APPENDIX

E

Computer Programs

Introduction

With the advent of computers and software packages, it is now possible to analyse any structure for any geometry and given loading conditions quickly. Though several standard software packages are available for the linear, non-linear, buckling or dynamic analysis of structures, very few design packages are available. Nonetheless, some analysis software packages do have some design routines, which are based on specific codes. Two computer programs written in Visual Basic are provided in this appendix, which can be run on any standard personal computer under the Windows environment. Visual Basic has been chosen because it uses an interactive approach to the development of computer programs. Visual Basic interprets the code as we enter it, catching and highlighting syntax or spelling errors. It also partly compiles the code as it is entered. While compiling the whole code, if the compiler finds an error, it is highlighted in the code. We can fix the error and continue compiling without having to start all over. Visual Basic also provides a variety of user interfaces in the form of text boxes, labels, radio button, pictures, etc. Visual Basic version 6.0 is used in this example, because it is simple and popular though VB.NET is the most recent version. More details about Visual Basic may be found in the Microsoft web site http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbcon98/html/vbconpart1visualbasicbasics.asp.

E.1 Design Programs

When writing design programs, it is much easier if the code of practice follows a logical sequence of requirements and checks. Also, the design stresses and factors should be given in closed-form mathematical expressions. Fortunately, the current version of IS: 800, satisfies the above requirements.

The software for the design programs provided here is also included in this CD. The guidelines for the installation of the Beam Design and BeamColumn Design software are as follows:

Beam Design Software

1. Installation of software
   (a) Insert the Design of Steel Structures CD that came with the book into the CD drive
   (b) Open the WinZip file Beam Program in the CD
   (c) Locate the file SETUP.exe and double click on this file
   (d) Follow the instructions
2. Execution of BeamDesign
   (a) Double click on BeamDesign icon either on desktop or navigating through the programs menu.
   (b) Two input screens will be displayed one after the other where you have to type in the data for the design
   (c) After typing all the data, click on 'Start the Design' button. The output will be saved under the working directory as BeamDesign.out.

Beam-Column Design Software

1. Installation of software
   (a) Insert the Design of Steel Structures CD that came with the book into the CD drive
   (b) Open the WinZip file Beam-Column Program in the CD
   (c) Locate the file SETUP.exe and double click on this file
   (d) Follow the instructions
2. Execution of Beam-Column Software
   (a) Double click on BeamDesign icon either on desktop or navigating through the programs menu.
   (b) Two input screens will be displayed one after the other where you have to type in the data for the design
   (c) After typing all the data, click on 'Start the Design' button. The output will be saved under the working directory as BeamColumn.out.

E.1.1 Beam Design Program

The simply supported steel beam design program, given in this section, designs beams of compact, semi-compact, and plastic sections. The beam arrangement is shown in Fig. E.1 and the flow chart is shown in Fig. E.2.

Only simple loading has been considered to reduce the length of the program (the user may modify it to suit his/her requirements). The notations used are the same as those given in IS: 800 (Section 8), and hence the program is easy to understand. The typical input screens of this program are given in Figs E.3 and E.4.

