THE ‘3RP’
COMPUTER PROGRAM FOR THE SOLUTION OF
THE THREE RESERVOIR PROBLEM IN HYDRAULICS

Qaisar Ali*

ABSTRACT

Computer software has become the driving force for modern scientific investigation and engineering problem solving. Software is virtually inescapable in a modern world.

This paper introduces a software named “The 3RP” which aims at studying and solving a problem of Fluid Mechanics and Hydraulics generally known as the Classical Three Reservoir Problem. Besides minimizing the hectic calculations involved in this problem, the 3RP is intended to serve as a pointer for Engineering Students and Practitioners that directs their attention towards the relatively unexplored vast area of software development for the subjects of Fluid Mechanics and Hydraulics.

INTRODUCTION

As the purpose of this paper is to introduce the software, so only a brief explanation of the related theory about the three reservoir problem is presented.

Suppose that three reservoirs A, B and C are connected to a common junction ‘J’ by pipes 1, 2 and 3 in which the friction losses are $h_1$, $h_2$, $h_3$ respectively. Moreover, it is assumed that all the pipes are sufficiently long, so that minor loses and velocity heads may be neglected.

Considering the given system, it is evident that the Continuity and Energy equations require that the flow entering the junction equals the flow leaving it and that the pressure head at ‘J’ (schematically shown by an open piezometer tube, with water at elevation ‘P’) be common to all pipes. So, for the condition in Fig. 1, $Q_1 = Q_2 + Q_3$.

The three problems that arise here are:

1. All pipe lengths and diameters are known. The surface elevations of two reservoirs are given and that of the third one is to be determined.

2. All the lengths and diameters of pipes are known. The elevations of water surfaces of two reservoirs and the flow to or from the third are given and the problem is to find the elevation of surface of the third reservoir.

3. All pipe lengths, diameters and the elevations of all three reservoirs are known. The problem is to find the flow in each pipe.

The reason behind choosing the third case for programming is that it is the one in which it is not immediately evident whether the flow is into or out of reservoir ‘B’ (see Fig. 1). For solving this problem an imaginary piezometer tube is considered at the junction box. The piezometer tube is then moved up or down by trial and adjustment until the resulting flow distribution satisfies the continuity relation. It is this area of the problem which involves lengthy computations but the use of computer technology has put an end to such an inconvenience.

PROGRAMME LANGUAGE

As widely understood, Object Oriented Languages are the most powerful programming tools amongst all the computer programming languages. So mainly, the programmer had two options, either to develop the software through Visual Basic or to do it with Visual C++.

Visual Basic is used as the programming language as it is fairly simple and uses common English words and phrases for the most part, so non programmers could

* Department of Civil Engineering, N.W.F.P University of Engineering and Technology, Peshawar.
understand the program as well. The language is not ambiguous, however, when a statement is written in Visual Basic language, the statement never has multiple meanings within the same context. Visual Basic programming language is one of the easiest programming languages in existence. It is simple but powerful, however, its simplicity does not translate to inability. Today Visual Basic is considered as one of the most powerful Windows programming languages in the market and supports advanced programming techniques.

On the other hand C started out as a language for technical programmers. Although C++ is now considered as a general-purpose language, still carries the baggage of its C ancestry. To master C++ completely, one must be more aware of the way computer works (which is not recommended, at least for a Civil Engineer) than most other languages require. The reason for this is that Visual Basic provides checks to ensure that you have not made a silly programming error when running your program. Although these checks slow the program down a little bit and sometimes stop the programmer from deliberately doing something technical with the computer but on the other hand C++ do not have these checks.

THE PROGRAM ' 3RP ' (DEVELOPED IN VISUAL BASIC)

*********************************************************************************************************************************
'Start of program

Dim EjGIVEN, X, Y, Z, j, mystring, i As Variant
Dim lowestE, highestE, minarray(3) As Integer
Const pi = 3.141592654
*********************************************************************************************************************************

Private Sub CmdStart_Click()

'To prompt the user in case he hasn't chosen the system of units
If Option1.Value = False And Option 2.Value = False Then
    MsgBox "CHOOSE SYSTEM OF UNITS, PLEASE!"
    Exit sub
End If

'In case of incomplete data, a message is displayed
    MsgBox "ENTER COMPLETE DATA PLEASE"
    Exit Sub
End If

