

Parameter Estimation in Different Enzyme Reactions

Sudarshan R. Nelatury¹, Charles F. Nelatury², Mary C. Vagula³

¹Department of ECE, School of Engineering, Penn State Erie, Erie, USA

²Department of ECE, College of Engineering, Drexel University, Philadelphia, USA

³Biology Department, Gannon University, Erie, USA

Email: vagula001@gannon.edu

Received 31 October 2013; revised 7 December 2013; accepted 25 December 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Enzyme kinetic parameters have been estimated using MATLAB software via the Wilkinson non-linear regression technique. The MATLAB script file written to implement this technique is short and very straightforward. Several software tools are commercially available for this purpose, with many graphical user interface (GUI) features. A routine use of these packages might offer immediate satisfaction of interactive hands-on experience; but in some cases the researcher might wish to write his/her own code and compare the results for further confirmation. Today MATLAB is in use in almost all the schools and laboratories as a standard software tool. So this paper is aimed at helping enzyme researchers to make use of this powerful software for estimation of parameters. It enables the incorporation of the analytical steps behind parameter estimation in an easy-to-follow manner and furnishes better visualization.

Keywords

Michaelis-Menten Equation; Dixon Plot; Hill Equation; Wilkinson Nonlinear Regression; MATLAB

1. Introduction

Over the past several years, different enzyme reactions governed by different equations have been identified in the field of biochemistry. Estimation of kinetic parameters of these enzyme reactions is one of the important problems [1]-[4]. The nonlinearity in the functional expression for the reaction rate, in terms of substrate concentration and other parameters, makes the parameter estimation problem difficult. Graphical techniques such as Lineweaver-Burk's method and the use of Dixon plot to estimate parameters [5]-[7] were in vogue for some time. These methods rely on linearizing the Michaelis-Menten equation in the case of simple enzyme reactions.

For complex enzyme reactions, such techniques do not furnish reliable results. This has prompted Wilkinson to propose non-linear regression for the estimation of enzyme parameters [8]. Using Wilkinson non-linear regression coupled with a modified Fibonacci search method, Atkins outlined a computer program for fitting the Hill equation directly to experimental data [9]. Following this, numerous software packages have been developed [10]-[11] to perform parameter estimation interactively. Most of these packages have fixed menus and do not allow the user to incorporate possible changes to the underlying algorithm.

The goal of this article is to explain the nonlinear regression technique briefly and show four simulation examples of enzyme reactions. The programming is done in MATLAB, a registered trade mark of MathWorks, Inc. MATLAB is a high-level language and interactive environment that enables one to perform computationally intensive tasks faster than with traditional programming languages. The syntax of various functions in MATLAB makes implementing difficult, mathematical steps very straightforward.

Today, use of MATLAB has almost become a *de facto* standard among engineers and computer scientists both in academia and industry. Faculty and students of biochemistry can easily learn this software using numerous tutorials available on-line or introductory books such as [18]-[21]. Further, it is easy for biologists to develop interaction with other engineers and scientists in the language of MATLAB. In this paper we shall discuss the basic principle behind nonlinear regression, and supplement it with simulation examples¹.

2. Parameter Estimation via Nonlinear Regression

One of the challenges faced in a biochemistry laboratory while studying enzyme kinetics is to decide comprehensive levels of the substrate concentration. For example, it is hard to determine the maximum concentration, minimum concentration, and to know how best to space the intermediate levels and so on. In practice, one may have to plot the data and go through a few iterations before endorsing the data for further processing. After satisfactory acquisition of the data, which invariably carries measurement inaccuracies, enzyme parameters appearing in the governing model can be estimated using the non-linear regression technique.

Let us consider a functional relationship that expresses the reaction rate v in terms of the independent variable(s) and some parameters. For example, in the case of a simple enzyme-substrate uninhibited reaction it is given by the Michaelis-Menten equation:

$$v = \left(\frac{V_{\max} s}{s + K_m} \right). \quad (1)$$

In Equation (1) above, the substrate concentration s is the independent variable, and V_{\max} and K_m are enzyme parameters representing the maximum reaction velocity and the Michaelis constant respectively. Numerically, K_m equals s when $v = 0.5V_{\max}$. In general, there may be more independent variables than s alone, which for convenience, could be put in a vector of N elements say, $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_N]^T$. Let the vector $\boldsymbol{\beta} = [\beta_1 \ \beta_2 \ \dots \ \beta_p]^T$ denote a set of P parameters. Here the superscript T denotes the transpose. This generalization would become clearer after three more examples that are discussed in this paper. The dependence of v on \mathbf{x} and $\boldsymbol{\beta}$ is denoted by $v(\mathbf{x}_m; \boldsymbol{\beta})$. Let y_m stand for the m^{th} data sample. The subscript m in all occurrences except in the Michaelis constant K_m indicates the m^{th} observation. Since measurements are subject to experimental errors, we shall write:

$$y_m = v(\mathbf{x}_m; \boldsymbol{\beta}) + \varepsilon_m \quad (2)$$

where the measurement error, $\varepsilon_m \sim \mathcal{N}(0, \sigma_m^2)$, is assumed to be a normally distributed random variable with a mean zero and a variance of σ_m^2 . Given a set of measured values of Equation (2), the parameter estimation problem involves finding an estimate of $\boldsymbol{\beta}$ that reduces a measure of error, usually the sum of weighted (with the weights w_m) residual squares $WRSS$, denoted by $J(\boldsymbol{\beta})$:

$$WRSS = J(\boldsymbol{\beta}) = \sum_m w_m \{y_m - v(\mathbf{x}_m; \boldsymbol{\beta})\}^2. \quad (3)$$

