Manual for Software Utility for Extracting Fragility Curves

Fragility\_Extractor\_1\_0\_0\_10.zip

# Provenance

This software tool was developed by Virginia Tech’s Resilient-Sustainable Buildings project c.2015-2019, with funding from the US National Science Foundation (CMMI #1455466). The tool was created by former VT student Haseeb Tahir and Co-PI Matthew Eatherton. PI Madeleine Flint finalized the documentation for the repository.

The software tool aids in the digitization and recovery of parameters associated with performance-based engineering analyses, e.g., “fragility curves.” The tool was used to obtain for Intensity Measure (IM)-Engineering Demand Parameter (EDP) curves and IM-Damage State (DS) curves. The tool is especially designed for IM-EDP curves coming from Incremental Dynamic Analysis (IDA), and can translate graphical representations of such curves into a limit number of parameters. The types of parameters produced are:

* Mean (*m*)
* Variance (*v*)
* Lognormal Mean ()
* Lognormal Standard deviation () / Dispersion (*β*)
* Median (*θ*)
* Spectral acceleration at collapse (median, *SaC*)

The tool uses least squares regression to fit a lognormal distribution to provided or digitized data. These parameters are produced by one of 6 methods appropriate to the available data.

# Use Cases

Researchers looking for a lightweight tool for customized for extraction of PBE data:

* Verify that your operating system is compatible with the tool (Section 3)
* Review Section 4

Researchers interested in building a more robust extractor:

* Scan Section 4
* Review the source code in Section 5 and modify as needed (in-tool digitization would be of great interest)

Researchers seeking to better understand the origin of parameters in related databases:

* Review Section 4.3 and related material in Section 6.2

# Technologies

Seven tools were developed in MATLAB and later on Visual studio was to combine these tools into one program and to develop a graphical user interface. The tool has been verified to run on MS Windows 10. It does not run on MacOS.

# How to Use

Go to the download directory and extract the FragilityExtractor.v1.0.0.1.zip file. Run (double-click) the ‘Fragility Curve Extractor.exe’ file. Navigate through the tabs. You will need to have already digitized points and retrieved their coordinates to use the tool.

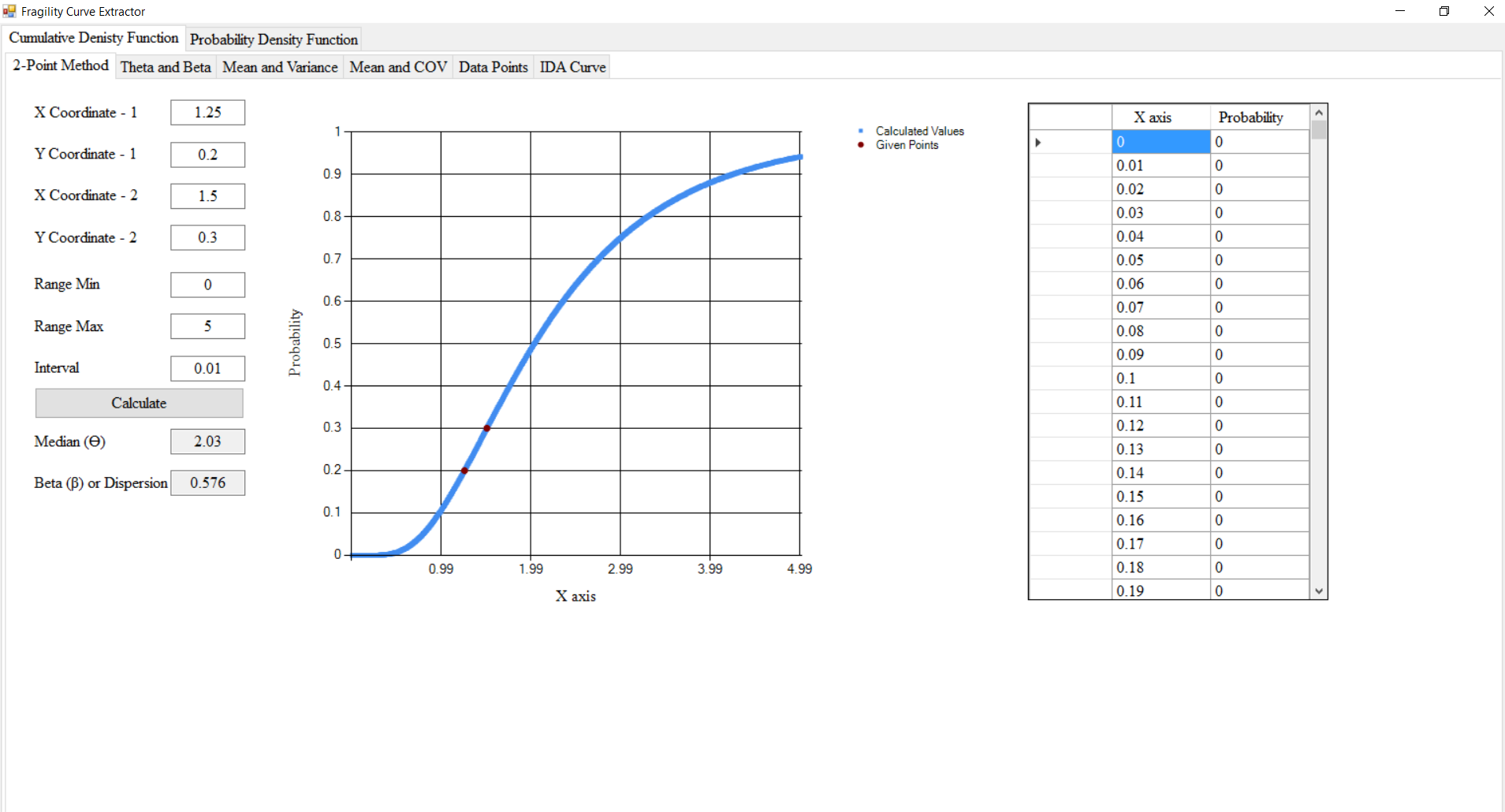
## Description of tabs

There are two main tabs: cumulative density function (CDF) and probability density function (PDF). The cumulative density function tab is further divided into following six sub-tabs.

