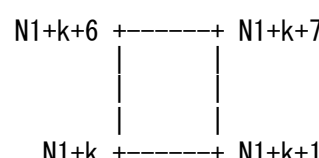


【課題 7-2】 手作業でデータを作るのは面倒なので、以下のようなデータ生成のプログラムを作成します。

```

Public nodeNo As Integer ' 節点番号
Public elemNo As Integer ' 要素番号
Sub genNode(Xnode, Ynode, BX, BY, FX, FY)
  nodeNo = nodeNo + 1: ii = nodeNo + 1
  With Worksheets("節点データ")
    .Cells(ii, 1) = nodeNo ' 節点番号
    .Cells(ii, 2) = Xnode: .Cells(ii, 3) = Ynode ' 座標値
    .Cells(ii, 4) = BX: .Cells(ii, 5) = BY ' 変位境界条件有無
    .Cells(ii, 6) = 0: .Cells(ii, 7) = 0 ' 境界条件の変位
    .Cells(ii, 8) = FX: .Cells(ii, 9) = FY ' 外力境界条件
  End With
End Sub
Sub genElement(N1, N2, N3)
  elemNo = elemNo + 1: ii = elemNo + 1
  With Worksheets("要素データ")
    .Cells(ii, 1) = elemNo ' 要素番号
    .Cells(ii, 2) = N1 ' 構成節点番号(1)
    .Cells(ii, 3) = N2 ' (2)
    .Cells(ii, 4) = N3 ' (3)
    .Cells(ii, 5) = 1 ' 材料番号
  End With
End Sub
Sub ボタン1_Click()
  nodeNo = 0: FX1 = 2: FX2 = -3.5: XX = 0: YY = 0 ' 各種初期値設定
  genNode XX, YY, 1, 1, FX1, 0: ' 節点データの生成 (左から正の水圧)
  For i = 1 To 4 ' 最底箇所はXY固定として生成
    XX = XX + 0.1
    genNode XX, YY, 1, 1, 0, 0
  Next
  XX = XX + 0.1 ' 右端は負の水圧
  genNode XX, YY, 1, 1, FX2, 0
  For j = 1 To 40
    YY = YY + 0.1: XX = 0 ' 高さ方向増加とXの初期値設定
    FX1 = FX1 - 0.1: If FX1 < 0 Then FX1 = 0 ' 水圧の絶対値を減らす
    FX2 = FX2 + 0.1: If FX2 > 0 Then FX2 = 0 ' 水圧の絶対値を減らす
    genNode XX, YY, 0, 0, FX1, 0 ' 左端は正の水圧
    For i = 1 To 4
      XX = XX + 0.1
      genNode XX, YY, 0, 0, 0, 0 ' 変位及び外力境界条件なし
    Next
    XX = XX + 0.1
    genNode XX, YY, 0, 0, FX2, 0 ' 右端は負の水圧
  Next
  elemNo = 0: N1 = 1 ' 要素データの生成
  For j = 1 To 40
    For k = 0 To 4
      genElement N1 + k, N1 + k + 1, N1 + k + 6
      genElement N1 + k + 6, N1 + k + 1, N1 + k + 7
    Next
    N1 = N1 + 6
  Next
End Sub

