

Determination of Parameters Associated with the Equilibrium Model

("Matlab Application for Fitting Progress Curves to the Equilibrium Model – Version 2.5, 15/4/10")

The general procedures for carrying out the necessary rate determinations required for accurate determination of the Equilibrium Model parameters, and fitting this data to the mathematical model to generate the parameters, are described in "Peterson, M.E., Daniel, R.M., Danson, M.J. & Eisenthal, R. (2007) The dependence of enzyme activity on temperature: determination and validation of parameters. *Biochemical Journal*, **402**, 331-337". It should be borne in mind that the Equilibrium Model equation contains exponentials of exponentials – quite small deviations from ideal behaviour, or a failure to obtain true V_{max} values, may lead to difficulty in obtaining reliable Equilibrium Model parameters.

INSTRUCTIONS FOR USING MATLAB PROGRESS CURVE FITTING SOFTWARE (for Windows XP and Vista)

Software is copyright, James McDowall, Nigel Goodhue, Karin Bryan, Daniel Hailstone, Colin Monk and Michelle Peterson, 2007. This software is available for download from The University of Waikato Research Commons, <http://hdl.handle.net/10289/3791> Please note: this application is strictly for non-commercial use only.

CONTENTS

A] To install the program

B] To Perform a Fit

1. Using the Example data file
2. Using your own data
 - i)Preparing the data
 - ii)Processing
 - iii)Checking the results
- Generating simulation plots
- Other checks

Appendix:

Equations for generating simulated data sets

A] To install the program:

Note that Administrator-level access is required to run the installation process, but not to run the installed program. This application is for Windows XP and Vista. Microsoft Excel™ is required for the application to run.

1. Download the compressed archive "EQM_Fitting_Software_ver.2.5.zip" to your hard drive.
2. Extract the contents of the archive to any convenient location.
3. Run the installation program "MCRInstaller.exe" which is located in the Program_Files folder — please note, if you have a previously installed version of this application, you will be prompted to either remove or repair the application. Select repair.
4. Continue installation as per instructions on screen.
5. Start menu and/or desktop shortcuts to the "fitting.exe" program must be created manually if they are desired.

B] To perform a fit:

1. Using the example data file

- a. Run "fitting.exe" (in "Program_Files" folder).
- b. You will be prompted to open an Excel .xls file – navigate to "Fitting_Files" folder and select either "Example Data - Progress Curves Matlab Input.xls" or "Example Data - Initial Rates Matlab Input.xls".
- c. The program will now perform the fit (low-resolution, no standard deviations or confidence intervals for the fit will be calculated).

- d. Once fitting is complete, you will be asked if you wish a high resolution fit to be performed (the runtime of this routine is dependent on a number of factors, please see the notes below).
- e. Note that the option to do the high-resolution fit only applies when the “Example Data - Progress Curves Matlab Input.xls” data is used.
- f. If the "Example Data - Initial Rates Matlab Input.xls" data is used, one graph of initial rates vs. temperature (actual data and fitted curve) is automatically produced (figure 1 below); the option to continue with the high-resolution analysis is not available.

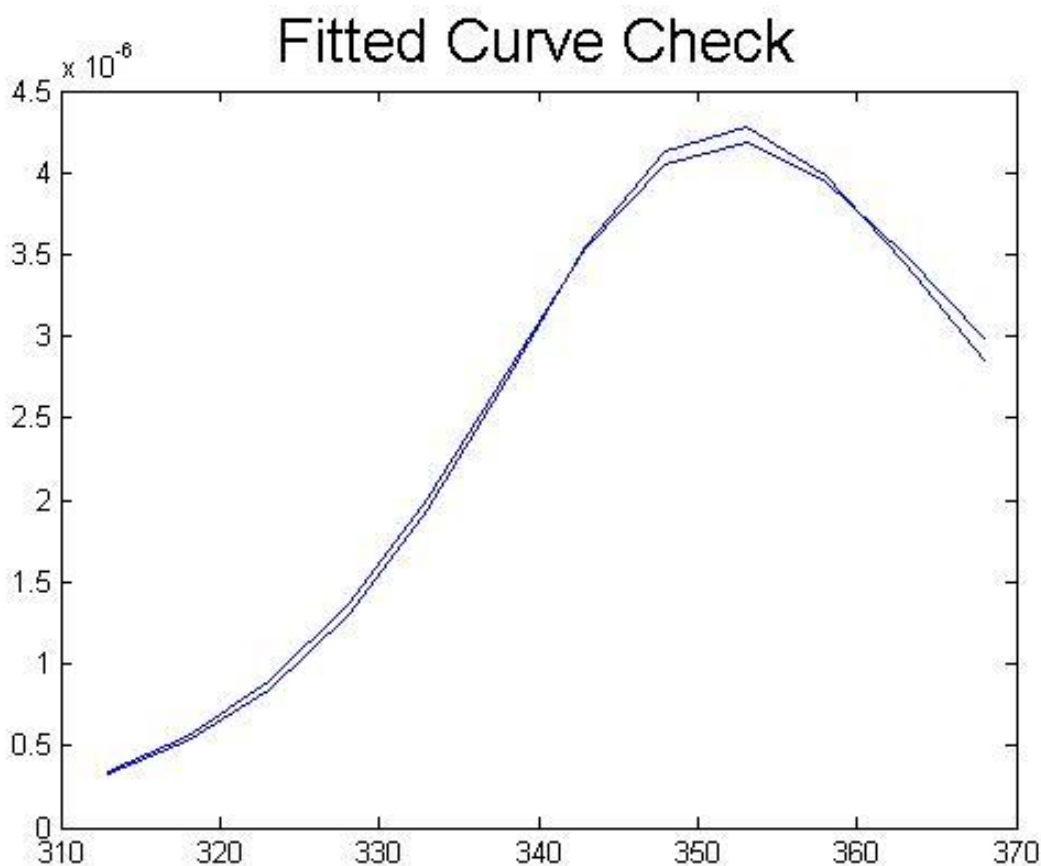


Figure 1

- g. If the “Example Data - Progress Curves Matlab Input.xls” data is used, and "No" is selected after the initial processing, the program will immediately plot three graphs (combined in a single figure; see figure 2 below): one that checks the fit against the experimental progress curve data, one that shows the residuals between the fit and the experimental data, and the third expressing the fit as reaction rates versus temperature versus time. (The axes are normally not labelled - the labels here have been manually inserted.)