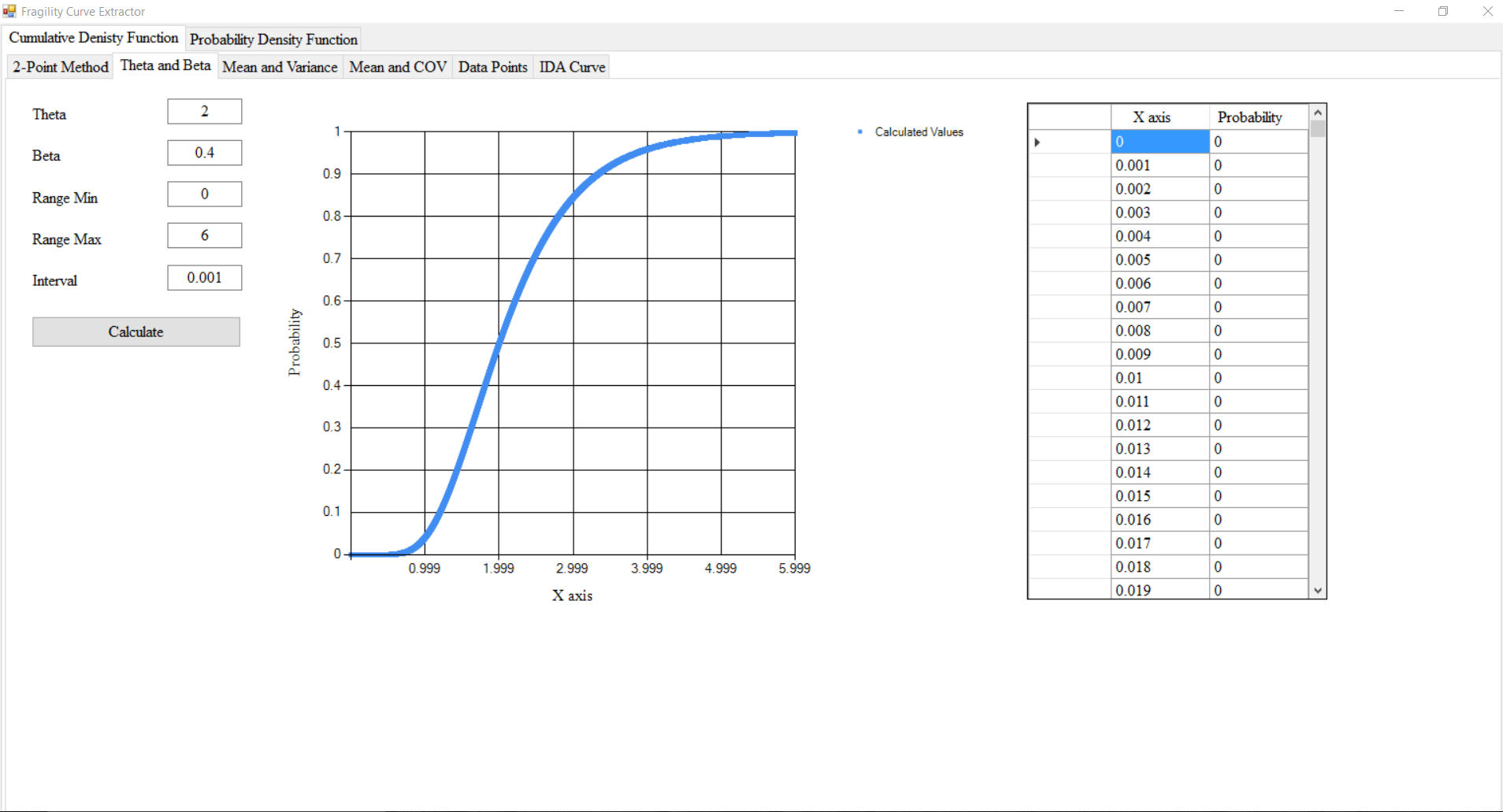
|  |  |
| --- | --- |
| *2-Point Method* | 2-Point Method can be used when coordinates of two points on a lognormal CDF are known. |
| *Theta and Beta* | Draw the CDF and calculate probability at intervals of x values given median (*θ*) and dispersion (*β*) values. |
| *Mean and Variance* | Takes input mean (*m*) and variance (*v*) and calculates lognormal distribution parameters. Can draw the CDF and calculate probabilities. |
| *Mean and COV* | Takes input mean (*m*) and coefficient of variation (COV=) and calculates lognormal distribution parameters. Draws the CDF and calculate probabilities. |
| *Data Points* | When points are not from a CDF it allows to input coordinates of more than two points. It is best to use this tool when data points are not on a smooth CDF that can be represented by one median and dispersion. |
| *IDA Curve* | Find lognormal dispersion values for IDA curves at regular intervals of x-axis values. Slope and Spectral acceleration at collapse values for 16%, 50%, and 84% quantiles (i.e., ) are input. See Figure 1 for an explanation of these terms. |
| *Probability Density Function* | Works similar to the 2-point method for CDFs, except the coordinates entered into this tool are from a probability density function. |

Each tab requires inputting ‘Range min’ and ‘Range max’ to define the bounds of the variable on the x-axis. It is also crucial that the median of CDF lies within this range. ‘Interval’ defines the interval of variables on the x-axis at which the probability values are calculated. All tools show the results in the form of plot and give output in a table format.

