

Fitting (Optimization)

Q: When and how should I use Bayesian fitting?

A: The SAAM II Bayesian feature allows the incorporation of prior knowledge of the model parameters into the modeling of the kinetic data. The additional information is entered as a mean and standard deviation for one or more of the parameters in the model. The values can come from previous individual experiments, analysis of a population, or from published results. The prior knowledge must be justifiable independently of the model.

In both the SAAM II Compartmental and Numerical applications, the use of Bayesian estimation results in an additional factor being included in the objective function during optimization (fitting). The final fitted value of any Bayesian parameter is thus influenced by the population mean and standard deviation used.

For example, suppose a model contains parameters $k(1,2)$, $k(2,1)$ and $k(0,1)$, but the data is not rich enough to allow reliable estimation of fitted values. If examples could be found in literature where $k(0,1)$ had been measured before, and values for $k(0,1)$ and a measure of its variability were provided, this information, expressed in terms of the mean and standard deviation, could be entered into the Compartmental model as the Bayesian mean and SD for parameter $k(0,1)$. The final fit will combine the information from the kinetic data and the prior knowledge on $k(0,1)$. In all cases, the influence of the prior knowledge on the final estimates and the results must be carefully assessed.

Q: How do I set initial conditions for a sum of exponentials?

A: We will use two simple Case Studies to explain this. For this example, remember that the relationship between a half-life and the governing exponential is:

$$a = \ln(2)/\text{half-life}$$

1 - Monoexponential decay - bolus injection protocol

This case study will describe how to obtain initial estimates for the coefficient "A" and exponential "a" of the simple monoexponential model

$$y(t) = A \cdot \exp(-a \cdot t)$$

You can plot the data in semi-logarithmic mode using SAAM II. You can either print a copy of the plot to continue this case study, or continue the next steps using the plot that appears on your screen (i.e. you can do the following steps by approximation using what is on the screen as opposed to formally working with a hard copy of the plot).

Fitting (Optimization)

1. Draw a "best" straight line through the monoexponentially decaying data.
2. The point at which your "best" line intersects the ordinate is an estimate for "A"; estimate A.
3. Calculate the half-time for an arbitrary point on the line. For example, a data value is 2,000 and it occurs at approximately 53 minutes. One half of 2000 is 1000 that occurs at 133 minutes. The half-time, that is, the time it takes to decay from 2000 to 1000, is thus about 80minutes. An estimate for "a" can thus be obtained.
4. Open the parameter dialog box, and enter your initial estimates for "A" and "a".
5. Solve and plot your results. 6. Fit to the data.

2 - Case Study: Biexponential decay - bolus injection protocol

This case study will describe how to obtain initial estimates for the coefficients A1 and A2 and the exponentials a1 and a2 of the biexponential model

$$y(t) = A1 \cdot \exp(-a1 \cdot t) + A2 \cdot \exp(-a2 \cdot t)$$

You can plot the data in the semi-logarithmic mode using SAAM II. You can either print a copy of the plot to continue this case study, or continue the next steps using the plot that appears on your screen.

1. Draw a "best" straight line (L1) through the monoexponentially decaying tail portion of the data.
2. The point at which this line intersects the ordinate provides an estimate for A2; estimate this value from your plot.
3. Calculate the half-time for an arbitrary point on this line. For example, the point 1600 that occurs at approximately 65 minutes has been chosen. One half of 1600 is 800 that occurs at approximately 135minutes. The half-time is approximately 70 minutes. Now you can obtain an estimate for a2, the smaller of the two exponentials.
4. Draw a line through the initial decaying portion of the data (L2).

Fitting (Optimization)

5. The point at which this line intersects the ordinate provides an estimate for "A1 +A2". An estimate for A1 can be obtained by subtracting your estimate for A2 from this value. Estimate A1.
6. Calculate the half-time for an arbitrary point on this line. For example, the point 4000 that occurs at approximately 2 minutes has been chosen. One half of 4000 is 2000 that occurs at approximately 22minutes. The half-time is approximately 20 minutes. We can obtain an estimate for a1, the larger of the two exponentials.
7. Open the parameter dialog box, and enter your estimates for A1, A2, and a1and a2.
8. Solve, and plot your results.
9. Fit to the data.