Our objective is to minimize $J(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$. It is hard to find the minimum of $J(\boldsymbol{\beta})$ by explicitly differentiating and solving for global minimum. Iterative methods are usually employed. Nonlinear regression is a convenient technique [8] [22] that allows us to estimate the unknown parameters by linearizing Equation (2) at

¹The MATLAB script for one of the examples is shown in Appendix I.

each iteration starting from an initial guess $\hat{\beta}_0$. This is done by first expressing $v(\mathbf{x}_m; \boldsymbol{\beta})$ with the first two terms in its Taylor's series expansion around an estimate, initially taken as $\hat{\beta}_0$. Then the m^{th} element in Equation (2) can be written as:

$$y_m \approx v(\mathbf{x}_m; \hat{\beta}_0) + \left[\frac{\partial v(\mathbf{x}_m; \boldsymbol{\beta})}{\partial \beta_1} \quad \frac{\partial v(\mathbf{x}_m; \boldsymbol{\beta})}{\partial \beta_2} \quad \dots \quad \frac{\partial v(\mathbf{x}_m; \boldsymbol{\beta})}{\partial \beta_p} \right]_{\boldsymbol{\beta}=\hat{\beta}_0} \begin{bmatrix} \hat{\beta}_{11} - \hat{\beta}_{10} \\ \hat{\beta}_{21} - \hat{\beta}_{20} \\ \dots \\ \hat{\beta}_{p1} - \hat{\beta}_{p0} \end{bmatrix} + \varepsilon_m \quad (4)$$

here $\hat{\beta}_1 - \hat{\beta}_0 = \Delta \hat{\beta}_0 = \begin{bmatrix} \hat{\beta}_{11} - \hat{\beta}_{10} \\ \hat{\beta}_{21} - \hat{\beta}_{20} \\ \dots \\ \hat{\beta}_{p1} - \hat{\beta}_{p0} \end{bmatrix}$ is assumed to be a small increment in the parameter vector. This can be done

for all the observations $m = 1$ to M .

Now treating the vector $[y_m - v(\mathbf{x}_m; \hat{\beta}_0)]$ as the residual \mathbf{r}_0 at 0^{th} iteration, we can write:

$$\mathbf{r}_0 \approx \left[\left[\frac{\partial v(\mathbf{x}_m; \boldsymbol{\beta})}{\partial \beta_1} \right] \quad \left[\frac{\partial v(\mathbf{x}_m; \boldsymbol{\beta})}{\partial \beta_2} \right] \quad \dots \quad \left[\frac{\partial v(\mathbf{x}_m; \boldsymbol{\beta})}{\partial \beta_p} \right] \right]_{\boldsymbol{\beta}=\hat{\beta}_0} (\hat{\beta}_1 - \hat{\beta}_0) + [\varepsilon_m] \quad (5)$$

This process can be iterated by replacing 0 and 1 with k and $k + 1$ respectively. Denoting the matrix involving the derivatives in Equation (5) at k^{th} iteration by \mathbf{G}_k we get,

$$\mathbf{r}_k \approx \mathbf{G}_k (\hat{\beta}_{k+1} - \hat{\beta}_k) + [\varepsilon_m]. \quad (6)$$

The weighted least squares solution for this can be written in terms of generalized inverse of \mathbf{G}_k [23] [24] as:

$$\hat{\beta}_{k+1} = \hat{\beta}_k + (\mathbf{G}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{G}_k)^{-1} \mathbf{G}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{r}_k \quad (7)$$

where

$$\boldsymbol{\Sigma} = (\text{cov}[\varepsilon_m]) \quad (8)$$

is the covariance matrix of $[\varepsilon_m]$.

Equation (7) helps in estimating the parameters of a given model recursively; but is in a generalized form. For the enzyme researchers or students to be able to code this in MATLAB, they need to understand each symbol in (7) and how to express the same for a given model. In the next section four types of enzyme models are considered and expressions for various elements of $\boldsymbol{\beta}$, \mathbf{G} , and \mathbf{r} are given as they relate to each model. In Appendix I, MATLAB code is provided for Example III. For other examples, the same code could be edited with minimum effort by replacing the expressions of these vectors/matrices appropriately and could be executed. With this idea in mind, in the next section, Equation (7) is specialized for four different models.

3. Four Models

3.1. Model I: A Simple Uninhibited Enzyme Substrate Reaction

Let us first consider the case of a simple uninhibited enzyme substrate reaction. As mentioned earlier in the paper, s denotes the concentration of the substrate, V_{\max} the maximum reaction velocity, and K_m the Michaelis constant, which equals the substrate concentration at half the maximum rate.

Governing equation (Michaelis-Menten equation):

$$v(\mathbf{x}; \boldsymbol{\beta}) = \left(\frac{V_{\max} s}{s + K_m} \right) \quad (9)$$

Variables and parameters:

$$\left. \begin{aligned} \mathbf{x} &= s \\ \boldsymbol{\beta} &= \begin{bmatrix} K_m \\ V_{\max} \end{bmatrix} \end{aligned} \right\} \quad (10)$$

The m^{th} element of the residue:

$$r_m = \left[y_m - \left(\frac{V_{\max} s_m}{s_m + K_m} \right) \right] \quad (11)$$

Elements of the m^{th} row of \mathbf{G} matrix:

$$[G_{m1} \quad G_{m2}] = \left[\frac{-V_{\max} s_m}{(s_m + K_m)^2} \quad \frac{s_m}{(s_m + K_m)} \right] \quad (12)$$

3.2. Model II: Competitive Inhibitor Enzyme Substrate Reactions

With other symbols remaining the same as in Model I, let i denote the concentration of inhibitor. The competitive reaction is governed by the Michaelis-Menten equation:

$$v(\mathbf{x}; \boldsymbol{\beta}) = \left(\frac{V_{\max} s}{s + K_m \left(1 + \frac{i}{K_I} \right)} \right) \quad (13)$$