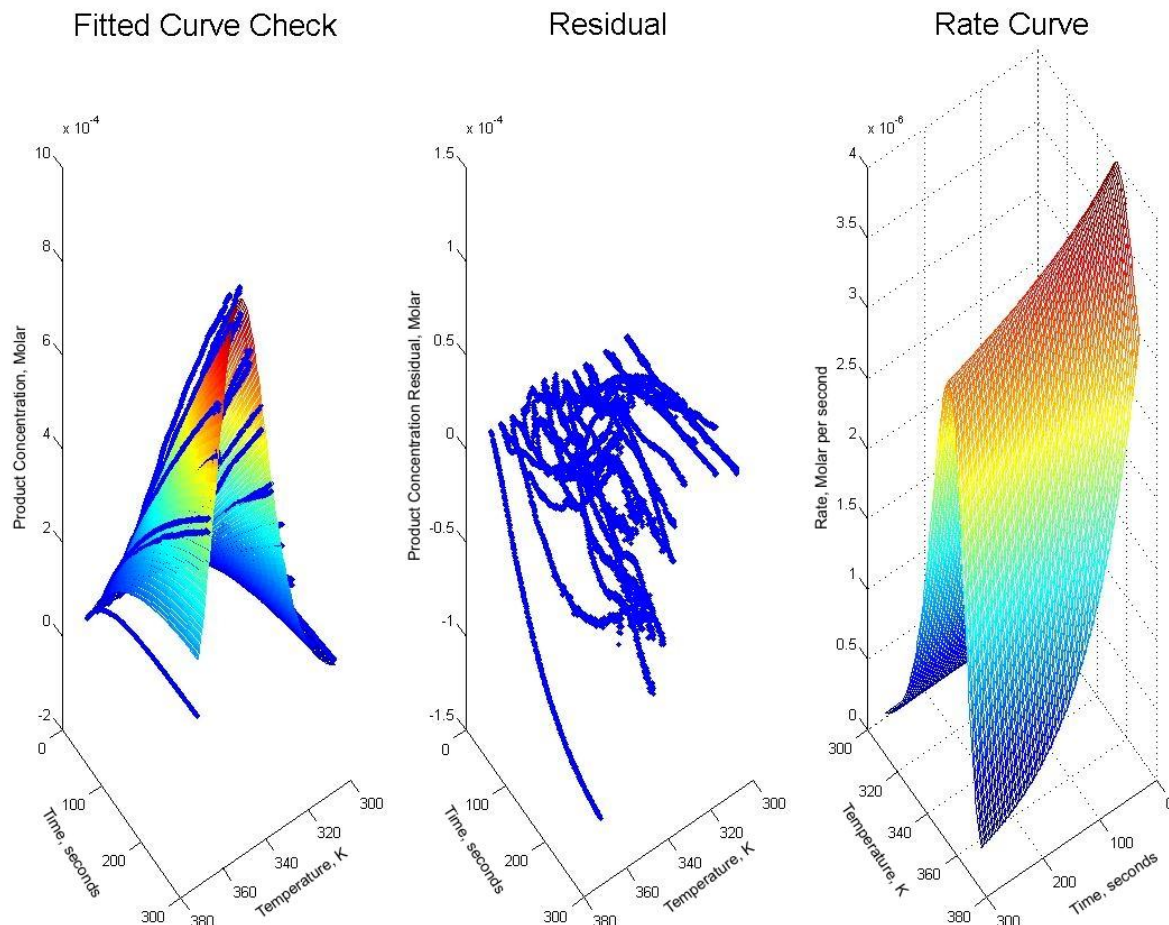


Figure 2

These graphs can be independently rotated in 3D, labels and colour bars may be added or removed and the set of graphs saved. Note that saving in the default (*.fig) format will create a file which can be opened and manipulated with a full Matlab software installation, but is not otherwise easily accessed. We routinely save the figure as a .jpg file. Also note that once a graph window is closed, the graph cannot be regenerated without re-running the analysis. When all graph windows are closed, the fitting.exe application terminates.

- h. If "Yes" is selected, the high resolution routine will run and once complete, the program will plot three graphs as above, and an additional two separate graphs will be plotted, illustrating the variability around the calculated value for each parameter (figure 3), and an error analysis around the residual mean squared error for the data set (figure 4). See also the comments in the "Processing" section below regarding reiterative processing.

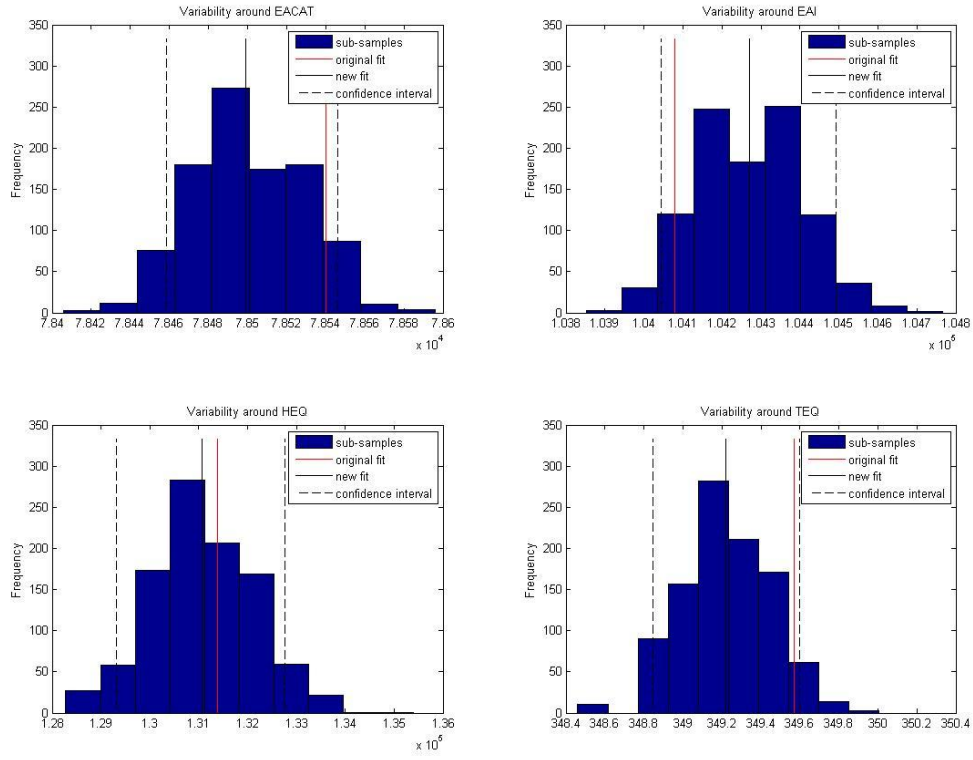


Figure 3

In the above plots, the vertical red line indicates the parameter as calculated by the initial low-resolution process; the solid black vertical line indicates the value as calculated by the high-resolution process. Ideally, the plots will be bell-shaped, with little difference between the positions of the red and black vertical lines.