Q: How do I analyze multiple data sets?

A: Analyzing multiple data sets is a necessity for many experimental designs. SAAM II allows the user to fit a model to more than one data set at the same time, allowing the option to estimate a proportionality factor (different for each data set) for the measurement error. See the links

- [What is the difference between absolute and relative weighting?](#)

for more information about this subject.

Q: How can I calculate Statistics if SAAM II has not converged?

A: You might want to do this to have an idea of the local a posteriori identifiability properties of the model you are investigating. Statistical information is not available if the fitting has reached the maximum allowed number of iterations or if one or more parameters have reached their upper and lower limits (either in the subproblem or have actually hit them).

To calculate the Covariance Matrix and obtain all the statistical information even if the program has not converged, set the maximum number of iterations in the Computational Settings Window to zero. The program will halt after a single fitting run, the parameter values will be unchanged and the complete statistical information (Confidence Limits, Covariance and Correlation Matrix and Objective Function) will be made available in the Statistics Window.

Fitting (Optimization)

Note that, even when using this approach, if a parameter (or more than one) has hit the Upper or Lower Limit, then its Statistics will not be available, and the other Statistics in the window will be evaluated keeping that parameter fixed to its current value and assuming for it a Standard Deviation of zero. Usually, this happens because the Confidence Limits on that parameter are very wide and the parameter estimate is therefore completely unreliable. Should you want the Statistics on that parameter anyway, and therefore have access to the complete statistical information, then its Upper and Lower Limits need to be relaxed (i.e. made wider, so that the parameter does not hit the Limits in the subproblem). In almost the totality of cases, if the Lower Limit is made mildly negative and the number of iterations is set to zero, the complete covariance matrix will be calculated and made available. You should, however, keep in mind that this kind of statistical information is for qualitative purposes only, and full convergence should be reached before putting any faith in the fitting results.

See also:

1. What should I do when the covariance matrix is unreliable?
2. What should I do when the fit goes on forever?
3. When and how should I use Bayesian fitting?

for more information on how to deal with this situation.

Q: How can I make the Area Under the Curve a fitted variable?

A: You can calculate the Area Under the Curve for any generic function of the parameters or of the compartmental masses that can be associated to a sample, such as in

$$s1 = q1^2 + q2/5 + \alpha$$

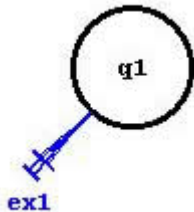
where alpha is an adjustable parameter and q1 and q2 are the masses in compartment 1 and 2 respectively. The Area Under the Curve can then be calculated by checking "Calculate AUCs" in the Settings window and can be viewed using the "AUCs" option in the "Show" menu.

One could conceive of a situation where the AUC for a generic function is an optimized variable. This is not currently possible using the "Calculate AUCs" option in SAAM II. However, it can be done anyway by exploiting some existing SAAM II features.

Suppose the user wants to calculate the AUC of s1 given above. The following steps show how:

Fitting (Optimization)

1. Draw a compartment with a single input as in this figure:



2. Define the equation describing $ex1$ to be equal to $s1$. From standard differential equation theory, it is apparent that the calculated value of $q1$ will be the value of AUC of $s1$ at any time.
3. Associate $q1$ to your value of AUC at the given time. Be sure to use Absolute Weighting (remember that Relative weighting is not appropriate when a data set is made of a single data point) #
4. Fit the model. The kinetic data and the AUC will be taken to be variables to be fitted at the same time

Q: How is the objective function calculated?

A: To produce a best fit, SAAM II minimizes an objective function based on maximum likelihood. This objective function allows for minimization of the model and the error associated with the data. The objective function is described on p. 147 of the SAAM II manual.

Q: What should I do when the covariance matrix is unreliable?

A: This could happen for various reasons. The most common is because the data are not informative enough to allow reliable estimation of the model parameters. The best way to solve this problem is to simplify the model of the system or redesign the kinetic experiment and collect additional data.