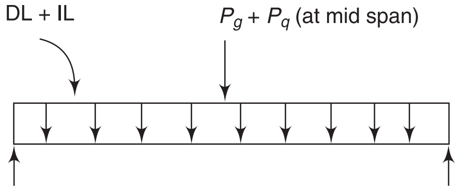

Fig. E.1 Beam and loading

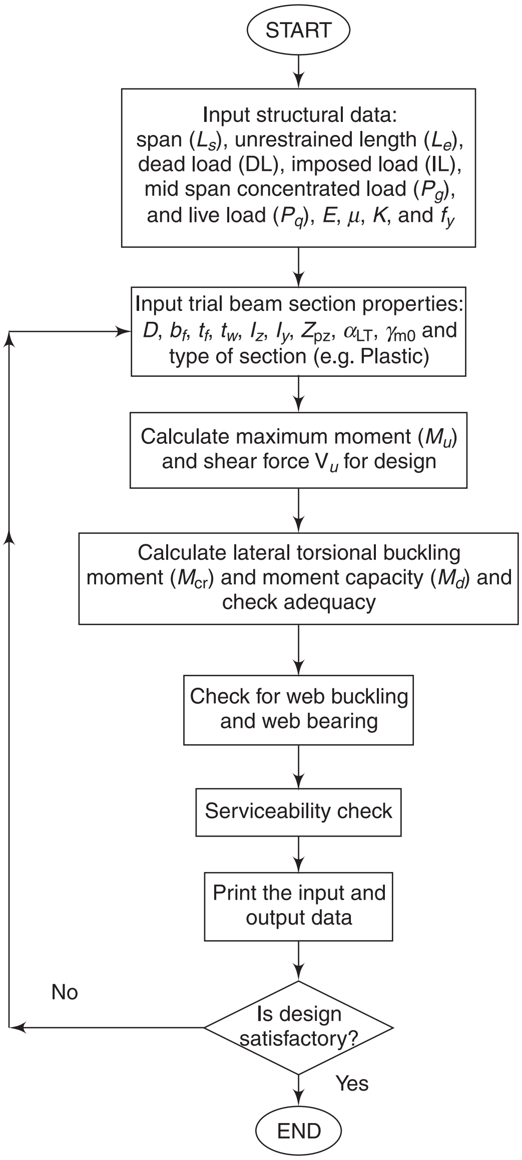

Fig. E.2 Flowchart of beam design program

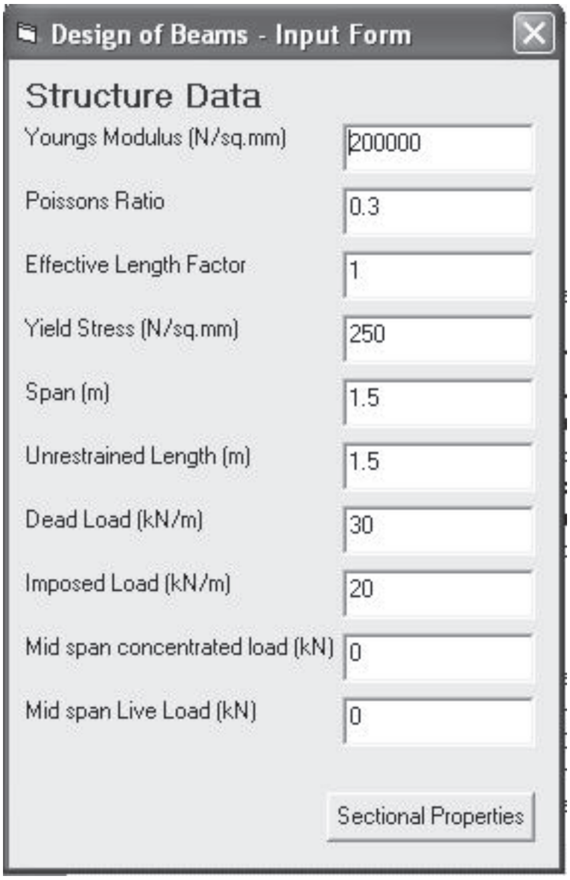

Fig. E.3 Input screen for structure data of beam design program

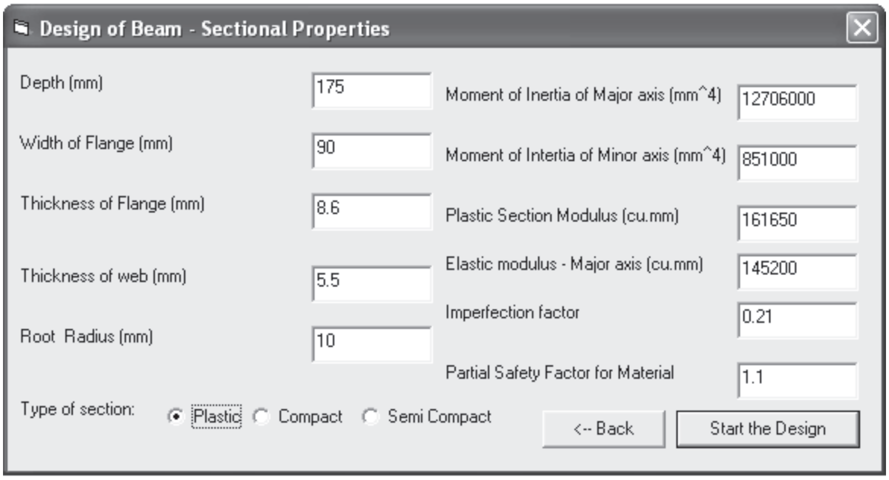

Fig. E.4 Input screen for sectional properties of beam design program

The following program is written in Visual Basic for beam design.

```
'Global Module

'Structure properties
Public E As Double
Public EMu As Double
Public K As Double
Public Fy As Double
Public LS As Double
Public Le As Double
Public DL As Double
Public IL As Double
Public PG As Double
Public PQ As Double

'Sectional Properties
Public D As Double
Public Bf As Double
Public tf As Double
Public tw As Double
Public R As Double
Public Iz As Double
Public Iy As Double
Public Zpz As Double
Public alphaLT As Double
Public GamaM As Double
Public Zz As Double

'intermediate values
Public Wu As Double
Public Vu As Double
Public Pu As Double
Public Mu As Double

Public Const PI As Double = 3.14159
Public sectionType As Integer

Public Const PLASTIC As Integer = 1
Public Const COMPACT As Integer = 2
Public Const SEMI_COMPACT As Integer = 3

Public blnLoaded As Boolean
Public FileName As String
```

```
'From Structure Data input form

Option Explicit
Private Sub cmdNext_Click()
'assign the data from input controls to global variables
Call UpdateStructureData
```

```
Unload Me
'Show the next input screen
frmSectionalProperties.Show
End Sub

Private Sub UpdateStructureData()
' transfer the data to global variables
E = Val(txtE)
EMu = Val(txtMu)
K = Val(txtK)
Fy = Val(txtFy)
LS = Val(txtSpan)
Le = Val(txtLE)
DL = Val(txtDeadLoad)
IL = Val(txtIL)
PG = Val(txtPG)
PQ = Val(txtPQ)

End Sub
— — — — — — — — — — — — — —

'From sectional properties input form, which also has design code
Option Explicit
Dim G As Double
Dim It As Double
Dim hf As Double
Dim Iw As Double
Dim Betaf As Double
Dim Mcr As Double
Dim LamdaLT As Double
Dim LamdaT1 As Double
Dim alphaLT As Double
Dim PhiLT As Double
Dim ChiLT As Double
Dim fbd As Double

Dim Md As Double
Dim Vd As Double
Dim Deltab As Double
Dim DeltaAll As Double

Dim b1 As Double
Dim n1 As Double
Dim Ab As Double
Dim I As Double
Dim A As Double
Dim rmin As Double
Dim Lamda As Double
Dim LamdaEff As Double
Dim Phi1 As Double
Dim Phi2 As Double
Dim fcd As Double
Dim SWB As Double
Dim n2 As Double
Dim Fw As Double
```

```
Dim blnSafe As Boolean

Private Sub cmdBack_Click()
  'Update the input controls with user entered Structure data
  Call LoadStructureData
  'Transfer user input sectional properties data to global variables
  Call UpdateSectionalProperties
  Unload Me

  frmBeamDesign1.Show

End Sub
Private Sub LoadStructureData()
  frmBeamDesign1.txtE = E
  frmBeamDesign1.txtMu = EMu
  frmBeamDesign1.txtK = Str(K)
  frmBeamDesign1.txtFy = Fy
  frmBeamDesign1.txtSpan = LS
  frmBeamDesign1.txtLE = Le
  frmBeamDesign1.txtDeadLoad = DL
  frmBeamDesign1.txtIL = IL
  frmBeamDesign1.txtPG = PG
  frmBeamDesign1.txtPQ = PQ
End Sub

Private Sub cmdDesign_Click()
'start the design process
'open the output file to print the results
  FileName = App.Path + "\BeamDesign.out"
  Open FileName For Output As 1
'pressume the design is safe
  blnSafe = True
  Call UpdateSectionalProperties
  Call Design
End Sub

Private Sub LoadSectionalProperties()
'if the data has been already transfered to the global variables
'then load them to input controls
'this check is necessary to sweap the default values with null
values
If blnLoaded Then
  frmSectionalProperties.txtDepth = Str(D)
  frmSectionalProperties.txtWidthOfFlange = Str(Bf)
  frmSectionalProperties.txtThickOfFlange = Str(tf)
  frmSectionalProperties.txtThickWeb = Str(tw)
  frmSectionalProperties.txtR = Str(R)
  frmSectionalProperties.txtMIMajor = Str(Iz)
  frmSectionalProperties.txtMIMinor = Str(Iy)
  frmSectionalProperties.txtPlastic = Str(Zpz)
  frmSectionalProperties.txtImpFactor = Str(alphaLT)
  frmSectionalProperties.txtSafetyFactor = Str(GamaM)
```

```
  frmSectionalProperties.txtZZ = Str(Zz)
  If sectionType = PLASTIC Then
    optPlastic = True
  ElseIf sectionType = COMPACT Then
    optCompact = True
  ElseIf sectionType = SEMI_COMPACT Then
    optSemiCompact = True
  End If
End If
End Sub

Private Sub UpdateSectionalProperties()
  D = Val(txtDepth)
  Bf = Val(txtWidthOfFlange)
  tf = Val(txtThickOfFlange)
  tw = Val(txtThickWeb)
  R = Val(txtR)
  Iz = Val(txtMIMajor)
  Iy = Val(txtMIMinor)
  Zpz = Val(txtPlastic)
  alphaLT = Val(txtImpFactor)
  GamaM = Val(txtSafetyFactor)
  Zz = Val(txtZZ)
  If optPlastic Then
    sectionType = PLASTIC
  ElseIf optCompact Then
    sectionType = COMPACT
  ElseIf optSemiCompact Then
    sectionType = SEMI_COMPACT
  End If
End Sub

Private Sub Form_Load()
  Call LoadSectionalProperties
  blnLoaded = True
End Sub

Private Sub Design()
  ‘ Print the input data into the output file
  Call PrintInput
  Call CalculateStdValues
  Call CalculateLTBucklingMomentCapacity
  ‘ continue the process only if so far the design is safe
  If blnSafe Then Call CalculateShearCapacity
  If blnSafe Then Call CalculateDeflection
  If blnSafe Then Call CheckForWebBuckling
  If blnSafe Then Call CheckForWebBearing
  Close #1
  If blnSafe Then End
End Sub
```

```
Private Sub CalculateStdValues()
  ' calculate maximum Mu and Vu for design
  Dim strMsg As String

  Wu = 1.5 * (DL + IL)
  Pu = 1.5 * (PG + PQ)
  Mu = Wu * LS * LS / 8# + Pu * LS / 4#
  Vu = (Wu * LS + Pu) / 2

  Print #1, Tab(5); "Output"
  Print #1, Tab(5); "----"

  Print #1, Tab(5); "Maximum bending moment for design (kNm)";
  Tab(60); Format(Mu, "#######0.0##")
  Print #1, Tab(5); "Maximum shear force for design (kN)"; Tab(60);
  Format(Vu, "######0.0##")
End Sub

Private Sub CalculateLTBucklingMomentCapacity()
Dim Betab As Double
Dim strMsg As String

  If sectionType = SEMI_COMPACT Then
    Betab = Zz / Zpz
  Else
    Betab = 1
  End If

  Betaf = 0.5
  G = E / (2 * (1 + EMu))
  hf = D - tf
  It = (2 * Bf * tf ^ 3 + hf * tw ^ 3) / 3
  Le = K * Le
  Iw = (1 - Betaf) * Betaf * Iy * hf ^ 2
  Mcr = (((PI ^ 2 * E * Iy) / (Le * 1000) ^ 2) * (G * It + (PI ^
  2 * E * Iw) / (Le * 1000) ^ 2)) ^ 0.5
  LamdaLT = (Betab * Zpz * Fy / Mcr) ^ 0.5
  LamdaLT1 = (1.2 * Zz * Fy / Mcr) ^ 0.5
  If LamdaLT > LamdaLT1 then
     LamdaLT = LamdaLT1
  End If
  alphaLT = 0.21
  PhiLT = 0.5 * (1 + alphaLT * (LamdaLT - 0.2) + LamdaLT ^ 2)
  ChiLT = 1# / (PhiLT + (PhiLT ^ 2 - LamdaLT ^ 2) ^ 0.5)
  If (ChiLT > 1#) Then
     ChiLT = 1#
  End If
  fbd = ChiLT * Fy / GamaM


