

## APPENDIX B

---

# Examples of Computer Programs in Visual BASIC

### B.1 Y0YC PROGRAM FOR CALCULATION OF NORMAL AND CRITICAL DEPTH IN A TRAPEZOIDAL CHANNEL

The Visual BASIC *form* is illustrated in Figure B.1. The *form code* extracts values entered in the text boxes and assigns them to variable names when the CALCULATE button is clicked. The channel parameters are  $Q$  = discharge in cfs;  $S$  = channel slope in ft/ft;  $b$  = channel bottom width in ft;  $m$  = channel sideslope ratio as run-to-rise; and  $n$  = Manning's roughness coefficient. The form code then calls a subprocedure Y0YC in the *module code*. The module calculates normal and critical depth in a trapezoidal channel using the bisection method. The results are then placed in the output text boxes. In the module code, the function  $F$  evaluates either the critical depth function or the normal depth function depending on whether NFUNC has the value 1 or 2, respectively. The BISECTION subprocedure is identical to that shown in Figure A.2a, except that additional variables have to be passed to BISECTION in the parameter list. The program input includes rectangular and triangular channel shapes by specifying  $m = 0$  or  $b = 0$ , respectively.

#### Y0YC Form Code

```
Option Explicit
Dim Q As Single, S As Single, b As Single, n As Single, m As Single
Dim Y0 As Single, YC As Single
'*****
Private Sub cmdCalculate_Click()
Q = Val(txtQ.Text)
S = Val(txtS.Text)
b = Val(txtB.Text)
n = Val(txtn.Text)
```

2 APPENDIX B: Examples of Computer Programs in Visual BASIC

**NORMAL AND CRITICAL DEPTH IN A TRAPEZOIDAL CHANNEL**

Discharge <u>Q</u> , cfs:	<input type="text"/>
Slope <u>S</u> , ft/ft:	<input type="text"/>
Bottom Width <u>b</u> , ft:	<input type="text"/>
Sideslope <u>m</u> :1	<input type="text"/>
Manning's <u>n</u> :	<input type="text"/>
<input type="button" value="CALCULATE"/>	
Normal Depth, ft =	<input type="text"/>
Critical Depth, ft =	<input type="text"/>
<input type="button" value="EXIT"/>	

**FIGURE B.1**

Visual BASIC form for determining normal and critical depth in a trapezoidal channel using the program Y0YC.

```

m = Val(txtSS.Text)
Call Y0YC(Q, S, b, m, n, Y0, YC)
txtY0.Text = Format(Y0, "###.000")
txtYC.Text = Format(YC, "###.000")
End Sub
'*****
Private Sub cmdExit_Click()
End
End Sub
'*****
    
```

**Y0YC Module Code**

```

Option Explicit
Dim Q As Single, S As Single, b As Single, m As Single
Dim n As Single, Y0 As Single, YC As Single
    
```

## APPENDIX B: Examples of Computer Programs in Visual BASIC 3

```

Sub YOYC(Q, S, b, m, n, Y0, YC)
Dim Y1 As Single, Y2 As Single, ER As Single
Dim NF As Integer
Y1 = 0.0001
Y2 = 100
ER = 0.0001
NF = 1
Call BISECTION(Y1, Y2, NF, ER, Q, S, b, m, n, YC)
Y1 = 0.0001
Y2 = 100
NF = 2
Call BISECTION(Y1, Y2, NF, ER, Q, S, b, m, n, Y0)
End Sub
'*****
Sub BISECTION(Y1, Y2, NFUNC, ER, Q, S, b, m, n, Y3)
Dim FY1 As Single, FY2 As Single, FY3 As Single, FZ As Single
Dim I As Integer
    FY1 = F(Y1, NFUNC, Q, S, b, m, n)
    FY2 = F(Y2, NFUNC, Q, S, b, m, n)
    If FY1 * FY2 > 0 Then Exit Sub
For I = 1 To 50
    Y3 = (Y1 + Y2) / 2
    FY3 = F(Y3, NFUNC, Q, S, b, m, n)
    FZ = FY1 * FY3
    If FZ = 0 Then Exit Sub
    If FZ < 0 Then Y2 = Y3 Else Y1 = Y3
    If Abs((Y2 - Y1) / Y3) < ER Then Exit Sub
Next I
End Sub
'*****
Function F(Y, NFUNC, Q, S, b, m, n) As Single
Dim A As Single, P As Single, R As Single, T As Single
    A = Y * (b + m * Y)
    P = b + 2 * Y * Sqr(1 + m ^ 2)
    R = A / P
    T = b + 2 * m * Y
    If NFUNC = 1 Then
        F = Q - Sqr(32.2) * A ^ 1.5 / T ^ 0.5
    Else
        F = Q - (1.486 / n) * A * R ^ (2 / 3) * S ^ (1 / 2)
    End If
End Function
End Function