<!--[if !supportEmptyParas]--> <!--[endif]-->

Sometimes this could also happen because the error is misspecified in magnitude (e.g. Absolute Weighting is used and a Fractional Standard Deviation of 20% instead of 3% is

Fitting (Optimization)

given). The use of Relative Weighting should give a better understanding of the true magnitude of the error.

Sometimes the following error message appears - in this example, concerning k(0,2):

WARNING: The following parameter limit(s) constrain further optimization:

k(0,2) hit lower limit

Usually, the parameter actually hits the limit. This problem can be easily solved by resetting the Upper and Lower Limits for that parameter.

However, this could occur even if **the parameter is not actually at limits** (by inspection of the Parameter Window). When this happens, the most likely reason is that the parameter(s) that hit the limits are not estimable from the data. The Statistics Window will contain the standard deviations of the parameters calculated with the parameters at limits held fixed. Possible suggestions to avoid this problem are to actually fix them in the fitting or to make them Bayesian parameters.

Please see:

1. What should I do when the fit goes on forever?
2. When and how should I use Bayesian fitting?
3. How can I calculate Statistics if SAAM II has not converged?

for more information on how to deal with this situation.

Q: What is the difference between absolute and relative weighting ?

A: When you choose Absolute Weighting, the underlying assumption is that the weights (i.e., the error affecting the data) are exactly known. If the heading of the data file is, for example:

(FSD 0.1)

the Fractional Standard Deviation of the data will be taken to be exactly 10%. This means that, if the value of a datum is 150, it will be taken to be 150, plus or minus 15.

Fitting (Optimization)

if, on the other hand, you choose **Relative Weighting**, then the Fractional Standard Deviation of the data will *not* be taken equal to 10%. A proportionality constant will be estimated from the data themselves, and reported as the Scaled Data Variance in the Statistics Window. This type of weighting is appropriate if:

1. the error in the data is not known exactly, but its structure (e.g. constant Standard Deviation, constant Fractional Standard Deviation) is known. Estimating the proportionality constant provides an extra degree of flexibility;
2. multiple data sets are being analyzed, and the user is uncertain about their error relative to one another.

The final FSD of the data might be larger or smaller than 10%, but it will reflect what the optimization algorithm estimates to be the error in the data, given the model.

Please see the manual and the on-line help for more details.

Q: What is the difference between absolute and relative weighting ?

A: When you choose Absolute Weighting, the underlying assumption is that the weights (i.e., the error affecting the data) are exactly known. If the heading of the data file is, for example:

(FSD 0.1)

the Fractional Standard Deviation of the data will be taken to be exactly 10%. This means that, if the value of a datum is 150, it will be taken to be 150, plus or minus 15.

if, on the other hand, you choose **Relative Weighting**, then the Fractional Standard Deviation of the data will *not* be taken equal to 10%. A proportionality constant will be estimated from the data themselves, and reported as the Scaled Data Variance in the Statistics Window. This type of weighting is appropriate if:

1. the error in the data is not known exactly, but its structure (e.g. constant Standard Deviation, constant Fractional Standard Deviation) is known. Estimating the proportionality constant provides an extra degree of flexibility;
2. multiple data sets are being analyzed, and the user is uncertain about their error relative to one another.

Fitting (Optimization)

The final FSD of the data might be larger or smaller than 10%, but it will reflect what the optimization algorithm estimates to be the error in the data, given the model.

Please see the manual and the on-line help for more details.

Q: What is the right model to use?

A: There is no such thing as the "right" model. The "right model" comes from a combination of goodness of fit and model structure.

The goal is to find the simplest model that explains the experimental data. The Akaike Information Criterion (AIC) and the Schwarz-Bayesian Information Criterion (BIC) are "parsimony criteria" useful to perform model comparison and discrimination, e.g. decide whether a sum of two or three exponentials, or a model with two or three compartments better fit the available data.

To this extent, AIC and BIC are a function both of the goodness of fit (given by the objective function $L(p^*, u^*)$), the number of adjustable model parameters P and variance parameters Q , and the total number of data points N . Between two or more rival models, the model with the lowest AIC or BIC (that is, the model that better explains the data with the least number of parameters) is the "best" one.

