AN APPLICATION PROGRAM TO ANALYZE EQUILIBRIUM pH USING VB.NET

Harry S.J. Koleangan[1)]

[1)]Program Studi Kimia FMIPA Universitas Sam Ratulangi, Manado 95115

ABSTRACT

An application program to analyze equilibrium pH of an anaerobic waste solution model had been developed. The equilibrium pH was analyzed using bisection method written in VB.NET 2008 Express Edition program language. Application of the program on the secondary data showed that pH of an anaerobic waste solution was 6.56.

Keywords: bisection method, speciation, VB.NET 2008

PROGRAM APLIKASI UNTUK MENGANALISA KESEIMBANGAN pH MENGGUNAKAN VB.NET

ABSTRAK

Telah dilakukan penelitian untuk membuat sebuah program aplikasi dalam menganalisis keseimbangan pH dari model sampah anaerob. Keseimbangan pH dianalisis menggunakan metode biseksi dengan bahasa pemrograman VB.NET 2008. Penerapan program pada data sekunder menunjukkan bahwa pH dari sampah anaerob adalah 6,56.

Kata kunci: metode biseksi, spesiasi, VB.NET 2008

INTRODUCTION

Concentration of H^+ in water (usually written as pH) has important role in water quatility determination because pH controls the dissolution and precipitation of solids, alters the solubility of gases, catalyzes many other reactions, and affects the interactions of chemicals with organisms.

pH analyzes can be divided into two categories. Firstly, determination of equilibrium pH in solution in which all of the substances used to prepare the solution have been known previously, and secondly, determination of the numbers of acid or base that should be added to reach a desired pH.

In the first category, pH analyzes can be carried out by deriving an equation that has one unknown variabel, that is H^+ concentration. The equation is usually a polynom which is difficult to solve analytically (exact method). Another way to solve this kind of equation is by using graphical method, that is by plotting the function on a graph and the point at which x-value gives f(x) = 0 will then the solution for the equation. The disadvantage of this method is it has low level of precision. Other way to solve the equation is by using trial and error method, that is by taking an x-value randomly and solving it to see whether of not f(x) equals 0. If not then another x-value is taken until f(x) = 0 or close to zero (Chapra and Canale, 2006). It is obvious that this method is extremely unefficient.

Bisection method is one of numerical methods that can be used to find the roots of polynomial equations. The method implements many systematic and arithmetic operations so that if the calculation is solved manually will be very exhausting and inefficient.

The systematic approach of bisection method can be completed easily and rapidly using a computer program so that the polynomial equation can be solved in a very short time and with a high degree of accuracy. The purpose of this research is to develop a computer program, using a VB.NET Express Edition programming language, that can be employed to determine the equilibrium pH and the distribution of chemical species from an anaerobic waste solution model in which all the substances used to make the solution are previously known.

RESEARCH METHODS

Data

The data are adapted from a problem in *Water Chemistry* textbook page 226, written by Mark M. Benjamin (2002), as follows: a model of anaerobic waste solution is prepared as a mixture of 10^{-2} *M* $NaHCO_3$, 10^{-4} *M* Na_2S, and 4 × 10^{-3} M HAC. Determine the solution pH at equilibrium. Upon dissolution, sodium sulfide (Na_2S) releases sulfide ion (S^{2-}), a relatively strong base that can protonate to form bisulfide ion (HS^-) or hydrogen sulfide (H_2S). The values of pK_{a1} and pK_{a2} for H_2S are 6.99 and 12.92 respectively, while those of H_2CO_3 are 6.35 and 10.33 respectively, and pK_a for acetic acid (HAC) is 4.76. Assume that the solution is ideal.

Equations for Solving The Problem

Equations needed to solve the problem are derived using procedure proposed by Benjamin (2002), that is by preparing species lists, writing the relevant equilibrium constants, mass balance(s), and charge balance equation as follows:

- Species list:
  Tipe a: H^+, OH^-.
  Tipe b: H_2CO_3, HCO_3^-,CO_3^{2-}, H_2S, HS^-, S^{2-}, HAc, Ac^-.
  Tipe c: Na^+.
- Equilibrium constants:
  Equilibrium constant of water:
  $$K_w = \{H^+\}\{OH^-\} = 10^{-14} \qquad (1)$$
  Equilibrium constants of carbonic acid:
  $$K_{a1} = \frac{\{H^+\}\{HCO_3^=\}}{\{H_2CO_3\}} \ldots\ldots\ldots\ldots \qquad (2)$$
  $$K_{a2} = \frac{\{H^+\}\{CO_3^{2=}\}}{\{HCO_3^-\}} \ldots\ldots\ldots\ldots\ldots \qquad (3)$$
  Equilibrium constants of hydrogen sulfide:
  $$K_{a1} = \frac{\{H^+\}\{HS^-\}}{\{H_2S\}} \ldots\ldots\ldots\ldots\ldots\ldots \qquad (4)$$
  $$K_{a2} = \frac{\{H^+\}\{S^{2-}\}}{\{HS^-\}} \ldots\ldots\ldots\ldots\ldots\ldots \qquad (5)$$
  Equilibrium constant of acetic acid:
  $$K_a = \frac{\{H^+\}\{Ac^-\}}{\{HAc\}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (6)$$
- Mass balances:
  $$TOTCO_3 = \{H_2CO_3\} + \{HCO_3^-\} + \{CO_3^{2-}\} = 10^{-2}M \ldots\ldots\ldots\ldots\ldots \qquad (7)$$
  $$TOTS = \{H_2S\} + \{HS^-\} + \{S^{2-}\} = 10^{-4}M \qquad (8)$$
  $$TOTAc = \{HAc\} + \{Ac^-\} = 4\times10^{-3}M \qquad (9)$$
  $$TOTNa = \{Na^+\} = 1{,}02\times10^{-4}M \ldots\ldots \qquad (10)$$
- Charge balance:
  $$\{H^+\} + \{Na^+\} = \{OH^-\} + \{HCO_3^-\} + 2\{CO_3^{2-}\} + \{HS^-\} + 2\{S^{2-}\} + \{Ac^-\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (11)$$

Subsequently, equation (11) must be manipulated in such away that it has only one unknown variable H^+, as follows:

From equation (10):
$$\{Na^+\} = 1{,}02\times10^{-4}M \ldots\ldots\ldots\ldots \qquad (12)$$
From equation (1):
$$\{OH^-\} = \frac{K_w}{\{H^+\}} = \frac{10^{-14}}{\{H^+\}} \ldots\ldots\ldots\ldots \qquad (13)$$
From equations (2) and (3):
$$\{HCO_3^-\} = \alpha_{1,H_2CO_3} \times TOTCO_3 \ldots\ldots \qquad (14)$$
$$\{CO_3^{2-}\} = \alpha_{2,H_2CO_3} \times TOTCO_3 \ldots\ldots \qquad (15)$$
where:
$$\alpha_{1,H_2CO_3} = \frac{\{HCO_3^-\}}{TOTCO_3}$$
$$= \frac{\{HCO_3^-\}}{\{H_2CO_3\} + \{HCO_3^-\} + \{CO_3^{2-}\}}$$
$$= \frac{1}{\frac{\{H_2CO_3\}}{\{HCO_3^-\}} + 1 + \frac{\{CO_3^{2-}\}}{\{HCO_3^-\}}}$$
$$= \frac{1}{\frac{\{H^+\}}{K_{a1}} + 1 + \frac{K_{a2}}{\{H^+\}}}$$
$$= \frac{K_{a1}\{H^+\}}{\{H^+\}^2 + K_{a1}\{H^+\} + K_{a1}K_{a2}}$$
$$\alpha_{2,H_2CO_3} = \frac{\{CO_3^{2-}\}}{TOTCO_3}$$

$$= \frac{\{CO_3^{2-}\}}{\{H_2CO_3\} + \{HCO_3^-\} + \{CO_3^{2-}\}}$$

$$= \frac{1}{\frac{\{H_2CO_3\}}{\{CO_3^{2-}\}} + \frac{\{HCO_3^-\}}{\{CO_3^{2-}\}} + 1}$$

$$= \frac{1}{\frac{\{H^+\}^2}{K_{a1}K_{a2}} + \frac{\{H^+\}}{K_{a2}} + 1}$$

$$= \frac{K_{a1}K_{a2}}{\{H^+\}^2 + K_{a1}\{H^+\} + K_{a1}K_{a2}}$$

From equations (4) and (5):

$$\{HS^-\} = \alpha_{1,H_2S} \times TOTS \quad \text{…………} \quad (16)$$

$$\{S^{2-}\} = \alpha_{2,H_2S} \times TOTS \quad \text{………………} \quad (17)$$

where:

$$\alpha_{1,H_2S} = \frac{\{HS^-\}}{TOTS}$$

$$= \frac{\{HS^-\}}{\{H_2S\} + \{HS^-\} + \{S^{2-}\}}$$

$$= \frac{1}{\frac{\{H_2S\}}{\{HS^-\}} + 1 + \frac{\{S^{2-}\}}{\{HS^-\}}}$$

$$= \frac{1}{\frac{\{H^+\}}{K_{a1}} + 1 + \frac{K_{a2}}{\{H^+\}}}$$

$$= \frac{K_{a1}\{H^+\}}{\{H^+\}^2 + K_{a1}\{H^+\} + K_{a1}K_{a2}}$$

$$\alpha_{2,H_2S} = \frac{\{S^{2-}\}}{TOTS}$$

$$= \frac{\{S^{2-}\}}{\{H_2S\} + \{HS^-\} + \{S^{2-}\}}$$

$$= \frac{1}{\frac{\{H_2S\}}{\{S^{2-}\}} + \frac{\{HS^-\}}{\{S^{2-}\}} + 1}$$

$$= \frac{1}{\frac{\{H^+\}^2}{K_{a1}K_{a2}} + \frac{\{H^+\}}{K_{a2}} + 1}$$

$$= \frac{K_{a1}K_{a2}}{\{H^+\}^2 + K_{a1}\{H^+\} + K_{a1}K_{a2}}$$

From equation (6): $\{Ac^-\} = \alpha_{1,HAc} \times TOTAc$ ……………… (18)

where:

$$\alpha_{1,HAc} = \frac{\{Ac^-\}}{TOTAc}$$

$$= \frac{\{Ac^-\}}{\{HAc\} + \{Ac^-\}}$$

$$= \frac{1}{\frac{\{HAc\}}{\{Ac^-\}} + 1}$$

$$= \frac{1}{\frac{\{H^+\}}{K_a} + 1}$$

$$= \frac{K_a}{\{H^+\} + K_a}$$

By incorporating equations (12) through (18) to equation (11) it is obtained one polinomial having only one unknown variable $\{H^+\}$, namely:

$$\{H^+\} + 1{,}02 \times 10^{-4} = \frac{K_w}{\{H^+\}} + \left(\alpha_{1,H_2CO_3} + 2\alpha_{2,H_2CO_3}\right)TOTCO_3 + \left(\alpha_{1,H_2S} + 2\alpha_{2,H_2S}\right)TOTS + \alpha_{1,HAc}TOTAc$$

….

Equation 19 can be solved numerically using bisection method written in VB.NET 2008 Express Edition programming language.

Algorithm of The Program and Source Code

Algorithm to solve a polynomial using the bisection method is presented in the following flow chart:

RESULTS AND DISCUSSION

The program that has been developed was tested using the existing data. When first running, the program asked to enter the desired level of accuracy of the calculations. After entering this value, the program showed the following display on figure 1

The input data were then processed by the program to calculate the equilibrium pH. This was carried out by pressing the pH equlibirum button and the program subsequently displayed the results of the calculations, as follows on figure 2

The results showed that the equilibrium pH of the wastewater model is 6.561396 with accuracy level up to 5 significant figures and prespecified tolerance (Es) = 0.0005%. It means that the value that we can use from pH calculation is 6.5614 (5 significant figures) with the certainty level of the calculation 100% - 0.0005% (= 99.9995%.) Furthermore, it was also acquired that the approximate percent relative error (Ea) is equals to 0.00033%. This means that the result has relative percent error of approximately 0.00033%. The picture also showed that the number of iterations required by the bisection method to solve equations (19) was 40 iterations.