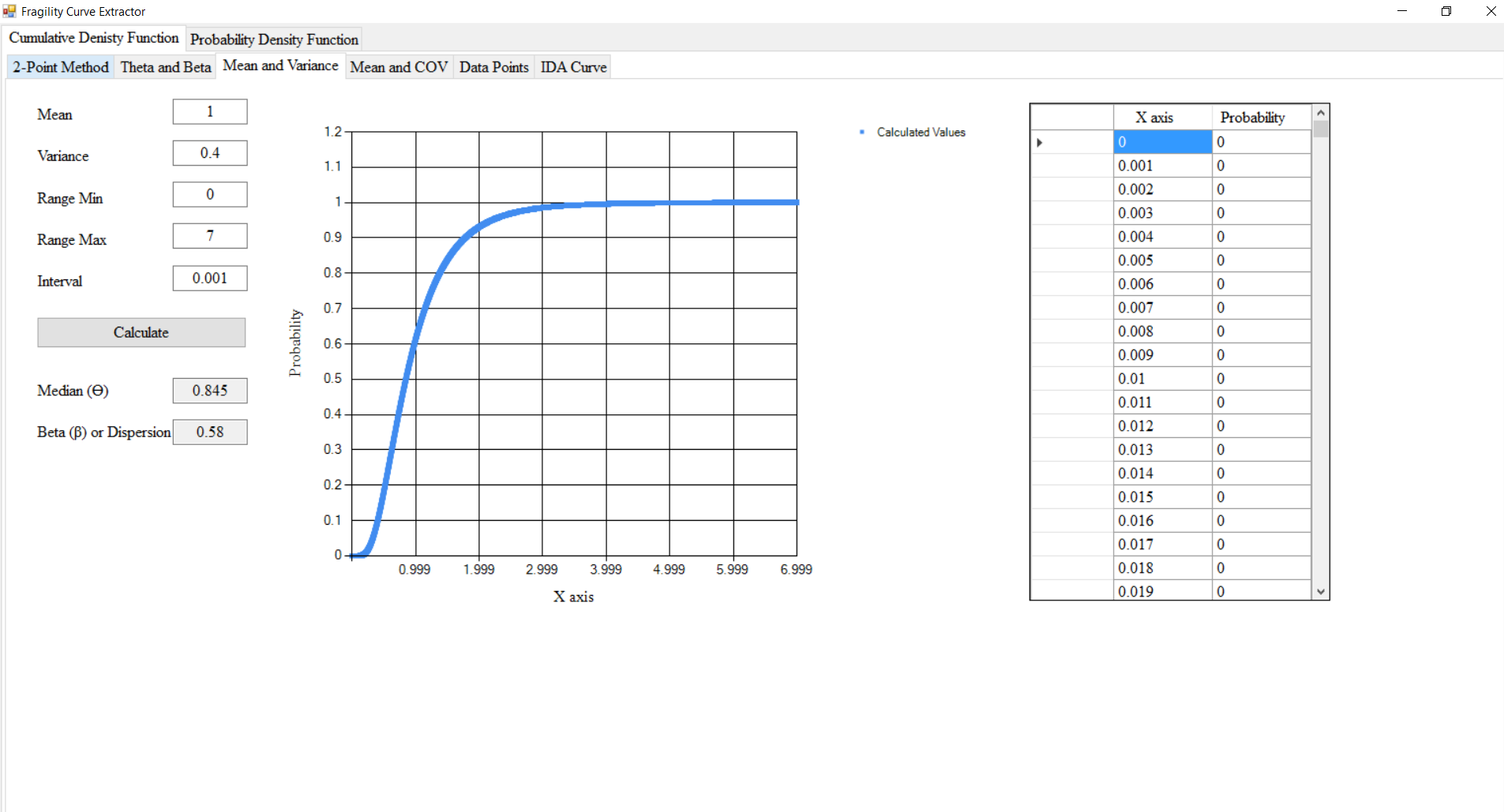
## Screen Shots



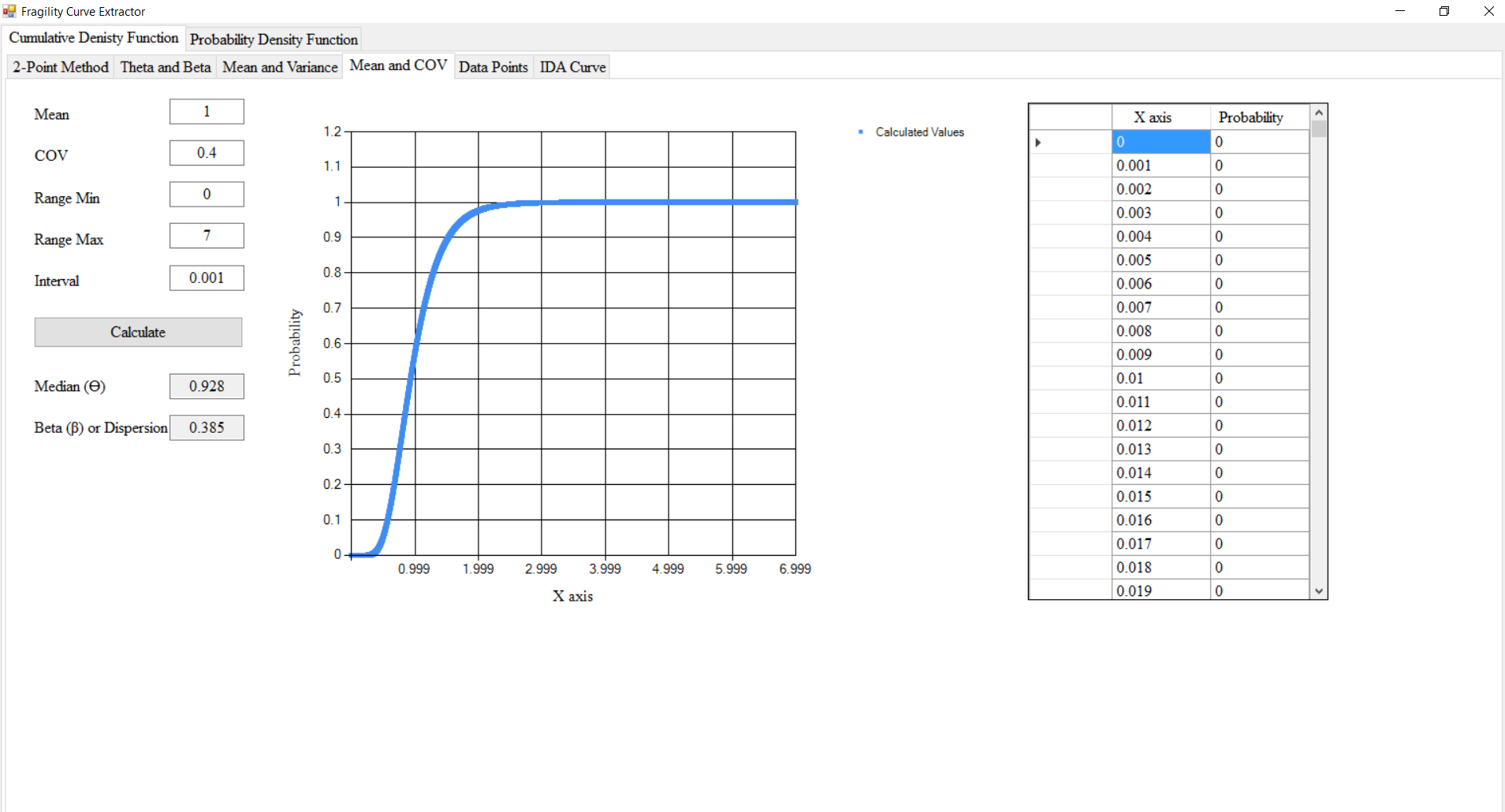
Screen Shot 1. 2-Point Method for CDFs tab