  Md = Betab * Zpz * fbd / 1000000#
  strMsg = "Torsional constant " + Str(It) + " mm^4" + vbCrLf
  strMsg = strMsg + "Warping constant " + Str(Iw) + " mm^6" +
  vbCrLf
  strMsg = strMsg + "Moment capacity of the section " + Str(Md) +
  " kNm" + vbCrLf
```

```
  If Md < Mu Then
    MsgBox strMsg + "Moment capacity is less than " + Str(Mu) + "
    Section is unsafe. Revise the section", , "Unsafe section"
    Unload Me
    blnSafe = False
    frmBeamDesign1.Show
    Exit Sub
  End If
  Print #1, Tab(5); "Torsional constant (mm^4) "; Tab(60);
  Format(It, "#######0.0##")
  Print #1, Tab(5); "Warping constant (mm^6)"; Tab(60); Format(Iw,
  "#######0.0##")
  Print #1, Tab(5); "Moment capacity of the section (kNm)"; Tab(60);
  Format(Md, "#######0.0##")
  Print #1, Tab(5); "Moment capacity of Trial section is adequate"
  Print #1, ""

End Sub

Private Sub CalculateShearCapacity()
  Dim strMsg As String

  Vd = ((Fy / (GamaM * Sqr(3))) * D * tw) / 1000
  strMsg = "Shear capacity = " + Str(0.6 * Vd) + " kN" + vbCrLf +
  "Shear force = " + Str(Vu) + " kN" + vbCrLf
  If 0.6 * Vd < Vu Then
    MsgBox strMsg + "Shear capacity is unsafe. Revise the section"
    Unload Me
    blnSafe = False
    frmBeamDesign1.Show
    Exit Sub
  Else
    Print #1, Tab(5); "Shear capacity (kN) "; Tab(60); Format(0.6
    * Vd, "#######0.0##")
    Print #1, Tab(5); "Shear force (kN) "; Tab(60); Format(Vu,
    "#######0.0##")
    Print #1, Tab(5); "Shear capacity of Trial section is adequate"
  End If

End Sub
Private Sub CalculateDeflection()

  Dim strMsg As String

  Deltab = (5 * (DL + IL) * (LS * 1000) ^ 4 / 384 + (PG + PQ) * LS
  ^ 3 / 48) / (E * Iz)
  DeltaAll = LS * 1000 / 300

  strMsg = "Delta allowable = " + Str(DeltaAll) + " mm" + vbCrLf
  + "Delta actual = " + Str(Deltab) + " mm" + vbCrLf
  If Deltab < DeltaAll Then
    'MsgBox strMsg + "Trial section is adequate for deflection
    check"
```

```
    Print #1, Tab(5); "Delta allowable (mm) "; Tab(60);
    Format(DeltaAll, "#######0.0##")
    Print #1, Tab(5); "Delta actual (mm) "; Tab(60); Format(Deltab,
    "#######0.0##")
    Print #1, Tab(5); "Trial section is adequate for deflection
    check"
    Print #1, ""
  Else
    MsgBox strMsg + "Revise the section for deflection consideration"
    Unload Me
    blnSafe = False
    frmBeamDesign1.Show
    Exit Sub
  End If
End Sub
Private Sub CheckForWebBuckling()
  Dim strMsg As String

  b1 = (Bf - tw) / 2
  n1 = D / 2
  Ab = (b1 + n1) * tw
  I = b1 * tw ^ 3 / 12
  A = b1 * tw
  rmin = Sqr(I / A)
  Lamda = 0.7 * (D - 2 * (tf + R)) / rmin
  LamdaEff = (Fy * Lamda ^ 2 / (PI ^ 2 * E)) ^ 0.5
  Phi1 = 0.5 * (1 + 0.49 * (LamdaEff - 0.2) + LamdaEff ^ 2)
  Phi2 = Phi1 + (Phi1 ^ 2 - LamdaEff ^ 2) ^ 0.5
  fcd = Fy / (Phi2 * GamaM)
  SWB = fcd * Ab / 1000

  strMsg = "Strength against web buckling = " + Str(SWB) + " kN"
  + vbCrLf + "Shear force = " + Str(Vu) + " kN"
  If SWB > Vu Then
    'MsgBox strMsg + "Safe against web buckling"
    Print #1, Tab(5); "Strength against web buckling (kN) ";
    Tab(60); Format(SWB, "#######0.0##")
    Print #1, Tab(5); "Shear force (kN) "; Tab(60); Format(Vu,
    "#######0.0##")
    Print #1, Tab(5); "Safe against web buckling"
  Else
    MsgBox strMsg + "Revise the section for web buckling"
    Unload Me
    blnSafe = False
    frmBeamDesign1.Show
    Exit Sub
  End If

End Sub
Private Sub CheckForWebBearing()
  Dim strMsg As String

  n2 = 2.5 * (tf + R)
  Fw = (b1 + n2) * tw * Fy / (GamaM * 1000)
```

```
  strMsg = "Strength against web bearing = " + Str(Fw) + " kN" +
  vbCrLf + "Shear force = " + Str(Vu) + " kN" + vbCrLf
  If Fw > Vu Then
    'MsgBox strMsg + "Safe against web bearing"
    MsgBox "Design is safe and " + vbCrLf + "The output is saved
    as BeamDesign.out in the working folder"
    Print #1, Tab(5); "Strength against web bearing (kN) ";
    Tab(60); Format(Fw, "#######0.0##")
    Print #1, Tab(5); "Shear force (kN) "; Tab(60); Format(Vu,
    "#######0.0##")
    Print #1, Tab(5); "Safe against web bearing"
    Print #1, ""
  Else
    MsgBox strMsg + "Revise the section for web bearing"
    Unload Me
    frmBeamDesign1.Show
    Exit Sub
  End If
End Sub

Private Sub optCompact_Click()
  If optCompact Then
  sectionType = COMPACT
  End If

End Sub

Private Sub optPlastic_Click()
  If optPlastic Then
    sectionType = PLASTIC
  End If
End Sub

Private Sub optSemiCompact_Click()
  If optSemiCompact Then
    sectionType = SEMI_COMPACT
  End If
End Sub
Private Sub PrintInput()

  Print #1, Tab(10); "Beam Design"
  Print #1, Tab(10); "------"
  Print #1, ""
  Print #1, Tab(5); "Structure Data"
  Print #1, Tab(5); "----------"

  Print #1, Tab(5); "Youngs Modulus (N/sq.mm) "; Tab(60); Format(E,
  "#######0.0##")
  Print #1, Tab(5); "Poisson's Ratio "; Tab(60); Format(EMu,
  "#######0.0##")
  Print #1, Tab(5); "Eff.Length factor "; Tab(60); Format(K,
  "#######0.0##")
```

```
  Print #1, Tab(5); "Yield Stress (N/sq.mm) "; Tab(60); Format(Fy,
  "#######0.0##")
  Print #1, Tab(5); "Span (m) "; Tab(60); Format(LS, "#######0.0##")
  Print #1, Tab(5); "Unrestrained length (m) "; Tab(60); Format(Le,
  "#######0.0##")
  Print #1, Tab(5); "Dead load (kN/m) "; Tab(60); Format(DL,
  "#######0.0##")
  Print #1, Tab(5); "Imposed load (kN/m) "; Tab(60); Format(IL,
  "#######0.0##")
  Print #1, Tab(5); "Mid span concentrated load (kN) "; Tab(60);
  Format(PG, "#######0.0##")
  Print #1, Tab(5); "Mid span live load (kN) "; Tab(60); Format(PQ,
  "#######0.0##")

  Print #1, Tab(5); ""
  Print #1, Tab(5); "Sectional Properties"
  Print #1, Tab(5); "------------"
  Print #1, Tab(5); "Depth (mm)"; Tab(60); Format(D, "#######0.0##")
  Print #1, Tab(5); "Width of flange (mm)"; Tab(60); Format(Bf,
  "#######0.0##")
  Print #1, Tab(5); "Thickness of flange (mm)"; Tab(60); Format(tf,
  "#######0.0##")
  Print #1, Tab(5); "Thickness of web (mm)"; Tab(60); Format(tw,
  "#######0.0##")
  Print #1, Tab(5); "Root radius (mm)"; Tab(60); Format(R,
  "#######0.0##")
  Print #1, Tab(5); "Moment of inertia of Major axis (mm^4)";
  Tab(60); Format(Iz, "#######0.0##")
  Print #1, Tab(5); "Moment of inertia of Minor axis (mm^4)";
  Tab(60); Format(Iy, "#######0.0##")
  Print #1, Tab(5); "Palstic section modulus (cu.mm)"; Tab(60);
  Format(Zpz, "#######0.0##")
  Print #1, Tab(5); "Elastic modulus about major axis (cu.mm)";
  Tab(60); Format(Zz, "#######0.0##")
  Print #1, Tab(5); "Imperfection factor "; Tab(60); Format(alphaLT,
  "#######0.0##")
  Print #1, Tab(5); "Partial safety factor for material "; Tab(60);
  Format(GamaM, "#######0.0##")
  Print #1, ""
End Sub
```

