

Modeling of Water and Wastewater Treatment Units

Hisham I. M. Abdel-Magid ¹, Yousif Ali Yousif ² and Elsadig Elhadi Elhassan ³

¹ Enginormatics Engineering Ltd.,
London, Postal Code N1 7GU, United Kingdom (hisham@enginormatics.com)

² School of Civil Engineering, Sudan University of Science and Technology,
Khartoum, P.O. Box 407, Sudan (yousifali@sustech.edu)

³ Secretariat of Academic Affairs, Sudan University of Science and Technology,
Khartoum, P.O. Box 407, Sudan (alsadigalhadi@sustech.edu)

Abstract

This paper outlines the significance of computer modeling as a vital part of the practice of environmental engineering. To meet this goal, the examples herein were constructed as computer software applications that demonstrate how to transform an environmental engineering problem into a working computer model that can be manipulated and edited. The models were written using Microsoft Visual Basic.NET 10 programming language. The Integrated Development Environment (IDE) used was Microsoft Visual Studio 2010 Professional Edition. Examples were test-ran under Microsoft Windows XP and 7 workstations, although the pre-compiled code should run on other platforms with .NET 4.x support, such as Linux with MonoDevelop installed.

Keywords: *Modelling, Water, Wastewater, Treatment, Development, VB.NET*

1. Preamble

It goes without saying that computers are dominating the world of information technology. What started as simple ideas of enhancing manual computations and aiding in math solving, has become the most complex, interconnected, multi-layered system of all times. Computers nowadays serve all sorts of functions: some are general every-day-purpose kind of things (writing your letters with a word processor, doing some tax calculations with an electronic calculator, playing video games, ...), while others are more specific to certain environments and specialized in purpose (a database server saving customers' data, a managerial program running a hospital management system, an autopilot system driving a combat airplane, an embedded Linux-derived kernel in your Android smartphone, ...), and those lists are far from exhaustive.

Using computers in engineering modeling is one of the recipes for being a successful engineer, traditional manual computations and hand-writing modular calculations are all being lost to the oblivion. There are multiple computer

modular designs, with Differences in how they are being implemented. Those implementations vary according to more than one factor such as [1]:

1. The underlying programming language used. Some modules are general in purpose, and as such are programmed using a general purpose language (such as Visual Basic, C/C++, Java, etc ...). Others are specialized in their working, necessitating a specialized programming language to be used (such as Prolog for artificial intelligence software, PHP for website programming, SQL for database access, ...).
2. The computer platform for target execution (most commonly used in market today being the Intel-compatible x86 '32-bit' or AMD64 '64-bit' architectures).
3. The operating system environment (again, common OSes in the market including – but not limited to – MS-Windows XP/7/8, Linux with all its flavors – Fedora, Mandriva, Gentoo, SuSE, among others; MAC OS, UNIX, Solaris, and much more)
4. And above all, the engineering principles that form the backbone for modular design.

The time for flat programs (that run in a linear fashion from the first instruction to the last) has been gone long ago. Programs are being multilayered, event-driven in nature. That means the program will need to be segmented into parts, each part dealing with (or responding to) a certain event. That is usually (but not necessarily) a part of an object-oriented programming (OOP) design. Today's programs usually are written for GUI (Graphical User Interface) environments, but this is not a necessity. Indeed, sometimes having the overhead of calling graphical routines lifted off the program presents a huge advantage in program's speed and execution, that the GUI is simply omitted, and the program is written as a text-based (or console) program (If you ever went to book a flight and saw your travel agent working that ugly black screen, then

you get it). Some programming languages are text-based by nature (such as Prolog), unless mixed with other languages, or being used with an extension (such as Visual Prolog extension for Prolog) [2].

In this work, all programs were written as GUI programs for windowing systems. As such, the programs have two parts: the GUI design (what the user sees), and the code doing the work behind the scenes. The code for each program is included in the text along each example presented. The code provided goes into the main program window source file (named Form1.vb by default). For the multi-windowed example, the sources are for Form1.vb, Form2.vb, and so on.

The programming code was designed to be at the minimum level of complexity and length needed to perform the task at hand, at the same time not to be too short, scrambled, or ambiguous, so that new and beginner programmers can follow without stumbling at each line. For each task performed, especially if a function performs a lengthy operation, or a specific programming concept deemed complex or advanced, the reader will find prompt documentation through the source code in the relevant parts to make following the code an easy and productive task. Visual Basic.NET was selected as a programming language for the work's examples for many reasons:

- MS-Windows is one of the most (if not the most) popular operating system in use today. Chances are, when buying a new PC or laptop (or even a mobile phone), the user will face this OS first and foremost.
- .NET Fx (pronounced "dot net framework") is becoming so popular as a dominant programming platform in computer industry, that it will be counterproductive not to learn and understand how to implement engineering models using this platform.
- Visual Basic.NET, which is a descendent of the old beginners' friend BASIC language, is holding its name as being an easy-to-learn language. It may be handicapped with its capabilities being less than other more sophisticated high-level languages, but the more important thing is that the language has so many layers of abstraction inserted between the programmer and the machine. That means the programmer doesn't need to bother about the low-level workings of the computer, he just focuses on programming. For this, BASIC, and its descendent VB, which then evolved to VB.NET, is considered one of the easy programming languages to learn, especially for beginners. And this is why it was the perfect choice for this text.