Variables and parameters:

$$\left. \begin{aligned} \mathbf{x} &= \begin{bmatrix} s \\ i \end{bmatrix} \\ \boldsymbol{\beta} &= \begin{bmatrix} K_m \\ K_I \\ V_{\max} \end{bmatrix} \end{aligned} \right\} \quad (14)$$

The m^{th} element of the residue:

$$r_m = \left[y_m - \left(\frac{V_{\max} s_m}{s_m + K_m \left(1 + \frac{i}{K_I} \right)} \right) \right] \quad (15)$$

Elements of the m^{th} row of \mathbf{G} matrix:

$$[G_{m1} \quad G_{m2} \quad G_{m3}] = \left[\frac{-V_{\max} s K_I (K_I + i)}{(s K_I + K_m K_I + K_m i)^2} \quad \frac{V_{\max} s K_m i}{(s K_I + K_m K_I + K_m i)^2} \quad \frac{s K_I}{(s K_I + K_m K_I + K_m i)} \right]. \quad (16)$$

In passing, it might be mentioned that if the data points are collected for various inhibitor values, one could vectorize them into a single column. Accordingly, the value for i in Equation (16) should be substituted.

3.3. Model III: Allosteric Monosubstrate Enzyme

For our third model, the allosteric enzyme reaction is considered. This is governed by the Hill equation involving an empirical parameter n .

$$v(\mathbf{x}; \boldsymbol{\beta}) = \left(\frac{V_{\max} s^n}{s^n + K_m} \right) \quad (17)$$

Variables and parameters:

$$\mathbf{x} = [s] \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \quad \mathbf{\beta} = \left[\begin{array}{c} K_m \\ n \\ V_{\max} \end{array} \right] \quad (18)$$

The m^{th} element of the residue:

$$r_m = \left[y_m - \left(\frac{V_{\max} s_m^n}{s_m^n + K_m} \right) \right] \quad (19)$$

Elements of the m^{th} row of \mathbf{G} matrix:

$$[G_{m1} \quad G_{m2} \quad G_{m3}] = \left[\begin{array}{ccc} \frac{-V_{\max} s_m^n}{(K_m + s_m^n)^2} & \frac{V_{\max} K_m s_m^n \ln s}{(K_m + s_m^n)^2} & \frac{s_m^n}{(K_m + s_m^n)} \end{array} \right] \quad (20)$$

3.4 Model IV: Allosteric Enzyme with Two Species of Ligand Sites

This is governed by two Hill equations combined as:

$$v(\mathbf{x}; \boldsymbol{\beta}) = \left(\frac{V_{\max 1} s_m^{n_1}}{s_m^{n_1} + K_1} \right) + \left(\frac{V_{\max 2} s_m^{n_2}}{s_m^{n_2} + K_2} \right) \quad (21)$$

Variables and parameters:

$$\mathbf{x} = [s] \quad \left. \begin{array}{l} \\ \\ \\ \\ \\ \\ \end{array} \right\} \quad \mathbf{\beta} = \left[\begin{array}{c} K_1 \\ n_1 \\ V_{\max 1} \\ K_2 \\ n_2 \\ V_{\max 2} \end{array} \right] \quad (22)$$

The m^{th} element of the residue:

$$r_m = \left[y_m - \left(\frac{V_{\max 1} s_m^{n_1}}{s_m^{n_1} + K_1} \right) - \left(\frac{V_{\max 2} s_m^{n_2}}{s_m^{n_2} + K_2} \right) \right] \quad (23)$$

The six elements of the m^{th} row of \mathbf{G} matrix:

$$[G_{m1} \quad G_{m2} \quad G_{m3} \quad \cdots \quad G_{m4} \quad G_{m5} \quad G_{m6}] = \left[\begin{array}{cccccc} \frac{-V_{\max 1} s_m^{n_1}}{(K_1 + s_m^{n_1})^2} & \frac{V_{\max 1} K_1 s_m^{n_1} \ln s_m}{(K_1 + s_m^{n_1})^2} & \frac{s_m^{n_1}}{(K_1 + s_m^{n_1})} & \cdots & \frac{-V_{\max 2} s_m^{n_2}}{(K_2 + s_m^{n_2})^2} & \frac{V_{\max 2} K_2 s_m^{n_2} \ln s_m}{(K_2 + s_m^{n_2})^2} & \frac{s_m^{n_2}}{(K_2 + s_m^{n_2})} \end{array} \right] \quad (24)$$

In the next section four simulation examples are given based on the preceding four models.

4. Simulation Examples

4.1 Example I—Simple Uninhibited Reaction

Data consisting of 10 samples are simulated in case of a hypothetical enzyme with $K_m = 240$ nM and $V_{\max} = 80$ mmol/min/mg substrate using MATLAB. A zero mean unit variance random number with Gaussian density is added as measurement error. The substrate and the reaction rate are shown in **Table 1**. We shall take Σ as a $10 \times$

10 unit matrix. Thus, WRSS can be called RSS.

The parameters estimated using the program written in MATLAB are:

$\hat{K}_m = 237.10$, $\hat{V}_{\max} = 80.24$, and the RSS = 0.14². **Figure 1** shows the enzyme reaction rate versus substrate concentration in the units specified in **Table 1**. In order to render a better visualization, we have provided the relief of the error surface defined in Equation (2) with unity weights in **Figure 2(a)**. Also, **Figure 2(b)** shows the contours of the same plot. Note that the contours are ever widening ellipses with the global minimum shown by an asterisk. Also plotted in this figure is the learning trajectory obtained by joining all the estimated parameters over 6 iterations, starting from a provisional guess of $K_m(0) = 20$ and $V_{\max}(0) = 25$.

