

Fitting the Michaelis-Menten Equation in MATLAB

Disclaimer: The goal of this example is to walk you through how to conduct a nonlinear fit in MATLAB and compare the results between a nonlinear fit and linear fit for a given data set. Before reviewing this example it is recommended that you read over *Fitting Curves to Data using Nonlinear Regression* and *Conducting a Nonlinear Fit Analysis in MATLAB* so that you can be familiar with the theory and functions involved.

Example Problem

The hydrolysis of carbobenzyglycyl-L-tryptophan catalyzed by pancreatic carboxypeptidase occurs according to the reaction:



The following data on the rate of formation of L-tryptophan at 25°C and pH 7.5 was obtained:

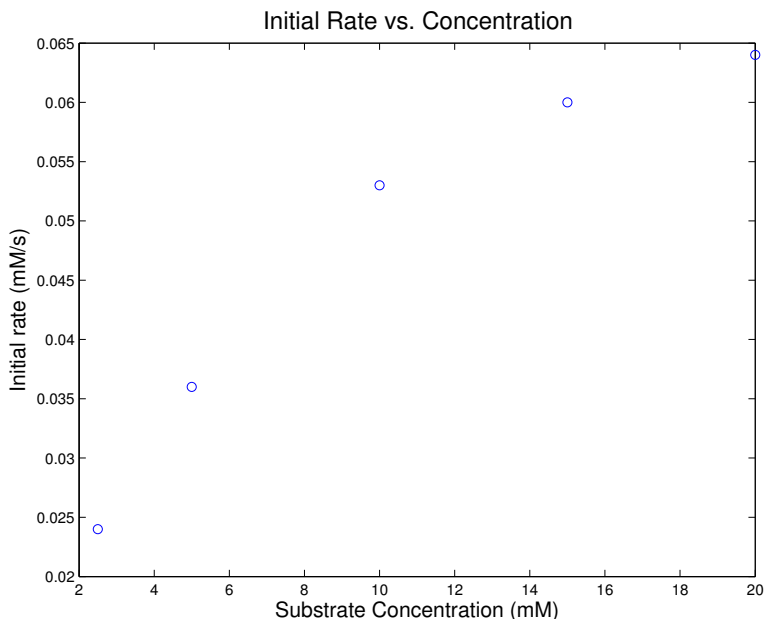
Substrate Concentration (mM)	Rate (mM s ⁻¹)
2.5	0.024
5.0	0.036
10.0	0.053
15.0	0.060
20.0	0.064

- Plot the initial rate as a function of substrate concentration. What shape does the data follow?
- Fit a nonlinear curve to this data of the form $f(x, \beta) = \frac{\beta_1 x}{\beta_2 + x}$ using the `nlinfit` function in MATLAB. Plot your fit and the data points on the same graph. Does the fit seem reasonable? Determine the 95% confidence intervals for β_1 and β_2 , and find the r^2 value for the fit. Are they what you expect?
- Make a residual plot to assess the fit from part b.
- Now linearize the model using the Lineweaver-Burk method and solve for V_{max} and K_M . Find the 95% confidence intervals for the slope and intercept of your Lineweaver-Burk plot and determine the r^2 value.
- Make a residual plot to assess the fit from part d.
- Conduct an F-test to see which model is the better fit. For help with F-tests see *Fitting Curves to Data Using Nonlinear Regression* and *Using the F-test to Compare Two Models*.
- Based on the above analysis, is one method preferable to the other? Compare the parameter values found using each method and the result from the F-test.

Example Problem Solution

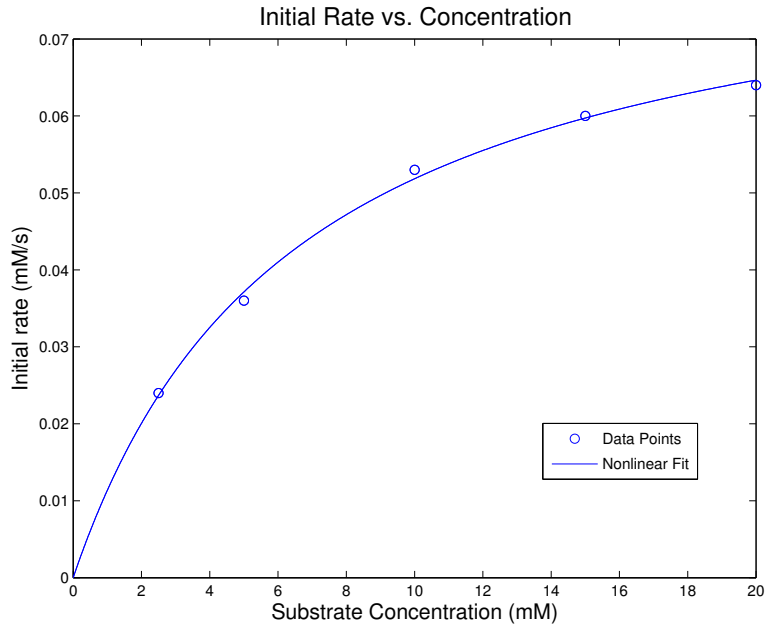
It is recommended that you try the problem before looking at the solution. For an example of code that could be used to come up with the solution see the Appendix.