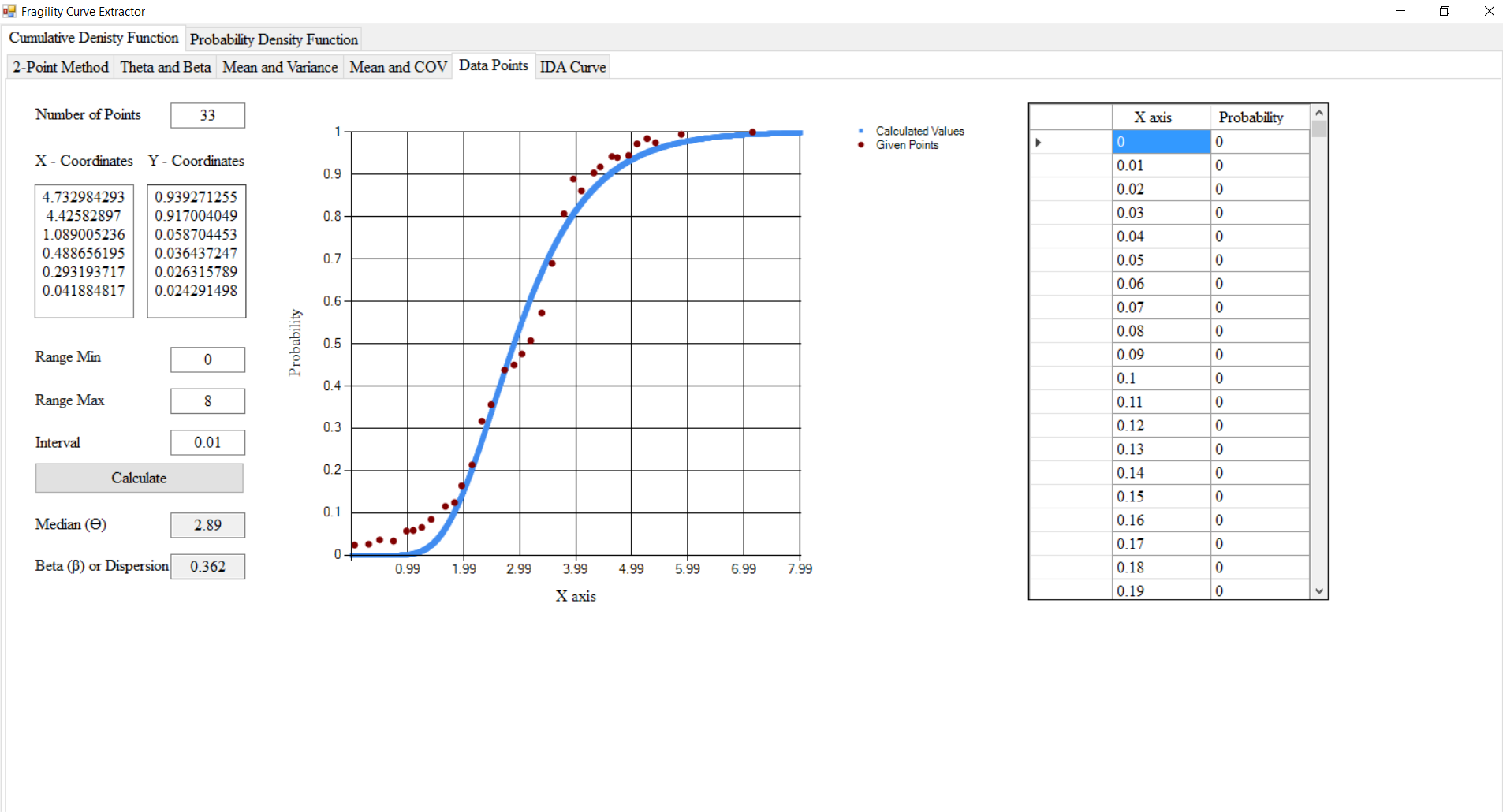
Screen Shot 2. Theta and Beta tab



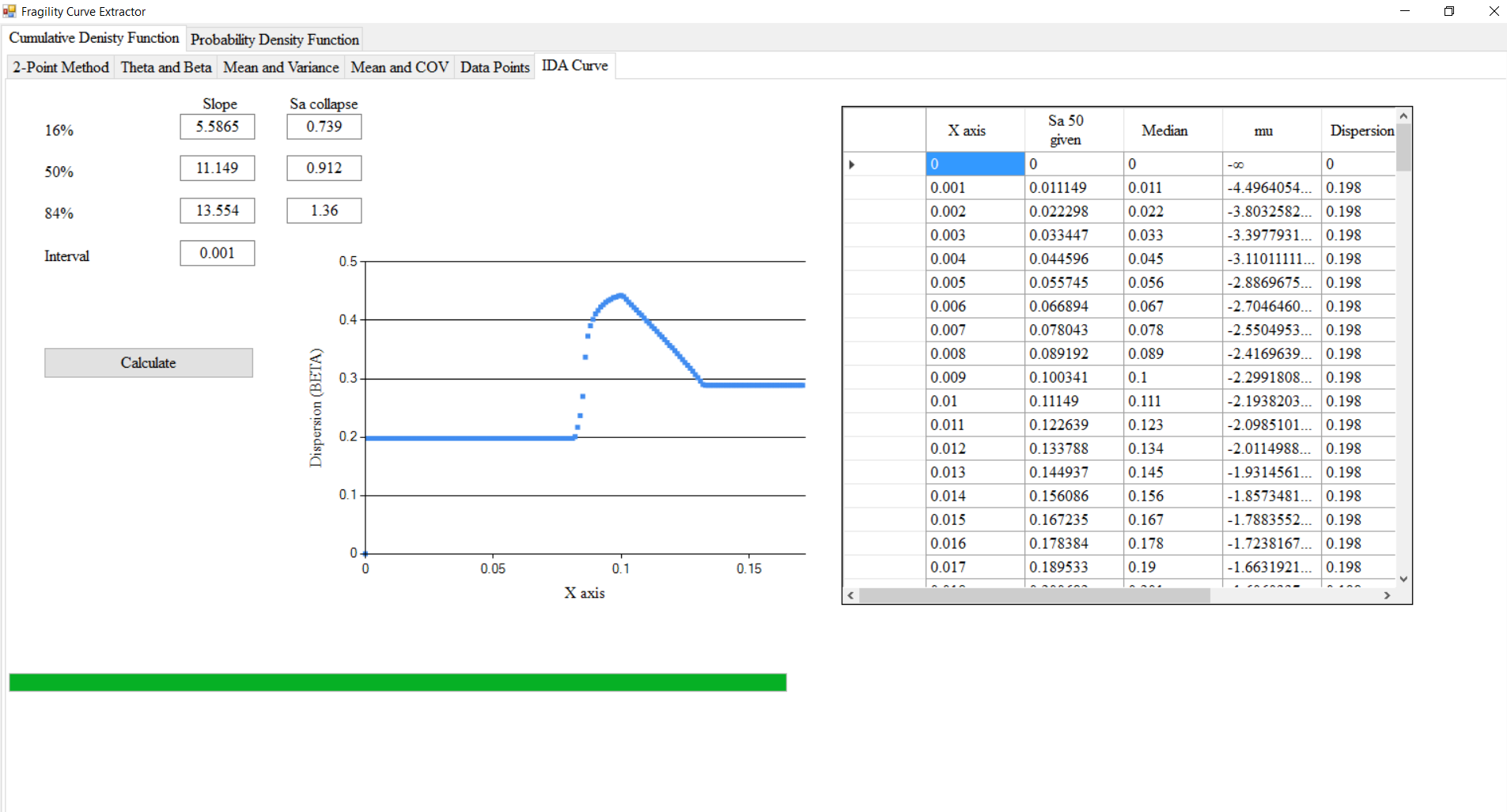
Screen Shot 3. Mean and Variance tab



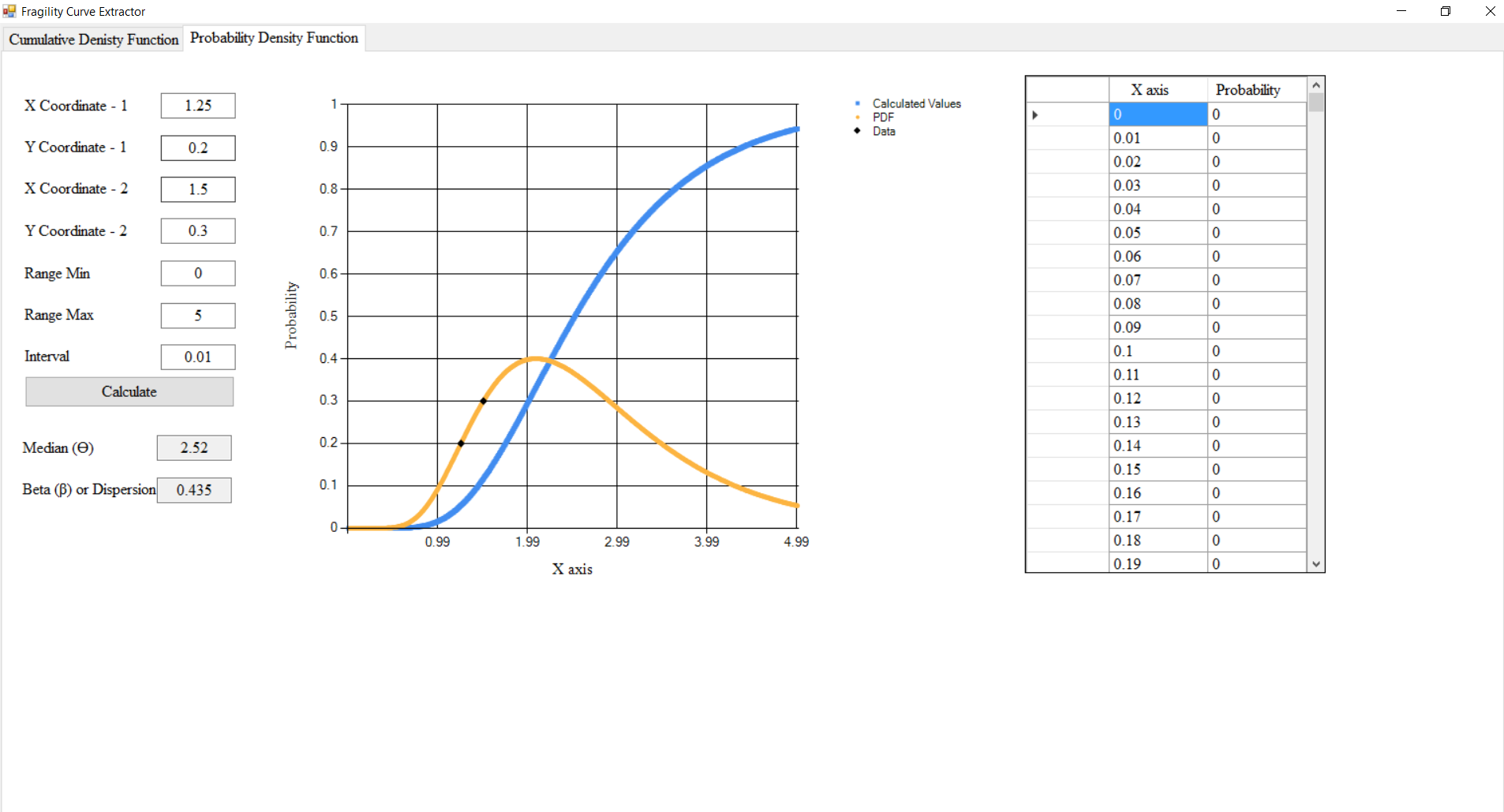
Screen Shot. Mean and COV tab



Screen Shot 5. Data Points tab



Screen Shot 6. IDA Curve tab



Screen Shot 7. Probability Density Function tab

## IDA (IM-EDP) Curves: Obtaining required slopes and collapse spectral accelerations

The original source IDA curves were reported with the 16%, 50%, and 84% quantiles. First, the curves were digitized using web plot digitizer software. After that curves were split into two parts at the point where curve becomes flat (‘Sa collapse’). Regression analysis (minimizing the sum of the square of residuals) was used to find the ‘Slope’ of each of the quantiles, as shown in Figure 1. These slopes and collapse accelerations are input into the *FragilityExtractor* tool to perform an additional regression analysis was used again to find the dispersion of the lognormal distribution assuming homoscedasticity.

Figure 1. Regression analysis of pre-collapse IDA values applied to data from Pitilakis et al., (2014)

# Visual Basic Source Code

Public  Class  Form1

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim  screenHeight  As  Integer  = My.Computer.Screen.Bounds.Height

Dim  screenWidth  As  Integer  = My.Computer.Screen.Bounds.Width

Me.Height  = screenHeight  -­‐ 35

Me.Width  = screenWidth

TabControl1.Height  = screenHeight -­‐  45

TabControl1.Width  = screenWidth -­‐  10

TabControl2.Height  = screenHeight -­‐  55

TabControl2.Width  = screenWidth -­‐  10

TextBox1.Text  = 1.25

TextBox2.Text  = 0.2

TextBox3.Text  = 1.5

TextBox4.Text  = 0.3

TextBox6.Text  = 0

TextBox7.Text  = 5

TextBox12.Text  = 0.01

TextBox57.Text  = 2

TextBox55.Text  = 0.4

TextBox53.Text  = 0

TextBox52.Text  = 6

TextBox51.Text  = 0.001

TextBox28.Text  = 1

TextBox26.Text  = 0.4

TextBox31.Text  = 0

TextBox30.Text  = 7

TextBox29.Text  = 0.001

TextBox38.Text  = 1

TextBox36.Text  = 0.4

TextBox34.Text  = 0

TextBox33.Text  = 7

TextBox32.Text  = 0.001

TextBox24.Text  = 1.25

TextBox23.Text  = 0.2

TextBox22.Text  = 1.5

TextBox21.Text  = 0.3