This simulation is repeated using the software Prism 5, a registered trademark of GraphPad, as well as the EXCEL template called anemona published in [12]. It was assuring to find that the parameters estimated were identical viz., $\hat{K}_m = 237.10$, and $\hat{V}_{\max} = 80.24$.

Table 1. Simulated data consisting of 10 samples in case of a hypothetical enzyme with $K_m = 240$ nM and $V_{\max} = 80$ $\mu\text{mol}/\text{min}/\text{mg}$ substrate for enzyme reaction model I.

Substrate Concentration, s (nM)	Reaction Rate v ($\mu\text{mol}/\text{min}/\text{mg}$ substrate)
0	0.00
50	14.12
100	23.86
150	31.10
200	36.69
400	50.33
600	57.47
800	61.87
1000	64.84
10000	78.45

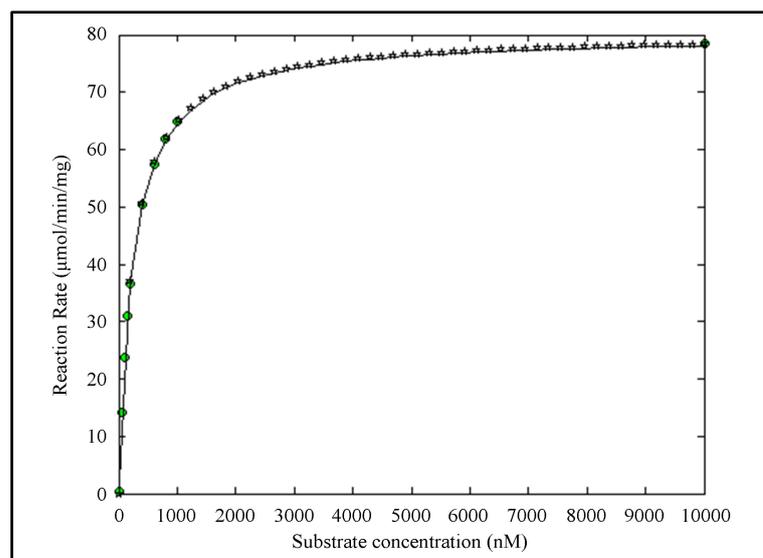


Figure 1. Reaction rate curve for Model I. The circles are simulated measurements, solid line is the theoretical curve, stars show the model using the estimated parameters.

²We denote the estimated value of a parameter by the hat symbol; for example, \hat{K}_m is the estimate of K_m .

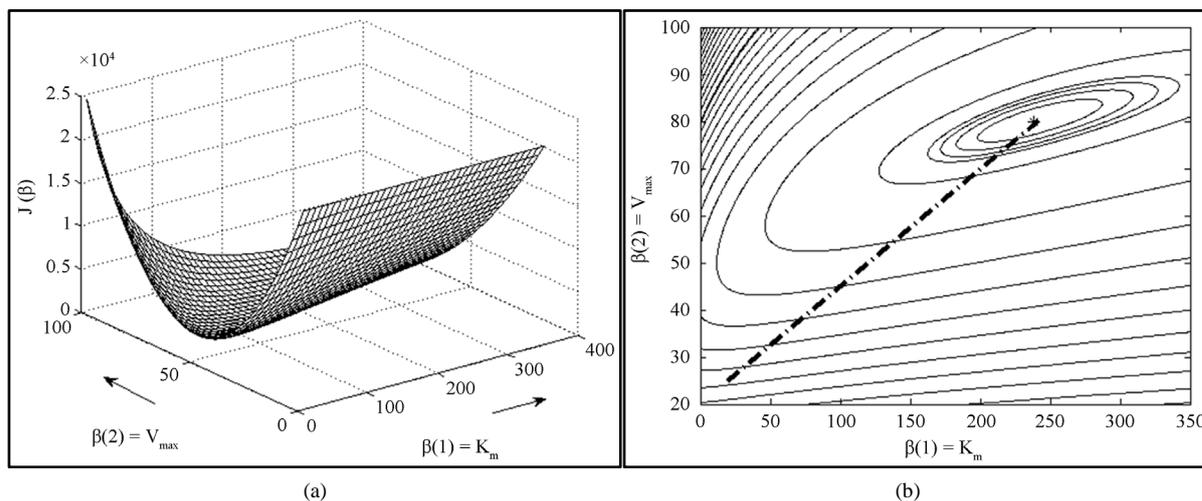


Figure 2. (a) Showing the relief of the error $J(\beta)$ as a function of the parameters. (b) Showing the contours of the error $J(\beta)$ as a function of the parameters. The global minimum is shown by an asterisk. Also shown is the learning trajectory.

4.2. Example II—Enzyme Competitive Inhibitor Case

As a second example, the data found in the GraphPad user manual [14] is used. It is reproduced in **Table 2** for convenience. The parameters for this data as given by Prism 5 are:

$$\hat{K}_m = 236.70, \hat{K}_i = 2.74, \hat{V}_{\max} = 79.92.$$

The parameters estimated by using MATLAB program via non-linear regression are:

$$\hat{K}_m = 237.12, \hat{K}_i = 2.75, \hat{V}_{\max} = 79.94.$$

Figure 3 shows the plot of reaction curves.