- a. The following figure shows the plot of the raw data. The slope appears to decrease for increasing concentrations, eventually leveling off. This is expected for enzyme kinetics data. As substrate concentration increases, the rate also increases. However, the rate approaches a maximum value above which no amount of additional substrate will influence it. This is often referred to as the saturation point.

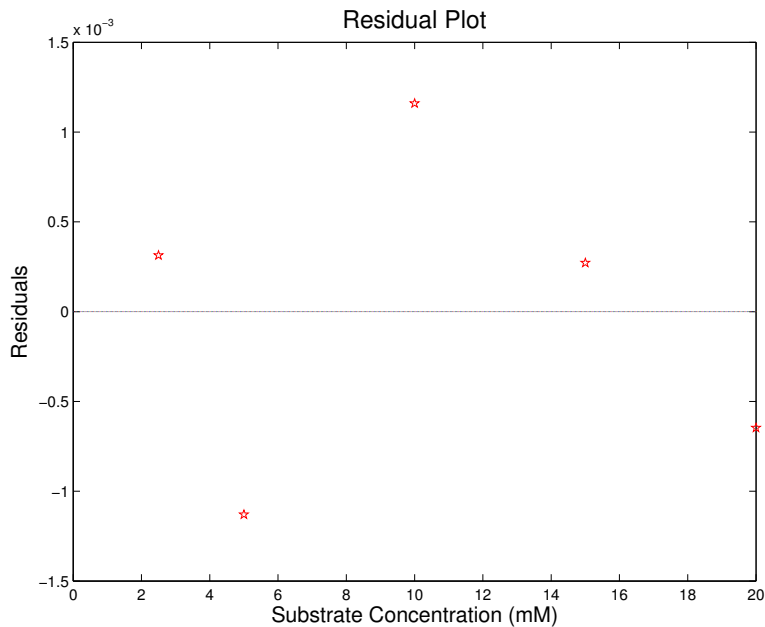


- b. Using MATLAB's `nlinfit` command, a curve for this data can be generated. The fit function was $f(x, \beta) = \frac{\beta_1 x}{\beta_2 + x}$, where β_1 was V_{max} and β_2 was K_M . The calculated constants with 95% confidence intervals were $\text{beta}(1) = 0.0859 \pm (0.0074)$ mM/s and $\text{beta}(2) = 6.5619 \pm (1.5235)$ mM. The r^2 value was 0.9972. These values were calculated using the methods detailed in the *Fitting Curves to Data using Nonlinear Regression* document.

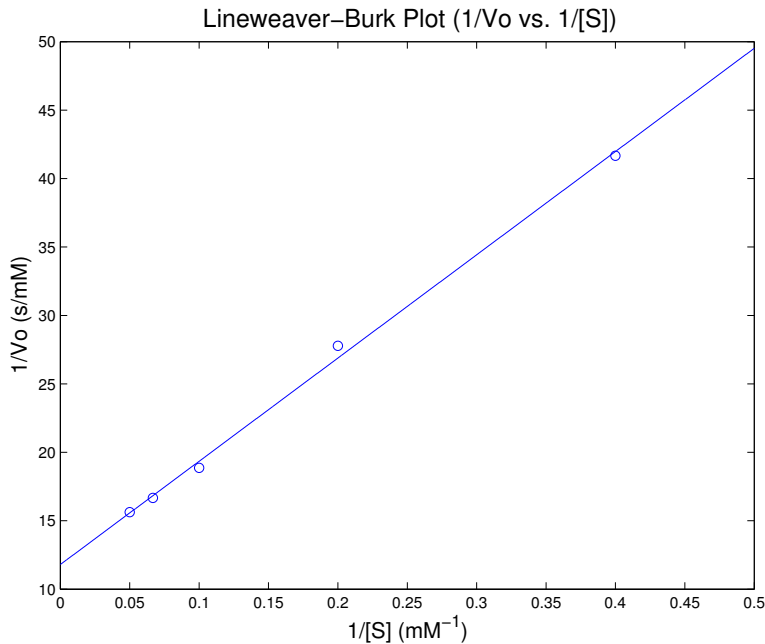
The plot of the raw data with this fit is shown below. Note that the curve appears to be a good fit for the data, passing through or near each of the data points. MATLAB has minimized the residual sum of squares to produce this model. It iterated through values for the constants β_1 and β_2 until it found values that minimized the sum of the squares of the residuals (data point minus point on curve).



