Visualization of the Geometric Properties for the Design of a Gusset Plate using a Spreadsheet with Scaled Graph Capability

Jorge A. Tito
Structural Analysis and Design Program
Department of Engineering Technology
University of Houston Downtown, Houston, TX 77002-1001

ABSTRACT – A gusset plate is used to connect two structural elements. The gusset plate design is based on the Whitmore area, which is obtained graphically by intersecting the perpendicular cross section at the end of a connection with a line traced from the start of the connection forming 30 degrees angle with the connection centerline. The pseudo-column lengths are also obtained graphically by tracing perpendicular lines from the ends, through the center of the Whitmore area, and finishing at the edges. The spreadsheet presented follows the classical graphical method using Analytical Geometry to find Whitmore area and the pseudo-column lengths presenting the results in a scaled drawing. This spreadsheet with graphical presentation permits the student a better visualization of a geometric problem, permitting further calculations.

I. Introduction

Gusset plates are used to connect individual members to form strong structures. The design of gusset plates is based on experimental works whose conclusions reveal failure modes at a specific zone called Whitmore width, which is shown in Figure 1, as defined by AISC (2005). Finding the Whitmore width is possible using a scaled graph, however it is time consuming and the results shall be used as input for other computer tools in order to verify the structural quality of the gusset plate.

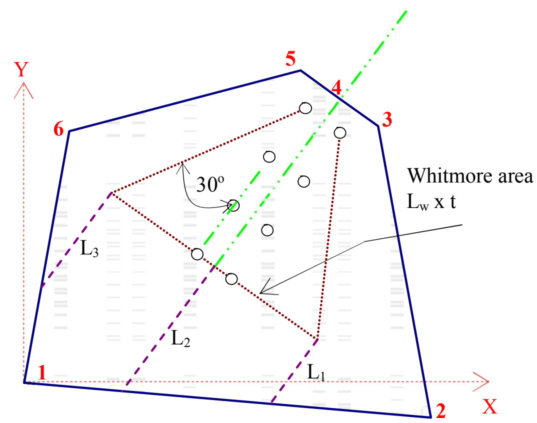


Figure 1. Typical gusset plate, Whitmore area and pseudo-column lengths

The points and lines defining the gusset plate can be represented with coordinates and incidence, which are two nodes defining the start and end of the line.

The capabilities of Excel (2003 and 2007) spreadsheet can be improved significantly using Visual Basic for Applications (VBA), and in this way a series of functions are created to find the intersections of lines, perpendicular lines, and parallel lines, among other applications. The theory of Analytical Geometry is used intensively to create these VBA functions.

Excel and VBA can also be used to draw the points and lines in the spreadsheet. The scale, type of lines, texts, colors, and other shapes or properties are user-defined.

II. Functions written in VBA using Analytical Geometry

Excel and VBA can be used to create functions which are not in the Excel library. The functions described in this work are based on Analytical Geometry theory.

An orthogonal two-dimensional coordinate system is used to define the coordinates *(x, y)* of the nodes, and the lines are defined using the concept of incidence, which are its start and end nodes.

```
Public Function InterceptL1L2(XY1 As Range, _
XY2 As Range, XY3 As Range, XY4 As Range)
Dim m1, m2, Xo As Double
Dim X1, X2, X3, X4 As Double
Dim Xintercept, Yintercept, Intercepta(2) As Variant

X1 = XY1(1): Y1 = XY1(2)
X2 = XY2(1): Y2 = XY2(2)
X3 = XY3(1): Y3 = XY3(2)
X4 = XY4(1): Y4 = XY4(2)

If X2 = X1 Then
   m1 = 10000000
Else
   m1 = (Y2 - Y1) / (X2 - X1)
End If
If X4 = X3 Then
   m2 = 10000000
Else
   m2 = (Y4 - Y3) / (X4 - X3)
End If

If m2 = m1 Then
   Xintercept = "n/a, //"
   Yintercept = "n/a, //"
Else
   Xintercept = (m2 * X4 - Y4 + Y2 - m1 * X2) / (m2 - m1)
   Xo = Xintercept
   Yintercept = m2 * (Xo - X4) + Y4
End If

Intercepta(0) = Xintercept
Intercepta(1) = Yintercept
InterceptL1L2 = Intercepta

End Function
```

