A UTILITARIAN COMPARISON OF NONLINEAR REGRESSION METHODS

By

CHETAN CHANDAK

Bachelor of Science in Chemical Engineering

Jawaharlal Nehru Technological University

Hyderabad, India

2007

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2009

A UTILITARIAN COMPARISON OF NONLINEAR

REGRESSION METHODS

Thesis Approved:

Dr. R. Russell Rhinehart

Thesis Adviser

Dr. Karen High

Dr. Sundar Madihally

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to thank Dr. R. Russell Rhinehart for guiding me through the research. It has been a great learning experience working under him.

It has not just been the research, but also the timely guidance he has provided for my course work at OSU. I appreciate his patience and confidence in my work, and his sincere concern towards his students to ensure a good future after their graduation. I would like to thank him again, and the Chemical Engineering department for providing me with the Graduate Assistantship during my degree. I also thank my committee members for their valuable inputs.

I am grateful to all the other members of the department and my friends at OSU for their support.

TABLE OF CONTENTS

Chapter Page

List of Tables vi

List of Figures vii

1. INTRODUCTION 1

1.1. Literature Review 4

1.1.1. Linear Regression 4

1.1.1.1. Least Squares Regression – Vertical distances 6

1.1.1.2. Least Squares Regression – Horizontal distances 7

1.1.1.3. Perpendicular regression 8

1.1.1.3.1. Perpendicular offsets 8

1.1.1.3.2. Shifting of axes 10

1.1.1.4. Geometric Mean Functional Relationship 13

1.1.1.5. Maximum Likelihood Method 13

1.1.2. Nonlinear Regression 14

1.2. Research Statement 17

2. METHOD 18

2.1. Scope of the study 19

2.2. Experiment 20

2.2.1. Titration 20

2.2.1.1. Interval halving root finding technique 23

2.2.1.2. Data generation 25

2.2.2. Packed bed reactor 30

Chapter Page

2.2.2.1. Output variable evaluation 32
2.2.2.1.1. Newton – Raphson method 32
2.2.2.2. Experimental data 36
2.3. Regression methods 39
2.3.1. Vertical distance method 39
2.3.2. Maximum likelihood method 40
2.3.2.1. Circular Contours 44
2.3.2.2. Propagation of uncertainty 46

3. EXPERIMENTAL 50
3.1. Titration 50
3.1.1 Data generation 50
3.1.2 Regression 50
3.1.2.1. R^3 cyclic direct search 52
3.1.2.2. Objective function evaluation 58
3.1.2.2.1. Least squares regression – Vertical distances 58
3.1.2.2.2. Maximum Likelihood and Normal distances 59
3.1.2.2.2.1. Golden Section Search 60
3.2 Packed bed reactor 64
3.2.1 Data generation 64
3.2.2 Regression 64
3.2.2.1 Parameter Optimization 64
3.2.2.2 Objective Function Evaluation 67
3.3. Assumptions 70
3.4. Result Analysis Techniques 72

4. RESULTS AND DISCUSSION 75

5. MODEL VALIDATION 93

6. CONCLUSIONS 95

Chapter Page

REFERENCES 97

APPENDICES

Appendix A – Titration Simulator Code..... 99

Appendix B – Packed Bed Simulator Code..... 112

Appendix C – Results: Parameter Distributions and Probability Plots..... 127

LIST OF TABLES

Table Page

4.1.1 Comparison of regression methods for $\sigma_x < \sigma_y$ 76
4.1.2 Comparison of regression methods for $\sigma_x = \sigma_y$ 80
4.1.3 Comparison of regression methods for $\sigma_x > \sigma_y$ 83
4.1.4 Comparison of regression methods for approximate variances for Maximum Likelihood method 85
4.1.5 Summary of Findings 88
4.2.1 Comparison of regression methods – Packed bed reactor 89

LIST OF FIGURES

Figure Page

1. Input vs. output for an illustrative nonlinear process 3
2. A prototype process simulator 5
3. Perpendicular distance – Linear regression 9
4. Perpendicular regression by Akaho's method 15
5. Drawback of Akaho's method 16
6. Interval halving method 24
7. $f\ vs.\ pH_g$ – Titration experiment 25
8. Normal distribution 26
9. Apparent vs. actual data – Titration experiment 27
10. Data generation – Titration experiment 28
11. A typical pH curve for regression 29
12. Packed bed reactor 31
13. Newton-Raphson method 33
14. $f\ vs.\ C_0$ – Packed bed reactor simulation 34
15. Newton Raphson flowchart 35
16. A typical inlet vs. outlet concentration plot for PBR simulation 36
17. Data generation – Packed bed reactor 37
18. Normal Distribution – Standard deviation and confidence intervals 38
19. Vertical distance method 39
20. Likelihood contours - Maximum likelihood approach 41
21. Maximum likelihood regression 42
22. Maximum likelihood regression 43
23. Circular contours – Maximum likelihood regression 45
24. Error propagation 46
25. R^3 direct search – one dimensional 52

Figure Page

26. R^3 direct search – two dimensional 54

27. Flowchart – R^3 cyclic direct search for titration parameter optimization 57

28. Flowchart – Vertical distance objective function evaluation 58

29. Golden section search 61

30. Flowchart – Golden section search 63

31. Flowchart - R^3 cyclic direct search for PBR parameter optimization 66

32. Flowchart - R^3 cyclic direct search for PBR objective function optimization (Normal distances and Maximum Likelihood method) 68

33 a. Typical 'A_0' distribution for vertical and maximum likelihood methods 73

b. Typical 'pK_a' distribution for vertical and maximum likelihood methods 73

34. Typical 'probability of deviation from the true value' plots for 'A_0' and 'pK_a' .. 74

35 a. 'A_0' distribution for vertical and maximum likelihood methods for σ_x = 0.25, σ_y= 0.5 78

35 b. 'pK_a' distribution for vertical and maximum likelihood methods for σ_x = 0.25, σ_y= 0.5 78

36. Probability of deviation from the true value plots for 'A_0' and 'pK_a' for σ_x = 0.25, σ_y= 0.5 79

37 a. 'A_0' distribution for vertical and maximum likelihood methods for σ_x = 0.25, σ_y= 0.25 81

37 b. 'pK_a' distribution for vertical and maximum likelihood methods for σ_x = 0.25, σ_y= 0.25 81

38. Probability of deviation from the true value plots for 'A_0' and 'pK_a' for σ_x = 0.25, σ_y= 0.25 82

39 a. 'A_0' distribution for vertical and maximum likelihood methods for σ_x = 0.1, σ_y= 0.25 with perturbed variance values for maximum likelihood objective function 85

39 b. 'pK_a' distribution for vertical and maximum likelihood methods for σ_x = 0.1, σ_y= 0.25 with perturbed variance values for maximum likelihood objective function 86

40. Probability of deviation from the true value plots for 'A_0' and 'pK_a' for σ_x = 0.1, σ_y= 0.25 with perturbed variance values for maximum likelihood objective function 86

41 a. Parameter 'k_0' distribution for vertical and maximum likelihood methods..... 90
41 b. Parameter 'k_1' distribution for vertical and maximum likelihood methods..... 90
41 c. Parameter 'E' distribution for vertical and maximum likelihood methods 91
42. Probability plots for packed bed regression parameters. 91

1. INTRODUCTION

Regression techniques are widely used to model empirical data. They find varied applications in many fields from formulating scientific rules and laws, to extrapolating data, and much more. Conventionally these methods employ the classic least squares (vertical distance) approach to estimate the model parameter values, which, however, is flawed in its underlining assumption itself that there are no uncertainties in the input variables. Input and output uncertainties are inherent to all practical experiments, and the vertical least square distance (VLSD) approach can cause a model parameter bias. The purpose of this study is to investigate the alternate approaches tried by other researchers and try and develop a new regression technique to overcome the above shortcomings.

Fig. 1 represents the output (Y) vs. input (X) of an illustrative nonlinear process. Due to uncertainty in the input values, if X_a is any nominally chosen setting or meter reading of the input given to the process, the actual input to the process lies anywhere between X_1 and X_2. The corresponding output for X_1 and X_2 may lie between Y_{11} and Y_{12}, Y_{21} and Y_{22} respectively. Hence for a given estimate of nominal, or target input X_a, the output value may lie anywhere within the region bounded by $\min\{Y_{11},\ Y_{12},\ Y_{21},\ Y_{22}\}$ and $\max\{Y_{11},\ Y_{12},\ Y_{21},\ Y_{22}\}$ depending upon the deviation from the true value. Similarly for a given input X_b, the output can lie anywhere within the region bounded by $\min\{Y_{51},\ Y_{52}, Y_{61},\ Y_{62}\}$ and $\max\{Y_{51},\ Y_{52}, Y_{61},\ Y_{62}\}$. ΔX and ΔY represent the probable uncertainty limits for the input and output values.

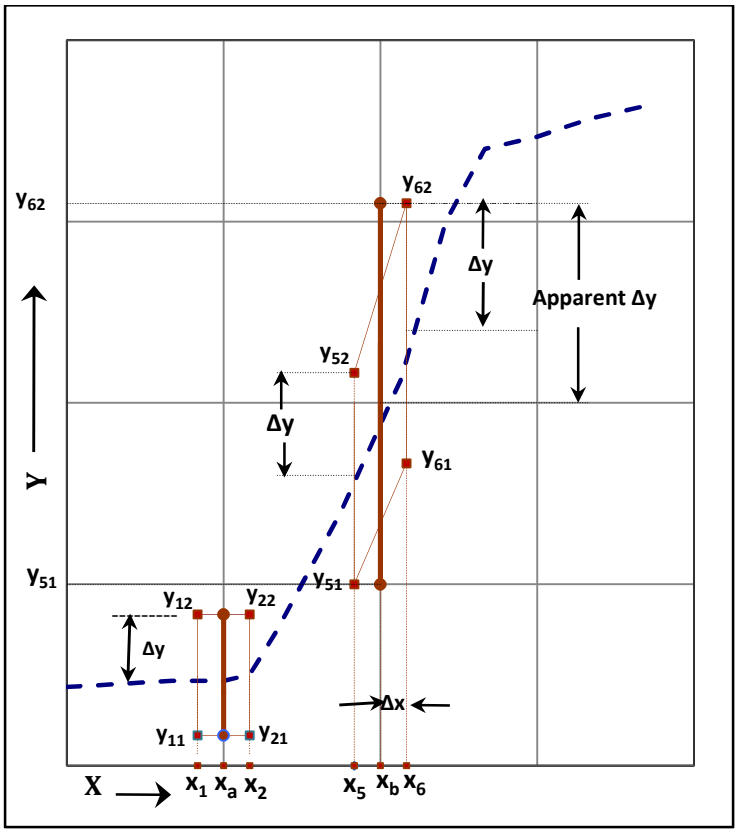


Figure 1: Input vs. output for an illustrative nonlinear process

Tradionally, developing an empirical model for the above process presumes that there are no uncertainties in the input values. Hence, if the experiment actually results in (X_6, Y_{62}) as the input-output pair, it would be erroneously assumed that the measurement corresponds to (X_b, Y_{62}). And, due to uncertainty in the output values, the measured output could lie anywhere along the vertical line intersecting X_b between Y_{62} and Y_{51}. This presumption might lead to the parameter values close to the actual ones when the slope of the curve is similar for all X-values. However, when the slope of the curve makes large changes, the relative importance of data shifts. For example, Fig. 1 shows that the possible ΔY deviation at X_b appears very large relative to the ΔY deviation at X_a even though the precision on Y is the same at both X-values. Using sum-of-squared output deviation $(\Sigma \Delta Y^2)$ as the objective function, would give too much importance to

the input X_b, relative to X_a. The truth is better indicated by revealing both X and Y uncertainty on the data and a likelihood region around the data point.

Hence, with the idea of sweeping all the uncertainty in the output variable being fundamentally incorrect, alternative approaches like the perpendicular regression through minimization of the perpendicular offsets [3, 7] or shifting of data points [8], the geometric mean functional relationship (GMFR) [4, 5, 9], the maximum likelihood method [1, 4, 5, 9], etc., were developed for linear regression. Nonlinear regression, however, hasn't been as extensively studied as the linear regression.

1.1 LITERATURE REVIEW

1.1.1 LINEAR REGRESSION

As cited in [1, 2], a model for linear regression does not just imply a straight line. A model is said to be linear in parameters if the second and higher order derivatives of the function with respect to the parameters are zero, i.e.,

$$\frac{\partial^2 G(\alpha, X)}{\partial \alpha_i \partial \alpha_j} = 0 \text{ [1]} \tag{1}$$

where 'G' is the function relating the model parameters $\alpha_i, \alpha_j ..$, with the independent variable X. Hence, a polynomial of the form, $G = \alpha + \beta X + \gamma X^2$ is linear in terms of regression as the second derivatives with respect to α, β, γ are zero. However, G is nonlinear in X as its second derivative with respect to X is non-zero. A few more examples could be $G_1 = \frac{1 + aX}{b}$, and $G_2 = aX^b$. G_1 is nonlinear in parameters but linear in X while G_2 is nonlinear in both the parameters and X.

Consider a model given by

$$Y_i = a + bX_i \tag{2}$$

where (X_i, Y_i) represent any of the 'N' experimental data pairs, and a, b are the true process parameters. Due to uncertainty in the input and output measurements, the apparent experimental data pair (X_i, Y_i) is actually due to the true values and the error associated with it, i.e.,

$$X_T = X_i \pm \delta_i, \quad \delta_i \sim N(0, \sigma_{\delta_i}) \tag{3}$$

$$Y_i = Y_T \pm \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma_{\varepsilon_i}) \tag{4}$$

where (X_T, Y_T) represent the true but unknowable measurements, and δ_i, ε_i represent the corresponding errors as a normal distribution with a mean of zero and variance $\sigma_{\delta_i}, \sigma_{\varepsilon_i}$ respectively. Schematically this may be summarized as

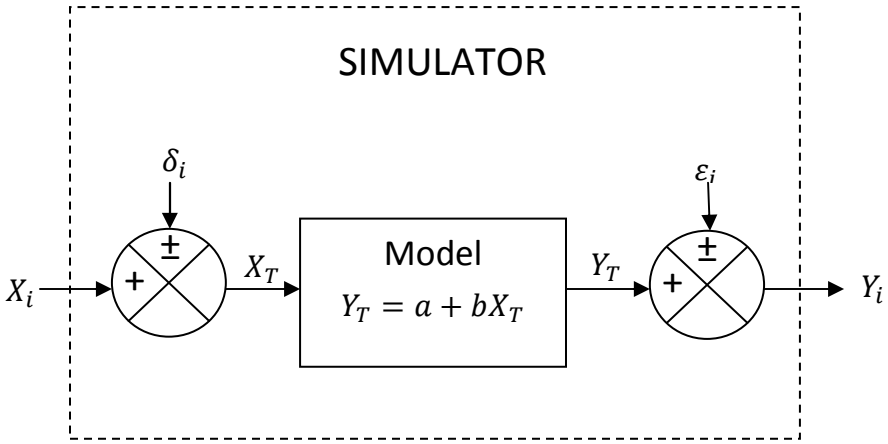


Figure 2: A prototype process simulator

The regression method employed would yield a model of the type

$$\hat{Y}_i = \hat{a} + \hat{b}\hat{X}_i \tag{5}$$

where $(\hat{X}_i, \hat{Y}_i)$ represent the corresponding model data pair, and $\hat{a}, \hat{b}$ the model parameter estimates. Illustrated below are a few of the most commonly used regression methods that could be employed to estimate $\hat{a}$ and $\hat{b}$.

1.1.1.1 Least Squares Regression – Vertical distances

Regressing Y on X, the square of the vertical distances between the experimental and model data pairs are minimized, i.e.,

$$\min_{\{a,b\}} J = \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2 \tag{6}$$

Analytically, the model parameters can be estimated by equating the derivative of the objective function with respect to the parameters to zero, resulting in [3]

$$\hat{a} = \frac{\left(\sum {X_i}^2\right) \cdot \left(\sum Y_i\right) - \left(\sum X_i Y_i\right) \cdot \left(\sum X_i\right)}{N\left(\sum {X_i}^2\right) - \left(\sum X_i\right)^2} \tag{7}$$

$$\hat{b} = \frac{N\left(\sum X_i Y_i\right) - \left(\sum X_i\right) \cdot \left(\sum Y_i\right)}{N\left(\sum {X_i}^2\right) - \left(\sum X_i\right)^2} \tag{8}$$

In a more simplified manner [4, 5], Eq. (7) and (8) can be written as

$$\hat{b} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sum (X_i - \bar{X})^2} = \frac{S_{XY}}{S_{XX}} \tag{9}$$

$$\hat{a} = \bar{Y} - b\bar{X} \tag{10}$$

where $\bar{X}$ and $\bar{Y}$ represent the mean of the experimental input and output measurements.

However, with the availability of faster computation techniques, numerical methods to optimize the objective function by varying the parameter values are widely used. The numerical methods iteratively minimize the objective function by varying the parameter values through an optimization routine.

The least squares regression is believed to provide better parameter estimates when the uncertainty in the input variable is minimal, or negligible, compared to the uncertainty in the output variable.

1.1.1.2 Least Squares Regression – Horizontal distances

While the conventional least squares regression minimizes the vertical distances, regressing X on Y minimizes the horizontal distances. This method is suitable when the uncertainty in the input variables is much larger compared to the uncertainty in the output variable, as the output variable is assumed to be perfectly known. The model equation is transformed to the type [6]

$$\hat{x} = \hat{a} + \hat{b}\hat{y} \tag{11}$$

And the parameter estimates by the analytical method are obtained by interchanging x and y in Eq. 7 and 8 as [6]

$$\hat{a} = \frac{\left(\sum {Y_i}^2\right) \cdot \left(\sum X_i\right) - \left(\sum X_i Y_i\right) \cdot \left(\sum Y_i\right)}{N\left(\sum {Y_i}^2\right) - \left(\sum Y_i\right)^2} \tag{12}$$

$$\hat{b} = \frac{N\left(\sum X_i Y_i\right) - \left(\sum X_i\right) \cdot \left(\sum Y_i\right)}{N\left(\sum {Y_i}^2\right) - \left(\sum Y_i\right)^2} \tag{13}$$

1.1.1.3 Perpendicular regression

Due to the limited applicability of the least square regression techniques, methods like the perpendicular regression were developed. The perpendicular distances can traditionally be minimized by two approaches

1. Minimizing the perpendicular offsets [3, 7]
2. Shifting the coordinate axis by an angle such that minimizing the vertical distances hence is equivalent to minimizing the perpendicular distances [8].

1.1.1.3.1 Perpendicular offsets

This method involves minimizing the square of the perpendicular distances between the experimental and model data pairs. If (X_i, Y_i) is an experimental data pair, then its perpendicular distance to the Model (Eq. 2) is given by [3, 7]

$$d = \frac{\left|Y_i - \left(\hat{a} + \hat{b}X_i\right)\right|}{\sqrt{1 + \hat{b}^2}} \tag{14}$$

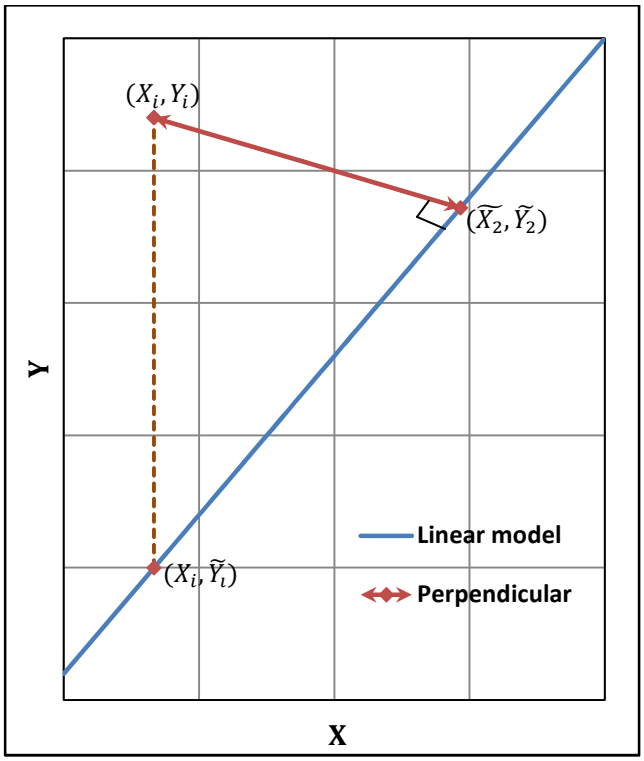


Figure 3: Perpendicular distance – Linear regression

In Fig. 3, $(\widetilde{X_2}, \widetilde{Y_2})$ is the foot of the perpendicular to (X_i, Y_i). However, minimizing the perpendicular distances does not necessarily require finding the foot of the perpendicular if the point (X_i, Y_i) and the model coefficients are known, as can be seen from Eq.14. The objective function is the same as minimizing the vertical distances but with the denominator term, i.e. [3, 7],

$$\min_{\{a,b\}} J = \sum_{i=1}^{N} \left(\frac{Y_i - (a + bX_i)}{\sqrt{1 + b^2}} \right)^2 = \sum_{i=1}^{N} \frac{(y_i - \hat{y})^2}{1 + b^2} \tag{15}$$

Hence, for the experimental point(X_i, Y_i) , a point $(X_i, \tilde{Y}_i)$ is estimated based on the model parameters and hence the Objective Function (Eq. 15) is evaluated. Alternatively, the perpendicular distance can be evaluated by determining the foot of the perpendicular $(\widetilde{X_2}, \widetilde{Y_2})$, and hence its distance from (X_i, Y_i).

The analytical estimation of the model parameters is given by [3]

$$\hat{a} = \frac{(1-r)\cdot\left(\sum {X_i}^2\right)\cdot\left(\sum Y_i\right) - \left(\sum X_i Y_i\right)\cdot\left(\sum X_i\right)}{N(1-r)\cdot\left(\sum {X_i}^2\right) - \left(\sum X_i\right)^2} \tag{16}$$

$$\hat{b} = \frac{N\left(\sum X_i Y_i\right) - \left(\sum X_i\right)\cdot\left(\sum Y_i\right)}{N(1-r)\cdot\left(\sum {X_i}^2\right) - \left(\sum X_i\right)^2} \tag{17}$$

where

$$r = \frac{\sum\left(Y_i - \tilde{Y}\right)^2}{(1+b^2)\cdot\left(\sum {X_i}^2\right)} \tag{18}$$

In a more simplified form the slope estimate $\hat{b}$, could be written as [4]

$$\hat{b} = \frac{S_{YY} - S_{XX} + \sqrt{(S_{YY} - S_{XX})^2 + 4{S_{XY}}^2}}{2S_{XY}} \tag{19}$$

where $S_{XX} = \sum(X_i - \bar{X})^2$, $S_{XY} = \sum(X_i - \bar{X})(Y_i - \bar{Y})$. The estimate of the intercept can be obtained as $\hat{a} = \bar{Y} - \hat{b}\bar{X}$.

1.1.1.3.2 Shifting of axes

Another approach of perpendicular regression is to shift the primary direction of the coordinate axes by an angle such that minimizing the vertical distances hence is the same as minimizing the perpendicular offsets with respect to the original direction.

The method begins by initializing the parameters of the Model (Eq. 2), by minimizing the vertical distances. The parameters hence are denoted as a_0, b_0, where the subscript '0'

represents the initial value of the parameters before shifting the axes. If the slope b_0 equals zero, the line is horizontal, and minimizing the vertical distances hence is the same as minimizing the perpendicular distances. Hence the axes are continually shifted by certain angles until this condition is satisfied, and once this is achieved, the parameter values with respect to the original axes are the required estimates.

The axes are shifted initially by an angle θ_{y0} such that vertical or the dependent variable axis aligns itself with the perpendicular ($\overrightarrow{n_0}$) to the initial model. For the parameters a_0, b_0 the angle is determined by the direction cosines of $\overrightarrow{n_0}$ as [8]

$$n_{x0} = \cos\theta_{x0} = \frac{-b_0}{\sqrt{1 + {b_0}^2}} \tag{20}$$

$$n_{y0} = \cos\theta_{y0} = \frac{1}{\sqrt{1 + {b_0}^2}} \tag{21}$$

where n_{x0} and $n_{y0}$are the component vectors of $\overrightarrow{n_0}$.

The transformed experimental data pair (x_i, y_i) is given by [8]

$$\begin{Bmatrix} \hat{x_i} \\ \hat{y_i} \end{Bmatrix} = \begin{bmatrix} \cos\theta_{y0} & -\cos\theta_{x0} \\ \cos\theta_{x0} & \cos\theta_{y0} \end{bmatrix} \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} = M_0 \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} \tag{22}$$

Or, from the expressions for direction cosines from Eqs. (20) and (21), [8]

$$\hat{x}_i = \frac{1}{\sqrt{1 + {b_0}^2}} (x_i + b_0 y_i) \tag{23}$$

$$\hat{y}_i = \frac{1}{\sqrt{1 + {b_0}^2}} (y_i - b_0 x_i)$$

(24)

Since the slope $b_0 \neq 0$, the parameter values are re-estimated by minimizing the vertical distances with respect to the new coordinate system and the process is repeated until $b_0 = 0$.

If $\theta_{y0}, \theta_{y1}, .. \theta_{yn}$ are the angles by which the axes were rotated for each iteration, the final transformed experimental data pair is given by [8]

$$\begin{Bmatrix} \hat{x_i} \\ \hat{y_i} \end{Bmatrix} = \begin{bmatrix} \cos\theta_{yn} & -\cos\theta_{xn} \\ \cos\theta_{xn} & \cos\theta_{yn} \end{bmatrix} \ldots \begin{bmatrix} \cos\theta_{y0} & -\cos\theta_{x0} \\ \cos\theta_{x0} & \cos\theta_{y0} \end{bmatrix} \begin{Bmatrix} x_i \\ y_i \end{Bmatrix} = M_0 \begin{Bmatrix} x_i \\ y_i \end{Bmatrix}$$

(25)

And to get back to the original coordinate system, the transpose of the cumulative rotation matrix is multiplied to unit normal vector $[0\ 1]^T$, to obtain the normal vector $\vec{n}$ to the final model. [8]

$$\vec{n} = \begin{Bmatrix} n_x \\ n_y \end{Bmatrix} = {M_c}^T . \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}$$

(26)

And hence the parameter estimates, [8]

$$\hat{a} = \frac{a_m}{n_y}, \qquad \hat{b} = \frac{-n_x}{n_y}$$

(27)

However this approach of perpendicular regression is a lot more complex than minimizing the perpendicular offsets. Hence unless this method has any advantages of faster convergence to the optimum or lesser iterations, which hasn't been cited by the author, it may not seem reasonable to select this approach.

1.1.1.4 Geometric Mean Functional Relationship

This method is another effort to overcome the pitfalls of the least squares method. The method requires to first regress Y on X to get the ordinary least square estimate of the slope as $\hat{b} = S_{XY}/\, S_{XX}$ [4, 5, 9]. Then regress X on Y and re-write the equation in the form of Y on X to get another estimate of the slope, $\hat{l} = S_{YY}/\, S_{XY}$ [4, 5, 9]. The GMFR estimate of the slope then is [4, 5, 9]

$$\hat{b}_{GMFR} = sign(S_{XY})\sqrt{\hat{b}\hat{l}} = \; sign(S_{XY})\sqrt{\frac{S_{YY}}{S_{XX}}} \tag{28}$$

The estimate of the intercept can be obtained as $\hat{a} = \bar{Y} - \; \hat{b}\bar{X}$.

As reported in [5], this method minimizes the sum of the geometric mean of squared vertical and horizontal distances of each experimental point to the model line. When the uncertainty in the input and output measurements are due to the errors and not due to the randomness in the variable itself, the parameter estimates for this method are the same as the one for maximum likelihood, and the method is best suited for this case [4].

1.1.1.5 Maximum Likelihood

The maximum likelihood is a more generic regression technique that maximizes the combined likelihood probabilities of the experimental data pairs. However, the principle of maximum likelihood is mostly used in estimating the unknown parameters of a distribution and its application to regression techniques has been limited. For the Linear Model (Eq. 2), the analytical estimate of the slope is determined by finding the likelihood probabilities of individual data points by the probability density function and equating the

derivative of the combined probability with respect to the parameters to zero. The slope estimate obtained hence is given by [4]

$$\hat{b} = \frac{S_{YY} - \lambda S_{XX} + \sqrt{(S_{YY} - \lambda S_{XX})^2 + 4\lambda {S_{XY}}^2}}{2S_{XY}} \tag{29}$$

A detailed explanation on this method for nonlinear regression, a more general case, is addressed in the later part of the report.

It is important to note that when the variances in the input and output measurements are the same, i.e., $\lambda \, (= {\sigma_\varepsilon}^2 / \, {\sigma_\delta}^2 \,) = 1$, the perpendicular regression parameter estimates are the same as the parameter estimates for the maximum likelihood method (Refer Eqs. 19 and 29) and hence the method is best suited in this case [4]. This has also been found true for nonlinear regression discussed later in the report.

1.1.2 NONLINEAR REGRESSION

The shortcomings of the least squares regression remain the same for both the linear and the nonlinear regression. And to overcome this, a few efforts were made by researchers to develop methods that were computationally realizable and understandable to an engineer's intellect. One among them is the Taylors series approximation to evaluate perpendicular distances as cited in [10].

The author, Akaho [10] used a first-order Taylor series approximation to determine a tangent at a point on the assumed model curve, vertically above/below the experimental data point, and then determine the foot of the perpendicular from the experimental point to the tangent.

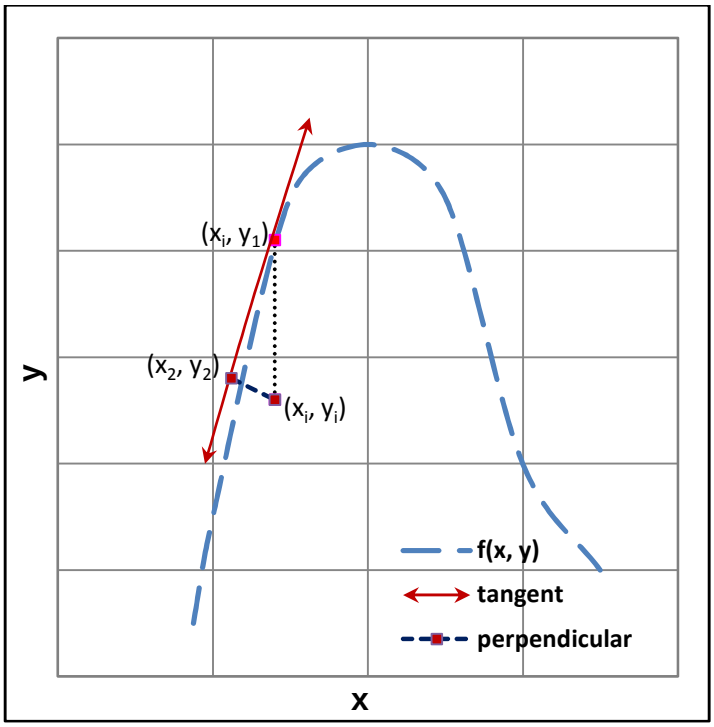


Figure 4: Perpendicular regression by Akaho's method

For any experimental data point (x_i, y_i) close to the assumed model curve $f(x, y)$ as shown in Fig. 4, the first order Taylor series approximation is given by

$$f(x,y) \cong f(x_i,y_i) + f_x(x_i,y_i) \cdot (x - x_i) + f_y(x_i,y_i) \cdot (y - y_i) \tag{30}$$

If (x, y) is a point on the curve,

$$f(x,y) = f(x) - y = 0 \tag{31}$$

$$f(x,y) \cong f(x_i,y_i) + f_x(x_i,y_i) \cdot (x - x_i) + f_y(x_i,y_i) \cdot (y - y_i) = 0 \tag{32}$$

Equation (32) represents a tangent to the curve at the point (x_i, y_1). For the point (x_i, y_i), Akaho's method requires finding a point (x_2, y_2) along this tangent which is the foot of the perpendicular on it from (x_i, y_i). This approach holds good when there are no significant changes in the slope of the assumed curve over the uncertainty range on 'x_i', as the tangent would remain relatively the same.