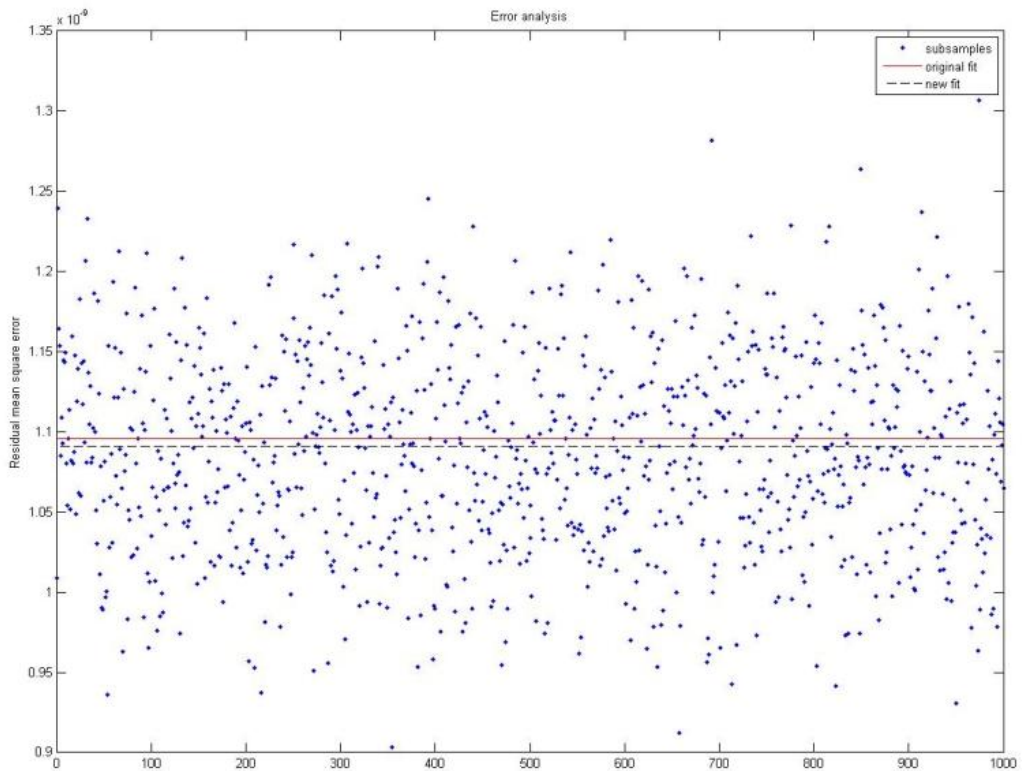


Figure 4

- i. On opening the example Excel file after the fitting is complete, the parameter values generated by the low-resolution analysis will appear in the green column of the "Settings-Output" sheet in this file; the parameter values generated by the high-resolution analysis (if it was run) will appear in the yellow column, with the calculated statistics in the uncoloured columns. If the "Example Data - Initial Rates Matlab Input.xls" data is used, EAI is not determined.
- j. Parameter values (EACAT, EAI, HEQ) are expressed in Joules per mole. TEQ is expressed in Kelvin. Please note, EACAT, EAI and HEQ correspond to $\Delta G_{\text{cat}}^{\ddagger}$, $\Delta G_{\text{inact}}^{\ddagger}$ and ΔH_{eq} respectively.
- k. An extra worksheet will also be generated in your Excel file with the reaction rate versus temperature versus time data for the fit. The original data is time, temperature, and product concentration. This applies for both low-resolution and high-resolution processing of the "Example Data - Progress Curves Matlab Input.xls" data, but not for the "Example Data - Initial Rates Matlab Input.xls" data.
- l. The example data set provided has been processed; the output and figures are in the folder "Fitting_Files \ Example Data - Output and Figures".

2. Using your own data

i) Preparing the data

- a. For Progress Curves, prepare your data file as per the "Data" sheet in the Excel file labelled "Example Data - Initial Rates Matlab Input.xls" (in "Fitting_Files" folder). An empty template spreadsheet is provided as "Template.xls". Note that the sequence of the worksheets must be maintained, as must the position of the table within the "Settings-Output" worksheet, to ensure correct reading and writing by the application.
- b. For Initial Rates, prepare your data file as per the "Data" sheet in the Excel file labelled "Example Data - Initial Rates Matlab Input.xls".
- c. It is advisable to copy the final formatted data to a fresh copy of the Template.xls file. We have found that preprocessing of all data (converting raw data to concentrations) in a single Excel file, and subsequently deleting redundant sheets to leave the two sheets as present in the template file, can cause the Matlab application to fail to run. This is due to Excel's maintenance of records of deleted sheets.
- d. Ensure that input data is expressed correctly—temperature in degrees Kelvin, time in seconds. Product and enzyme concentration must be expressed in MOLAR (that is, moles per litre). In the case of initial rates, the rate must be expressed in MOLAR per second.
- e. Enter the enzyme concentration into the cell labelled "E0" in the "Start" column in the "Settings-Output" worksheet.
- f. Time courses should all be of equal length and equal data frequency and so have equal numbers of data points.
- g. Some exceptions to the above comment will still allow the process to run, but others will have various effects according to the number and positions of omissions made. Some examples of the effects of omissions (with an empty cell in the data matrix) are:
 - i. Removing a small proportion (~2%) of the points at random (including one zero time value) doesn't affect the outcome.
 - ii. Removal of a contiguous block of about 5% of the points from a single run gives only slightly changed output, but no accuracy or regression output.
 - iii. Removal of the first two points from a single run will cause the whole process to fail, with no change from the starting parameters.
 - iv. Removing the first five points from all of the runs produces output similar to the control; in this case there are no empty cells, rather, the entire rows were deleted, with the data table starting at a non-zero time value.
- h. Although it is not necessary to have the columns (temperatures) in sequence, if performing an initial rates analysis it is best to have them so, as the graph produced has the experimental data points connected point-to-point in order of spreadsheet columns, not in order of temperature values.
- i. If the data set for the Progress Curves process contains more than one set of values for a given temperature, those values will be averaged before processing begins. This is not the case for Initial Rates data sets.
- j. If the data set consists of negative values, for example from collecting decreasing absorbance values, the sign must be changed.
- k. Check that the data in the final spreadsheet are numerical values, and not results of calculations.
- l. Save the file to disk.

ii) Processing

- Follow the procedures given above for the example data set, to perform the analysis.
- Runtime for the high-resolution routine is dependent on the number of data points, which in turn has an effect on the standard deviation of the results. On our desktop PCs (typically 2 GHz, 1GB RAM) a calculation involving ~30,000 data points (11 temperatures in duplicate, 330 second assay, data points taken at intervals of 0.25 second) takes about 2.5 hours to process.
- The relationship between the number of sample points and the standard deviation of the fit of the results is less simple; the standard error will depend on data reproducibility, and where in the time course the points are taken.
- The table below provides a very rough guide to the effects of data point reduction on standard error for the experimental run described, and on run time.