In order to show the use of this program, the data given in Example 10.9 was used and the resulting output is as follows:

```
      Beam Design
      ----------

Structure Data
-------------
Youngs Modulus <N/sq. mm>                        200000.0
Poisson's Ratio                                  0.3
Eff.Length factor                                1.0
```

```
Yield Stress <N/sq.mm>                                   250.0
Span <m>                                                 1.5
Unrestrained length <m>                                  1.5
Dead load <kN/m>                                         30.0
Imposed load <kN/m>                                      20.0
Mid span concentrated load <kN>                          0.0
Mid span live load <kN>                                  0.0

Sectional Properties
-------------------
Depth <mm>                                               175.0
Width of flange <mm>                                     90.0
Thickness of flange <mm>                                 8.6
Thickness of web <mm>                                    5.5
Root radius <mm>                                         10.0
Moment of inertia of Major axis <mm^4>                   12706000.0
Moment of inertia of Minor axis <mm^4>                   851000.0
Palstic section modulus <cu.mm>                          161650.0
Elastic modulus about major axis <cu.mm>                 145200.0
Imperfection factor                                      0.21
Partial safety factor for material                       1.1

Output
-----
Maximum bending moment for design <kNm>                  21.094
Maximum shear force for design <kN>                      56.25
Torsional constant <mm^4>                                47391.627
Warping constant <mm^6>                                  5890826240.0
Moment capacity of the section <kNm>                     31.044
Moment capacity of Trial section is adequate

Shear capacity <kN>                                      75.777
Shear force <kN>                                         56.25
Shear capacity of Trial section is adequate
Delta allowable <mm>                                     5.0
Delta actual <mm>                                        1.297
Trial section is adequate for deflection check

Strength against web buckling<kN>                        119.165
Shear force <kN>                                         56.25
Safe against web buckling
Strength against web bearing <kN>                        110.938
Shear force <kN>                                         56.25
Safe against web bearing
```

E.1.2 Beam-Column Design Program

The design of a beam-column, using a hot-rolled I-section, is considered in the program given in this section. In all cases, axial compression is taken as positive. The sign convention for the moments at the top and bottom ends of the column is positive for moments applied clockwise and negative for moments applied anti-clockwise.

Hot rolled I- or H- plastic, compact, or semi-compact sections can be designed using this program. There are two sets of input data in this program, as in the case of the beam program. The first is the structural data consisting of the following: factored axial force (N), factored bending moment at the top and bottom about the major axis (M_{Z1}, M_{Z2}), factored bending moment at the top and bottom about the minor axis (M_{Y1}, M_{Y2}), length in major and minor axis (L_z, L_y), effective length factors (K_z, K_y), Young's modulus (E), and Poisson's ratio (μ). The second set of data is the member data consisting of the following: depth of the section (D), Area (A), breadth of flange (b_f), thickness of flange and web (t_f and t_w), root radius (R), radius of gyration in the major and minor axis (r_z, r_y) moment of inertia about minor axis (I_y), elastic modulus about major and minor axis (Z_z and Z_y), yield strength of material (f_y) and partial factor of safety of material ($\gamma_{m0,}$) and type of section (plastic, compact, or semi-compact).

Based on the preceding data, the plastic modulus about the major and minor axis (Z_{pz}, Z_{py}) and the warping and torsional rigidity (I_w, I_t) are calculated. The member buckling resistance in compression about the major axis and minor axis (P_{dz} and P_{dy}) are calculated based on clause 7.1.2 of the code. The member buckling resistance in bending is computed based on clause 8.2.2, about both the axes (M_{dz} and M_{dy}).