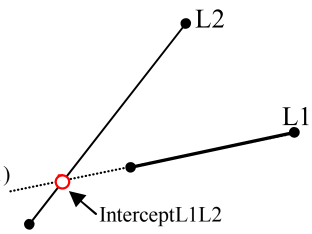


Figure 2. VBA code for the function to find the interception of two lines

```
Public Function PerpL1L2(XYo As Range, XY1 As Range, _
XY2 As Range, XY3 As Range, XY4 As Range)
Dim m1, m2, XX As Double
Dim Xo, Yo, X1, Y1, X2, Y2, X3, Y3, X4, Y4, Xp, Yp As
Double
Dim PerpL1L2Temp(2) As Double

Xo = XYo(1): Yo = XYo(2)
X1 = XY1(1): Y1 = XY1(2)
X2 = XY2(1): Y2 = XY2(2)
X3 = XY3(1): Y3 = XY3(2)
X4 = XY4(1): Y4 = XY4(2)

If X2 = X1 Then
   m1 = 100000
Else
   m1 = (Y2 - Y1) / (X2 - X1)
End If
If X4 = X3 Then
   m2 = 100000
Else
   m2 = (Y4 - Y3) / (X4 - X3)
End If

If m1 = 0 Then
   Xp = Xo
Else
   Xp = 1 / (m2 + 1 / m1) * (Xo / m1 + Yo + m2 * X3 - Y3)
End If

Yp = m2 * (Xp - X3) + Y3

PerpL1L2Temp(0) = Xp
PerpL1L2Temp(1) = Yp
PerpL1L2 = PerpL1L2Temp
End Function
```

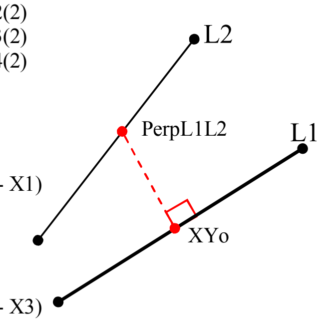


Figure 3. VBA code for the function to find a perpendicular to L_1 from XYo and intersecting L_2

The function *InterceptL1L2()* is created to find the coordinates of the intersection point of two lines. Figure 2 shows the VBA code for this function. The lines are defined using the incidence nodes of L_1:XY_1-XY_2, and L_2:XY_3-XY_4.

```
Public Function PerpPointL1(XY1 As Range, _
XY2 As Range, XYp As Range)
Dim m1, Xo, Yo As Double
Dim PerpPointL1Temp(2) As Double
Dim X1, Y1, X2, Y2, Xp, Yp As Double

X1 = XY1(1): Y1 = XY1(2)
X2 = XY2(1): Y2 = XY2(2)
Xp = XYp(1): Yp = XYp(2)

If X2 = X1 Then
   m1 = 100000000
Else
   m1 = (Y2 - Y1) / (X2 - X1)
End If

If Y1 = Y2 Then
   Xo = Xp
   Yo = Y1
Else
   Xo = (m1 * X1 - Y1 + Xp / m1 + Yp) / (m1 + 1 / m1)
   Yo = -1 / m1 * (Xo - Xp) + Yp
End If

PerpPointL1Temp(0) = Xo
PerpPointL1Temp(1) = Yo
PerpPointL1 = PerpPointL1Temp

End Function
```

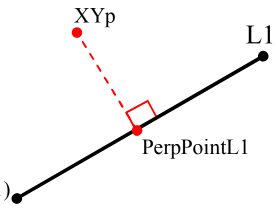


Figure 4. VBA code for the function to find a perpendicular from an outer point XYp toward the line L1

The function *PerpL1L2()* is used to define a perpendicular line to L_1 from a specified point and intersecting a second line L_2. As shown in Figure 3, the function finds a point belonging to line L_2:$(XY_3$-$XY_4)$ that when is joined to the given point *XYo* forms a perpendicular to the line L_1:$(XY_1$-$XY_2)$.

Figure 4 shows the VBA code for the function *PerpPointL1()* which finds a perpendicular line from a given outer point *XYo* toward the line L_1.