Parameter	Relative (and actual) standard deviation		
	30,000 data points	3000 data points	300 data points
$\Delta G_{\text{inact}}^{\ddagger}$	1x (0.15KJ/Mol)	1.6x (0.24KJ/Mol)	5.4x (0.80KJ/Mol)
$\Delta G_{\text{cat}}^{\ddagger}$	1x (0.035KJ/Mol)	3.6x (0.13KJ/Mol)	10.6x (0.37KJ/Mol)
ΔH_{eq}	1x (1.4KJ/Mol)	1.7x (2.4KJ/Mol)	3.3x (4.6KJ/Mol)
T_{eq}	1x (0.39K)	2.7x (1.1K)	8.7x (3.4K)
Run Time	~8400sec	~840sec	~90sec

- The application is CPU-intensive, so it is recommended that no other CPU-intensive processes be run concurrently on single-core systems. Multiple-core CPUs can perform simultaneous analyses with negligible overhead, but it may be difficult to match the output graphs with the corresponding datasets at the end of the runs.
- A minimum of 512MB of RAM is recommended, but the run time is not significantly affected by an increase from 1 GB to 2 GB.
- The graphs generated by the application may be manipulated to a limited extent prior to saving as a jpg file.
- Rotation in 3D, resizing, and the addition of labels and colour bars is possible.
- Note that the program does not prompt to save the graphs if the windows are closed, and that the graphs cannot be regenerated once they are closed, even if the application is still running, so care needs to be taken to ensure that they are saved. Regenerating a lost graph requires reprocessing the data.
- The application will close when all graph windows are closed.
- We have found it useful, when starting from arbitrary initial values for the parameters, to run the quick, low resolution process (selecting "No" at the prompt for full processing), and to use the output parameters as initial values for a second run. This process is repeated for a third low-resolution run, and the output from this run is used to start a fourth run, which is carried on through the high-resolution process.
The output from a good data set will not change significantly with repeated runs, but noisier data, or inappropriate starting values, will produce improving output values, indicated by decreasing standard deviation values, and an improvement in the shape of the histograms in Matlab Figure 2. This effect can be seen in the figures included in the folder "Example Data - Output and Figures".
 R^2 values improve as the runs are performed; values should ideally be better than 0.99.

iii) Checking the results

- The fitting process used to generate Equilibrium Model parameters is a least-square regression analysis aimed at finding a global minimum of the sum of the squares of the ordinal differences between the original data from the temperature profile and those depicted by the Equilibrium Model.
- Occasionally the algorithm fails to identify the global minimum and becomes stuck in a local minimum from which it cannot extract itself. In our experience this happens rarely, say once every 20 runs, with continuous assay data, but may be more common with discontinuous assays which tend to be less accurate and have fewer data points. This will result in the generation of parameter values that do not fit the data. It is sometimes easy to identify where this has happened, as the values for these parameters are implausible (e.g., a T_{eq} of 500K).
- However, in some cases it may not be so clear, and **users should routinely verify their results by comparing the experimental results with the 3D concentration/time/temperature plot (e.g. Figure 6) generated by the derived parameters** (see below).
- It is sometimes possible to solve the problem by assigning "better" starting values for the parameters. These are currently set to an arbitrary set of default values, and for most data sets the starting values do not affect the final values. However, see comment ii) k above.

- e. However, if the conflict between these starting values and the actual values is too large, the algorithm may get trapped in a local minimum. Starting values can sometimes be estimated by comparing raw assay data to existing successful fits.
- f. Should this be difficult to carry out, zero-time model values derived from the same data set can be used as starting values (although an estimated EINACT value would need to be provided independently). The zero-time parameters should first be verified by comparing the raw and generated zero-time activity plots.
- g. It must also be noted that data sets fitted using this application that do not follow the general guidelines for collecting data set out in Peterson *et al* (2007) may also generate parameter values that do not fit the data.
- h. The most likely reason for this is inaccurate or insufficient data, or rates that are not true V_{\max} values. In our hands these problems occur most frequently with "stopped" assays, which tend to have larger errors and fewer points than continuous assays, or where it has been difficult to maintain V_{\max} over both the time span and the temperature range of the assay.
- i. Failure to maintain V_{\max} may be due to the tendency of the K_m for the substrate to increase, sometimes dramatically, at temperatures above T_{eq} . The graph below (figure 5) shows this effect in the enzyme used as the source of the example data set; this increase is quite moderate. Increases of 5 to 10-fold are not unusual.