For example, see Carson, Cobelli and Finkelstein, *The Mathematical Modeling of Metabolic and Endocrine Systems*, New York: John Wiley, 1983, p.251 for a case study. Note that AIC and BIC are calculated in a slightly different fashion.

See also:

1. What are the AIC and BIC?

for more information about this subject.

Q: How can I analyze an algebraic model in the Compartmental application?

A: Although the Compartmental application is designed for complex compartmental models, it can handle algebraic models as well. They must, however, be constructed within the experimental framework used in the Compartmental application. A simple example is to construct the model for a straight line and fit it to data. Follow these steps:

1. Start the **Compartmental** application.

Fitting (Optimization)

2. Click **Experiment Attributes...** on the **Set** menu. In the Experiment Attributes box, enter **x** for the **Independent Variable** name and **10.0** for the **End At** value. Click the **Done** button to close the box.
3. An experiment must be created in order to use the computational machinery in the Compartmental application. Click the **Experiment** button in the **Toolbox**. Use the default values in the **Create Experiment** box (rename the experiment if desired), and click the **Create** button.
4. A sample object is the link between the function that will be defined and the data to which it will be fitted. To create a sample, click on the **Sample** tool in the **Toolbox**, and then click anywhere in the main model (white) window. The sample should appear as a red circle labeled **s1** as shown in Figure 1.

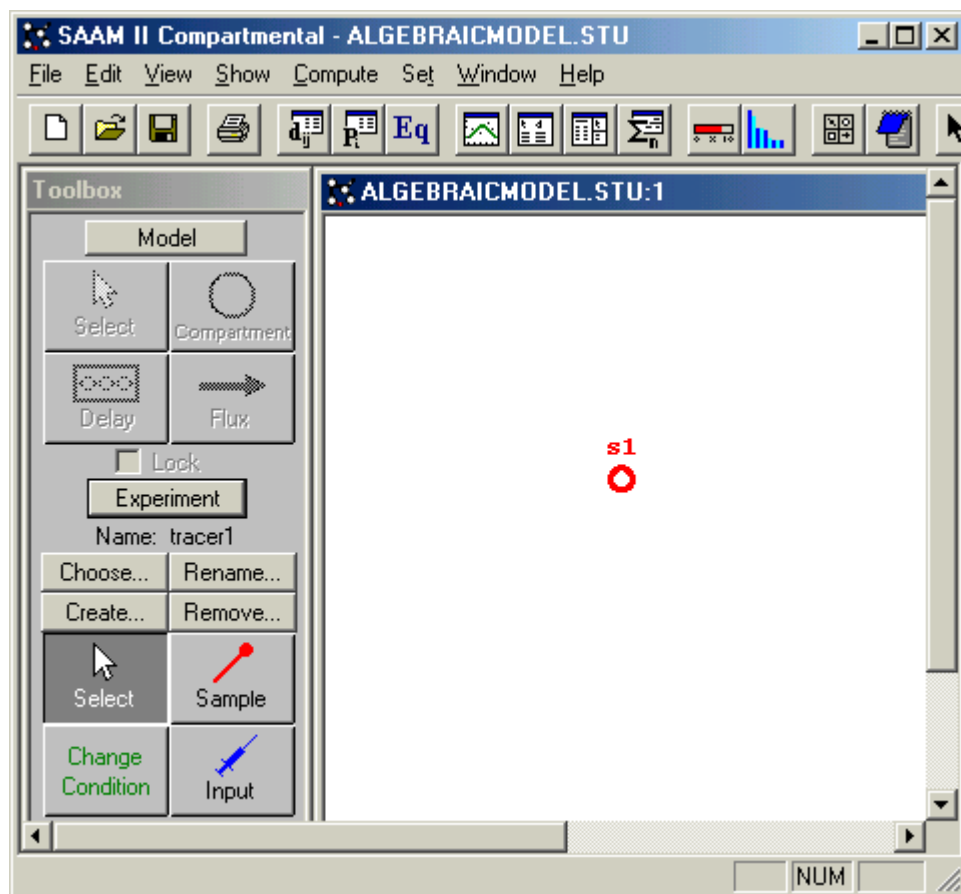


Figure 1

Fitting (Optimization)

- Double-click on the sample object to open the **Sample Attributes** box. The name for the data could be entered in the **Associate with Data Name** area now, but for this example, it will be done after the data has been typed in. Click in the **Equations** area, enter the equation shown below and then click the **Done** button.

$$s1 = m * x + b$$

- Open the **Data** window by clicking **Data** on the **Show** menu. Type in the data as shown in Figure 2. To check that the format is satisfactory, click **Check Data Format** on the **Edit** menu. If a mistake has been made an error message will appear; otherwise a **Data Format is okay** message will appear at the bottom of the Data window. Click the **X** in the upper right corner of the Data window to close it (or click **Close** on the **File** menu).