Figure 5 shows the VBA code for the function *ParallelL1d()* which is used to find the incidence points of a line L_2 which is parallel to the given line L_1 at a defined distance "*d*". If *d* is positive then the line L_2 is located counterclockwise of L_1.

```
Public Function ParallelL1d(XY1 As Range, _
 XY2 As Range, d)
Dim Parallel(2, 2) As Double

XY3 = PointDistFromPoOfL1(XY1, XY1, XY2, d)
XY4 = PointDistFromPoOfL1(XY2, XY1, XY2, d)

Parallel(0, 0) = XY3(0)
Parallel(0, 1) = XY3(1)
Parallel(1, 0) = XY4(0)
Parallel(1, 1) = XY4(1)

ParallelL1d = Parallel
End Function

Public Function PointDistFromPoOfL1(XYo As Range, _
XY1 As Range, XY2 As Range, DistFromL1)
Dim m1, a, b, C, d, XX, SignoDist As Double
Dim Xo, Yo, X1, Y1, X2, Y2, Xd, Yd As Double
Dim PointDistTempo(2) As Double

Xo = XYo(1): Yo = XYo(2)
X1 = XY1(1): Y1 = XY1(2)
X2 = XY2(1): Y2 = XY2(2)

If DistFromL1 = 0 Then
   Xd = Xo
   Yd = Yo
   GoTo 100
End If

SignoDist = DistFromL1 / Abs(DistFromL1)

RadioAngle = ConvertCartesianPolar(X2 - X1, Y2 - Y1)
Angle = RadioAngle(1)
If Angle > 180 Then SignoDist = -SignoDist

If X2 = X1 Then
   m1 = 100000000
   Xd = X2 - DistFromL1
Else
   m1 = (Y2 - Y1) / (X2 - X1)
End If

If m1 = 0 Then
   Yd = Yo + DistFromL1
   Xd = Xo
Else
   d = m1 * Xo - m1 * X1 + Y1 + Xo / m1
   a = 1 + 1 / m1 ^ 2
   b = -2 * Xo - 2 * d / m1 + 2 * Yo / m1
   C = Xo ^ 2 + d ^ 2 - 2 * Yo * d + Yo ^ 2 - DistFromL1 ^ 2
   If X2 = X1 Then
      Xd = X2 - DistFromL1
   Else
      Xd = (-b - SignoDist * (b ^ 2 - 4 * a * C) ^ 0.5) / (2 * a)
   End If
   Yd = -1 / m1 * (Xd - Xo) + Yo
End If
100
PointDistTempo(0) = Xd
PointDistTempo(1) = Yd
PointDistFromPoOfL1 = PointDistTempo

End Function
```

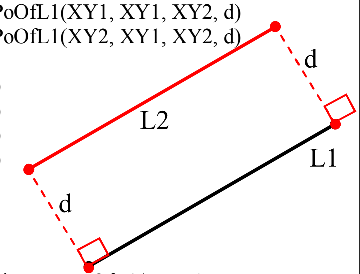


Figure 5. VBA code for the function to find a parallel line

```
Public Function ConvertCartesianPolar(X, Y)
Dim Radio, Angle1, Angle2 As Single
Dim CPc(2) As Single
Dim Quad As Integer

Radio = (X ^ 2 + Y ^ 2) ^ 0.5

If X = 0 And Y = 0 Then
   CPc(0) = Radio
   CPc(1) = 0
Else
   Angle1 = WorksheetFunction. _
   Degrees(WorksheetFunction.Atan2(Abs(X), Abs(Y)))
End If

If X >= 0 And Y >= 0 Then
   Quad = 1
ElseIf X < 0 And Y >= 0 Then
   Quad = 2
ElseIf X < 0 And Y < 0 Then
   Quad = 3
ElseIf X >= 0 And Y < 0 Then
   Quad = 4
End If
Select Case Quad
Case 1
   Angle2 = Angle1
Case 2
   Angle2 = 180 - Angle1
Case 3
   Angle2 = 180 + Angle1
Case 4
   Angle2 = 360 - Angle1
End Select

CPc(0) = Radio
CPc(1) = Angle2
ConvertCartesianPolar = CPc

End Function
```

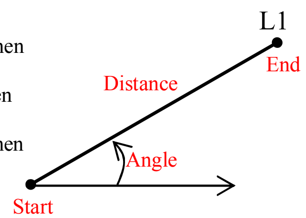


Figure 6. VBA code for the function to find the distance and direction of the line L1

Another function used to find the distance between the incidence points and the angle with respect to the horizontal line is called *ConvertCartesianPolar()*, its VBA code is presented in Figure 6, where the *X* means the difference of ordinates (X_{end} - X_{start}) and *Y* the difference of abscissas (Y_{end} - Y_{start}).