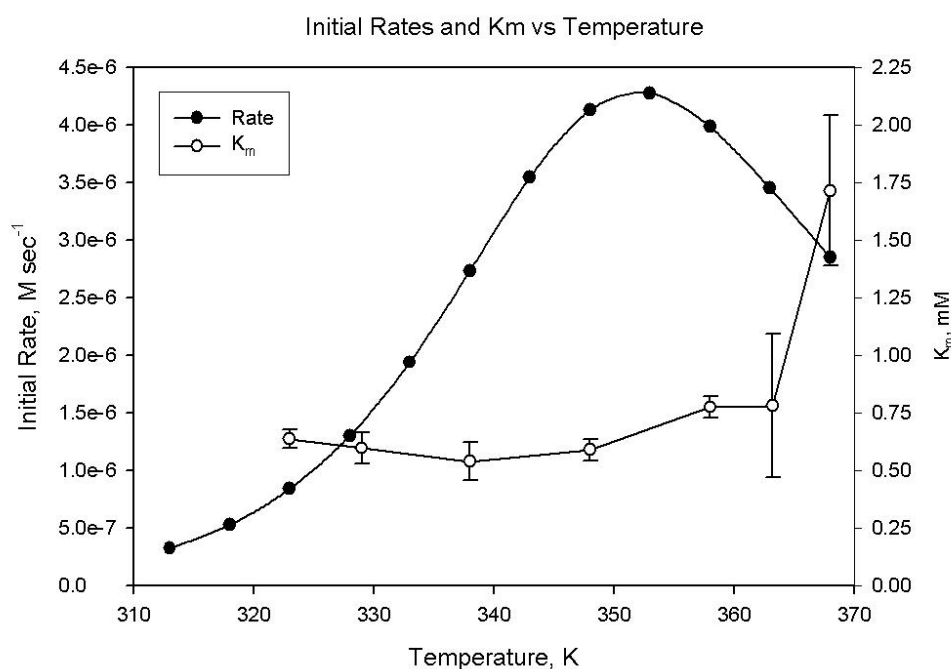


Figure 5

- j. An alternative possibility is that the enzyme does not obey the Equilibrium Model, although of the 50 or so enzymes data sets derived so far all fit the model when true V_{\max} data has been obtained. Please let us know if you find an exception!
- k. **A general method of confirming that the data matches the derived parameters is to compare the experimental product concentration results with a 3D plot generated by the parameters (see below). This comparison should be carried out with all datasets.**

Plots of the experimental data and of simulated 3D plot based on the parameters generated from the Equilibrium Model can be generated as follows:

The product concentration data from the original Matlab matrix, reformatted, can be plotted in SigmaPlot™ or similar. The Matlab data sheet needs to be reformatted into XYZ triplets. The data may be the averages of replicate runs at each temperature. The format we use for SigmaPlot™ is:

Column A: time values (unchanged)
 Column B: first temperature value copied down for same number of rows as the time values
 Column C: product concentration data corresponding to first temp value
 Column D: second temperature value, as for Column B
 Column E: product concentration data corresponding to second temperature
 Repeat Temperature / Data pairs until complete.

In SigmaPlot (or similar), plot overlaid 3D scatter plots of XYZ triplets (respectively Time, Temperature and Product Concentration Data) corresponding to columns ABD; ADE; AFG etc until complete. These plots produce the black lines (actually sets of discrete points) shown in figure 6.

Using the parameters produced by the Matlab application in the equation provided in the appendix below (Product Concentration), the function-plotting option produces a simulated data set, for comparable time and temperature ranges. From this, a 3D mesh plot can be generated. This plot can be superimposed on the experimental data plots. These two plots should show a correspondence in shape and axes.

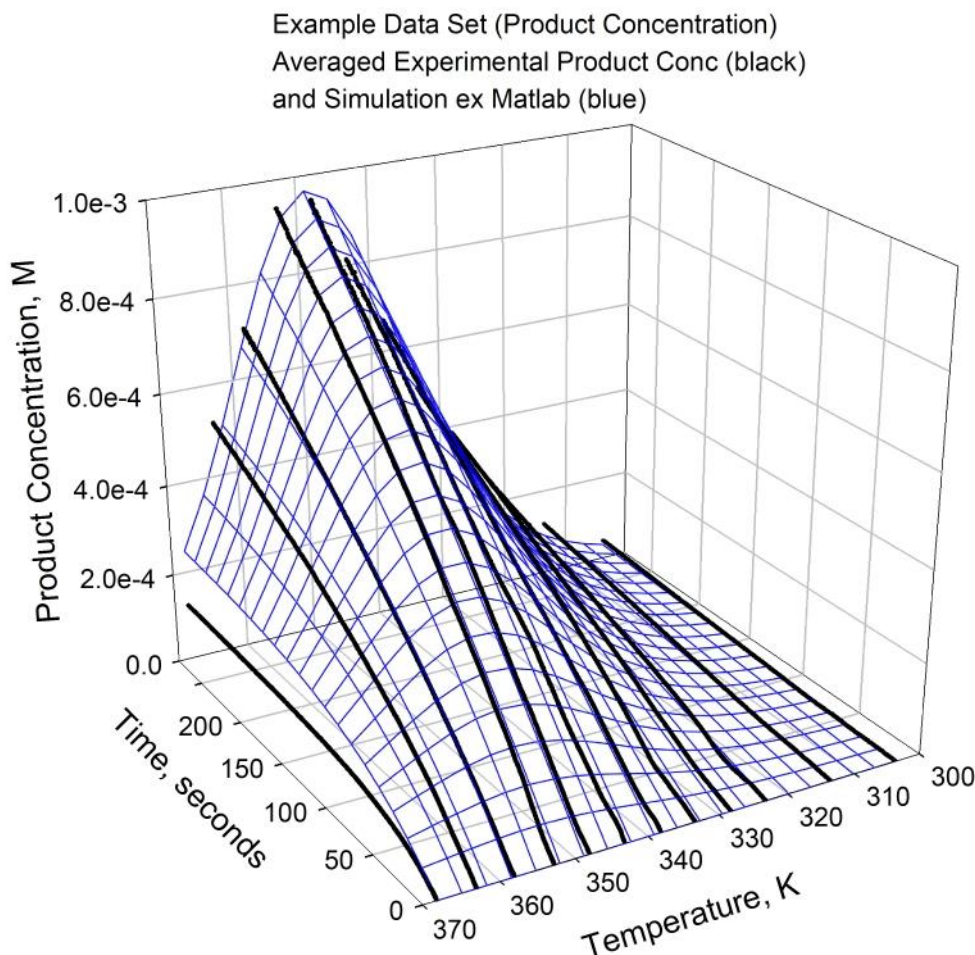


Figure 6

Determining Residual values:

Differences between experimental product concentration values, and the values predicted by the model for the same temperature and time, necessitates a simulation with temperature and time points corresponding to the experimental data. The simulation generated by the process described earlier, and plotted in figure 6, is based on uniform intervals within the temperature and time ranges, and relatively few time points, which probably will not correspond to the experimental values.

Excel functions ("Excel Product Concentration Simulation"; "Excel Residual Calculation"; and "Excel Residual Percentage Calculation" provided in the appendix below), will generate respectively the required simulation, the residual values, or the residual value as a percentage of the experimental value. The functions for calculation of the residuals includes the function for generating the simulation, so it is possible to generate a set of simulated product concentration points, or either set of residuals, or all.

The procedure is to start with a copy of the Excel file after the final run has been performed - ie the high-resolution parameter values are present in the "Settings-Output" sheet. The "Rate Data Actual" sheet can be cleared, or a new sheet inserted. The contents of the "Data" sheet is replaced by averaged replicate values, and, if desired, filtered or truncated

time values. The top row, and the first column (respectively temperatures and times) are copied from the "Data" sheet to a blank sheet. One of the functions provided below is copied into the first cell corresponding to a data value in the "Data" sheet, and then copied through the corresponding columns and rows.