```

## B.2 YCOMP PROGRAM FOR FINDING MULTIPLE CRITICAL DEPTHS IN A COMPOUND CHANNEL

Input data for the program Ycomp include the discharge  $Q$  and the slope  $S_0$ , the latter of which is used only for the computation of normal depth for comparison with critical depth(s). The input data are placed in text boxes in the Visual

4 APPENDIX B: Examples of Computer Programs in Visual BASIC

**NORMAL AND CRITICAL DEPTH IN A COMPOUND CHANNEL**

**DATA INPUT**

Discharge  $Q$ , cfs =

Slope  $S_0$ , ft/ft =

INPUT DATA FILE 

Current File

CALCULATE

**RESULTS**

CROSS SECTION:

LOWER  $Y_c$ , FT

UPPER  $Y_c$ , FT

LOWER  $Q$ , CFS

UPPER  $Q$ , CFS

$Y_{normal}$ , FT

**FIGURE B.2**

Visual BASIC form for determining multiple critical depths in a compound channel using the program Ycomp.

BASIC form shown in Figure B.2. An input data file that gives the compound channel geometry and roughness is required and is specified by clicking on the INPUT DATA FILE button. The form code is given in Sub cmdInput. A sample data file is shown at the end of the module code. It consists of the station-elevation pairs, which define the cross-section geometry, and the subsection locations with corresponding Manning's  $n$  values. The subsection locations must correspond to ground points, and vertical banks are not allowed. The module code finds the locations of the right and left banks, assuming that the main channel consists of only one subsection.

## APPENDIX B: Examples of Computer Programs in Visual BASIC 5

The results are computed by the module code, which includes a bisection sub-procedure to solve for the critical depths based on the compound channel function given by the function subprocedure FC. The program determines if there are values of critical depth in the main channel (lower  $Y_c$ ) and floodplain (upper  $Y_c$ ). It also determines the discharge range, if any, over which multiple critical depths in the main channel and floodplain exist. For discharges greater than the upper  $Q$ , only the upper critical depth occurs, while only the lower critical depth exists for discharges less than the lower  $Q$ . In between these two discharges, both lower and upper critical depths occur. If the compound channel has only one critical depth, then the lower  $Q$  equals the upper  $Q$ .

**Ycomp Form Code**

```
Option Explicit
Dim DIST(100), ELEV(100), XSUB(20), RN(20)
Dim Q As Single, YC1 As Single, YC2 As Single, YN As Single
Dim QL As Single, QU As Single, S0 As Single
Dim NP As Integer, NSUB As Integer
'*****
Private Sub cmdCalculate_Click()
Q = txtQ.Text
S0 = txtS0.Text
Call Ycomp(DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, S0, YN, YC1, YC2, QL, QU)
txtYN.Text = Format(YN, "###.000")
If YC1 < 0 Then
    txtYC1.Text = "Does Not Exist"
Else
    txtYC1.Text = Format(YC1, "###.000")
End If
If YC2 < 0 Then
    txtYC2.Text = "Does Not Exist"
Else
    txtYC2.Text = Format(YC2, "###.000")
End If
txtQL.Text = Format(QL, "###.")
txtQU.Text = Format(QU, "###.")
End Sub
'*****
Private Sub cmdInput_Click()
Dim K As Integer
Dim strFileIn As String, strXS As String
cdbFile.ShowOpen
strFileIn = cdbFile.FileName
If strFileIn = "" Then Exit Sub
txtFName.Text = strFileIn
Open strFileIn For Input As #1
Input #1, strXS, NP, NSUB
For K = 1 To NP
```