A visual display of the surface of the cost function J is not possible as it is now a function of three variables: V_{\max} , K_m and K_i . So an attempt is made to visualize this optimization problem in two steps. First, we note that when $i = 0$, Model II degenerates to Model I; as such, K_m and V_{\max} are found using the data of the first and second columns of **Table 2**. Secondly, using these parameters in the remaining data, J is plotted in **Figure 4**. The global minimum is found at $K_i = 2.75$ in conformity with the method of non-linear regression. In MATLAB, the min command obtains minimum of a given vector. Further, it may be pointed out that the modified Fibonacci search method used by Atkins in [9] also readily finds application to accomplish this.

4.3. Example III—pNPP Hydrolysis by Alkaline Phosphatase (Hill Equation)

Determination of kinetic parameters for the hydrolysis of p-nitrophenylphosphate (pNPP) by alkaline phosphatase was reported in [17] using a program called SigrafW. **Figures 5(a), (b)** show the familiar structure of pNPP and AP. For convenience, the data points of **Table 1** in [17] are reproduced in **Table 3** and are used in this example. The parameters obtained by SigrafW, and those obtained through MATLAB simulations, as well as Prism 5 of GraphPad, are shown in **Table 4**.

This exercise has been attempted using anemona reported in [12], but this tool has yielded absurd results. Both the MATLAB program and Prism are found to yield identical values for the parameters.

Using MATLAB we find that RSS is less and correlation is better. **Figure 6** shows the reaction rate curves using both the results. We note that the curve obtained using the parameters reported in [17] does not pass through the observations as accurately as reported therein. The MATLAB script file for this representative example is given in Appendix I. Modifying the code for other examples is straightforward as mentioned earlier.

4.4 Example IV—ATP Hydrolysis by Alkaline Phosphatase (Sum of Two Hill Equations)

Again we take the data available in [17] for the ATP hydrolysis by alkaline phosphatase. This reaction is characterized by the sum of two Hill equations. **Figure 5(c)** depicts the ATP structure. The measured data is found

Table 2. Data for competitive enzyme reaction (Model II) taken from the GraphPad user manual [14]. Here i denotes the concentration of inhibitor.

Reaction velocity for various inhibitors				
Substrate Concentration s , (nM)	$i = 0$ (nM)	$i = 3$ (nM)	$i = 10$ (nM)	$i = 30$ (nM)
0	0.00	0.00	0.00	0.00
50	17.00	7.20	3.60	0.00
100	22.00	13.30	8.10	0.10
150	30.60	17.80	13.60	4.80
200	33.60	25.10	10.80	5.20
400	50.40	38.00	21.90	14.60
600	58.40	45.00	25.50	13.50
800	62.70	50.50	31.80	15.90
1000	63.50	52.60	38.10	20.90
10000	81.00	74.30	70.90	61.70

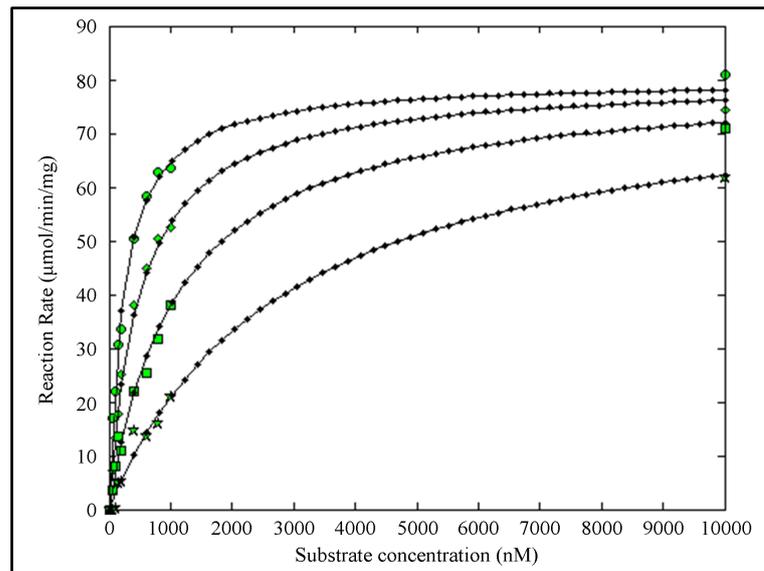


Figure 3. Reaction rate curves for Model II for various values of inhibitor concentrations. Circles: $i = 0$, diamonds $i = 3$, squares $i = 10$, stars $i = 30$. The solid line is obtained with values given by prism, dots are using MATLAB.

in **Table 2** of [17] reproduced here for convenience in **Table 5**.

For better numerical accuracy, all the substrate values are first up-scaled uniformly by the factor 10^8 before parameter estimation. They are later scaled down by the same factor. The parameters reported in [17], and those obtained through MATLAB program, as well as Prism 5 are shown in **Table 6**. Note that MATLAB and Prism 5 yield identical results except for a miniscule difference in n_2 and RSS.

The reaction rate curves are shown in **Figure 7**. In SigraW reported in [17] for the purpose of a good fit, a baseline concept is used. This amounts to a form of piece-wise regression. But the work in this paper does not rely on any baseline, as we treat the reaction as a single phenomenon. Reaction rate curve, using the parameters in [17] without the baseline, is also plotted in **Figure 7**. It does not pass through the measured data as seen in **Figure 7**.

Table 3. Hydrolysis of PNPP by alkaline phosphatase in 50 mM of 2-amino-2-hydroxymethylpropane 1 - 3 diol buffer, pH 9.4 containing 2 mM MgCl₂ [17].

