Parametric Studies on a Mitchell-Ossberger Turbine (Numerical Approach)

Ismaila O Alabi[1], David B Oke[1], Ademola A Dare[2]

[1]Mechanical Engineering Department, The Polytechnic Ibadan, Nigeria
[2]Mechanical Engineering Department, University of Ibadan, Nigeria

Abstract The parametric studies play important role when it comes to performance optimization of the locally manufactured Mitchell-Ossberger Turbine (MOT). Several literatures on hydro-turbine had discussed extensively on the design of MOTs as well as their advantages over the other types of hydro-turbines. But none had ever discussed extensively, the parametric studies involved in optimizing an efficient MOT. This present study focused on some selected parameter towards designing an efficient and optimal MOT. This was achieved via formulation of a turbine parametric model, development of turbine computational interface (module) and decoding of the obtained data with Visual Basic (VB) 10. The algorithm, which contains step by step solution to the problem, was developed for the program. The flow formed was designed to have a data grid view which displays data loaded into the application. The flow interface had eight buttons based on certain parametric entries. The strength of this interface lied in its codes and after data which were sorted on descending order has been loaded to the memory via Visual Basic programming software. It was discovered from the obtained results that as the angle of attack increases, the turbine power developed and the efficiency decreases marginally. The effect of the flow rate was that, it increases as other parameter increases. Then, all necessary recommendations were made.

Keywords MOT, Parametric studies, Computational interface, VB

Introduction

Mitchell-Ossberger Turbine (MOT) or Cross-Flow Turbine (CFT) is a special type of impulse turbine that transform water potential to mechanical energy. Although it is technically classified as an Impulse turbine [1-4] because it is not entirely immersed in water. It is used in low-head, high-flow systems. MOT is known as wide range of heads overlapping those of Kaplan, Francis and Pelton. It can operate with heads between 5 and 200 m. It allows the water to pass through the runner and crosses it two times before leaving the turbine. Its simplicity in design makes it cheap and easy to repair in case of runner brakes due to the important mechanical stresses [5]. However, several literatures on hydro-turbine have discussed the design and construction of a MOT as well as their advantages over the other types of hydro-turbines. But none had ever discussed extensively, the parametric studies involved in optimizing an efficient MOT design. This present study focuses on some selected parameter towards designing an efficient MOT hydro turbine via MOT-model formulation, parametric analysis for optimal performance, as well as development of computational interface with Visual Basic 10 programming software. There are many different types of turbines, and proper selection requires considerable expertise. A pelton design, for example, works best with high head. Across flow design works better with low Head but high flow. Likewise, other turbine types such as Francis, Turgo and Kaplan, each have optimum applications.

Basically, the two methods which help in selection of turbine are as follows:

  A. **Thumb Rule:** Different types of turbines can be selected to best suit given head and flow conditions. Figure 1 shows the typical application ranges of various turbines.
  B. **Scientific Method:** Turbines have been mainly selected scientifically based on the specific speed. One of the important parameters of a turbine is the Specific Speed denoted as n_s, which is defined as the speed in rpm at which a turbine of homologous design would operate, if the runner were to reduce to a size which would develop one metric horse power under one meter head. It is given by the following relation:

$$n_s = \frac{n\sqrt{P*1.358}}{H^{5/4}}$$

Where, n_s = Specific speed of turbine in revolutions per minute (r.p.m.)
n = Rated speed of turbine in revolutions per minute
P = turbine output in kW, and
H = Rated head in meters