Using these values, the strength of the cross section is checked based on the interaction equations given in clause 9.3.1 of the code. Finally, the overall member strength is checked as per the interaction equation given in clause 9.3.2 of the code. If any one of the checks is not satisfied, the user has to change the trial section properties and run the program again. The program written in Visual Basic and typical input screens are given below:

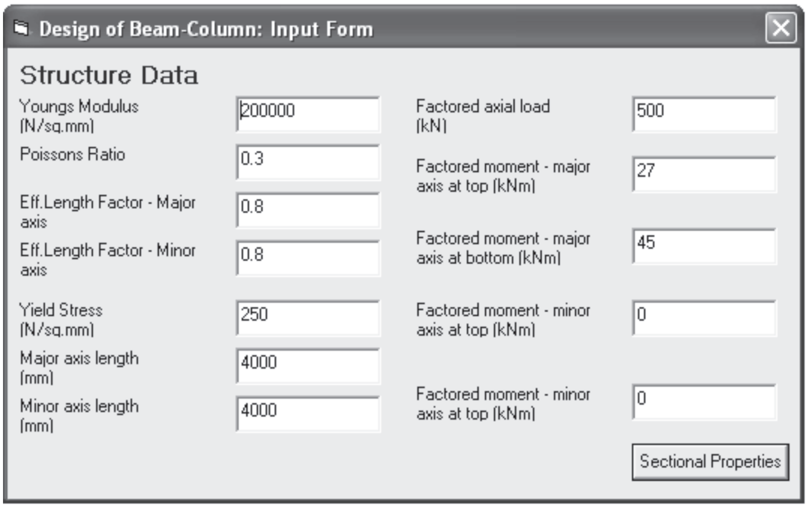

Fig. E.5 Input screen for structure data of beam-column program

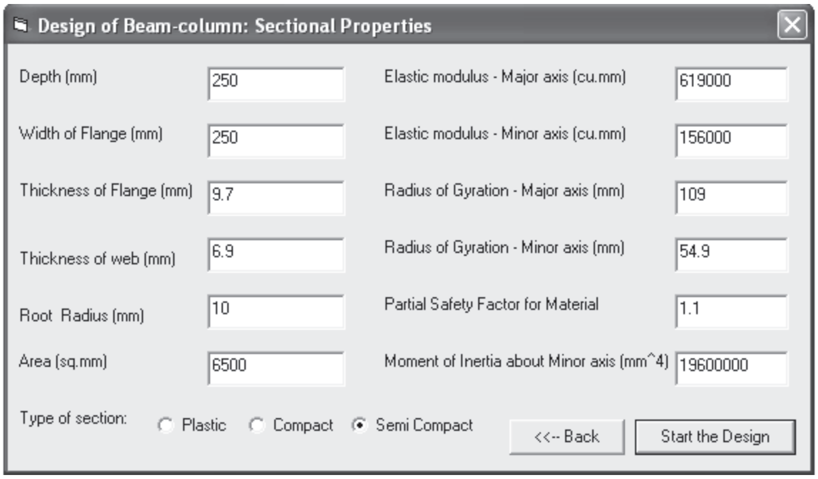

Fig. E.6 Input screen for sectional properties of Beam-Column program

```
Option Explicit
'Structure properties
Public E As Double
Public EMu As Double
Public Fy As Double
Public Lz As Double
Public Ly As Double
Public Kz As Double
Public Ky As Double
Public N As Double
Public Mz1 As Double
Public Mz2 As Double
Public My1 As Double
Public My2 As Double

'Sectional Properties
Public D As Double
Public Bf As Double
Public tf As Double
Public tw As Double
Public R As Double
Public rz As Double
Public ry As Double
Public Zz As Double
Public Zy As Double
Public Zpz As Double
Public Zpy As Double
Public GamaM As Double
Public A As Double
Public Iy As Double
Public Pd As Double

Public sectionType As Integer
```

```
Public Const PLASTIC As Integer = 1
Public Const COMPACT As Integer = 2
Public Const SEMI_COMPACT As Integer = 3
Public Const PI As Double = 3.14159

Public blnLoaded As Boolean
Public FileName As String
```

```
Option Explicit
Private Sub cmdNext_Click()
  Call UpdateStructureData
  Unload Me
  frmSectionalProperties.Show
End Sub

Private Sub UpdateStructureData()
  E = Val(txtE)
  EMu = Val(txtMu)
  Kz = Val(txtKz)
  Ky = Val(txtKy)
  Fy = Val(txtFy)
  Ly = Val(txtLY)
  Lz = Val(txtLZ)
  My1 = Val(txtMY1)
  My2 = Val(txtMY2)
  Mz1 = Val(txtMZ1)
  Mz2 = Val(txtMZ2)
  N = Val(txtN)
End Sub
```

```
Option Explicit
Dim G As Double
Dim It As Double
Dim hf As Double
Dim Iw As Double
Dim Betaf As Double

Dim Mcr As Double
Dim LambdaLT As Double
Dim alphaLT As Double
Dim PhiLT As Double
Dim ChiLT As Double
Dim fbd As Double

Dim Md As Double
Dim Vd As Double
Dim Deltab As Double
Dim DeltaAll As Double

Dim b1 As Double
Dim n1 As Double
Dim Ab As Double
```

```
Dim I As Double
Dim A As Double
Dim rmin As Double
Dim Lamda As Double
Dim LamdaEff As Double
Dim Phi1 As Double
Dim Phi2 As Double
Dim fcd As Double
Dim SWB As Double
Dim n2 As Double
Dim Fw As Double
Dim Mz As Double
Dim Mzm As Double
Dim My As Double
Dim Mym As Double
Dim Nd As Double
Dim Mdz, Mdz1 As Double
Dim Mdy As Double
Dim StressRatio As Double
Dim StressRatio1 As Double
Dim Lrz As Double
Dim Lry As Double
Dim AlphaZ As Double
Dim AlphaY As Double
Dim Betab As Double
Dim LambdaZ As Double
Dim LambdaY As Double
Dim Pdz As Double
Dim Pdy As Double
Dim blnSafe As Boolean

Private Sub cmdBack_Click()
  Call UpdateSectionalProperties
  Call LoadStructureData
  Unload Me
  frmBeamColumnDesign1.Show
End Sub
Private Sub LoadStructureData()
  frmBeamColumnDesign1.txtE = E
  frmBeamColumnDesign1.txtMu = EMu
  frmBeamColumnDesign1.txtKz = Str(Kz)
  frmBeamColumnDesign1.txtKy = Str(Ky)
  frmBeamColumnDesign1.txtFy = Fy
  frmBeamColumnDesign1.txtLY = Ly
  frmBeamColumnDesign1.txtLZ = Lz
  frmBeamColumnDesign1.txtMY1 = My1
  frmBeamColumnDesign1.txtMY2 = My2
  frmBeamColumnDesign1.txtMZ1 = Mz1
  frmBeamColumnDesign1.txtMZ2 = Mz2
  frmBeamColumnDesign1.txtN = N
End Sub
```

```
Private Sub cmdDesign_Click()
  FileName = App.Path + "\BeamColumn.out"
  Open FileName For Output As 1
  blnSafe = True
  Call UpdateSectionalProperties
  Call Design
End Sub

Private Sub LoadSectionalProperties()
  If blnLoaded Then
    frmSectionalProperties.txtDepth = Str(D)
    frmSectionalProperties.txtWidthOfFlange = Str(Bf)
    frmSectionalProperties.txtThickOfFlange = Str(tf)
    frmSectionalProperties.txtThickWeb = Str(tw)
    frmSectionalProperties.txtR = Str(R)
    frmSectionalProperties.txtRZ = Str(rz)
    frmSectionalProperties.txtRY = Str(ry)
    frmSectionalProperties.txtA = Str(A)
    frmSectionalProperties.txtZZ = Str(Zz)
    frmSectionalProperties.txtZY = Str(Zy)
    frmSectionalProperties.txtSafetyFactor = Str(GamaM)
    frmSectionalProperties.txtIy = Str(Iy)
    If sectionType = PLASTIC Then
      optPlastic = True
    ElseIf sectionType = COMPACT Then
      optCompact = True
    ElseIf sectionType = SEMI_COMPACT Then
      optSemiCompact = True
    End If
  End If
End Sub

Private Sub UpdateSectionalProperties()
  D = Val(txtDepth)
  Bf = Val(txtWidthOfFlange)
  tf = Val(txtThickOfFlange)
  tw = Val(txtThickWeb)
  R = Val(txtR)
  rz = Val(txtRZ)
  ry = Val(txtRY)
  A = Val(txtA)
  Zz = Val(txtZZ)
  Zy = Val(txtZY)
  GamaM = Val(txtSafetyFactor)
  Iy = Val(txtIy)
  If optPlastic Then
    sectionType = PLASTIC
  ElseIf optCompact Then
    sectionType = COMPACT
  ElseIf optSemiCompact Then
    sectionType = SEMI_COMPACT
  End If
End Sub
```

```
Private Sub Form_Load()
  Call LoadSectionalProperties
  blnLoaded = True
End Sub

Private Sub Design()
  blnSafe = True
  Call PrintInput
  Call ComputeInteractionEqnLocal
  If blnSafe Then Call DesignCompressionResisMajor
  If blnSafe Then Call DesignCompressionResisMinor
  If blnSafe Then Call DesignBucklingResisMajor
  If blnSafe Then Call DesignResOfCrossSection
  If blnSafe Then Call DesignShearResistance
  If blnSafe Then Call DesignComBendingAxialForce
  Close #1
  If blnSafe Then End
End Sub
Private Sub PrintInput()

  Print #1, Tab(10); "Beam-Column Design"
  Print #1, Tab(10); "----------"
  Print #1, ""
  Print #1, Tab(5); "Structure Data"
  Print #1, Tab(5); "-------"
  Print #1, Tab(5); "Youngs Modulus (N/sq.mm)  "; Tab(60); Format(E,
  "#######0.0##")
  Print #1, Tab(5); "Poisson's Ratio "; Tab(60); Format(EMu,
  "#######0.0##")
  Print #1, Tab(5); "Yield Stress (N/sq.mm) "; Tab(60); Format(Fy,
  "#######0.0##")
  Print #1, Tab(5); "Major axis Length (mm) "; Tab(60); Format(Lz,
  "#######0.0##")
  Print #1, Tab(5); "Minor axis Length (mm) "; Tab(60); Format(Ly,
  "#######0.0##")