## 6 APPENDIX B: Examples of Computer Programs in Visual BASIC

```

Input #1, DIST(K), ELEV(K)
Next K
For K = 1 To NSUB
Input #1, XSUB(K), RN(K)
Next K
Close 1
txtXS.Text = strXS
End Sub
'*****
Private Sub cmdExit_Click()
End
End Sub

```

**Ycomp Module Code**

```

Option Explicit
Dim DIST(100), ELEV(100), XSUB(20), RN(20)
Dim Q As Single, YC1 As Single, YC2 As Single, YN As Single
Dim QU As Single, QL As Single, S0 As Single
Dim NP As Integer, NSUB As Integer
'*****
Sub Ycomp(DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, S0, YN, YC1, YC2, QL, QU)
Dim EMIN As Single, EMAX As Single, ELEVBANKFULL As Single
Dim WS As Single, WS1 As Single, WS2 As Single, WS3 As Single, WSMAX As Single
Dim FROUDE As Single, FROUDE1 As Single, FROUDEMAX As Single
Dim ELCRIT1 As Single, ELCRIT2 As Single, E1 As Single, DELTAY As Single
Dim J As Integer, KMIN As Integer, K As Integer, ILTBANK As Integer
Dim ITER As Integer, KLT As Integer, KRT As Integer, IRTBANK As Integer
Dim strC As String
Const G = 32.2!
If ELEV(1) >= ELEV(NP) Then 'find maximum elevation
    EMAX = ELEV(1)
Else
    EMAX = ELEV(NP)
End If
KMIN = 1: EMIN = ELEV(KMIN)
For K = 2 To NP 'find minimum elevation in main channel
If ELEV(K) < EMIN Then
    EMIN = ELEV(K)
    KMIN = K
End If
Next K
For J = 1 To NSUB 'find subsection number of banks
If XSUB(J) > DIST(KMIN) Then
    IRTBANK = J
    ILTBANK = J - 1
Exit For
End If
Next J

```

## APPENDIX B: Examples of Computer Programs in Visual BASIC 7

```

For K = 1 To NP 'find point number of left bank
If DIST(K) = XSUB(ILTBANK) Then
    KLT = K
    Exit For
End If

Next K
For K = 1 To NP 'find point number of right bank
If DIST(K) = XSUB(IRTBANK) And DIST(K) = DIST(K + 1) Then
    KRT = K + 1
    Exit For
ElseIf DIST(K) = XSUB(IRTBANK) Then
    KRT = K
    Exit For
End If

Next K
If ELEV(KLT) <= ELEV(KRT) Then
    ELEVBANKFULL = ELEV(KLT)
Else
    ELEVBANKFULL = ELEV(KRT)
End If

WS1 = EMIN + 0.01: WS2 = EMAX: strC = "NORM" 'calculate normal depth
Call Bisection(WS1, WS2, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, S0, strC, _
    WS3)
YN = WS3 - EMIN
WS = ELEVBANKFULL: strC = "CRIT" 'calculate critical depth(s)
FROUDE1 = FC(WS, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, S0, strC)
QU = Q / FROUDE1
If Q > QU Then
    ELCRIT1 = -1: YC1 = -1 'calculate upper yc
    WS1 = ELEVBANKFULL: WS2 = EMAX: strC = "CRIT"
    Call Bisection(WS1, WS2, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, _
        S0, strC, WS3)
    ELCRIT2 = WS3: YC2 = ELCRIT2 - EMIN
Else 'calculate lower yc
    ELCRIT2 = -1: YC2 = -1 'possibly no upper yc
    WS1 = EMIN + 0.01: WS2 = ELEVBANKFULL: strC = "CRIT"
    Call Bisection(WS1, WS2, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, _
        S0, strC, WS3)
    ELCRIT1 = WS3: YC1 = ELCRIT1 - EMIN
End If
E1 = ELEVBANKFULL
DELTAY = 0.01 * (ELEVBANKFULL - EMIN)
WS = ELEVBANKFULL + 2! * DELTAY: strC = "CRIT"
FROUDE = FC(WS, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, S0, strC) 'calculate F+
If FROUDE < FROUDE1 Then
    QL = QU 'Case I-only upper yc with no multiple yc
    Exit Sub 'or Case III-only lower yc with no multiple yc
End If
FROUDEMAX = FROUDE1 'find the maximum Froude number
For ITER = 1 To 100