'Data is obtained through the following variables
E1 = Text2.Text 'Elevation of first Reservoir
L1 = Text3.Text 'Length of pipe connected to first Reservoir
D1 = Text4.Text 'Diameter of pipe connected to first Reservoir
f1 = Text5.Text  'Friction Loss in the first pipe
E2 = Text7.Text  'Elevation of Second Reservoir
L2 = Text8.Text  'Length of pipe connected to second Reservoir
D2 = Text9.Text  'Diameter of pipe connected to second Reservoir
F2 = Text10.Text 'Friction Loss in the second pipe
E3 = Text12.Text 'Elevation of third Reservoir
L3 = Text13.Text 'Length of pipe connected to third Reservoir
D3 = Text14.Text 'Diameter of pipe connected to third Reservoir
F3 = Text15.Text 'Friction Loss in the third pipe
Ej = Text16.Text 'Elevation of the junction box

minarray(0) = Val(Text2.Text)  'To find minimum value of E
minarray(1) = Val(Text7.Text)
minarray(2) = Val(Text12.Text)
lowestE = Val(minarray(0))

For j = 0 To 2
    If minarray(j) < lowestE Then
        lowestE = minarray(j)
    End If
Next

minarray(0) = Val(Text2.Text)  'To find maximum value of E
minarray(1) = Val(Text7.Text)
minarray(2) = Val(Text12.Text)
highestE = Val(minarray(0))

For j = 0 To 2
    If minarray(j) > highestE Then
        highestE = minarray(j)
    End If
Next

If Text16.Text = "" Then 'WHEN Ej IS NOT KNOWN
    For Ej = lowestE To highestE
        HL1 = E1 - Ej
        HL2 = E2 - Ej
        HL3 = Ej - E3
    Next

'For metric system of units
If Option1.Value = True Then
    g = 9.8
End If

'For English system of units
If Option2.Value = True Then
    g = 32.2
End If

Exit For
End If
End If

If Q2 > Q1 + Q3 Then
    S1 = Q1 + Q3 + 2
    S2 = Q1 + Q3 - 2
    Q2 = v2 * a2
If \( Q2 < S1 \) And \( Q2 > S2 \) Then

\[
\text{Text1.Text} = (Q1) \\
\text{Text6.Text} = (Q2) \\
\text{Text11.Text} = (Q3) \\
\text{Shape1.Visible} = \text{True} \\
\text{Text16.Text} = "" \\
\text{Text16.Text} = \text{Ej} \\
\text{Label47.Visible} = \text{True} \\
\text{Label32.Visible} = \text{True} \\
\text{Text17.Visible} = \text{True}
\]

Exit For

End If

End If

Next Ej

End If

---

*When user gives value of ej, the following code executes*

*In case the user needs greater accuracy*

If \( \text{Len(Text16.Text)} > 0 \) Then

\[
\text{Ej} = \text{Val(Text16.Text)} \\
\text{HL1} = \text{Ej} - \text{Ej} \\
\text{HL2} = \text{E2} - \text{Ej} \\
\text{HL3} = \text{Ej} - \text{E3}
\]

If Option1.Value = True Then

\[ g = 9.8 \]

Elseif Option2.Value = True Then

\[ g = 32.2 \]

*To find the velocities and discharge through the Darcy's law*

\[
X1 = (2 \times g \times D1 \times \text{HL1}) \\
X2 = (2 \times g \times D2 \times \text{HL2}) \\
X3 = (2 \times g \times D3 \times \text{HL3})
\]

If \( X1 < 0 \) Then

\[ X1 = -1 \times X1 \]

Elseif \( X2 < 0 \) Then

\[ X2 = -1 \times X2 \]

Elseif \( X3 < 0 \) Then

\[ X3 = -1 \times X3 \]

End If

\[
\begin{align*}
\text{v1} &= (\text{Sqr}(X1)) / (\text{Sqr}(f1 * L1)) \\
\text{v2} &= (\text{Sqr}(X2)) / (\text{Sqr}(f2 * L2)) \\
\text{v3} &= (\text{Sqr}(X3)) / (\text{Sqr}(f3 * L3)) \\
\text{a1} &= (3.14159 / 4) \times (D1 ^ 2) \\
\text{a2} &= (3.14159 / 4) \times (D2 ^ 2) \\
\text{a3} &= (3.14159 / 4) \times (D3 ^ 2)
\end{align*}
\]

\[
\begin{align*}
\text{Q1} &= \text{v1} \times \text{a1} \\
\text{Q2} &= \text{v2} \times \text{a2} \\
\text{Q3} &= \text{v3} \times \text{a3}
\end{align*}
\]

If \( Q3 > Q1 + Q2 \) Then

\[
\begin{align*}
\text{S1} &= Q1 + Q2 + 2 \\
\text{S2} &= Q1 + Q2 - 2 \\
\text{Q3} &= \text{v3} \times \text{a3}
\end{align*}
\]