This process can be repeated on a further worksheet for the other functions, if desired.

As the Matlab process averages replicated temperatures before processing, and as the process described earlier for producing an overlay of the experimental data lines and a simulated mesh plot (figure 6) is best done with averaged replicates, the Excel simulation and residuals are also produced from averaged data.

Note that the functions refer to named sheets and cells, so altering the "Data" or "Settings-Output" sheets' names, or changing the allocation of rows and columns within the sheets, will prevent the functions from running.

Once obtained, these data sets can be reformatted as described earlier, and plotted as a set of 3D scatter plots. Figure 7 shows the experimental scatter plots in black, and the corresponding residual values in red.

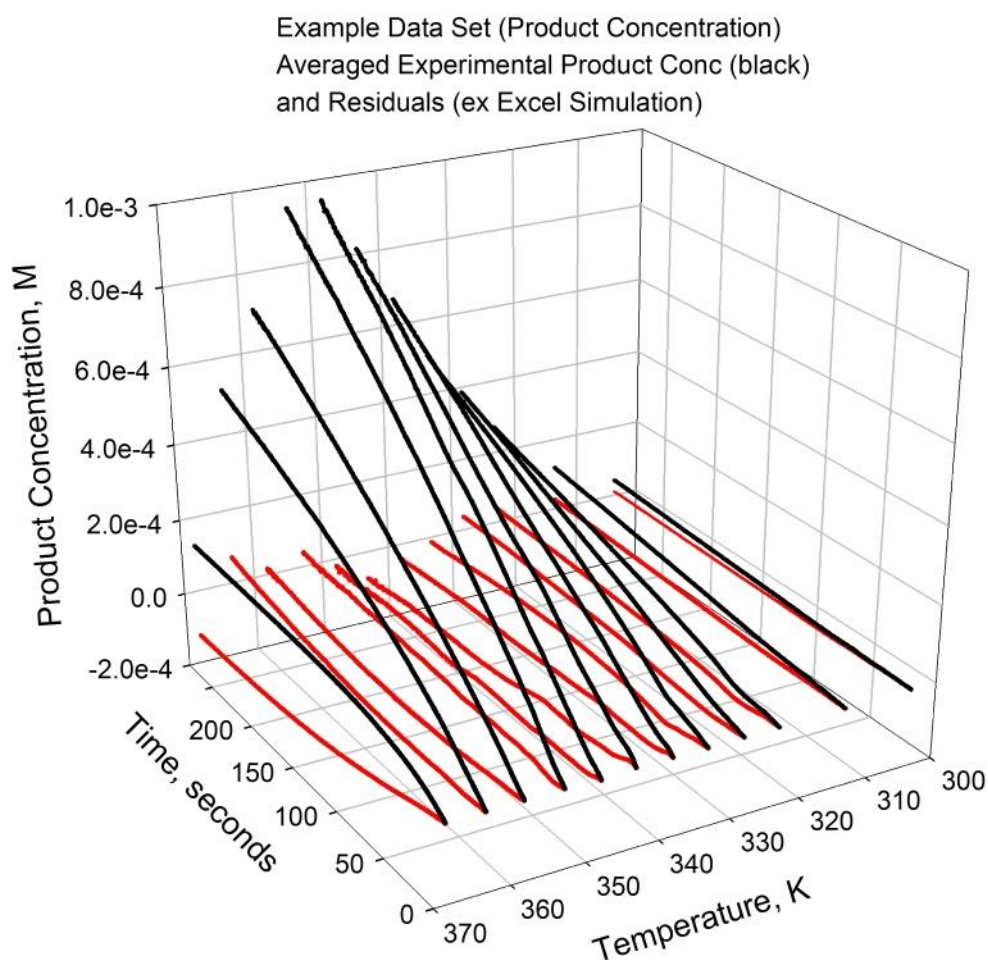


Figure 7

Other checks

Calculate the initial rates from the experimental progress curves originally fitted to the Model, then fit this data using the "initial rates" option, and compare the three parameters generated by the this fitting routine with those generated from the full data set. The three parameters generated should be close to the corresponding parameters from the full data set.

Zero-time initial rate data should calculated from raw progress curves can be plotted against temperature. An Initial Rates simulations can be generated in SigmaPlot™ (see figure 8 below), using the parameters output by Matlab (from either the

"initial rates" option, or from the full process, but without using the $\Delta G_{\text{inact}}^{\ddagger}$ value), in the "Initial Rates" equation given below.

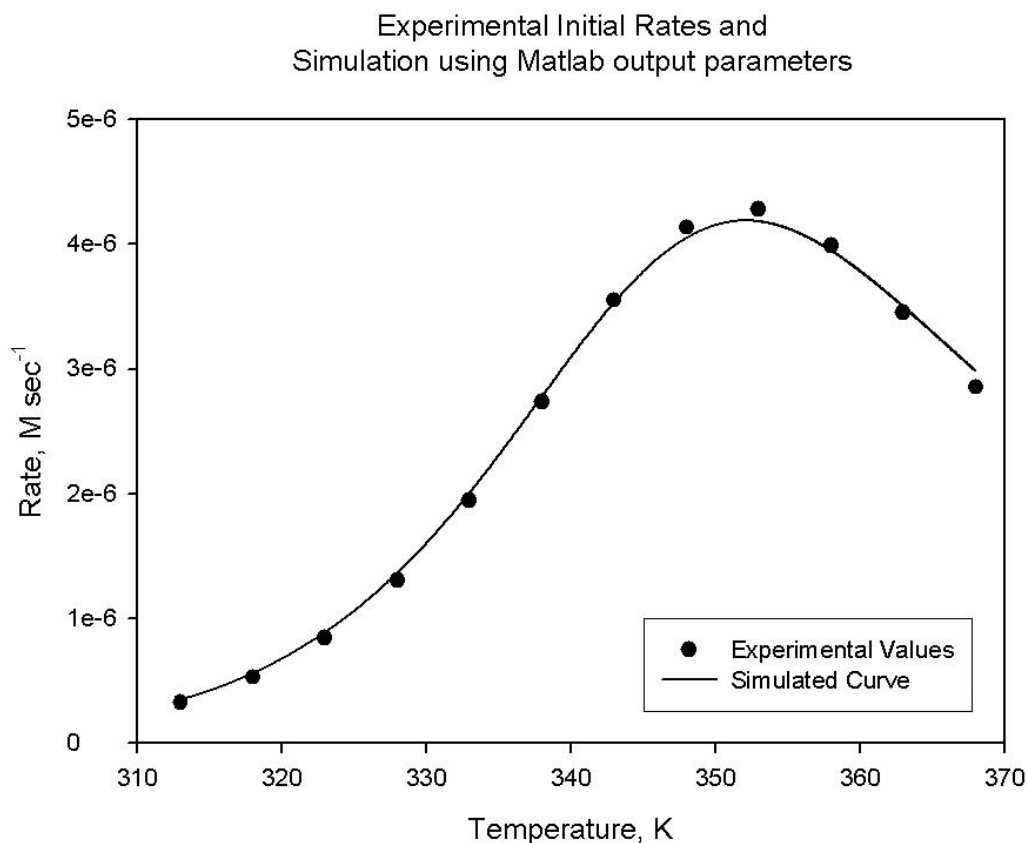


Figure 8

This plot can then be compared with the plot generated using the "initial rates" option in the fitting program (figure 1). The plots should have similar shapes and magnitude.