Once the specific speed (n_s) is determined, the chart given in Figure 1(b) may be used to determine the type of turbine that may be adopted for the particular work.
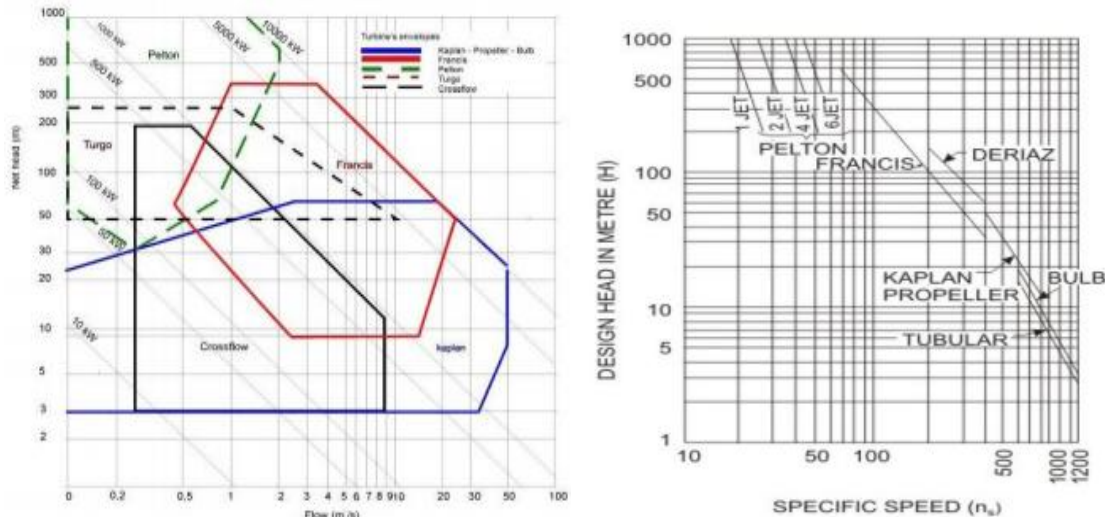

Figure 1: (a) The Thumb Rule   (b) Scientific Method Chart [6-7]

According to literatures, various charts have been developed to aid the selection of a suitable turbine for varying site conditions of the flow head. However, the two selection charts above are limited to four major parameters, namely; Flow rate (Q), Head (H), Specific Speed (Ns) and Power (P), as discussed in the relevant literatures. This present study takes into account other parameters for designing an efficient Mitchell-Ossberger Turbine in addition to the existing parameters.

Materials and Method

Modules and processes were developed for hydro-turbine computational purposes. The paper followed the general way of developing applications in Visual Basic express edition. This present study utilized the approach adopted to develop a computational interface for a Mitchell-Ossberger Turbine using Visual Basic 10 programming software [8]. The process is as follows:

  i.   **Planning**: It is at this stage that the conceptual view of the program is gotten, its structure, target users and ultimately the purpose for which the program was written.
  ii. **Mathematical modeling**: it involves the gathering of parameters that to be used in programming the MOT for generating useful power.
  iii. **Numerical Solution**: the method for solving those parameters involved is basically of numerical approach or method.
  iv. **Creating the project:** this is the creation of all files necessary for the application which is known as the project and when more than one is referred to, it is known as solution.
  v.   **User interface Design:** it involves dragging various controls onto the design surface or form. Then the properties that define the appearance and behavior of the form and the contents are set.
  vi. **Coding:** this involves writing of the visual basic code that defines how the application behaves and interacts with the user.
  vii. **Debugging:** it is possible to have bugs in the application, bugs are errors that disallow the application from producing desired results, it is from this stage that the application is tested and errors are removed or debugging.
  viii. **Compilation:** The program is then compiled in an executable format (.exe) extension which runs on windows operating system.

Description of Program Interface
There is a need for a deeper design understanding than the architectural design and configuration specification as it contains the procedural methods with which the inputs are transformed to output. The algorithm for the program is also developed, which contains step by step solution to the problem.