The basic idea behind the programs is to explain how to transform certain engineering problems into valid programming concepts and working computer modules.

That means the programs are easy to read and transform into another programming language. They can also be reprogrammed using another language under another platform other than Windows (for example using Qt/C++ to program under Linux).

Aims and objectives of the work undertaken incorporate the following:

- Develop computer programs that aid design of treatment units and final disposal.
- Advocate rational approaches towards sustainable wastewater management via pollution prevention, appropriate treatment, resource recovery and re-use at both centralized and decentralized levels.
- Relate relevant physical, chemical and biological processes and their mutual relationships within various sanitation and waste stream components.
- Evaluate various urban drainage and sewerage schemes, wastewater, sludge and solid waste treatment technologies.
- Promote environmental protection, improvement and sustainability, enhancing the overall quality of life.
- Demonstrate effective problem-solving abilities
- Utilize current technology of waste collection, handling, treatment, final disposal and management.
- Apply appropriate regulations and guidelines in safeguarding environmental conditions.
- Use effective oral and written communication skills.
- Capture ability of reasonable scientific judgment and concepts of appropriate decision making.

The paper significantly supplements, complements and guides to basic principles of environmental and sanitary engineering. Besides, it can be sourced as a standalone problem solving script in environmental engineering. It is estimated to have valuable benefits to students, practicing and professional sanitary and environmental engineers to be a treasured support to student-centered learning. To accomplish such objectives great attention was paid to offer solutions to selected problems in a well-defined, clear and discrete layout exercising step-by-step procedure and explanation of the related solution employing crucial procedures, methods, approaches, equations, data, figures and calculations.

2. Materials and Methods

Generated computer programs targeted water and wastewater treatment units and processes with emphasis on [2-12]:

- Water and wastewater characteristics (of importance are physical, radiation, chemical biological, bacteriological and microbiological. Physical

properties are subject to natural forces making it easier to measure & determine their values & effects. Physical properties of significance include: concentration of solids, turbidity, taste, odor, color, temperature, electrical conductivity, salinity, density, standard volume, viscosity, surface tension, moisture content, humidity, radiation and dissolved oxygen.

- Water Treatment, storage and distribution.
- Wastewater volumes, footprints, collection and transportation
- Wastewater Volumes, Collection and Transportation
- Wastewater Treatment and final disposal.

3. Examples of Design Units

For explanation purposes in this paper, three examples were chosen to reflect the design of Dorr settling tank, a facultative waste stabilization pond and Streeter Phyleps equation for disposal of treated effluents into a water course (if allowed as per governing authority regulations and permits). The aforementioned examples as comprehended in the following sections, whilst Figure 1 illustrates an example of a program screen shot for an additional treatment unit (i.e. a Grit Removal unit).

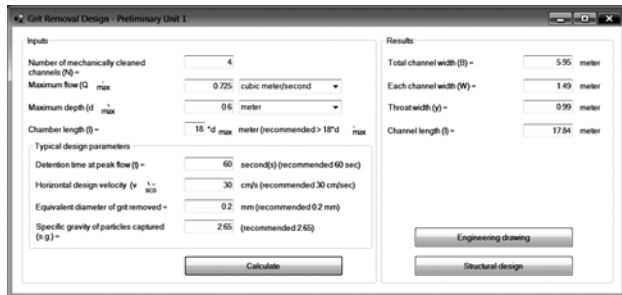


Fig. 1 Screen shot for Grit Removal unit design.

3.1 Circular Settling Tanks (Dorr Settling Tanks)

In tanks with circular shapes, the flow is horizontal. Circular tanks can be used for primary as well as secondary sedimentation purposes. The maximum diameter for circular tanks is considered to be 50 m, with an average diameter of 30 m, and a maximum water depth of 4 m, with an average water depth of 2.5 m. In the design of circular tanks, the following points merit consideration (refer to Figure 2) [14-16].

- Ratio of diameter to clarification depth may be found as follows:
 - $6 \leq D/H \leq 10$ for circular tanks with diameters of 16 to 30 m.