TextBox19.Text  = 0

TextBox18.Text  = 5

TextBox13.Text  = 0.01

TextBox50.Text  = 4

TextBox48.Text  = 0

TextBox49.Text  = 0

TextBox46.Text  = 0

TextBox45.Text  = 5

TextBox40.Text  = 0.01

TextBox42.Text  = 5.5865

TextBox44.Text  = 0.739

TextBox41.Text  = 11.149

TextBox43.Text  = 0.912

TextBox39.Text  = 13.554

TextBox37.Text  = 1.36

TextBox17.Text  = 0.001

TextBox24.Text  = 1.25

TextBox23.Text  = 0.2

TextBox22.Text  = 1.5

TextBox21.Text  = 0.3

TextBox19.Text  = 0

TextBox18.Text  = 5

TextBox13.Text  = 0.01

End  Sub

Private Sub Button1\_Click\_1(sender As Object, e As EventArgs) Handles Button1.Click

Cursor  = Cursors.WaitCursor

Dim  x1  As  Double

Dim  y1  As  Double

Dim  x2  As  Double

Dim  y2  As  Double

Dim  min  As  Double

Dim  max  As  Double

Dim  interval  As  Double

Dim  mu  As  Double

Dim  sigma  As  Double

Dim  p1  As  Double

Dim  p2  As  Double

Dim  p As  Double

Dim  check  As  Integer

Dim  theta  As  Double

Dim  diff  As  Double

Dim  diffcheck  As  Double

x1  = Nothing

y1  = Nothing

x2  = Nothing

y2  = Nothing

min  = Nothing

max  = Nothing

interval  = Nothing

mu  = Nothing

sigma  = Nothing

p1  = Nothing

p2  = Nothing

p  = Nothing

theta  = Nothing

diff  = Nothing

diffcheck  = 1000

check  = 0

Me.Chart1.Series(0).Points.Clear()

Me.Chart1.Series(1).Points.Clear()

Me.DataGridView1.Rows.Clear()

If  Not  IsNumeric(TextBox1.Text)  Then

TextBox1.Clear()

MsgBox("Please  enter numbers  only.",  vbInformation)

Cursor = Cursors.Default

Exit  Sub

Elseif  Not  IsNumeric(TextBox2.Text)  Then

TextBox2.Clear()

MsgBox("Please  enter numbers  only.", vbInformation)

Cursor = Cursors.Default

Exit  Sub

Elseif  Not  IsNumeric(TextBox3.Text)  Then

TextBox3.Clear()

MsgBox("Please  enter numbers  only.",  vbInformation)

Cursor = Cursors.Default

Exit  Sub

Elseif  Not  IsNumeric(TextBox4.Text)  Then

TextBox4.Clear()

MsgBox("Please  enter numbers  only.",  vbInformation)

Cursor = Cursors.Default

Exit  Sub

Elseif  Not  IsNumeric(TextBox6.Text)  Then

TextBox6.Clear()

Cursor  = Cursors.Default

MsgBox("Please  enter numbers  only.",  vbInformation)   
Exit Sub

Elseif  Not  IsNumeric(TextBox7.Text)  Then

TextBox7.Clear()