If \( Q3 < S1 \) And \( Q3 > S2 \) Then

\[
\begin{align*}
\text{Text1.Text} &= (Q1) \\
\text{Text6.Text} &= (Q2) \\
\text{Text11.Text} &= (Q3) \\
\text{Text16.Text} &= "" \\
\text{Text16.Text} &= \text{Ej} \\
\text{Shape1.Visible} &= \text{True} \\
\text{Label47.Visible} &= \text{True} \\
\text{Label32.Visible} &= \text{True} \\
\text{Text17.Visible} &= \text{True}
\end{align*}
\]

Exit For

End If

End If
'To check the direction of flow and to find the applicable continuity equation

If Q1 > Q2 + Q3 Then
   S1 = Q2 + Q3 + 2
   S2 = Q2 + Q3 - 2
   Q1 = v1 * a1
If Q1 < S1 And Q1 > S2 Then
End If
   X1 = (2 * g * D1 * HL1)
   X2 = (2 * g * D2 * HL2)
   X3 = (2 * g * D3 * HL3)
If X1 < 0 Then
   X1 = -1 * X1
Elseif X2 < 0 Then
   X2 = -1 * X2
Elseif X3 < 0 Then
   X3 = -1 * X3
End If
   v1 = (Sqr(X1)) / Sqr(f1 * L1)
   v2 = (Sqr(X2)) / Sqr(f2 * L2)
   v3 = (Sqr(X3)) / Sqr(f3 * L3)
a1 = ((pi / 4) * (D1 ^ 2))
a2 = ((pi / 4) * (D2 ^ 2))
a3 = ((pi / 4) * (D3 ^ 2))
Q1 = v1 * a1
Q2 = v2 * a2
Q3 = v3 * a3
Text1.Text = Q1
Text6.Text = Q2
Text11.Text = Q3
Shape1.Visible = True
Label47.Visible = True
Label32.Visible = True
Text17.Visible = True
End If

Private Sub CmdEnd_Click()
   End
   End Sub

Private Sub Command4_Click()
   Text1.Text = ""
   Text6.Text = ""
   Text11.Text = ""
   Text16.Text = ""
   Text17.Text = ""
End Sub

Private Sub Form_Load()
   OptEnglish.Value = False
   OptMetric.Value = False
End Sub

Private Sub OptEnglish_Click()
   OptEnglish.Value = True
   OptMetric.Value = False
   Text1.Text = ""
   Text6.Text = ""
   Text11.Text = ""
   Text16.Text = ""
   For i = 0 To 9
      Label37(i).Caption = "ft"
   Next
   Label33.Caption = "cubic feet/sec"
   Label34.Caption = "cubic feet/sec"
   Label35.Caption = "cubic feet/sec"
End Sub
Private Sub OptMetric_Click()
    OptMetric.Value = True
    OptEnglish.Value = False
    Text1.Text = ""
    Text6.Text = ""
    Text11.Text = ""
    Text16.Text = ""
    For i = 0 To 9
        Label37(i).Caption = “m”
    Next
    Label33.Caption = “cubic metre/sec”
    Label34.Caption = “cubic metre/sec”
    Label35.Caption = “cubic metre/sec”
End Sub

Private Sub Text16_Change()
    Text17.Text = ""
End Sub

********************************************************************************

Private Sub Text1_Click()
If Text17.Text = "" Then
    Text17.Text = Text1.Text
Else
    X = Val(Text17.Text) + Val(Text1.Text)
    Text17.Text = Text17.Text + "+" + Text1.Text + "=" + Str(X)
End If
End Sub

********************************************************************************

Private Sub Text11_Click()
If Text17.Text = "" Then
    Text17.Text = Text11.Text
Else
    Y = Val(Text17.Text) + Val(Text11.Text)
End If
End Sub

********************************************************************************

RESULTS

The 3RP determines the flow through each pipe, the check being the fact that the flow entering the junction must be equal to the flow leaving it, so the output is displayed in four text boxes, see Fig. 2, three shows the required discharge, the fourth one being the elevation of the junction box. As the answers can be rechecked by addition of two discharge values, so a calculator has been installed at the bottom of the form to facilitate the user.

CONCLUSION

The basic problem is that a scientist could just examine the fundamental Fluid Mechanics associated with frictional loss of energy in pipe flow. From this point onwards an engineer has to take the charge who has to apply these fundamentals to various types of complex practical pipe flow problems in the shortest possible time with greatest possible accuracy. The best solution to this problem is computer. Programming takes time and care, but once
setup, there is great flexibility and many man-hours of labor can be saved.

REFERENCES


4. Greg Perry. Teach yourself Visual Basic 5 in 24 hours Sams Publishing