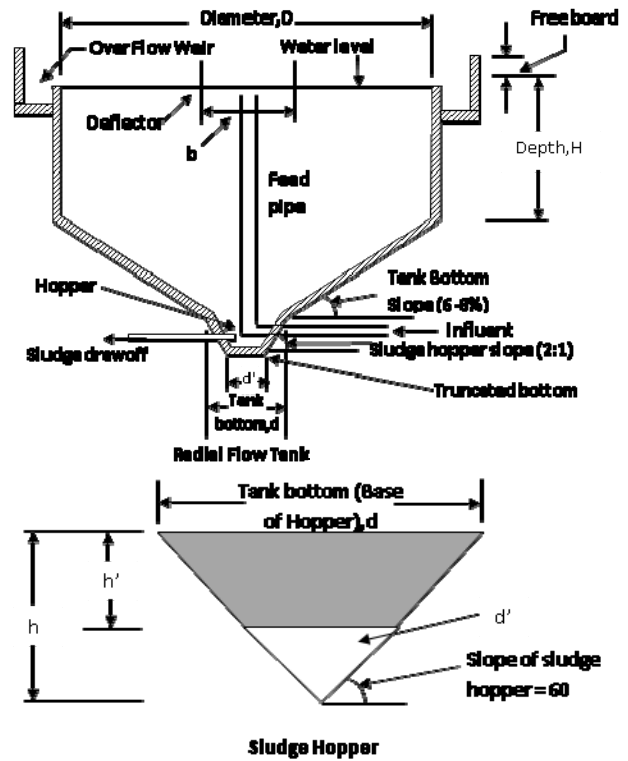


Fig. 2 Radial flow sedimentation tank.

- $16 \leq D/H \leq$ for circular tanks with diameters of 30 to 50 m.
- $20 \leq D/H \leq 25$ for circular tanks serving wastewaters with large amounts of organic matter (high settling velocities).
- Ratio of diameter of circular deflector to diameter of tank may be taken as:

$$b = (10 \text{ to } 20\%) \times D \tag{1}$$
 Where:
 - b = Diameter of circular deflector (may be taken as $b \approx 0.15 \times D$), m.
 - D = Diameter of circular tank, m

Assuming a single weir in tank:

$$\text{surface loading} \times \text{area} = \text{weir loading} \times \text{length}$$

$$\pi (D - b)^2 \times v_s / 4 = \pi \times D \times q_{\text{weir}} \tag{2}$$

Where:

D = Diameter of tank, m

b = Diameter of deflector, m

v_s = Surface loading rate, m/d

q_{weir} = Weir loading (overflow rate), $\text{m}^3/\text{d}/\text{m}$ length

Taking $b = 0.15 \times D$, then:

$$D = 5.54 \times q_{\text{weir}} / v_s \tag{3}$$

- Tank depth may be determined as:

$$H = Q \times t / N \times A \tag{4}$$

Where:

A = Effective area of settlement, m²

$$H = \pi \times (D_a - b)^2 / 4 \quad (5)$$

Where:

N = Number of settling tanks, dimensionless

D_a = Actual diameter of tank, m

Thus,

$$H = 4 \times t \times Q / \pi \times N \times (D_a - b)^2 \quad (6)$$

- Slope of settling tank bottom, β, may be taken as β = 6 to 8° (or 5 to 10°).
- Slope of sludge hopper, α, may be taken as α = 2:1 (≥ 60°).
- Detention time may be taken as 2 h at peak flow.
- Volume of sludge hopper is to hold 12 to 24 h of sludge production, and it may be determined as:

$$V_h = Q \times C_0 \times \text{Eff} / a \times C_c \quad (7)$$

Where:

V_h = Volume of sludge hopper, m³

Q = Flow rate, m³/d

C₀ = Concentration of solids entering sedimentation tank, kg/m³

Eff = Efficiency of solids removal in tank, dimensionless

a = Fraction of flow to be stored in hopper (as sludge)

C_c = Solids content of final sludge, kg/m³
 = (1 – m.c.) × 1000, where m.c.
 = moisture content of sludge (dimensionless).

Considering hopper as a truncated cone, then:

$$V_h = (\pi d^2 \times h / 12) - (\pi d'^2 \times h' / 1) \quad (8)$$

Where:

d = Diameter of hopper base (tank floor), m

h = Hopper imaginary depth (depth of total cone), m

d' = Diameter of truncated cone (hopper floor), m

h' = Depth of truncated cone, m

Taking a slope of sludge hopper as α = 60, then:

$$(d/2)/h = (d'/2)/h' = \tan 30 \quad (9)$$

As such:

$$V_h = 0.227(d^3 - d'^3) \quad (10)$$

- The feed pipe is to be located 20 to 30 cm below water level.