c. The residual plot for this data is shown below. To plot the residuals, plot the R value returned by the **nlinfit** function as a function of the independent variable. Each value in the R vector is the data point subtracted by the value the model estimated for each value of the independent variable. When looking at residual plots you must search for a pattern or long runs of positive or negative values. It is also best for the residuals to have small relative magnitudes. The plot below fits the criteria for a good fit (i.e. no apparent pattern, small relative magnitudes as noted by the scaling on the y-axis). However, there are only five data points so it is difficult to conclude with certainty.



d. The data can be linearized using the equation: $\frac{1}{V_0} = \frac{K_M}{V_{max}} \frac{1}{[S]} + \frac{1}{V_{max}}$. For a derivation of this equation refer to your textbook or class notes. The plot of the linearized data, along with the linear fit, is shown below.



MATLAB can be used to solve for the intercept and slope. Note 95% confidence intervals are also obtained. If you are unsure how to obtain these values please review the linear fitting tutorial.

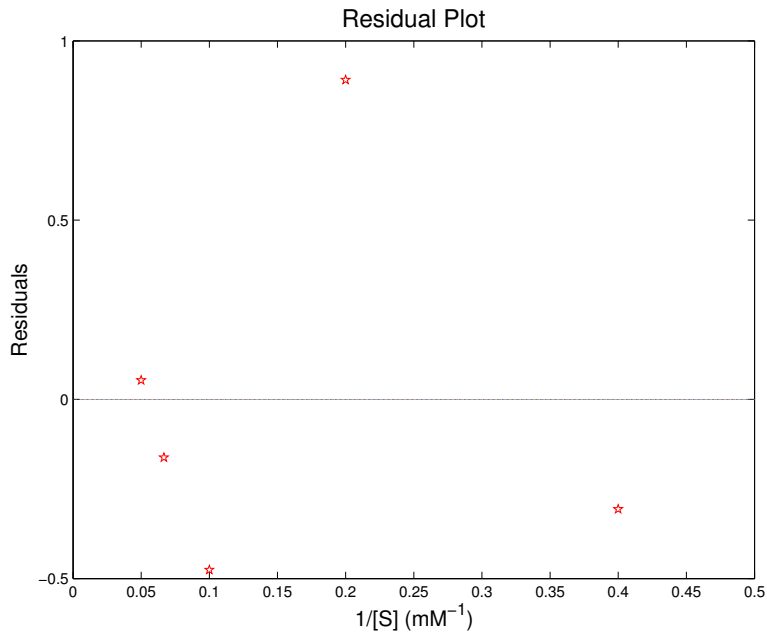
- intercept= $1/V_{max} = 11.8 \pm 1.42$
- slope= $K_m/V_{max} = 75.43 \pm 7.05$

Extracting V_{max} and K_M gives:

- $V_{max} = 0.0847$ mM/s
- $K_m = 6.3923$ mM

Finally, the r^2 value was calculated to be $r^2 = 0.9976$. The method used was the same as in the nonlinear case and is discussed in the *Fitting Curves to Data using Nonlinear Regression* document.

- e. The residual plot for the linearized data is shown below. Once again, look for patterns, long runs of positive or negative values, or large relative magnitudes. In this case none of these trends can be seen and the residual plot is indicative of a good fit. However, the fact there are only five data points limits the conclusions that can be made from the residual plot alone.



- f. Conducting the F-test with the linear fit as model 1 and the nonlinear fit as model 2 gives an F-statistic of $3.5603e+05$ and a p-value of $7.9913e-09$. Interpreting this result, the nonlinear fit is significantly better than the linearized, transformed fit. This result makes sense, because transforming data tends to exaggerate errors.
- g. Based on this analysis, both methods provide good fits for the data. The nonlinear case returned 0.0859 ± 0.0074 mM/s for V_{max} and 6.5619 ± 0.5235 mM for K_M , while the linearized method returned 0.0847 mM/s for V_{max} and 6.3923 mM for K_M . The values for the linearized method fall within the confidence bounds of the nonlinear version. Additionally, both models had r^2 values close to 1 and residual plots that were indicative of good fits. Despite the apparent similarity in these results, the F-test revealed that the nonlinear fit is statistically better than the linear fit. Linearizing the data makes the parameters easier to calculate by hand, but if a computer is available, the nonlinear method is the better choice.