But, for large changes in the slope of the curve as shown in Fig. 5, the Taylor series approximation for the point (x_i, y_i) will result in a tangent at a point (x_i, y_1) on the curve, and the foot of the perpendicular evaluated along this tangent would be (x_2, y_2) instead of (x, y), the actual foot of the perpendicular. This distortion of perpendicular distances is a drawback of Akaho's method.

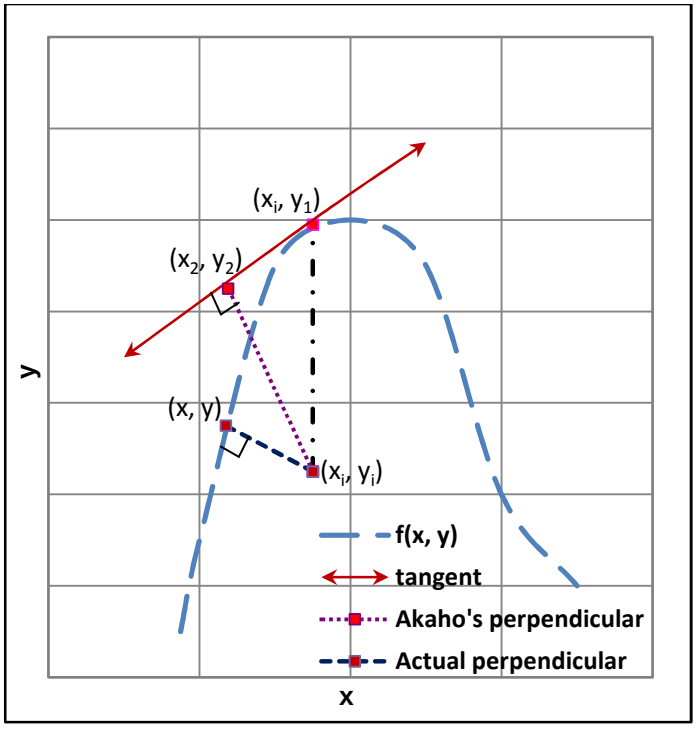


Figure 5: Drawback of Akaho's method

1.2 RESEARCH STATEMENT

Based on the above literature review it was established that though several regression techniques were devised for least squares regression, each method had its own limitations. The maximum likelihood method seemed the most promising of all, but the research was mostly restricted to linear regression.

In our study, we intended to analyze the shortcomings the least square regression through Monte Carlo simulations and explore alternative regression techniques including the maximum likelihood approach for nonlinear regression.

2. METHOD

This work evaluates the quality of a regression method by its accuracy and consistency in predicting the regression parameter values. The closeness of the parameter values estimated through a regression method, to the true values, determines the accuracy of the method. Consistency may be established through repeated estimations of the parameter values for different sets of data realizations (A data realization is a realistic approximation of experimental data for computer simulations). The work also evaluates the practicability of a regression method by computational load and user complexity.

The methods analyzed were — the least squares method, the normal distance method, and maximum likelihood method. The least squares method is the conventional method adopted for regression. The maximum likelihood method was developed through our research, and the normal or the perpendicular distance method is just a consequence of the maximum likelihood method when the uncertainties in the input and the output variables, σ_x and σ_y, are the same.

Monte Carlo simulations were run for a weak acid-strong base titration experiment and a packed bed reactor experiment. As cited in [9], "in Monte Carlo studies, the investigator plays God by choosing the true values of the regression parameters, which in real life, can only be estimated from error-prone data. The investigator then programs a computer to simulate an experiment by adding random errors to the true values of X

and Y," and tests the regression capabilities of different methods based on their estimate of the parameter values.

2.1 SCOPE OF THE STUDY

The scope of this project is:

1. **Number of regression parameters, input and output variables**: The titration simulation comprised of two regression parameters (A_0, pK_a), with the input and output variables being the volume of base added (V) and the pH of the solution respectively. The packed bed simulation was more complex with three regression parameters $-$ k_0, k_1, E, three input variables $-$ the inlet concentration (C_{in}), the volumetric flowrate (ν), and the reactor temperature (T), and the outlet concentration (C_0) as the single output variable.

2. **Distribution of the errors**: The input and output variables were assumed to follow normal independent distributions with a mean of zero.

Note: In the discussion henceforth, the input and output error variances σ_δ, σ_ε, are denoted by σ_x and σ_y respectively.

2.2 EXPERIMENT

2.2.1 TITRATION

The titration experiment chosen to test the methods is usually known for its nonlinearity in the pH values vs. the volume of titrant added, and hence is a good test for the robustness of any regression technique. The experiment was simulated in Microsoft Visual Basic based on the equations governing the titration process. Base was presumed to be added dropwise to the batch of acid, and the pH of the mixture changed instantly. The volume readings of the base added were the input to the process, and the corresponding pH values were the output. Uncertainty was included in both the input and output measurements to simulate reality. The theoretical equations were derived as follows:

Let A_0 and B_0 be the initial acid and base concentrations. Let 'v' ml of the base be added to a liter of the acid solution. Due to increase in volume of the mixture through addition of base, the acid concentration decreases to 'a' mol/lit and base concentration to 'b' mol/lit.

$$a = \frac{A_0}{\left(1 + \frac{v}{1000}\right)} \frac{mol}{lit} \tag{33}$$

$$b = \frac{B_0 \cdot \frac{v}{1000}}{(1 + \frac{v}{1000})} \; mol/lit \tag{34}$$

Let 'x' moles of acid and 'y' moles of water dissociate upon addition of the base. Since it's a dilute acid, the water content is large compared to acid and virtually remains unaffected despite the dissociation.

Under equilibrium,

$$(a-x)HA \rightleftharpoons xH^+ + xA^- \tag{35}$$

$$(\infty - y)H_20 \rightleftharpoons yH^+ + yOH^- \tag{36}$$

$$(b-b)BOH \rightleftharpoons bB^+ + bOH^- \tag{37}$$

Hence, the total hydrogen ion concentration: $[H^+] = (x+y)$ (38)

Concentration of A^-: $[A^-] = x$ (39)

Concentration of OH^-: $[OH^-] = y + b$ (40)

Concentration of the undissociated acid, HA: $[HA] = a - x$ (41)

The dissociation constants for the acid and water from Eqs. (35) and (36) are

$$k_a = \frac{[H^+][A^-]}{[HA]} \tag{42}$$

$$k_w = [H^+][OH^-] \tag{43}$$

The dissociation constant of the water has a nominal value of 10^{-14}.

pH of any aqueous solution is defined as

$$pH = -\log_{10}[H^+] \tag{44}$$

Substituting the variables in the above three equations gives

$$k_a = \frac{(x+y)\cdot x}{(a-x)} \tag{45}$$

$$k_w = (x+y)(y+b) \tag{46}$$

$$pH = -\log_{10}(x + y) \tag{47}$$

Analytically, combining Eqs. (45), (46) and (47) to evaluate the pH corresponding to the volume of base added gives a cubic expression in either x or y. Only one of the three roots would be the desired value, which, may not be easy to determine.

Alternatively, numerical root finding methods such as the interval halving method can be implemented by guessing a value of pH and determining the corresponding pH value calculated through the above equations. If the calculated pH is the same as the initial guess, then it is the desired root. The numerical method successively iterates with the calculated pH value as the new guess until both the values are the same. The method is implemented in the following manner:

1. Guess a value of pH between 0 and 14: pH_g

2. Calculate the total hydrogen ion concentration corresponding to pH_g,

$$[H^+] = (x + y) = 10^{-pH_g} \tag{48}$$

3. From (45) and (48): $x_c = \frac{a \cdot K_a}{K_a + 10^{-pH_g}}$

4. From (46) and (48): $y_c = k_w \cdot 10^{pH_g} - b$

5. Calculated total hydrogen ion concentration: $(x_c + y_c)$

6. Calculated pH: $pH_c = -\log_{10}(x_c + y_c)$

The desired root would be the pH guess (pH_g) that makes its difference with the calculated value zero, or $f = pH_g - pH_c = 0$. However, since the calculated total

hydrogen ion concentration ($x_c + y_c$) can be negative at times for a bad pH guesses, and the logarithm of negative numbers is not defined, f is alternatively defined as the difference of the total acid concentration for the initial and calculated pH values, and the pH value (pH_g) that makes $f = (x + y) - (x_c + y_c) = 0$ is the desired root.

Hence the pH of the solution for a given set of parameter values, and the volume of the base added can be determined through root finding techniques. The advantage of this method is: (i) There is a unique solution between 0 and 14. (ii) Each step in the procedure is an explicit calculation.

2.2.1.1 Interval halving root finding technique

The interval halving method begins by localizing the root between two limits. For a function $f(x)$ shown in Fig.6, the root a_3 lies between a_1 and b_1. The midpoint of a_1 and b_1, b_2 is evaluated. Since $f(a_1)$ and $f(b_2)$ are of opposite signs, the root is now bracketed between these two limits. The process of midpoint calculation of the limits and bracketing the root continues until the interval size reduces to the desired limit.

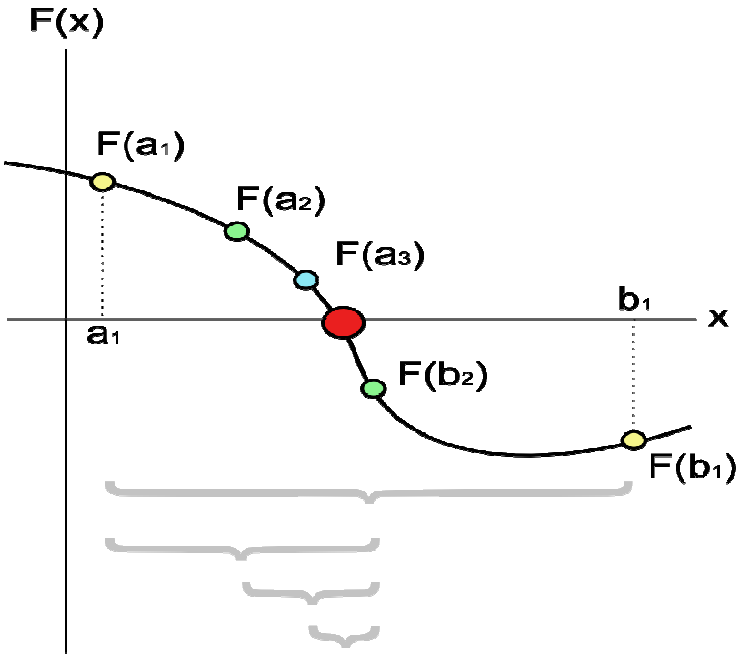


Figure 6: Interval Halving Method (Reproduced from [12])

(Note: the notations, a_1, a_2, a_3, b_1, b_2, do not correspond to the parameters of the Linear Model (Eq. 2)).

For the titration process, the interval halving method is applied with pH_g as the search variable, and the function f defined as $f = (x + y) - (x_c + y_c)$. A typical relation between f *vs.* pH_g for a given set of parameter values and volume is shown in Fig. 7.

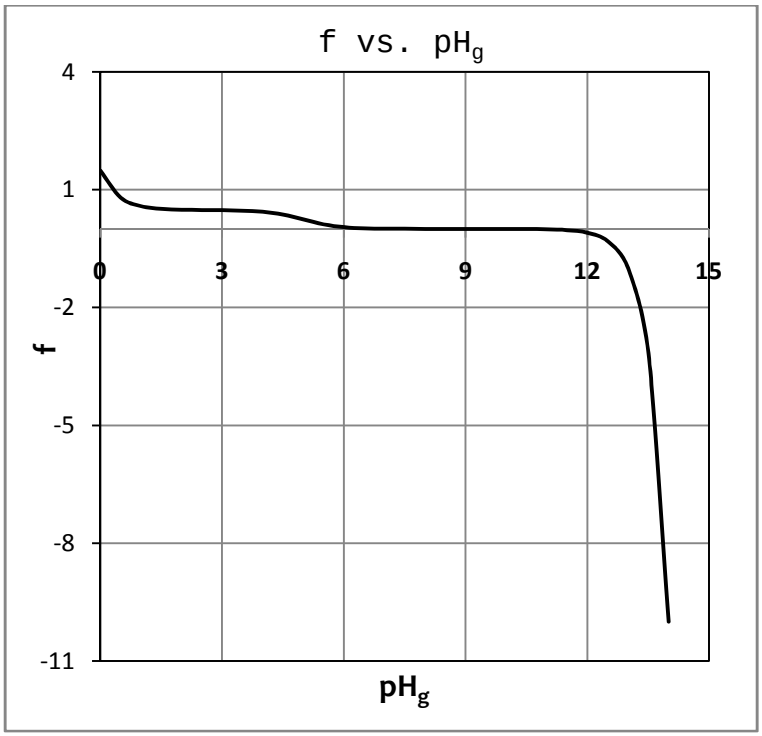


Figure 7: $f\ vs.\ pH_g$ – Titration experiment

Since the nominal pH of any substance lies between 0 and 14, these are chosen are the minimum and maximum bounds. Subsequent reductions in the interval size based on the interval halving method were continued for twenty iterations, i.e., until the interval size reduces to an order of 10^{-5}. $(Range = (\Delta X)\left(\frac{1}{2}\right)^N = (14-0)(0.5)^{20} = 1.3E-5)$

2.2.1.2 Data generation

The experimental data were generated by adding uncertainties to the nominal values of the input and output variables. The uncertainties were presumed to follow Gaussian distributions, a standard statistical assumption [13].

A Gaussian distribution, also called the normal distribution is a continuous probability distribution of random variables and depicts a bell shaped pattern, symmetric on either side as shown in Fig. 8.

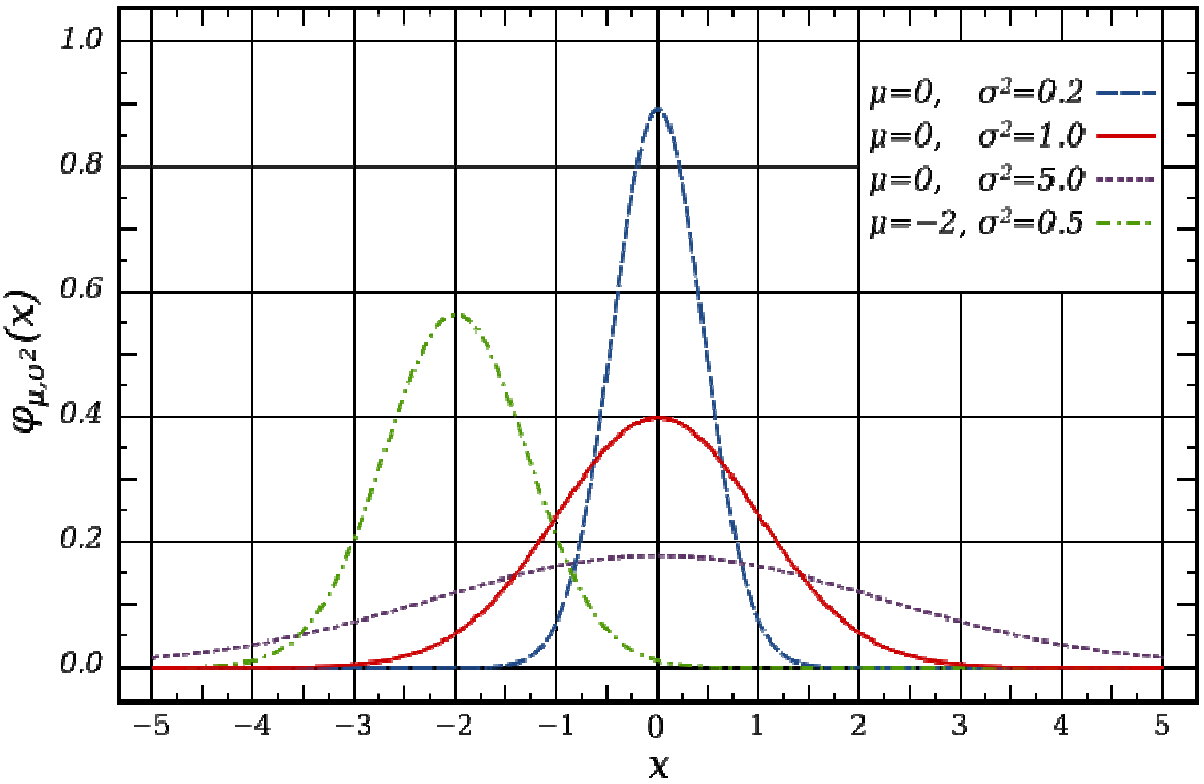


Figure 8: Normal distribution (Reproduced from [11])

It is a plot of the random variable vs. their corresponding probability density function values (φ) calculated as [11]

$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{49}$$

and is characterized by the mean (μ) and the variance (σ^2). The mean represents the average value of the distribution and the variance indicates the dispersion of the distribution, the higher the variance, the more dispersed the distribution. The distribution with $\mu = 0$ and $\sigma = 1$ is called the standard normal distribution.

A Normal Independent Distribution (NID), that is a normal distribution where each individual observation is independent of the other, with a mean of zero and standard

deviation of σ, is represented as NID(0, σ), and is a practical approximation to uncertainty associated with experimental measurements [2]. In Microsoft Visual Basic the noise can be approximated to NID(0, σ) through the following expression [3]

$$Noise = \sigma \cdot Sqr(-2 * Log(Rnd())) * Sin(2 * 3.14159 * Rnd()) \qquad (50)$$

The noise level can be varied by varying the magnitude of the σ value.

Fig. 9 shows a typical comparison of the actual data of the process with the apparent data known to us for $\sigma_x = \sigma_y = 0.25$.

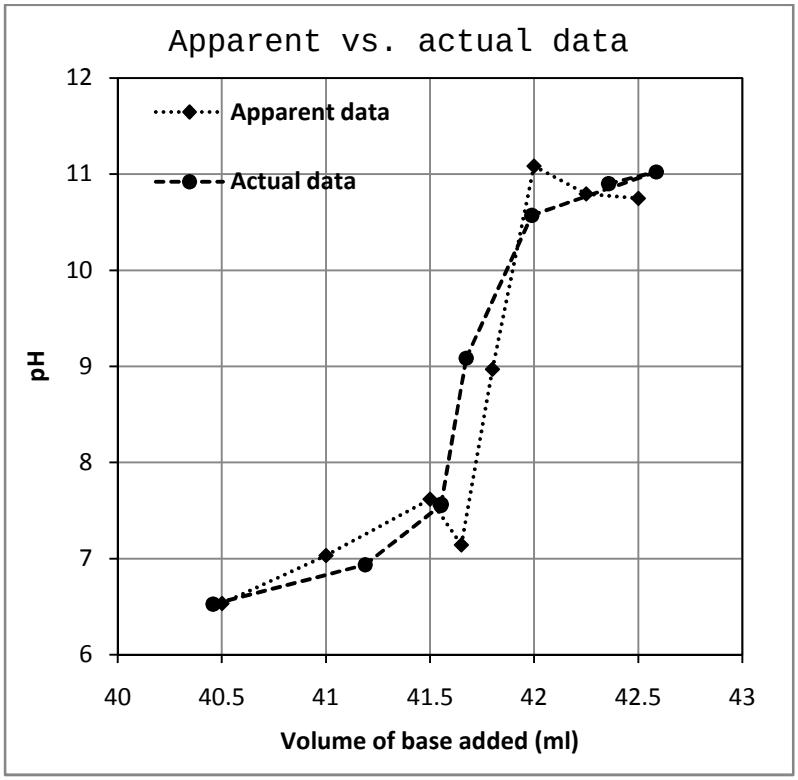


Figure 9: Apparent vs. actual data – Titration experiment

The actual data in the figure represents idealized data, the input and output values after, and before, the addition of their respective noises, or in other words, the true data fed into, and obtained from the process simulator. The apparent data represents the input and

output values before and after the addition of their respective noises, that is, the data evident to us.

The data generation process can be summarized as shown in Fig. 10.

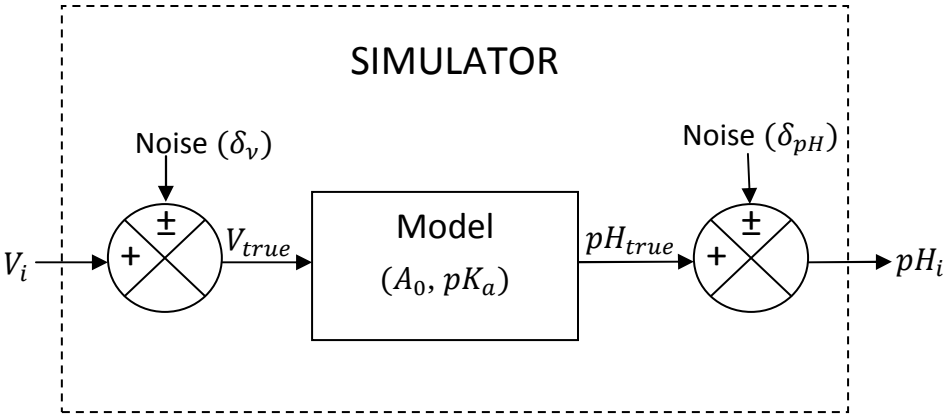


Figure 10: Data Generation – Titration experiment

The initial acid concentration (A_0) and the pK_a value of the acid were chosen as the regression parameters to be estimated. Since $pk_a = -\log_{10} K_a$, a small change in pK_a effects the dissociation constant (K_a) by several orders of magnitude and has a significant impact on the shape of the pH curve. The experimental data pairs i.e., the input-output readings were chosen to lie along, or close to the steep part of the curve as shown in Fig. 11 to ensure nonlinearly in the regression process. Heuristically, a minimum of three experimental data pairs are required per regression parameter. Hence, eight experimental data pairs were taken to determine the two model parameter values.

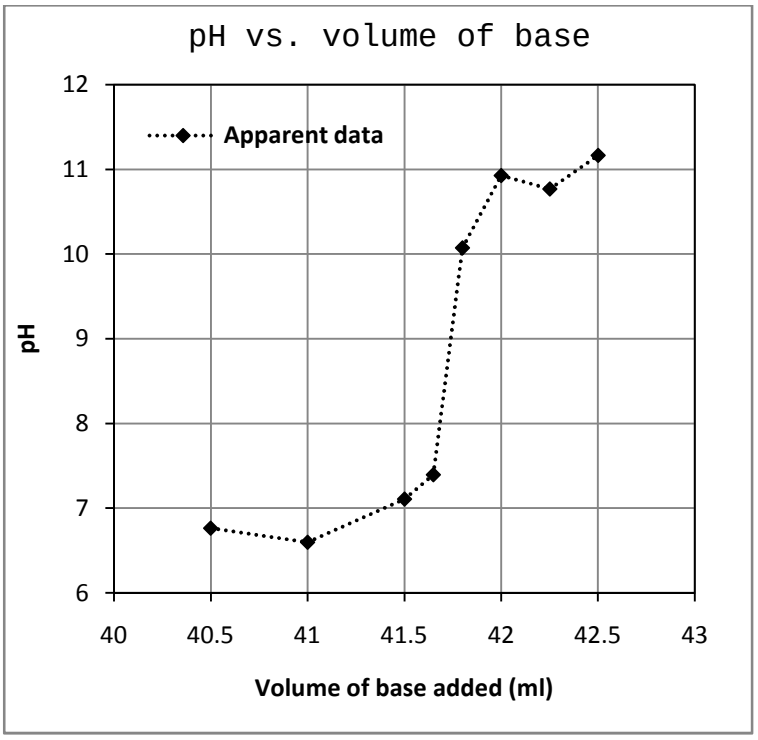


Figure 11: A typical pH curve for regression

For each set of experimental data, the parameter values were estimated by each of the three methods (vertical distance, normal distance, maximum likelihood). The process was repeated for different realizations of the eight-pair sets of data for a thousand trials, and the mean and the standard deviation of the distribution of the parameter values obtained by each of the method provided data for the comparison of the regression methods. The closer the mean to the true parameter value (accuracy) and lower the standard deviation (consistency), the better the method.

2.2.2 PACKED BED REACTOR

To illustrate chemical engineering applications of regression techniques, a packed bed reactor model was chosen. Packed bed reactors are a classification of a more generic type of continuous tubular flow reactors, called the plug flow reactors. They find important applications in catalytic processes, predominant in chemical industries.

For the simulation, a prototype chemical reaction with the kinetics defined as

$$A \xrightarrow{cat} R$$

$$(-r_A) = \frac{k_0 e^{-E/RT} C_A}{1 + k_1 C_A} \tag{51}$$

was chosen, where k_0 $(gmol/L\text{-}s)$ denotes the pre-exponential reaction rate coefficient, k_1 $(L/gmol)$ - the mass transfer coefficient, C_A $(gmol/lit)$ - the outlet concentration, E $(KJ/Kmol)$ - the activation energy of the reaction, R $(KJ/Kmol\text{-}K)$ – the gas constant, and T (K) – the reactor temperature. The reaction was presumed to take place under isothermal conditions following Hougan- Watson kinetics with the deactivation rate of the catalyst safely ignored. The contents within the reactor were assumed to be at steady state, following plug flow conditions.

The reactor model was derived through a mole balance on an elemental section of the reactor and integrating it over the entire length of the reactor.

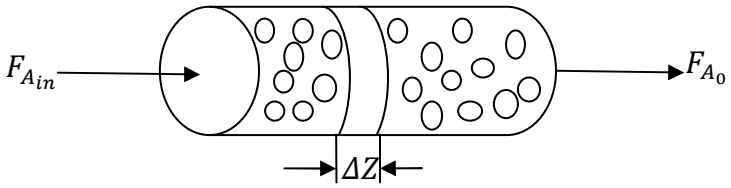


Figure 12: Packed bed reactor

For the reactor shown in Fig. 12, applying the mole balance about the element Z,

$$F_A|_Z - F_A|_{Z+\Delta Z} - \int (-\mathrm{r_A})\,\mathrm{dV} = \frac{dN_A}{dt} \quad (moles/sec) \tag{52}$$

At steady state,

$$\frac{dN_A}{dt} = 0 \tag{53}$$

Hence,

$$F_A|_Z - F_A|_{Z+\Delta Z} - \int (-\mathrm{r_A})\,(\pi \mathrm{r}^2)\,\mathrm{dZ} = 0 \tag{54}$$

Differentiating w. r. t Z,

$$\frac{dF_A}{dZ} = (-\mathrm{r_A})\,(\pi \mathrm{r}^2) \tag{55}$$

Substituting $F_A = \nu C_A$, where ν is the volumetric flowrate, and the rate equation (Eq. 51)

$$\nu \frac{dC_A}{dZ} = \left(\frac{k_0 e^{-E/RT} C_A}{1 + k_1 C_A}\right)(\pi \mathrm{r}^2) \tag{56}$$

Integrating over the entire length of the reactor, L,

$$\int_{C_{in}}^{C_0} \frac{1}{\left(\frac{k_0 e^{-\frac{E}{RT}} C_A}{1 + k_1 C_A}\right)} dC_A = \frac{(\pi \mathrm{r}^2)}{\nu} \int_0^{\mathrm{L}} \mathrm{dZ} \tag{57}$$

$$\int_{C_{in}}^{C_0}\left(\frac{1+k_1C_A}{C_A}\right)\,dC_A = \frac{(\pi r^2)}{\nu}(k_0e^{-\frac{E}{RT}})\int_0^{\mathrm{L}}\mathrm{dZ} \tag{58}$$

$$k_1(C_{in}-C_0)+\ln\left(\frac{C_{in}}{C_0}\right)=\frac{Vk_0}{\nu}e^{-E/RT} \tag{59}$$

Eq. (59) was the reactor model used in the simulation. The volume of the reactor, V, was assigned a value of 1000 mL. The inlet concentration, C_{in}, the volumetric flowrate, ν, and the reactor temperature, T were chosen as the input variables. The outlet concentration (C_0) was the output variable. Reaction kinetic coefficients k_0, k_1 and E were chosen as the regression parameters.

2.2.2.1 Output variable evaluation

The reactor model is an implicit equation and hence, the evaluation of outlet concentration for given input variables and model parameter values, requires the application of numerical root finding techniques. The Newton Raphson method was chosen, for the reasons explained below.

2.2.2.1.1 Newton-Raphson method

The Newton-Raphson method is based on linear approximation of small segments of a function to evaluate tangents that guide towards the root.

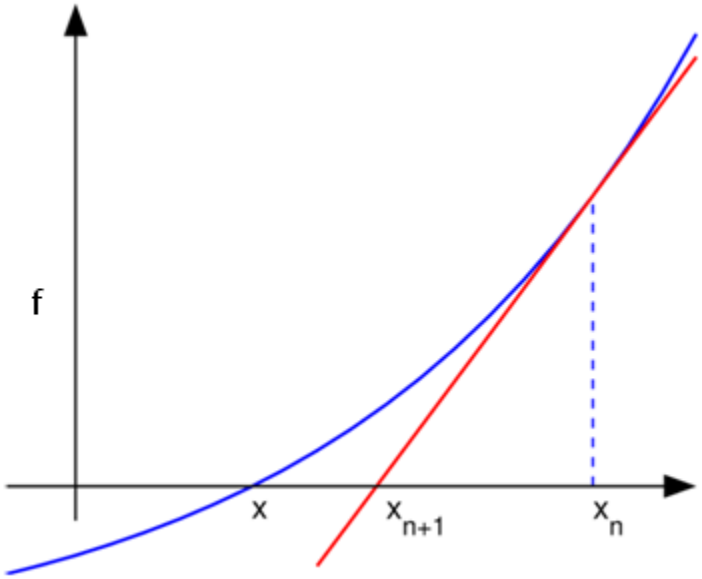


Figure 13: Newton-Raphson method (Reproduced from [14])

For the function shown in Fig. 13, the method is begun by initializing a guess, x_n, chosen such that successive values of 'x' progressively descend towards the root. At x_n, the function is linearly approximated by a tangent through the Taylors series expansion until the first derivative as

$$f(x_{n+1}) = f(x_n) + (x_{n+1} - x_n) \cdot f'(x_n) \tag{60}$$

The tangent is extended until it intersects the independent variable axis, and the point of intersection is the new guess.

Hence from Eq. 60,

$$f(x_{n+1}) = 0 \tag{61}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{62}$$

Eq. (62) is an iterative formula used to evaluate subsequent guesses by constantly updating the $x_n$value with the previous guesses. The process is repeated until $|x_{n+1} - x_n|$ reduces to the desired extent.

An analogous plot of $f\ vs. C_0$ for the packed bed reactor simulation, where f is written as

$$f = K_1(C_{in} - C_0) + \ln\left(\frac{C_{in}}{C_0}\right) - \frac{VK_0}{\nu}e^{-E/RT} \tag{63}$$

reveals a linear nature as shown in Fig. 14

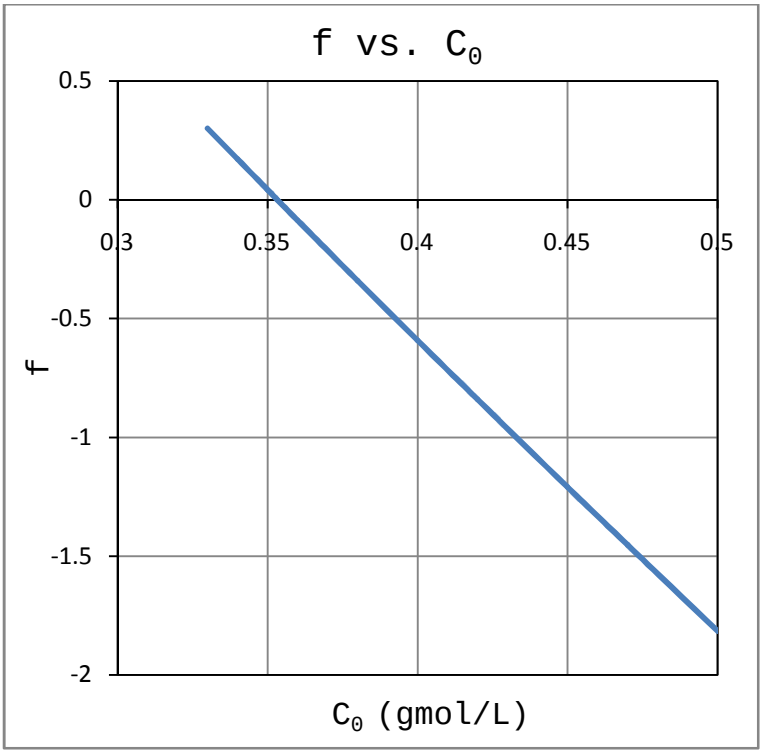


Figure 14: f vs. C_0 – Packed bed reactor simulation

Hence, the Newton-Raphson method, which is based on the linear approximation of a function, converges fast to the desired root.

