input data into the sketch.

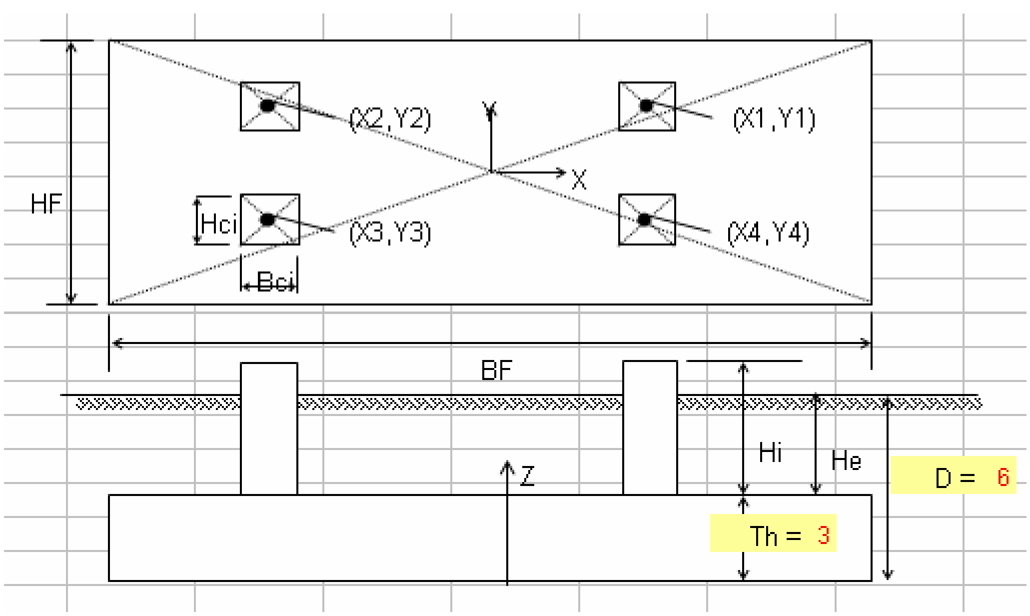
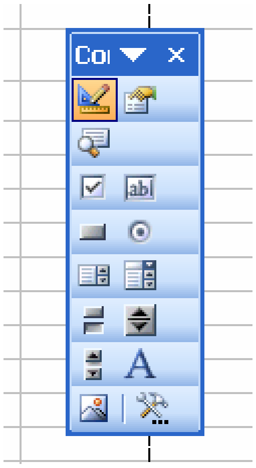
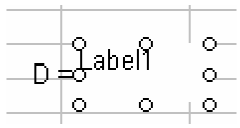
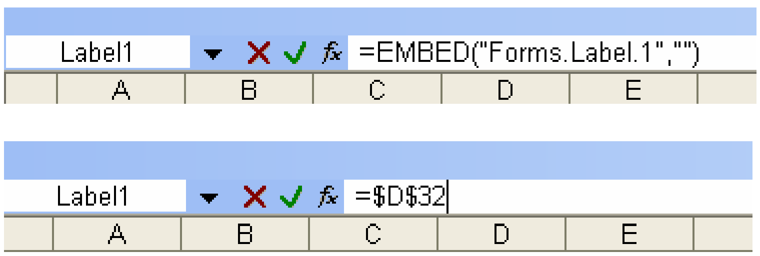


Figure 1. Sketch used to describe the variables.



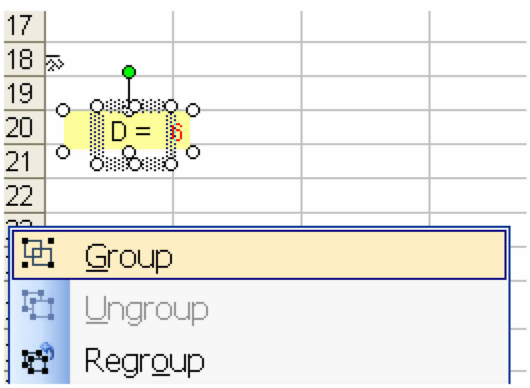


a. From the toolbox of VBA select label or text box. Drag the icon to the sketch, together with the description.



b. In the ‘formula bar’ of excel replace the default command (EMBED…..) by the cell correspondent to the variable to be described.

c. The VBA tool and the description can be grouped for clarity. If the description is back of the tool, use the ‘send to front’ excel tool of ‘Draw’.



d. Group both the VBA tool and the description.

Figure 2. Sequence to Include the Input Data into the Sketch

The input data is located in one block, marked with clear color because the print out in black and white is clearer. The cells used for input shall be un-locked permitting the use of a protection for all the spreadsheet, except the input cells.

Intermediate calculations are performed using Excel functions. VBA becomes more effective when the calculations involve iterations or a significant amount of conditionals.

The problem conclusions may be presented using the capabilities of Excel, presenting programmed comments if the assumptions satisfy the design criteria.

Finally, and this is the main objective of this paper, the spreadsheet can present a scaled drawing of the problem. If the drawing has too much information, it is better to present a new sheet exclusive for the drawing. In the new spreadsheet there is no input, all the information respect to the problem is automatically imported from the calculation sheet. Only the drafting related information is manipulated on this sheet, such as scale and location of the drawing.

Drawing using VBA

Scale definition. Create a scale options for the user. The Excel's "Data Validation" tool may be used to select the required scale for the drawing. This tool is found in Data – Validation – Setting – List, as shown in Figure 3.