The Flow Interface
The flow form was designed to have a data grid view which displays data loaded into the application. The flow interface has eight buttons. The first three buttons are for parametric entry, where the selected parameters were computed as entered. The fourth button is for material selection (WELDED STEEL and PVC only). The fifth and sixth buttons were used for table selection and independent parameter generation while the seventh and eight buttons were used for graph plotting selection. The strength of this interface lies in its codes and after data has been loaded to the memory, they are sorted in descending order using the Visual Basic Method, and a rank is given to each value, the ranking was programmed to give the same rank to equal values and skip the next rank depending on how many values are equal.
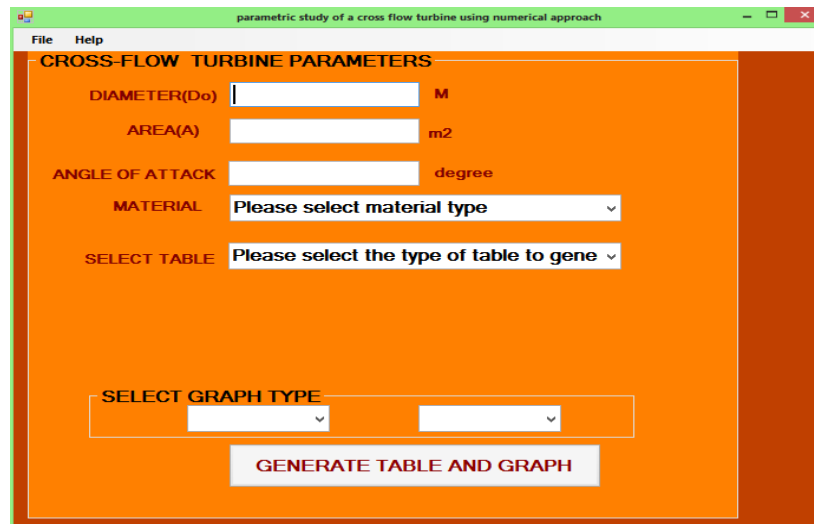
Figure 2: The Mitchell-Ossberger or Cross-flow Turbine Computational Interface

The Turbine Graph Selection Interface
This interface is responsible for accepting the user input parameters, it has eight buttons, the first button is for diameter selection, the second button for area selection, the third button for angle of attack selection, the forth button for material selection, the fifth button for table selection and the sixth and seventh button for graph selection while the last button is for table and graph generation command.

Results and Discussion
The results obtained from the data generated are presented and discussed below:

Results
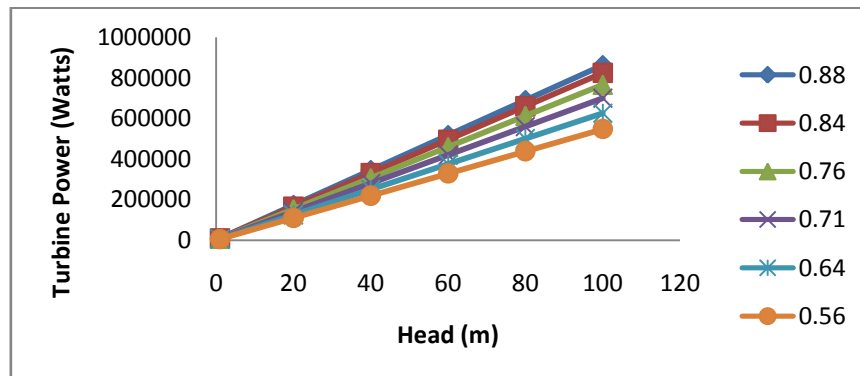**Effects of Cross-flow Selected Design Parameters on Turbine Efficiency**

Figure 3: Plot of Turbine Power against Head

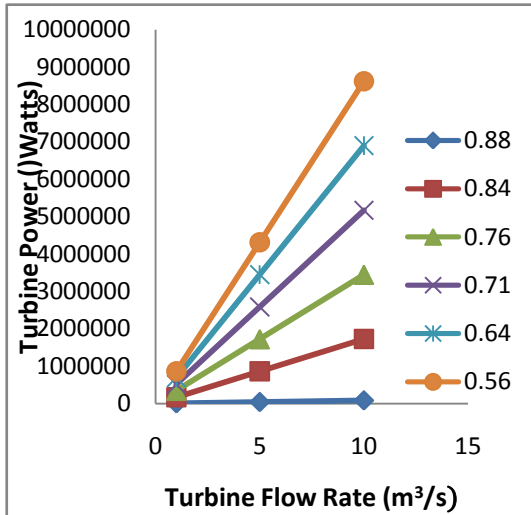**Effects of Turbine Power and Flow Rate on Cross-flow Turbine Efficiency at 16-45⁰ Angle of Attack**
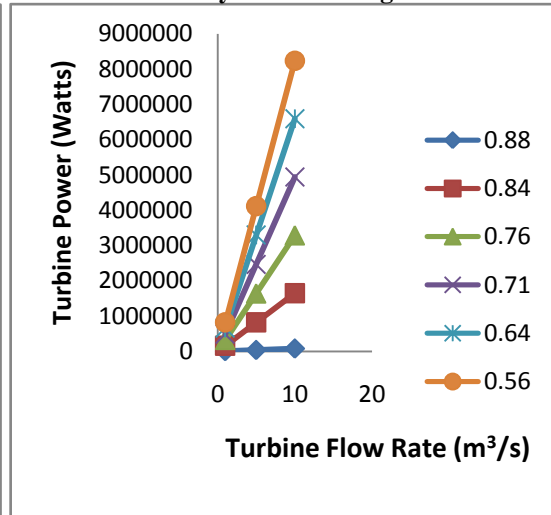