MsgBox("Please  enter numbers  only.",  vbInformation)

Cursor = Cursors.Default

Exit  Sub

Elseif  Not  IsNumeric(TextBox12.Text)  Then

TextBox12.Clear()

MsgBox("Please  enter numbers  only.",  vbInformation)

Cursor = Cursors.Default

Exit  Sub

End  If

x1  = TextBox1.Text

y1  = TextBox2.Text

x2  = TextBox3.Text

y2  = TextBox4.Text

min  = TextBox6.Text

max  = TextBox7.Text

interval  = TextBox12.Text

For  m As  Double  = min  To  max  Step  interval

For d As  Double  = 0  To  1 Step  0.001

mu = Math.Log(m)

p1  = MathNet.Numerics.Distributions.LogNormal.CDF(mu,  d, x1)

p2  = MathNet.Numerics.Distributions.LogNormal.CDF(mu,  d, x2)   
diff  = (y1  -­‐ p1)  ^ 2 + (y2  -­‐ p2) ^ 2

If  diff  < diffcheck  Then

theta = m

sigma = d

diffcheck  = diff

End  If

Next

Next

TextBox8.Text  = Math.Round(theta,  3)

TextBox9.Text  = Math.Round(sigma,  3)

For  x As  Double  = min  To  max  Step  interval

p  = MathNet.Numerics.Distributions.LogNormal.CDF(Math.Log(theta),  sigma, x) Me.Chart1.Series(0).Points.AddXY(Math.Round(x,  3), Math.Round(p,  3))

Me.DataGridView1.Rows.Add(Math.Round(x,  3), Math.Round(p,  3))

Next

Me.Chart1.Series(1).Points.AddXY(x1,  y1)

Me.Chart1.Series(1).Points.AddXY(x2,  y2)

Cursor  = Cursors.Default

End  Sub

Private  Sub  Button2\_Click(sender  As  Object,  e As  EventArgs)  Handles  Button2.Click

Cursor  = Cursors.WaitCursor     
Dim x1  As  Double     
Dim y1  As  Double     
Dim x2  As  Double     
Dim y2  As  Double     
Dim min  As  Double     
Dim max  As  Double     
Dim interval  As  Double     
Dim mu  As  Double     
Dim sigma  As  Double     
Dim p1  As  Double     
Dim p2  As  Double

Dim  p As  Double     
Dim check  As  Integer     
Dim theta  As  Double     
Dim diff  As  Double     
Dim diffcheck  As  Double

x1  = Nothing

y1  = Nothing

x2 = Nothing

y2 = Nothing

min = Nothing

max = Nothing

interval = Nothing

mu = Nothing

sigma = Nothing

p1 = Nothing

p2 = Nothing

p = Nothing

theta = Nothing

diff = Nothing

diffcheck = 1000

check = 0

Me.Chart2.Series(0).Points.Clear()

Me.Chart2.Series(1).Points.Clear()

Me.Chart2.Series(2).Points.Clear()

Me.DataGridView2.Rows.Clear()

If  Not  IsNumeric(TextBox24.Text)  Then

TextBox24.Clear()

MsgBox("Please  enter numbers  only.",  vbInformation)   
Cursor Cursors.Default   
Exit Sub

Elseif  Not  IsNumeric(TextBox23.Text)  Then

TextBox23.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)   
Cursor Cursors.Default   
Exit Sub

Elseif  Not  IsNumeric(TextBox22.Text)  Then

TextBox22.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)   
Cursor Cursors.Default   
Exit Sub

Elseif  Not  IsNumeric(TextBox21.Text)  Then

TextBox21.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)   
Cursor Cursors.Default   
Exit Sub

Elseif  Not  IsNumeric(TextBox19.Text)  Then

TextBox19.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)   
Cursor Cursors.Default   
Exit Sub

Elseif  Not  IsNumeric(TextBox18.Text)  Then

TextBox18.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)

Cursor  = Cursors.Default   
Exit Sub

Elseif  Not  IsNumeric(TextBox13.Text)  Then

TextBox13.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)   
Cursor Cursors.Default   
Exit Sub

End  If

x1  = TextBox24.Text

y1  = TextBox23.Text

x2 = TextBox22.Text

y2 = TextBox21.Text

min  = TextBox19.Text

max = TextBox18.Text

interval = TextBox13.Text

For  m As  Double  = min  To  max  Step  interval

For d As  Double  = 0  To  1 Step  0.001   mu = Math.Log(m)

p1  = MathNet.Numerics.Distributions.LogNormal.PDF(mu,  d, x1)

p2  = MathNet.Numerics.Distributions.LogNormal.PDF(mu,  d, x2)

diff  = (y1  -­‐ p1)  ^ 2 + (y2  -­‐ p2) ^ 2

If  diff  < diffcheck  Then

theta = m sigma = d diffcheck  = diff

End  If

Next

Next

TextBox16.Text  = Math.Round(theta,  3)

TextBox15.Text  = Math.Round(sigma,  3)

For  x As  Double  = min  To  max  Step  interval

p  = MathNet.Numerics.Distributions.LogNormal.CDF(Math.Log(theta),  sigma, x) Me.Chart2.Series(0).Points.AddXY(Math.Round(x,  3), Math.Round(p,  3))

Me.DataGridView2.Rows.Add(Math.Round(x,  3), Math.Round(p,  3))

Next

For  x As  Double  = min  To  max  Step  interval

p  = MathNet.Numerics.Distributions.LogNormal.PDF(Math.Log(theta),  sigma, x)

Me.Chart2.Series(1).Points.AddXY(Math.Round(x,  3), Math.Round(p,  3))

Next

Me.Chart2.Series(2).Points.AddXY(x1,  y1)

Me.Chart2.Series(2).Points.AddXY(x2,  y2)

Cursor  = Cursors.Default

End  Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

Cursor  = Cursors.WaitCursor     
Dim mean  As  Double     
Dim variance  As  Double     
Dim min  As  Double     
Dim max  As  Double     
Dim interval  As  Double     
Dim mu  As  Double     
Dim sigma  As  Double     
Dim p As  Double     
Dim theta  As  Double     
Dim beta  As  Double

min  = Nothing

max  = Nothing

interval = Nothing

mu = Nothing

sigma = Nothing

p = Nothing

mean = Nothing

variance = Nothing

theta = Nothing

beta = Nothing

Me.Chart3.Series(0).Points.Clear()

Me.DataGridView3.Rows.Clear()

If  Not  IsNumeric(TextBox28.Text)  Then

TextBox28.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox26.Text)  Then

TextBox26.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox31.Text)  Then

TextBox31.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox30.Text)  Then

TextBox30.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox29.Text)  Then

  TextBox29.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub

End  If

mean  = TextBox28.Text

variance  = TextBox26.Text

min = TextBox31.Text

max = TextBox30.Text interval  = TextBox29.Text

mu  = Math.Log((mean  ^ 2) / Math.Sqrt(variance  + mean ^ 2))

sigma  = Math.Sqrt(Math.Log(variance  / (mean ^ 2)  + 1))

beta = sigma

theta = mean / Math.Sqrt(Math.Exp(beta  ^ 2))

TextBox11.Text  = Math.Round(theta,  3)

TextBox10.Text  = Math.Round(beta,  3)