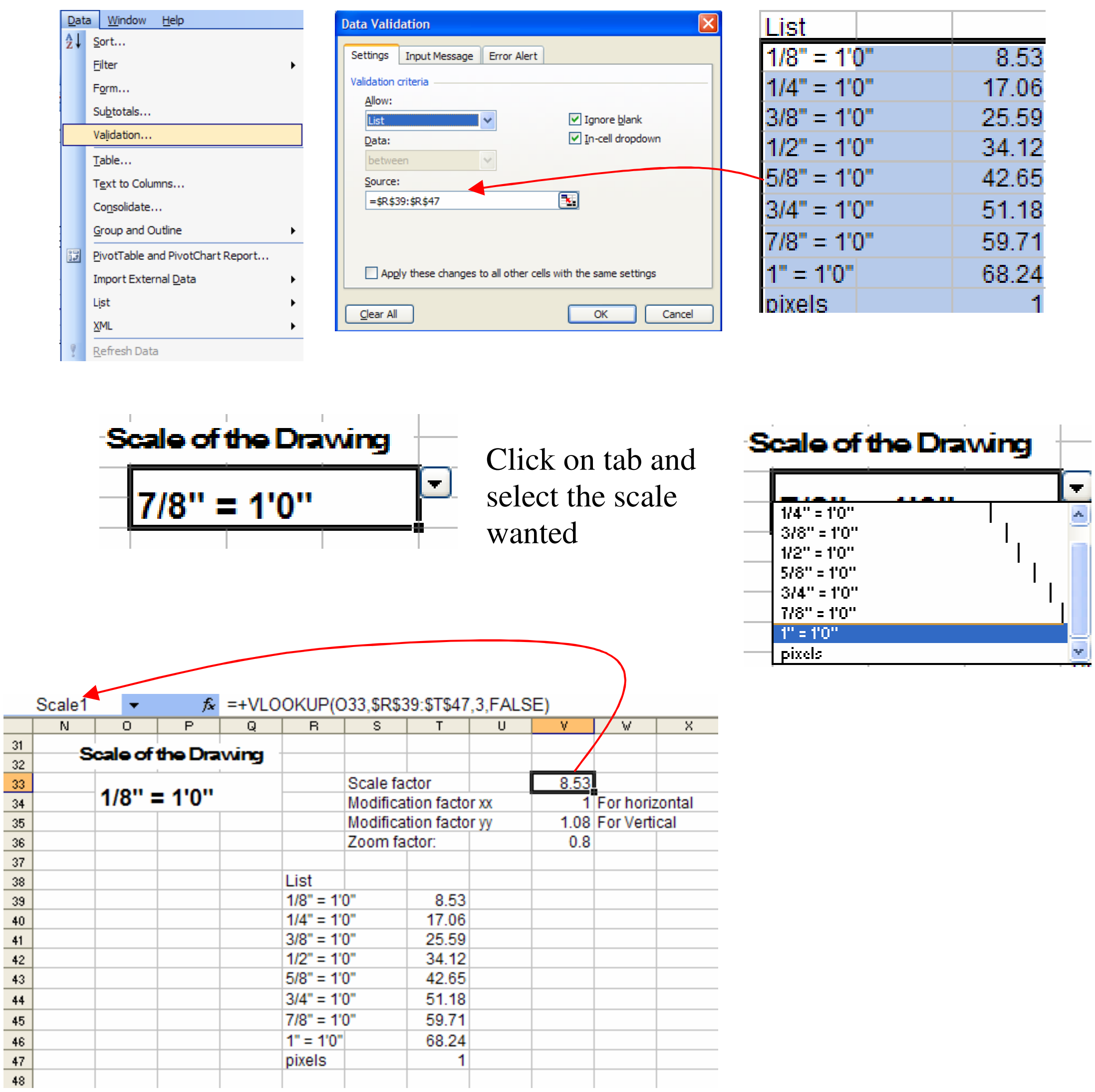


Figure 3. Scale selection

When the scale is selected, the program employs the "vlookup" function and opts for a factor that is used to scale the coordinates, as indicated in Figure 3. The user adjusts the modification factors for horizontal and vertical dimensions accordingly to the printer available. The zoom factor depends on the percentage of the normal size selected in the page setup for the spreadsheet and it can be automatically found using the VBA expression "ActiveSheet.PageSetup.Zoom / 100". The scale factor, the modification factors and the zoom factor are used to scale the point coordinates useful for drawing.

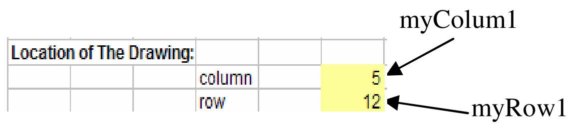
Delete Old Drawings. This control button is necessary to clear the spreadsheet of old lines, circles, or any other drawing. The VBA code is shown in Figure 4. This Delete button shall be activated (clicked) before the drawing of the final sketch.

```
Private Sub cmdEraser_Click()
Set myShapes = ActiveSheet.Shapes
   'Delete Polylines
   For Each shp In myShapes
      If Left(shp.Name, 8) = "Freeform" Then
         shp.Delete
      End If
   Next

   'Delete Lines
   For Each shp1 In myShapes
      If Left(shp1.Name, 4) = "Line" Then
         shp1.Delete
      End If
   Next

      'Delete Rectangles
   For Each shp1 In myShapes
      If Left(shp1.Name, 9) = "Rectangle" Then
         shp1.Delete
      End If
   Next
```

```
      'Delete Shapes
   For Each shp1 In myShapes
      If Left(shp1.Name, 9) = "AutoShape" Then
         shp1.Delete
      End If
   Next

      'Delete Text Box
   For Each shp1 In myShapes
      If Left(shp1.Name, 8) = "Text Box" Then
         shp1.Delete
      End If
   Next

      'Delete oval
   For Each shp1 In myShapes
      If Left(shp1.Name, 4) = "Oval" Then
         shp1.Delete
      End If
   Next
End Sub
```

Figure 4. VBA Code to Delete Old Drawings in the Spreadsheet

Drawing Location. The drawing can be located in the place where better adjust for the user. VBA defines the origin of coordinates in the upper left corner of the screen, the x-axis is positive toward right and the y-axis is positive downward. It is possible to translate the origin to other position using the columns and rows of the spreadsheet. The width of each column and the height of each row are defined in "points", each point is 1/72 inch, and they may be found using VBA codes. The code given in Figure 5 helps to move the origin of coordinates to the desired position.



Column and Row desired for the origin of coordinates for the drawing

```
'Locate the position of (0,0)
'PositionX and PositionY are the new coordinates in points of the origin for the drawing
PositionX = 0
PositionY = 0
myColumn = ActiveSheet.Range("myColumn1")    'Read the Location desired for the Drawing
myRow = ActiveSheet.Range("myRow1")          'Read the Location desired for the Drawing
'Loop to find the value in points of origin to be used for the drawing
For I = 1 To myColumn
    PositionX = PositionX + ActiveSheet.Columns(I).Width
Next I
For I = 1 To myRow
    PositionY = PositionY + ActiveSheet.Rows(I).Height
Next I
```

Figure 5. Excel Input Cells and VBA Code to Find the Origin of Coordinates for the Drawing

Reading Data. Using the standard commands of VBA, the input data is read from the spreadsheet and stored in previously dimensioned VBA variables. It is preferable to define with explicit names the cells to be used as input, as this permits future changes in the spreadsheet without affecting the original programming. For example, the cell V33, which defines the scale factor, may be called "Scale1" in the spreadsheet using the function "insert-name-define", as shown in Figure 3. Note that the name of the cells is case-sensitive.

Table 1. Data Needed to Draw Objects in VBA

Object	*Coordinates of*	*Dimensions*	*Code*
Line	Start and End	--	Shapes.AddLine Xstart,Ystart,Xend,Yend
Rectangle	Lower left corner. 'Y' is positive downward	Width and Height	Shapes.AddShape msoShapeRectangle, X, Y, Width, Height
Oval or circle	Center	Width and Height	Shapes.AddShape msoShapeOval, X, Y, Width, Height
Polyline	Vertices of the polygon. It may be opened or closed. The coordinates shall be stored in an array		Dim CoordPoints(1 To M, 1 To N) As Single CoordPoints (1, 1) = a CoordPoints (1, 2) = b …. CoordPoints (2, 1) = c CoordPoints (2, 2) = d …. CoordPoints (M, 1) = e …. CoordPoints (M, N) = f Shapes.**AddPolyline** CoordPoints

Drawing. The dimensions corrected by the scale and modification factors, used for the scaled drawing, shall be assigned to different variables in order to maintain the original dimensions for other purposes. The drawing may be performed using different geometric options. It may be used different objects, like circles, rectangles, polylines, lines, text boxes, etc. It is important to define the coordinates of the shape respect to the origin of the drawing previously defined, remembering that VBA always use the positive x-axis toward the right and the y-axis downward of the screen. The VBA codes have options to change the line width, color, continuity, and other attributes of the object. VBA has specific codes to draw each one of the objects. Table 1 shows the coordinates needed to draw some objects. VBA has a help function that is a good start for the code writing.