Figure 4: Angle of Attack at 16⁰

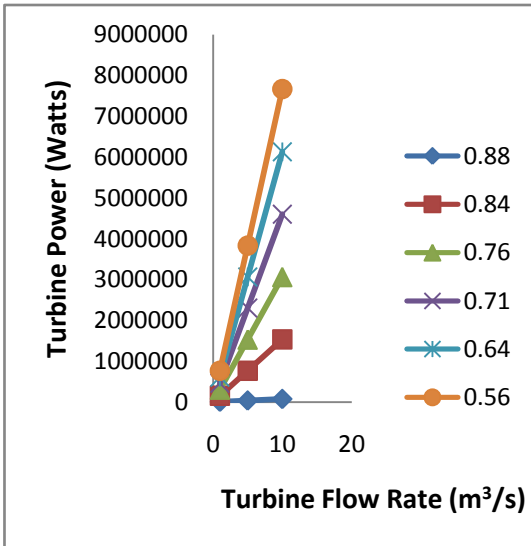

Figure 5: Angle of Attack at 20⁰
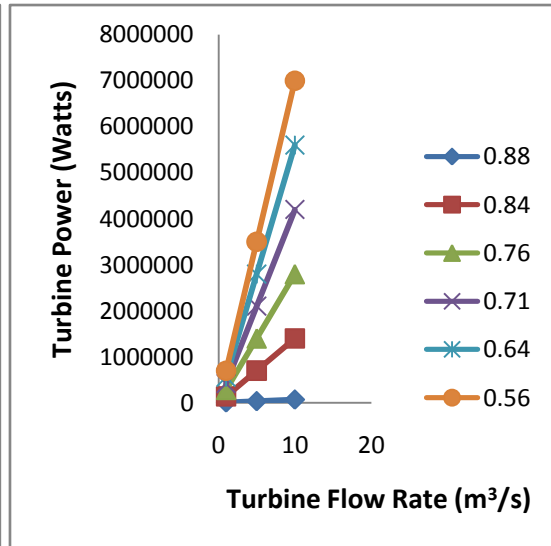

Figure 6: Angle of Attack at 25⁰

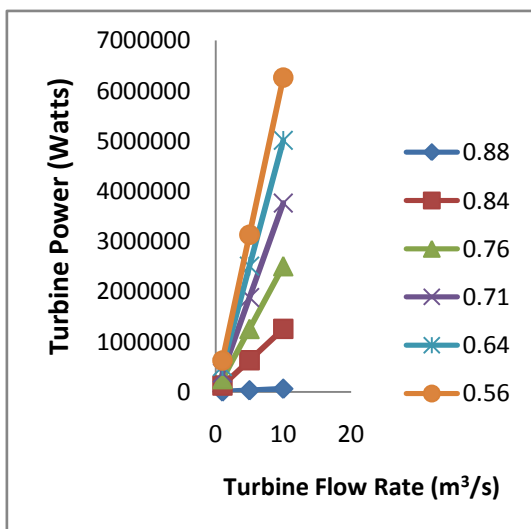*Figure 7: Angle of Attack at 30⁰*

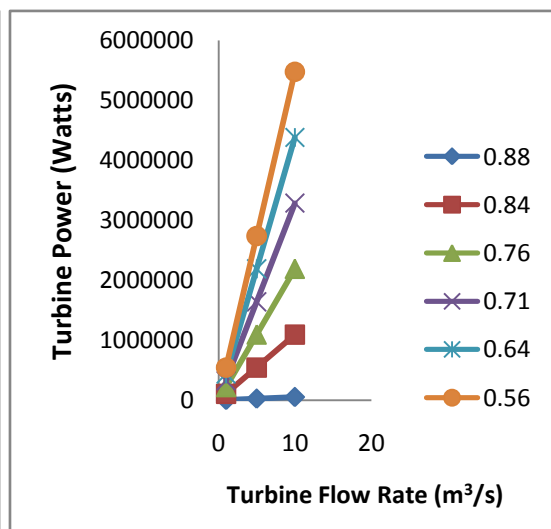*Figure 8: Angle of Attack at 35⁰*

Figure 9: Angle of Attack at 40⁰

Discussion

Fig. 3 showed that the turbine power is directly proportional to the head, as the turbine power increases, the head also increases. This is an indication that the higher the selected parameters, the higher the