  Print #1, Tab(5); "Eff.Length factor - Major axis "; Tab(60);
  Format(Kz, "#######0.0##")
  Print #1, Tab(5); "Eff.Length factor - Minor axis "; Tab(60);
  Format(Ky, "#######0.0##")
  Print #1, Tab(5); "Factored axial load (kN) "; Tab(60); Format(N,
  "#######0.0##")
  Print #1, Tab(5); "Major axis factored moment at top (kNm)";
  Tab(60); Format(Mz1, "#######0.0##")
  Print #1, Tab(5); "Major axis factored moment at bottom (kNm)";
  Tab(60); Format(Mz2, "#######0.0##")
  Print #1, Tab(5); "Minor axis factored moment at top (kNm)";
  Tab(60); Format(My1, "#######0.0##")
  Print #1, Tab(5); "Minor axis factored moment at bottom (kNm)";
  Tab(60); Format(My2, "#######0.0##")

  Print #1, Tab(5); ""
  Print #1, Tab(5); "Sectional Properties"
```

```
    Print #1, Tab(5); "----------"
    Print #1, Tab(5); "Depth (mm)"; Tab(60); Format(D, "#######0.0##")
    Print #1, Tab(5); "Width of flange (mm)"; Tab(60); Format(Bf,
    "#######0.0##")
    Print #1, Tab(5); "Thickness of flange (mm)"; Tab(60); Format(tf,
    "#######0.0##")
    Print #1, Tab(5); "Thickness of web (mm)"; Tab(60); Format(tw,
    "#######0.0##")
    Print #1, Tab(5); "Root radius (mm)"; Tab(60); Format(R,
    "#######0.0##")
    Print #1, Tab(5); "Radius of Gyration major axis (mm)"; Tab(60);
    Format(rz, "#######0.0##")
    Print #1, Tab(5); "Radius of Gyration minor axis (mm)"; Tab(60);
    Format(ry, "#######0.0##")
    Print #1, Tab(5); "Elastic modulus about major axis (cu.mm)";
    Tab(60); Format(Zz, "#######0.0##")
    Print #1, Tab(5); "Elastic modulus about minor axis (cu.mm)";
    Tab(60); Format(Zy, "#######0.0##")
    Print #1, Tab(5); "Partial safety factor for material "; Tab(60);
    Format(GamaM, "#######0.0##")
    Print #1, Tab(5); "Area (sq.mm)"; Tab(60); Format(A,
    "#######0.0##")
    Print #1, Tab(5); "Moment of intertia about minor axis (mm^4)";
    Tab(60); Format(Iy, "#######0.0##")
    Print #1, ""
End Sub
Private Sub ComputeInteractionEqnLocal()
    G = E / (2 * (1 + EMu))
    If (Abs(Mz1) > Abs(Mz2)) Then
       Mz = Mz1
       Mzm = Mz2
    Else
       Mz = Mz2
       Mzm = Mz1
    End If

    If (Abs(My1) > Abs(My2)) Then
       My = My1
       Mym = My2
    Else
       My = My2
       Mym = My1
    End If
    'Properties of the  Cross-section
    Zpz = 2 * Bf * tf * (D - tf) / 2 + tw * (D - 2 * tf) ^ 2 / 4
    Zpy = 2 * tf * Bf * Bf / 4 + (D - 2 * tf) * tw * tw / 4
    It = (2 * tf ^ 3 * Bf + (D - tf) * tw ^ 3) / 3
    Iw = (D - tf) ^ 2 * Iy / 4
    Nd = A * Fy / (GamaM * 1000#)
    ' Minor Axis buckling resistance in bending
```

```
  If sectionType = SEMI_COMPACT Then
    Mdy = Zy * Fy / (GamaM * 1000000#)
  Else
    Mdy = Zpy * Fy / (GamaM * 1000000#)
  End If
  Dim strMsg As String
  strMsg = "Plastic modulus about Major axis " + Str(Zpz) + "
  cu.mm" + vbCrLf
  strMsg = strMsg + "Plastic modulus about Minor axis " + Str(Zpy)
  + " cu.mm" + vbCrLf
  strMsg = strMsg + "Torsional constant" + Str(It) + " mm^4" +
  vbCrLf
  strMsg = strMsg + "Warping constant " + Str(Iw) + " mm^6"
  'MsgBox strMsg
  'MsgBox "Design Resistance in Bending about Minor Axis" + Str(Mdy)
  + " kNm" + vbCrLf + "Applied Moment about Minor Axis " + Str(My)
  + " kNm"
  If sectionType = SEMI_COMPACT Then
    Betab = Zz / Zpz
  Else
    Betab = 1
  End If
  Mdz = Betab * Zpz * Fy / (GamaM * 1000000#)
  Print #1, Tab(5); "Output"
  Print #1, Tab(5); "-----"

  Print #1, Tab(5); "Plastic modulus about Major axis (cu.mm) ";
  Tab(60); Format(Zpz, "#######0.0##")
  Print #1, Tab(5); "Plastic modulus about Minor axis (cu.mm)";
  Tab(60); Format(Zpy, "#######0.0##")
  Print #1, Tab(5); "Torsional constant (mm^4)"; Tab(60); Format(It,
  "#######0.0##")
  Print #1, Tab(5); "Warping constant (mm^6)"; Tab(60); Format(Iw,
  "#######0.0##")
  Print #1, Tab(5); "Design Resistance in Bending about Minor
  Axis (kNm)"; Tab(60); Format(Mdy, "#######0.0##")
  Print #1, Tab(5); "Applied Moment about Minor Axis (kNm) ";
  Tab(60); Format(My, "#######0.0##")
  Print #1, ""
End Sub
Private Sub DesignResOfCrossSection()
Dim Alpha1, Alpha2, Nr, Mndz, Mndy As Double

'Local Capacity Check
  If sectionType = SEMI_COMPACT Then
    StressRatio = (N / Nd) + (Mz / Mdz) + (My / Mdy)
  Else
    Nr = N / Nd
    Mndz = 1.11 * Mdz * (1 - Nr)
    If Mndz > Mdz Then
      Mndz = Mdz
    End If
    If Nr <= 0.2 Then
      Mndy = Mdy
```

```
    Else
      Mndy = 1.56 * Mdy * (1 - Nr) * (Nr + 0.6)
    End If
    Alpha1 = 5 * Nr
    If Alpha1 < 1 Then
      Alpha1 = 1
    End If
    Alpha2 = 2
    ' section strength interaction equation
    StressRatio = (My / Mndy) ^ Alpha1 + (Mz / Mndz) ^ Alpha2

  End If
  If StressRatio < 1# Then
    'MsgBox "Interaction equation value = " + Str(StressRatio) +
    vbCrLf + " Section is safe"
    Print #1, Tab(5); "Interaction equation (section strength
    value "; Tab(60); Format(StressRatio, "#######0.0##"); "
    Section is safe"
    Print #1, ""
  Else
    MsgBox "Interaction equation value = " + Str(StressRatio) +
    vbCrLf + "Section unsafe due to member capacity check " +
    vbCrLf + "Revise the section"
    blnSafe = False
    Unload Me
    frmBeamColumnDesign1.Show
  End If
End Sub
Private Sub DesignCompressionResisMajor()
Dim Fcrz, Fcdz, Phiz, Phiz1 As Double
'Compression resistance - Major Axis
  Lz = Kz * Lz
  Ly = Ky * Ly
  Lrz = Lz / rz
  Lry = Ly / ry
  If (D / Bf) > 1.2 And tf <= 40 Then
    AlphaZ = 0.21
    AlphaY = 0.34
  ElseIf (D / Bf) > 1.2 And tf < 100 Then
    AlphaZ = 0.34
    AlphaY = 0.49
  ElseIf (D / Bf) <= 1.2 And tf <= 100 Then
    AlphaZ = 0.34
    AlphaY = 0.49
  Else
    AlphaZ = 0.76
    AlphaY = 0.76
  End If

  Fcrz = PI * PI * E / (Lrz * Lrz)
  LambdaZ = (Fy / Fcrz) ^ 0.5
  Phiz = 0.5 * (1 + AlphaZ * (LambdaZ - 0.2) + LambdaZ ^ 2)
```

```
  Phiz1 = Phiz + (Phiz ^ 2 - LambdaZ ^ 2) ^ 0.5
  Fcdz = Fy / (GamaM * Phiz1)
  If Fcdz > (Fy / GamaM) Then
    Fcdz = Fy / GamaM
  End If
  Pdz = Fcdz * A / 1000#
  Dim strMsg As String
  strMsg = "Compression resistance about Major Axis " + Str(Pdz)
  + " kN" + vbCrLf
  If Pdz > N Then
    'MsgBox strMsg + "Section is safe"