Examples

Footing Design. In this example, the dimension of a footing is input to the excel spreadsheet to verify the soil capacity using the classical methods of foundation engineering[2]. After the verification of the soil pressures, the foundation is drawn using the VBA code in another spreadsheet. This combination of analysis and drafting allows the student to visualize the problem and the solution found, appreciating the effect of the different parameters, like a virtual laboratory. Figure 6 shows the spreadsheet used to verify the foundation. This spreadsheet computes the geometric properties of the footing, combine the loads, translate the loads from the top of the pedestal to the bottom of the footing, computes the soil pressure in the four corners of the footing, and compare the allowable soil bearing pressure with the actual soil stress.

Figure 7 shows the scaled drawing of the different views of the footing, which is done using VBA; this drawing can be exported to other Computed Assisted Drafting (CAD) programs. Figure 8 shows the VBA programming code to make the drawing. The programming code is divided in data reading, location of the origin, and drawing, as explained before.

Reinforced Concrete Design – Beams. This spreadsheet permits the user verify a given cross section of a rectangular or T-beam, including the longitudinal steel and stirrups. The spreadsheet is based on the ACI-318-05[3]. The user can input the geometry with the help of sketches and the steel dimensions for each layer of reinforcement. Excel functions are used to verify the beam capacity and a different sheet is used to present a scaled cross section of the beam. Figure 9 shows the spreadsheet calculations and the scaled drawing.

Using this spreadsheet, the student can appreciate the geometry of the beam, the location and size of the rebar and the stirrup with the specified cover. All this information is important for the beam design and it can be copied to the final drawings. This spreadsheet is also used to study the effect of the change of the design parameters on the final design, permitting the global understanding of the reinforced concrete beam design.

UNIVERSITY OF HOUSTON - DOWNTOWN   HOUSTON, TX

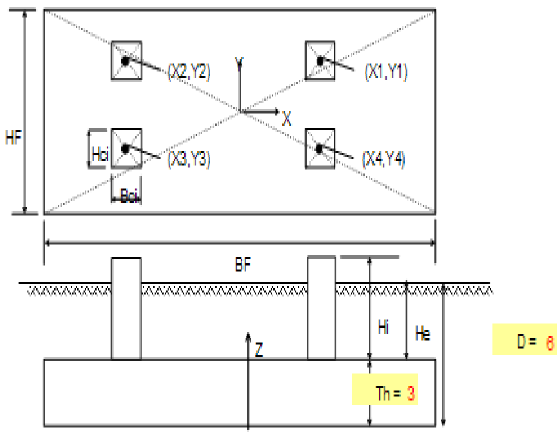
DESIGN BY: DATE: 1/8/2007   SUBJECT: SHEET N^{O} OF

CHKD. BY: DATE: JOB N^{O}

FILE: FILE: E:\Paper ASEE2007\[Footing-MultipleColumn.xls]Calculation

Program Name:

E:\Paper ASEE2007\[Footing-MultipleColumn.xls]Drawing



GENERAL EVALUATION OF THE FOOTING: OK

GENERAL DATA

FOUNDATION GEOMETRY:

Top of Pavement Elevation =		100 ft		
BF =	20 ft			
HF =	15 ft			
Th =	3 ft			
D =	6 ft		Bottom of Foundation Elevation =	94 ft
He =	3 ft		Top of Footing Elevation =	97 ft
Density of Concrete = γc =	150 pcf			
Density of Soil = γs =	110 pcf			

LOCATION OF COLUMNS:

COLUMN:	1	2	3	4			
Shape	Rectangular	Rectangular	Rectangular	Rectangular			
ToC Elevation =	101	101	101	101 ft			
Hi=	4	4	4	4 ft			
Xi=	8	-8	-8	8 ft			
Yi=	5.5	5.5	-5.5	-5.5 ft			
Zi=	7	7	7	7 ft			
Bci=	2	2	2	2 ft			
Hci=	2	2	2	2 ft			
; OF EACH COLUMN:	24	24	24	24 kip			
PLAN AREA:	4	4	4	4 ft^2 ==>	ΣAc =	16 ft^2	

Figure 6. Spreadsheet to Verify the Soil Pressure under a Proposed Footing (Part I / III)

UNIVERSITY OF HOUSTON - DOWNTOWN

HOUSTON, TX

DESIGN BY: DATE: 1/8/2007 SUBJECT: SHEET N° OF

CHKD. BY: DATE: JOB N°

FILE: FILE: E:\Paper ASEE2007\[Footing-MultipleColumn.xls]Calculation

Program Name:

E:\Paper ASEE2007\[Footing-MultipleColumn.xls]Drawing

WEIGHT OF COLUMNS:	10	kip	
WEIGHT OF EARTH:	94	kip	
WEIGHT OF SLAB:	135	kip	
TOTAL WEIGHT OF FOOTING:	238	kip	TO BE ADDED TO DEAD LOAD CASE ONLY
TOTAL MOMENT DUE COLUMN WEIGHT:	0	kip-ft	TO BE ADDED TO DEAD LOAD CASE ONLY
TOTAL MOMENT DUE COLUMN WEIGHT:	0	kip-ft	TO BE ADDED TO DEAD LOAD CASE ONLY

LOADS AT TOP OF COLUMN: DATA

DEAD LOAD

COLUMN	Vx	Vy	AXIAL	Mx	My	TORSION	Mx at	My at	X	Y	Z	Vy*Z	P*Y	Vx*Z	P*X
	kip	kip	kip	kip-ft	kip-ft	kip-ft	base	base							
1	-15	-20	-100	-80	-60				8	5.5	7	-140	-550	-105	-800
2	-20	-25	-150	-60	-70	0			-8	5.5	7	-175	-825	-140	1200
3	-20	-30	-125	-60	-90	0			-8	-5.5	7	-210	687.5	-140	1000
4	-15	-25	-90	-95	-110	0			8	-5.5	7	-175	495	-105	-720
							212.5	-1500				-700	-192.5	-490	680

LIVE LOAD

COLUMN	Vx	Vy	AXIAL	Mx	My	TORSION	Mx at	My at	X	Y	Z	Vy*Z	P*Y	Vx*Z	P*X
	kip	kip	kip	kip-ft	kip-ft	kip-ft	base	base							
1	0	0	0	0	0	0			8	5.5	7	0	0	0	0
2	0	0	0	0	0	0			-8	5.5	7	0	0	0	0
3	0	0	0	0	0	0			-8	-5.5	7	0	0	0	0
4	0	0	0	0	0	0			8	-5.5	7	0	0	0	0
							0	0				0	0	0	0