The Newton-Raphson method was applied with the inlet concentration value as the initial guess and the method was continued until the absolute value of the difference of the subsequent guesses of C_0 was less than 10^{-10}. The central finite method was applied to evaluate the derivatives. A flowchart of the method is shown in Fig 15.

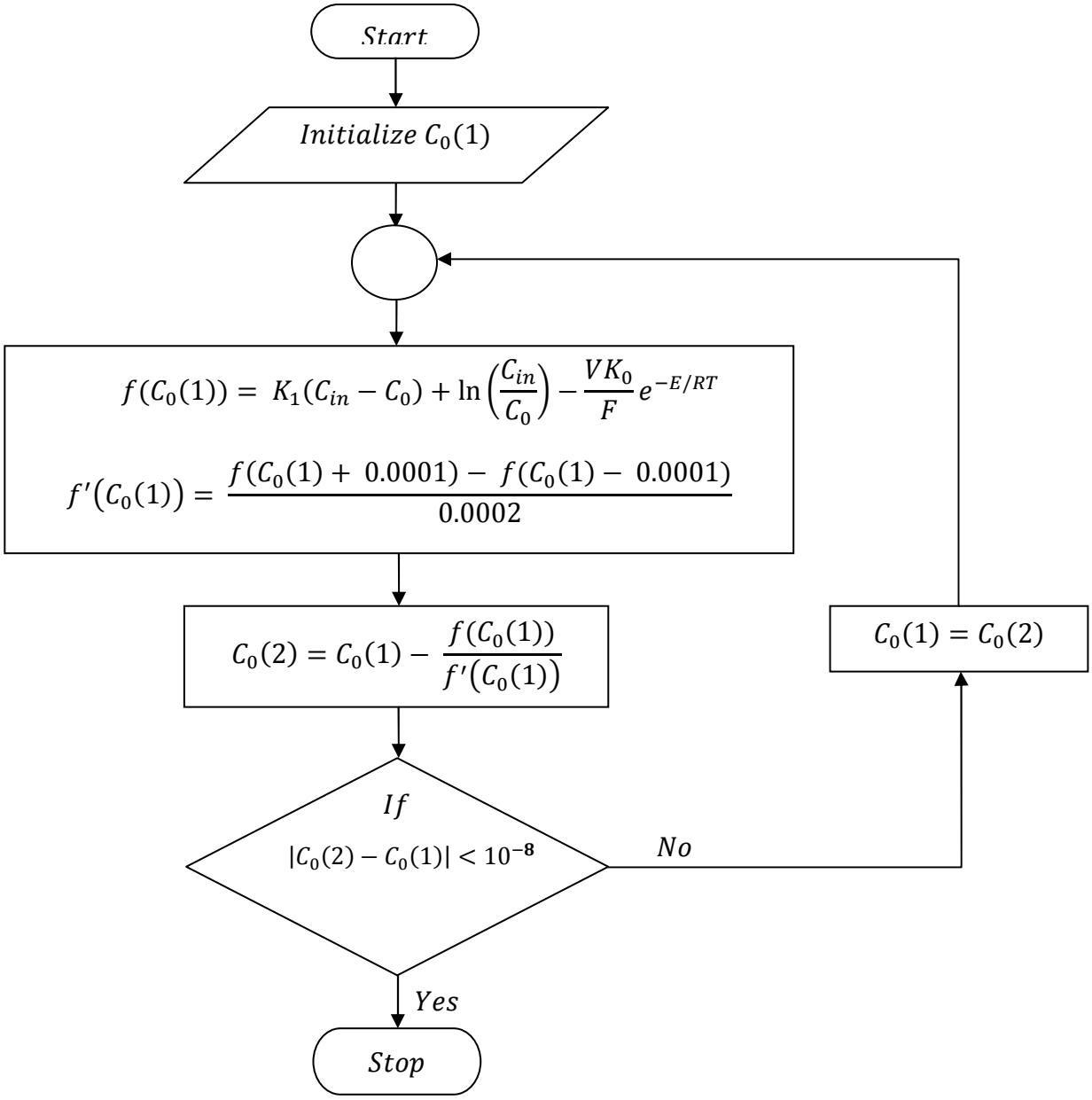


Fig. 15: Newton Raphson flowchart

2.2.2.2 Experimental data

The input values to generate experimental data for the packed bed reactor simulation were chosen in accord with the standard 'Design of Experiments' (DOE) procedure. That is, the repeated measurements of the input and output variables under similar conditions, to ensure accuracy. A typical plot for the inlet and outlet concentrations of the experimental data is shown in Fig.16.

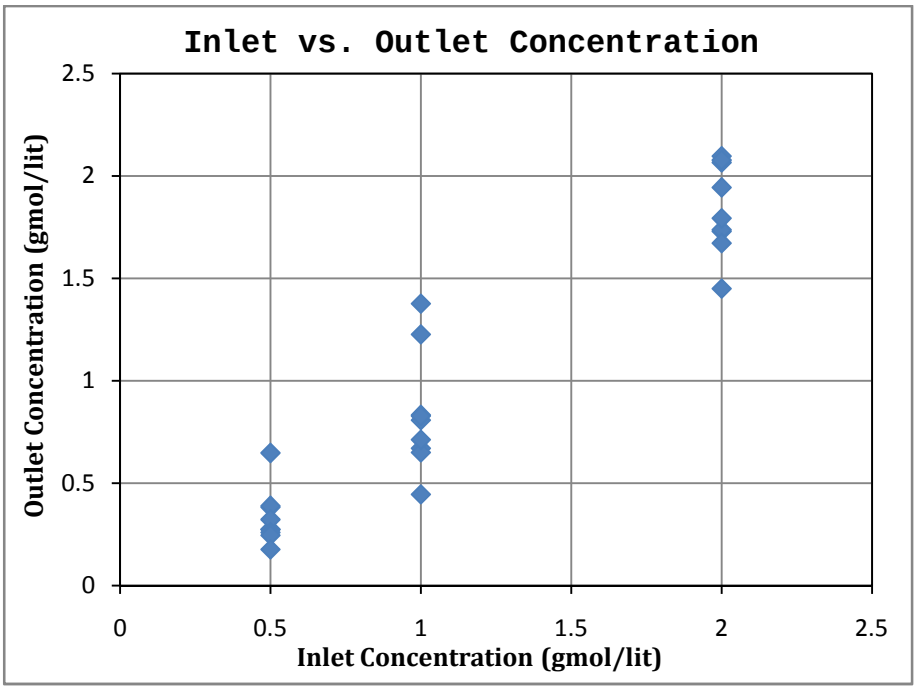


Figure 16: A typical inlet vs. outlet concentration plot for the PBR simulation

The DOE however, with the input values not spread out over the entire range, and confined to certain fixed values, does not provide a good data set as the data pattern along the unspecified region is unknown. But since practical experiments conform to the DOE procedure for the ease of measurements, this was adopted.

The experimental data were generated in a similar fashion as the titration experiment. Fig. 17 summarizes the data generation process.

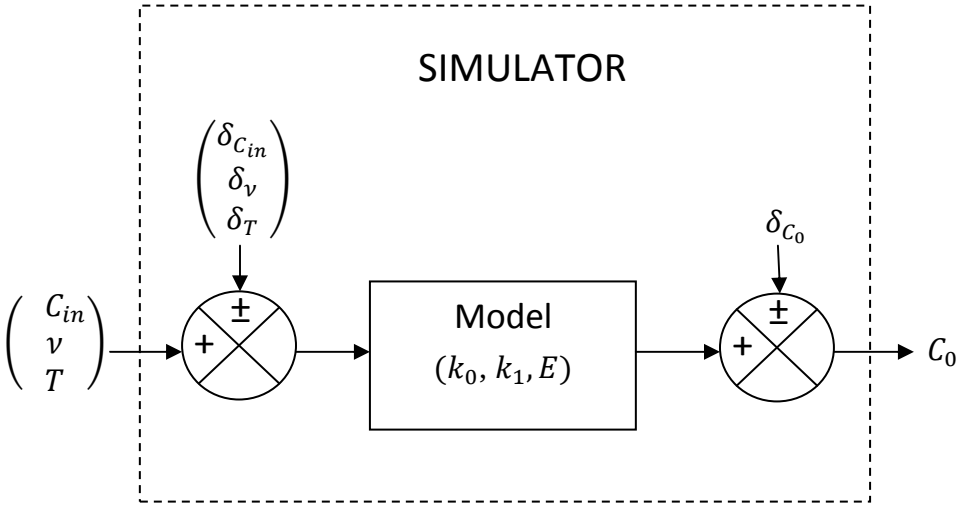


Figure 17: Data generation – Packed bed reactor

The uncertainty values for the input and output variables, for both the titration and the packed bed reactor simulation, were decided based on practical guesses of what the maximum error associated with each of the variables possibly could be.

For instance, if the maximum error in concentration measurement is presumed to be $0.5 gmol/lit$, and the certainty of getting this as the maximum value is 99% of the total measurements, then from the normal distribution plot shown in the Fig.18, 3σ would approximately correspond to $(34.1 + 13.6 + 2.1) * 2 \cong 99.7\%$ of the range, hence approximating $0.5 gmol/lit$ to 3σ, we get σ = 0.5/3 = 0.167

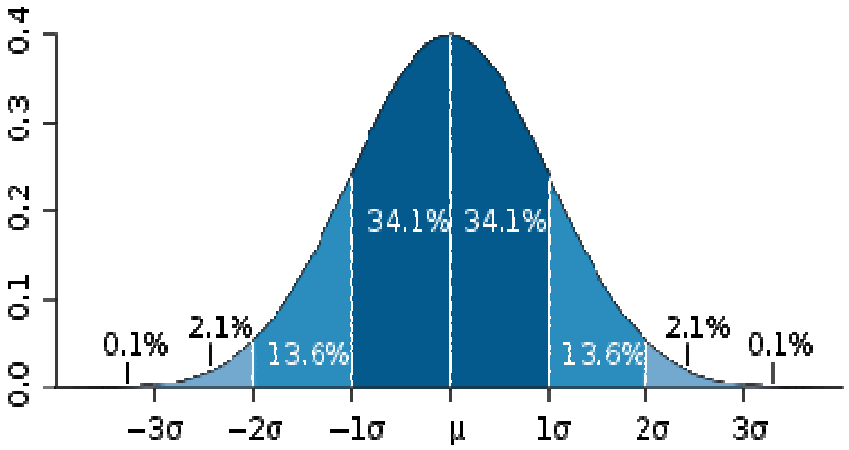


Figure 18: Normal distribution – Standard deviation and Confidence intervals (Reproduced from [11])

The parameter values were estimated by all the three regression methods for varying uncertainty levels in the input and output variables. The results were analyzed on the basis of the mean and the standard deviation of the parameter values for a thousand trials.

2.3 REGRESSION METHODS

2.3.1 VERTICAL DISTANCE METHOD

The vertical distance method assumes that the uncertainty in an experimental reading is only through the output measurements and no error exits in the input readings. That is, the apparent input measurements are presumed to be true values inputted to the process (or simulator for this study) and the only uncertainty that exits through the output measurements are minimized through vertical distances. Hence, theoretically this method holds good only when σ_x is negligible compared to σ_y.

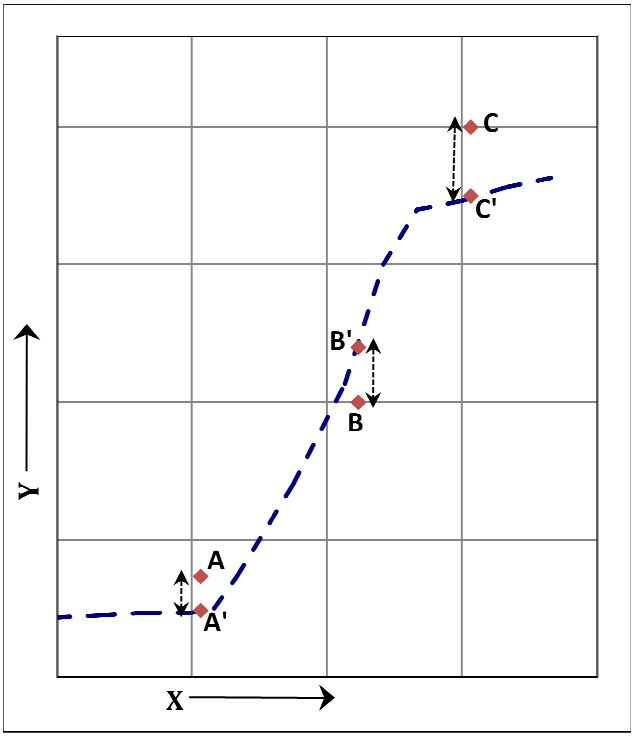


Figure 19: Vertical distance method

Fig. 19 illustrates the vertical distance method for a nonlinear process. The points A, B, and C denote three experimental data pairs, and the corresponding points on the model curve shown by the dotted line are the model predictions vertically closest to A, B, and C

respectively.

2.3.2 MAXIMUM LIKELIHOOD METHOD

The maximum likelihood method maximizes the combined likelihood of the experimental data points.

Consider an experimental data pair. Due to uncertainty in the measurements, the value inputted to the process and the output recorded for different trials would be different. Depending upon the uncertainty associated, the experimental data pair may lie anywhere within a certain space as explained earlier (refer Fig.1). However if the measurements are repeated several times, the probability of the average being close to the true data pair increases, and the distribution of the input and output measurements, or the uncertainty associated with the measurements, follows a normal distribution, and as discussed earlier, a normal independent distribution where many, small, independent effects contribute to each observation, holds good in the case of uncertainties.

For an uncertainty of NID $(0, \sigma_x)$ in the input, and NID $(0, \sigma_y)$ in the output values, Fig. 20a depicts the possible Gaussian distribution of the input and output values.

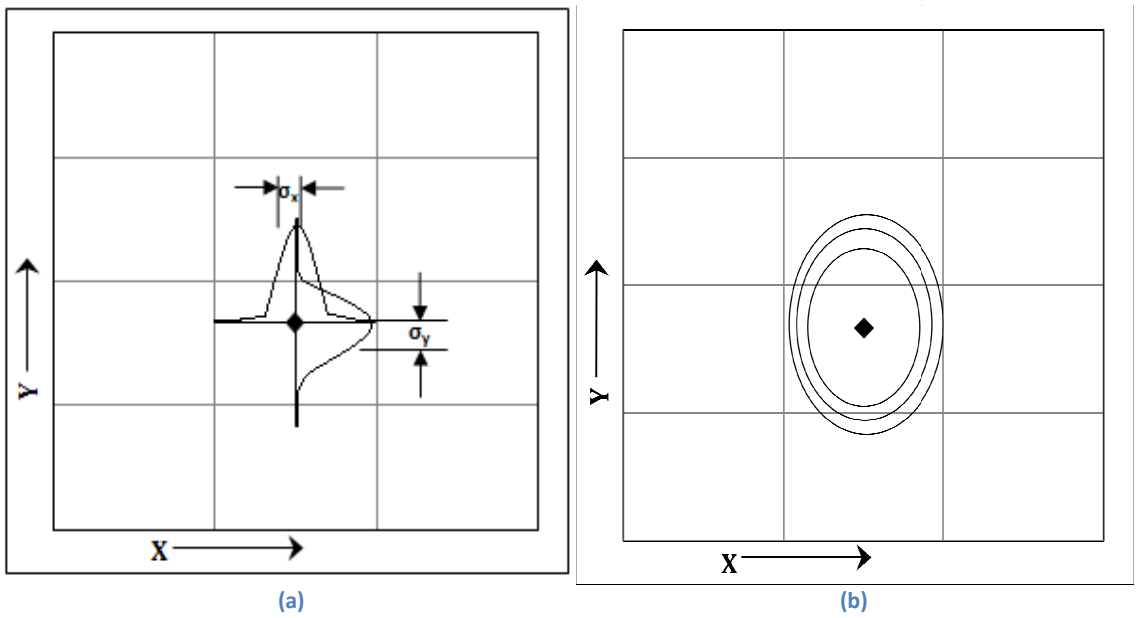


Figure 20: Likelihood contours - Maximum likelihood approach [3]

The central point in the figure is the combination of the most probable input and output values. The joint probability, likelihood contours corresponding to the input and output distribution are shown in Fig. 20b. Depending upon the standard deviation from the central point, the contours increase in size, the larger ones corresponding to higher multiples of σ_x and σ_y. They are shown vertically ellipsoidal indicating the uncertainty in the output values is greater than the uncertainty in the input values. If the uncertainties in the input and output values are the same, then the contours would be circular.

Each contour has a probability value associated to it. The closer the contour to the most probable point, the higher the probability. The most probable point however is unknowable but repeated measurements of the data pair ascertains the proximity to it. The maximum likelihood approach tries to fit the curve trying to maximize the combined probability of the all the data points as shown in Fig. 21.

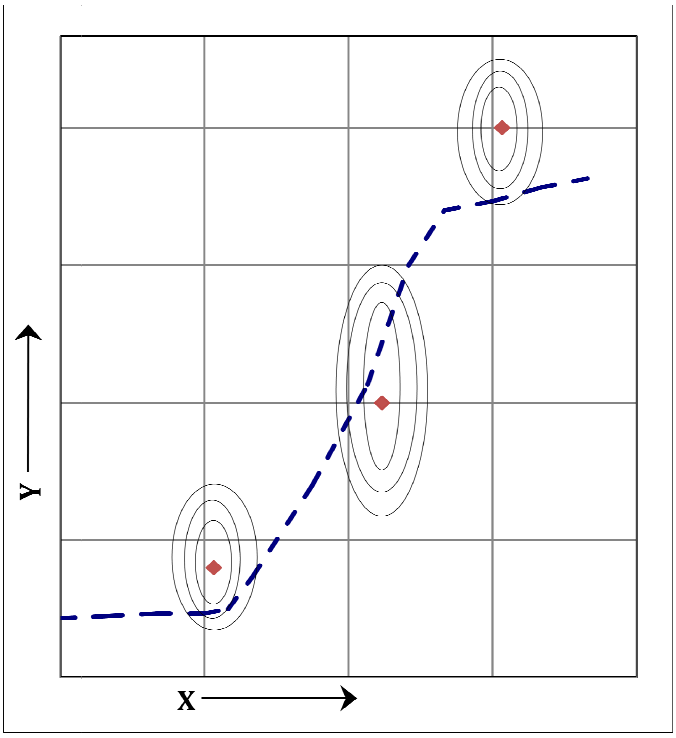


Figure 21: Maximum likelihood regression [3]

The point origin of each concentric set of likelihood contours shown in the figure represents the experimental data pair. That is, the experimental data pair is considered as the most probable point. This is not necessarily true, but due to its limited number of repeated measurements, this is a reasonable assumption. The maximum likelihood method adjusts the model parameter values such that a point on the model curve maximizes the likelihood of proximity to the experimental data set. The point on the model curve lies at the seat of the contour just touching the model curve shown by the dotted line, further elaborated in Fig. 22.

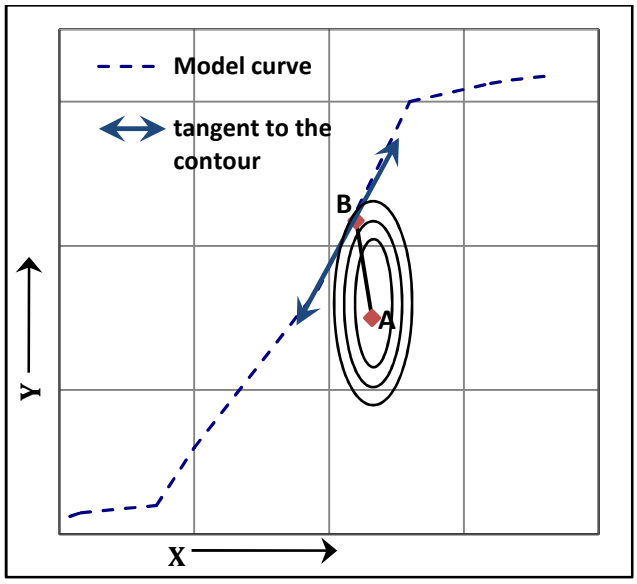


Figure 22: Maximum likelihood regression

Point ‘A’ in the above figure represents an experimental data pair and is the origin for the series of ellipsoidal contours emanating from it. Any point along a contour has the same likelihood probability value. The contours are continuous in space, in a sense that there exist several other contours with different probability values within the gap between two illustrated contours, and depending upon the parameter values, the model curve touches different contours with the model data pair, ‘B’ lying at the point of tangency to a contour.

The probability values for the likelihood contours are calculated as

$$P(X_i, Y_i) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left[\left(\frac{X_i-\tilde{X}_i}{\sigma_x}\right)^2+\left(\frac{Y_i-\tilde{Y}_i}{\sigma_y}\right)^2\right]} \quad (64)$$

where (X_i, Y_i) is the experimental data pair and $(\tilde{X}_i, \tilde{Y}_i)$ is the model predicted data pair. The maximum likelihood method tries to maximize the joint probability of all the experimental data points, i.e.,

$$\max_{\{X_{m,Ym}\}} J = \prod_{i=1}^{N} P(X_i, Y_i) = \prod_{i=1}^{N} \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left[\left(\frac{X_i-\tilde{X}_i}{\sigma_x}\right)^2+\left(\frac{Y_i-\tilde{Y}_i}{\sigma_y}\right)^2\right]} \tag{65}$$

This is the same as minimizing the negative of the index in the exponent term, i.e.,

$$\min_{\{X_{m,Ym}\}} J = \sum_{i=1}^{N} \left(\frac{X_i - \tilde{X}_i}{\sigma_x}\right)^2 + \left(\frac{Y_i - \tilde{Y}_i}{\sigma_y}\right)^2 \tag{66}$$

The term, $\left(\frac{X_i-\tilde{X}_i}{\sigma_x}\right)^2 + \left(\frac{Y_i-\tilde{Y}_i}{\sigma_y}\right)^2$ basically defines the shape of the contours. If $\sigma_x \neq \sigma_y$, it represents the equation of an ellipse, and hence the contours are ellipsoidal. If $\sigma_x = \sigma_y$, it represents circular contours, and the objective function reduces to

$$\min_{\{X_{m,Ym}\}} J = \sum_{i=1}^{N} \left(X_i - \tilde{X}_i\right)^2 + \left(Y_i - \tilde{Y}_i\right)^2 \tag{67}$$

which basically is minimizing the sum of ordinary distances.

2.3.2.1 Circular contours

When $\sigma_x = \sigma_y$, the contours are a set of concentric circles. Any model prediction would lie on a circle, distanced from the experimental pair by the radius.

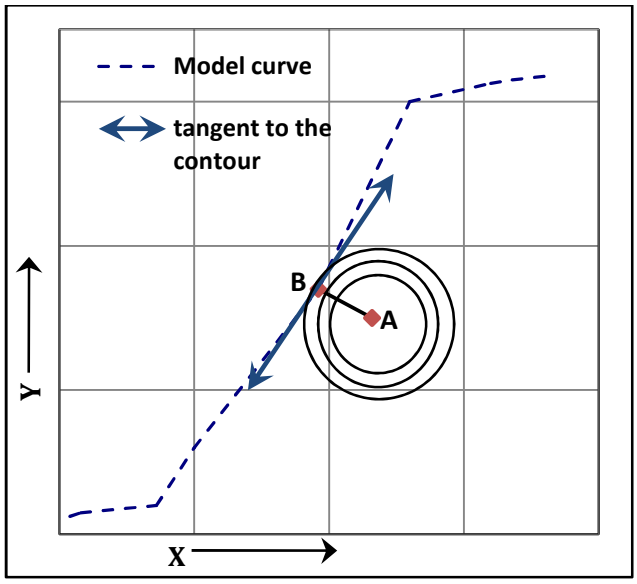


Figure 23: Circular contours – Maximum likelihood regression

For the contours shown in Fig. 23, for an experimental data pair, '*A*', the model curve locates a corresponding prediction '*B*', along the contour. Since concentric circles do not intersect (a fundamental property of circles), depending upon the probability values, the model pair would lie along different contours, with the radius of circle being the shortest distance. It is also known that the tangent to any circle is perpendicular to the radius. So a tangent at the model prediction is perpendicular to the distance between the experimental and model data pairs, or the distance between an experimental pair and the corresponding model prediction along the presumed model curve is the perpendicular distance, with the model prediction being the foot of the perpendicular to the experimental data pair.

Hence when $\sigma_x = \sigma_y$, maximizing the likelihood contour probability is the same as minimizing the perpendicular or the ordinary distances from the experimental points. The point obtained by minimizing ordinary distances is the same as that obtained by minimizing normal distances has also been cited in [4], but for linear regression. The same can also be extended for nonlinear models through the above discussion. This

however, is a logical argument and not a mathematical proof.

The maximum likelihood approach, however, requires information on the input and output variances, often not available. If an estimate of σ_x is available, σ_y can be evaluated through propagation of uncertainty principles that follows.

2.3.2.2 Propagation of uncertainty

If $Y = f(X)$ relates the input (X) and output (Y) of a particular process, any error in the input variable will propagate a corresponding error in the output variable depending upon the slope of the curve at that input value.

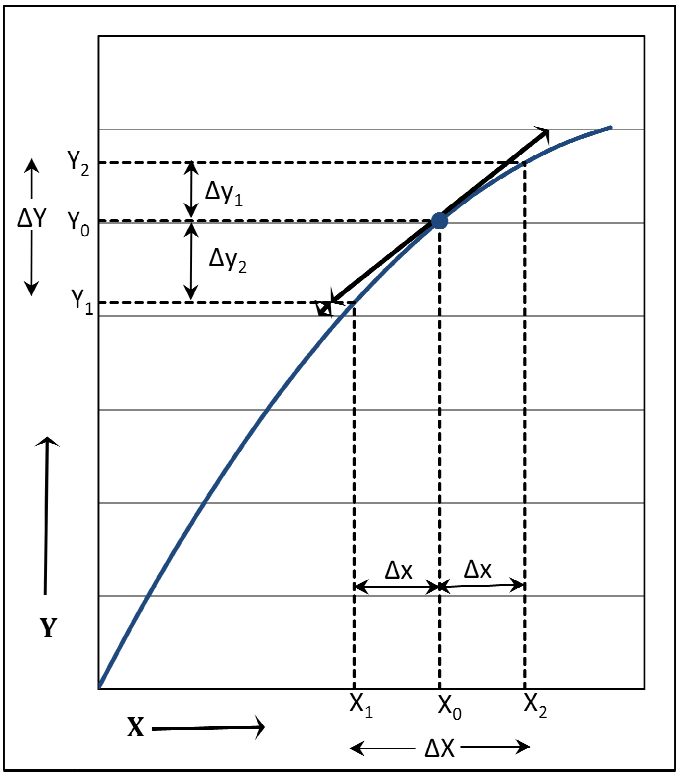


Figure 24: Error Propagation [3]

As shown in the curve above, if Δx is the maximum possible error in the input, the value may lie anywhere between X_1 and X_2, and correspondingly the output may lie between Y_1 and Y_2. The interval range between X_1 and X_2, Y_1 and Y_2 are denoted by ΔX and ΔY

respectively.

If ΔX is small, such that the curve may be linearly approximated within that region, then,

$$|\Delta Y| = \left|\left(\frac{\partial y}{\partial x}\Big|_{X_0}\right)\right| \cdot |\Delta X| \tag{68}$$

Eliminating the absolute values by squaring on both sides gives

$$\Delta Y^2 = \left(\frac{\partial y}{\partial x}\Big|_{X_0}\right)^2 \cdot \Delta X^2 \tag{69}$$

Although the rigorous derivation is complicated, the variances in the input and output value are related in a similar way as above. Hence for an uncertainty of σ_x in the input variable, the uncertainty propagated in the output variable σ_{yx} is given by

$$\sigma_{yx}{}^2 = \left(\frac{\partial y}{\partial x}\Big|_{X_0}\right)^2 \cdot \sigma_x{}^2 \tag{70}$$

The slope can be approximated through the central finite difference method as

$$\frac{\partial y}{\partial x}\Big|_{X_0} = \frac{f(X_0 + \delta x) - f(X_0 - \delta x)}{2(\delta x)} \tag{71}$$

where δx is a small change in the input variable.

Since there also exists inherent uncertainty in the measurement of the output variable, the total uncertainty in the output value (σ_{yt}) is the sum of the uncertainty propagated due to error in the input measurement (σ_x) and uncertainty in the output measurement (σ_{ym}), i.e.,

$$\sigma_{yt}{}^2 = \sigma_{ym}{}^2 + \left(\frac{\partial y}{\partial x}\Big|_{X_0}\right)^2 \cdot \sigma_x{}^2 \tag{72}$$

The total uncertainty in the output variable can be evaluated by calculating the standard deviation of repeated output measurements at constant input conditions. Hence the inherent uncertainty in output measurements can be estimated as

$$\sigma_{ym}{}^2 = \sigma_{yt}{}^2 - \left(\frac{\partial y}{\partial x}\Big|_{X_0}\right)^2 \cdot \sigma_x{}^2 \tag{73}$$

Note: σ_{ym} is the same as σ_y designated earlier.

For the titration experiment, σ_x can be estimated through the manufacturer's tolerances data, the maximum possible error through an equipment under normal conditions of operation, standard for the different types and sizes of volumetric equipments used.

The procedure for determining the inherent uncertainty in the output measurements for known uncertainty in the input measurements is outlined as follows

1. Conduct the experiment several times at constant input conditions (X_0), to obtain a set of output measurements. The standard deviation of the output measurements gives σ_{yt}.
2. Repeat the experiment at different input conditions and obtain the standard deviations for the each corresponding set of output measurements.
3. To evaluate the slope of the function at a given input condition, X_0, an estimate of the model parameter value is required to define the function over the entire range of input values. To do so, each set of output measurements is averaged and the model

parameter values are estimated for the input vs. the averaged output measurements through the vertical or the normal distance method. The slope can be then be evaluated using the central finite difference formula given by Eq. (71)

4. Once the slopes are known, the output uncertainty for each corresponding input measurement can be evaluated through Eq. (73)

The number of repeated output measurements required however cannot be determined. The greater the number of repetitions, the more accurate the estimates would be.

3. EXPERIMENTAL

3.1 TITRATION

3.1.1 DATA GENERATION

The experimental data were generated through the simulator which calculates the pH of a weak acid for a given volume of strong base added. The pH values were obtained through the interval halving method based on Eqs. (33) through (48).

3.1.2 REGRESSION

The parameters, A_0 and pK_a were estimated by the vertical distance method, the normal distance method and the maximum likelihood method. The estimation process involves optimizing the parameter values by minimizing the objective function. This is a two stage nested procedure. The parameters are optimized through a two-dimension search logic based on the objective function values and the ones that yield the least objective function value are the required estimates. The objective function for the vertical distance method is a straight forward evaluation of vertical distances based on the parameter values, but is a one-dimensional line search along the input variable for the normal distance and the maximum likelihood method for a given set of parameter values. The optimization statement may hence be written as

$$\min_{\{A_0, pK_a\}} J = \sum_{i=1}^{N} J_i \tag{74}$$

$$\min_{\{\tilde{V}_i\}} J_i = {d_i}^2 \tag{75}$$

$$S.T \quad \widetilde{pH}_i = f(\tilde{V}_i, A_0, pK_a) \tag{76}$$

Eq. (74) denotes the two-dimensional search process along A_0 and $pK_a$to find the least objective function value. The objective function, J is the sum of the least individual objective function values (J_i) for each experimental data pair. Based on the parameters A_0 and pK_a, a line search along V_i establishes the J_i values for the normal distance and the maximum likelihood methods (Eq. 75). For the vertical distance method, as discussed earlier, the objective function evaluation does not require a line search as the model and the experimental input volumes are the same ($V_i = \tilde{V}_i$). The pH values for each model data pair, ($\widetilde{pH}_i$), for all the methods, are evaluated based on the parameter values at a given model volume, $\tilde{V}_i$ (Eq. 76).