The scaled drawing of the nodes and lines is done using the VBA programs presented by Tito and Gomez-Rivas (2007). The nodes and line incidence is used as input to make the graphical presentation.

III. Gusset Plate

Figure 7 shows the input parameters with their positive sign; in this form, a negative value means the opposite direction.

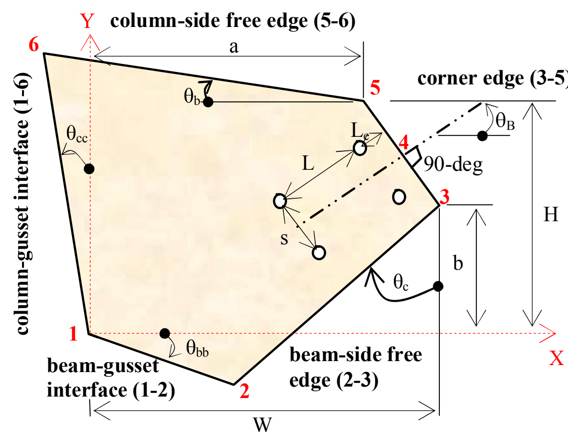


Figure 7. Schematic of a gusset plate and the information considered as input

Table 1. Information used as input

Symbol	Description
W	*Horizontal projection of the beam-gusset interface and beam-side free edge*
H	*Height from the beam-gusset interface up to the column-side free edge and corner edge intersect*
a	*Horizontal distance from Y-axis to point 5, it is the X-coordinate of point 5.*
b	*Vertical distance from X-axis to point 3, it is the Y-coordinate of point 3*
θ_b	*Angle of column-side free edge when measured to a horizontal line **
θ_c	*Angle of beam-side free edge when measured to a vertical line **
θ_{bb}	*Angle from the beam-gusset interface when measured to the X-axis **
θ_{cc}	*Angle from the column-gusset interface when measured to the Y-axis **
L	*Distance from the first hole to the last*
Le	*Edge distance from the first hole to the corner edge*
s	*Distance between parallel row of holes*
**** Note:*** *the angles are positive in the direction indicated in Figure 7*	

As shown in Figure 7, the coordinates used for this paper are referred as the orthogonal axis X and Y whose origin is at point *1*. The coordinates of the corners defining the gusset plate are described below.

$$\text{Node 1: } X_1 = 0 \qquad Y_1 = 0 \qquad (1)$$

$$\text{Node 3: } X_3 = W \qquad Y_3 = b \qquad (2)$$

$$\text{Node 5: } X_5 = a \qquad Y_5 = H \qquad (3)$$

The node 4 is at mid distance between the nodes 3 and 5, then:

$$\text{Node 4: } X_4 = (W+a)/2 \qquad Y_4 = (b+H)/2 \qquad (4)$$

The coordinates of node *2* are found using the intersection of the line *∠1-101* and *∠3-102*, where the node *101* belongs to the line *∠1-2*, and the node *102* belongs to the line *∠3-2*, located at an arbitrary distance "*d*", in this form:

Assume: d = 10

Node 101:

$$X_{101} = X_1 + d \,.\, \cos(\theta_{bb}) \qquad (5)$$

$$Y_{101} = Y_1 - d \,.\, \sin(\theta_{bb}) \qquad (6)$$

Node 102:

$$X_{102} = X_3 - d \,.\, \sin(\theta_c) \qquad (7)$$

$$Y_{102} = Y_3 - d \,.\, \cos(\theta_c) \qquad (8)$$

The node *2* is found using the VBA function:

$$=InterceptL1L2(X_1{:}Y_1,\ X_{101}{:}Y_{101},\ \_ \quad X_3{:}Y_3,\ X_{102}{:}Y_{102}) \qquad (9)$$