efficiency. Fig. 4 showed that as the turbine power increases, the flow rate also increases. The higher the flow rate, the lower the efficiency at an angle of attack of 16^0. Fig. 5 to 8 produced results similar to fig. 4. That is, as the turbine power increases, the flow rate also increases. As the flow rate increase, the efficiency drops. In general, the turbine power decreases as the angle of attack increases and efficiency drops. The maximum efficiency was obtained at a 16^0 Angle of Attack. The flow rate increases as the turbine power increases.
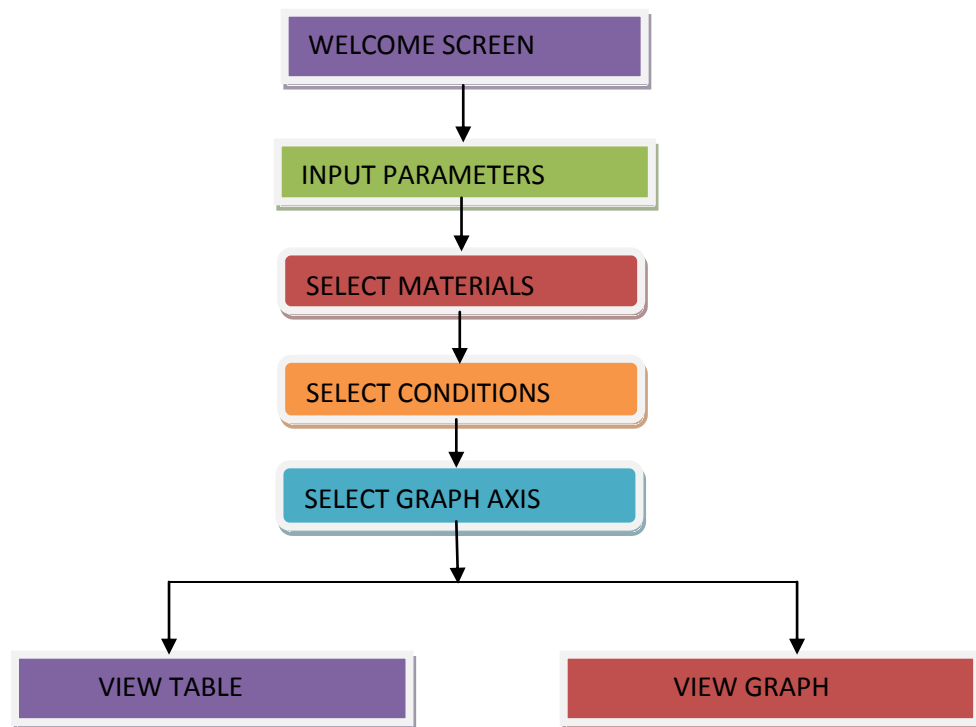
Conclusion