Various methods are available for optimizing the parameter values. Gradient based methods like the Cauchy's steepest descent, Newton-Raphson, Levenberg Marquardt, successive quadratic, etc. require the knowledge of function derivatives and fail at function discontinuities. Direct search methods like the R^3 cyclic method [3], Hooke-Jeeves, Nelder-Mead, etc., however, optimize based on function evaluations and are more robust. The R^3 cyclic method was chosen for its simplicity and versatility. It, in many cases, requires the least number of function evaluations among all the direct search methods which adds to its advantages.

3.1.2.1 R^3 Cyclic Direct Search Method

The direct search method is an alternating search along the decision variable with steps of varying magnitude directed towards the optimum. The steps sizes are increased by an expansion factor when the objective function moves towards the desired optimum, and decreased by a contraction factor when the objective function moves away from the optimum. The expansion and contraction factors are used for the purpose of speeding up the search process once the right direction is found.

Consider a one-dimension direct search to find the minimum of the objective shown in Fig. 25.

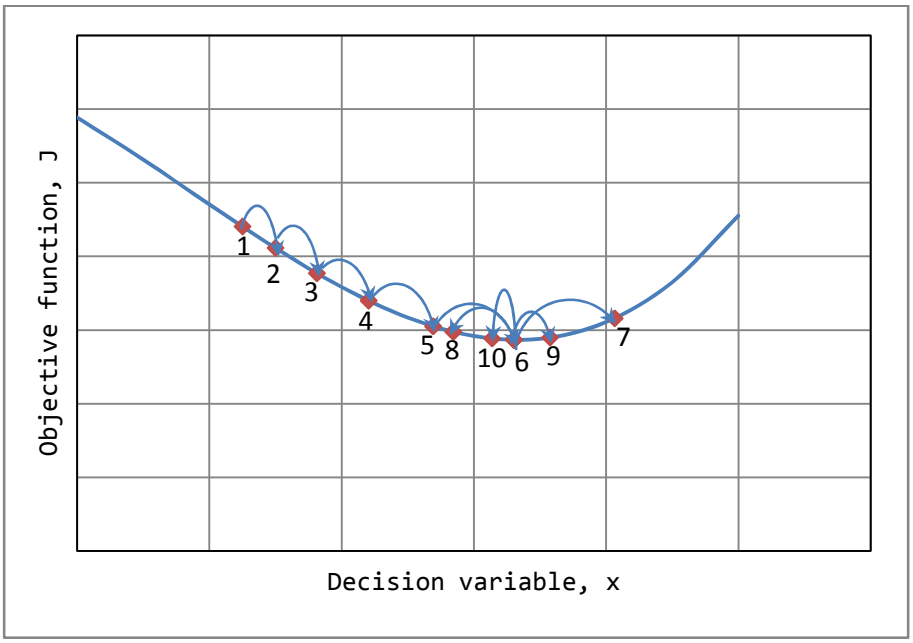


Figure 25: R^3 direct search – one dimensional [3]

For a random start at point 1, a step change of certain magnitude is made in the value of the decision variable, 'x'. Since the objective function value at point 2 is lower than the value at 1, the subsequent step size is increased by an expansion factor of 1.25. The process is continued until point 6 as the objective function value continues to decrease. A further increase in the step size leads to point 7 where the objective function value is

greater than the previous value. The trial solution is hence returned to point 6 and then proceeds in the opposite direction to point 8 with a decreased step size due to contraction factor. The objective function at 8 however, is still higher than the value at 6, and so the trial solution is again returned to the historical best value at 6 and the next step proceeds in the opposite direction with a further reduced step size to point 9. The step size continues to reduce upon each reversal of the search direction and finally reduces to an extent such that, its value, and the change in objective function due to step changes, become insignificant and the search process is stopped.

The cyclic direct search can also be used for functions with more than one decision variable. The direct search steps are cycled individually between the decision variables after a step change in each of the decision variables, and their respective moves for the next iteration is decided. The search logic for a two-dimensional optimization problem is illustrated in Fig. 26.

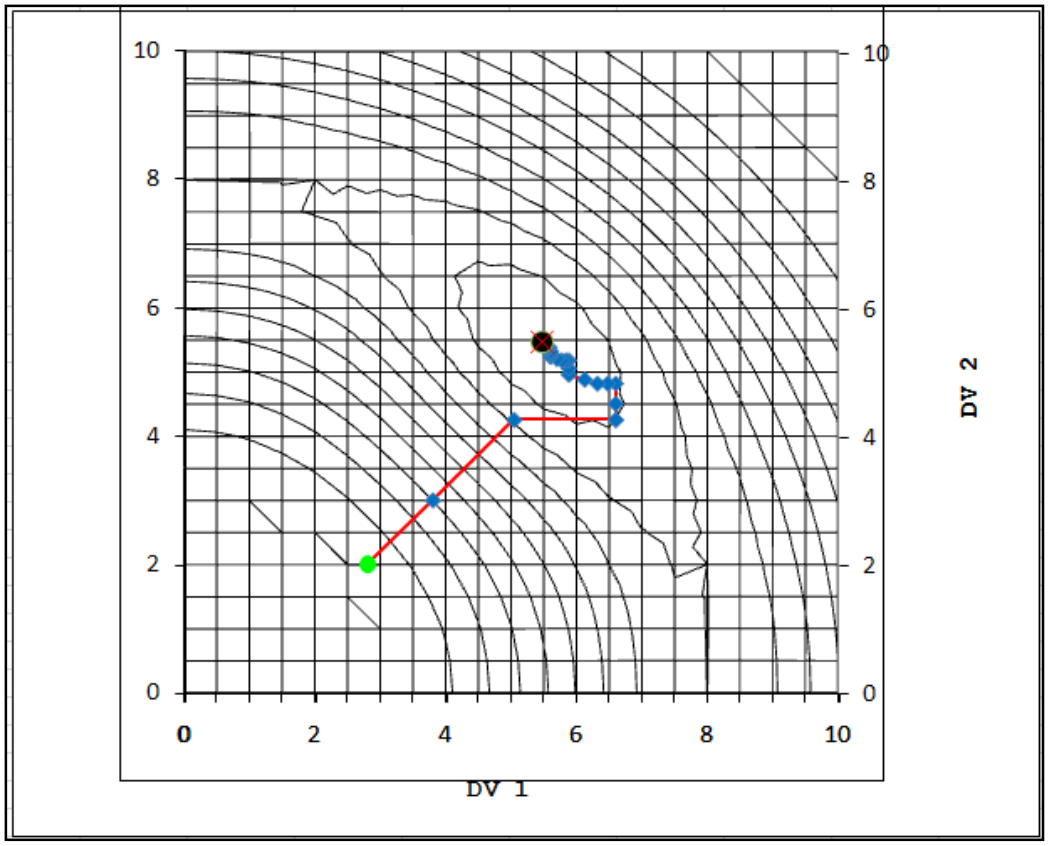


Figure 26: R^3 direct search – two dimensional [3]

The graph in Fig. 26 is a two-dimensional contour plot of a function vs. its decision variables. The objective is to find the minimum of the function which lies at the center trough. The search process is begun at a random point in the space, shown by a round marker here at around (2.75, 2). For a change in DV 1 by a magnitude of 0.5, the function value was found to be lower and hence the subsequent change in DV 1 would be multiplied by an expansion factor of magnitude 1.25, i.e., the subsequent change in DV 1 would be 0.5 * 1.25 = 0.625. Similarly DV 2 was incremented by a magnitude of 0.5 and the function value was found to be lower, the subsequent change would be 0.625. The initial magnitudes for the change in the DV's are usually a certain percent of their actual values. The initial changes in DV 1 and 2 complete the first iteration. The small diamond shaped markers shown along the DV path mark the end of each iteration. The changes in

DV 1 and 2 for the second iteration, as can be seen from the plot, are greater than the first iteration. For the third iteration, the increment in DV 2 caused the function to be higher than its previous value and hence the change was reversed and the subsequent change in DV 2 would be decremented by contraction factor of 0.75. The process of alternatively incrementing or decrementing the decision variables continues until the stopping criteria on the total number of iterations is affected. Other stopping criteria like the threshold limit on the absolute or relative change in the objective function value or the decision variables, the Rhinehart steady state stopping criteria [15, 16] etc. can also be implemented.

The search logic for the titration process involves optimization of the parameter values (A_0, pK_a) by minimizing the objective function corresponding to the regression technique chosen. In order to ensure that the search logic is robust, the parameters were initialized with random guesses deviating around 100% from the true values. The step increments for the parameters were begun with 10% of their initial value, and an expansion and a contraction factor of 1.25 and 0.75 respectively were chosen. The stopping criteria on the change in objective function and the step increments in the decision variables to the order of 10^{-10} were chosen such that it does not affect the search process. The veracity of the algorithm was verified by running the simulator to generate data with no uncertainty to it, and the parameter estimates by either of the methods should yield the actual values.

For each realization (a set of experimental data) the parameter values were estimated by all the three regression methods. In order to ensure the search logic does not create a bias for any of the regression techniques, the optimization algorithm was maintained

exactly the same for all the three regression techniques including the initialization and the stopping criteria. The objective function evaluation (described in the next section), however, differs for each of the methods. The algorithm and the flowchart of the R^3 cyclic direct search method for the titration process are explained as follows.

Algorithm:

Initialization:

1. Guess $X\begin{pmatrix}1\\2\end{pmatrix} = \begin{pmatrix}A_0\\pK_a\end{pmatrix}$
2. Calculate the objective function, J_0. $J_2 = J_0$
3. Initialize ΔX, expansion and contraction factor, $i = 1.\ i \in Z,\ \forall Z = [1, 2]$

Search Logic:

4. $X(i) = X(i) + \Delta X(i)$
5. Calculate $J_1 = f(X_1, X_2)$
6. If $J_1 < J_0$, then $J_0 = J_1$

   $\Delta X(i) = \Delta X(i) * Exp.factor$

   Else,

   $X(i) = X(i) - \Delta X(i)$

   $\Delta X(i) = -\Delta X(i) * \frac{Cont.\ factor}{Exp.\ factor}$

   End if

$Next\ i$

Stopping Criteria:

7. If $(|(J_0 - J_2)| < 10^{-10})$ & $(|(\Delta X)| < 10^{-10})$ $then$,

   Exit Loop

   Else

   $J_2 = J_0$

   End if

Note:

J_0 = the least objective function value at the time of reading

J_1= the current objective function value at the time of reading

J_2 = the initial objective function value before starting a new iteration

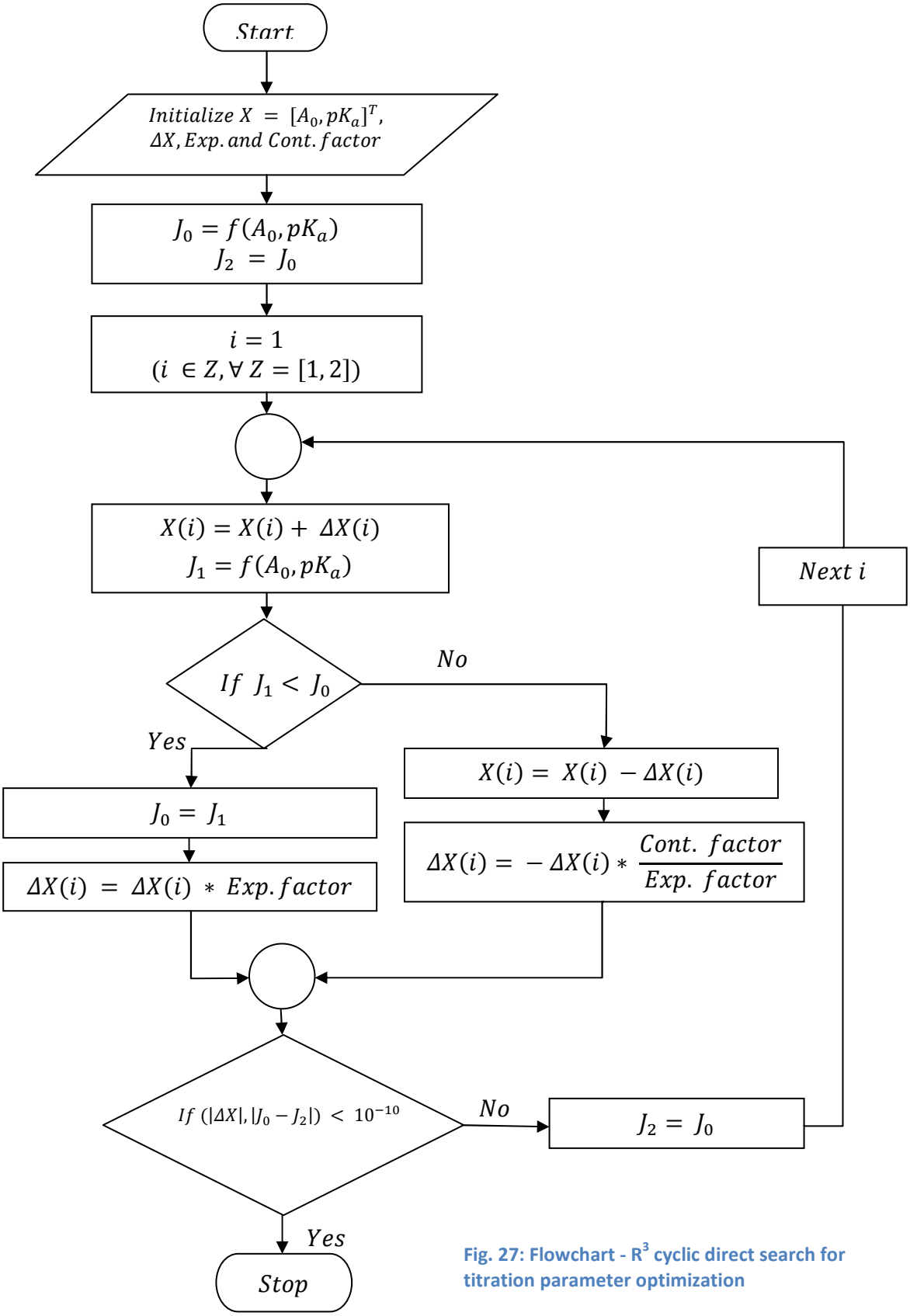


Fig. 27: Flowchart - R^3 cyclic direct search for titration parameter optimization

3.1.2.2 Objective function evaluation

This section explains the objective function evaluation (Eq. 75 & 76) for the three regression methods.

3.1.2.2.1 Least squares regression – Vertical distances

The objective function for the least squares regression is evaluated by calculating the output variable for a given set of parameter values, with the input variable as the same as that of the experimental data, and hence calculating the vertical distances. The sum of all the vertical distances gives the required objective function. The algorithm for the titration process can be explained through a flowchart as shown below.

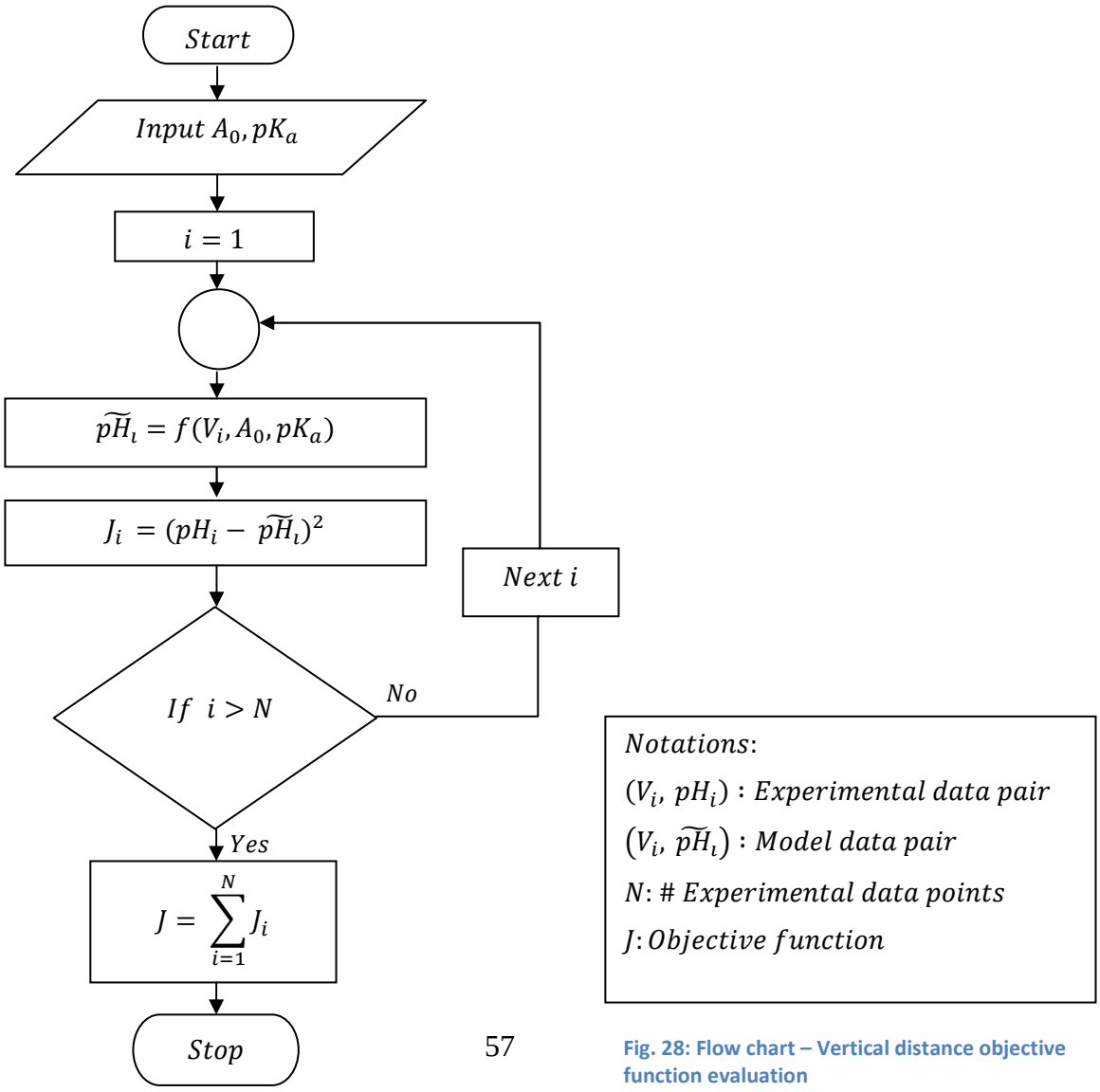


Fig. 28: Flow chart – Vertical distance objective function evaluation

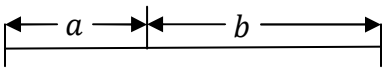
3.1.2.2.2 Maximum likelihood and Normal distances

The objective function evaluation for the normal distance and the maximum likelihood method is a one-dimensional optimization problem. For a given experimental data pair, a line search is required along the volume to find the corresponding model data pair that best minimizes the objective function. Several methods like the successive quadratic, golden section, Newton-Raphson, and the marching method can be implemented for this purpose.

For the golden section method, based on the experimental data, the minimum and maximum bounds on the volume of base added for each model data pair can be established as a certain percentage of the least and the highest experimental volume, say 90 and 110% respectively, making its applicability possible. The Newton-Raphson method and the marching method require just one initial guess to start the search process, but the Newton-Raphson method involves evaluation of derivatives which is not always convenient (The marching method is an exhaustive search for the optimum through small step increments along the decision variable).The successive quadratic method can also be implemented, but it requires three initial guesses. Hence, the marching method is the easiest to implement. However, while conceptually simple, the marching method, due to large number of function evaluations, slows down the search process considerably. The golden section search, alternatively, with its ability to discard about 38% of the range per iteration, can be used for faster convergence to the desired optimum. This work uses the golden section search.

3.1.2.2.2.1 Golden Section Search

If a line segment is split in a way such that the ratio of the smaller segment to the larger segment is equal to the ratio of the larger segment to the length of the line, then the ratio is called the golden ratio and has a numerical value of 0.61803398...



$$\frac{a}{b} = \frac{b}{a+b} = \gamma$$

$$then\ \gamma = 0.61803398\ \ldots \tag{77}$$

The numerical value for γ can be obtained as follows:

Modifying Eq. (77) gives

$$\left(\frac{a}{b}\right)^2 + \frac{a}{b} - 1 = 0 \tag{78}$$

Eq. (78) is a quadratic in a/b and can be solved for the roots as

$$\frac{a}{b} = \frac{-1 \pm \sqrt{5}}{2} \tag{79}$$

a/b represents a ratio, and can take only positive values, which gives

$$\frac{a}{b} = \gamma = 0.61803398\ \ldots \tag{80}$$

The golden section optimizes the objective function by successively narrowing down the line search range on the decision variable based on the golden ratio.

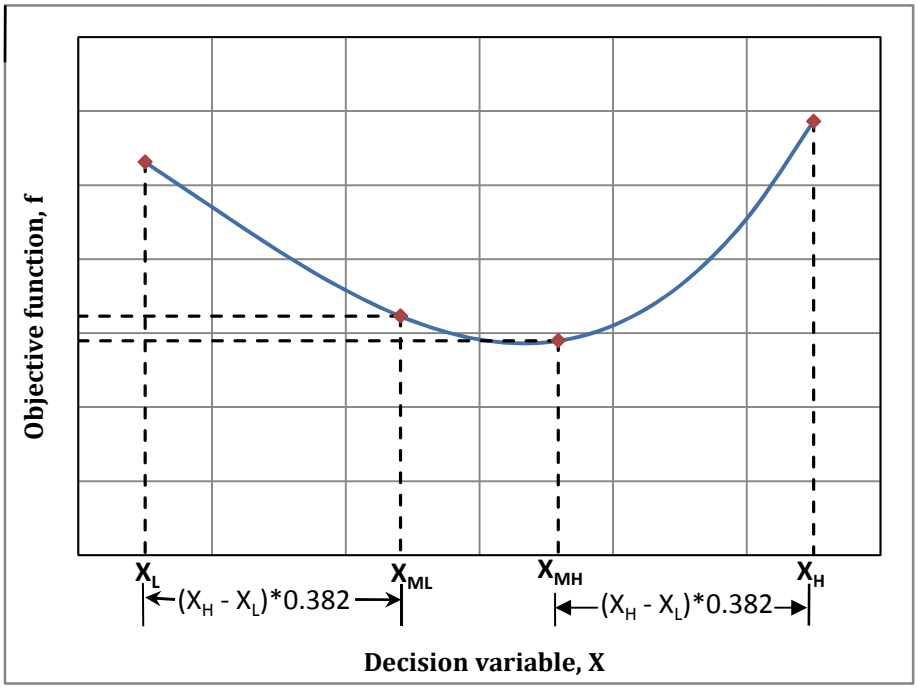


Figure 29: Golden section search

For the objective function shown above, the optimum is bound the two limits X_L and X_H. The points X_{ML} and X_{MH} are determined using the golden ratio as shown in Eqs. (81) and (82), and since the value of the objective function at X_{ML} is lower than the value at X_{MH}, the minimum is now bound between X_L and X_{MH}.

$$X_{ML} = X_L + (X_H - X_L) * (1- \gamma) \tag{81}$$

$$X_{MH} = X_H - (X_H - X_L) * (1- \gamma) \tag{82}$$

The search range is now reduced by about 38.2%, and X_L and X_{MH}, are denoted as the lower and upper limits, X_L and X_H respectively, and the search process is again repeated. The process goes on until the range or the interval size reduces to the desired limit.

It is important to note that the input and output variables need to be scaled while evaluating the normal distances or the maximum likelihood objective function. This ensures that the objective function is not biased towards any variable due to its magnitude which is simply a result of the choice of the units, and equal weightage is given to all the variables. For the titration experiment the variables were scaled between 0 and 1 through Eqs. (83) and (84)

$$V' = \left(\frac{V - V_{min}}{V_{max} - V_{min}}\right) \tag{83}$$

$$pH' = \left(\frac{pH - pH_{min}}{pH_{max} - pH_{min}}\right) \tag{84}$$

where $V_{min}, V_{max}, pH_{min}, pH_{max}$ are the minimum and maximum experimental volumes and pHs.

The scaled data was used for both the normal and the maximum likelihood methods. Accordingly, to proportionate the variances in the input and output variables, these were scaled as follows.

$$\sigma_x{}' = \left(\frac{\sigma_x}{V_{max} - V_{min}}\right) \tag{85}$$

$$\sigma_y{}' = \left(\frac{\sigma_y}{pH_{max} - pH_{min}}\right) \tag{86}$$

The flowchart for the golden section search process for the maximum likelihood method is shown in Fig. 30. If $\sigma_x{}' = \sigma_y{}'$ then it represents the normal distance method.

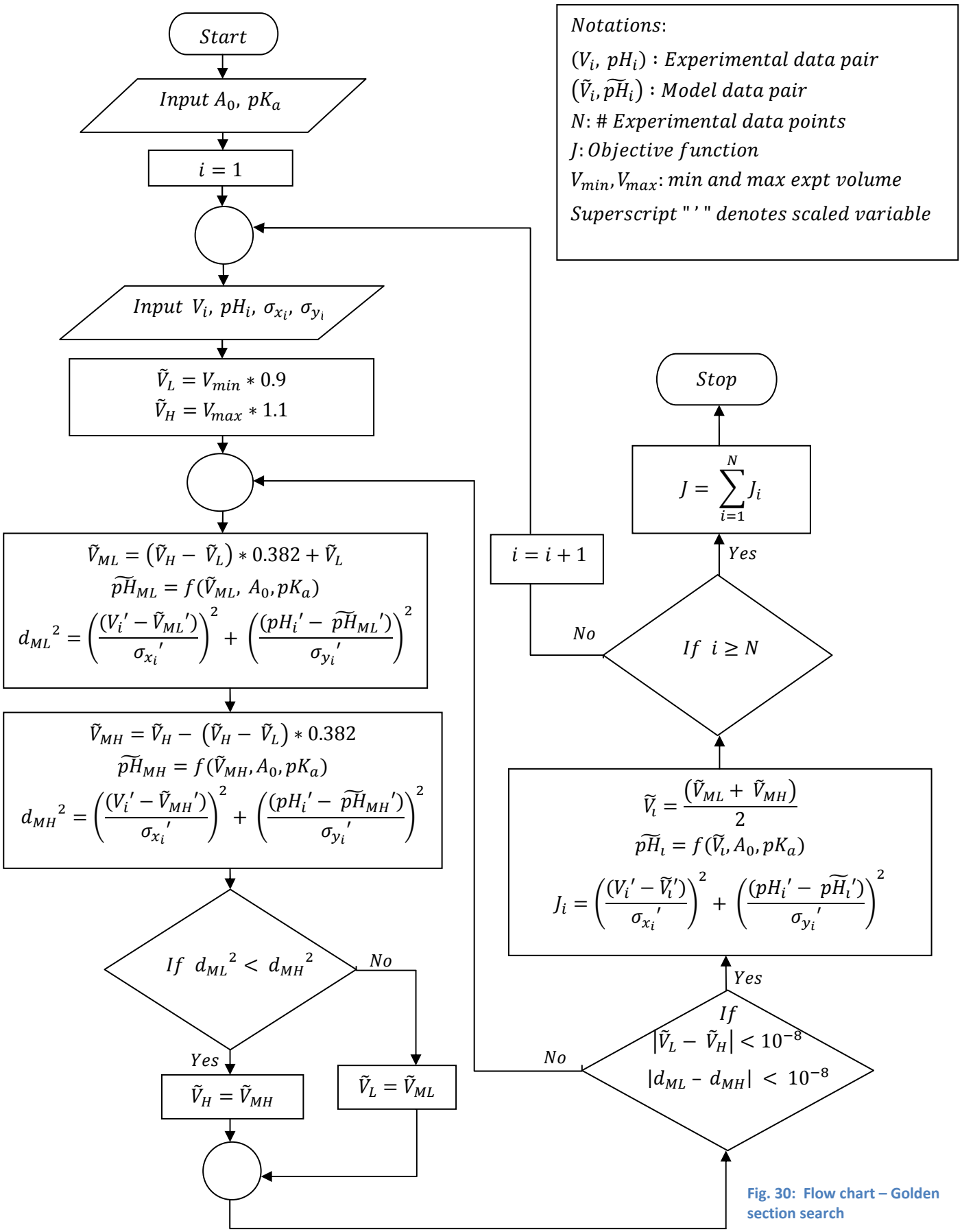


Fig. 30: Flow chart – Golden section search

3.2 PACKED BED REACTOR

3.2.1 DATA GENERATION

The data generation process is as described in Section 2.2.2.2. For a given set of input and the parameter values, the output values were obtained through the Newton Raphson root finding technique.

3.2.2 REGRESSION

Like the titration experiment, the regression parameter values for the packed bed model (K_0, K_1, and E) were estimated by the vertical distance, the perpendicular distance and the maximum likelihood method. The optimization process is a two stage nested procedure involving the optimization of the parameter values through minimizing the objective function. The optimization statement can be written as

$$\min_{\{K_0,K_1,E\}} J = \sum_{i=1}^{N} J_i \tag{87}$$

$$\min_{\{\widetilde{C_{in_i}},\tilde{v}_i,\tilde{T}_i\}} J_i = {d_i}^2 \tag{88}$$

$$S.T \qquad \widetilde{C_i} = f(\widetilde{C_{in_i}}, \tilde{v}_i, \tilde{T}_i, A_0\,, pKa) \tag{89}$$

3.2.2.1 Parameter Optimization

Eq. (87) denotes the three-dimensional search along the parameters, K_0, K_1, and E. The R^3 cyclic direct search was used for this purpose. The parameters were initialized by random guesses deviating around 100% from the true values. The initial step increments were 10% of their starting values, and an expansion factor of 1.25 and a contraction

factor of 0.75 were chosen for the optimization algorithm. The parameter values were constrained between their individual nominal limits to avoid impractical values if the search direction followed the wrong path. This was done through a 'soft constraint' by adding a penalty to the objective function once the parameter values go beyond the nominal limits, thereby reversing the search direction. The penalty chosen was the square of the magnitude of deviation from the nominal limits.

$$If\ X > X_n, penalty = \ (X - X_n)^2$$

$$If\ X < X_n,\ penalty = 0 \tag{90}$$

The objective function hence would be: $J + penalty$ (91)

If more than one parameter deviated from the constrained limits, the greater of their deviations was added as the penalty. The penalty could be added in different ways depending upon the choice of the programmer.

The stopping criteria for the optimization logic was to restrict the change in the objective function and the decision variables beyond a magnitude of 10^{-10}. The optimization logic remained the same for all the three regression methods to avoid bias.

Fig. 31 illustrates the flowchart for the optimization routine. The penalty is designated by 'P' in the flowchart.