Similarly, the coordinates of the node 6 are found using the intersection of the line *∠1-103* and *∠3-104*, where the nodes *103* and *104* are auxiliary points belonging to the lines *∠1-6* and *∠3-6*, respectively, at an arbitrary distance "*d*", yielding:

Assume: d = 10

Node 103:

$$X_{103} = X_1 - d \,.\, \sin(\theta_{cc}) \qquad (10)$$

$$Y_{103} = Y_1 + d \,.\, \cos(\theta_{cc}) \qquad (11)$$

Node 104:

$$X_{104} = X_5 - d \,.\, \cos(\theta_b) \qquad (12)$$

$$Y_{104} = Y_5 + d \,.\, \sin(\theta_b) \qquad (13)$$

The node *6* is found using the VBA function:

$$=InterceptL1L2(X_1{:}Y_1,\ X_{103}{:}Y_{103},\ \_ \quad X_5{:}Y_5,\ X_{104}{:}Y_{104}) \qquad (14)$$

The next step consists of finding the angle between the connection centerline with the horizontal and the coordinates of the first and last holes. It is assumed two parallel lines of holes with their centerline intercepting the gusset plate at node *4*.

As shown in Figure 8, the node *8* belongs to a line perpendicular to the line *∠4-5* at distance L_e from the node *4*; therefore, its coordinates (X_8, Y_8) are found using the following expression:

$$=PointDistFromPoOfL1(X_4{:}Y_4,X_4{:}Y_4,X_5{:}Y_5,L_e) \qquad (15)$$

Similar for the node *9*, but at distance $L+L_e$:

$$=PointDistFromPoOfL1(X_4{:}Y_4,X_4{:}Y_4,X_5{:}Y_5,L+L_e) \qquad (16)$$

The nodes *10* and *11* form a line parallel to the line *∠8-9* at a distance *–s/2*, for which, the negative sign indicates clockwise from line *∠8-9*. The following VBA function is used:

=parallell1d(X_8:Y_8, X_9:Y_9, -s/2) (17)

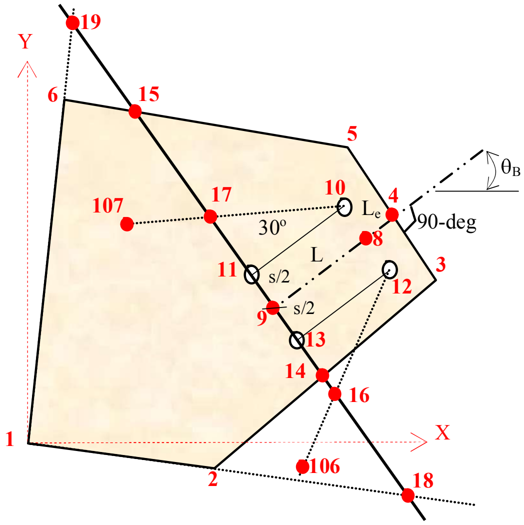


Figure 8. Definition of nodes

The nodes *12* and *13* are found considering a distance *s/2*:

=parallell1d(X_8:Y_8, X_9:Y_9, s/2) (18)

The angle forming the line *∠9-8* with the horizontal is defined as θ_B. The following VBA function provides the distance between points *9* and *8* and the required angle θ_B.

=ConvertCartesianPolar(X_8–X_9, Y_8–Y_9) (19)

IV. Whitmore Width

As shown in Figure 8, the nodes *106* and *107* belong to the line forming 30-deg with the lines *∠10-11* and *∠12-13* starting from the nodes *10* and *12*, respectively. The following equation is used:

Assume: d = 10

Node 106:

$$X_{106} = X_{12} - d \cdot \cos(\theta_B + 30) \quad (20)$$

$$Y_{106} = Y_{12} - d \cdot \sin(\theta_B + 30) \quad (21)$$

Node 107:

$$X_{107} = X_{10} - d \cdot \cos(\theta_B - 30) \quad (22)$$

$$Y_{107} = Y_{10} - d \cdot \sin(\theta_B - 30) \quad (23)$$

The node *16* is the intersection of the line *∠12-106* with the line *∠9-13*:

=InterceptL1L2(X_9:Y_9, X_{13}:Y_{13}, X_{12}:Y_{12}, X_{106}:Y_{106}) (24)

Similarly, node *17* is found as the interception of the line *∠9-13* with the line *∠10-107*:

=InterceptL1L2(X_9:Y_9, X_{13}:Y_{13}, X_{10}:Y_{10}, X_{107}:Y_{107}) (25)

The other nodes along the line 9-13 are:

Node 14: Interception of line 9-13 with side 2-3

Node 15: Interception of line 9-13 with side 5-6

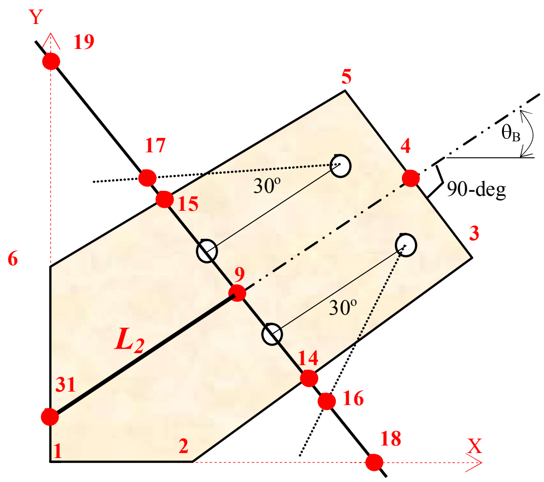
Node 18: Interception of line 9-13 with side 1-2

Node 19: Interception of line 9-13 with side 1-6

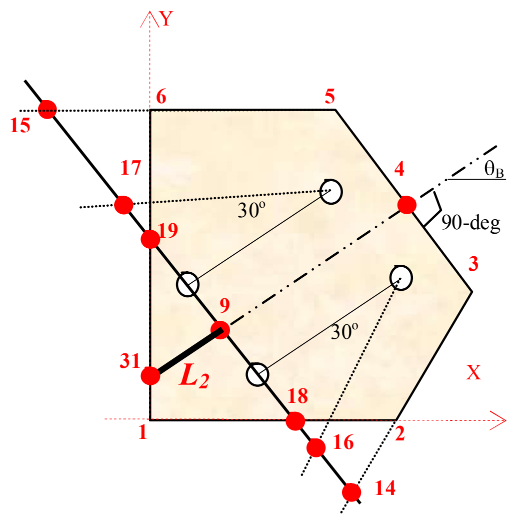
When node *17* is inside the gusset plate, then the distance *Ŝ16-17* is the Whitmore width, L_w. Figure 9 shows a pseudo-code that verifies the correct value of L_w, and Figure 10 shows the particular situations to find the correct value of L_w.

```
If Ŝ9-16 <= Ŝ9-14 and Ŝ9-16 <= Ŝ9-18 then
    Lw1 = Ŝ9-16
Else If Ŝ9-16 > Ŝ9-14 and Ŝ9-14 <= Ŝ9-18 then
    Lw1 = Ŝ9-14
Else Ŝ9-16 > Ŝ9-18 and Ŝ9-18 <= Ŝ9-14 then
    Lw1 = Ŝ9-18
End if

If Ŝ9-17 <= Ŝ9-15 and Ŝ9-17 <= Ŝ9-19 then
    Lw2 = Ŝ9-17
Else If Ŝ9-17 > Ŝ9-15 and Ŝ9-15 <= Ŝ9-19 then
    Lw2 = Ŝ9-15
Else Ŝ9-17 > Ŝ9-19 and Ŝ9-19 <= Ŝ9-15 then
    Lw2 = Ŝ9-19
End if

Lw = Lw1 + Lw2
```

Figure 9. Pseudo code to find Whitmore width, L_w



a. ***Lw = Ŝ14-15*** ***L2 = Ŝ9-31***



b. ***Lw = Ŝ18-19*** ***L2 = Ŝ9-31***

Figure 10. Particular cases for Whitmore width

V. Pseudo-Column Lengths

The pseudo-columns are used in different forms to compute the compression capacity of a gusset plate. The three values commonly used correspond to the free distance between the end of the Whitmore width (L_1 and L_3) and at the centerline (L_2), as shown in Figure 11. The auxiliary nodes are found using the following interceptions:

Node 30: Interception of line Ł1-2 with the perpendicular to Ł9-13 from point 9.