```

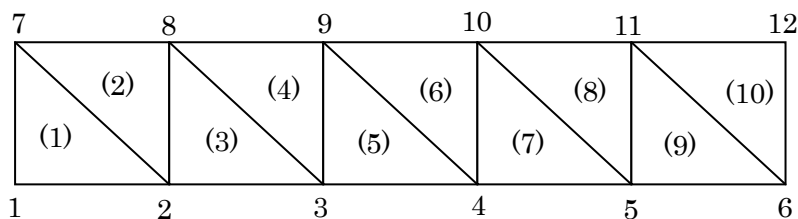


【課題解答例】

■**節点データ** 以下が生成された節点データです。

| | A | B | C | D | E | F | G | H | I |
|----|------|------|------|-------|-------|-----|-----|------|------|
| 1 | 節点番号 | X座標 | Y座標 | X拘束条件 | Y拘束条件 | X変位 | Y変位 | X方向力 | Y方向力 |
| 2 | 1 | 0.00 | 0.00 | 1 | 1 | 0 | 0 | 2.0 | 0.0 |
| 3 | 2 | 0.10 | 0.00 | 1 | 1 | 0 | 0 | 0.0 | 0.0 |
| 4 | 3 | 0.20 | 0.00 | 1 | 1 | 0 | 0 | 0.0 | 0.0 |
| 5 | 4 | 0.30 | 0.00 | 1 | 1 | 0 | 0 | 0.0 | 0.0 |
| 6 | 5 | 0.40 | 0.00 | 1 | 1 | 0 | 0 | 0.0 | 0.0 |
| 7 | 6 | 0.50 | 0.00 | 1 | 1 | 0 | 0 | -3.5 | 0.0 |
| 8 | 7 | 0.00 | 0.10 | 0 | 0 | 0 | 0 | 1.9 | 0.0 |
| 9 | 8 | 0.10 | 0.10 | 0 | 0 | 0 | 0 | 0.0 | 0.0 |
| 10 | 9 | 0.20 | 0.10 | 0 | 0 | 0 | 0 | 0.0 | 0.0 |
| 11 | 10 | 0.30 | 0.10 | 0 | 0 | 0 | 0 | 0.0 | 0.0 |
| 12 | 11 | 0.40 | 0.10 | 0 | 0 | 0 | 0 | 0.0 | 0.0 |
| 13 | 12 | 0.50 | 0.10 | 0 | 0 | 0 | 0 | -3.4 | 0.0 |
| 14 | 13 | 0.00 | 0.20 | 0 | 0 | 0 | 0 | 1.8 | 0.0 |
| 15 | 14 | 0.10 | 0.20 | 0 | 0 | 0 | 0 | 0.0 | 0.0 |
| 16 | 15 | 0.20 | 0.20 | 0 | 0 | 0 | 0 | 0.0 | 0.0 |

■**要素データ** 以下のような順に生成されます。なお、括弧付き番号は要素番号、括弧が付いていない番号は節点番号です。



| | A | B | C | D | E |
|----|------|-----|-----|-----|------|
| 1 | 要素番号 | 節点1 | 節点2 | 節点3 | 材料番号 |
| 2 | 1 | 1 | 2 | 7 | 1 |
| 3 | 2 | 7 | 2 | 8 | 1 |
| 4 | 3 | 2 | 3 | 8 | 1 |
| 5 | 4 | 8 | 3 | 9 | 1 |
| 6 | 5 | 3 | 4 | 9 | 1 |
| 7 | 6 | 9 | 4 | 10 | 1 |
| 8 | 7 | 4 | 5 | 10 | 1 |
| 9 | 8 | 10 | 5 | 11 | 1 |
| 10 | 9 | 5 | 6 | 11 | 1 |
| 11 | 10 | 11 | 6 | 12 | 1 |
| 12 | 11 | 7 | 8 | 13 | 1 |
| 13 | 12 | 13 | 8 | 14 | 1 |
| 14 | 13 | 8 | 9 | 14 | 1 |
| 15 | 14 | 14 | 9 | 15 | 1 |
| 16 | 15 | 9 | 10 | 15 | 1 |