The program targets the hydraulic and structural design of a circular sedimentation tank given the relevant design data of rate of flow, Q (m³/s); loading rate, v_s (m³/h/m²); influent solids concentration, C₀ (kg/m³); tank efficiency, Eff (%); sludge moisture content, m.c. (%); sludge production that can be stored by the sludge hopper; weir loading (overflow rate), q_{weir} (m³/d/m weir length); retention time, t (s) at peak flow; and bottom radius

(truncated radius), d' (m). Verification of program has been done for the following data for designing a circular sedimentation tank with the following information:

- Rate of flow, Q = 15 m³/min.
- Loading rate (settling velocity), v_s = 1.4 m³/h/m²
- Influent solids concentration, C₀ = 250 mg/L
- Tank efficiency, Eff = 55%
- Sludge moisture content, m.c. = 96%
- Size of sludge hopper sufficient to hold 12 h sludge production
- Weir loading (overflow rate), q_{weir} = 200 m³/d/m
- Retention time, t = 2 h at peak flow
- Bottom radius (truncated radius), d' = 0.5 m

Figure 3 represents the logical flowchart for circular sedimentation tank design, whilst Figure 4 illustrates the VB.Net code enlisting of the same.

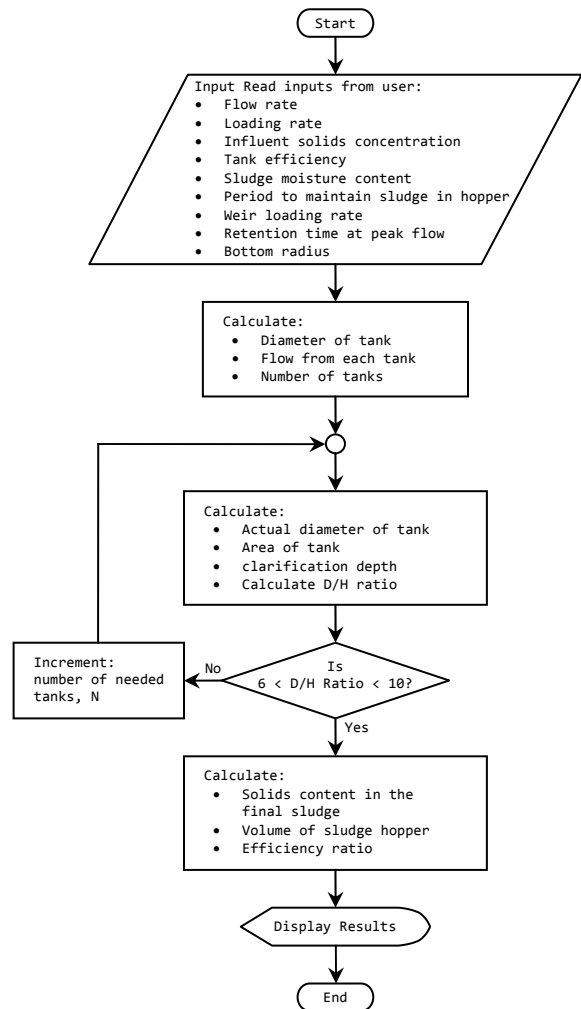


Fig. 3 Flowchart of circular sedimentation tank design.

```
Public Class Form1
    Const pi = 3.142857
    Dim Q, Vs, Co, Eff, mc, tr, Qweir, t, rdash As Double
    Private Sub Form1_Load(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles MyBase.Load
        Me.Text = "Program: Designs a circular sedimentation tank"
        Button1.Text = "&Calculate"
        Label11.Text = "Enter flow rate (m3/min):"
        Label12.Text = "Enter loading rate (m3/hr/m2):"
        Label13.Text = "Enter influent solids concentration (mg/L):"
        Label14.Text = "Enter tank efficiency (%):"
        Label15.Text = "Enter sludge moisture content (%):"
        Label16.Text = "Enter period to maintain sludge in hopper (hr):"
        Label17.Text = "Enter weir loading rate (m3/d/m):"
        Label18.Text = "Enter retention time at peak flow (hr):"
        Label19.Text = "Enter bottom radius (m):"
        GroupBox1.Text = " Output: "
        Label10.Text = "Number of tanks needed:"
        Label111.Text = "Diameter of tank (m):"
        Label112.Text = "Depth of clarification zone (m):"
        Label113.Text = "Diameter of sludge hopper (m):"
    End Sub

    Sub calculateResults()
        Q = Val(TextBox1.Text)
        Vs = Val(TextBox2.Text)
        Co = Val(TextBox3.Text)
        Eff = Val(TextBox4.Text)
        mc = Val(TextBox5.Text)
        tr = Val(TextBox6.Text)
        Qweir = Val(TextBox7.Text)
        t = Val(TextBox8.Text)
        rdash = Val(TextBox9.Text)
        Dim Cc, dh, Vh, D, Q1, N, A, H, rat As Double
        Dim OK, t1 As String
        Vs = Vs * 24 'convert to m/day
        Q = Q * 60 * 24 'convert to m3/day