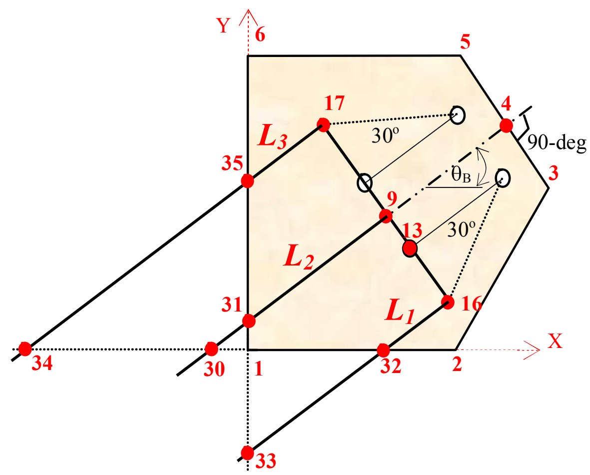
Node 31: Interception of line Ł1-6 with the perpendicular to Ł9-13 from point 9.

Node 32: Interception of line Ł1-2 with the perpendicular to Ł9-13 from point 16.

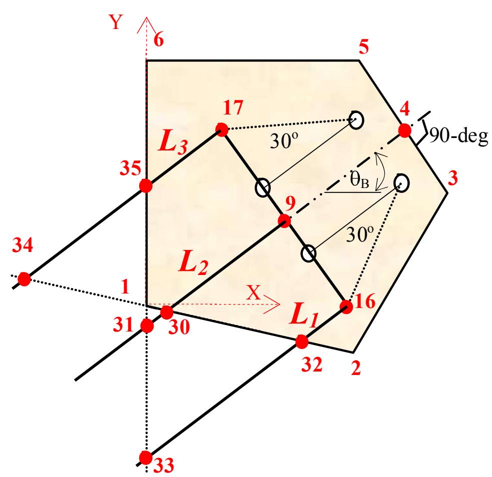
Node 33: Interception of line Ł1-6 with the perpendicular to Ł9-13 from point 16.

Node 34: Interception of line Ł1-2 with the perpendicular to Ł9-13 from point 17.

Node 35: Interception of line Ł1-6 with the perpendicular to Ł9-13 from point 17.



L_1 = Ŝ16-32 ***L_2 = Ŝ9-31*** ***L_3 = Ŝ17-35***



L_1 = Ŝ16-32 ***L_2 = Ŝ9-30*** ***L_3 = Ŝ17-35***

Figure 11. Pseudo-column length

The following relationships apply to find the pseudo-column lengths:

L_1: least value between Ŝ16-32 and Ŝ16-33

L_2: least value between Ŝ9-31 and Ŝ9-30

L_3: least value between Ŝ17-35 and Ŝ17-34

L_1 does not exist if node *16* is outside from the gusset plate, or when L_{w1} is lower than half of L_w. Similarly, L_3 does not exist if node *17* is outside the gusset plate, or when L_{w2} is lower than half of L_w. Figure 10 show cases where only L_2 exists.

VI. Verification of the proposed methodology

Chambers and Bartley (2007) published an algorithm to find the Whitmore width and the associated pseudo columns based on geometric concepts. The twelve cases presented are satisfactorily verified using the spreadsheet based in the Analytical Geometry presented in this paper.

The advantage of this method is its capability to make a scaled drawing, because the coordinates of the nodes and the line incidences are computed during the process. The graphical presentation permits the students to visualize the geometry of the gusset plate and the parameters under study.

Figure 12 shows the spreadsheet input and numerical results for one of the examples given by Chambers and Bartley, observing that both are equivalent. Finally, Figure 13 presents the scaled graph done with the information previously input. The program uses the coordinates and incidences to make this graph.