[PNPP] (M)	Reaction rate (nmol/min/mg)
5.00e-5	1.60
1.00e-4	3.20
2.00e-4	5.50
3.00e-4	8.40
5.00e-4	13.50
7.00e-4	16.60
1.00e-3	19.00
2.00e-3	29.00
3.00e-3	36.00
5.00e-3	43.00
7.00e-3	43.50
1.00e-2	45.70
1.50e-2	47.00
2.00e-2	48.00

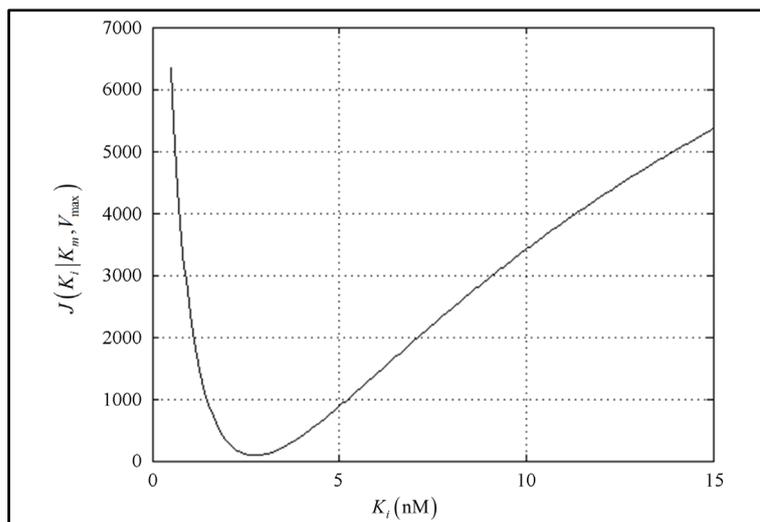
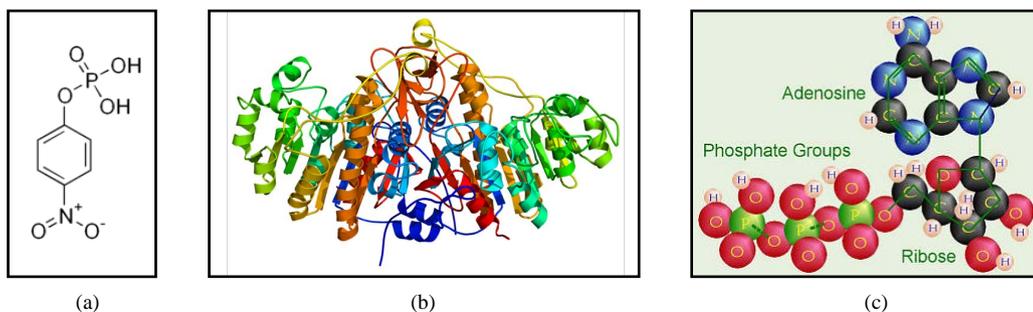
**Figure 4.** Plot of Cost J as a function of K_i given the values of K_m and V_{max} .**Figure 5.** Structures of (a) para-Nitrophenylphosphate (pNPP) (b) Alkaline Phosphatase (AP) (source: <http://pfam.sanger.ac.uk/family/PF00245>) (c) Adenosine triphosphate (ATP) (source: <http://www.abmservice.com/MeasuringATP.html>).

Table 4. Various parameters for Example III compared with those reported in [17] using SigrafW as well as Prism 5. Note that the MATLAB program developed in this paper and Prism 5 yield identical results.

Parameter	Reported in [16] Using SigrafW	Using MATLAB	Using Prism 5
K (M)	1.43e-3	8.41e-4	8.41e-4
n	1.08	1.08	1.08
V_{\max} (nmol/min/mg)	51.32	51.32	51.32
RSS	260.17	12.18	12.18
Correlation	0.99	1.00	1.00

Table 5. Hydrolysis of ATP by alkaline phosphatase in 50 mM Tris-HCl buffer, pH 7.5, containing 2 mM MgCl₂ [17].

[ATP] (M)	Reaction rate (Nmol/min/mg)	[ATP] (M)	Reaction rate (Nmol/min/mg)
1.00e-8	3.00	3.00e-5	27.90
2.50e-8	3.20	5.00e-5	32.20
5.00e-8	4.00	6.00e-5	33.70
7.50e-8	4.50	7.50e-5	36.40
1.00e-7	6.00	1.00e-4	38.90
2.50e-7	11.00	2.00e-4	50.70
5.00e-7	14.00	3.00e-4	61.50
7.50e-7	16.00	4.00e-4	64.70
1.00e-6	21.70	5.00e-4	69.50
2.00e-6	22.30	6.00e-4	77.50
3.00e-6	24.00	7.50e-4	86.50
5.00e-6	23.00	1.50e-3	91.30
7.50e-6	23.50	2.00e-3	102.60
1.00e-5	25.30	3.00e-3	102.90
2.00e-5	25.80		

Table 6. Various Parameters in Example IV using SigrafW, MATLAB and Prism 5.

Parameter	Reported in [17]	Using MATLAB	Using Prism 5
K_1 (M)	4.25e-7	2.67e-7	2.67e-7
n_1	1.32	0.96	0.96
$V_{\max 1}$ (nmol/min/mg)	22.02	24.78	24.78
K_2 (M)	4.28e-4	1.69e-3	1.69e-3
n_2	1.17	1.14	1.13
$V_{\max 2}$ (nmol/min/mg)	88.72	87.71	87.71
RSS	8165.9	93.71	89.81
Correlation	0.96	1.00	1.00