■変位境界条件や外力境界条件の表示 課題のたびに境界条件を図示するのは面倒なので、この部分もプログラム化します。ただし、変位境界条件については三角形をどの向きに描くかを指定しなければいけません。一方、変位境界条件は境界条件の有無を非ゼロ、ゼロで判別しています。そこで、境界条件の正負で三角形の向きを指定することにします。詳細についてはプログラム中の注釈を参照されたい。

```

Sub drawExternalForce()
  With Worksheets("節点データ")
    Sheets("図").Select      ' 右向き/上向きが正, 左向き/下向きが負
    i = 2                    ' となるので矢印の先が節点座標となる
    Do While .Cells(i, 1) <> ""
      X2 = extX(Val(.Cells(i, 2))): Y2 = extY(Val(.Cells(i, 3))) ' シート座標に変換
      DX = Val(.Cells(i, 8)) * 50: DY = Val(.Cells(i, 9)) * 50
      BX = Val(.Cells(i, 4)): BY = Val(.Cells(i, 5))
      X1 = X2 - DX: Y1 = Y2 + DY
      If DX <> 0 Or DY <> 0 Then drawVector X1, Y1, X2, Y2, 1, 8, "Arrow_" & i - 1
      If BX <> 0 Or BY <> 0 Then drawDispBoundary i, X2, Y2, BX, BY
      i = i + 1
    Loop
  End With
End Sub
Sub drawVector(X1, Y1, X2, Y2, W, C, Aname) ' 矢印の表示
  On Error Resume Next
  ActiveSheet.Shapes(Aname).Delete: Err = 0
  ActiveSheet.Shapes.AddLine(X1, Y1, X2, Y2).Select
  Selection.Name = Aname
  With Selection.ShapeRange.Line
    .EndArrowheadStyle = msoArrowheadTriangle
    .EndArrowheadLength = msoArrowheadLengthMedium
    .EndArrowheadWidth = msoArrowheadWidthMedium
    .Visible = msoTrue
    .Weight = W
    .Style = msoLineSingle
    .ForeColor.SchemeColor = C ' C=10:赤, 12:青, 17:緑, 8:黒
  End With
End Sub
Sub drawDispBoundary(Eno, X, Y, BX, BY) ' 変異境界条件表示(ただし変位 = 0)
  If X = 0 And BY = 0 Then Exit Sub ' 境界条件なしのとき終了
  C = RGB(0, 0, 0): X1 = X: Y1 = Y: DH1 = 6: DH2 = 10
  iflag = False ' 境界条件≠0のとき境界条件ありとするので
  If BX >= 0 Then ' 境界条件の正負により表示の向きを変えるものとする
    X2 = X - DH1: X3 = X + DH1: DY = DH2 ' BX>=0, BY>0のとき上向き(下端用)
    If BY < 0 Then DY = -DY ' BX>=0, BY<0のとき下向き(上端用)
    Y2 = Y + DY: Y3 = Y2
  End If
  If BX = 0 Then
    iflag = True
    XX1 = X2: XX2 = X3
    YY1 = Y2 + DY / 5: YY2 = YY1
  End If
  End Sub

```

(次ページに続く)

(前ページから続く)

```

Else
  Y2 = Y - DH1: Y3 = Y + DH1: DX = DH2 ' BX<0, BY>0 のとき右向き (左端用)
  If BY < 0 Then DX = -DX ' BX<0, BY>0 のとき左向き (右端用)
  X2 = X + DX: X3 = X2
  If BY = 0 Then
    iflag = True
    XX1 = X2 + DX / 5: XX2 = X3
    YY1 = Y2: YY2 = Y3
  End If
End If
If BX <> 0 And BY <> 0 Then
  C = RGB(0, 0, 0)
Else
  C = RGB(255, 255, 0)
End If
With ActiveSheet.Shapes.BuildFreeform(msoEditingAuto, X1, Y1)
  .AddNodes msoSegmentLine, msoEditingAuto, X2, Y2
  .AddNodes msoSegmentLine, msoEditingAuto, X3, Y3
  .AddNodes msoSegmentLine, msoEditingAuto, X1, Y1
  .ConvertToShape.Select
End With
With Selection
  .Name = "B_" & Eno
  .ShapeRange.Line.Visible = Visible
  If BX <> 0 And BY <> 0 Then
    .ShapeRange.Fill.ForeColor.RGB = RGB(255, 0, 0)
    .ShapeRange.Fill.Visible = msoTrue
    .ShapeRange.Fill.Solid = True
  Else
    .ShapeRange.Fill.ForeColor.RGB = RGB(255, 255, 0)
    .ShapeRange.Fill.Visible = msoTrue
  End If
  .ShapeRange.Line.Visible = msoTrue
  .ShapeRange.Line.ForeColor.SchemeColor = 8
  .ShapeRange.Line.Weight = 1
End With
If iflag Then
  ActiveSheet.Shapes("BL_" & Eno).Delete: Err = 0
  ActiveSheet.Shapes.AddLine(XX1, YY1, XX2, YY2).Select
  With Selection
    .Name = "BL_" & Eno
    .ShapeRange.Line.Visible = msoTrue
    .ShapeRange.Line.ForeColor.SchemeColor = 8
    .ShapeRange.Line.Weight = 1
  End With
End If
End Sub
Function extX(X) As Double
  extX = X * 100 + 200
End Function
Function extY(Y) As Double
  extY = -Y * 100 + 600
End Function

```

(次ページに続く)