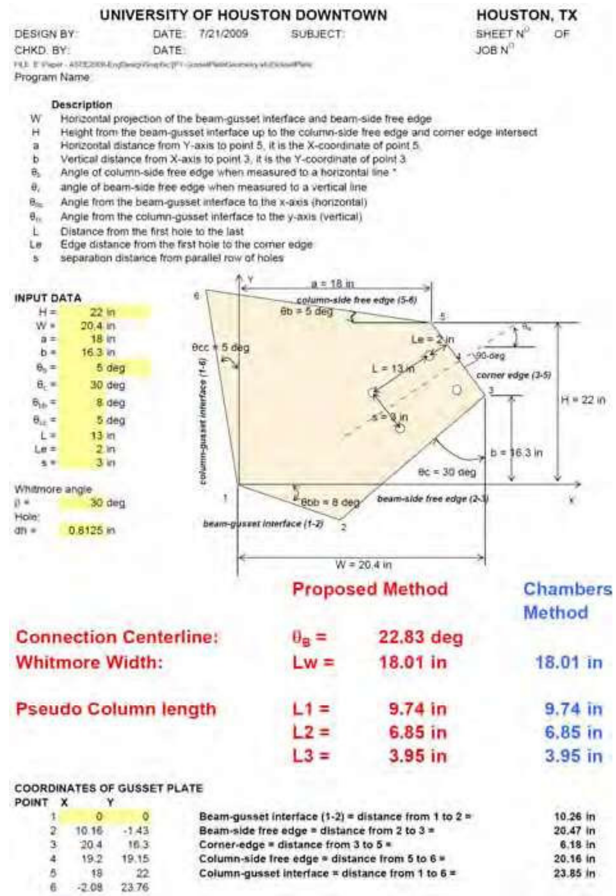


Figure 12. Spreadsheet: Input (in yellow) and Results

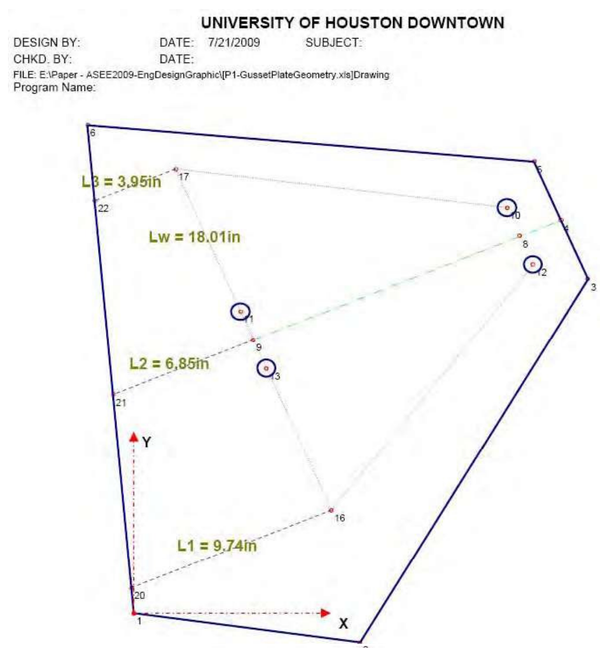


Figure 13. Scaled graph of the gusset plate, the Whitmore width and the pseudo column

VII. Conclusions

The paper presents a methodology that combines Analytic Geometry with an Excel spreadsheet prepared with VBA programs that draw scaled graphs, and with VBA functions to compute interceptions, parallel lines, and perpendicular lines, among other applications. The graphical presentation permits the students a visualization of the gusset plate design and also obtains parameters that can be used for further calculations. In the engineering practice, this type of problems is frequently solved only with graphical methods.

VIII. References

American Institute of Steel Construction, AISC (2005), *Steel Construction Manual*, 13^{th} Ed., Chicago, Il.

Chambers, J., Bartley, T. (2007). *Geometric Formulas for Gusset Plate Design*, Engineering Journal, American Institute of Steel Design, Third Quarter, pg 255-267, 2007.

Microsoft Office Excel 2003 and 2007, *part of Microsoft Office Professional Edition 2003 and 2007.* Microsoft Corporation.

Tito, J., Gomez-Rivas, A. (2007). *Use of Spreadsheets with Scaled Graphics to Teach Structural Engineering*, American Society for Engineering Education, AC2007-397.