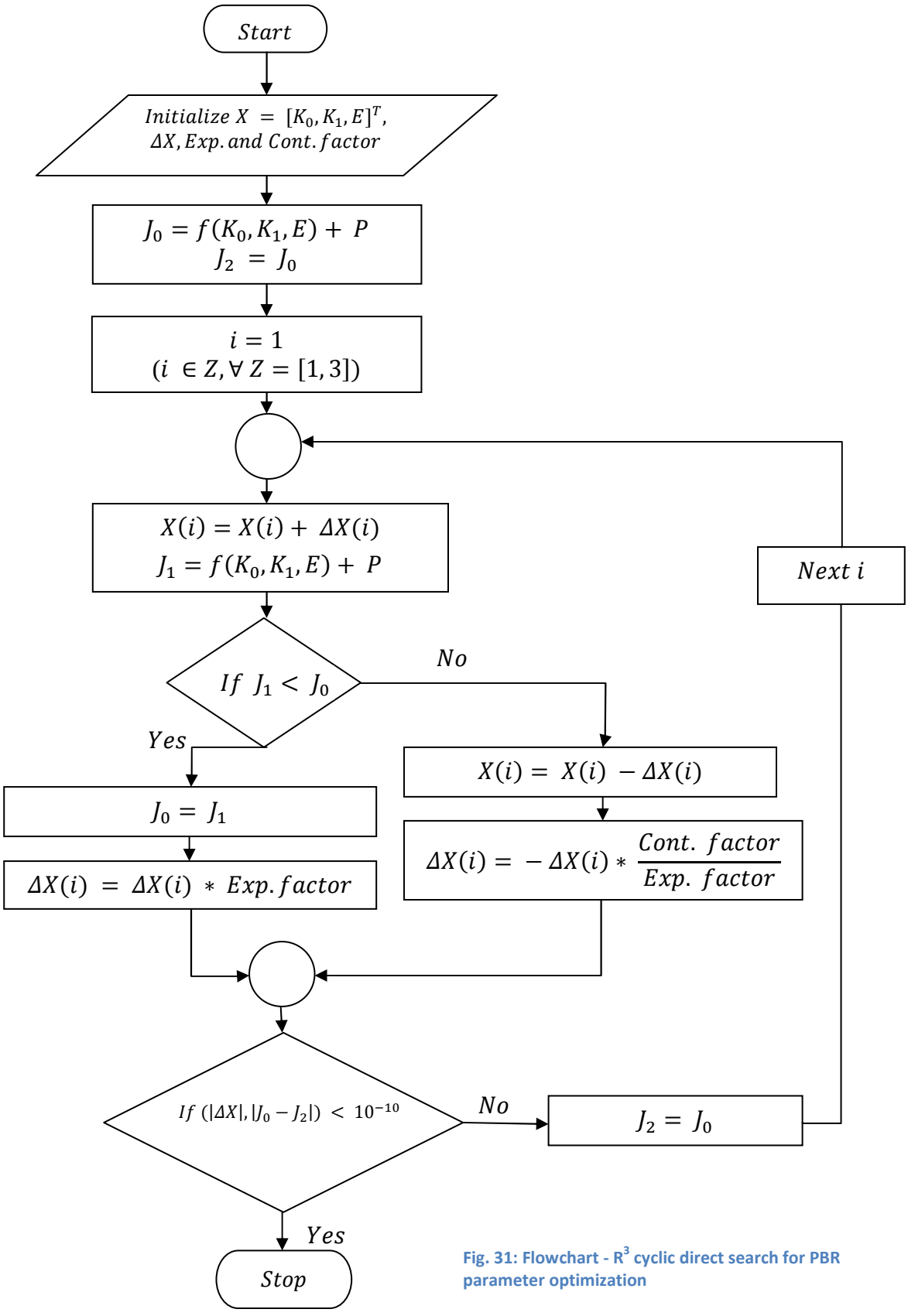


Fig. 31: Flowchart - R^3 cyclic direct search for PBR parameter optimization

3.2.2.2 Objective Function Evaluation

Eq. (88) represents the three-dimensional search along the input variables, $\widetilde{C_{in_i}}$, $\tilde{v_i}$, and $\tilde{T_i}$, to evaluate the objective function for the normal distance and the maximum likelihood methods. Unlike the titration experiment, the golden section search, which is a one-dimensional search, cannot be used in this case. The R^3 cyclic direct search, as discussed earlier, is well suited for multivariable searches, and was implemented again.

The objective function evaluation flowchart for the above said methods is illustrated in Fig. 32. The input and output variables were scaled between 0 and 1. The variances were also proportioned by scaling them.

The objective function was evaluated as

$$J_k = \left(\sum_{P=1}^{3}\left(\frac{(X_k{}'(P) - \tilde{X}_k{}'(P)}{\sigma_k'(P)}\right)^2\right) + \left(\frac{(C_0{}'(k) - \widetilde{C_0}'(k)}{\sigma_{C_0}'(k)}\right)^2 \tag{92}$$

$$J = \sum_{k=1}^{N} J_k \tag{93}$$

where $X_k{}'$ represents the scaled input variables for the k^{th} measurement of the experimental data and $\tilde{X}_k{}'$ represents the corresponding scaled input variables for the model.

$$X_k{}' = [C_i{}'(k), v'(k), T'(k)]^T \tag{94}$$

$$X_k{}'(1) = C_i{}'(k);\ \ X_k{}'(2) = v'(k);\ \ X_k{}'(3) = T'(k) \tag{95}$$

When $\sigma_k' = \ \sigma_{C_0}'$, the algorithm represents the normal distance method.

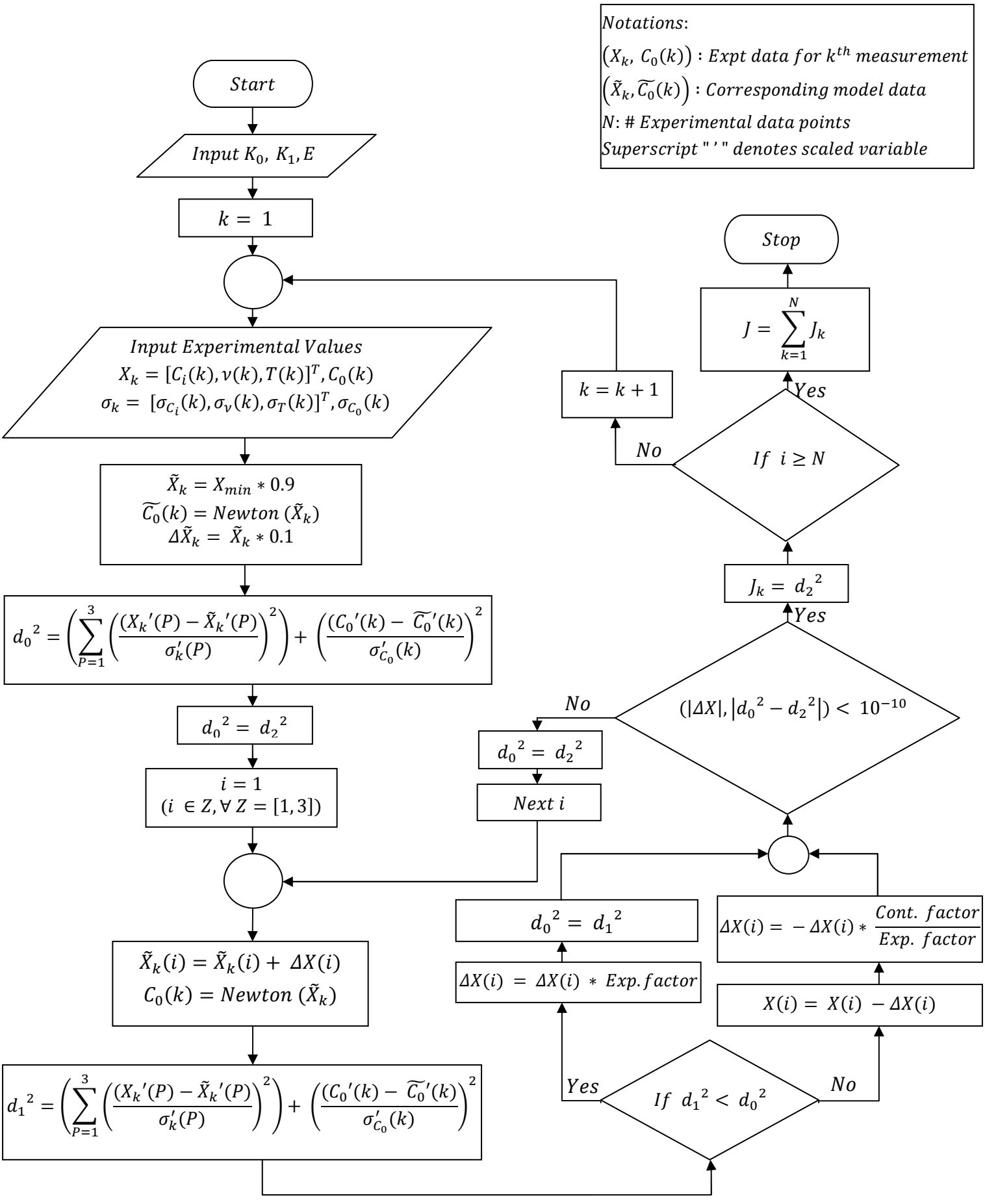


Fig. 32: Flowchart - R^3 cyclic direct search for PBR objective function optimization (Normal distance and Maximum likelihood method)

The objective function evaluation for the vertical distance method does not require the nested optimization search as the model input variables are the same as the experimental data, i.e., $\widetilde{C_{in_i}} = C_{in_i}, \tilde{v}_i = v_i, \widetilde{T_i} = T_i$.

3.3 ASSUMPTIONS

Before proceeding further it is important to understand the assumptions that were involved in the above methods and their possible implications.

1. Gaussian uncertainty

It was assumed that the experimental measurements, or more specifically the uncertainties associated with it, could be approximated by Gaussian distribution. This would imply that the variable would have to be measured several times for it to represent a Gaussian distribution. However, the actual requirement as pointed out in [13], is the assumption that the distribution of variable is symmetric. A Gaussian approximation may be used for this purpose but is not a necessity.

2. No systematic uncertainty

Systematic errors are a characteristic of an experiment or the system involved in a process and cannot be generalized. The only possible way to account for systematic errors is to introduce additional terms in the model function that negates its deviation. Since systematic errors are specific to the process, any model modification for the simulation would be unrealistic and should be dealt when dealing with the actual process.

Systematic uncertainties require a non-Gaussian distribution with a non-zero mean [2]. Hence to avoid systematic uncertainties, a Gaussian distribution with a mean of zero was implemented.

3. Independent Measurements

The experimental uncertainties were assumed to be independent, and hence the Gaussian or the normal independent distribution. This is a standard assumption for statistical analysis [2].

4. The model approximation

The simulator that generates the experimental data uses the same model and procedure to calculate output values for the regression. Although it is unlikely that engineering models ever exactly express the natural phenomena of a process, this situation of functional identity between the simulator and model permits the evaluation of accuracy in regressed parameter values.

3.4 RESULTS ANALYSIS TECHNIQUES

The regression methods were tested for different conditions of input and output uncertainties (σ_x and σ_y), each for a thousand realizations, and the distribution of the parameter values including the mean and the standard deviation were evaluated. A large number of realizations ensured that the results reflected the average values and were unique to the regression method.

Bias (or mean) and standard deviation:

The relative deviation of the mean from the true value of the parameter, also called the bias, should be small for a regression method to be "good". For a parameter with true value β, and $\hat{\beta}$, its estimate from a particular method, [9]

$$bias = \left| \frac{\bar{\hat{\beta}} - \beta}{\beta} \right| \tag{96}$$

$\bar{\hat{\beta}}$ in the above equation denotes the mean of parameter estimates. The standard deviation of the distribution for the "good" method should also be lower.

Frequency Distribution:

Apart from the mean and standard deviation, the distributions of the parameters were analyzed by creating histograms. Typical distributions of the parameters for the titration experiment for the vertical and maximum likelihood methods with $\sigma_x = 0.25$, $\sigma_y = 0.1$ are shown in Fig. 33 (a and b).

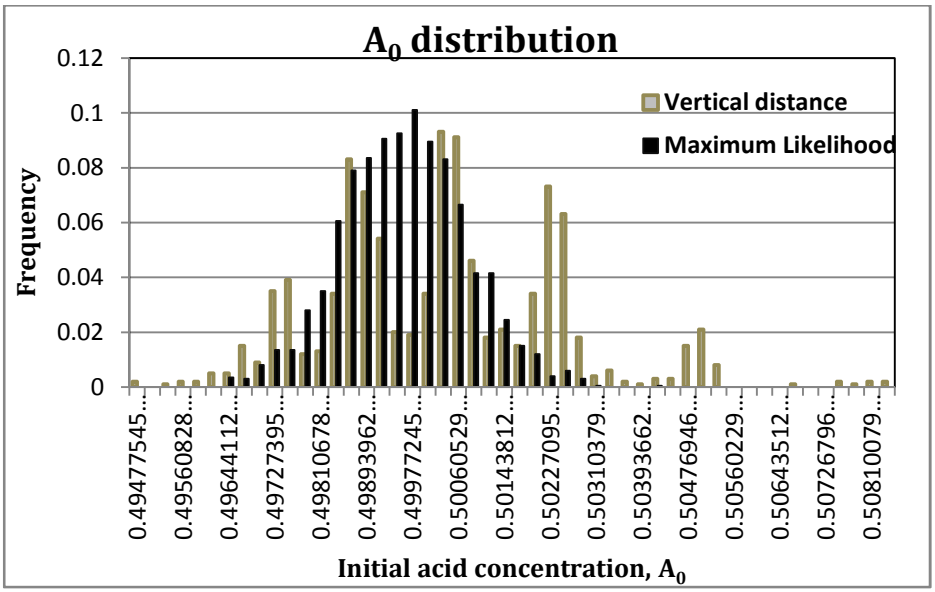


Figure 33a: Typical A_0 distribution for vertical and maximum likelihood method

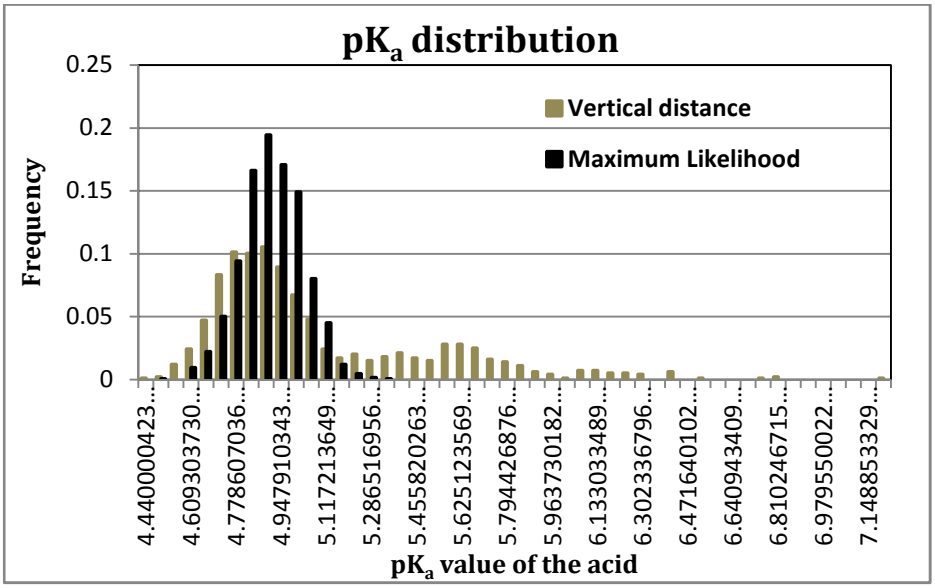


Figure 33b: Typical pK_a distribution for vertical and maximum likelihood method

As can be seen from the figures, the maximum likelihood method yields a symmetric distribution, more like the normal distribution, with a lower variance compared to the vertical distance method. Additionally, the distributions of the parameter values for the vertical distance method do not specifically follow any conventional

pattern. Lower variance indicates better consistency of a method in predicting the parameter values.

(Note: A comparison of all the three methods together in Fig. 33 obscures the distinction of the histograms corresponding to each of the method, hence only two were plotted).

Probability of worse deviation:

A third test, the probabilities for obtaining parameter values beyond a certain deviation from the true value were evaluated for each of the methods by counting the number of times the parameter values exceed the required deviation limit. The lower the probability, the better the method. Typical probability plots for the titration experiment with $\sigma_x = 0.25,\ \sigma_y = 0.1$ are shown in Fig. 34.

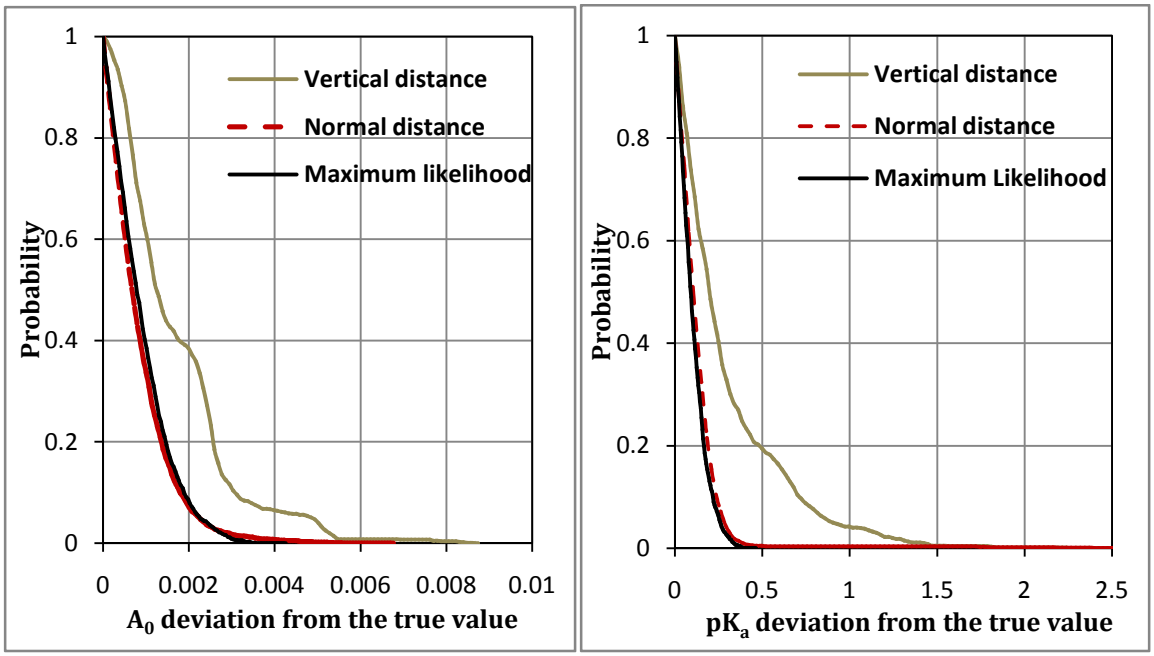


Figure 34: Typical 'probability of deviation from true value' plots for A_0 and pK_a

Fig. 34 depicts the probability of finding a parameter value at a certain deviation from the true value is always higher for the vertical distance method as compared to the other two methods, indicating its predictions are far worse than the other two methods.

4. RESULTS & DISCUSSION

The vertical distance, the normal distance, and the maximum likelihood methods were tested for varying magnitudes of uncertainty in the input and output variables. The magnitudes were basically decided through intuitive guesses on the possible errors associated with the variables, but were also considered beyond the typical limits to test the robustness of the method.

4.1 TITRATION

For the titration simulation, the apparent data range for the input variable, the volume of the base added, varied from 40.5ml to 42.5ml. Assuming a maximum error in the volume measurement to be 0.75ml (i.e., 37.5% of the range), with a probability of 99% would imply $3\sigma_x \cong 0.75ml$ (Refer Fig. 18), or $\sigma_x \cong 0.75/3 \cong 0.25ml$. Extending the range beyond nominal limits, σ_x values were varied from 0.1ml to 0.7ml.

Similarly, the usual pH range for the apparent data varied from 6.5 to 11.5$units$. Assuming a maximum error of 1unit would imply $3\sigma_y \cong 1.0$ or $\sigma_y \cong 1.0/3 \cong 0.33units$. Hence it was reasonable to vary σ_y value along the same range as σ_x, that is from 0.1$units$ to 0.7$units$.

The methods were tested basically for three cases

i. $\sigma_x < \sigma_y$ ii. $\sigma_x = \sigma_y$ iii. $\sigma_x > \sigma_y$

The true values for A_0 and pK_a in all the cases were chosen as 0.5 and 5 respectively.

4.1.1 $\sigma_x < \sigma_y$

Theoretically, when $\sigma_x < \sigma_y$, and σ_y is significantly higher than σ_x, the vertical distance method which is based on the premise that all the uncertainty rests in the output measurements and $\sigma_x = 0$, should predict parameters reasonably close to the true values. However, the results tabulated in Table 4.1.1 suggest that this does not hold true always, particularly for high uncertainty values.

		Vertical Distance		Normal Distance		Maximum Likelihood	
		Bias	Standard Deviation	Bias	Standard Deviation	Bias	Standard Deviation
σ_x = 0.1	A_0	0.000132	0.000985	0.000046	0.000806	0.000083	0.000874
σ_y = 0.25	pK_a	0.002532	0.234873	0.008217	0.219850	0.009365	0.240614
σ_x = 0.1	A_0	0.000148	0.001170	0.000163	0.001301	0.000064	0.000999
σ_y = 0.5	pK_a	0.000282	0.356470	0.020523	0.392295	0.014865	0.346681
σ_x = 0.25	A_0	0.000754	0.002270	0.000222	0.001674	0.000236	0.001710
σ_y = 0.5	pK_a	0.016562	0.505455	0.031143	0.417657	0.029934	0.377001

Table 4.1.1 Comparison of the regression methods for $\sigma_x < \sigma_y$

When $\sigma_x = 0.1$, $\sigma_y = 0.25$, the bias for the parameter A_0 is the least for the normal distance method, while for pK_a, it is the least for the vertical distance method. pK_a is the more dominating parameter of the two, significantly affecting the model curve, and as can be seen from the results, there is a considerable difference in the magnitudes of biases and standard deviations of A_0 and pK_a, for all the cases, for all the methods. Hence comparatively it is more important to get better results for pK_a than A_0. Hence in terms of biases, the vertical distance is the preferred method. However, the standard deviations

for A_0 and pK_a estimates are the least for the normal distance method. The difference of the standard deviations among the methods though, is not substantial. The standard deviation actually is a more reliable test than the bias, as while evaluating the bias, the positive deviations of the parameter values annul the negative deviations, thereby despoiling the very essence of the test.

Hence for the present sub-case, all the methods could be considered at par. However, for the simplicity of the logic, and the ease of computational burden, the vertical distance method could be chosen over the others.

For σ_x = 0.1, σ_y = 0.5, due to the increase in the uncertainty in the pH measurements, the predictions of each of the methods deteriorate compared to the previous case. And, while bias for pK_a is the least for the vertical distance method, there is not much difference in the standard deviations for the vertical and the maximum likelihood methods. Due to the decrease in the ratio of σ_x to σ_y, the variation in the results for the vertical and the normal distance methods become obvious. The vertical distance method, followed by the maximum likelihood and the normal distance methods, could be the favored order of preference.

When σ_x = 0.25, σ_y = 0.5, due to the rise in the uncertainty in the input measurements the predictions further deteriorate compared to the previous case. However the normal distance had better predictions than the vertical distance method, but the maximum likelihood method could be ascertained the best among the three.

Typical distributions for the parameters and their probability plots for σ_x = 0.25, σ_y = 0.5 are shown in the Figs. 35 and 36 respectively.

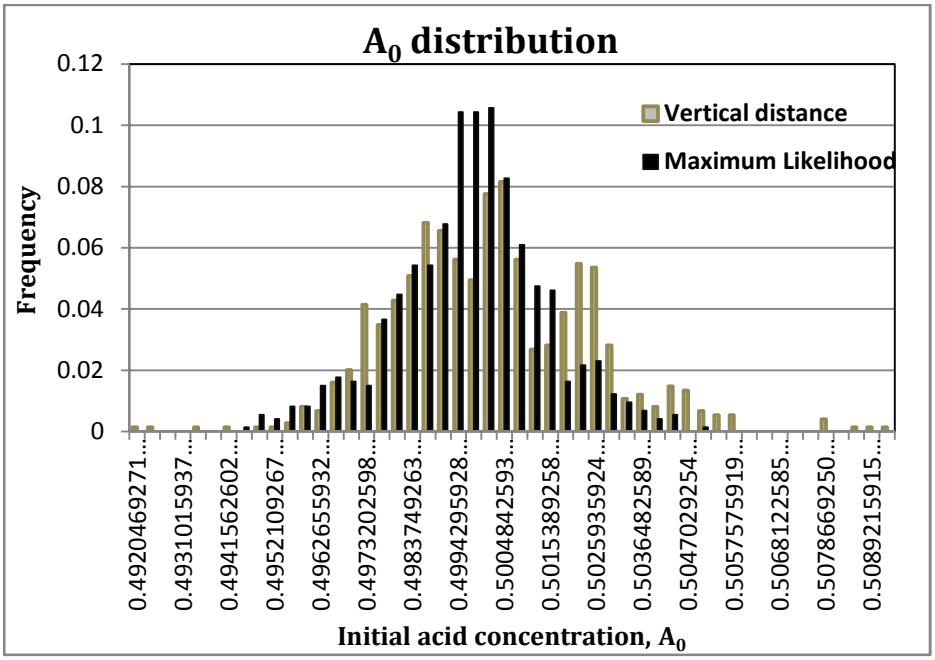


Figure 35a: A_0 distribution for vertical and maximum likelihood method for σ_x = 0.25, $\sigma_y = 0.5$

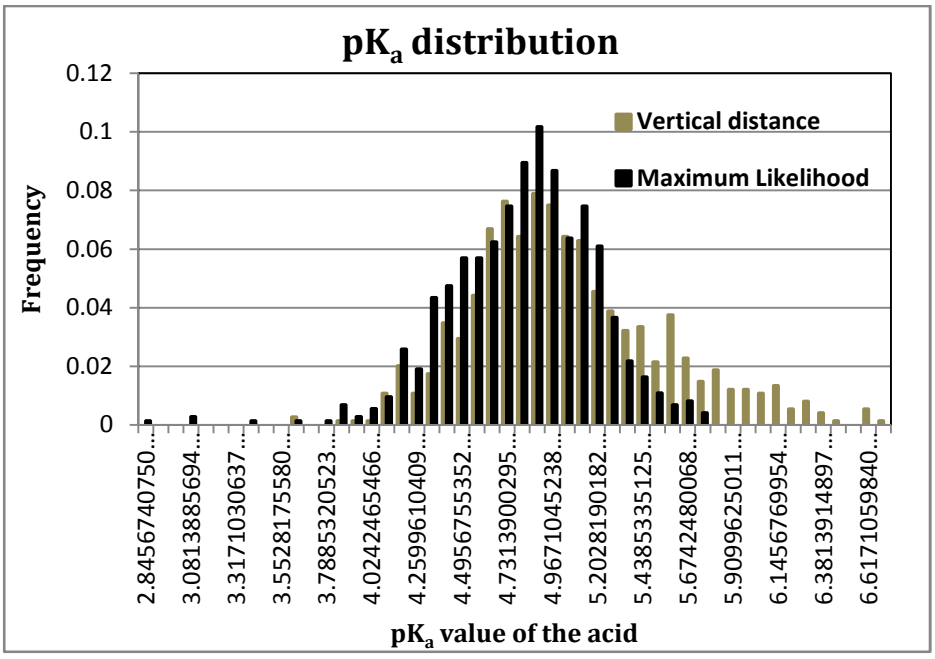


Figure 35b: pK_a distribution for vertical and maximum likelihood method for σ_x = 0.25, $\sigma_y = 0.5$

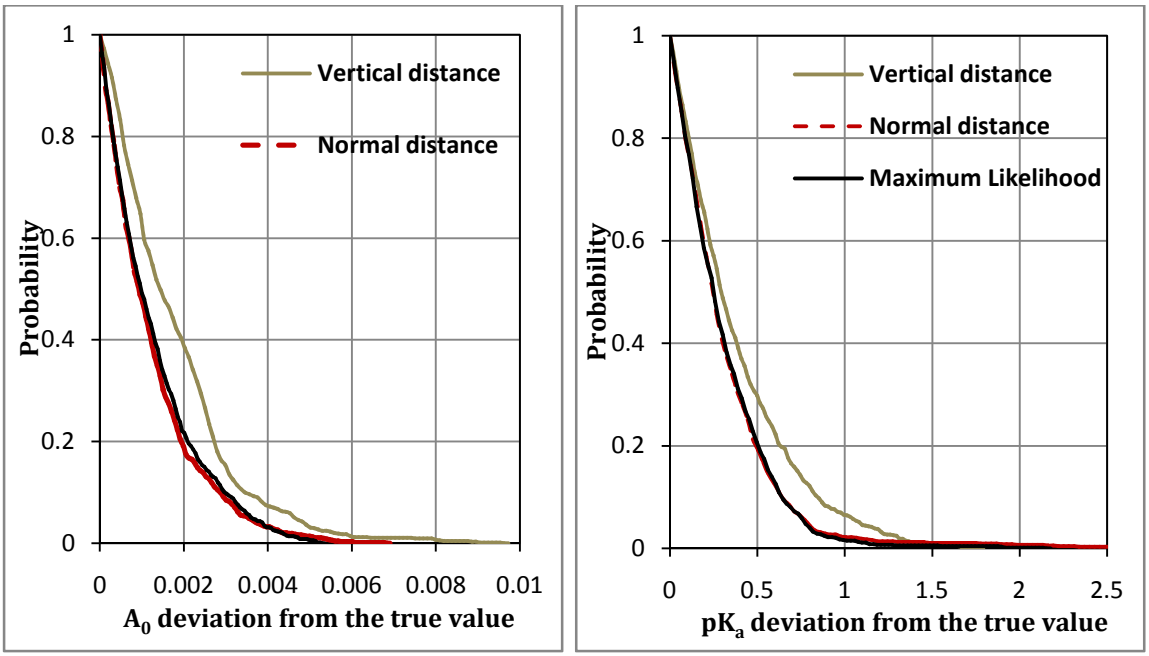


Figure 36: 'Probability of deviation from true value' plots for A_0 and pK_a for σ_x = 0.25, $\sigma_y = 0.5$

As expected from the tabulated results for σ_x = 0.25, $\sigma_y = 0.5$, the parameter distributions for the vertical distance method are more dispersed compared to the maximum likelihood. The probability plots also indicate the better of the methods, with the normal distance and the maximum likelihood curves almost overlapping each other, clearly separated from the vertical distance curves.

Hence from the above tests when $\sigma_x < \sigma_y$, the maximum likelihood predictions were at par with the best predictions in each sub-case. Among the vertical and normal distance methods, the vertical distance could be selected for low input uncertainties, but as the magnitude rises, the normal distance method could be chosen.

4.1.2 $\sigma_x = \sigma_y$

When $\sigma_x = \sigma_y$, the maximum likelihood method is exactly the same as the normal distance method. The predictions of the vertical and the maximum likelihood method were compared for σ values ranging from 0.1 to 0.7.

In each of the cases in Table 4.1.2, the maximum likelihood method provided better results than the vertical distance method. The variation in the results became more obvious as the σ values increased due to the initial premise of the vertical distance method.

		Vertical Distance		Maximum Likelihood	
		Bias	Standard Deviation	Bias	Standard Deviation
$\sigma = 0.1$	A_0	0.000113	0.000993	0.000068	0.000686
	pK_a	0.003953	0.193984	0.003967	0.119569
$\sigma = 0.25$	A_0	0.000652	0.002128	0.000019	0.001396
	pK_a	0.017518	0.420125	0.017119	0.216889
$\sigma = 0.5$	A_0	0.003161	0.005154	0.000236	0.003384
	pK_a	0.058861	0.737612	0.052285	0.467059
$\sigma = 0.7$	A_0	0.012160	0.036223	0.004937	0.040177
	pK_a	0.096469	0.998426	0.066694	0.727321

Table 4.1.2 Comparison of the regression methods for $\sigma_x = \sigma_y$

Typical distributions and the probability plots for σ_x = 0.25, $\sigma_y = 0.25$ are shown in the Figs. 37 and 38 respectively.