```

## 8 APPENDIX B: Examples of Computer Programs in Visual BASIC

```

E1 = E1 + DELTAY
If E1 > EMAX Then Exit For
WS = E1: strC = "CRIT"
FROUDE = FC(WS, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, S0, strC)
If FROUDE > FROUDEMAY Then
    FROUDEMAY = FROUDE
    WSMAX = E1
End If
Next ITER
QL = QU / (FROUDEMAY / FROUDE1)
If Q > QU Then Exit Sub 'Case I-only upper yc with multiple yc
If Q > QL Then
    WS1 = WSMAX: WS2 = EMAX 'Case II-both lower and upper yc
    strC = "CRIT"
    Call Bisection(WS1, WS2, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, _
    S0, strC, WS3)
    ELCRIT2 = WS3: YC2 = ELCRIT2 - EMIN
    Exit Sub
Else 'Case III-only lower yc with multiple yc
    ELCRIT2 = -1: YC2 = -1
End If
End Sub
'*****
Sub Bisection(WS1, WS2, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, S0, strC, WS3)
Dim WS As Single, FWS1 As Single, FWS2 As Single, FWS3 As Single, FZ As Single
Dim I As Integer
Const ER = 0.001!
WS = WS1
FWS1 = FC(WS, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, S0, strC) - 1
WS = WS2
FWS2 = FC(WS, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, S0, strC) - 1
If FWS1 * FWS2 > 0 Then frmYcomp.Print "!!NO ROOT!!": Exit Sub
For I = 1 To 50
    WS3 = (WS1 + WS2) / 2
    WS = WS3
    FWS3 = FC(WS, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, S0, strC) - 1
    FZ = FWS1 * FWS3
    If FZ < 0 Then
        WS2 = WS3
    ElseIf FZ > 0 Then
        WS1 = WS3
    Else
        Exit Sub
    End If
    If Abs(WS2 - WS1) < ER Then Exit Sub
Next I
frmYcomp.Print "ERROR CRITERION NOT SATISFIED"
End Sub
'*****
Function FC(WS, DIST(), ELEV(), XSUB(), RN(), NP, NSUB, Q, S0, strC) As Single
Dim ASUB(20) As Single, PSUB(20) As Single, CSUB(20) As Single