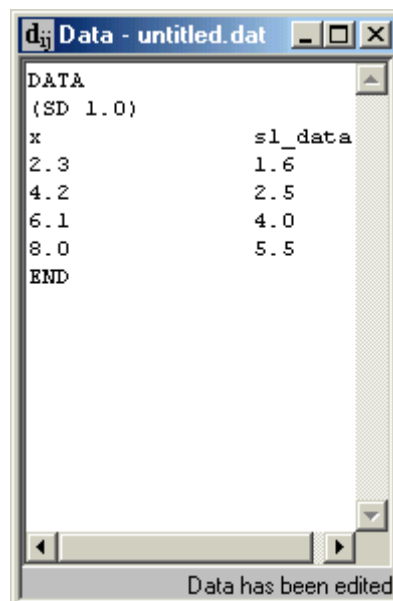


Figure 2

- To inform the application that the data labeled *s1_data* is to be associated with the sample *s1*, click **Associations** on the **Show** menu to open the **Associate Samples and Data** dialog box. In the box, *s1* (in the left pane) and *s1_data* (in the right pane) should already be highlighted. Click the **Associate** button. The entry in the left pane will change to *s1 : s1_data* to indicate that the sample and data are now associated. Click the **Done** button to close the box. Notice that the sample object is now solid red, which indicates that *s1* is now associated with data.

Fitting (Optimization)

8. Before the model can be fitted to the data, approximate values are required for the parameters. Open the Parameter window by clicking **Parameters** on the **Show** menu, or by clicking the **Parameters** toolbar button. Notice that m and b are listed as parameters. Parameters are defined as any undefined variables in the model, i.e. they appear only on the right of the equal sign in one or more equations. In this case, m and b appear in the equation for $s1$. Approximate values and upper and lower limits are required for each parameter to constrain the n-dimensional space ($n = 2$ here) that the optimizer must search for the values that minimize the objective function, i.e. create the best fit of the model to the data. Since the y-intercept, b , is already highlighted, enter 0.0 for the **Value**, -1.0 for the **Low Limit**, 1.0 for the **High Limit**, and then click the **Save** button. Next, double-click the line in the list containing m to place it into the edit area at the bottom of the box. Enter 1.0 for the **Value** and click the **Save** button. Notice that default values are supplied for the upper and lower limits. Click the **Done** button to close the box.
9. All information required to fit the model to the data has now been entered. Click **Fit** on the **Compute** menu, or click the **Fit** toolbar button (looks like blue bar graph).
10. Reopen the Parameter window. Notice that the values for both m and b have been adjusted to provide the best fit of the model to the data. Close the Parameter window.
11. To view a plot of the fitted line, click **Plot** on the **Show** menu, or click the **Plot** toolbar button (looks like a small plot). In the left pane, select the $s1 : s1_data$ entry, and click the **Done** button. You should see the plot in Figure 3.

Fitting (Optimization)