WIND LOAD SOUTH-NORTH

COLUMN	Vx	Vy	AXIAL	Mx	My	TORSION	Mx at	My at	X	Y	Z	Vy*Z	P*Y	Vx*Z	P*X
	kip	kip	kip	kip-ft	kip-ft	kip-ft	base	base							
1	0	0	0	0	0	0			8	5.5	7	0	0	0	0
2	0	0	0	0	0	0			-8	5.5	7	0	0	0	0
3	0	0	0	0	0	0			-8	-5.5	7	0	0	0	0
4	0	0	0	0	0	0			8	-5.5	7	0	0	0	0
							0	0				0	0	0	0

WIND LOAD WEST-EAST

COLUMN	Vx	Vy	AXIAL	Mx	My	TORSION	Mx at	My at	X	Y	Z	Vy*Z	P*Y	Vx*Z	P*X
	kip	kip	kip	kip-ft	kip-ft	kip-ft	base	base							
1	0	0	0	0	0	0			8	5.5	7	0	0	0	0
2	0	0	0	0	0	0			-8	5.5	7	0	0	0	0
3	0	0	0	0	0	0			-8	-5.5	7	0	0	0	0
4	0	0	0	0	0	0			8	-5.5	7	0	0	0	0
							0	0				0	0	0	0

Page 2 of 3

Figure 6. Spreadsheet to Verify the Soil Pressure under a Proposed Footing (Part II / III)

UNIVERSITY OF HOUSTON - DOWNTOWN

HOUSTON, TX

DESIGN BY: DATE: 1/8/2007 SUBJECT: SHEET N[o] OF

CHKD. BY: DATE: JOB N[o]

FILE: FILE: E:\Paper ASEE2007\[Footing-MultipleColumn.xls]Calculation

Program Name:

E:\Paper ASEE2007\[Footing-MultipleColumn.xls]Drawing

LOADS AT BOTTOM AND CENTER OF FOUNDATION

	DEAD	LIVE	WIND S-N	WIND W-E
AXIAL LOAD	-703	0	0	0
Vx	-70	0	0	0
Vy	-100	0	0	0
Mx	213	0	0	0
My	-1500	0	0	0

NG PROPERTIES

Area =	300 ft^2	Cx =	10 ft	Cy =	7.5 ft
Ixx =	5625 ft^4	Iyy =	10000 ft^4		

STRESSES DUE LOAD CASES

	DEAD LOAD (ksf)	LIVE LOAD (ksf)	WIND S-N (ksf)	WIND W-E (ksf)	X	Y
Point 1	0.561	0.000	0.000	0.000	10	7.5
Point 2	3.561	0.000	0.000	0.000	-10	7.5
Point 3	4.128	0.000	0.000	0.000	-10	-7.5
Point 4	1.128	0.000	0.000	0.000	10	-7.5

COMBINATIONS

COMBO	DL	LL	WL_{N-S}	WL_{E-W}	Max. qa	Min qa
1	1.0	1.0	0.0	0.0	4.5	0.0
2	1.0	0.5	1.0	0.0	6	0.0
3	1.0	0.5	-1.0	0.0	6	0.0
4	1.0	0.5	0.0	1.0	6	0.0
5	1.0	0.5	0.0	-1.0	6	0.0
6	0.6	0.0	1.0	0.0	6	0.0
7	0.6	0.0	-1.0	0.0	6	0.0
8	0.6	0.0	0.0	1.0	6	0.0
9	0.6	0.0	0.0	-1.0	6	0.0

COMBO	q1 (ksf)	q2 (ksf)	q3 (ksf)	q4 (ksf)	Max q (ksf)	Min q (ksf)	CRITERIA		
1	0.56	3.56	4.13	1.13	4.13	0.56	OK	0	
2	0.56	3.56	4.13	1.13	4.13	0.56	OK	0	
3	0.56	3.56	4.13	1.13	4.13	0.56	OK	0	
4	0.56	3.56	4.13	1.13	4.13	0.56	OK	0	
5	0.56	3.56	4.13	1.13	4.13	0.56	OK	0	
6	0.34	2.14	2.48	0.68	2.48	0.34	OK	0	
7	0.34	2.14	2.48	0.68	2.48	0.34	OK	0	
8	0.34	2.14	2.48	0.68	2.48	0.34	OK	0	
9	0.34	2.14	2.48	0.68	2.48	0.34	OK	0	0

Page 3 of 3

Figure 6. Spreadsheet to Verify the Soil Pressure under a Proposed Footing (Part III / III)

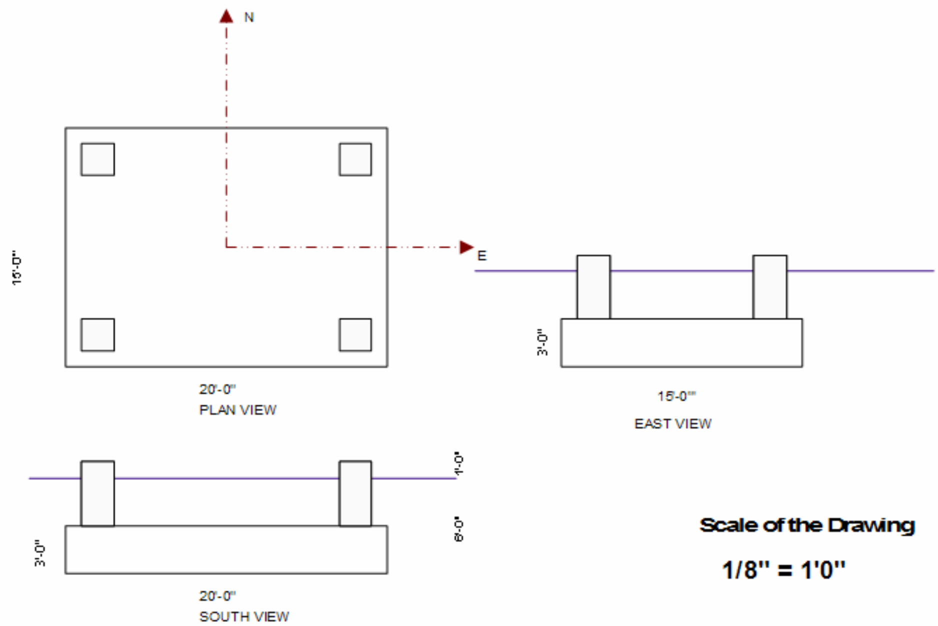
UNIVERSITY OF HOUSTON - DOWNTOWN HOUSTON, TX

DESIGN BY: DATE: 1/8/2007 SUBJECT: SHEET N° OF

CHKD. BY: DATE: JOB N°

FILE: FILE: E:\Paper ASEE2007\[Footing-MultipleColumn.xls]Drawing

Program Name:



Scale of the Drawing

1/8" = 1'0"

Page 1 of 1

Draw the Rectangle and Circle

Delete Old Figures

	Ground from top of footing =			3	ft		
	Top of Column to earth =			1	ft		
		b	**h**	**t**			
		ft	**ft**	**ft**			
	Cube	20	15	3			
Number of Columns =			4			E	N
		Along X	Along Y	Height		Center Location	
1	Rectangular	2	2	4	ft	8	5.5
2	Rectangular	2	2	4	ft	-8	5.5
3	Rectangular	2	2	4	ft	-8	-5.5
4	Rectangular	2	2	4	ft	8	-5.5