```



## APPENDIX B: Examples of Computer Programs in Visual BASIC 9

```

Dim AT As Single, BT As Single, CT As Single, RT As Single, PT As Single
Dim ALPHA As Single, S1 As Single, S2 As Single, S3 As Single
Dim Y1 As Single, Y2 As Single, DY As Single, DX As Single, DP As Single
Dim A As Single, P As Single, B As Single, DPDY As Single, YN As Single
Dim R As Single, C As Single, TERM1 As Single, TERM2 As Single, DALDY As Single
Dim J As Integer, K As Integer, L As Integer
Const G = 32.2!
AT = 0: BT = 0: CT = 0: RT = 0: PT = 0: ALPHA = 0
S1 = 0: S2 = 0: S3 = 0: A = 0: P = 0: B = 0: DPDY = 0
For L = 1 To NSUB
ASUB(L) = 0!: PSUB(L) = 0!: CSUB(L) = 0!
Next L
J = 1 'begin geometric evaluation; J=subsection no.
Y1 = WS - ELEV(1)
For K = 2 To NP
Y2 = WS - ELEV(K)
DX = DIST(K) - DIST(K - 1)
If Y2 > Y1 Then DY = Y2 Else DY = Y1
If DY = Y2 Then YN = Y1 Else YN = Y2
If (Y1 * Y2) < 0 Then
    DX = DX * DY / (DY - YN)
    A = A + 0.5 * DX * DY
    DP = Sqr(DX * DX + DY * DY)
    P = P + DP
    B = B + DX
    DPDY = DPDY + DP / DY
ElseIf (Y1 + Y2) > 0 Then
    A = A + 0.5 * DX * (Y1 + Y2)
    DP = Sqr(DX * DX + (Y1 - Y2) * (Y1 - Y2))
    P = P + DP
    B = B + DX
    If YN = 0 Then DPDY = DPDY + DP / DY
End If
Y1 = Y2
'*****Update subsection properties if end of subsection reached
If DIST(K) = XSUB(J) Then
    If A = 0 Then
        ASUB(J) = 0!
        PSUB(J) = 0!
        CSUB(J) = 0!
    Else
        A = Abs(A): B = Abs(B)
        R = A / P
        C = 1.49 * A * R ^ (2! / 3!) / RN(J)
        ASUB(J) = A: PSUB(J) = P: CSUB(J) = C
        AT = AT + A: BT = BT + B
        CT = CT + C: PT = PT + P
        S1 = S1 + (C / A) ^ 3 * (3! * B - 2! * R * DPDY)
        S2 = S2 + C ^ 3 / A ^ 2
        S3 = S3 + C / A * (5! * B - 2! * R * DPDY)
    End If

```

## 10 APPENDIX B: Examples of Computer Programs in Visual BASIC

```

        J = J + 1
        A = 0: P = 0: B = 0: DPDY = 0
    End If
Next K
If AT = 0 Then Exit Function
RT = AT / PT
ALPHA = S2 / (CT ^ 3 / AT ^ 2)
TERM1 = AT ^ 2 * S1 / CT ^ 3
TERM2 = S2 * (2 * AT * BT / CT ^ 3 - AT ^ 2 * S3 / CT ^ 4)
DALDY = TERM1 + TERM2
TERM1 = ALPHA * Q ^ 2 * BT / (G * AT ^ 3)
TERM2 = Q ^ 2 * DALDY / (2 * G * AT ^ 2)
If strC = "CRIT" Then
    FC = Sqr(TERM1 - TERM2)
ElseIf strC = "NORM" Then
    FC = Q / (CT * S0 ^ (1 / 2))
End If
End Function

```

**Data File**

```

"CSA",8,3 'Name of cross section, number of ground points, number of subsections
0,110 'Ground points: Station, elevation (from left to right looking downstream)
4,106
604,106
610,100
682,100
688,106
1288,106
1292,110
604 , .08 , 688 , .03 , 1292 , .08 'Location of subsections: Station,
'Manning's n on the left side

```

**B.3 WSP PROGRAM FOR WATER SURFACE PROFILE COMPUTATION**

The program WSP computes normal and critical depth, classifies the water surface profile, and computes the water surface profile in a trapezoidal channel using the direct step method. The Visual BASIC form is shown in Figure B.3. First, the channel discharge,  $Q$ ; slope,  $S$ ; bottom width,  $b$ ; side slope ratio,  $m$ ; and Manning's  $n$  are entered in the text boxes on the form. Then, the values of critical depth,  $Y_C$ , and normal depth,  $Y_0$ , are calculated as in the program Y0YC. For the water surface profile computation, the channel length,  $XL$ , and the control depth,  $Y_{CONT}$ , are entered next. If critical depth is the control, zero can be entered in the control depth box, and the program will assign a depth slightly greater or less than critical