A MATLAB Code

```
1 % Assign the data vectors
2 substrate=[2.5 5 10 15 20];
3 vo=[.024 .036 .053 .06 .064];
4
5 % Plot the raw data to get a look at the shape
6 figure(1)
7 clf
8 plot(substrate,vo,'o')
9 xlabel('Substrate Concentration (mM)','FontSize',14)
10 ylabel('Initial rate (mM/s)','FontSize',14)
11 title('Initial Rate vs. Concentration','FontSize',16)
12 print -depsc rawdataplot
13
14 % Calculate unknown coefficients in the model using nlinfit
15 model=@(b,x) b(1).*substrate./(b(2)+substrate);
16 initialguess=[1 1];
17 [beta,R,J,CovB,MSE] = nlinfit(substrate,vo,model, initialguess);
18 % 95% CI of coefficients
19 betaci = nlparci(beta,R,J);
20
21 % Plot the nonlinear fit with the raw data
22 figure(2)
23 plot(substrate,vo,'o')
24 hold on
25 x=linspace(0,20,1000);
26 f=beta(1).*x./(beta(2)+x);
27 plot(x,f,'-')
28 xlabel('Substrate Concentration (mM)','FontSize',14)
29 ylabel('Initial rate (mM/s)','FontSize',14)
30 title('Initial Rate vs. Concentration','FontSize',16)
31 legend('Data Points','Nonlinear Fit',0)
32 print -depsc rawdatawithfit
33
34 % Make a residual plot
35 figure(3)
36 clf
37 plot(x,0,'-',substrate,R,'rp')
38 xlabel('Substrate Concentration (mM)','FontSize',14)
39 ylabel('Residuals','FontSize',14)
40 title('Residual Plot','FontSize',16)
41 print -depsc nonlinresiduals
42
43 % Compute the r^2 value
44 ssresnonlin=sum(R.^2);
45 sstotnonlin=sum((vo-mean(vo)).^2);
46 rsqrnonlin=1-(ssresnonlin/sstotnonlin);
47
48 % Plot the transformed data, it should be linear
49 figure(4)
50 clf
51 plot(1./substrate,1./vo,'o')
52 xlabel('1/[S] (mM^{-1})','FontSize',14)
53 ylabel('1/Vo (s/mM)','FontSize',14)
54 title('Lineweaver-Burk Plot (1/Vo vs. 1/[S])','FontSize',16)
```

```

55
56 % Set up fittype and options
57 ft = fittype( 'poly1' );
58 opts = fitoptions( ft );
59 opts.Lower = [-Inf -Inf];
60 opts.Upper = [Inf Inf];
61
62 % Fit model to data and find 95% confidence bounds
63 [xData, yData] = prepareCurveData( 1./substrate, 1./vo );
64 [fitresult, gof] = fit( xData, yData, ft, opts );
65 p=polyfit(1./substrate,1./vo,1);
66
67 % Plot linear fit
68 x=linspace(0,.5,1000);
69 line=p(1).*x+p(2);
70 figure(4)
71 hold on
72 plot(x,line,'-')
73 print -depsc rawlin
74
75 % Make residual plot
76 Rlin=(1./vo)-((p(1).*(1./substrate)+p(2)));
77 figure(5)
78 clf
79 plot(1./substrate,Rlin,'rp')
80 hold on
81 plot(x,0,'-')
82 xlabel('1/[S] (mM-1)', 'FontSize',14)
83 ylabel('Residuals', 'FontSize',14)
84 title('Residual Plot', 'FontSize',16)
85 print -depsc linres
86
87 % Determine constants from linear fit
88 Vmax=1/(p(2));
89 Km=Vmax.*p(1);
90
91 % Determine the r2 value
92 ssreslin=sum(Rlin.^2);
93 sstotlin=sum((1./vo-mean(1./vo)).^2);
94 rsqrlin=1-(ssreslin/sstotlin);
95
96 % F-test between nonlinear and linear models
97 F=ssreslin./ssresnonlin;
98 df=length(vo)-2;
99 P=1-fcdf(F,df,df);

```