If they do not, it is possible that your enzyme reaction is not "ideal", or that you have a poor data set. In the former case, the three parameters from the zero time set may still be valid. A useful check may be to truncate all your data sets to shorter time periods (e.g., from 3-5 minutes to 30 seconds) and run the simulation with the truncated dataset.

Other tests, such as comparing $\Delta G_{\text{inact}}^{\ddagger}$ or $\Delta G_{\text{cat}}^{\ddagger}$ values to literature values for similar enzymes may seem useful but such values are likely to have been determined under different conditions and assumptions.

Presentation of results

The fit of data to the model can be expressed as shown in Figure 6. A more complete presentation of results can be gained by generating a plot of time vs temperature vs activity, and overlaying this onto to a similar plot generated from the 4 Equilibrium Model parameters, thus directly comparing the experimental results with the predictions of the Equilibrium Model. This necessitates smoothing the experimental data to reduce the total number of points sufficiently to allow generation of the surface (a limitation of the graphing software). Unfortunately all the 3D data smoothing algorithms we have tried result in a significant lowering of the temperature optimum "ridge", so that a comparison of the two surfaces may be somewhat misleading (compare Figure 9A with Figure 6).

One solution to this problem is to smooth each individual experimental curve (or averaged replicate curves), take a subset of the smoothed values (using about 20 time points per curve) and to generate a mesh plot of this reduced data set against time and temperature, which gives a more accurate representation of the fit of the experimental data to the model (Figure 9B).

Example Data - Product Concentration
 red: experimental data, smoothed as XYZ matrix
 blue: simulated data, using Matlab output parameters

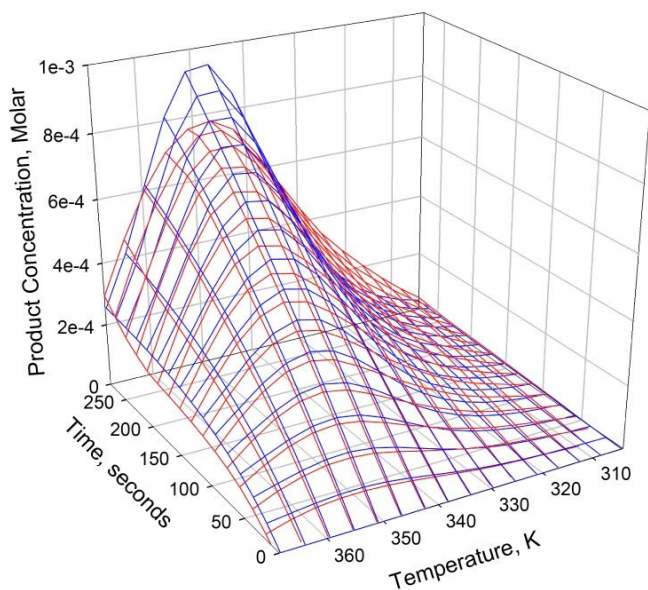


Figure 9A

Example Data - Product Concentration
 red: experimental data smoothed and subsampled
 blue: simulated data, using Matlab output parameters

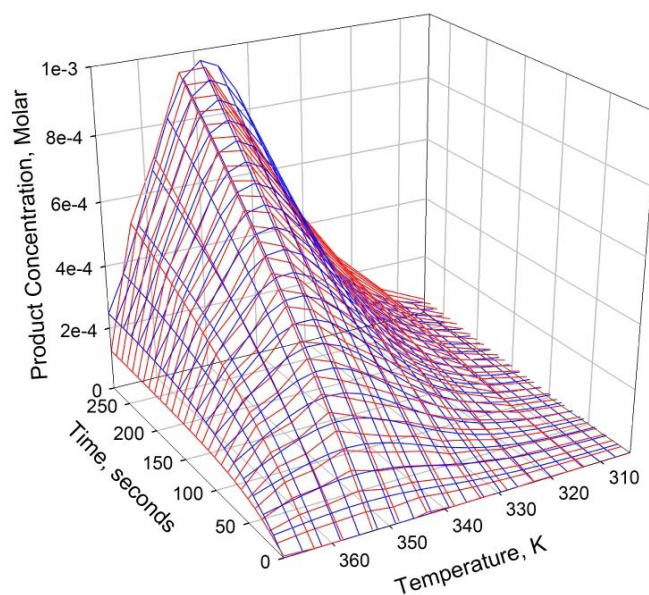


Figure 9B

A more intuitive comparison of results can be obtained by comparing the surfaces derived from Rate curves. These can be generated from the raw progress curves (product concentration). Rates are calculated from the difference between successive product concentration values, divided by the time interval. Product concentrations are Molar, and time is in seconds. As described above, the reaction rate data can, after smoothing, be plotted as a 3D mesh plot, and overlaid with a simulated data set generated from the Equilibrium Model parameters using the Progress Curves equation provided in the appendix (Figure 10A). Again, the smoothing process will cause a mismatch in the comparison with the simulated data set.

Again, a better match between the plots of the rates derived from the experimental data and the simulated data is obtained if the rates are calculated from the smoothed experimental data sets, and a subset of those points used to generate the mesh plot (Figure 10B).