The parametric studies in the MOT and improvement in various parameters should not be underestimated when it comes to performance optimization of the locally manufactured MOT. This present study applied various parameters needed in designing a cross flow hydro turbine optimally via the stated objectives using a Visual basic codes. The visual basic program was also used to develop an interface to validate the experimental results already obtained in the literature [9]. It was discovered from the obtained results that as the angle of attack increases, the turbine power developed and the efficiency decreases marginally. The effect of the flow rate was that, it increases as other parameter increases. Thus, parametric studies would help in predicting the expected output of OMT in the hydropower plant. For existing sites that are not performing optimally, the parameters can be checked via this OMT interface to study why there is drop in output and how best to boost the output. It presents a fast and more accurate means of design calculation when compared with manual estimation which on the other hand involves dealing with large figures that can easily induce human error.

References

[1]. Barglazan, M. (2005) "Design optimization of cross-flow hydraulic turbine", Scientific bulletien of the Politechnica, University Timisoaro, Volume 50(64). 24-28

[2]. Felix M, Ramadhani, W.A, Towo, S.K., Makhanu, O. M. & Binyam, A (2010). 'Design and Fabrication of Cross Flow Turbine'. NBCBN, Local Action Research-Tanzania: 1-15

[3]. Chiyembekezo S. K, Cuthbert Z. K & Torbjorn K. N (2014$_a$) Experimental study on a simplified crossflow turbine. International Journal of Energy And Environment. Volume 5, Issue 2:155-182.

[4]. Chiyembekezo S. K, Cuthbert Z. K & Torbjorn K. N (2014$_b$). A numerical investigation of flow profile and performance of a low cost Crossflow turbine. International Journal of Energy and Environment. Volume 5, Issue 3: 275-296.

[5]. Tamil, A.C, Anil, G. & Chandapillai, J (2010). Development and Testing of A Cross Flow Turbine. IGHEM-AHEC, IIT Roorkee, India: 1-8

[6]. Saurabh S, Arpit G, & Dinesh K (2013). Review of Optimal Selection of Turbines for Hydroelectric Projects. International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 3: 424-430

[7]. Dele, I.S, Isaiah, A.A, Olufriopo, S.A, & Adio T.A (2014). An Assessment of The Small Hydro Potential of Opeki River, Southwestern Nigeria. Science Journal of Energy Engineering; 2(3): 25-31.

[8]. Ajao K.R, Olabode O.F. & Sule O. (2014). Development of a Computational Interface for Hydropower Plants. Sustainable Energy, 2014, Vol. 2, No. 2: 63-80.

[9]. Alabi, I.O. (2015). Design and Construction of A 20kW Low Head Cross-flow Turbine. An unpublished M.Sc. Thesis Submitted to the Department of Mechanical Engineering, University of Ibadan, Nigeria. March, 2015. 1-50.

APPENDIX A

```
WELCOME SCREEN
      ↓
INPUT PARAMETERS
      ↓
SELECT MATERIALS
      ↓
SELECT CONDITIONS
      ↓
SELECT GRAPH AXIS
   ↓         ↓
VIEW TABLE   VIEW GRAPH
```

Appendix B
Public Class Form6
Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Dim one, against, pi, d0, n, Hn, A, Q, g, density, alpha, c, file, Hg, Hh, eff, dp, L, V, tg, Pt, tj, Ns, Sb, tb, nb, rb, di, ds As Double
Dim one1, against1 As String
If Val (TextBox4.Text) = 0 Then
MsgBox ("PLEASE ENTER THE RANGE FOR DISHARGE RATE")
ElseIf Val (TextBox6.Text) > 0 Then
MsgBox ("MAX FLOW RATE CANNOT EXCEED 10metre3/sec")
ElseIf Val (TextBox7.Text) > 200 Then
MsgBox ("MAX HEADLOSS CANNOT EXCEED 200m")
ElseIf Val (TextBox3.Text) = 0 Then
MsgBox ("PLEASE ENTER A VALUE FOR THE AREA")
ElseIf Val (TextBox5.Text) = 0 Then
MsgBox ("PLEASE ENTER A VALUE FOR THE ANGLE OF ATTACK")
Else
If ComboBox2.SelectedIndex = 0 Then ' AT CONSTANT FLOW RATE '
ComboBox2.SelectedIndex = 0
If Val (TextBox7.Text) = 0 Or Val(TextBox4.Text) = 0 Then
MsgBox ("Please select a range for Headloss")
Else
For Hn = Val (TextBox4.Text) To Val(TextBox7.Text)
Q = Val (TextBox1.Text)
d0 = Val (TextBox2.Text)
A = Val (TextBox3.Text)
alpha = Val(TextBox5.Text)
If ComboBox1.SelectedIndex = 0 Then
n = 0.012
Else