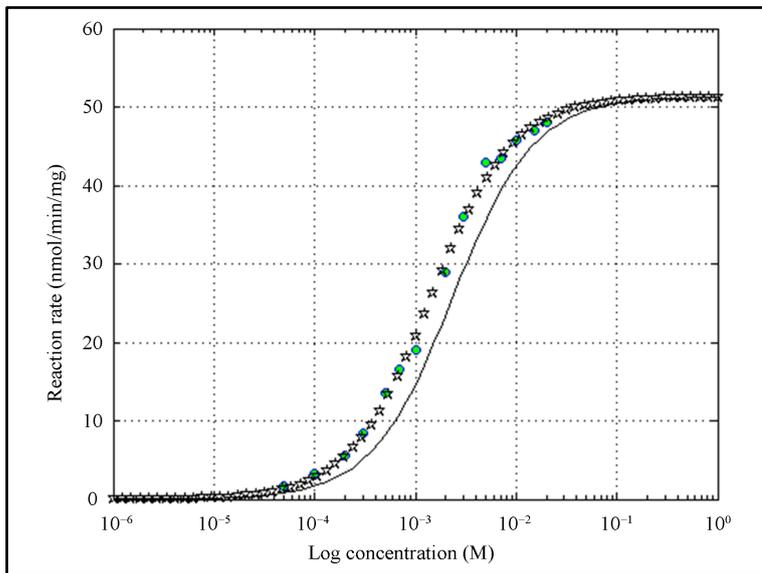


Figure 6. Showing the reaction rate in case of PNPP hydrolysis through alkaline phosphatase.

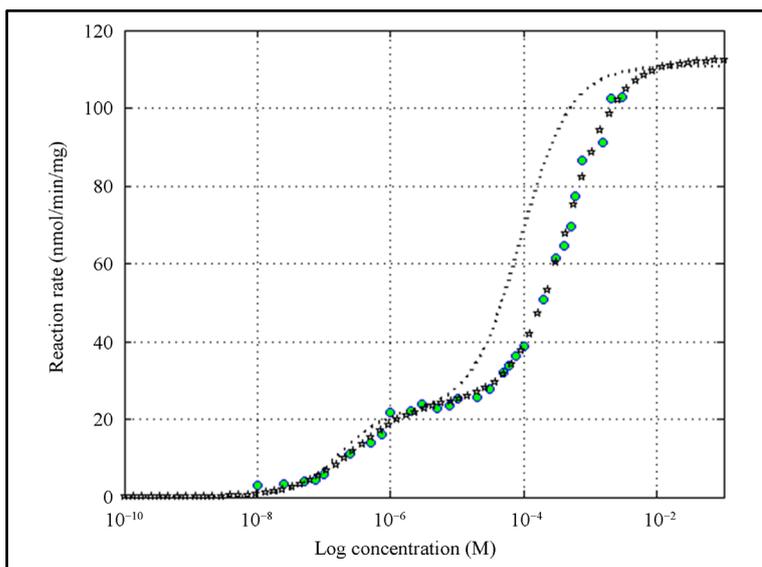


Figure 7. Reaction curve for the Example IV of allosteric response in ATP hydrolysis using alkaline phosphatase. Circles are measured values. Starred curve is obtained with the parameters estimated using MATLAB. In [17] the notion of baseline is used to obtain a good fit. Dotted curve shows the reaction rate without the use of baseline.

5. Discussion and Conclusion

Enzyme parameters can be found using graphical aids such as Lineweaver-Burk’s plot, or Dixon’s plot in the case of simple enzyme reactions. But for complex enzyme reactions, parameter estimation becomes computationally intensive. With the advent of software tools, this task is performed interactively on a computer. In the open literature, several software packages are reported. Almost all of them allow the user to feed the data and select the type of reaction for estimation of enzyme kinetic parameters; but they do not allow the user to write the source code for non-linear regression. This paper provides the mathematical expressions needed in non-linear regression for four types of enzyme reactions and suggests the use of MATLAB. Nonlinear regression is

briefly discussed first in general terms, and then specialized to four different models. The elements of various vectors and matrices are explicitly given for the convenience of programming in MATLAB. The MATLAB script for one of the examples is given in Appendix I. MATLAB offers better visualization and the ability to write the source code. Also, this study helps us to use these algorithms in more complex cases. Although in this paper we have given analytical expressions for the G matrix, one could use numerical differentiation for performing parameter estimation. Plotting the cost surface helps in visualizing where optimal values are located in the parameter space and helps to confirm the results obtained by non-linear regression. None of the available packages depict the surface of cost function and learning trajectory of estimation process wherever possible. But using MATLAB, this could be accomplished—thus, enabling the analysts to appreciate the analytical steps involved in the non-linear regression technique.

Acknowledgements

The authors thank Dr. Mary Connerty and Dr. Craig Warren of Penn State Erie, for helpful suggestions.