    Print #1, Tab(5); "Compression resistance about Major Axis
    (kN)"; Tab(60); Format(Pdz, "#######0.0##"); " Section is
    safe"
    Print #1, ""
  Else
    MsgBox strMsg + "Revise the Section"
    blnSafe = False
    Unload Me
    frmBeamColumnDesign1.Show
  End If

End Sub
Private Sub DesignCompressionResisMinor()
Dim Fcry, Fcdy, Phiy, Phiy1  As Double
Dim strMsg As String

  'Compression Resistance- Minor Axis
  Fcry = PI * PI * E / (Lry * Lry)
  LambdaY = (Fy / Fcry) ^ 0.5
  Phiy = 0.5 * (1 + AlphaY * (LambdaY - 0.2) + LambdaY ^ 2)
  Phiy1 = Phiy + (Phiy ^ 2 - LambdaY ^ 2) ^ 0.5
  Fcdy = Fy / (GamaM * Phiy1)
  If Fcdy > (Fy / GamaM) Then
    Fcdy = Fy / GamaM
  End If
  Pdy = Fcdy * A / 1000#

  strMsg = "Compression resistance about Minor Axis " + Str(Pdy)
  + " kN" + vbCrLf
  If Pdy > N Then
    Print #1, Tab(5); "Compression resistance about Minor Axis
    (kNm)"; Tab(60); Format(Pdy, "#######0.0##"); " Section is
    safe"
    Print #1, ""
  Else
    MsgBox strMsg + "Revise the section"
    blnSafe = False
    Unload Me
    frmBeamColumnDesign1.Show
  End If
```

```
  If Pdz > Pdy Then
    Pd = Pdy
  Else
    Pd = Pdz
  End If
  Print #1, Tab(5); "Compression Resistance (kN)"; Tab(60);
  Format(Pd, "#######0.0##")
  Print #1, Tab(5); "Compressive Force (kN)"; Tab(60); Format(N,
  "#######0.0##")
  Print #1, ""
End Sub

Private Sub DesignBucklingResisMajor()
Dim PhiLT, PhiLT1, chi, C1, Chi1, Mcr1, fcrb As Double
Dim strMsg As String

'Member Buckling resistance
' Major Axis
  If Abs(Mz1) < Abs(Mz2) Then
    Chi1 = Mz1 / Mz2
  Else
    Chi1 = Mz2 / Mz1
  End If

  'Eqn suggested by Gardner and Nethercot
   C1 = 1.88 - 1.4 * Chi1 + 0.52 * Chi1 ^ 2
   If C1 > 2.7 Then C1 = 2.7

   fcrb = C1 * (1473.5 / Lry) ^ 2 * ((1 + (1 / 20) * (Lry / (D /
   tf)) ^ 2) ^ 0.5)
   Mcr = fcrb * Zz
   LambdaLT = (Betab * Zpz * Fy / Mcr) ^ 0.5
   LambdaLT1 = (1.2 * Zz * Fy / Mcr) ^ 0.5
   If LambdaLT > LambdaLT1 Then
      LambdaLT = LambdaLT1
   End If
   LambdaLT1 = (1.2 * Zz * Fy / Mcr) ^ 0.5
   If LambdaLT > LambdaLT1 Then
      LambdaLT = LambdaLT1
   End If
  alphaLT = 0.21
  PhiLT = 0.5 * (1 + alphaLT * (LambdaLT - 0.2) + LambdaLT ^ 2)
  PhiLT1 = PhiLT + (PhiLT ^ 2 - LambdaLT ^ 2) ^ 0.5
  chi = 1# / PhiLT1
  If chi > 1# Then
    chi = 1#
  End If
  fbd = chi * Fy / GamaM

  Mdz1 = fbd * Betab * Zpz / 1000000#
```

```
  strMsg = "Buckling Resistance in Bending about Major Axis" +
  Str(Mdz) + "kNm" + vbCrLf + "Applied Moment about Major axis "
  + Str(Mz) + "kNm" + vbCrLf
  If Mdz1 > Mz Then
    Print #1, Tab(5); "Buckling Resistance in Bending about Major
    Axis (kNm)"; Tab(60); Format(Mdz1, "#######0.0##")
    Print #1, Tab(5); "Applied Moment about Major axis (kNm)";
    Tab(60); Format(Mz, "#######0.0##")
    Print #1, Tab(5); "Buckling Resistance in Bending about Major
    Axis is safe"
    Print #1, ""
  Else
    MsgBox strMsg + "Revise the Section"
    blnSafe = False
    Unload Me
    frmBeamColumnDesign1.Show
  End If

End Sub


Private Sub DesignShearResistance()

Dim MSF, Av, Vp As Double
Dim strMsg As String

  'Shear Resistance of the cross-section
  MSF = Abs(Mz1 - Mz2) / (Lz / 1000)
  Av = D * tw
  Vp = Av * Fy / (Sqr(3) * GamaM * 1000#)

  strMsg = "Shear Resistance Parallel to Web " + Str(Vp) + " kN"
  + vbCrLf + " Max.S.F. " + Str(MSF) + " kN" + vbCrLf
  If Vp > MSF Then
    Print #1, Tab(5); "Shear Resistance Parallel to Web (kN)";
    Tab(60); Format(Vp, "#######0.0##")
    Print #1, Tab(5); "Maximum Shear Force (kN)"; Tab(60);
    Format(MSF, "#######0.0##")
    Print #1, Tab(5); "Shear Resistance Parallel to Web is safe"
    Print #1, ""
  Else
    MsgBox strMsg + "Revise the Section"
    blnSafe = False
    Unload Me
    frmBeamColumnDesign1.Show
  End If

End Sub

Private Sub DesignComBendingAxialForce()
Dim Psiz, BetaMz, MulT, Mufz, Muz, Muy, KKz, Kky, psiy, BetaMy,
nz, Cmz, Cmy, CmLT, ny, klt, klt1 As Double
Dim strMsg As String
```

```
'Member Bending Resistance in Combined Bending and axial Compression
'Major Axis
    Psiz = Mzm / Mz
    nz = N / Pdz
    Kz = 1 + (LambdaZ - 0.2) * nz
    Cmz = 0.6 + 0.4 * Psiz
    If Cmz < 0.4 Then
       Cmz = 0.4
    End If
    If Kz > 1 + 0.8 * nz Then
       Kz = 1 + 0.8 * nz
    End If
    klt = 0
    Cmy = 0
    'Minor axis
    If Not (Mym = 0 And My = 0) Then
        psiy = Mym / My
        ny = N / Pdy
        Ky = 1 + (LambdaY - 0.2) * ny

        If Ky > 1 + 0.8 * ny Then
           Ky = 1 + 0.8 * ny
        End If
        Cmy = 0.6 + 0.4 * psiy
        If Cmy < 0.4 Then
           Cmy = 0.4
        End If
        CmLT = 0.6 + 0.4 * psiy
        If CmLT < 0.4 Then
           CmLT = 0.4
        End If
        klt = 1 - 0.1 * LambdaLT * ny / (CmLT - 0.25)
        klt1 = 1 - 0.1 * ny / (CmLT - 0.25)
        If klt > klt1 Then
           klt = klt1
        End If
    End If
    'Check with Interaction Formula for overall buckling

    StressRatio1 = (N / Pdy) + (Ky * Cmy * Abs(My) / Mdy) + (klt
    * Abs(Mz) / Mdz1)
    StressRatio2 = (N / Pdz) + (0.6 * Ky * Cmy * Abs(My) / Mdy) +
    (Kz * Cmz * Abs(Mz) / Mdz1)
    strMsg = "Interaction equation value  = " + Str(StressRatio1)
    + vbCrLf
    If StressRatio1 < 1# And StressRatio2 < 1 Then
      Print #1, Tab(5); "Interaction equation value 1 "; Tab(60);
      Format(StressRatio1, "#######0.0##")
      Print #1, Tab(5); "Interaction equation value 2 "; Tab(60);
      Format(StressRatio2, "#######0.0##")
     Print #1, Tab(5); "Member is safe under combined Axial Force
     and B.M."
     Print #1, ""
```

```
    MsgBox "Member is safe" + vbCrLf + "The output is stored as
    BeamColumn.out under the working folder", , "Desgin Over"
  Else
    MsgBox strMsg + "Revise the section"
    blnSafe = False
      Unload Me
    frmBeamColumnDesign1.Show

  End If

End Sub

Private Sub optCompact_Click()
  If optCompact Then
  sectionType = COMPACT
  End If

End Sub

Private Sub optPlastic_Click()
  If optPlastic Then
    sectionType = PLASTIC
  End If
End Sub

Private Sub optSemiCompact_Click()
  If optSemiCompact Then
    sectionType = SEMI_COMPACT
  End If
End Sub
```