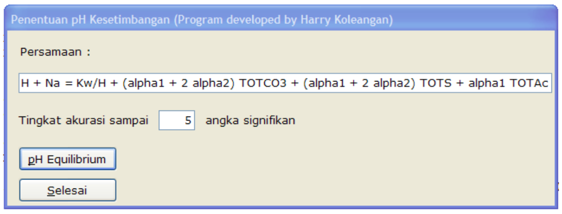


Figure 1. Display After Data Input

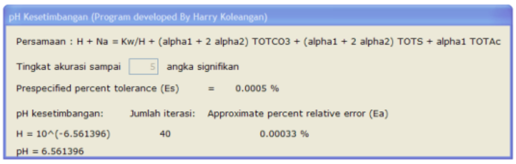
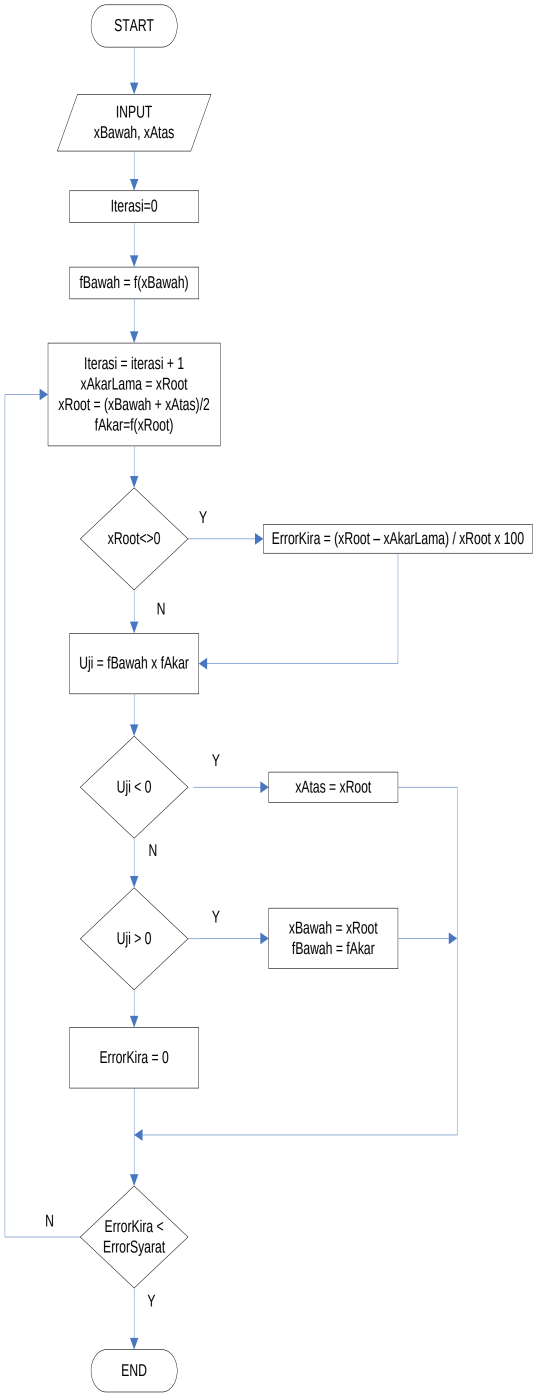


Figure 2. Results Display

CONCLUSION

Based on the test program above it can be concluded that the program can be used for pH equilibrium analysis of model waste water. This program can also be used for pH equilibrium analysis of other solutions provided that an equation similar to equation (19) can be derived, and then the equation is inserted in the program. The advantages of using this program are the calculations can be carried out more quickly and untiringly, and the results obtained have a high level of accuracy.



The source code is written down in the attachment.

REFERENCES

Benjamin, M. M. 2002. *Water Chemistry*. McGraw-Hill, New York, NY10020.

Caruso, J.A., K.L. Sutton and K.L. Ackley. 2000. *Elemental Speciation. Comprehensive Analytical Chemistry, Vol. XXXII*, Elsevier Science B.V.

Chapra, S. C. and R. P. Canale. 2006. *Numerical Methods for Engineers. 5th Ed.* McGraw-Hill, New York, NY 10020.

Conte, S. D. and Carl de Boer. 1980. *Elementary Numerical Analysis. An Algorithmic Approach. 3rd Ed.* McGraw-Hill, New York.

Petroutsos, E., M. Ridgeway. 2008. *Mastering Visual Basic 2008*, SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. 2007. *Numerical Recipes. The Art of Scientific Computing. 3rd Ed.*, Cambridge University Press.