For  x As  Double  = min  To  max  Step  interval

p  = MathNet.Numerics.Distributions.LogNormal.CDF(mu,  sigma, x)

Me.Chart3.Series(0).Points.AddXY(Math.Round(x,  3), Math.Round(p,  3))

Me.DataGridView3.Rows.Add(Math.Round(x,  3), Math.Round(p,  3))

Next

Cursor  = Cursors.Default

End  Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

Cursor = Cursors.WaitCursor   
Dim mean  As  Double     
Dim cov  As  Double     
Dim variance  As  Double     
Dim min  As  Double     
Dim max  As  Double     
Dim interval  As  Double     
Dim mu  As  Double     
Dim sigma  As  Double     
Dim p As  Double     
Dim theta  As  Double     
Dim beta  As  Double

cov  = Nothing

min  = Nothing

max = Nothing

interval = Nothing

mu = Nothing

sigma = Nothing

p = Nothing

mean = Nothing

variance  = Nothing

theta = Nothing

beta = Nothing

Me.Chart4.Series(0).Points.Clear()

Me.DataGridView4.Rows.Clear()

If  Not  IsNumeric(TextBox38.Text)  Then TextBox38.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox36.Text)  Then TextBox36.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox34.Text)  Then TextBox34.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox33.Text)  Then TextBox33.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox32.Text)  Then TextBox32.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub

End  If

mean  = TextBox38.Text

cov  = TextBox36.Text

min = TextBox34.Text

max = TextBox33.Text

interval  = TextBox32.Text

variance  = (mean \*  cov) ^ 2

mu  = Math.Log((mean  ^ 2) / Math.Sqrt(variance  + mean ^ 2))

sigma = Math.Sqrt(Math.Log(variance  / (mean ^ 2)  + 1))

beta = sigma

theta = mean / Math.Sqrt(Math.Exp(beta  ^ 2))

TextBox25.Text  = Math.Round(theta,  3)

TextBox14.Text  = Math.Round(beta,  3)

For  x As  Double  = min  To  max  Step  interval

p  = MathNet.Numerics.Distributions.LogNormal.CDF(mu,  sigma, x)

Me.Chart4.Series(0).Points.AddXY(Math.Round(x,  3), Math.Round(p,  3))

Me.DataGridView4.Rows.Add(Math.Round(x,  3), Math.Round(p,  3))

Next

Cursor  = Cursors.Default

End  Sub

Private  Sub  Button5\_Click(sender  As  Object,  e As  EventArgs)  Handles  Button5.Click

Cursor Cursors.WaitCursor     
 Dim x1  As  Double     
 Dim y1  As  Double     
 Dim x2  As  Double     
 Dim y2  As  Double     
 Dim min  As  Double     
 Dim max  As  Double     
 Dim interval  As  Double     
 Dim mu  As  Double     
 Dim sigma  As  Double     
 Dim p As  Double     
 Dim check  As  Integer     
 Dim l As  Integer     
 Dim theta  As  Double     
 Dim diff  As  Double     
 Dim diffcheck  As  Double

l  = Nothing

x1  = Nothing

y1 = Nothing

x2 = Nothing

y2 = Nothing

min = Nothing

max = Nothing

interval = Nothing

mu = Nothing

sigma = Nothing

p = Nothing

theta = Nothing

diffcheck = 1000

check  = 0

Me.Chart5.Series(0).Points.Clear()

Me.Chart5.Series(1).Points.Clear()

Me.DataGridView5.Rows.Clear()

If  Not  IsNumeric(TextBox40.Text)  Then

TextBox40.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox45.Text)  Then

TextBox45.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default     
 Exit Sub     
Elseif  Not  IsNumeric(TextBox46.Text)  Then

TextBox46.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox50.Text)  Then

TextBox50.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub

End  If

l  = TextBox50.Text     
Dim coor(l,  2) As  Double     
Dim prob(l)  As  Double

If  l <> TextBox48.Lines.Count Or l <> TextBox49.Lines.Count Or TextBox48.Lines.Count <> TextBox49.Lines.Count Then   
 MsgBox("Number  of Coordinates  is not Right",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub

End  If

For  x As  Integer  = 0  To  l -­‐  1

coor(x,  1) = TextBox48.Lines(x)

coor(x, 2) = TextBox49.Lines(x)

Next

min  = TextBox46.Text

max  = TextBox45.Text

interval = TextBox40.Text

For  m As  Double  = min  To  max  Step  interval

For  d As  Double  = 0  To  1 Step  0.001

mu = Math.Log(m)

For  x As  Integer  = 0  To  l -­‐  1

prob(x) = MathNet.Numerics.Distributions.LogNormal.CDF(mu, d, coor(x, 1))

Next

diff  = 0

For  x As  Integer  = 0  To  l -­‐  1

diff  = diff  + (prob(x)  -­‐ coor(x, 2))  ^ 2

Next

If  diff  < diffcheck  Then

theta  = m sigma = d

diffcheck = diff

End  If

Next

Next

TextBox35.Text  = Math.Round(theta,  3)

TextBox27.Text  = Math.Round(sigma,  3)

For  x As  Double  = min  To  max  Step  interval

p  = MathNet.Numerics.Distributions.LogNormal.CDF(Math.Log(theta),  sigma, x) Me.Chart5.Series(0).Points.AddXY(Math.Round(x,  3), Math.Round(p,  3))

Me.DataGridView5.Rows.Add(Math.Round(x,  3), Math.Round(p,  3))

Next

For  x As  Integer  = 0  To  l -­‐  1

Me.Chart5.Series(1).Points.AddXY(coor(x,  1), coor(x, 2))

Next

Cursor  = Cursors.Default

End  Sub

Private  Sub  Button6\_Click(sender  As  Object,  e As  EventArgs)  Handles  Button6.Click

Cursor  = Cursors.WaitCursor     
Dim theta  As  Double     
Dim beta  As  Double     
Dim mean  As  Double     
Dim variance  As  Double     
Dim min  As  Double     
Dim max  As  Double     
Dim interval  As  Double     
Dim mu  As  Double     
Dim sigma  As  Double     
Dim p As  Double

min  = Nothing

max  = Nothing

interval = Nothing

mu  = Nothing

sigma = Nothing

p = Nothing

mean = Nothing

variance = Nothing

theta = Nothing

beta = Nothing

Me.Chart6.Series(0).Points.Clear()

Me.DataGridView6.Rows.Clear()

If  Not  IsNumeric(TextBox57.Text)  Then

TextBox57.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)   
Cursor Cursors.Default   
Exit Sub

Elseif  Not  IsNumeric(TextBox55.Text)  Then

TextBox55.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)   
Cursor Cursors.Default   
Exit Sub

Elseif  Not  IsNumeric(TextBox53.Text)  Then

TextBox53.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)   
Cursor Cursors.Default   
Exit Sub

Elseif  Not  IsNumeric(TextBox52.Text)  Then

TextBox52.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)   
Cursor Cursors.Default   
Exit Sub

Elseif  Not  IsNumeric(TextBox51.Text)  Then

TextBox51.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)   
Cursor Cursors.Default   
Exit Sub