To illustrate the use of the program, a trial run of the program was conducted for example 13.1 and the resulting output is as follows:

```
Beam-Column Design
-----------------

Structure Data
-------------
Youngs Modulus <N/sq.mm>                          200000.0
Poisson's Ratio                                   0.3
Yield Stress <N/sq.mm>                            250.0
Major axis Length <mm>                            4000.0
Minor axis Length <mm>                            4000.0
Eff.Length factor - Major axis                    0.8
Eff.Length factor - Minor axis                    0.8
Factored axial load <kN>                          500.0
Major axis factored moment at top <kNm>           27.0
Major axis factored moment at bottom <kNm>        45.0
Minor axis factored moment at top <kNm>           0.0
Minor axis factored moment at bottom <kNm>        0.0
```

```
Sectional Properties
--------------------
Depth <mm>                                        250.0
Width of flange <mm>                              250.0
Thickness of flange <mm>                          9.7
Thickness of web <mm>                             6.9
Root radius <mm>                                  10.0
Radius of Gyration major axis <mm>                109.0
Radius of Gyration minor axis <mm>                54.9
Elastic modulus about major axis <cu.mm>          619000.0
Elastic modulus about minor axis <cu.mm>          156000.0
Partial safety factor for material                1.1
Area <sq.mm>                                      6500.0
Moment of intertia about minor axis <mm^4>        19600000.0

Output
-----
Plastic modulus about Major axis <cu.mm>          674456.721
Plastic modulus about Minor axis <cu.mm>          305869.717
Torsional constant <mm^4>                         178425.738
Warping constant <mm^6>                           282946041000.0
Design Resistance in Bending about
Minor Axis <kNm>                                  35.455
Applied Moment about Minor Axis <kNm>             0.0
Compression resistance about Major Axis <kN>      1407.624 Section
                                                  is safe
Compression resistance about Minor Axis <kN>      1110.522 Section
                                                  is safe
Compression Resistance <kN>                       1110.522
Compressive Force <kN>                            500.0

Buckling Resistance in Bending about
Major Axis <kNm>                                  128.525
Applied Moment about Major axis <kNm>             45.0
Buckling Resistance in Bending about
Major Axis is safe

Interaction equation value                        0.658 Section is
                                                  safe

Shear Resistance Parallel to Web <kN>             226.348
Maximum Shear Force <kN>                          5.625
Shear Resistance Parallel to Web is safe

Interaction equation value                        0.45
Interaction equation value2                       0.663
Member is safe under combined Axial Force and B.M.
```

The inclusion of these computer programs is only to illustrate the usefulness of computers in repetitive and complex calculations, which are time consuming if done manually. However, the publisher and the author do not provide any warranty for their use to design problems. The user is advised to test the programs thoroughly, before using them.