Example Data - Progress Curves
 red: experimental Data, smoothed as XYZ matrix
 blue: simulated data, using Matlab output parameters

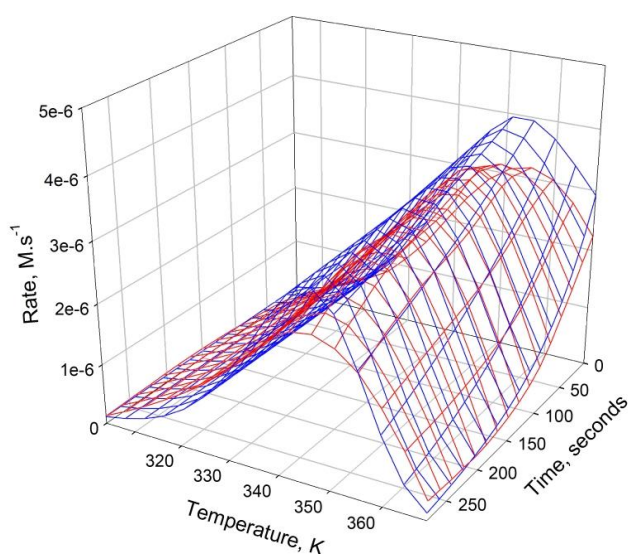


Figure 10A

Example Data - Progress Curves
 red: experimental data smoothed; rates calculated and subsampled
 blue: simulated data, using Matlab output parameters

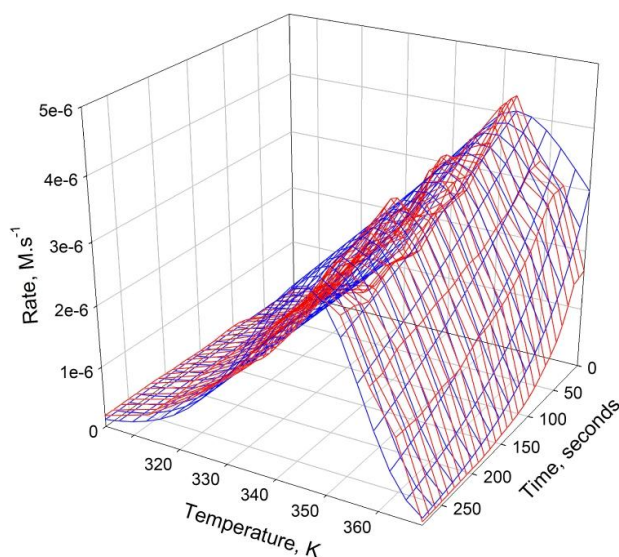


Figure 10B

APPENDIX

Equations for generating simulated data sets

These equations are provided in text format in the file **Equations for Simulations.txt**

The equation for generating the Progress Curves (Rates) simulation is:

$$((k*T*\exp(-C/(R*T))*E)*\exp(-(k*T*\exp(-I/(R*T))*\exp((Q*(1/S-1/T))/R)*t)/(h*(1+\exp(Q*(1/S-1/T)/R))))/(h*(1+\exp(Q*(1/S-1/T)/R)))$$

The equation for generating the Product Concentration simulation is:

$$-((k*T/h)*\exp(-C/(R*T))/(1+\exp((-Q/R)*(1/T-1/S)))*E/(((k*T/h)*\exp(-I/(R*T))*\exp((-Q/R)*(1/T-1/S))/(1+\exp((-Q/R)*(1/T-1/S))))) * \exp(-(((k*T/h)*\exp(-I/(R*T))*\exp((-Q/R)*(1/T-1/S))/(1+\exp((-Q/R)*(1/T-1/S))))) * t) + (k*T/h)*\exp(-C/(R*T))/(1+\exp((-Q/R)*(1/T-1/S)))*E/(((k*T/h)*\exp(-I/(R*T))*\exp((-Q/R)*(1/T-1/S))/(1+\exp((-Q/R)*(1/T-1/S)))))$$

The equation for generating the Initial Rates simulation is:

$$(k*T/h)*\exp(-C/(R*T))*E/(1+\exp((-Q/R)*(1/T-1/S)))$$

The equation for generating a Progress Curve (Rates) simulation according to the "Classical Model" is:

$$(((k*T)/h)*\exp(-(C/(R*T))))*(E)*\exp(-(((k*T)/h)*\exp(-I/(R*T)))*t)$$

Parameters to be applied in these equations are:

k = 1.3806503E-23 (Boltzmann constant)

R = 8.314472 (Molar gas constant)

I = EAI = $\Delta G_{inact}^\ddagger$

Q = HEQ = ΔH_{eq}

h = 6.62606876E-34 (Planck constant)

E = E0 = Enzyme Concentration, Molar

C = EACAT = ΔG_{cat}^\ddagger

S = TEQ = T_{eq}

T and t are respectively temperature and time; ranges and intervals are specified as variables in SigmaPlot™.

Time and $\Delta G_{inact}^\ddagger$ are not used in the Initial Rates simulations. The variable names above are changed from those used in the Excel template file, as SigmaPlot™ requires single-character identifiers for the terms.

The Excel functions used to generate the point-to-point simulations, and to derive the residuals, are:

Excel Product Concentration Simulation is:

=(-(1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$6/(8.32*Data!B\$1))*'Settings-Output'!\$C\$4/(((1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$5/(8.32*Data!B\$1))*exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8)))))*exp(-((1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$5/(8.32*Data!B\$1))*exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8)))/(1+exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8))))*Data!A2)+(1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$6/(8.32*Data!B\$1))*'Settings-Output'!\$C\$4/(((1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$5/(8.32*Data!B\$1))*exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8))))))

Excel Residual Calculation (= Experimental - Simulation):

=Data!B2-(-(1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$6/(8.32*Data!B\$1))*'Settings-Output'!\$C\$4/(((1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$5/(8.32*Data!B\$1))*exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8)))))*exp(-((1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$5/(8.32*Data!B\$1))*exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8)))/(1+exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8))))*Data!A2)+(1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$6/(8.32*Data!B\$1))*'Settings-Output'!\$C\$4/(((1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$5/(8.32*Data!B\$1))*exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8))))))

Excel Residual Percentage Calculation (= Experimental - Simulation):

=((Data!B2-(-(1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$6/(8.32*Data!B\$1))*'Settings-Output'!\$C\$4/(((1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$5/(8.32*Data!B\$1))*exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8)))))*exp(-((1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$5/(8.32*Data!B\$1))*exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8)))/(1+exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8))))*Data!A2)+(1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$6/(8.32*Data!B\$1))*'Settings-Output'!\$C\$4/(((1.38E-23*Data!B\$1/6.63E-34)*exp(-'Settings-Output'!\$E\$5/(8.32*Data!B\$1))*exp((- 'Settings-Output'!\$E\$7/8.32)*(1/Data!B\$1-1/'Settings-Output'!\$E\$8))))))/Data!B2)*100

The source code for the Matlab Application has also been provided, in plain text format, in the folder named "Matlab Code"