◉ Oval ○ Rectangular

Location of The Drawing:				
		column	5	
		row	12	
Space From Plan View to Lateral View =			10	ft
	Scale factor		8.53	
	Modification factor xx		1	For horizontal
	Modification factor yy		1.08	For Vertical
	Zoom factor:		0.8	

List	
1/8" = 1'0"	8.53
1/4" = 1'0"	17.06
3/8" = 1'0"	25.59
1/2" = 1'0"	34.12
5/8" = 1'0"	42.65
3/4" = 1'0"	51.18
7/8" = 1'0"	59.71
1" = 1'0"	68.24
pixels	1

Figure 7. Scaled Drawing of the Proposed Footing Designed and Data Necessary

```
Private Sub cmdDrawFooting_Click()

Dim Ncolumns, I As Integer
Dim Scale1, myColumn, myRow, myZoom As Double
Dim PositionX, PositionY, ModifX, ModifY As Double
Dim ScaleX, ScaleY, Fact, ColEarthDist As Double
Dim Base, Length, Thickness, DistViews As Double
Dim Base1, Length1, Thickness1 As Double
Dim Longi As String
Dim Radius, CenterEast, CenterNorth As Double
Dim EarthOn, FoundDepth As Double
Dim Ncols, fila As Integer
Dim TypeCol As String
Dim LengthX, LengthY, HeightCol As Double

'GENERAL INPUT DATA
'Modify scale according to the printer used
ModifX = ActiveSheet.Range("ModifX")
ModifY = ActiveSheet.Range("ModifY")
Fact = ModifX / ModifY
Base1 = ActiveSheet.Range("Base1")
Length1 = ActiveSheet.Range("Heigth1")
Thickness1 = ActiveSheet.Range("Thickness1")
'The Scale of the lines is defined.  Correct for a PageSetup
'different of 100%
ActiveSheet.Range("MyZoom") =
ActiveSheet.PageSetup.Zoom / 100
myZoom = ActiveSheet.PageSetup.Zoom / 100
Scale1 = ActiveSheet.Range("Scale1") / myZoom
ScaleX = Scale1 * ModifX
ScaleY = Scale1 * ModifY
'Read Earth Depth on Footing
EarthOn = Range("EarthOn") * ScaleY
FoundDepth = Range("EarthOn") + Thickness1
ColEarthDist = Range("ColEarthDist")

'LOCATE THE POSITION (0,0)
'PositionX and PositionY are the new coordinates in 'points
of the origin for the drawing
PositionX = 0
PositionY = 0
myColumn = ActiveSheet.Range("myColumn1")   'Read the
Location desired for the Drawing
myRow = ActiveSheet.Range("myRow1")        'Read 'the
Location desired for the Drawing
'Loop to find the value in points of origin to be used 'for the
drawing
For I = 1 To myColumn
   PositionX = PositionX + ActiveSheet.Columns(I).Width
Next I
For I = 1 To myRow
   PositionY = PositionY + ActiveSheet.Rows(I).Height
Next I
```

```
'Variables for drawing
Base = Base1 * ScaleX
Length = Length1 * ScaleY
Thickness = Thickness1 * ScaleY
DistViews = Range("DistViews") * ScaleY

'DRAWING
'DRAW RECTANGLE: Plan View
a = PositionX
b = PositionY
c = Base
d = Length
With_ ActiveSheet.Shapes.AddShape(msoShapeRectangle,_
a, b, c, d)
       .Line.DashStyle = msoLineSolid
       .Fill.ForeColor.RGB = RGB(255, 255, 255)
End With

'DRAW RECTANGLE: South View
a = PositionX
b = PositionY + Length + DistViews
c = Base
d = Thickness
With _ ActiveSheet.Shapes.AddShape(_
msoShapeRectangle, a, b, c, d)
       .Line.DashStyle = msoLineSolid
       .Fill.ForeColor.RGB = RGB(255, 255, 255)
End With

'DRAW RECTANGLE: East View
a = PositionX + Base + DistViews
b = PositionY + Length - Thickness
c = Length * Fact
d = Thickness
With   ActiveSheet.Shapes.AddShape(_
msoShapeRectangle, a, b, c, d)
       .Line.DashStyle = msoLineSolid
       .Fill.ForeColor.RGB = RGB(255, 255, 255)
End With

'DRAW AXIS X or EAST-WEST
a = PositionX + Base / 2
b = PositionY + Length / 2
c = PositionX + Base + DistViews * 0.5
d = b
With ActiveSheet.Shapes.AddLine(a, b, c, d).Line
   .DashStyle = msoLineDashDotDot
   .ForeColor.RGB = RGB(125, 0, 0)
   .EndArrowheadLength = msoArrowheadLong
   .EndArrowheadStyle = msoArrowheadTriangle
   .EndArrowheadWidth = msoArrowheadWide
End With
```

Figure 8. VBA Code to Draw the Proposed Footing (part I / III)

```
With  ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationHorizontal, c, d, 15, 15)
   .Line.Transparency = 1
   .TextFrame.Characters.Text = "E"
End With

'DRAW AXIS Y OR SOUTH-NORTH
a = PositionX + Base / 2
b = PositionY + Length / 2
c = a
d = PositionY + Length / 2 - Length
With ActiveSheet.Shapes.AddLine(a, b, c, d).Line
   .DashStyle = msoLineDashDotDot
   .ForeColor.RGB = RGB(125, 0, 0)
   .EndArrowheadLength = msoArrowheadLong
   .EndArrowheadStyle = msoArrowheadTriangle
   .EndArrowheadWidth = msoArrowheadWide
End With
With ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationHorizontal, c * 1.03, d, 15, 15)
   .Line.Transparency = 1
   .TextFrame.Characters.Text = "N"
End With

'WRITE DIMENSIONS
'Write Base in the Plan-View
a = PositionX + Base / 2 - 20
b = PositionY + Length + 10
Longi = Int(Base1) & "'-" & Round((Base1 - Int(Base1)) *
12, 1) & """"
With ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationHorizontal, a, b, 40, 15)
   .Line.Transparency = 1
   .TextFrame.Characters.Text = Longi
   '.Fill.ForeColor.RGB = RGB(250, 0, 250)
End With
a = PositionX + Base / 2 - 20
b = PositionY + Length + 25
Longi = "PLAN VIEW"
With ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationHorizontal, a, b, 80, 15)
   .Line.Transparency = 1
   .TextFrame.Characters.Text = Longi
End With

'Write Base in the South-View
a = PositionX + Base / 2 - 20
b = PositionY + Length + Thickness + DistViews + 10
Longi = Int(Base1) & "'-" & Round((Base1 - Int(Base1)) *
12, 1) & """"
With ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationHorizontal, a, b, 40, 15)
   .Line.Transparency = 1
   .TextFrame.Characters.Text = Longi
End With
a = PositionX + Base / 2 - 20
b = PositionY +Length+ Thickness + DistViews + 25
Longi = "SOUTH VIEW"
With  ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationHorizontal, a, b, 80, 15)
   .Line.Transparency = 1
   .TextFrame.Characters.Text = Longi
End With

'Write Length in the Plan-View
a = PositionX - 40
b = PositionY + Length / 2
Longi = Int(Length1) & "'-" & _
Round((Length1 - Int(Length1)) * 12, 1) & """" "
With ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationUpward, a, b, 15, 40)
   .Line.Transparency = 1
   .TextFrame.Characters.Text = Longi
End With

'Write Length in the East-View
a = PositionX + Base + DistViews + Base / 2 - 45
b = PositionY + Length + 15
Longi = Int(Length1) & "'-" & Round((Length1 -
Int(Length1)) * 12, 1) & """"  "
With ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationHorizontal, a, b, 45, 15)
   .Line.Transparency = 1
   .TextFrame.Characters.Text = Longi
End With
a = PositionX + Base + DistViews + Base / 2 - 60
b = PositionY + Length + 35
Longi = "EAST VIEW"
With ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationHorizontal, a, b, 60, 15)
   .Line.Transparency = 1
   .TextFrame.Characters.Text = Longi
End With

'Write Thickness in the East-View
a = PositionX + Base + DistViews - 20
b = PositionY + Length - Thickness / 2 - 12
Longi = Int(Thickness1) & "'-" & Round((Thickness1 -
Int(Thickness1)) * 12, 1) & _ """"
With ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationUpward, a, b, 15, 25)
   .Line.Transparency = 1
   .TextFrame.Characters.Text = Longi
End With
'Write Thickness in the South-View
a = PositionX - 25
b = PositionY + Length + DistViews + Thickness / 2 - 10
Longi = Int(Thickness1) & "'-" & Round((Thickness1 -
Int(Thickness1)) * 12, 1) & """" "
```