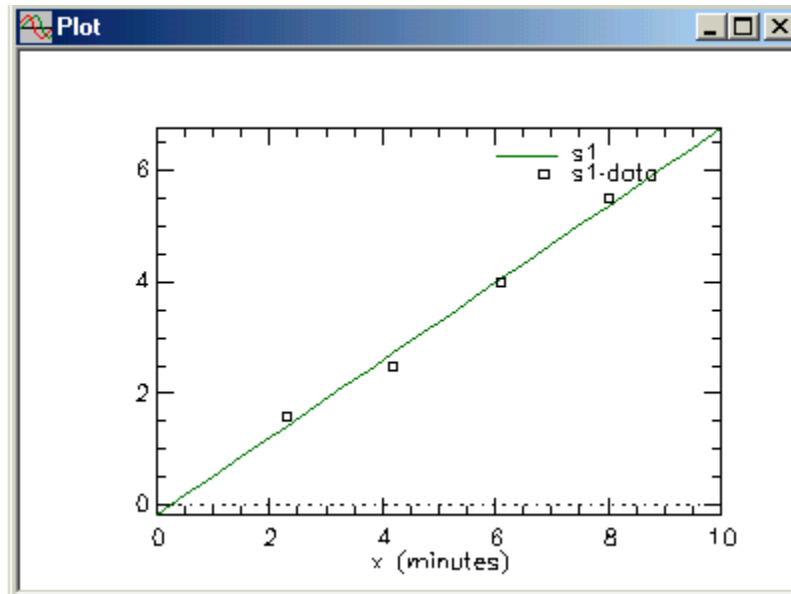


Figure 3

12. When finished, close the **Plot** box, save the model (if desired), and exit the Compartmental Application.

Q: What should I do when the fit goes on forever?

A: The reason for this is that the optimization algorithm used in the program has trouble converging, given the model and the data. Usually, this happens because the data are not informative enough and/or the model is too complex to guarantee identifiability.

Possible solutions are:

1. Check the *a priori* identifiability of your model: it might be that your model is not *a priori* identifiable from data, that is, the solution for the unknown parameters of the model is not unique, given the model structure and the experimental design. For a formal definition and more information about *a priori* identifiability and software tools to check for it. E.g, see S. Audoly, L. D'Angio', M. P. Saccomani and C. Cobelli, "Global Identifiability of Linear Compartmental Models - A Computer Algebra Algorithm", *IEEE Transactions on Biomedical Engineering*, 45: 1, January 1998.
2. If you have some additional information about the parameters, make them Bayesian parameters by giving SAAM II their Population Mean and Standard Deviation.

Fitting (Optimization)

3. Simplify the model structure, deleting nonaccessible compartments and creating links between the parameters when appropriate.
4. Redesign the kinetic experiment and collect additional data, then attempt again to fit your model.

See also:

1. What should I do when the covariance matrix is unreliable?
2. When and how should I use Bayesian fitting?

for more information on how to deal with this situation.

Q: How do I implement time-dependent conditionals?

A: Conditional statements traditionally provide a mechanism for switching an execution path or the value assigned to one or more variables. In many programming languages the conditional takes the form of an if statement. In SAAM II, the conditional is combined with the syntax for a general assignment statement like:

```
x = if (variable > 0.0) then 1.0 else 2.0
```

This acts like a switch such that the first value is assigned to x if the condition is true, and the second value is assigned to x if it's false.

In SAAM II there are two types of conditionals defined as explicit and implicit. An explicit conditional is one where the time during the experiment at which the switching action is to take place can be predicted in advance. An implicit conditional is one where the time is unknown until the condition itself is evaluated. An example of each type would be:

Explicit: $x = \text{if } (t \geq 10.0) \text{ then } 1.0 \text{ else } 2.0$

Implicit: $x = \text{if } (q1 < q2) \text{ then } 1.0 \text{ else } 2.0$

In the explicit case, the application can predict in advance that the value of x will change at time $t = 10.0$. In the implicit case, however, there is no way to predict in advance when the value of $q1$ will be less than $q2$. This distinction is critically important in the SAAM II Compartmental application because of the integration that occurs during a *Solve* or *Fit*. In order for the integrator to calculate correct results, the function upon which it is operating must be both differentiable and continuous. Discontinuities within an experiment are acceptable if the time at which they occur is known in advance so that the integrator can be

Fitting (Optimization)

reset and restarted. For this reason, *implicit* conditionals are not allowed in the Compartmental application.

Explicit conditionals are allowed in the Compartmental application, but are implemented as Change Conditions to provide ease of entry and added flexibility. In a Change Condition, the time or times at which the change is to be active are specified along with an equation, such as $x = 1.0$. This equation remains active for the duration of the specified time period (see the SAAM II User Guide for more information on Change Conditions).