Journal of Scientific and Engineering Research

```
n = 0.009
End If
pi = Math.Atan(1) * 4
tb = (0.174 * d0)
g = 9.81
Density = 9810
c = 0.98
file = 0.98
Hg = 1.06 * Hn
Hh = 0.06 * Hg
eff = 0.5 * (c ^ 2) * (1 + file) * (Math.Cos((file) * (pi / 180))) ^ 2
V = Q / A
L = ((g * Hn * tg) / V)
dp = 2.69 * (((n ^ 2 * Q ^ 2 * L) / (Hh)) ^ 0.1875)
tg = 9.81
Pt = density * g * eff * Hn * Q
tj = 0.233 * Q / (L * ((Hn) ^ 0.5))
Ns = (513.25 * (Hn ^ 0.745)) / (Pt ^ 0.5)
nb = 3.142 * (d0 / tb)
Sb = (3.142 * d0 - nb * tb) / nb
rb = 0.163 * d0
di = d0 - 2 * A
ds = (0.222 * d0)
'code to update the tble lies here now'
Form2.Show ()
Form2.DataGridView1.Rows.Add(Math.Round(Q, 0), Math.Round(Hn, 0), Math.Round(Hg, 0),
Math.Round(Hh, 0), Math.Round(eff, 0), Math.Round(dp, 0), Math.Round(L, 0), Math.Round(Pt, 0),
Math.Round(tj, 0), Math.Round(Ns, 0), Math.Round(d0, 0), Math.Round(di, 0), Math.Round(tb, 0),
Math.Round(Sb, 0), Math.Round(nb, 0), Math.Round(rb, 0), Math.Round(ds, 0))
'code to update chart lies here too'
If ComboBox3.SelectedIndex = 0 Then
one = eff
one1 = "EFFICIENCY"
ElseIf ComboBox3.SelectedIndex = 1 Then
one = Ns
one1 = "SPECIFIC SPEED"
ElseIf ComboBox3.SelectedIndex = 2 Then
one = Hn
one1 = "HEADLOSS"
ElseIf ComboBox3.SelectedIndex = 3 Then
one = Q
one1 = "DISCHARGE RATE"
ElseIf ComboBox3.SelectedIndex = 4 Then
one = Pt
one1 = "TURBINE POWER"
End If
If ComboBox4.SelectedIndex = 0 Then
against = eff
against1 = "EFFICIENCY"
ElseIf ComboBox4.SelectedIndex = 1 Then
against = Ns
against1 = "SPECIFI SPEED"
ElseIf ComboBox4.SelectedIndex = 2 Then
against = Hn
against1 = "HEADLOSS"
ElseIf ComboBox4.SelectedIndex = 3 Then
against = Q
against1 = "DISCHARGE RATE|"
```

```
ElseIf ComboBox4.SelectedIndex = 4 Then
against = Pt
against1 = "Turbinepower"
End If
Form5.Text = "GRAPH OF" + one1 + "   " + "against" + "   " + against1
Form5.Show()
Form5.BringToFront()
Form5.Chart1.Series("the result").Points.Add(one, against)
Next
End If
ElseIf ComboBox2.SelectedIndex = 1 Then 'at constant head loss'
If Val(TextBox1.Text) = 0 Or Val(TextBox6.Text) = 0 Then
MsgBox("Please select a range for FLOW RATE")
Else
For Q = Val(TextBox1.Text) To Val(TextBox6.Text)
Hn = Val(TextBox4.Text)
d0 = Val(TextBox2.Text)
A = Val(TextBox3.Text)
alpha = Val(TextBox5.Text)
If ComboBox1.SelectedIndex = 0 Then
n = 0.012
ElseIf ComboBox1.SelectedIndex = 1 Then
n = 0.009
End If
pi = Math.Atan(1) * 4
tb = 0.174 * d0
g = 9.81
density = 9810
c = 0.98
file = 0.98
Hg = 1.06 * Hn
Hh = 0.06 * Hg