Figure 8. VBA Code to Draw the Proposed Footing (part II / III)

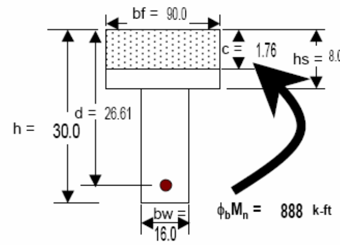
```
With ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationUpward, a, b, 15, 25)
  .Line.Transparency = 1
  .TextFrame.Characters.Text = Longi
End With
'Write Earth-Depth in the South-View
a = PositionX + Base + 40
b = PositionY + Length + DistViews + Thickness -
FoundDepth * ScaleY / 2 - 10
Longi = Int(FoundDepth) & "'-" & Round((FoundDepth -
Int(FoundDepth)) * 12, 1) & """ "
With ActiveSheet.Shapes.AddTextbox(_
msoTextOrientationUpward, a, b, 15, 25)
  .Line.Transparency = 1
  .TextFrame.Characters.Text = Longi
End With

'DRAW THE PLAN VIEW OF COLUMNS
Ncols = ActiveSheet.Range("Ncols")
fila = Range("Ncols").Row + 1
For I = 1 To Ncols
TypeCol = ActiveSheet.Cells(fila + I, 2)
LengthX = ActiveSheet.Cells(fila + I, 4) * ScaleX
LengthY = ActiveSheet.Cells(fila + I, 5) * ScaleY
CenterEast = ActiveSheet.Cells(fila + I, 8) * ScaleX
CenterNorth = ActiveSheet.Cells(fila + I, 9) * ScaleY
If TypeCol = "Oval" Then
  a = PositionX + Base / 2 + CenterEast - LengthX / 2
  b = PositionY + Length / 2 - CenterNorth - LengthY/2
  With ActiveSheet.Shapes.AddShape(_
msoShapeOval, a, b, LengthX, LengthY)
    .Line.DashStyle = msoLineSolid
    .Fill.ForeColor.RGB = RGB(250, 250, 250)
  End With
ElseIf TypeCol = "Rectangular" Then
 a = PositionX + Base / 2 + CenterEast - LengthX / 2
 b = PositionY + Length / 2 - CenterNorth - LengthY / 2
With ActiveSheet.Shapes.AddShape(_
msoShapeRectangle, a, b, LengthX, LengthY)
    .Line.DashStyle = msoLineSolid
    .Fill.ForeColor.RGB = RGB(250, 250, 250)
  End With
Else
MsgBox ("Review Data respect to Column Type, the
program accept Oval or Rectangular")
End If
Next I

'DRAW THE EARTH LINE
a = PositionX * 0.9
b = PositionY + Length + DistViews - EarthOn
c = (PositionX + Base) * 1.1
d = PositionY + Length + DistViews - EarthOn
With ActiveSheet.Shapes.AddLine(a, b, c, d).Line
  .DashStyle = msoLineSolid
  .ForeColor.RGB = RGB(50, 0, 128)
End With
a = (PositionX + Base + DistViews) * 0.9
b = PositionY + Length - Thickness - EarthOn
c = (PositionX + Base + DistViews + Length) * 1.1
d = PositionY + Length - Thickness - EarthOn
With ActiveSheet.Shapes.AddLine(a, b, c, d).Line
  .DashStyle = msoLineSolid
  .ForeColor.RGB = RGB(50, 0, 128)
End With

'DRAW THE ELEVATION VIEW OF COLUMNS
For I = 1 To Ncols
  LengthX = ActiveSheet.Cells(fila + I, 4) * ScaleX
  LengthY = ActiveSheet.Cells(fila + I, 5) * ScaleY
  HeightCol = ActiveSheet.Cells(fila + I, 6) * ScaleY
  CenterEast = ActiveSheet.Cells(fila + I, 8) * ScaleX
  CenterNorth = ActiveSheet.Cells(fila + I, 9)*ScaleY
  a = PositionX + Base / 2 + CenterEast - LengthX / 2
  b = PositionY + Length + DistViews - HeightCol
  With ActiveSheet.Shapes.AddShape(_
msoShapeRectangle, a, b, LengthX, HeightCol)
    .Line.DashStyle = msoLineSolid
    .Fill.ForeColor.RGB = RGB(250, 250, 250)
  End With
a = PositionX + Base + DistViews + Length * Fact /2 _ +
CenterNorth * Fact - LengthY * Fact / 2
b = PositionY + Length - Thickness - HeightCol
With ActiveSheet.Shapes.AddShape(_
msoShapeRectangle, a, b, LengthY * Fact, HeightCol)
    .Line.DashStyle = msoLineSolid
    .Fill.ForeColor.RGB = RGB(250, 250, 250)
  End With
Next I

End Sub
```

Figure 8. VBA Code to Draw the Proposed Footing (part III / III)

UNIVERSITY OF HOUSTON - DOWNTOWN **HOUSTON, TX**

BY ____________________ DATE _____ SUBJECT ____ SHEET NO _______ OF _____
CHKD BY ________________ DATE _____ ____ JOB NO ________________