        Co = Co / 1000 'convert to kg/m3
        t = t / 24 'convert to days
        D = 5.54 * Qweir / Vs 'diameter of tank
        Q1 = pi / 4 * D ^ 2 * (1 - 0.15) ^ 2 * Vs 'flow from each tank
        N = CInt(Q / Q1) 'number of tanks

    Cycle:
        D = (Q / (N * pi * 0.85 ^ 2 * Vs / 4)) ^ 0.5 'actual diameter
        A = pi / 4 * (0.85 * D) ^ 2 'area of tank
        H = Q * t / (N * A) 'clarification depth
        'check D/H
        rat = D / H
        If rat > 6 And rat < 10 Then
            OK = "T" : t1 = "D/H ratio=" + rat.ToString + " is O.K."
        Else
            OK = "F" : N = N + 1 : GoTo cycle
        End If
        'solids content in the final sludge
        Cc = (1 - mc / 100) * 1000
        'volume of sludge hopper Vh
        Eff = Eff / 100 'efficiency ratio
        Vh = Q * Co * Eff / (24 * tr * Cc)
        dh = (Vh / 0.227 + (2 * rdash) ^ 3) ^ (1 / 3)
        TextBox10.Text = fn(N)
        TextBox11.Text = fn(D)
        TextBox12.Text = fn(H) + " : " + t1
        TextBox13.Text = fn(dh)
    End Sub

    Function fn(ByVal N As Double) As String
        'formats a number N into two decimal places and returns it as string
        Return FormatNumber(N, 2).ToString
    End Function
    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
        calculateResults()
    End Sub
End Class
```

Fig. 4 Enlisting of VB.Net code to design a circular sedimentation tank.

3.2 Facultative Waste Stabilization Ponds

Waste Stabilization Ponds are biological treatment processes whereby stabilization of organic matter is achieved by the aid of microorganisms in shallow basins. They are regarded as low cost treatment systems especially for developing countries. The factors that need to be considered in design aspects include [17-20]:

- Availability of required land area.
- Site selection with respect to topography, climatological conditions and hydrological aspects.
- Wastewater flow pattern and characteristics.

- Loading rates and available nutrients.
- Pond design methodologies.

A computer program is written to determine the size of a facultative pond needed to treat wastewater of a certain BOD₅ for example 400 mg/l) given the other required data such as effluent BOD (say 60 mg/l), wastewater flow (say 2000 m³/d), depth of pond (say 1.5 m), lowest temperature (say 24° C), removal rate constant for pond, k_p (say 0.35/day at 20° C). Assumed figures were used to verify the program.

Assuming complete mixing conditions then,

$$\frac{L_e}{L_i} = \frac{1}{1 + k_p t} \quad (11)$$

Where

L_e = Effluent BOD, mg/L.

L_i = Influent BOD, mg/L.

t = Retention time, or mean hydraulic retention time, d.

k_p = Removal rate constant for waste stabilization pond, d⁻¹.

$$[k_p]_{24} = [k_p]_{20} \{1.05\}^{[T-20]} \quad (12)$$

Program enlisting for determining size of a facultative pond needed to treat wastewater is as shown on Figure 5.