**WATER SURFACE PROFILE IN A TRAPEZOIDAL CHANNEL**

**RESULTS**

<b>Discharge <u>Q</u>, cfs:</b>	<input type="text"/>	<b>Channel Length, ft:</b>	<input type="text"/>
<b>Slope <u>S</u>, ft/ft:</b>	<input type="text"/>	<b>Control Depth, ft: (0 for critical)</b>	<input type="text"/>
<b>Bottom Width <u>b</u>, ft:</b>	<input type="text"/>		
<b>Sideslope <u>m</u>:1</b>	<input type="text"/>		
<b>Manning's <u>n</u>:</b>	<input type="text"/>		
	<b>CALCULATE <u>Y</u><sub>0</sub>, <u>Y</u><sub>C</sub></b>		<b>CALCULATE PROFILE</b>
<b>Normal Depth, ft =</b>	<input type="text"/>	<b>Profile Type =</b>	<input type="text"/>
<b>Critical Depth, ft =</b>	<input type="text"/>	<b>PRINT</b>	<b>EXIT</b>



**FIGURE B.3**

Visual BASIC form for calculating water surface profiles in trapezoidal channels by the direct step method using the program WSP.

## 12 APPENDIX B: Examples of Computer Programs in Visual BASIC

depending on the profile classification in order to avoid starting on the wrong profile. The CALCULATE PROFILE button is clicked, and the profile type is classified and shown. The profile is calculated in Sub PROFILE using the direct step method by dividing the depth interval appropriate to the particular profile type into 30 increments. The specific energy and slope of the energy grade line are calculated in Sub SPENERGY. Interpolation is applied to obtain the depth at the end of the specified length of channel. The channel stations increase in the downstream direction regardless of whether the computation proceeds in the upstream direction (subcritical) or downstream direction (supercritical). If normal depth or critical depth is approached within 0.1 percent before the specified channel length is reached, the computation is stopped. The results for depth, YP; velocity, VP; and momentum function, MP, are printed at each step, XP, in the RESULTS picture box. The output can be directed to a printer with the PRINT button. The program accommodates horizontal, mild, and steep slopes.

**WSP Form Code**

```
Option Explicit
Dim Q As Single, S As Single, b As Single, n As Single, m As Single
Dim Y0 As Single, YC As Single, XL As Single, YCONT As Single
Dim strPR As String
Dim J As Integer, NP As Integer
Dim XP(100), YP(100), VP(100), MP(100)
'*****
Private Sub cmdCalculate_Click()
Q = Val(txtQ.Text)
S = Val(txtS.Text)
b = Val(txtB.Text)
n = Val(txtn.Text)
m = Val(txtSS.Text)
Call Y0YC(Q, S, b, m, n, Y0, YC)
If Y0 = 1000# Then
    txtY0.Text = "does not exist"
Else
    txtY0.Text = Format(Y0, "###.000")
End If
txtYC.Text = Format(YC, "###.000")
End Sub
'*****
Private Sub cmdPROFILE_Click()
XL = Val(txtXL.Text)
YCONT = Val(txtYCONT.Text)
Call PROFILE(Q, S, b, m, n, Y0, YC, XL, YCONT, strPR, NP, XP(), YP(), VP(), MP())
txtPROFILE.Text = strPR
picResults.Cls
picResults.Print " X, ft", " Y,ft", "V, ft/s", "Mom. F, cuft"
picResults.Print
For J = 1 To NP
```