FILE: E:\Paper ASEE2007\[Reinf.Concrete-BeamDesign-2006.xls]Beam Design

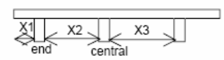


CONCRETE T-BEAM WITH STEEL IN COMPRESSION

FLEXURE

ACI 318-05 Ch. 10

Concrete Strength 28 days, f'_c: 5,000 psi
Ultimate Strain at Concrete = ε_{cu} = 0.003 Art. 10.3.3
0.65< β_1 = 1.05 - 0.05*f'c/1000 < 0.85= 0.80 art. 10.2.7.3
Longitudinal Steel Yielding, f_y: 60,000 psi It may be grade 40, 60 or 75
Modulus of Elasticiy of Steel, E = 29,000,000 psi
Stirrup Steel Yielding, fys: 60,000 psi Grade 60 or lower. Art 11.5.2
Height of T-Beam (h): 30.0 in
Width of T-Beam (bw): 16.0 in
Central, End or Isolated T-Beam? Central End Central Isolated
Width of Flange for Isolated T-Beam = bf' in Input this value for Isolated T-Beam



Clear Span of the T-Beam = ln = 30.0 ft
Cantilever dist.(End T-Beam Only), X1 0.0 ft Effective Cantilever = 0 in
Clear Dist. Between T-Beams @Right=X2 10.0 ft --
Clear Dist. Between T-Beams @Left=X3 10.0 ft
Thickness of Slab = hs = 8.0 in -- **Note: For Rectangular beams assume hs = h**
For Central T-Beam: bf = min(ln/4, X1/2+bw+X2/2, 8hs+bw+8hs) 90 136 144 90 in
For End T-Beam: bf = min(min(X1,6hs,ln/12)+bw+ln/12, min(X1,6hs,ln/12)+bw+X2/2,min(X1,6hs,ln/12)+bw+ 6hs) 46 in
For Isolated T-Beam: Verify that bf = min(bf',4*bw) and hs>=bw/2 64 in
Effective Width of the Flange = bf = 90.0 in art. 8.10

Structure Type (B,E,S): Beam Beam Joist Slab
E Pour Against a Form Pour Against Earth

Ultimate Moment = Mu = (+:Tension at bottom, -:Tension at top) 550 kip-ft 550 kip-ft **Tension at: Bottom**
Minimum Steel Ratio = ρmin = 0.0035 For General Beams: GB 0.0071 For Isostatic Cantilever Beams:ICB
Type of Beam GB Simply supported or Continuous Bea Isostatic Cantilever Beams
Minimum Steel Ratio = ρ_{min} = 0.00354
As minimum = ρ_{min}*bw*d 1.51 in^2

VERIFICATION OF BEAM

Width of Beam (bw): 16.0 in
Effective Width of the Flange = bf' = 90.0 in **Tension at: Bottom**
Minimum cover: 1.50 in art. 7.7.1
Stirrup #: 4 Number of Stirrup Legs: 2
Nominal Stirrup Diameter, dbst: 0.50 in
Min. Distance Between Stirrup Face and Face of Adjacent Rebar: 1 in

Layer # 1: Rebar in Tension

Number of bars	Rebar Size	Nominal Rebar Area (in^2)	Nominal Diameter db(in)	Max db (in)	Steel Area Layer	Centroid of Layer (in)	Minimum Spacing between rebars (in)	Spacing between rebars (in)	Verification of Beam Width
4	10	1.27	1.27						OK if
0	9	1	1.128						splices are
0	8	0.79	1	1.27	5.08	2.64	1.27	2.06	not used!!!

Layer # 2: Spacing with layer #1 1 in

Number of bars	Rebar Size	Nominal Rebar Area (in^2)	Nominal Diameter db(in)		Steel Area Layer	Centroid of Layer (in)	Minimum Spacing between rebars (in)	Spacing between rebars (in)	Verification of Beam Width
2	10	1.27	1.27						
0	10	1.27	1.27						
0	10	1.27	1.27	1.27	2.54	4.91	1.27	9.46	OK

Total Area of Rebar (A_s) : 7.62 in^2
Centroid of Rebars, x_{cg} = 3.39 in
Effective depth = d = h - cg = 26.61 in

Layer # 3: Compression Rebar

Number of bars	Rebar Size	Nominal Rebar Area (in^2)	Nominal Diameter db(in)		Steel Area Layer	Centroid of Layer (in)	Minimum Spacing between rebars (in)	Spacing between rebars (in)	Verification of Beam Width
2	9	1	1.128						
0	10	1.27	1.27						
0	10	1.27	1.27	1.128	2	2.56	1.13	8.87	OK

Page 1 of 2

Figure 9. Spreadsheet to Verify Reinforced Concrete Beams (Part I / III)

UNIVERSITY OF HOUSTON - DOWNTOWN **HOUSTON, TX**

BY ____________________ DATE _____ SUBJECT ____ SHEET NO _______ OF _____
CHKD BY ________________ DATE _____ ____ JOB NO ________________

FILE: E:\Paper ASEE2007\[Reinf.Concrete-BeamDesign-2006.xls]Beam Design

Area of Rebar in Compression (A_{spc}):	2.00	in^2					
Effective depth = d' =	2.56	in					
Area of Rebar in Tension Asp1=As-Asp=	5.62						

Part I: Use if a <= hs

Whitney block=a=Asp1*fy / (0.85*f'c*bf)=	0.88	in			c = a/β1 =	1.10	in
Strain at compression steel $\varepsilon'_s=\varepsilon_{cu}/c*(c-d')$=	-0.00398		εy = fy / Es =	0.00207 in/in			
Recalculation of c if $\varepsilon_s' < \varepsilon_y$. c =	1.76		A = 306000	B = -283200		D= -446136	
Strain at compression steel $=\varepsilon_{cu}/c*(c-d')$ =	-0.00138						
Stress at compression steel = fcs =	0	psi					
Whitney block adjusted: a =	1.40	in					
M_{n1} = (As*fy-Aspc*fcs)*(d-a/2))/12000 =	987.1	k-ft					
M_{n2} = (Aspc*fcs)*(d-d'))/12000 =	0.0	k-ft					
$M_n = M_{n1} + M_{n2}$ =	987.1	k-ft					
Verification than a<hs:	OK, a<=hs				hs =	8	in

Part II: Use if a > hs

Steel As1 = 0.85*(bf-bw)*hs*fc / fy =	0.00 in^2					
Mn1 = As1 * fy * (d-hs/2)/12000 =	0.0 kip-ft					
Asp2 = Asp1 - As1	0.00 in^2					
a = Asp2 * fy / (0.85*f'c*bw) =	0.00 in			c = a/β1 =	0.00 in	
Strain at compression steel = ecu/c*(c-d') =	0.00000					
Recalculation of c if $\varepsilon_s' < \varepsilon_y$. c =	0.00		A = 306000	B = 2232800		D= -446136
Strain at compression steel = $\varepsilon_{cu}/c*(c-d')$ =	0.0000					
Stress at compression steel = fcs =	0	psi				
Whitney block adjusted: a =	0.00					
Mn2 = (As*fy-Asp2*fcs) * (d-a/2) / 12000 =	0.0 kip-ft					
Mn3 = Aspc*fcs*(d-d')/12000 =	0.0 kip-ft					
Mn = Mn1 + Mn2 + Mn3=	0.0 kip-ft					