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles MyBase.Load
        Me.Text = "Size of facultative pond"
        Me.MaximizeBox = False
        Me.FormBorderStyle = Windows.Forms.FormBorderStyle.FixedSingle
        Label11.Text = "Effluent BOD, Le (mg/L)"
        Label12.Text = "Influent BOD, Li (mg/L)"
        Label13.Text = "Wastewater flow (m3/d)"
        Label14.Text = "Depth of pond (m)"
        Label15.Text = "Removal rate constant at 20C"
        Label16.Text = "Lowest Temp. (C)"
        Label17.Text = ""
        Button1.Text = "&Calculate"
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
        Dim Le, Li, Q, d, kp, kp24 As Double
        Dim t, temp, V, SA As Double
        Le = Val(TextBox1.Text)
        Li = Val(TextBox2.Text)
        Q = Val(TextBox3.Text)
        d = Val(TextBox4.Text)
        kp = Val(TextBox5.Text)
        temp = Val(TextBox6.Text)
        kp24 = kp * (1.05 ^ (temp - 20))
        t = ((Li / Le) - 1) / kp24
        V = t * Q
        SA = V / d
        Label17.Text = "Detention time = " + FormatNumber(t, 2) + " days"
        Label17.Text += vbCrLf + "Volume = " + FormatNumber(V, 2) + " m3"
        Label17.Text += vbCrLf + "Surface area = " + FormatNumber(SA, 2) + " m2"
    End Sub
End Class
```

Fig. 5 Enlisting of VB.Net code for sizing a facultative pond needed to treat wastewater.

3.3 Disposal into Receiving Bodies of Water

The effects of raw discharges into bodies of water include [2-10]:

- Change in characteristics of receiving water.
- Endangering of aquatic life.

- Creation of mal odor and nuisance.
- Production of taste and odor problems.
- Build-up of concentration of toxic materials.
- Generation of public health hazards.
- Deceleration of the process of self-purification.

The chosen example enables writing a computer program to compute temperature of the mixture of wastewater discharged from a factory and that of the stream, amount of the 5-day BOD of the mixture, critical oxygen deficit, and location of the critical oxygen deficit given amount of wastewater (say 21600 m³/d) finding its way into the neighboring stream, stream flow rate (say 10 m³/s), velocity of the stream, temperature of the stream (say 20° C), temperature of the discharged wastewater (say 30° C), the 5-day BOD at 20° C of the wastewater (say 400 mg/l), BOD of the stream (say 2.5 mg/l), amount of dissolved oxygen contained in the stream (say 25 percent of the saturation value), and DO of wastewater (say anaerobic) and k₁ and k₂ (at 20° C say 0.3 and 0.7 /day respectively). Certain actual data were used for program verification.

Program enlisting for computing concentration of pollutant from factory wastewater discharged to a stream is as shown on Figure 6.

4. Conclusions

The main conclusions drawn from this work may be summarized as follows:

- Visual Basic .NET as a selected language for writing computer programs proved its suitability and easiness in writing the programs.
- Flow charts are a useful aid when designing computer modules, which helps to ensure the correctness of the module and eases the design process.
- Structural design concepts for treatment units could be developed while offering hydraulic and their engineering designs.
- A dynamic design working platform may be advocated for joining different programs in an effort to integrate units design.
- A wider platform is therefore proposed in order to facilitate future linkage and expansion of several design modules for water treatment units. Such platform could cater for the conceptualization, analysis, design, and optimization of water/wastewater treatment units in many dimensions from the hydraulic and structural points of view.