eff = 0.5 * (c ^ 2) * (1 + file) * (Math.Cos((file) * (pi / 180))) ^ 2
V = Q / A
L = ((g * Hn * tg) / V)
dp = 2.69 * (((n ^ 2 * Q ^ 2 * L) / (Hh)) ^ 0.1875)
tg = 9.81
Pt = density * g * eff * Hn * Q
tj = 0.233 * Q / (L * ((Hn) ^ 0.5))
Ns = (513.25 * (Hn ^ 0.745)) / (Pt ^ 0.5)
nb = 3.142 * (d0 / tb)
Sb = (3.142 * d0 - nb * tb) / nb
rb = 0.163 * d0
di = d0 - 2 * A
ds = (0.222 * d0)
'code to update the tble lies here now'
Form3.Show()
Form3.DataGridView1.Rows.Add(Math.Round(Q, 0), Math.Round(eff, 0), Math.Round(dp, 0), Math.Round(L,
0), Math.Round(Pt, 0), Math.Round(tj, 0), Math.Round(Ns, 0), Math.Round(d0, 0), Math.Round(di, 0),
Math.Round(tb, 0), Math.Round(rb, 0), Math.Round(ds, 0), Math.Round(nb, 0))
'code to update chart lies here too'
If ComboBox3.SelectedIndex = 0 Then
one = eff
one1 = "EFFICIENCY"
ElseIf ComboBox3.SelectedIndex = 1 Then
one = Ns
one1 = "SPECIFI SPEED"
ElseIf ComboBox3.SelectedIndex = 2 Then
```

```
one = Hn
one1 = "HEADLOSS"
ElseIf ComboBox3.SelectedIndex = 3 Then
one = Q
one1 = "DISCHARGE RATE"
ElseIf ComboBox3.SelectedIndex = 4 Then
one = Pt
one1 = "TURBINE POWER"
End If
If ComboBox4.SelectedIndex = 0 Then
against = eff
against1 = "EFFICIENCY"
ElseIf ComboBox4.SelectedIndex = 1 Then
against = Ns
against1 = "SPECIFI SPEED"
ElseIf ComboBox4.SelectedIndex = 2 Then
against = Hn
against1 = "HEADLOSS"
ElseIf ComboBox4.SelectedIndex = 3 Then
against = Q
against1 = "DISCHARGE RATE|"
ElseIf ComboBox4.SelectedIndex = 4 Then
against = Pt
against1 = "Turbinepower"
End If
Form5.Text = "GRAPH OF" + "   " + one1 + "against" + "   " + against1
Form5.Show()
Form5.BringToFront()
Form5.Chart1.Series("the result").Points.Add(one, against)
Next
End If
End If
End If
End Sub
Private Sub Form6_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
GroupBox2.Hide()
GroupBox3.Hide()
ComboBox1.Text = "Please select material type"
ComboBox2.Text = "Please select the type of table to generate"
End Sub
Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
End Sub
Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox2.SelectedIndexChanged
If ComboBox2.SelectedIndex = 0 Then ' AT CONSTANT FLOW RATE'
GroupBox2.Show()
GroupBox3.Show()
TextBox6.Hide()
Label13.Hide()
TextBox7.Show()
Label14.Show()
ElseIf ComboBox2.SelectedIndex = 1 Then
GroupBox2.Show()
GroupBox3.Show()
TextBox7.Hide()
Label14.Hide()
```

```
TextBox6.Show()
Label13.Show()
End If
End Sub
Private Sub GroupBox1_Enter (ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
GroupBox1.Enter
End Sub
End Class
```