Attachment

Source Code

```
'*********************************************************************
'A program  to analyze speciation of an anaerobic waste solution model
'Developed by: Ir. Harry S. J. Koleangan, M.Si.
'Chemistry Department, FMIPA Unsrat, Manado
'*********************************************************************

Option Strict On
Imports System.Math
Public Class Persamaan
    Dim xLower, xUpper As Double                '= selang yang digunakan untuk mencari
akar persamaan
    Dim ErrorSyarat, ErrorKira As Double        '= prespecified dan approximate error
    Dim xRoot As Double                         '= akar penyelesaian yang diperoleh dari
metode bisection
    Dim iterasi As Integer                      '= banyaknya iterasi yang digunakan pada
metode bisection
    Dim u As Integer                            '= banyaknya titik potong dengan sb x
    Dim u As Integer
    Dim Kw, Ka_HAc, Ka1_H2CO3, Ka2_H2CO3, Ka1_H2S, Ka2_H2S As Double
    Dim TOT_Ac, TOT_CO3, TOT_S As Double

    Sub nilaiAwal()
        xLower = 0
        xUpper = 1
        ErrorSyarat = BatasError(CInt(txtAkurasi.Text))
        Kw = 10 ^ -14
        Ka_HAc = 10 ^ -4.3           'Ka of acetic acid
        Ka1_H2CO3 = 10 ^ -6.35       'Ka1 of carbonic acid
        Ka2_H2CO3 = 10 ^ -10.33      'Ka2 of carbonic acid
        Ka1_H2S = 10 ^ -6.99         'Ka1 of hydrogen sulfide
        Ka2_H2S = 10 ^ -12.92        'Ka2 of hydrogen sulfide
        TOT_Ac = 4 * 10 ^ -3         'total concentration of Ac
        TOT_CO3 = 10 ^ -2            'total concentration of CO3
        TOT_S = 10 ^ -4              'total concentration of S
    End Sub

    Function evaluasiFungsi(ByVal x As Double) As Double
        Dim TOT_Na As Double = 1.02 * 10 ^ -2   'total concentration of Na
        Dim pembagiHAc As Double = x + Ka_HAc
        Dim alpha1_HAc As Double = Ka_HAc / pembagiHAc
        Dim pembagiH2CO3 As Double = x ^ 2 + Ka1_H2CO3 * x + Ka1_H2CO3 * Ka2_H2CO3
        Dim alpha1_H2CO3 As Double = Ka1_H2CO3 * x / pembagiH2CO3
        Dim alpha2_H2CO3 As Double = Ka1_H2CO3 * Ka2_H2CO3 / pembagiH2CO3
        Dim pembagiH2S As Double = x ^ 2 + Ka1_H2S * x + Ka1_H2S * Ka2_H2S
        Dim alpha1_H2S As Double = Ka1_H2S * x / pembagiH2S
        Dim alpha2_H2S As Double = Ka1_H2S * Ka2_H2S / pembagiH2S
        evaluasiFungsi = x + TOT_Na - Kw / x - (alpha1_H2CO3 + 2 * alpha2_H2CO3) * TOT_CO3
- (alpha1_H2S + 2 * alpha2_H2S) * TOT_S - alpha1_HAc * TOT_Ac
    End Function

    Sub ttkPotong()
        Dim t As Double
        Dim jarak As Double = (xUpper - xLower) / 10000
        Dim oldY, Y As Double
        oldY = evaluasiFungsi(xLower)
        u = 0
        For t = xLower To xUpper Step jarak
            Y = evaluasiFungsi(t)
            If oldY * Y < 0 Then
                u = u + 1
            End If
            oldY = Y
        Next
    End Sub

    Function BatasError(ByVal syarat As Double) As Double
        BatasError = 0.5 * 10 ^ (2 - syarat)
    End Function

    Sub Rekeng(ByVal xBawah As Double, ByVal xAtas As Double)
        Dim fBawah, fAkar, Uji As Double
        Dim xAkarLama As Double
```

```
        fBawah = evaluasiFungsi(xBawah)
        iterasi = 0
        Do
            iterasi = iterasi + 1
            xAkarLama = xRoot
            xRoot = (xBawah + xAtas) / 2
            fAkar = evaluasiFungsi(xRoot)
            If xRoot <> 0 Then
                ErrorKira = Math.Abs((xRoot - xAkarLama) / xRoot) * 100
            End If
            Uji = fBawah * fAkar
            If Uji < 0 Then
                xAtas = xRoot
            ElseIf Uji > 0 Then
                xBawah = xRoot
                fBawah = fAkar
            Else
                ErrorKira = 0
            End If
            If ErrorKira < ErrorSyarat Then
                Exit Sub