```
Public Class Form1
    Dim Cs(31) As Double
    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
        Dim Qw, Qs, V, Tw, Ts, BODs, BODw As Double
        Dim Dow, Dos, k1, k2 As Double
        Dim Tm, BODm As Double
        Qw = Val(TextBox1.Text)
        Qs = Val(TextBox2.Text)
        V = Val(TextBox3.Text)
        Ts = Val(TextBox4.Text)
        Tw = Val(TextBox5.Text)
        BODs = Val(TextBox6.Text)
        BODw = Val(TextBox7.Text)
        Dow = Val(TextBox8.Text)
        k1 = Val(TextBox9.Text)
        k2 = Val(TextBox10.Text)
        Qw /= (24 * 60 * 60)
        V *= 24
        Dos = 0.85 * (Cs(20))
        Tm = ((Tw * Qw) + (Ts * Qs)) / (Qw + Qs)
        TextBox11.Text = "Temp. of mixture = " + FormatNumber(Tm, 2) + " C"
        TextBox11.Text += vbCrLf
        BODm = ((BODw * Qw) + (BODs * Qs)) / (Qw + Qs)
        TextBox11.Text += "BOD of mixture = " + FormatNumber(BODm, 2) + " mg/l"
        TextBox11.Text += vbCrLf
        'Use Streeter-Phelps equation to determine the critical oxygen deficit
        'correct k1 and k2 to mixture temp. Tm
        Dim xc, dc, tc, k1T, k2T, BODu, _Do As Double
        k1T = k1 * (1.135 ^ (Tm - 20))
        k2T = k2 * (1.024 ^ (Tm - 20))
        'calculate ultimate BOD
        BODu = (1.25 * BODm) / (1 - (10 ^ (-k1 * 5)))
        TextBox11.Text += "Ultimate BOD = " + FormatNumber(BODu, 2) + " mg/l"
        TextBox11.Text += vbCrLf
        _Do = (Dow * Qw + Dos * Qs) / (Qw + Qs)
        _Do = Cs(20) - _Do
        tc = (1 / (k2T - k1T)) * Math.Log((k2T / k1T) *
            * (1 - ((Do * (k2T - k1T)) / (k1T * BODu))))
        TextBox11.Text += "Time to reach critical point = "
            + FormatNumber(tc, 2) + " days"
        TextBox11.Text += vbCrLf
        dc = (k1T / k2T) * BODu * (Math.E ^ (-k1T * tc))
        xc = V * tc
        TextBox11.Text += "Dc = " + FormatNumber(dc, 2) + " mg/l"
        TextBox11.Text += vbCrLf
        TextBox11.Text += "Location of critical oxygen deficit = "
            + FormatNumber(xc, 2) + " km"
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles MyBase.Load
        Me.Text = "Sample program"
        Me.FormBorderStyle = Windows.Forms.FormBorderStyle.FixedSingle
        Me.MaximizeBox = False
        Label1.Text = "Wastewater discharge flow, Qw (m3/d)"
        Label2.Text = "Stream discharge, Qs (m3/s)"
        Label3.Text = "Stream velocity (km/hr)"
        Label4.Text = "Stream temp., Ts (C)"
        Label5.Text = "Wastewater temp., Tw (C)"
        Label6.Text = "BOD5 of stream, BODs (mg/l)"
        Label7.Text = "BOD5 of wastewater, BODw (mg/l)"
        Label8.Text = "Dissolved oxygen in wastewater"
        Label9.Text = "First order reaction rate const, k1 (/d)"
        Label10.Text = "Reaeration constant, k2 (/d)"
        TextBox11.Multiline = True
        TextBox11.ScrollBars = ScrollBars.Vertical
        TextBox11.Height = 65
        Me.Height += 65
        Button1.Text = "&Calculate"
        'initialize table of oxygen concentration saturation
        Cs(0) = 14.6
        Cs(1) = 14.2 : Cs(2) = 13.8 : Cs(3) = 13.5
        Cs(4) = 13.1 : Cs(5) = 12.8 : Cs(6) = 12.5
        Cs(7) = 12.2 : Cs(8) = 11.9 : Cs(9) = 11.6
        Cs(10) = 11.3 : Cs(11) = 11.1 : Cs(12) = 10.8
        Cs(13) = 10.6 : Cs(14) = 10.4 : Cs(15) = 10.2
        Cs(16) = 10.0 : Cs(17) = 9.7 : Cs(18) = 9.5
        Cs(19) = 9.4 : Cs(20) = 9.2 : Cs(21) = 9.0
        Cs(22) = 8.8 : Cs(23) = 8.7 : Cs(24) = 8.5
        Cs(25) = 8.4 : Cs(26) = 8.2 : Cs(27) = 8.1
        Cs(28) = 7.9 : Cs(29) = 7.8 : Cs(30) = 7.6
    End Sub
End Class
```

Fig. 6 Enlisting of VB.Net code for computing concentration of pollutant from factory wastewater discharged to a stream.

Acknowledgments

Sincere thanks and words of gratitude are addressed to Mr. Dr. Mohammed I. M. Abdel-Magid for his insights about the programmed algorithms and VB.Net syntax. Further appreciation is directed to those great people who didn't

hesitate a second in helping with every possible way and form: Dr. Abdel-Ghani Abdel-Galil (Manager of Environmental Audit, Inspection and Monitoring Project in Qatar) of Environmental Solutions Inc. KSA; Dr. Baraka Mohamed Kabir, Dr. Suhair Abdel-Qayoum, Dr. Mohamed Abugaib, Eng. Abusabah Elfatih, Eng. Alsadig Almaleeh, Eng Kattab Ahmed, Mrs. Sabah Sid Ahmed and Ms. Buloud Abdel-Marouf at the College of Water and Environmental Engineering, SUST; Dr. Abdallah M. T. Shigidi at College of Engineering, SUST; Eng. Amal Rabbah (Deputy General Manager) of Industrial Research and Consultancy Center; Dr. Abdul-Rahman ben Salih Hariri (Dean College of Engineering), Prof. Isam Mohamed, Dr. Sami Abullah Osman of Environmental Engineering Department - University of Dammam, Kingdom of Saudi Arabia; Dr. AbdelWahid Hago (Head of Department of Civil Engineering), Sultan Qaboos University, Sultanate of Oman; Dr. Elnasir Abdelwahab of Water Resources Planning Model, NBI, Ethiopia; Eng. Samar Kamaledin of Parsons Engineering Consultants, Sultanate of Oman.

References

- [1] M. Halvorson, Microsoft Visual Basic 2010: Step by Step, Microsoft Press, Washington, 2010.