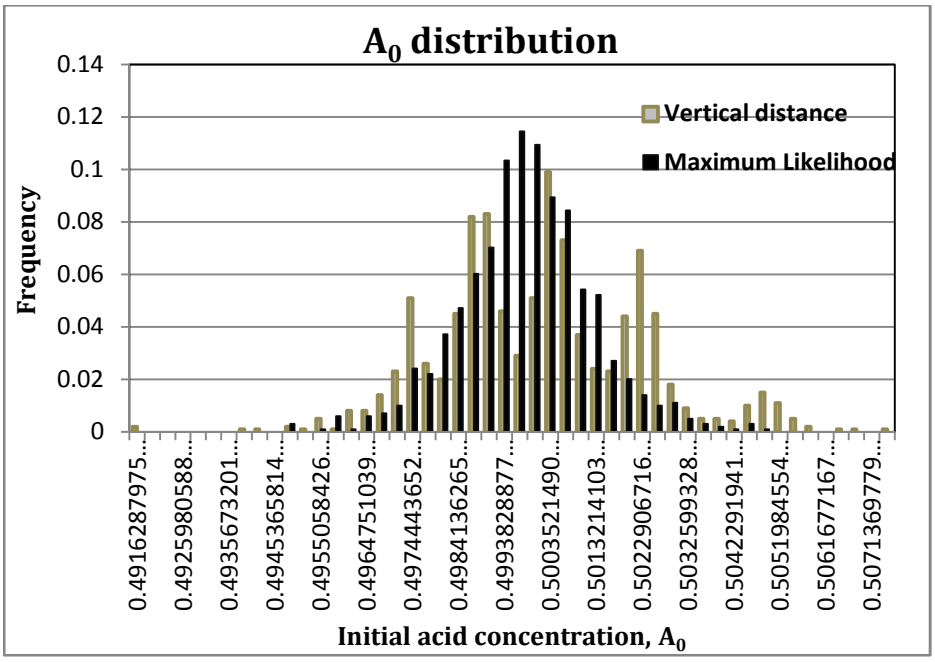


Figure 37a: A_0 distribution for vertical and maximum likelihood method for σ_x = 0.25, $\sigma_y = 0.25$

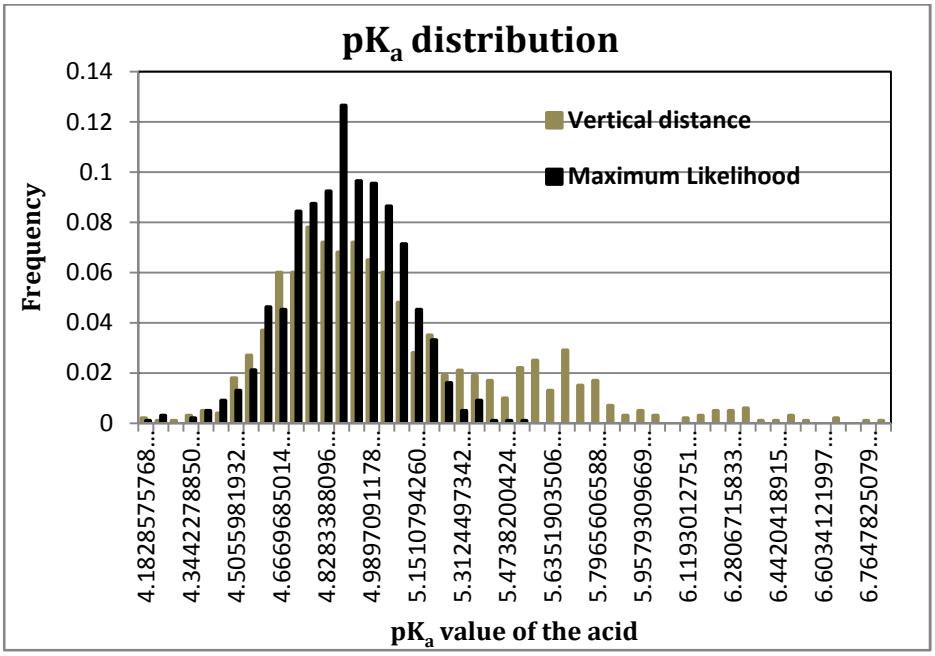


Figure 37b: pK_a distribution for vertical and maximum likelihood method for σ_x = 0.25, $\sigma_y = 0.25$

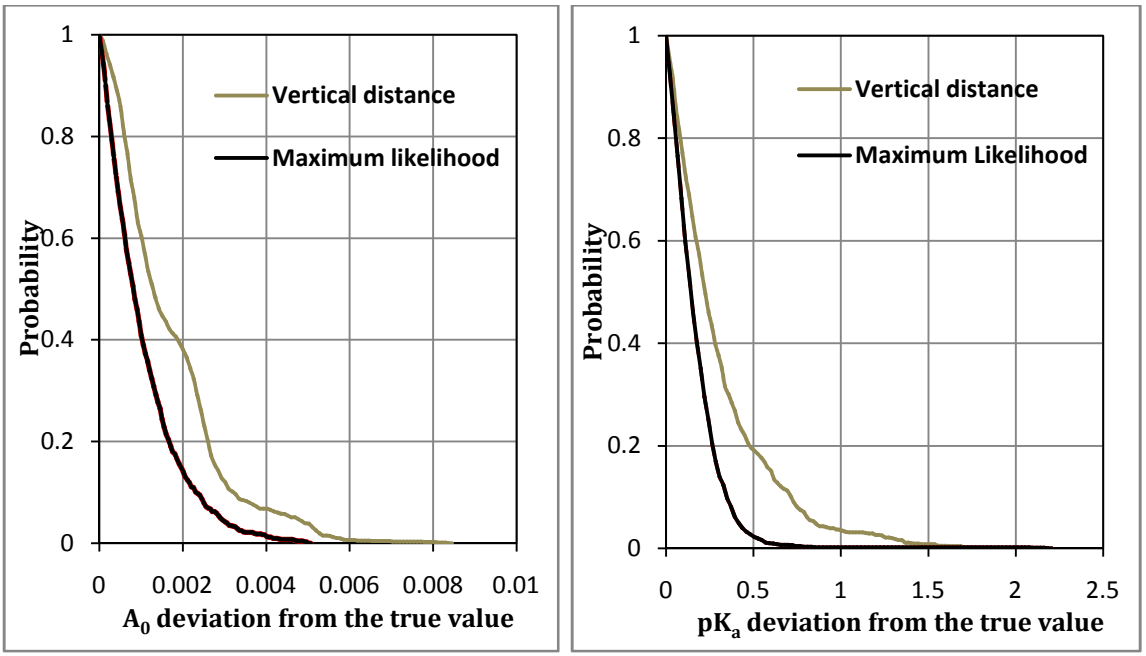


Figure 38: 'Probability of deviation from true value' plots for A_0 and pK_a for σ_x = 0.25, $\sigma_y = 0.25$

The nature of the distributions and the probability plots are similar to the ones shown in Figs. 35, and 36. The maximum likelihood hence, is the better of the methods when $\sigma_x = \sigma_y$

4.1.3 $\sigma_x > \sigma_y$

Theoretically, when $\sigma_x > \sigma_y$, the vertical distance method's supposition that σ_x can be neglected, no longer holds good, and its predictions further deteriorate compared to the previous two cases of $\sigma_x < \sigma_y$ and $\sigma_x = \sigma_y$. The normal distance method is a better representation of the situation than the vertical distance method. The results obtained from the simulations are presented in Table 4.1.3.

		Vertical Distance		Normal Distance		Maximum Likelihood	
		Bias	Standard Deviation	Bias	Standard Deviation	Bias	Standard Deviation
σ_x = 0.25	A_0	0.000915	0.002120	0.000076	0.001148	0.000517	0.001126
σ_y = 0.1	pK_a	0.022009	0.414877	0.015659	0.173191	0.012219	0.116610
σ_x = 0.5	A_0	0.003534	0.006926	0.000504	0.002300	0.002796	0.002199
σ_y = 0.1	pK_a	0.061475	0.684288	0.035114	0.276258	0.051410	0.262570
σ_x = 0.5	A_0	0.003467	0.004671	0.000499	0.002301	0.001878	0.002305
σ_y = 0.25	pK_a	0.068533	0.711066	0.035584	0.294769	0.053500	0.331088

Table 4.1.3 Comparison of the regression methods for $\sigma_x > \sigma_y$

From the above table, when σ_x = 0.25, σ_y = 0.1, while the results for the maximum likelihood method were better than the other two methods, there was a considerable difference in the standard deviations of pK_a for the vertical distance method with the other two methods. Increasing the uncertainty levels in the next two sub-cases, the predictions for the vertical distance method deteriorate, but while the results for the normal distance and the maximum likelihood method are at par for σ_x = 0.5, σ_y = 0.1, the preference shifts to normal distance method for the last sub-case. The parameter

distributions and the probability curves for a prototype case of σ_x = 0.25, σ_y = 0.1 were discussed previously through Figs. 33 and 34 respectively.

Hence while the maximum likelihood works well for lower uncertainty values when $\sigma_x > \sigma_y$, the normal distance could be adopted for high uncertainty values.

In all the above cases, the σ_x and σ_y values for evaluating the maximum likelihood objective function were taken the same as the true, but practically unknowable uncertainty values, chosen to generate the experimental data. However, it is highly improbable to get the exact estimate of the uncertainty associated with any measurement. Hence to test the feasibility of maximum likelihood method in realistic situations, the σ_x and σ_y used for the maximum likelihood objective function evaluation, were approximated by values deviating by around 50% from the true values chosen to generate the experimental data.

The σ_x values for all the data pairs were perturbed by the same magnitude from the true value, assuming the uncertainty in all the input measurements would more or less be the same. Hence if the true σ_x value is 0.1, the perturbed value would be a number close to 0.05 or 0.15. The σ_y values were perturbed by different magnitudes for each data pair as the uncertainty level would vary depending upon the linear/nonlinear region of the model curve. The values were generated at random through a Visual Basic code written as

$$\sigma_{perturbed} = \sigma_{true} + (Rnd() - 0.5) * \sigma_{true} \tag{97}$$

The σ_x and σ_y values for each data pair however, were maintained the same for all realizations. The results were tested for simulations with σ_x = 0.1, σ_y = 0.25 and

σ_x = 0.25, σ_y = 0.1. In both the cases the maximum likelihood and the normal distance were equivalent and were better than the vertical distance method. The results are tabulated in Table 4.1.4

		Vertical Distance		Normal Distance		Maximum Likelihood	
		Bias	Standard Deviation	Bias	Standard Deviation	Bias	Standard Deviation
σ_x = 0.1	A_0	0.000132	0.000985	0.000046	0.000806	0.000068	0.000827
σ_y = 0.25	pK_a	0.002532	0.234873	0.008217	0.219850	0.007635	0.175719
σ_x = 0.25	A_0	0.000915	0.002120	0.000076	0.001148	0.000200	0.001258
σ_y = 0.1	pK_a	0.022009	0.414877	0.015659	0.173191	0.009392	0.122903

Table 4.1.4 Comparison of the regression methods for approximate variances for Maximum likelihood

The parameter histograms and the probability plots for the simulation with the experimental data generated through σ_x = 0.1, σ_y = 0.25 are shown in Figs. 39 and 40 respectively.

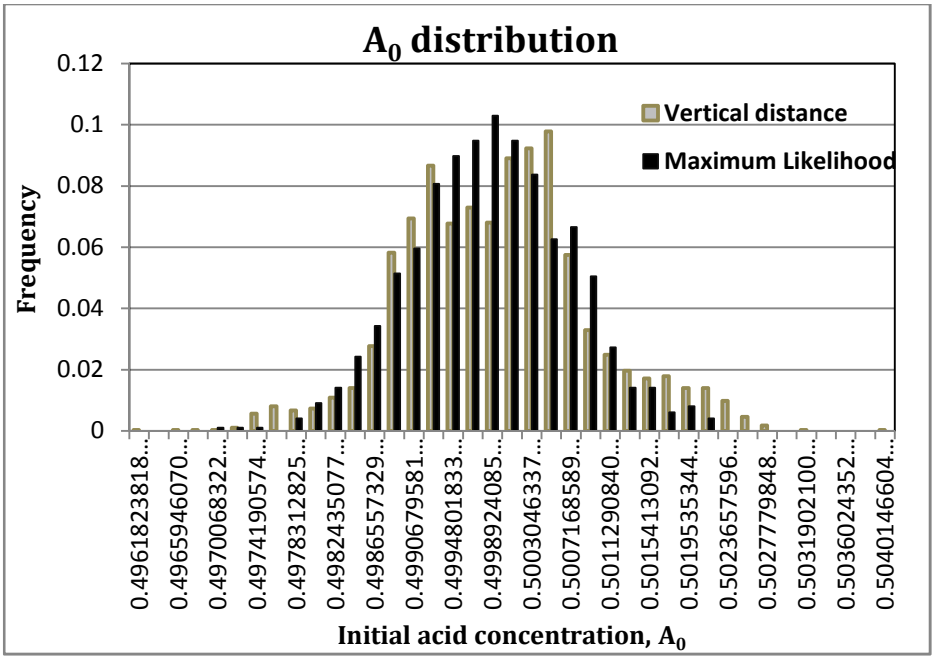


Figure 39a: A_0 distribution for vertical and maximum likelihood method for σ_x = 0.1, σ_y = 0.25 with perturbed variance values for maximum likelihood objective function

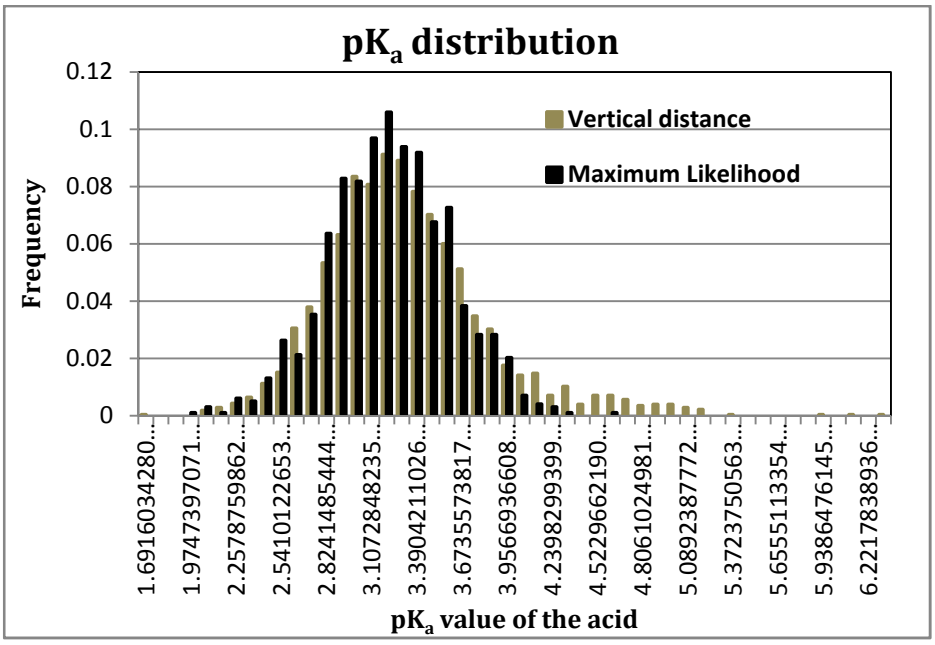


Figure 39b: pK_a distribution for vertical and maximum likelihood method for σ_x = 0.1, $\sigma_y = 0.25$ with perturbed variance values for maximum likelihood objective function

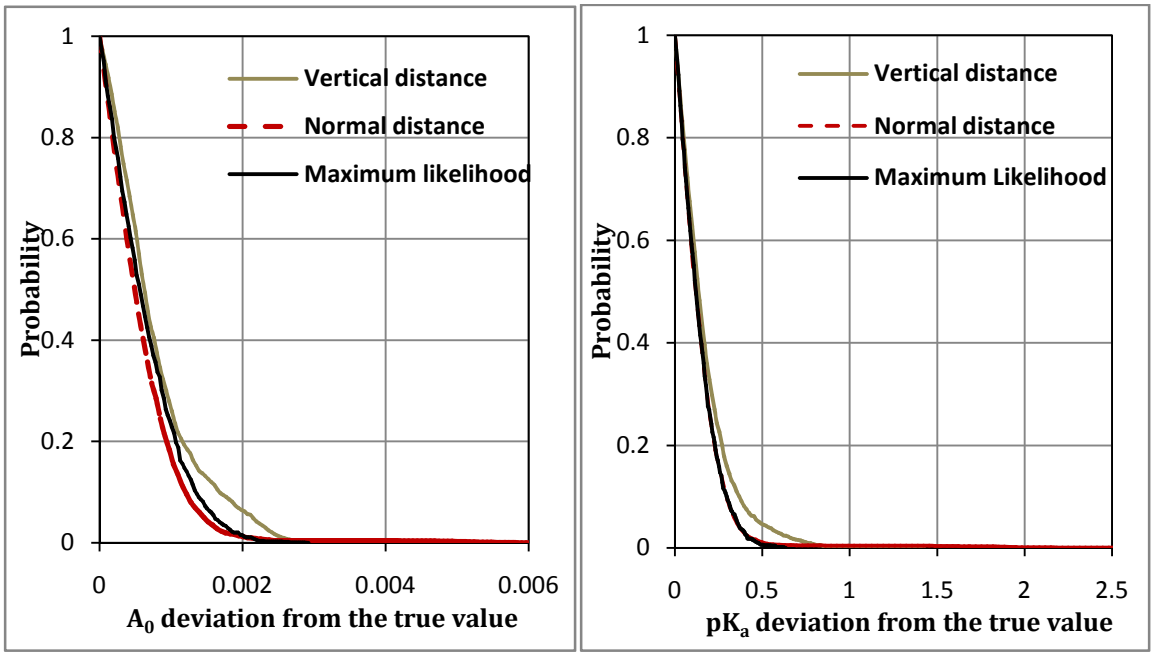


Figure 40: 'Probability of deviation from true value' plots for A_0 and pK_a for σ_x = 0.1, $\sigma_y = 0.25$ with perturbed variance values for maximum likelihood objective function

While the parameter distribution for the vertical distance were more dispersed than the maximum likelihood method, the probability plots were almost overlapping for

all the three methods, but the normal distance had its tail extending till the end and the vertical distance curve was slightly above the two.

The normal distance method, which is the same as the maximum likelihood method, but with $\sigma_x = \sigma_y$, can also be considered as a case of the variances for the maximum likelihood objective function evaluation deviating from the true values.

Hence from the above tests, the following could be concluded

1. When $\sigma_x < \sigma_y$, the maximum likelihood works well for all the sub-cases, but for lower magnitudes of uncertainty the vertical distance can be chosen to minimize computational burden.
2. When $\sigma_x = \sigma_y$, the maximum likelihood method works well for all the cases and is significantly better than the vertical distance method
3. When $\sigma_x > \sigma_y$, the maximum likelihood method could be chosen for lower uncertainty values, but as the uncertainty levels increases, the normal distance method would be a better option.

The findings are summarized in Table 4.1.5. The roman numericals in each case indicate the order of preference based on the results obtained.

		Vertical Distance	Normal Distance	Maximum Likelihood
1. Parameter Precision and Accuracy				
$\sigma_x < \sigma_y$				
	σ_x = 0.1 σ_y = 0.25	I	I	I
	σ_x = 0.1 σ_y = 0.5	I	II	I
	σ_x = 0.25 σ_y = 0.5	III	II	I
$\sigma_x = \sigma_y$				
	σ = 0.1	II	I	I
	σ = 0.25	II	I	I
	σ = 0.5	II	I	I
	σ = 0.7	II	I	I
$\sigma_x > \sigma_y$				
	σ_x = 0.25 σ_y = 0.1	II	I	I
	σ_x = 0.5 σ_y = 0.1	III	I	II
	σ_x = 0.5 σ_y = 0.25	III	I	II
2. Programming Burden		I	II	II
3. User Complexity		I	I	II

Table 4.1.5 Summary of Findings

4.2 PACKED BED SIMULATION

The packed bed simulations were more complex than the titration ones due to the three-parameter optimization, and the three-variable search for the objective function evaluations, for the normal distance and the maximum likelihood methods. Due to the large computational requirements, these simulations were not as comprehensively studied as the titration ones.

Based on the practical inference on the maximum possible uncertainties in the input variables C_{in}, v, T, and the output variable C_0, as $0.05 gmol/lit$, $0.1 ml/sec$, $1^0 C$ and $0.05 gmol/lit$, the variances were assigned values of 0.0167, 0.033, 0.333, and 0.0167 respectively. The simulations were run for a two hundred and fifty realizations, with the initialization of the parameters by 30% deviation from the true values. The results obtained are tabulated as follows.

	Vertical Distance		Normal Distance		Maximum Likelihood	
	Bias	Standard Deviation	Bias	Standard Deviation	Bias	Standard Deviation
k_0	0.227193	0.085813	0.284356	0.092555	0.292860	0.094249
k_1	0.229042	8.185125	0.290425	8.881987	0.303765	9.068874
E	0.004754	939.021278	0.004774	943.248619	0.005304	943.193565

Table 4.2.1 Comparison of the regression methods – Packed bed reactor

From the above table, the vertical distance method for all the parameters had slightly better predictions than the normal and the maximum likelihood methods. But, variation in the results is insignificant to affirm the better of the three. Hence, either of the methods could be chosen to regress the data. The parameter distributions and probability plots for

the vertical distance and the maximum likelihood methods for the above results are shown in Figs. 41 and 42

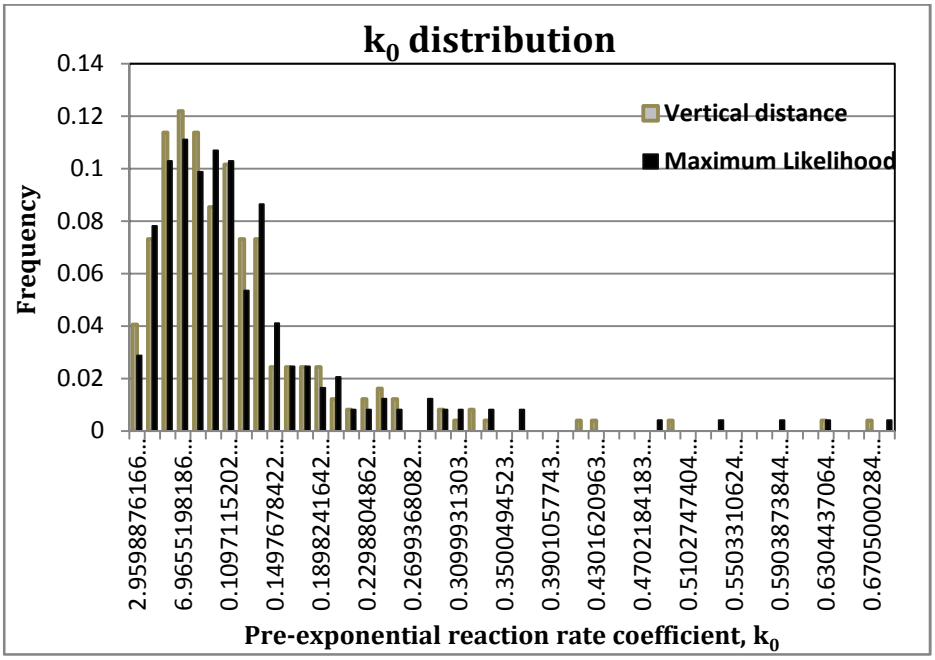


Figure 41a: Parameter 'k_0' distribution for vertical distance and maximum likelihood methods

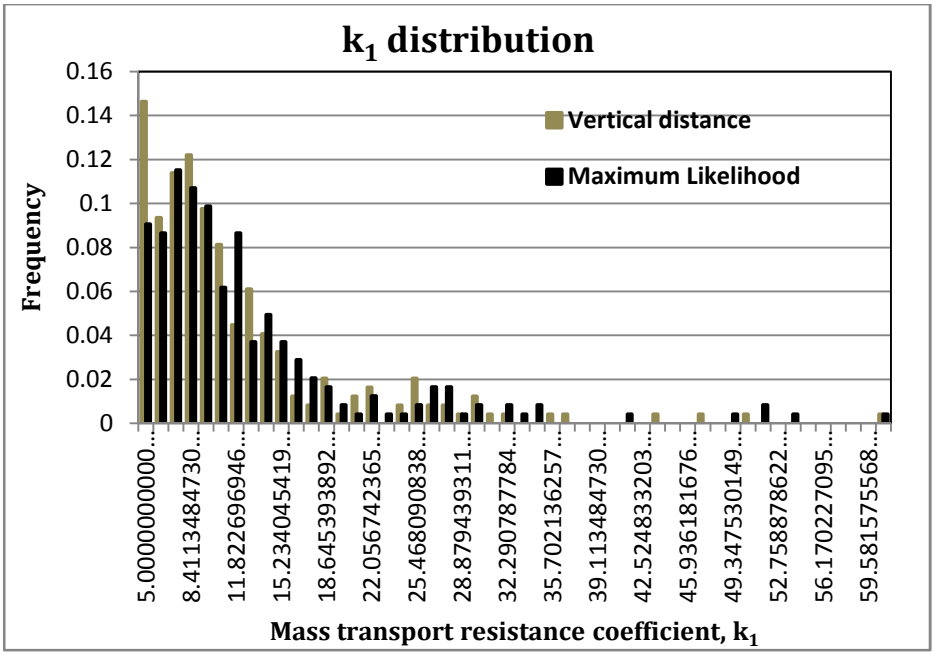


Figure 41b: Parameter 'k_1' distribution for vertical distance and maximum likelihood methods

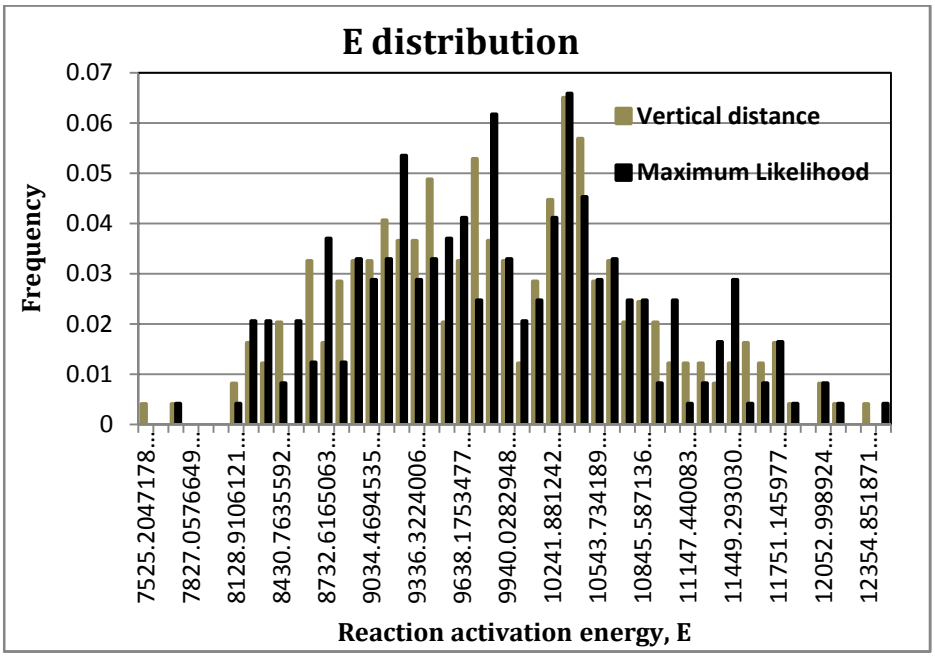


Figure 41c: Parameter 'E' distribution for vertical distance and maximum likelihood methods

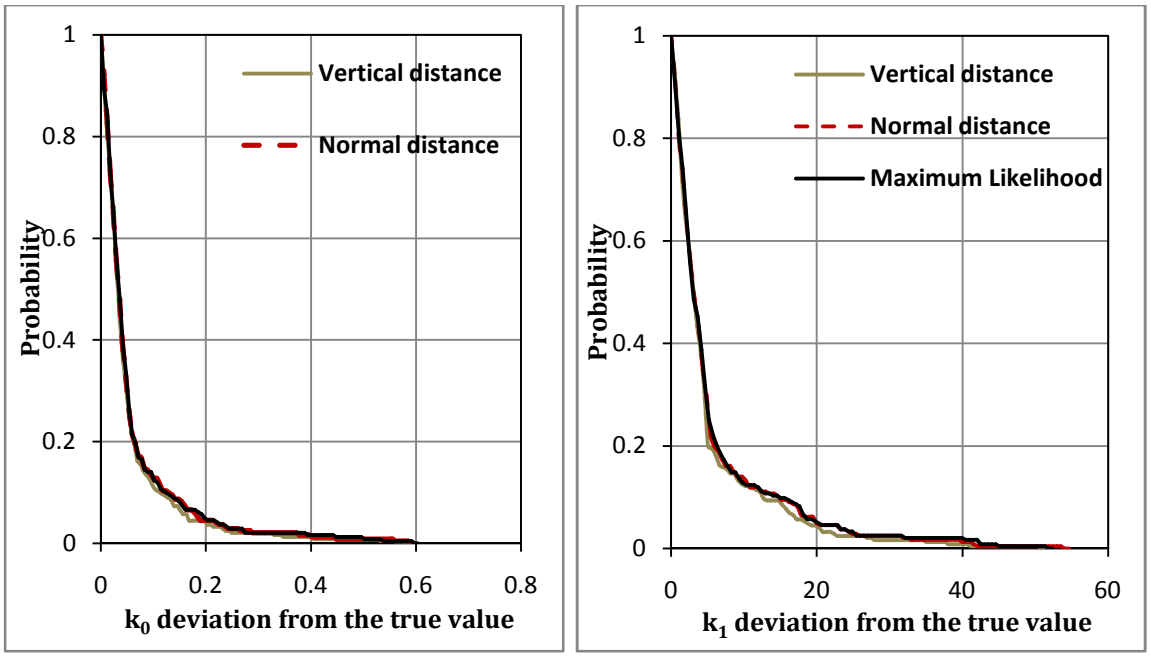


Figure 42 (a and b): Probability plots for packed bed regression parameters, k_0 and k_1

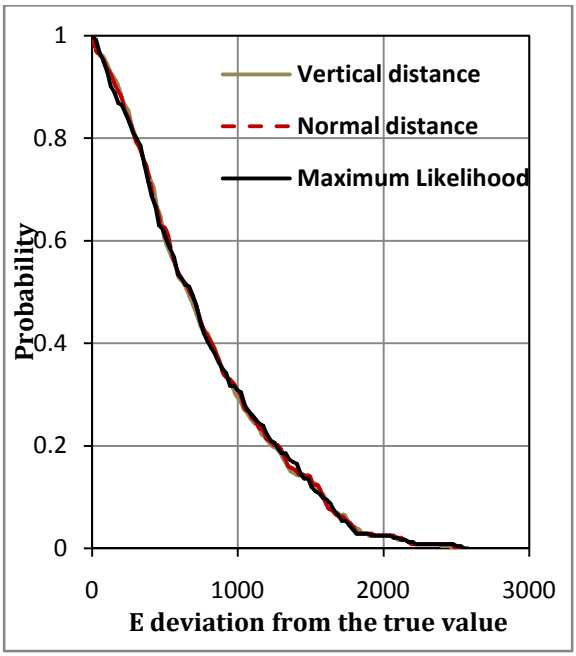


Figure 42c: Probability plots for packed bed regression parameter, E

As can be seen from the probability plots, the curves for all the methods were overlapping, suggesting all the methods were at par. The parameter histograms were also inconclusive and followed similar pattern for both the regression methods.

The computational burden was found to be more or less the same for all the methods, with the vertical distance requiring a lot more iterations than the maximum likelihood method. Due to the large time requirements, the simulations for the specified variances were only studied for two hundred and fifty realizations. Increasing the uncertainty levels, the parameter predictions for the vertical and the normal distance methods, constrained to the extreme limits, while the maximum likelihood method predicted reasonable values.

Hence, though the vertical distance method could be adopted for lower uncertainty levels, the maximum likelihood may have to be chosen for higher magnitudes of uncertainty.

5. MODEL VALIDATION

To ascertain the veracity of the code employed for the simulations, a few basic tests were conducted.

1. The vertical and the normal distance methods for both the titration and the packed bed simulations were tested for data with no experimental uncertainties in the input and output measurements ($\sigma_x = 0, \sigma_y = 0$). Both the methods yielded the exact same parameter values used in generating the data, confirming the trueness of the algorithm.

   Since with no experimental uncertainties the maximum likelihood objective function results with zeroes in the denominator, data generated with very small uncertainty values were used to test the method. The maximum likelihood method yielded parameter values close enough to the true ones, ascertaining the veracity of the code.

2. Another test adopted was to repeat the regression several times for the same set of data but with different initializations for the parameters, varying upto 50% of their true values. All the methods yielded the same parameter values as their previous estimation, but upto four decimal places for A_0 and two decimal places for pK_a, suggesting that each time the same optimum was achieved. The precision of the parameter values could be increased beyond the specified decimal places by further reducing the tolerance limits in the stopping criteria, but this would result in a greater

number of iterations, which did not seem necessary.

3. To test if a thousand realizations was a sufficient number to conclude the results, a few titration simulations were conducted by doubling the number of realizations. The mean and the standard deviation of the parameter distributions for the vertical distance method remained approximately the same, but there was considerable variation in the results for the normal and the maximum likelihood methods. Hence the titration simulations were run on an average for about 3000 realizations before concluding the results for the cases discussed earlier.
4. While evaluating the results, any outliers in the titration parameter estimates as negative values for A_0, or pK_a values less than 1.0 were omitted by deleting the entire results for that particular realization, for that particular method. The packed bed regression parameters were constrained between their nominal limits as defined in the problem statement.