References

- [1] Cornish-Bowden, A. (2004) *Fundamentals of Enzyme Kinetics*. 3rd Edition, Portland Press Limited, London.
- [2] Engel, P. C. (1977) *Enzyme Kinetics—The Steady State Approach*. Chapman and Hill Limited, London.
- [3] Dixon, M. and Webb, E.C. (1964) *Enzymes*. 2nd Edition, Academic Press, New York.
- [4] Keener, J. and Snyed, J. (2004) *Mathematical Physiology*. Springer, New York.
- [5] Lineweaver, H. and Burk, D. (1934) The Determination of Enzyme Dissociation Constants. *Journal of American Chemical Society*, **56**, 658. <http://dx.doi.org/10.1021/ja01318a036>
- [6] Dixon, M. (1953) The Determination of Enzyme Inhibitor Constants. *Biochemical Journal*, **55**, 170-171.
- [7] Fukushima, Y., Ushimaru, M. and Takahara, S. (2002) On the Error of the Dixon Plot for Estimating the Inhibition Constant between Enzyme and Inhibitor. *Biochemistry and Molecular Biology Education*, **30**, 90-92. <http://dx.doi.org/10.1002/bmb.2002.494030020022>
- [8] Wilkinson, G.N. (1961) Statistical Estimation of Enzyme Kinetics. *Biochemical Journal*, **80**, 324-332.
- [9] Atkins, G.L. (1973) A Simple Digital Computer Program for Estimating the Parameters of the Hill Equation. *European Journal of Biochemistry*, **33**, 175-180. <http://dx.doi.org/10.1111/j.1432-1033.1973.tb02667.x>
- [10] Williams, P.A. (1983) ENZPACK: A microcomputer program to aid in the teaching of enzyme kinetics. *Biochemical Education*, **11**, 141-143. [http://dx.doi.org/10.1016/0307-4412\(83\)90096-1](http://dx.doi.org/10.1016/0307-4412(83)90096-1)
- [11] Leatherbarrow, R.J. (1988) Enzfitter: A Non-Linear Regression Data Analysis Program for IBM PC. *Journal of American Chemical Society*, **110**, 4100.
- [12] Hernandez, A. and Ruiz, M.T. (1998) An EXCEL Template for Calculation of Enzyme Kinetic Parameters by Non-Linear Regression. *Bioinformatics (Applications Note)*, **14**, 227-228. <http://dx.doi.org/10.1093/bioinformatics/14.2.227>
- [13] Heinrich, R., Melendez-Hevia, E. and Cabezas, H. (2002) Optimization of Kinetic Parameters of Enzymes. *Biochemistry and Molecular Biology Education*, **30**, 184-188. <http://dx.doi.org/10.1002/bmb.2002.494030030065>
- [14] Miller, J.R. (2003) GraphPad Prism version 4.0 Step-by-step Examples. GraphPad Software Inc., San Diego, 111-119.
- [15] Gonzalez-Cruz, J., Rodreguez-Sotres, R. and Rodriguez-Penagos, M. (2003) On the Convenience of Using a Computer Simulation to Teach Enzyme Kinetics to Undergraduate Students with Biological Chemistry-Related Curricula. *Biochemistry and Molecular Biology Education*, **31**, 93-101. <http://dx.doi.org/10.1002/bmb.2003.494031020193>
- [16] Clark, A.G. (2004) Lucenz Simulator—A Tool for the Teaching of Enzyme Kinetics. *Biochemistry and Molecular Biology Education*, **32**, 201-206. <http://dx.doi.org/10.1002/bmb.2004.494032030350>
- [17] Leone, F.A., Baranauskas, J.A., and Furiel, R.P.M. (2005) SigrafW: An Easy to Use Program for Fitting Enzyme Kinetic Data. *Biochemistry and Molecular Biology Education*, **33**, 399-403. <http://dx.doi.org/10.1002/bmb.2005.49403306399>
- [18] Palm III, W.J. (2008) *A Concise Introduction to MATLAB*. 1st Edition. McGraw Hill, New York.
- [19] Daku, B. (2005) *MATLAB Tutorial CD: Learning MATLAB Superfast*. Wiley, New York.
- [20] Pratap, R. (2005) *Getting Started with MATLAB 7—A Quick Introduction for Scientists and Engineers*. Oxford University Press, Oxford.
- [21] Davis, T.A. and Sigmon K. (2004) *MATLAB Primer*. CRC Press. <http://dx.doi.org/10.1201/9781420034950>

- [22] Cobelli, C. and Carson, E. (2008) Introduction to Modeling in Physiology and Medicine. Academic Press, Waltham, 212-220.
- [23] Moore, E.H. (1920) On the Reciprocal of the General Algebraic Matrix. (Abstract) *Bulletin of American Mathematical Society*, **26**, 394-395.
- [24] Penrose, R.A. (1955) A Generalized Inverse for Matrices. *Proceedings of Cambridge Philosophical Society*, **51**, 406-413. <http://dx.doi.org/10.1017/S0305004100030401>

Appendix I

```

%% Matlab script for Example III. To reproduce Figure 5
clear all; close all
%% data
s=[5e-5;1e-4;2e-4;3e-4;5e-4;7e-4;1e-3;2e-3;3e-3; ...
  5e-3;7e-3; 1e-2;1.5e-2;2e-2];
v=[1.6;3.2;5.5;8.4;13.5;16.6;19;29;36;43;43.5;45.7;47;48];
m=length(s);
%% semilog plot of the data
semilogx(s,v,'o','MarkerSize',5,'MarkerFacecolor','g'); hold
on
% Initial values
Km=1e-3; n=1; Vmax=20;
for q=1:10
g1 = -Vmax*s.^n./((Km+s.^n).^2);
g2 = Vmax*s.^n.*log(s)./ ...
  (Km+s.^n)-Vmax*((s.^n).^2).*log(s)./(Km+s.^n).^2;
g3 = s.^n./(Km+s.^n);
r = v - Vmax*s.^n./(Km+s.^n); Z=[g1 g2 g3]; G=Z\r;
Km= (Km+G(1)); n = (n+G(2)); Vmax=(Vmax+G(3));
end
si=logspace(-6, 0,69);
vio = Vmax*si.^n./(Km+si.^n);
semilogx(si,vio,'pk','MarkerSize',7)
% the following are from ref[17]
VMAX= 5.132e1;
KM=1.427e-3;
N=1.081;
vi = VMAX*si.^N./(KM+si.^N);
semilogx(si,vi,'k')
xlabel('Log concentration (M)','FontSize',14)
ylabel('Reaction rate (nmol/min/mg)','FontSize',14)
grid; grid minor
axis ([1e-6 1 0 55])
%% residual sum of squares and correlation
RSS = (norm(v-VMAX*s.^N./(KM+s.^N)))^2
c=corrcoef(v,VMAX*s.^N./(KM+s.^N))
RSSO = (norm(v-Vmax*s.^n./(Km+s.^n)))^2
cO=corrcoef(v,Vmax*s.^n./(Km+s.^n))
% end of code

```