- [2] I. M. Abdel-Magid and M. I. M. Abdel-Magid, Problem Solving in Environmental Engineering, Dammam University Press, Dammam, 2015 (under publication).
- [3] American Water Works Association and American Society of Civil Engineers, Water Treatment Plant Design 5/E, McGraw-Hill Professional; 5 Ed., 2012.
- [4] N. Gray, Water Technology, Taylor & Francis; 3 Ed., 2010.
- [5] J. E. Drinan, and F. Spellman, Water and Wastewater Treatment: A Guide for the Non-engineering Professional, CRC Press; 2 Ed., 2012.
- [6] L. W. Mays, Water Resources Engineering, Wiley; 2 Ed., 2010.
- [7] Water Environment Federation, Operation of Municipal Wastewater Treatment Plants (3-Volume Set), McGraw-Hill Professional; 6 Ed., 2007.
- [8] F. R. Spellman, Handbook of Water and Wastewater Treatment Plant Operations, CRC Press; 2 Ed., 2008.
- [9] Degremont, Water treatment handbook, Lavoisier; 7th Ed., 2007.
- [10] N. K. Shammass, and L. K. Wang, Fair, Geyer, and Okun's, Water and Wastewater Engineering: Water Supply and Wastewater Removal, Wiley; 3 Ed., 2010.
- [11] M. J. Hammer, Water and Wastewater Technology, Prentice Hall; 2011.
- [12] S. Kawamura, Integrated Design and Operation of Water Treatment Facilities, Wiley; 2 Ed., 2000.
- [13] W. McCabe, J. Smith and P. Harriott, Unit Operations of Chemical Engineering, McGraw Hill Chemical Engineering Series, 2004.
- [14] K. Williamson, Civil Engineering: Water & Wastewater Treatment Review, Kaplan Publishing, 2007.
- [15] Inc. Metcalf & Eddy, G. Tchobanoglous, H. D. Stensel, R. Tsuchihashi, and F. Burton, Wastewater Engineering: Treatment and Resource Recovery, McGraw-Hill Science/Engineering/Math; 5 Ed., 2013.
- [16] J. A. Nathanson, Basic Environmental Technology: Water Supply, Waste Management & Pollution Control, Prentice Hall, 2007.
- [17] K. J. Howe, D. W. Hand, J. C. Crittenden and R. R. Trussell, Principles of Water Treatment, Wiley; 1 Ed., 2012.
- [18] J. R. Mihelcic, and J. B. Zimmerman, Environmental Engineering: Fundamentals, Sustainability, Design, Wiley; 1 Ed., 2009.
- [19] W. Viessman, M. J. Hammer, E. M. Perez, and P. A. Chadik, Water Supply and Pollution Control, Prentice Hall; 8 Ed., 2008.
- [20] S. Lin, and C. Lee, Water and Wastewater Calculations Manual, McGraw-Hill Professional; 2 Ed., 2007.

Hisham I. M. Abdel-Magid (M.Sc. B.Sc. MIWA MPMI AMASCE GMICE GSEC MSES FMWTS) is a Civil & Infrastructures Engineer by profession, who is interested in software programming for specialized engineering models in the field of GIS for water resources and environmental engineering his field, as well as being interested in computer graphics designing. Eng. Hisham is currently an independent Hydroinformatics engineer and director of his own firm (Enginormatics Engineering Ltd.) in London, UK. He worked as a regional GIS & Hydroinformatics Consultant for Eastern Nile Technical Regional Office (ENTRO), Nile Basin Initiative (NBI), Ethiopia; Senior Drains Engineer with Kenana Engineering & Technical Services for their DSR Project in Nigeria; Lecturer and Acting Head of Environmental Engineering Dept. of the College of Water and Environmental Engineering at Sudan University for Science and Technology (SUST); Civil & Infrastructures Engineer for Dar Al-Handasah Shair and Partners, Qatar; Research Engineer at Institute for Technological Research (ITR) of The National Center for Research (NCR), Sudan. He patents the co-development of *CN-REM* model for rainfall-runoff estimation under ArcGIS, as well as *WISAM* (a decision support platform for the design of wastewater treatment units). Eng. Hisham is completing his PhD in Environmental Engineering at SUST, Sudan, while obtaining an MSc degree in Hydrogeology from University of Strathclyde, UK. In 2007 he obtained his MSc in Civil Engineering from SUST, and his BSc (Honours) in Civil Engineering from Al-Zaiem Al-Azhari University, Sudan, 2004.

Yousif Ali Yousif (Ph.D. M.Sc. B.Sc.) is Associate Professor at School of Civil Engineering, College on Engineering at Sudan University of Science and Technology (SUST). He has been the former Dean, College of Water and Environmental Engineering for almost ten years. He also worked in many projects related to water engineering and sanitation design. He have various academic achievements in his field of specialization; ranging from international publications, book authoring, scientific supervision, technical leadership, research advancement as well as being actively participating in technical committees, conferences, workshops and dialogues.

Elsadig Elhadi Elhassan (Ph.D. M.Sc. BTech) is the Secretary of Academic Affairs at Sudan University of Science and Technology (SUST), Sudan. He held several technical and professional posts that ranged from being: Head of Civil Engineering department, College of Engineering; Deputy Dean College of Water and Environmental Engineering (CWEE), Head department of water resources Engineering at CWEE, Head of Irrigation and Drainage Engineering Division at SUST. He is a member of many academic societies and professional institutions regionally and globally. He published several papers, articles and books in his field of specialization.