            End If
        Loop
    End Sub

    Function hitung_HA(ByVal konsentrasi_H As Double, ByVal Ka As Double) As Double
        hitung_HA = konsentrasi_H / (konsentrasi_H + Ka)
    End Function

    Function hitung_A_minus_1(ByVal konsentrasi_H As Double, ByVal Ka As Double) As Double
        hitung_A_minus_1 = Ka / (konsentrasi_H + Ka)
    End Function

    Function hitung_H2A(ByVal konsentrasi_H As Double, ByVal Ka1 As Double, ByVal Ka2 As
Double) As Double
        hitung_H2A = konsentrasi_H ^ 2 / (konsentrasi_H ^ 2 + Ka1 * konsentrasi_H + Ka1 *
Ka2)
    End Function

    Function hitung_HA_minus_1(ByVal konsentrasi_H As Double, ByVal Ka1 As Double, ByVal
Ka2 As Double) As Double
        hitung_HA_minus_1 = (konsentrasi_H * Ka1) / (konsentrasi_H ^ 2 + Ka1 *
konsentrasi_H + Ka1 * Ka2)
    End Function

    Function hitung_A_minus_2(ByVal konsentrasi_H As Double, ByVal Ka1 As Double, ByVal
Ka2 As Double) As Double
        hitung_A_minus_2 = (Ka1 * Ka2) / (konsentrasi_H ^ 2 + Ka1 * konsentrasi_H + Ka1 *
Ka2)
    End Function

    Dim AkarForm As New AkarPersamaan
    Private Sub btAkar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btAkar.Click
        nilaiAwal()
        ttkPotong()
        If u = 0 Then
            MsgBox("There is no solution in the range of " & xLower & " to " & xUpper)
            Exit Sub
        End If
        Rekeng(xLower, xUpper)

        'calculation of HAc and Ac concentration
        Dim HAc, Ac As Double
        HAc = hitung_HA(xRoot, Ka_HAc) * TOT_Ac
        Ac = hitung_A_minus_1(xRoot, Ka_HAc) * TOT_Ac

        'calculation of H2CO3, HCO3, and CO3 concentration
        Dim H2CO3, HCO3, CO3 As Double
        H2CO3 = hitung_H2A(xRoot, Ka1_H2CO3, Ka2_H2CO3) * TOT_CO3
        HCO3 = hitung_HA_minus_1(xRoot, Ka1_H2CO3, Ka2_H2CO3) * TOT_CO3
        CO3 = hitung_A_minus_2(xRoot, Ka1_H2CO3, Ka2_H2CO3) * TOT_CO3

        'calculation of H2S, HS, and S concentration
        Dim H2S, HS, S As Double
        H2S = hitung_H2A(xRoot, Ka1_H2S, Ka2_H2S) * TOT_S
        HS = hitung_HA_minus_1(xRoot, Ka1_H2S, Ka2_H2S) * TOT_S
```

```
        S = hitung_A_minus_2(xRoot, Ka1_H2S, Ka2_H2S) * TOT_S

        With AkarForm
            .lblIterasi.Text = ""
            .lblEa.Text = ""
            .lblIterasi.Text = Format(iterasi, "0")
            .lblEa.Text = Format(ErrorKira, "0.####0") & " %"
            .lblPers.Text = "Equation : " & txtPers.Text
            .txtAkurasi.Text = txtAkurasi.Text
            .lblEs.Text = Format(ErrorSyarat, "0.#####") & " %"

            .lblPH.Text = "pH = " & Format(-Log10(xRoot), "0.000000")

            .lbl_H2CO3.Text = "{H2CO3} = " & Format(H2CO3, "0.0000000000") & " M, or pX =
" & Format(-Log10(H2CO3), "#0.0000")
            .lbl_HCO3.Text = "{HCO3_-} = " & Format(HCO3, "0.0000000000") & " M, or pX = "
& Format(-Log10(HCO3), "#0.0000")
            .lbl_CO3.Text = "{CO3_2-} = " & Format(CO3, "0.0000000000") & " M, or pX = " &
Format(-Log10(CO3), "#0.0000")

            .lbl_H2S.Text = "{H2S} = " & Format(H2S, "0.0000000000") & " M, or pX = " &
Format(-Log10(H2S), "#0.0000")
            .lbl_HS.Text = "{HS_-} = " & Format(HS, "0.0000000000") & " M, or pX = " &
Format(-Log10(HS), "#0.0000")
            .lbl_S.Text = "{S_2-} = " & Format(S, "0.0000000000") & " M, or pX = " &
Format(-Log10(S), "#0.0000")

            .lbl_HAc.Text = "{HAc} = " & Format(HAc, "0.0000000000") & " M, or pX = " &
Format(-Log10(HAc), "#0.0000")
            .lbl_Ac.Text = "{Ac_-} = " & Format(Ac, "0.0000000000") & " M, or pX = " &
Format(-Log10(Ac), "#0.0000")

            .CancelButton = btKlaar
            .Show()
        End With
    End Sub

    Private Sub btKlaar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btKlaar.Click
        End
    End Sub
End Class
```