Verification of Ductility and Calculation of ϕb

β1 =	0.80					
Block of Whitney = a =	1.40 in					
Neutral Axis, c =	1.76 in					
Ultimate Strain at Concrete = ε_{cu} =	0.003		Art. 10.3.3			
Strain at steel = $\varepsilon_t = \varepsilon_{cu}/c*(d-c)$ =	0.0425		OK!!! art. 10.3.5			
ϕ_b=0.65+(7250*εt-0.25*fy)/(145-fy)<=0.9	0.90		fy in ksi			
M_n =	**987.1**	**k-ft**		**$\phi_b M_n$ =**	**888**	**k-ft**
From Data: M_u =	**550.0**	**k-ft**	**OK!!! Flexure**	**Diff =**	**61.5%**	

SHEAR STRENGTH Vu <= ϕ_v * Vn = ϕ_v * (Vcn + Vsn) ACI 318- Note: For seismic areas see Chapter 21

Ultimate Acting Shear = Vu =	65 kip		Type of element: Beam	
ϕ_v =	0.75	art. 9.3.2.3		
Concrete Capacity, Vc:				
$f'c^{0.5}$	70.71 psi	art 11.1.2	For Slabs: $f'c^{0.5}$ <= 100 psi	
Vc =2*(f'c)$^{0.5}$ * bw*d =	60.21 kip	art.11.3.1.1	beams have no limit, the minimum shear steel change.	
ϕ_v*Vcn =	45.16 kip	continue		
Stirrup Contribution, Vs:				
Is necessary stirrups?	Provide Stirrups, Vu>fv*Vc		Note: For Slabs, footings, Joist as art 8.11 and shallow beam is not necessary minimum stirrups. (art 11.5.5.1)	
Vs = Vu/ϕ_v - Vc	26.46 kip			
4*$f'c^{0.5}$ * bw*d =	120.4 kip	art. 11.5.4.3		
		If Vs>=4*$f'c^{0.5}$*b*d ==>Maximum Spacing of Stirrups shall be reduced 50%		
Max Vs <= 8*$f'c^{0.5}$ * bw*d =	240.84 kip	art. 11.5.6.9	Continue	
Stirrup #:	4			
Number of Stirrup Legs:	2	(Closed stirrups are used, then this is an even number. For slabs use "0")		
Nominal Stirrup Dia:	0.50 in			
Nominal Stirrup Area:	0.2 in^2 per leg			
Maximum Spacing art. 11.5.4.1 = smax =	13.305 in	art. 11.5.4		
Spacing of Stirrups = s =	10 in	OK		
Av = Number of Legs * stirrup area =	0.4 in^2			
Min. Av =max(0.75*$f'c^{0.5}$,50)*bw*s/fys =	0.1410 in^2	OK minimum Av	art. 11.5.5.3	
Provided Vsn = Av * fys * d / s =	63.86 kip	OK Shear		
ϕ_v * Vs =	47.9			
Vn =	**124.08 kip**		**ϕ_v * Vn =**	**93.06 kip**
From Data: V_u =	**65.0 k-ft**	**OK Shear**	**Diff =**	**43.2%**

Figure 9. Spreadsheet to Verify Reinforced Concrete Beams (Part II / III)

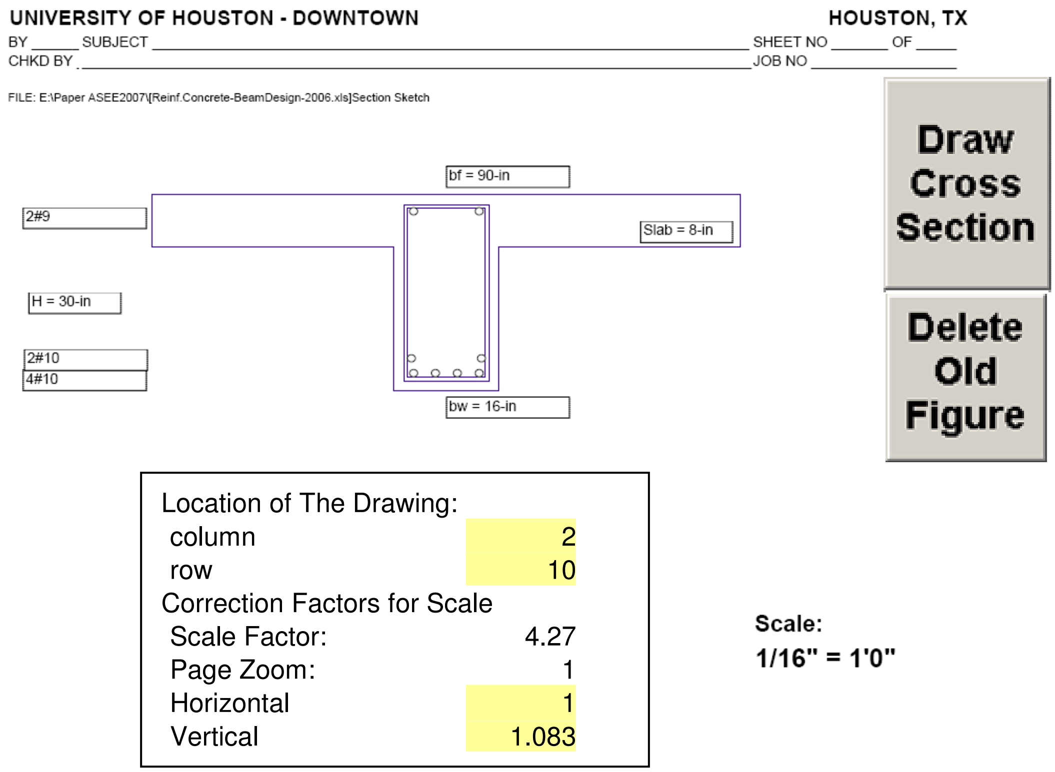


Figure 9. Spreadsheet to Verify Reinforced Concrete Beams (Part III / III)

Conclusions

These spreadsheets with the help of scaled graphic commands are very useful educational tools; they permit the student trials with different materials, geometry and other variables. The graphical spreadsheets avoid impractical solutions, mainly because the designer can visualize the possible design. As the spreadsheet is easy to change, the students can create their own spreadsheet based on the one given by the instructor.

Finally, the students may perform their own research about the relative importance of the parameters involved in the design. These tools are practically virtual laboratories.

Bibliography

1. Microsoft Office Excel 2003, part of Microsoft Office Professional Edition 2003. Microsoft Corporation.
2. Bowles, Joseph E., "Foundation Analysis and Design", 4^{th} Ed., McGraw-Hill, 1998.
3. American Concrete Institute, ACI 318-05, Building Code Requirements for Structural Concrete and Commentaries.

Acknowledgement

The authors want recognize the support of Scholars Academy of the University of Houston Downtown for their support.