## APPENDIX B: Examples of Computer Programs in Visual BASIC 13

```

XP(J) = Format(XP(J), " #####.0")
YP(J) = Format(YP(J), "###.000")
VP(J) = Format(VP(J), "###.00")
MP(J) = Format(MP(J), "#####.0")
picResults.Print XP(J), YP(J), VP(J), MP(J)
Next J
End Sub
'*****
Private Sub cmdPrint_Click()
Printer.FontName = "Arial"
Printer.FontSize = 12
Printer.CurrentY = 1500
Printer.Print Spc(30), "WATER SURFACE PROFILE - "; strPR
Printer.Print
Printer.Print Spc(20), " n="; n; Tab; "Q, cfs ="; Q; Spc(5); "Slope="; S
Printer.Print Spc(20), " m="; m; Tab; "b, ft ="; b
Printer.Print
Y0 = Format(Y0, "###.000")
YC = Format(YC, "###.000")
If S = 0 Then
Printer.Print Spc(20), "Y0 does not exist"
Else
Printer.Print Spc(20), " Y0,ft ="; Y0
End If
Printer.Print Spc(20), " YC,ft ="; YC
Printer.Print
Printer.Print Spc(20), " X, ft", "Y, ft", "V, ft/s", "Mom. F."
For J = 1 To NP
XP(J) = Format(XP(J), " #####.0")
YP(J) = Format(YP(J), "###.000")
VP(J) = Format(VP(J), "###.00")
MP(J) = Format(MP(J), "#####.0")
Printer.Print Spc(20), XP(J), YP(J), VP(J), MP(J)
Next J
Printer.EndDoc
End Sub
'*****
Private Sub cmdExit_Click()
End
End Sub

```

**WSP Module Code**

```

Option Explicit
Dim Q As Single, S As Single, b As Single, m As Single
Dim n As Single, Y0 As Single, YC As Single
'*****
Sub Y0YC(Q, S, b, m, n, Y0, YC)
Dim Y1 As Single, Y2 As Single, ER As Single

```

## 14 APPENDIX B: Examples of Computer Programs in Visual BASIC

```

Dim NF As Integer
Y1 = 0.0001
Y2 = 100
ER = 0.0001
NF = 1
Call BISECTION(Y1, Y2, NF, ER, Q, S, b, m, n, YC)
If S = 0 Then
    Y0 = 1000#
    Exit Sub
End If
Y1 = 0.0001
Y2 = 100
NF = 2
Call BISECTION(Y1, Y2, NF, ER, Q, S, b, m, n, Y0)
End Sub
'*****
Sub PROFILE(Q, S, b, m, n, Y0, YC, XL, YCONT, strPR, NP, XP(), YP(), VP(), MP())
Dim XSIGN As Single, YLIM As Single, E As Single, SEGL As Single
Dim DX As Single, DY As Single, Y As Single, SEGL1 As Single, E1 As Single
Dim SEGL2 As Single, E2 As Single, SEGLBAR As Single, VEL As Single, FM As Single
Dim I As Integer, NS As Integer
If YCONT = 0 Then YCONT = YC
If YC < Y0 Then 'classify mild slope profiles
    If YCONT > Y0 Then
        strPR = "M1": XSIGN = -1: YLIM = 1.001 * Y0
    ElseIf YCONT < YC Then
        strPR = "M3": XSIGN = 1: YLIM = 0.999 * YC
    Else
        strPR = "M2": XSIGN = -1: YLIM = 0.999 * Y0
        If YCONT = YC Then YCONT = 1.001 * YC
    End If
Else
    If YCONT > YC Then 'classify steep slope profiles
        strPR = "S1": XSIGN = -1: YLIM = 1.001 * YC
    ElseIf YCONT < Y0 Then
        strPR = "S3": XSIGN = 1: YLIM = 0.999 * Y0
    Else
        strPR = "S2": XSIGN = 1: YLIM = 1.001 * Y0
        If YCONT = YC Then YCONT = 0.999 * YC
    End If
End If
If S = 0 Then 'classify horizontal slope profiles
    If YCONT >= YC Then
        strPR = "H1": XSIGN = -1: YLIM = YCONT + 3# * YC
        If YCONT = YC Then YCONT = 1.001 * YC
    Else
        strPR = "H2": XSIGN = 1: YLIM = 0.999 * YC
    End If
End If
NS = 30 'initialize variables