End  If

theta  = TextBox57.Text

beta  = TextBox55.Text

min = TextBox53.Text

max = TextBox52.Text

interval  = TextBox51.Text

mean  = theta  \* Math.Sqrt(Math.Exp(beta  ^ 2))

variance  = mean ^  2 \* (Math.Exp(beta  ^ 2) -­‐ 1)

For  x As  Double  = min  To  max  Step  interval

mu  = Math.Log((mean  ^ 2) / Math.Sqrt(variance  + mean ^ 2))

sigma = Math.Sqrt(Math.Log(variance  / (mean ^ 2)  + 1))

p = MathNet.Numerics.Distributions.LogNormal.CDF(mu,  sigma, x)

Me.Chart6.Series(0).Points.AddXY(Math.Round(x,  3), Math.Round(p,  3))

Me.DataGridView6.Rows.Add(Math.Round(x,  3), Math.Round(p,  3))

Next

Cursor  = Cursors.Default

End  Sub

Private  Sub  Button7\_Click(sender  As  Object,  e As  EventArgs)  Handles  Button7.Click

Cursor  = Cursors.WaitCursor

ProgressBar1.Value = 0

Me.Chart7.Series(0).Points.Clear()       
Dim sa16  As  Double     
Dim s16  As  Double     
Dim m16  As  Double     
Dim sa50  As  Double     
Dim s50  As  Double     
Dim m50  As  Double     
Dim sa84  As  Double     
Dim s84  As  Double     
Dim m84  As  Double     
Dim interval  As  Double     
Dim mu  As  Double     
Dim sigma  As  Double     
Dim p1  As  Double     
Dim p2  As  Double     
Dim p3  As  Double     
Dim p As  Double     
Dim check  As  Integer     
Dim theta  As  Double     
Dim maxedp  As  Double     
Dim progress  As  Integer     
Dim diff  As  Double     
Dim diffcheck  As  Double

sa16  = Nothing

s16  = Nothing

m16 = Nothing

sa50 = Nothing

s50 = Nothing

m50 = Nothing

sa84 = Nothing

s84 = Nothing

m84 = Nothing

interval = Nothing

mu = Nothing

sigma = Nothing

p1 = Nothing

p2 = Nothing

p3 = Nothing

p = Nothing

theta = Nothing

maxedp = Nothing

check  = 0

Me.DataGridView7.Rows.Clear()

If  Not  IsNumeric(TextBox37.Text)  Then

  TextBox37.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox39.Text)  Then

TextBox39.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox41.Text)  Then

TextBox41.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub

Elseif  Not  IsNumeric(TextBox42.Text)  Then

TextBox42.Clear()   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Cursor Cursors.Default   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox43.Text)  Then

TextBox43.Clear()   
 Cursor Cursors.Default   
 MsgBox("Please  enter numbers  only.",  vbInformation)   
 Exit Sub     
Elseif  Not  IsNumeric(TextBox44.Text)  Then

TextBox44.Clear()   
 Cursor Cursors.Default   
MsgBox("Please  enter numbers  only.",  vbInformation)   
Exit Sub

Elseif  Not  IsNumeric(TextBox17.Text)  Then

TextBox17.Clear()   
MsgBox("Please  enter numbers  only.",  vbInformation)

Cursor = Cursors.Default   
Exit Sub

End  If

s16  = TextBox42.Text

sa16  = TextBox44.Text

s50 = TextBox41.Text

sa50 = TextBox43.Text

s84  = TextBox39.Text

sa84 = TextBox37.Text

interval = TextBox17.Text

m16 = sa16 / s16

m50 = sa50 / s50

m84 = sa84 / s84

maxedp  = 1.3 \*  Math.Max((Math.Max(m16,  m50)), m84)

progress  = maxedp /  interval

ProgressBar1.Maximum = progress  + 1

progress = 0

For  edp  As  Double  = 0  To  maxedp  Step  interval

mu  = Nothing

sa50  = Nothing     
sigma  = Nothing

diff = Nothing

diffcheck = 1000

If  edp  < m16  Then

sa16  = edp  \* s16   
 Else  sa16 = s16 \* m16

End  If

If  edp  < m50  Then

sa50  = edp  \* s50   
Else  sa50 = s50 \* m50

End  If

If  edp  < m84  Then

sa84  = edp  \* s84   
Else  sa84 = s84 \* m84

End  If

mu  = Math.Log(sa50)

For  d As  Double  = 0  To  1 Step  0.001   
  p1  = MathNet.Numerics.Distributions.LogNormal.CDF(mu,  d, sa16)

p2  = MathNet.Numerics.Distributions.LogNormal.CDF(mu,  d, sa84)

p3 = MathNet.Numerics.Distributions.LogNormal.CDF(mu,  d, sa50)   
  diff  = (0.16  -­‐ p1) ^  2 + (0.84 -­‐  p2) ^ 2 + (0.5  -­‐ p3) ^ 2

If  diff  < diffcheck  Then

sigma = d diffcheck = diff

End  If

Next

theta  = sa50

Me.Chart7.Series(0).Points.AddXY(edp,  Math.Round(sigma,  3))   
Me.DataGridView7.Rows.Add(edp, sa50, Math.Round(theta, 3), mu, Math.Round(sigma, 3), diff, sa50)

ProgressBar1.Value  = progress progress  = progress  + 1

Next

ProgressBar1.Value =  ProgressBar1.Maximum

Cursor  = Cursors.Default

End  Sub

End  Class

# Additional Information

## Citation

Flint M.M, Eatherton M.R., Reichard G., Tahir H., Ladipo O. (2019). Manual for Software Utility for Extracting Fragility Curves v1.0.0.10. *Seismic, Hurricane, and Tsunami Vulnerability Database and Pinch-Point Taxonomy for Mid-Rise Commercial Buildings*. DesignSafe-CI PRJ 1723.

## Related sources and documentation

* Flint M.M., Eatherton M.R., Reichard G., Tahir H., Ladipo O. (2019). Theoretical Basis for Simplified Representation of IDA Curves. *Seismic, Hurricane, and Tsunami Vulnerability Database for Mid-Rise Commercial Buildings*. DesignSafe-CI PRJ 1723. DOI: TBD.
* Tahir, H. (2016). Development of Fragility Curve Database for Multi-Hazard Performance Based Design. Virginia Tech. <http://hdl.handle.net/10919/71794>.
  + In folder as *Tahir-2016-Development of Fragility Curve Database for Multi-Hazard Performance Based Design.pdf*
* ﻿ ﻿Zaker Esteghamati, M, Lee, J, Musetich, M, et al. (2019). INSSEPT: An open-source relational database of seismic performance estimation to aid with early design of buildings. *Earthq. Spectra*, in review.
  + Contains data obtained using tool
  + Flint, M.M., Zaker Esteghamati M., Lee J., Musetich M. (2019). *Inventory of Seismic Structural Evaluations, Performance functions and Taxonomies for buildings (INSSEPT)*. DesignSafe-CI. DOI: ﻿10.17603/ds2-k1g7-vg97.

## Log

c.2015-02 to 2016-05 Development of tool by Haseeb Tahir with oversight by Matthew Eatherton

c.2016-05 to 2017-10 Development of additional documentation by Haseeb Tahir

2019-12-10 Text retrieved from Tahir Thesis by Madeleine Flint; minor formatting updates to create document; addition of provenance, log, and other sections

2019-12-17 Addition of source code.

2019-12-19 Final updates to citation and related sources