   The number of outliers for the titration simulation were usually the highest for the maximum likelihood method followed by the normal distance and the vertical distance method. However the percentage of outliers to the total number of realizations were less than 1% in most cases for all the methods, but were as high as upto 6% for certain cases with high variance values for the maximum likelihood method ((i). $\sigma_x = \sigma_y = 0.7$, (ii). $\sigma_x = 0.5, \sigma_y = 0.1$, (iii). $\sigma_x = 0.5, \sigma_y = 0.25$). For the packed bed simulation, the number of realizations with parameters constraining to the extreme limits were highest for the vertical distance method, followed by normal distance and the maximum likelihood methods.

6. CONCLUSIONS

1. Based on the results of the titration and the packed bed simulation the maximum likelihood method worked best in most cases. The vertical and the normal distance methods have individual preferences over the other depending upon the relative magnitude of the input and output uncertainties. The reality check for the maximum likelihood method through approximate variance values (deviation upto 40% from the true values) while evaluating the objective function, also yielded better results than the other two methods, suggesting the scope of an educated guess for the variances if a close estimate cannot be determined.

2. However, the maximum likelihood and the normal distance methods are a lot more computationally intensive than the vertical distance method. And, since for lower uncertainty levels there isn't a big variation in the results of either of the methods, the vertical distance method can be safely implemented without sacrificing much accuracy. However, for higher uncertainty levels the need for a better method is obvious.

3. Though a few researchers have investigated the possibility of the maximum likelihood method, they have concluded in the favor the vertical distance method due to the complexity and the computational burden [1]. However, one reason for the computational burden could be the optimization algorithm. Most researchers seemed to have tried the gradient based algorithms such as the Levenberg Marquardt,

Cauchy's steepest descent, Newton-Raphson etc. However the R^3 cyclic direct search considerably eases the search process with good consistency in the parameter predictions for distant initializations from the true values.

The vertical distance method does take a lot lower time for the titration simulations (approximately one –tenth of the time for the other two methods), but the R^3 cyclic direct search is a good improvisation for the search process over the gradient based methods. The time consumed for the packed bed simulation by all the methods was more or less the same, with the vertical distance method requiring a lot more iterations on average than the maximum likelihood method, thereby compromising on its simpler logic.

4. The parameters of the titration and the packed bed simulation were covariant. That is, they were inter-dependent and were not individually optimized without affecting the other. This has been cited in the view of one of the statements in [2] explaining the necessity for covariant parameters in computer simulations.

5. It is important to understand that the results of either of the simulations do not necessitate the certainty of always obtaining better predictions through the best method. It only reflects the higher probability. Hence, if an approximate estimate of parameters is known, the data could be regressed through all the three methods, and the predicted parameters closest to the approximate estimates could be selected.

REFERENCES

1. Johnson M. L., Frasier S. G., Nonlinear Least – Square Analysis, *Methods in Enzymology* [Online] **1985**, 117, pp. 301 – 342.
2. Johnson M. L., Use of Least-Squares Techniques in Biochemistry, *Methods in Enzymology* [Online] **1994**, 240, pp. 1 – 22.
3. Rhinehart R. R., Optimization Applications – Lecture notes. Oklahoma State University, 2008.
4. Leng L., Zhang T., Kleinman L., Zhu Wei., Ordinary Least Square Regression, Orthogonal Regression, Geometric Mean Regression and their Applications in Aerosol Science, *Journal of Physics* [Online] **2007**, 78, pp.1-5.
5. Draper N. R., Yang Y., Generalization of geometric mean functional relationship, *Computational Statics and Data Analysis* [Online], **1995**, 23, pp. 355-372.
6. Weibull.com, http://www.weibull.com/LifeDataWeb/least_squares.htm, (accessed: 20th February, 2009)
7. Weisstein, E. W., "Least Squares Fitting-Perpendicular Offsets", WolframMathWorld.com, http://mathworld.wolfram.com/LeastSquaresFittingPerpendicularOffsets.html, (accessed: 20th February, 2009).
8. Sampaio Jr. J. H. B., An iterative procedure for perpendicular offsets linear least squares fitting with extension to multiple regression, *Applied Mathematics and Computation* [Online] **2006**, 176, pp. 91-98.

9. Riggs D. S., Guarnieri J. A., Addelman S., Fitting straight lines when both variables are subject to error, *Life Sciences* [Online] **1978**, 22, pp. 1305-1360.
10. Shotaro A., Curve fitting that minimizes the mean square of perpendicular distances from sample points, *citeseerx.ist.psu.edu*, (accessed: 20th February, 2009).
11. Wikipedia.org, http://en.wikipedia.org/wiki/Normal_distribution, (accessed: 20th February, 2009).
12. Wikipedia.org, http://en.wikipedia.org/wiki/Interval_halving, (accessed: 20th February, 2009).
13. Griliches Z., Ringstad V., Error – in – the variables bias in nonlinear contexts, *Econometrica* [Online] **1970**, 38, pp.368-370.
14. Wikipedia.org, http://en.wikipedia.org/wiki/Newton's_method, (accessed: 20th February, 2009).
15. Padmanabhan V., Rhinehart. R.R., A novel termination criterion for optimization, *American Controls Conference* [Online] **2005**, pp. 1042-1047
16. Cao S., Rhinehart., R.R. Critical values for a steady-state identifier, *J. Proc. Cont.* [Online] **1997**, 7, 2, pp.149-152

APPENDIX A – TITRATION SIMULATOR CODE

The following Visual Basic code was adopted for the titration simulation.

Code:

```
Option Explicit
' Author :  Chetan Chandak
' Created date: 20- Dec-2007
' Description        : Titration model to test regression methods
'_____________________________________________________________________________________

' _____________________________________Declaring Common Variables_____________________

Global A0 As Double, h1 As Double       ' A0, pKa: model parameters
Global pKa As Double, h2 As Double      ' h1, h2 are the step increments
Global T(0 To 2) As Double              ' stores the objective function value
                                        ' T(0): least objective function value at the time of reading
                                        ' T(1): current objective function value at the time of reading
                                        ' T(2): objective function value before a new set of changes in
                                        ' the parameters.

Global Vmax As Double                   ' max volume of base (apparent) added during the titration process
Global Vmin As Double                   ' min volume of base (apparent) added.
Global pHmax As Double                  ' max pH (apparent) attained during the titration process
Global pHmin As Double                  ' min pH (apparent) attained during the titration process

Global z As Integer                     ' # realizations (variable)
Global N As Integer                     ' # best of N trials
Global w As Integer                     ' # best of N trials (variable)
Global p As Integer                     ' # iterations in the optimization process (variable)
Global rand(1 To 100) As Double         ' random # for initial guesses to start the prediction of pka & A0
'_____________________________________________________________________________________

' _____________________________________Experimental Data Generation Module_____________

Sub Regression()
Dim vn As Double                        ' noise to the apparent volume
Dim v As Double                         ' true but unknowable volume
Dim pHun As Double                      ' true but unknowable pH corresponding to v
Dim pH As Double                        ' apparent pH

Dim i As Integer                        ' # experimental data points
Dim r As Integer                        ' # realizations
Dim sig(1 To 2) As Double               ' std dev (noise) in the pH and volume readings

'-----------------------------------------------------Initialize Variables-----------------------------------------------

N = Sheet4.Cells(13, 8).Value
r = Sheet4.Cells(10, 8).Value
sig(1) = Sheet4.Cells(11, 8).Value
sig(2) = Sheet4.Cells(12, 8).Value

Vmin = Sheet4.Cells(15, 1).Value
Vmax = Sheet4.Cells(16, 1).Value
pHmax = Sheet4.Cells(16, 2).Value
```

```
pHmin = Sheet4.Cells(15, 2).Value

'----------Data Generation----------

For z = 1 To r Step 1
    A0 = Sheet4.Cells(3, 8).Value
    pKa = Sheet4.Cells(2, 8).Value

    Randomize
        For i = 1 To N Step 1
        rand(i) = (1 + (3 * (Rnd() - 0.5) * 0.2))       'range from 0.7 to 1.3
    Next i

    For i = 1 To 8
         vn = sig(1) * Sqr(-2 * Log(Rnd())) * Sin(2 * 3.14159 * Rnd())
        v = Sheet4.Cells(6 + i, 1).Value + vn

        pHun = pHpredict(v)
        pH = pHun + sig(2) * Sqr(-2 * Log(Rnd())) * Sin(2 * 3.14159 * Rnd())
        Sheet4.Cells(6 + i, 2) = pH

        Sheet4.Cells(6 + i, 4) = v
        Sheet4.Cells(6 + i, 5) = pHun
    Next i

    Call data_substitution          ' prints expt. data in the vertical, normal distance and max. likelihood
                                    ' worksheets

    Sheet5.Activate                 ' runs the vertical distance module
    Call vertical

    Sheet1.Activate                 ' runs the normal distance module
    Call Normal

    Sheet6.Activate                 ' runs the Maximum likelihood module
    Call Max_Likelihood

Next z

End Sub
'----------pH Evaluation----------

Function pHpredict(v As Double)

' Predicts pH for a given volume of base added
' Code taken from Dr. Rhinehart's excel file

Dim pHmin As Double, pHmax As Double, pHmid As Double
Dim fmin As Double, fmax As Double, fmid As Double
Dim j As Integer

pHmax = 14
pHmin = 0
```

```
For j = 1 To 20                         ' Interval Halving Method
   fmax = func(pHmax, v)
   fmin = func(pHmin, v)
   pHmid = (pHmax + pHmin) / 2
   fmid = func(pHmid, v)

   If (fmax * fmid < 0) Then
        pHmin = pHmid
        fmin = fmid
   Else
        pHmax = pHmid
        fmax = fmid
   End If
Next j

If pHmid < pHmin Then pHmid = pHmin
If pHmid > pHmax Then pHmid = pHmax

pHpredict = pHmid

End Function
' ----------------------------------------------------------------------------------------------------------------------------

Function func(pH As Double, v As Double)
'Code taken from Dr. Rhinehart's excel file

Dim x As Double
Dim y As Double
Dim a As Double
Dim b As Double
Dim ka As Double
Dim Hconc As Double

Dim b0 As Double
Dim kw As Double

b0 = Sheet4.Cells(4, 8).Value
kw = 0.0000000000001

ka = 10 ^ (-pKa)
a = A0 / (1 + (v / 1000))
b = (b0 * (v / 1000)) / (1 + (v / 1000))

x = a * ka / (ka + (10 ^ -pH))
y = (kw * (10 ^ pH)) - b
Hconc = x + y
func = (10 ^ -pH) - Hconc

End Function
' ----------------------------------------------------------------------------------------------------------------------------
```

```
' ----------------------------------------------------------------------------------------------------------------------------

Sub data_substitution()
'Substitutes the experimental data in the normal, vertical and max_likelihood sheets
'
' data_substitution Macro
' Macro recorded 3/12/2008 by chetan
'
    Sheets("data  generation").Select
    Sheet4.Range("A7:B14").Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Vertical_dist").Select
    Sheet5.Range("B12").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Sheets("Normal_dist").Select
    Sheet1.Range("B12").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Sheets("Max_Likelihood").Select
    Sheet6.Range("B12").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False

End Sub

'_____________________________________________________________________________________________

' _____________________________________Vertical Distance Module_____________________________

'-----------------------------------------------------Parameter Optimization-----------------------------------------

Sub vertical()
Dim best As Double                  ' best objective function value from N trials
Dim g As Integer                    ' stores the iteration corresponding to the best objective function value

Sheet5.Range(Cells(4, 9), Cells(1000, 15)).ClearContents

best = 10000

'-----------------------------------------------------R^3 Cyclic Direct Search----------------------------------------

For w = 1 To N Step 1

    A0 = Sheet4.Cells(3, 8).Value * rand(w)
    pKa = Sheet4.Cells(2, 8).Value * rand(w)

    Sheet5.Cells(8, 6) = pKa
    Sheet5.Cells(9, 6) = A0

    Sheet5.Cells(3 + w, 9) = w
    Sheet5.Cells(3 + w, 10) = A0
    Sheet5.Cells(3 + w, 11) = pKa

    T(0) = sumvert
    Sheet5.Cells(9, 2) = T(0)
```

```
T(2) = T(0)                          'storing the initial objective fn value in T(2)

h1 = A0 * 0.1                        'Initializing step increments
h2 = pKa * 0.1

For p = 1 To 200 Step 1

   A0 = A0 + h1
   Sheet5.Cells(9, 6) = A0
   Sheet5.Cells(9, 7) = h1
   T(1) = sumvert
   Sheet5.Cells(9, 2) = T(1)

   If (T(1) < T(0)) Then
      T(0) = T(1)                    'T(0) = best possible value till now
      h1 = h1 * 1.25                 'Expansion factor
   Else
      A0 = A0 - h1
      h1 = -h1 * (0.75 / 1.25)       'Contraction factor
      Sheet5.Cells(9, 6) = A0
      Sheet5.Cells(9, 7) = h1
   End If

   pKa = pKa + h2
   Sheet5.Cells(8, 6) = pKa
   Sheet5.Cells(8, 7) = h2
   T(1) = sumvert
   Sheet5.Cells(9, 2) = T(1)

   If (T(1) < T(0)) Then
      T(0) = T(1)
      h2 = h2 * 1.25                 'Expansion factor
   Else
      pKa = pKa - h2
      h2 = -h2 * (0.75 / 1.25)       'Contraction factor
      Sheet5.Cells(8, 6) = pKa
      Sheet5.Cells(8, 7) = h2
      T(1) = sumvert
      Sheet5.Cells(9, 2) = T(1)
   End If

   Sheet5.Cells(8, 2) = p

   If Abs(h1) < (10 ^ -10) And Abs(h2) < (10 ^ -10) And Abs(T(2) - T(0)) < (10 ^ -10) Then Exit For
   T(2) = T(0)

Next p

Sheet5.Cells(9, 7) = Abs(h1)
Sheet5.Cells(8, 7) = Abs(h2)

Sheet5.Cells(3 + w, 12) = A0
Sheet5.Cells(3 + w, 13) = pKa
Sheet5.Cells(3 + w, 14) = T(0)
Sheet5.Cells(3 + w, 15) = p
```

```
    If T(0) < best Then                 ' Best of N check
        best = T(0)
        g = w + 3
    End If

Next w

A0 = Sheet5.Cells(g, 12)
pKa = Sheet5.Cells(g, 13)
p = Sheet5.Cells(g, 15)
T(0) = sumvert

Sheet5.Cells(z + 24, 1) = z
Sheet5.Cells(z + 24, 2) = A0
Sheet5.Cells(z + 24, 3) = pKa
Sheet5.Cells(z + 24, 4) = T(0)
Sheet5.Cells(z + 24, 5) = p

End Sub

'-----------------------------------------------------------Objective function calculation-----------------------------------

Function sumvert() As Double

Dim Vi As Double                    ' Subscript i refers to experimental terms
Dim pHi As Double                   ' Subscript r refers to regressed terms
Dim pHr As Double

Dim d As Double
Dim sum As Double

Dim q As Integer                    ' # experimental data points

sum = 0
For q = 1 To 8 Step 1

    Vi = Sheet5.Cells(q + 11, 2).Value
    pHi = Sheet5.Cells(q + 11, 3).Value

    pHr = pHpredict(Vi)
    d = (pHi - pHr) ^ 2
    sum = sum + d

    Sheet5.Cells(q + 11, 4) = Vi
    Sheet5.Cells(q + 11, 5) = pHr
    Sheet5.Cells(q + 11, 6) = d

Next q

Sheet5.Cells(20, 6) = sum
sumvert = sum

'Worksheets("Vertical_dist").Calculate

End Function
' ----------------------------------------------------------------------------------------------------------------------------
```

```
'_____________________________________________________________________________________________
' _____________________________________Normal Distance Module_____________________________________
'-----------------------------------------------------Parameter Optimization-----------------------------------------

Sub normal()
Dim best As Double                  ' best objective function value from N trials
Dim g As Integer                    ' stores the iteration corresponding to the best objective function value

Sheet1.Range(Cells(4, 9), Cells(1000, 15)).ClearContents

best = 10000

'-----------------------------------------------------R^3 Cyclic Direct Search-----------------------------------------

For w = 1 To N Step 1

   A0 = Sheet4.Cells(3, 8).Value * rand(w)
   pKa = Sheet4.Cells(2, 8).Value * rand(w)

   Sheet1.Cells(8, 6) = pKa
   Sheet1.Cells(9, 6) = A0
   Sheet1.Cells(3 + w, 9) = w
   Sheet1.Cells(3 + w, 10) = A0
   Sheet1.Cells(3 + w, 11) = pKa

   T(0) = norm
   Sheet1.Cells(9, 2) = T(0)

   T(2) = T(0)                      'storing the initial objective fn value in T(2)

   h1 = A0 * 0.1                    'Initializing step increments
   h2 = pKa * 0.1

   For p = 1 To 200 Step 1

      A0 = A0 + h1
      Sheet1.Cells(9, 6) = A0
      Sheet1.Cells(9, 7) = h1
      T(1) = norm
      Sheet1.Cells(9, 2) = T(1)

      If (T(1) < T(0)) Then
         T(0) = T(1)                'T(0) = best possible value till now
         h1 = h1 * 1.25             'Expansion factor
      Else
         A0 = A0 - h1
         h1 = -h1 * (0.75 / 1.25)   'Contraction factor
         Sheet1.Cells(9, 6) = A0
         Sheet1.Cells(9, 7) = h1
      End If

      pKa = pKa + h2
      Sheet1.Cells(8, 6) = pKa
      Sheet1.Cells(8, 7) = h2
      T(1) = norm
      Sheet1.Cells(9, 2) = T(1)
```

```
        If (T(1) < T(0)) Then
            T(0) = T(1)
            h2 = h2 * 1.25                  'Expansion factor
        Else
            pKa = pKa - h2
            h2 = -h2 * (0.75 / 1.25)        'Contraction factor
            Sheet1.Cells(8, 6) = pKa
            Sheet1.Cells(8, 7) = h2
            T(1) = norm
            Sheet1.Cells(9, 2) = T(1)
        End If

        Sheet1.Cells(8, 2) = p

        If Abs(h1) < (10 ^ -10) And Abs(h2) < (10 ^ -10) And Abs(T(2) - T(0)) < (10 ^ -10) Then Exit For
        T(2) = T(0)

    Next p

    Sheet1.Cells(9, 7) = Abs(h1)
    Sheet1.Cells(8, 7) = Abs(h2)

    Sheet1.Cells(3 + w, 12) = A0
    Sheet1.Cells(3 + w, 13) = pKa
    Sheet1.Cells(3 + w, 14) = T(0)
    Sheet1.Cells(3 + w, 15) = p

    If T(0) < best Then                 ' Best of N check
        best = T(0)
        g = w + 3
    End If

Next w

A0 = Sheet1.Cells(g, 12)
pKa = Sheet1.Cells(g, 13)
p = Sheet1.Cells(g, 15)
T(0) = norm

Sheet1.Cells(z + 24, 1) = z
Sheet1.Cells(z + 24, 2) = A0
Sheet1.Cells(z + 24, 3) = pKa
Sheet1.Cells(z + 24, 4) = T(0)
Sheet1.Cells(z + 24, 5) = p

End Sub

'-----------------------------------------------------Objective function calculation-----------------------------------

Function norm() As Double

Dim Vi As Double                    ' Subscript i refers to experimental terms
Dim pHi As Double                   ' Subscript r refers to regressed terms
Dim Vr As Double
Dim pHr As Double
```

```
Dim Va As Double                    ' Va, Vb: lower and higher limits of the golden section line search
Dim Vb As Double
Dim Vl As Double                    ' Vl, Vh: intermediate lower and higher limits of golden section search
Dim Vh As Double
Dim pH As Double
Dim l As Double                     ' l = Vb - Va
Dim q As Integer
Dim sum As Double
Dim d(1 To 2) As Double
'-----------------------------------------------------Golden Section Search-----------------------------------------------

sum = 0
For q = 1 To 8 Step 1

   Vi = Sheet1.Cells(q + 11, 2).Value
    pHi = Sheet1.Cells(q + 11, 3).Value

    Va = Vmin * 0.9                 ' Initializing the golden section boundary limits
    Vb = Vmax * 1.1

    Do
       l = Vb - Va
       Vl = Va + (0.382 * l)
       Vh = Vb - (0.382 * l)

       pH = pHpredict(Vl)
       d(1) = dist(Vi, pHi, Vl, pH, 1, 1)

       pH = pHpredict(Vh)
       d(2) = dist(Vi, pHi, Vh, pH, 1, 1)

       If d(2) < d(1) Then
          Va = Vl
          Vr = Vh
       Else
          Vb = Vh
          Vr = Vl
       End If

    Loop Until Abs(d(2) - d(1)) < 0.00001 And l < 0.00001       ' Stopping criteria

    pHr = pHpredict(Vr)
    d(2) = dist(Vi, pHi, Vr, pHr, 1, 1)

    Sheet1.Cells(q + 11, 4) = Vr
    Sheet1.Cells(q + 11, 5) = pHr
    Sheet1.Cells(q + 11, 6) = d(2)
    sum = sum + d(2)

Next q

Sheet1.Cells(20, 6) = sum
norm = sum

End Function
' ------------------------------------------------------------------------------------------------------------------------
```

```
' ------------------------------------------------------------------------------------------------------------------------

'Objective function
Function dist(v1 As Double, pH1 As Double, v2 As Double, pH2 As Double, sigx As Double, sigy As
Double) As Double

    dist = ((v2 - v1) / (sigx * (Vmax - Vmin))) ^ 2 + ((pH2 - pH1) / (sigy * (pHmax - pHmin))) ^ 2

End Function
'_____________________________________________________________________________________________

' _____________________________________Maximum Likelihood Module_____________________________

'-----------------------------------------------------------Parameter Optimization-----------------------------------------

Sub Max_Likelihood()
Dim best As Double                  ' best objective function value from N trials
Dim g As Integer                    ' stores the iteration corresponding to the best objective function value

Sheet6.Range(Cells(4, 9), Cells(1000, 15)).ClearContents
best = 10000

'-----------------------------------------------------------R^3 Cyclic Direct Search----------------------------------------

For w = 1 To N Step 1

   A0 = Sheet4.Cells(3, 8).Value * rand(w)
   pKa = Sheet4.Cells(2, 8).Value * rand(w)

   Sheet6.Cells(8, 6) = pKa
   Sheet6.Cells(9, 6) = A0

   Sheet6.Cells(3 + w, 9) = w
   Sheet6.Cells(3 + w, 10) = A0
   Sheet6.Cells(3 + w, 11) = pKa

   T(0) = MaxP
   Sheet6.Cells(9, 2) = T(0)

   T(2) = T(0)                      'storing the initial objective fn value in T(2)

   h1 = A0 * 0.1                    'Initializing step increments
   h2 = pKa * 0.1

   For p = 1 To 200 Step 1

      A0 = A0 + h1
      Sheet6.Cells(9, 6) = A0
      Sheet6.Cells(9, 7) = h1
      T(1) = MaxP
      Sheet6.Cells(9, 2) = T(1)

      If (T(1) < T(0)) Then
         T(0) = T(1)                'T(0) = best possible value till now
         h1 = h1 * 1.25             'Expansion factor
      Else
         A0 = A0 - h1
         h1 = -h1 * (0.75 / 1.25)   'Contraction factor
```

```
        Sheet6.Cells(9, 6) = A0
        Sheet6.Cells(9, 7) = h1
      End If

      pKa = pKa + h2
      Sheet6.Cells(8, 6) = pKa
      Sheet6.Cells(8, 7) = h2
      T(1) = MaxP
      Sheet6.Cells(9, 2) = T(1)

      If (T(1) < T(0)) Then
        T(0) = T(1)
        h2 = h2 * 1.25                  'Expansion factor
      Else
        pKa = pKa - h2
        h2 = -h2 * (0.75 / 1.25)        'Contraction factor
        Sheet6.Cells(8, 6) = pKa
        Sheet6.Cells(8, 7) = h2
        T(1) = MaxP
        Sheet6.Cells(9, 2) = T(1)
      End If

      Sheet6.Cells(8, 2) = p
      If Abs(h1) < (10 ^ -10) And Abs(h2) < (10 ^ -10) And Abs(T(2) - T(0)) < (10 ^ -10) Then Exit For
      T(2) = T(0)

    Next p

    Sheet6.Cells(9, 7) = Abs(h1)
    Sheet6.Cells(8, 7) = Abs(h2)

    Sheet6.Cells(3 + w, 12) = A0
    Sheet6.Cells(3 + w, 13) = pKa
    Sheet6.Cells(3 + w, 14) = T(0)
    Sheet6.Cells(3 + w, 15) = p

    If T(0) < best Then                 ' Best of N check
      best = T(0)
      g = w + 3
    End If

Next w

A0 = Sheet6.Cells(g, 12)
pKa = Sheet6.Cells(g, 13)
p = Sheet6.Cells(g, 15)
T(0) = MaxP

Sheet6.Cells(z + 24, 1) = z
Sheet6.Cells(z + 24, 2) = A0
Sheet6.Cells(z + 24, 3) = pKa
Sheet6.Cells(z + 24, 4) = T(0)
Sheet6.Cells(z + 24, 5) = p

End Sub
' ----------------------------------------------------------------------------------------------------------------------
```

```
'-----------------------------------------------------Objective function calculation-----------------------------------

Function MaxP() As Double

Dim Vi As Double                    ' Subscript i refers to experimental terms
Dim pHi As Double                   ' Subscript r refers to regressed terms
Dim Vr As Double
Dim pHr As Double

Dim Va As Double                    ' Va, Vb: lower and higher limits of the golden section line search
Dim Vb As Double
Dim Vl As Double                    ' Vl, Vh: intermediate lower and higher limits of golden section search
Dim Vh As Double
Dim pH As Double
Dim l As Double                     ' l = Vb - Va

Dim q As Integer
Dim sum As Double
Dim d(1 To 2) As Double

Dim sigx As Double                  ' std dev in the experimental volume
Dim sigy(1 To 8) As Double          ' std dev in the experimental pH

'-----------------------------------------------------Golden Section Search-------------------------------------------

sigx = Sheet6.Cells(12, 7).Value / (Vmax - Vmin)

sum = 0
For q = 1 To 8 Step 1

   sigy(q) = Sheet6.Cells(11 + q, 8).Value / (pHmax - pHmin)

   Vi = Sheet6.Cells(q + 11, 2).Value
   pHi = Sheet6.Cells(q + 11, 3).Value

   Va = Vmin * 0.9                  ' Initializing the golden section boundary limits
   Vb = Vmax * 1.1

   Do
      l = Vb - Va
      Vl = Va + (0.382 * l)
      Vh = Vb - (0.382 * l)

      pH = pHpredict(Vl)
      d(1) = dist(Vi, pHi, Vl, pH, sigx, sigy(q))

      pH = pHpredict(Vh)
      d(2) = dist(Vi, pHi, Vh, pH, sigx, sigy(q))

      If d(2) < d(1) Then
         Va = Vl
         Vr = Vh
      Else
         Vb = Vh
         Vr = Vl
      End If
```

```
    Loop Until Abs(d(2) - d(1)) < 0.00001 And l < 0.00001        ' Stopping criteria for Golden section search

    Vr = (Vl + Vh) / 2
    pHr = pHpredict(Vr)
    d(2) = dist(Vi, pHi, Vr, pHr, sigx, sigy(q))

    Sheet6.Cells(q + 11, 4) = Vr
    Sheet6.Cells(q + 11, 5) = pHr
    Sheet6.Cells(q + 11, 6) = d(2)

    sum = sum + d(2)

Next q

Sheet6.Cells(20, 6) = sum
MaxP = sum

'Worksheets("Max_Likelihood").Calculate

End Function
'-----------------------------------------------------------------------------------------------------------------------------------
```

APPENDIX B – PACKED BED SIMULATOR CODE

The following Visual Basic code was adopted for the packed bed simulation.