(前ページから続く)

```

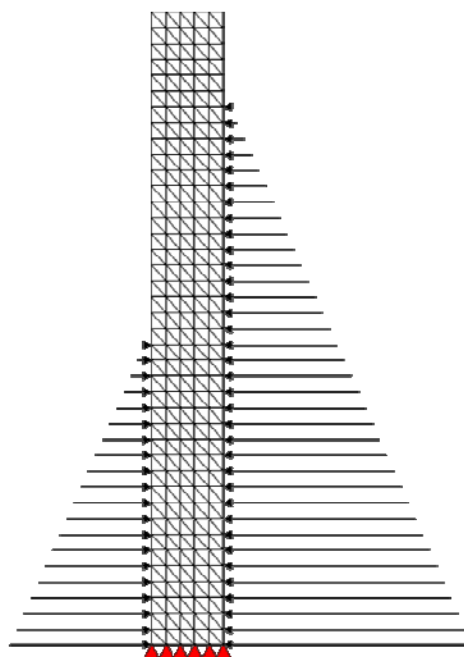
Sub drawElementData()
  i = 2
  Do While Worksheets("要素データ").Cells(i, 1) <> 0
    With Worksheets("節点データ")
      ii = Worksheets("要素データ").Cells(i, 2) + 1 ' 構成点 1
      XX = Val(.Cells(ii, 2)): YY = Val(.Cells(ii, 3))
      X1 = extX(XX): Y1 = extY(YY)
      ii = Worksheets("要素データ").Cells(i, 3) + 1 ' 構成点 2
      XX = Val(.Cells(ii, 2)): YY = Val(.Cells(ii, 3))
      X2 = extX(XX): Y2 = extY(YY)
      ii = Worksheets("要素データ").Cells(i, 4) + 1 ' 構成点 3
      XX = Val(.Cells(ii, 2)): YY = Val(.Cells(ii, 3))
      X3 = extX(XX): Y3 = extY(YY)
    End With
    With ActiveSheet.Shapes.BuildFreeform(msoEditingAuto, X1, Y1) ' 三角形
      .AddNodes msoSegmentLine, msoEditingAuto, X2, Y2
      .AddNodes msoSegmentLine, msoEditingAuto, X3, Y3
      .AddNodes msoSegmentLine, msoEditingAuto, X1, Y1
      .ConvertToShape.Select
    End With
    With Selection
      .Name = "Elm_" & Eno ' 図形名
      .ShapeRange.Fill.ForeColor.RGB = RGB(255, 255, 255) ' 塗潰し色
      .ShapeRange.Fill.Visible = msoTrue
      .ShapeRange.Line.Visible = msoTrue
      .ShapeRange.Line.ForeColor.SchemeColor = 8 ' 線の色
      .ShapeRange.Line.Weight = 1 ' 線の太さ
    End With
    i = i + 1
  Loop
End Sub
Sub 図_ボタン1_Click()
  drawElementData
  drawExternalForce
End Sub

```

■生成結果の図示と解析結果

生成した結果を上記プログラムで図示すると右図のようになり、生成結果が正しいことが分かります。

従来のプログラムが入ったファイルのシートに節点データと要素データを複写し、材料データを入力して実行します。



■実行結果

材料データは以下のように指定して実行します。

| | A | B | C | D |
|---|------|--------|-------|----|
| 1 | 材料番号 | ヤング率 | ポアソン比 | 板厚 |
| 2 | 1 | 800000 | 0.2 | 1 |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |

解析結果の表示オプションは以下のように指定して表示することにしました。なお、表示の際若干線が太いように感じましたので、線の太さ（Weight）を1に設定するようプログラムを変更しています。

| | A | B | C | D |
|---|--|---|-------|-----|
| 1 | | | 最小X | 100 |
| 2 | <input type="checkbox"/> 色付けをモノクロに | | 最大Y | 600 |
| 3 | <input checked="" type="checkbox"/> 節点番号表示なし | | スケール | 100 |
| 4 | | | 変位倍率 | 5 |
| 5 | <input checked="" type="checkbox"/> 要素番号表示なし | | 色付け対象 | 2 |

底部の左側に圧縮応力、右側に引張り応力が発生しています。このような場合、応力に関する評価も大事ですが、転倒に関する評価、転倒防止のための対策（下部の幅を大きくする等）を行う必要があります。