```

## APPENDIX B: Examples of Computer Programs in Visual BASIC 15

```

NP = NS + 1
DY = (YLIM - YCONT) / NS
If XSIGN = 1 Then XP(1) = 0 Else XP(1) = XL
Y = YCONT
Call SPENERGY(Y, Q, S, b, m, n, E, SEGL, VEL, FM)
YP(1) = Y: VP(1) = VEL: MP(1) = FM
E1 = E: SEGL1 = SEGL
For I = 2 To NS + 1 'direct step method loop
Y = Y + DY
Call SPENERGY(Y, Q, S, b, m, n, E, SEGL, VEL, FM)
YP(I) = Y: VP(I) = VEL: MP(I) = FM
E2 = E: SEGL2 = SEGL
SEGLBAR = 0.5 * (SEGL1 + SEGL2)
DX = (E2 - E1) / (S - SEGLBAR)
XP(I) = XP(I - 1) + DX
E1 = E2: SEGL1 = SEGL2
If XP(I) > XL Then 'interpolation for specified length
    YP(I) = YP(I) - DY * (XP(I) - XL) / DX
    XP(I) = XL
    NP = I
    Call SPENERGY(YP(I), Q, S, b, m, n, E, SEGL, VEL, FM)
    VP(I) = VEL: MP(I) = FM
    Exit For
End If
If XP(I) < 0# Then 'interpolation for specified length
    YP(I) = YP(I) - DY * XP(I) / DX
    XP(I) = 0#
    NP = I
    Call SPENERGY(YP(I), Q, S, b, m, n, E, SEGL, VEL, FM)
    VP(I) = VEL: MP(I) = FM
    Exit For
End If
Next I
End Sub
'*****
Sub BISECTION(Y1, Y2, NFUNC, ER, Q, S, b, m, n, Y3)
Dim FY1 As Single, FY2 As Single, FY3 As Single, FZ As Single
Dim I As Integer
    FY1 = F(Y1, NFUNC, Q, S, b, m, n)
    FY2 = F(Y2, NFUNC, Q, S, b, m, n)
    If FY1 * FY2 > 0 Then Exit Sub
For I = 1 To 50
    Y3 = (Y1 + Y2) / 2
    FY3 = F(Y3, NFUNC, Q, S, b, m, n)
    FZ = FY1 * FY3
    If FZ = 0 Then Exit Sub
    If FZ < 0 Then Y2 = Y3 Else Y1 = Y3
    If Abs((Y2 - Y1) / Y3) < ER Then Exit Sub
Next I
End Sub

```

## 16 APPENDIX B: Examples of Computer Programs in Visual BASIC

```

*****
Function F(Y, NFUNC, Q, S, b, m, n) As Single
Dim A As Single, P As Single, R As Single, T As Single
    A = Y * (b + m * Y)
    P = b + 2 * Y * Sqr(1 + m ^ 2)
    R = A / P
    T = b + 2 * m * Y
    If NFUNC = 1 Then
        F = Q - Sqr(32.2) * A ^ 1.5 / T ^ 0.5
    Else
        F = Q - (1.486 / n) * A * R ^ (2 / 3) * S ^ (1 / 2)
    End If
End Function
*****
Sub SPENERGY(Y, Q, S, b, m, n, E, SEGL, VEL, FM)
Dim A As Single, P As Single, R As Single
Dim KN As Single, G As Single
    G = 32.2: KN = 1.486
    A = Y * (b + m * Y)
    P = b + 2 * Y * Sqr(1 + m ^ 2)
    R = A / P
    VEL = Q / A
    FM = b * Y ^ 2 / 2 + m * Y ^ 3 / 3 + Q ^ 2 / (G * A)
    E = Y + Q ^ 2 / (2 * G * A ^ 2)
    SEGL = n ^ 2 * Q ^ 2 / (KN ^ 2 * A ^ 2 * R ^ (4# / 3#))
End Sub

```