Code:

```
Option Explicit
' Author :  Chetan Chandak
' Created date: 02- Feb-2009
' Description        : Packed bed model to test regression methods
'_____________________________________________________________________________

' _____________________________________Declaring Common Variables_____________________________

Global Ci As Double             ' Experimental variables
Global C0 As Double
Global F As Double
Global T As Double

Global Cir As Double             ' Regression or model variables
Global C0r As Double
Global Fr As Double
Global Tr As Double

Global k0 As Double             ' Model parameters
Global k1 As Double
Global E As Double
Global R As Double              ' Gas constant
Global V As Double              ' volume of the reactor

Global Cimin As Double          ' min and max input and output experimental values
Global Cimax As Double
Global C0min As Double
Global C0max As Double
Global Tmin As Double
Global Tmax As Double
Global Fmin As Double
Global Fmax As Double

Global z As Integer             ' # realizations, variable
Global l As Integer             ' # BON realizations, variable
Global chk As Integer           ' checks if the conc predicted by Newtons / Succesive substitution method
                                '  goes negative
Global i As Integer             ' common variables
Global j As Integer
Global q As Integer

Global rand As Double
Global sig(1 To 4) As Double

Global G(0 To 2) As Double      ' T(0): least possible objective function value at the time of reading
                                ' T(1): current objective function value at the time of reading
                                ' T(2): objective function value before a new set of changes in the parameters
Global d(0 To 2) As Double
```

```
'_____
' _____Experimental Data Generation Module_____
Sub Regression()

Dim N As Integer                    ' # realizations, value
Dim p As Integer                    ' # realizations for BON, value

N = Sheet3.Cells(2, 12).Value
p = Sheet3.Cells(3, 12).Value

R = 8.314
V = 1000

Cimin = Sheet2.Cells(36, 2).Value
Cimax = Sheet2.Cells(37, 2).Value
C0min = Sheet2.Cells(36, 5).Value
C0max = Sheet2.Cells(37, 5).Value
Tmin = Sheet2.Cells(36, 4).Value
Tmax = Sheet2.Cells(37, 4).Value
Fmin = Sheet2.Cells(36, 3).Value
Fmax = Sheet2.Cells(37, 3).Value

For z = 1 To N

   k0 = Sheet3.Cells(3, 6).Value
   k1 = Sheet3.Cells(2, 6).Value
   E = Sheet3.Cells(4, 6).Value
'----------Data Generation----------

   For i = 1 To 4 Step 1
      sig(i) = Sheet3.Cells(i + 1, 2).Value
   Next i

   Randomize

   For i = 1 To 27 Step 1
l1:    Ci = Sheet3.Cells(i + 10, 2).Value + sig(1) * Sqr(-2 * Log(Rnd())) * Sin(2 * 3.14159 * Rnd())
      F = Sheet3.Cells(i + 10, 3).Value + sig(2) * Sqr(-2 * Log(Rnd())) * Sin(2 * 3.14159 * Rnd())
      T = Sheet3.Cells(i + 10, 4).Value + sig(3) * Sqr(-2 * Log(Rnd())) * Sin(2 * 3.14159 * Rnd())

      If Ci < 0 Then GoTo l1

      C0 = Newton(Ci, F, T)
      If chk = 1 Then GoTo l1

      Sheet3.Cells(i + 10, 5) = C0 + sig(4) * Sqr(-2 * Log(Rnd())) * Sin(2 * 3.14159 * Rnd())
      If Sheet3.Cells(i + 10, 5) < 0 Then GoTo l1

      Sheet3.Cells(i + 10, 7) = Ci          ' Printing actual values
      Sheet3.Cells(i + 10, 8) = F
      Sheet3.Cells(i + 10, 9) = T
      Sheet3.Cells(i + 10, 10) = C0

   Next i
'----------
```

```
   Call data_substitution                    ' prints the experimental data in the regression method sheets

   For l = 1 To p
      rand = (1 + (2 * (Rnd() - 0.5) * 0.2))

      Sheet1.Activate                        ' runs the vertical distance module
      Call Regression_vertical

      Sheet2.Activate                        ' runs the normal distance module
      Call Regression_normal

      Sheet5.Activate                        ' runs the maximum likelihood module
      Call Regression_max
   Next l

Next z

End Sub
'-------------------------------------------------Newton Raphson method-------------------------------------------------

Function Newton(Ci As Double, F As Double, T As Double) As Double
Dim C(1 To 2) As Double
Dim i As Integer

C(1) = Ci * 0.9

For i = 1 To 100 Step 1

   C(2) = C(1) - (func(Ci, F, T, C(1)) / fder(Ci, F, T, C(1)))
   If C(2) < 0 Then C(2) = Ci * (1 + Rnd())
   If Abs(C(2) - C(1)) < 0.00000001 Then Exit For
   C(1) = C(2)

Next i

If i > 100 Then
   chk = 1
Else
   chk = 0
End If

Newton = (C(2) + C(1)) / 2

End Function
'-----------------------------------------------------------------------------------------------------------------------

Function func(Ci As Double, F As Double, T As Double, C As Double) As Double

   If C < 0 Then GoTo 2
   func = k1 * (Ci - C) + Log(Ci / C) - (V * k0 / F) * Exp(-E / (R * T))
2
End Function

'-----------------------------------------------------------------------------------------------------------------------
```

```
'-----------------------------------------------------------------------------------------------------------------------

Function fder(Ci As Double, F As Double, T As Double, C As Double) As Double

    fder = (func(Ci, F, T, C + 0.001) - func(Ci, F, T, C - 0.001)) / 0.002

End Function
'-----------------------------------------------------------------------------------------------------------------------

Sub data_substitution()
'
' data_substitution Macro
'
'
    Sheets("Sheet3").Select
    Range("A11:E37").Select
    Selection.Copy
    Sheets("Sheet1").Select
    Range("A8").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Sheets("Sheet3").Select
    Range("A11:E37").Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Sheet2").Select
    Range("A8").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Sheets("Sheet3").Select
    Range("A11:E37").Select
    Application.CutCopyMode = False
    Selection.Copy
    Sheets("Sheet5").Select
    Range("A8").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
End Sub
'_____________________________________________________________________________________________________

' _____________________________________Vertical distance Module_____________________________________

'-----------------------------------------------------Parameter Optimization-----------------------------------------

Sub Regression_vertical()

Dim a As Double
Dim b As Double
Dim C As Double
'-----------------------------------------------------------R^3 Cyclic Direct Search-----------------------------------

k0 = Sheet3.Cells(3, 6).Value * rand             ' Initializing
k1 = Sheet3.Cells(2, 6).Value * rand
E = Sheet3.Cells(4, 6).Value * rand

Sheet1.Cells(3, 9) = k0
Sheet1.Cells(4, 9) = k1
Sheet1.Cells(5, 9) = E
```

```
G(0) = Obj_vertical() + constraint()
Sheet1.Cells(4, 14) = G(0)

G(2) = G(0)                              ' storing the initial objective fn value in G(2)

a = k0 * 0.5
b = k1 * 0.5
C = E * 0.5

For i = 1 To 5000 Step 1

   k0 = k0 + a
   Sheet1.Cells(3, 9) = k0
   Sheet1.Cells(3, 10) = a
   G(1) = Obj_vertical() + constraint()
   Sheet1.Cells(4, 14) = G(1)

If (G(1) < G(0)) Then
   G(0) = G(1)                           ' G(0) = best possible value till now
   a = a * 1.25                          ' Expansion factor
Else
   k0 = k0 - a
   a = -a * (0.75 / 1.25)                ' Contraction factor
   Sheet1.Cells(3, 9) = k0
   Sheet1.Cells(3, 10) = a
End If

   k1 = k1 + b
   Sheet1.Cells(4, 9) = k1
   Sheet1.Cells(4, 10) = b
   G(1) = Obj_vertical() + constraint()
   Sheet1.Cells(4, 14) = G(1)

If (G(1) < G(0)) Then
   G(0) = G(1)
   b = b * 1.25                          ' Expansion factor
Else
   k1 = k1 - b
   b = -b * (0.75 / 1.25)                ' Contraction factor
   Sheet1.Cells(4, 9) = k1
   Sheet1.Cells(4, 10) = b
   G(1) = Obj_vertical() + constraint()
   Sheet1.Cells(4, 14) = G(1)
End If

E = E + C
   Sheet1.Cells(5, 9) = E
   Sheet1.Cells(5, 10) = C
   G(1) = Obj_vertical() + constraint()
   Sheet1.Cells(4, 14) = G(1)

If (G(1) < G(0)) Then
   G(0) = G(1)
   C = C * 1.25                          ' Expansion factor
Else
```

```
   E = E - C
   C = -C * (0.75 / 1.25)                    ' Contraction factor
   Sheet1.Cells(5, 9) = E
   Sheet1.Cells(5, 10) = C
   G(1) = Obj_vertical() + constraint()
   Sheet1.Cells(4, 14) = G(1)
End If

Sheet1.Cells(5, 14) = i

If Abs((G(2) - G(0)) / G(2)) < 10 ^ -6 And Abs(C / E) < 10 ^ -6 _
          And Abs(a / k0) < 10 ^ -6 And Abs(b / k1) < 10 ^ -6 Then      ' Stopping criteria
   Exit For
Else
   G(2) = G(0)
End If

Next i

Sheet1.Cells(3, 10) = Abs(a)                 ' reports the magnitude of the last change in parameter values
Sheet1.Cells(4, 10) = Abs(b)
Sheet1.Cells(5, 10) = Abs(C)
Sheet1.Cells(4, 14) = G(0)                   ' final value of the objective function

'Sheet4.Cells(l + 4, 11) = k0
'Sheet4.Cells(l + 4, 12) = Abs(a)
'Sheet4.Cells(l + 4, 13) = k1
'Sheet4.Cells(l + 4, 14) = Abs(b)
'Sheet4.Cells(l + 4, 15) = E
'Sheet4.Cells(l + 4, 16) = Abs(C)
'Sheet4.Cells(l + 4, 17) = G(0)
'Sheet4.Cells(l + 4, 18) = i

Sheet1.Cells(z + 40, 1) = z
Sheet1.Cells(z + 40, 2) = k0
Sheet1.Cells(z + 40, 3) = k1
Sheet1.Cells(z + 40, 4) = E
Sheet1.Cells(z + 40, 5) = i
Sheet1.Cells(z + 40, 7) = Abs(a)
Sheet1.Cells(z + 40, 8) = Abs(b)
Sheet1.Cells(z + 40, 9) = Abs(C)

End Sub
'-----------------------------------------------------Objective Function Evaluation-----------------------------------------

Function Obj_vertical() As Double

Dim sum As Double

For q = 1 To 27 Step 1
   Ci = Sheet1.Cells(7 + q, 2).Value
   F = Sheet1.Cells(7 + q, 3).Value
   T = Sheet1.Cells(7 + q, 4).Value
   C0 = Sheet1.Cells(7 + q, 5).Value

   C0r = Newton(Ci, F, T)
```

```
    d(0) = (C0 - C0r) ^ 2

    Sheet1.Cells(7 + q, 10).Value = C0r
    Sheet1.Cells(7 + q, 11).Value = d(0)

    sum = sum + d(0)

Next q

    Obj_vertical = sum                          ' Objective function value
    Sheet1.Cells(35, 11) = sum

End Function
'-----------------------------------------------------------Constraint check-----------------------------------------------------

Function constraint() As Double
Dim del(1 To 2) As Double

If k1 < 5 Then del(1) = Abs((k1 - 5) / k1)
If del(1) > del(2) Then del(2) = del(1)

If k1 > 100 Then del(1) = Abs((k1 - 100) / k1)
If del(1) > del(2) Then del(2) = del(1)

If k0 < 0.0001 Then del(1) = Abs((k0 - 0.0001) / k0)
If del(1) > del(2) Then del(2) = del(1)

If k0 > 1 Then del(1) = Abs((k0 - 1) / k0)
If del(1) > del(2) Then del(2) = del(1)

If E < 5000 Then del(1) = Abs((E - 5000) / E)
If del(1) > del(2) Then del(2) = del(1)

If E > 40000 Then del(1) = Abs((E - 40000) / E)
If del(1) > del(2) Then del(2) = del(1)

constraint = del(2)
End Function
'_____________________________________________________________________________________________________________________
' _____________________________________Normal distance Module_____________________________________
'-----------------------------------------------------Parameter Optimization-----------------------------------------------

Sub Regression_normal()
Dim a As Double
Dim b As Double
Dim C As Double
'-----------------------------------------------------R^3 Cyclic Direct Search-----------------------------------------------

k0 = Sheet3.Cells(3, 6).Value * rand         ' Initializing
k1 = Sheet3.Cells(2, 6).Value * rand
E = Sheet3.Cells(4, 6).Value * rand

Sheet2.Cells(3, 9) = k0
Sheet2.Cells(4, 9) = k1
Sheet2.Cells(5, 9) = E
```

```
G(0) = Obj_normal() + constraint()
Sheet2.Cells(4, 14) = G(0)

G(2) = G(0)                          ' storing the initial objective fn value in G(2)

a = k0 * 0.5
b = k1 * 0.5
C = E * 0.5

For i = 1 To 5000 Step 1

   k0 = k0 + a
   Sheet2.Cells(3, 9) = k0
   Sheet2.Cells(3, 10) = a
   G(1) = Obj_normal() + constraint()
   Sheet2.Cells(4, 14) = G(1)

If (G(1) < G(0)) Then
   G(0) = G(1)                       ' G(0) = best possible value till now
   a = a * 1.25                       ' Expansion factor
Else
   k0 = k0 - a
   a = -a * (0.75 / 1.25)             ' Contraction factor
   Sheet2.Cells(3, 9) = k0
   Sheet2.Cells(3, 10) = a
End If

   k1 = k1 + b
   Sheet2.Cells(4, 9) = k1
   Sheet2.Cells(4, 10) = b
   G(1) = Obj_normal() + constraint()
   Sheet2.Cells(4, 14) = G(1)

If (G(1) < G(0)) Then
   G(0) = G(1)
   b = b * 1.25                       ' Expansion factor
Else
   k1 = k1 - b
   b = -b * (0.75 / 1.25)             ' Contraction factor
   Sheet2.Cells(4, 9) = k1
   Sheet2.Cells(4, 10) = b
   G(1) = Obj_normal() + constraint()
   Sheet2.Cells(4, 14) = G(1)
End If

E = E + C
   Sheet2.Cells(5, 9) = E
   Sheet2.Cells(5, 10) = C
   G(1) = Obj_normal() + constraint()
   Sheet2.Cells(4, 14) = G(1)

If (G(1) < G(0)) Then
   G(0) = G(1)
   C = C * 1.25                       ' Expansion factor
Else
```

```
   E = E - C
   C = -C * (0.75 / 1.25)                          ' Contraction factor
   Sheet2.Cells(5, 9) = E
   Sheet2.Cells(5, 10) = C
   G(1) = Obj_normal() + constraint()
   Sheet2.Cells(4, 14) = G(1)
End If

Sheet2.Cells(5, 14) = i

If Abs((G(2) - G(0)) / G(2)) < 10 ^ -6 And Abs(C / E) < 10 ^ -6 _
            And Abs(a / k0) < 10 ^ -6 And Abs(b / k1) < 10 ^ -6 Then      ' Stopping criteria
   Exit For
Else
   G(2) = G(0)
End If

Next i

Sheet2.Cells(3, 10) = Abs(a)                    ' reports the magnitude of the last change in parameter values

Sheet2.Cells(4, 10) = Abs(b)
Sheet2.Cells(5, 10) = Abs(C)
Sheet2.Cells(4, 14) = G(0)                      ' Final Objective function value

'Sheet4.Cells(l + 4, 2) = k0
'Sheet4.Cells(l + 4, 3) = Abs(a)
'Sheet4.Cells(l + 4, 4) = k1
'Sheet4.Cells(l + 4, 5) = Abs(b)
'Sheet4.Cells(l + 4, 6) = E
'Sheet4.Cells(l + 4, 7) = Abs(C)
'Sheet4.Cells(l + 4, 8) = G(0)
'Sheet4.Cells(l + 4, 9) = i

Sheet2.Cells(z + 40, 1) = z
Sheet2.Cells(z + 40, 2) = k0
Sheet2.Cells(z + 40, 3) = k1
Sheet2.Cells(z + 40, 4) = E
Sheet2.Cells(z + 40, 5) = i
Sheet2.Cells(z + 40, 7) = Abs(a)
Sheet2.Cells(z + 40, 8) = Abs(b)
Sheet2.Cells(z + 40, 9) = Abs(C)

End Sub
'-----------------------------------------------------Objective Function Evaluation-----------------------------------------

Function Obj_normal() As Double

Dim a1 As Double
Dim b1 As Double
Dim C1 As Double
Dim sum As Double
'-----------------------------------------------------R^3 Cyclic Direct Search-----------------------------------------

For q = 1 To 27 Step 1
   Ci = Sheet2.Cells(7 + q, 2).Value
```

```
    F = Sheet2.Cells(7 + q, 3).Value
    T = Sheet2.Cells(7 + q, 4).Value
    C0 = Sheet2.Cells(7 + q, 5).Value

    Cir = Cimin * 0.9                       '  Initializing
    Tr = Tmin * 0.9
    Fr = Fmin * 0.9

    a1 = Cir * 0.1
    b1 = Tr * 0.1
    C1 = Fr * 0.1

    C0r = Newton(Cir, Fr, Tr)
    d(0) = dist(Ci, F, T, C0, Cir, Fr, Tr, C0r, 1, 1, 1, 1) + chk * 1000

    d(2) = d(0)

    For j = 1 To 2000 Step 1

       Cir = Cir + a1
       If Cir < 0 Then GoTo 5

       C0r = Newton(Cir, Fr, Tr)
       d(1) = dist(Ci, F, T, C0, Cir, Fr, Tr, C0r, 1, 1, 1, 1) + chk * 1000

       If d(1) < d(0) Then
          d(0) = d(1)
          a1 = a1 * 1.25                    ' Expansion factor
       Else
5         Cir = Cir - a1
          a1 = -a1 * 0.75 / 1.25            ' Contraction factor
       End If

       Tr = Tr + b1
       If Tr < 0 Then GoTo 6

       C0r = Newton(Cir, Fr, Tr)
       d(1) = dist(Ci, F, T, C0, Cir, Fr, Tr, C0r, 1, 1, 1, 1) + chk * 1000

       If d(1) < d(0) Then
          d(0) = d(1)
          b1 = b1 * 1.25                    ' Expansion factor
       Else
6         Tr = Tr - b1
          b1 = -b1 * 0.75 / 1.25            ' Contraction factor
       End If

       Fr = Fr + C1
       If Fr < 0 Then GoTo 7

       C0r = Newton(Cir, Fr, Tr)
       d(1) = dist(Ci, F, T, C0, Cir, Fr, Tr, C0r, 1, 1, 1, 1) + chk * 1000

       If d(1) < d(0) Then
          d(0) = d(1)
          C1 = C1 * 1.25                    ' Expansion factor
```

```
        Else
7           Fr = Fr - C1
            C1 = -C1 * 0.75 / 1.25                  ' Contraction factor
        End If

        If Abs((d(2) - d(0)) / d(2)) < 10 ^ -10 And Abs(C1 / Fr) < 10 ^ -10 And Abs(a1 / Cir) < 10 ^ -10 And
        Abs(b1 / Tr) < 10 ^ -10 Then                ' Stopping criteria
            Exit For
        Else
            d(2) = d(0)
        End If

    Next j

    Sheet2.Cells(7 + q, 7).Value = Cir
    Sheet2.Cells(7 + q, 8).Value = Fr
    Sheet2.Cells(7 + q, 9).Value = Tr
    Sheet2.Cells(7 + q, 10).Value = C0r
    Sheet2.Cells(7 + q, 11).Value = d(0)

    sum = sum + d(0)

Next q

    Obj_normal = sum                            ' Objective function value
    Sheet2.Cells(35, 11) = sum

End Function
'-------------------------------------------------------------------------------------------------------------------------

Function dist(Ci As Double, F As Double, T As Double, C0 As Double, Cir As Double, Fr As Double, Tr
As Double, C0r As Double, sig1 As Double, sig2 As Double, sig3 As Double, sig4 As Double) As Double


dist = ((Ci - Cir) / ((Cimax - Cimin) * sig1)) ^ 2 + ((F - Fr) / ((Fmax - Fmin) * sig2)) ^ 2 + ((T - Tr) /
((Tmax - Tmin) * sig3)) ^ 2 + ((C0 - C0r) / ((C0max - C0min) * sig4)) ^ 2


End Function
'_____________________________________________________________________________________________
' _____________________________________Maximum Likelihood Module_____________________________
'-----------------------------------------------------Parameter Optimization-----------------------------------------

Sub Regression_max ()
Dim a As Double
Dim b As Double
Dim C As Double
'-----------------------------------------------------R3 Cyclic Direct Search-----------------------------------------

k0 = Sheet3.Cells(3, 6).Value * rand         ' Initializing
k1 = Sheet3.Cells(2, 6).Value * rand
E = Sheet3.Cells(4, 6).Value * rand

Sheet5.Cells(3, 9) = k0
```

```
Sheet5.Cells(4, 9) = k1
Sheet5.Cells(5, 9) = E

G(0) = ObjML() + constraint()
Sheet5.Cells(4, 14) = G(0)

G(2) = G(0)                          ' storing the initial objective fn value in G(2)

a = k0 * 0.5
b = k1 * 0.5
C = E * 0.5

For i = 1 To 5000 Step 1

   k0 = k0 + a
   Sheet5.Cells(3, 9) = k0
   Sheet5.Cells(3, 10) = a
   G(1) = ObjML() + constraint()
   Sheet5.Cells(4, 14) = G(1)

If (G(1) < G(0)) Then
   G(0) = G(1)                       ' G(0) = best possible value till now
   a = a * 1.25                       ' Expansion factor
Else
   k0 = k0 - a
   a = -a * (0.75 / 1.25)             ' Contraction factor
   Sheet5.Cells(3, 9) = k0
   Sheet5.Cells(3, 10) = a
End If

   k1 = k1 + b
   Sheet5.Cells(4, 9) = k1
   Sheet5.Cells(4, 10) = b
   G(1) = ObjML() + constraint()
   Sheet5.Cells(4, 14) = G(1)

If (G(1) < G(0)) Then
   G(0) = G(1)
   b = b * 1.25                       ' Expansion factor
Else
   k1 = k1 - b
   b = -b * (0.75 / 1.25)             ' Contraction factor
   Sheet5.Cells(4, 9) = k1
   Sheet5.Cells(4, 10) = b
   G(1) = ObjML() + constraint()
   Sheet5.Cells(4, 14) = G(1)
End If

E = E + C
   Sheet5.Cells(5, 9) = E
   Sheet5.Cells(5, 10) = C
   G(1) = ObjML() + constraint()
   Sheet5.Cells(4, 14) = G(1)

If (G(1) < G(0)) Then
   G(0) = G(1)
```

```
   C = C * 1.25                                ' Expansion factor
Else
   E = E - C
   C = -C * (0.75 / 1.25)                      ' Contraction factor
   Sheet5.Cells(5, 9) = E
   Sheet5.Cells(5, 10) = C
   G(1) = ObjML() + constraint()
   Sheet5.Cells(4, 14) = G(1)
End If

Sheet5.Cells(5, 14) = i

If Abs((G(2) - G(0)) / G(2)) < 10 ^ -6 And Abs(C / E) < 10 ^ -6 _
            And Abs(a / k0) < 10 ^ -6 And Abs(b / k1) < 10 ^ -6 Then      ' Stopping criteria
   Exit For
Else
   G(2) = G(0)
End If

Next i

Sheet5.Cells(3, 10) = Abs(a)                   ' reports the magnitude of the last change in parameter values
Sheet5.Cells(4, 10) = Abs(b)
Sheet5.Cells(5, 10) = Abs(C)
Sheet5.Cells(4, 14) = G(0)                     ' Final Objective function value

'Sheet4.Cells(l + 4, 20) = k0
'Sheet4.Cells(l + 4, 21) = Abs(a)
'Sheet4.Cells(l + 4, 22) = k1
'Sheet4.Cells(l + 4, 23) = Abs(b)
'Sheet4.Cells(l + 4, 24) = E
'Sheet4.Cells(l + 4, 25) = Abs(C)
'Sheet4.Cells(l + 4, 26) = G(0)
'Sheet4.Cells(l + 4, 27) = i

Sheet5.Cells(z + 40, 1) = z
Sheet5.Cells(z + 40, 2) = k0
Sheet5.Cells(z + 40, 3) = k1
Sheet5.Cells(z + 40, 4) = E
Sheet5.Cells(z + 40, 5) = i
Sheet5.Cells(z + 40, 7) = Abs(a)
Sheet5.Cells(z + 40, 8) = Abs(b)
Sheet5.Cells(z + 40, 9) = Abs(C)

End Sub
'-------------------------------------------------Objective Function Evaluation-----------------------------------------

Function ObjML() As Double

Dim a1 As Double
Dim b1 As Double
Dim C1 As Double
Dim sigma(1 To 4) As Double
Dim sum As Double
```

```
sigma(1) = sig(1) / (Cimax - Cimin)
sigma(2) = sig(2) / (Fmax - Fmin)
sigma(3) = sig(3) / (Tmax - Tmin)
sigma(4) = sig(4) / (C0max - C0min)

'----------------------------------------------------R^3 Cyclic Direct Search----------------------------------------------

For q = 1 To 27 Step 1

   Ci = Sheet5.Cells(7 + q, 2).Value
   F = Sheet5.Cells(7 + q, 3).Value
   T = Sheet5.Cells(7 + q, 4).Value
   C0 = Sheet5.Cells(7 + q, 5).Value

   Cir = Cimin * 0.9                          '  Initializing
   Tr = Tmin * 0.9
   Fr = Fmin * 0.9

   a1 = Cir * 0.1
   b1 = Tr * 0.1
   C1 = Fr * 0.1

   C0r = Newton(Cir, Fr, Tr)
   d(0) = dist(Ci, F, T, C0, Cir, Fr, Tr, C0r, sigma(1), sigma(2), sigma(3), sigma(4)) + chk * 1000

   d(2) = d(0)

   For j = 1 To 2000 Step 1

      Cir = Cir + a1
      If Cir < 0 Then GoTo 1

      C0r = Newton(Cir, Fr, Tr)
      d(1) = dist(Ci, F, T, C0, Cir, Fr, Tr, C0r, sigma(1), sigma(2), sigma(3), sigma(4)) + chk * 1000

      If d(1) < d(0) Then
         d(0) = d(1)
         a1 = a1 * 1.25                       ' Expansion factor
      Else
1        Cir = Cir - a1
         a1 = -a1 * 0.75 / 1.25               ' Contraction factor
      End If

      Tr = Tr + b1
      If Tr < 0 Then GoTo 3

      C0r = Newton(Cir, Fr, Tr)
      d(1) = dist(Ci, F, T, C0, Cir, Fr, Tr, C0r, sigma(1), sigma(2), sigma(3), sigma(4)) + chk * 1000

      If d(1) < d(0) Then
         d(0) = d(1)
         b1 = b1 * 1.25                       ' Expansion factor
      Else
3        Tr = Tr - b1
         b1 = -b1 * 0.75 / 1.25               ' Contraction factor
      End If
```

```
        Fr = Fr + C1
        If Fr < 0 Then GoTo 4

        C0r = Newton(Cir, Fr, Tr)
        d(1) = dist(Ci, F, T, C0, Cir, Fr, Tr, C0r, sigma(1), sigma(2), sigma(3), sigma(4)) + chk * 1000

        If d(1) < d(0) Then
           d(0) = d(1)
           C1 = C1 * 1.25                    ' Expansion factor
        Else
4          Fr = Fr - C1
           C1 = -C1 * 0.75 / 1.25            ' Contraction factor
        End If

        If Abs((d(2) - d(0)) / d(2)) < 10 ^ -10 And Abs(C1 / Fr) < 10 ^ -10 And Abs(a1 / Cir) < 10 ^ -10 And
        Abs(b1 / Tr) < 10 ^ -10 Then          ' Stopping criteria
           Exit For
        Else
           d(2) = d(0)
        End If

     Next j

     Sheet5.Cells(7 + q, 7).Value = Cir
      Sheet5.Cells(7 + q, 8).Value = Fr
     Sheet5.Cells(7 + q, 9).Value = Tr
     Sheet5.Cells(7 + q, 10).Value = C0r
     Sheet5.Cells(7 + q, 11).Value = d(0)

     sum = sum + d(0)

Next q

     ObjML = sum                           ' Objective function value
     Sheet5.Cells(35, 11) = sum

End Function
'-------------------------------------------------------------------------------------------------------------------------
```

APPENDIX C – RESULTS: PARAMETER DISTRIBUTIONS AND PROBABILITY PLOTS

The following Visual Basic code was used to generate the parameter histograms and the probability plots

Code:

```
Option Explicit
' Author :  Chetan Chandak
' Created date: 20- Dec-2007
' Description        : Result analysis through parameter distribution and probability plots
'_____________________________________________________________________________________
'_____________________________________Parameter Distributions_____________________________
Public r As Integer                    ' number of realizations

Sub count()
Dim i As Integer
Dim j As Integer
Dim N As Integer                       ' number of intervals, taken as 50 here

Dim X As Double
Dim Xmax As Double
Dim Xmin As Double
Dim R1 As Double                       ' interval size
Dim P(0 To 200) As Double              ' for interval spacing
Dim C(1 To 200) As Integer             ' counts the number of values within an interval

Dim Y As Double
Dim Ymax As Double
Dim Ymin As Double
Dim R2 As Double                       ' interval size
Dim Q(0 To 200) As Double              ' for interval spacing
Dim D(1 To 200) As Integer             ' counts the number of values within an interval

Sheet1.Range(Cells(25, 6), Cells(125, 10)).ClearContents

r = Sheet1.Cells(1, 9).Value
N = Sheet1.Cells(2, 9).Value
Xmin = Sheet1.Cells(2, 4).Value
Xmax = Sheet1.Cells(3, 4).Value
Ymin = Sheet1.Cells(2, 5).Value
Ymax = Sheet1.Cells(3, 5).Value

' Since the regression methods would have diferent ranges depending upon each of its min and max values,
' a common min and max for A0 and pka is chosen to facilitate plotting the distribution on the same graph

R1 = (Xmax - Xmin) / N
R2 = (Ymax - Ymin) / N
```

```
For i = 0 To N Step 1
      P(i) = (i * R1) + Xmin
      Q(i) = (i * R2) + Ymin
Next i

For i = 1 To N Step 1
   C(i) = 0
   D(i) = 0
Next i

For i = 1 To r
   X = Sheet1.Cells(24 + i, 2).Value
   Y = Sheet1.Cells(24 + i, 3).Value

   If X = Xmax Then C(N) = C(N) + 1
   If Y = Ymax Then D(N) = D(N) + 1

   For j = 1 To N Step 1
      If (X > P(j - 1) Or X = P(j - 1)) And X < P(j) And X < Xmax Then
      C(j) = C(j) + 1
      End If

      If (Y > Q(j - 1) Or Y = Q(j - 1)) And Y < Q(j) And Y < Ymax Then
      D(j) = D(j) + 1
      End If
   Next j

Next i

For i = 1 To N
   Sheet1.Cells(24 + i, 6) = P(i - 1) & " - " & P(i)
   Sheet1.Cells(24 + i, 7) = C(i)
   Sheet1.Cells(24 + i, 9) = Q(i - 1) & " - " & Q(i)
   Sheet1.Cells(24 + i, 10) = D(i)
Next i

Call probability

End Sub
'_____________________________________________________________________________________

' _____________________________________Probability Plots_____________________________________

Sub probability()

Dim i As Integer
Dim j As Integer
Dim N As Integer                    ' number of intervals, taken as 100 here

Dim X As Double                     ' X = A0
Dim Y As Double                     ' Y = pKa

Dim Xact As Double                  ' true value of A0
Dim Yact As Double                  ' true value of pKa

Dim d1 As Double                    ' deviation of a particular A0 value from the true value
Dim d2 As Double                    ' deviation of a particular pKa value from the true value
```

```
Dim R1 As Double                ' interval or step size of d1
Dim R2 As Double                ' interval or step size of d2

Dim C1 As Double                ' count of # A0 values beyond a particular deviation
Dim C2 As Double                ' count of # pKa values beyond a particular deviation

Dim dmax1 As Double             ' max deviation of A0 value from the true value
Dim dmax2 As Double             ' max deviation of pKa value from the true value
Dim count As Integer            ' counts the total number of realizations

Sheet1.Range(Cells(25, 12), Cells(225, 15)).ClearContents

count = 0
r = Sheet1.Cells(1, 9).Value
N = Sheet1.Cells(3, 9).Value
Xact = Sheet1.Cells(3, 2).Value
Yact = Sheet1.Cells(4, 2).Value

For i = 1 To r Step 1

   X = Sheet1.Cells(24 + i, 2).Value
   Y = Sheet1.Cells(24 + i, 3).Value

   If X = 0 Or Y = 0 Then GoTo l1
   d1 = Abs(X - Xact)
   d2 = Abs(Y - Yact)

   If d1 > dmax1 Then dmax1 = d1
   If d2 > dmax2 Then dmax2 = d2
   count = count + 1

l1: Next i

R1 = dmax1 / N
R2 = dmax2 / N

j = 0
While (d1 <= dmax1) Or (d2 <= dmax2)

   d1 = j * R1
   d2 = j * R2
   C1 = 0
   C2 = 0

   For i = 1 To 1000 Step 1
      X = Sheet1.Cells(24 + i, 2).Value
      Y = Sheet1.Cells(24 + i, 3).Value

      If X = 0 Or Y = 0 Then GoTo l2
      If Abs(X - Xact) > d1 Then
         C1 = C1 + 1
      End If

      If Abs(Y - Yact) > d2 Then
         C2 = C2 + 1
```

```
        End If
l2: Next i

    Sheet1.Cells(25 + j, 12).Value = d1
    Sheet1.Cells(25 + j, 13).Value = C1 / count
    Sheet1.Cells(25 + j, 14).Value = d2
    Sheet1.Cells(25 + j, 15).Value = C2 / count

    j = j + 1

Wend

End Sub
'-------------------------------------------------------------------------------------------------------------------------------
```

VITA

Chetan Chandak

Candidate for the Degree of

Master of Science

Thesis: A UTILITARIAN COMPARISON OF NONLINEAR REGRESSION METHODS

Major Field: Chemical Engineering

Biographical:

Education:

- Completed the requirements for Master of Science in Chemical Engineering at Oklahoma State University, Stillwater, Oklahoma in May 09.
- Bachelor of Technology in Chemical Engineering, Jawaharlal Nehru Technological University, Hyderabad, India – May 07.

Experience:

- Research Assistant to Dr. Russell Rhinehart, OSU, Aug'07-May 09
- Project Intern under Dr. B. Satyavathi, Indian Institute of Chemical Technology, Hyderabad, India, – Dec'06-May 07

Professional Memberships:

- Member of the International Society of Automation
- Member of the Golden Key International Honor Society
- Member of the American Institute for Chemical Engineers' – OSU Chapter

Name: Chandak, Chetan Date of Degree: May 2009

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: A UTILITARIAN COMPARISON OF NONLINEAR REGRESSION METHODS

Pages in Study: 97 Candidate for the Degree of Master of Science

Major Field: Chemical Engineering

Findings and Conclusions:

To overcome the shortcomings of the least squares regression method, two methods – the normal distance, and the maximum likelihood, were developed. The maximum likelihood is a more generic method, with the normal distance being a consequence of it when error variances in the input and output measurements are equal. The methods were compared with the least squares method through Monte Carlo simulations for Titration and Packed Bed Reactor models. The methods were tested for varying magnitudes of uncertainty, for a sufficient number of realizations to ensure the results reflected the average parameter estimates, and were unique to the regression method.

The results for the maximum likelihood method were found to be at par with the best method in most cases. The vertical and the normal distance method had individual preferences depending upon the relative magnitudes of uncertainty. However the programming burden for the maximum likelihood and the normal distance method, apart from the estimate of uncertainty variances for the maximum likelihood method, were the drawbacks. But, approximate estimate of the variances for the maximum likelihood method also yielded good results, as tested for a few cases. Hence for a more accurate estimate of regression parameters, the maximum likelihood method could be adopted with a higher probability of getting the desired results as compared to the other two methods.

ADVISER'S APPROVAL: Dr. R. Russell Rhinehart