There are no Change Conditions in the Numerical application, but the *explicit* form of the *if* statement is allowed. The conditional within the statement may only contain the independent variable (usually time). The *explicit* example shown above would be permitted in the Numerical application.

There is a workaround for the implicit conditional problem, however, that involves using an approximation for the switch that is both continuous and differentiable, known as a Heaviside function. One approximation uses the normalized arctangent (one of the built-in functions of SAAM II):

$$\text{heaviside} = 0.5 * (1.0 + \text{atan}(\text{lambda} * (x - \text{xlag})) * 2.0 / 3.141592653)$$

In this implementation the $(x - \text{xlag})$ expression acts as the switch. When it's positive, the value of heaviside will be approximately 1.0; when it's negative, heaviside will be approximately 0.0. Lambda controls the sharpness of the value of heaviside, i.e. how quickly the value switches, and how closely to 1.0 and 0.0 that values become. Lambda should be tuned during each use to work best with the magnitude of $(x - \text{xlag})$.

This approximation can be used in SAAM II in the following manner:

$$y = \text{heaviside} * \text{expression1} + (1.0 - \text{heaviside}) * \text{expression2}$$

When $(x - \text{xlag})$ is positive, the value of heaviside will be approximately 1.0, and y will be set equal to expression1. When $(x - \text{xlag})$ becomes negative, the value of heaviside will become approximately 0.0, and y will be set equal to expression2. Expression1 and expression2 can be a single value or an expression like $(q1 / \text{vol})$.

Fitting (Optimization)

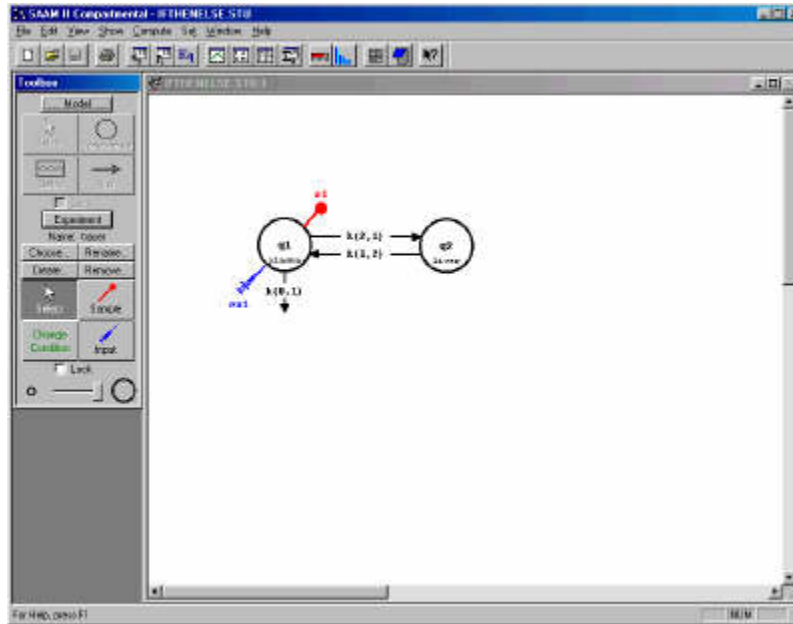


Figure 1

As an example, supposing the exogenous input in the model in Figure 1 is an infusion of magnitude 5000.0 that is active throughout the experiment, but it must be turned on or off in order to maintain the concentration in compartment $q1$ at a value of $1.0e5$. There is no way of predicting in advance when and how many times this will occur. To implement this switch, write the following equations in the **Equations** window:

```

ambda = 1.0e8
condition = 1.0e5 - q1
heaviside = 0.5*(1.0 + atan(lambda * (condition))) * 2.0 / 3.141592653
input = heaviside * 5000.0
    
```

In this example, *condition* will be positive, *heaviside* will be approximately 1.0, and *input* will be 5000.0 whenever $q1$ is less than $1.0e5$. Note that the second part of the approximation containing expression2 is not required in this case since the value for the infusion input will be 0.0 (off) whenever $q1$ exceeds $1.0e5$. The only remaining task is to open the **Exogenous Input** box and create the infusion by selecting the **Input Type equation**, setting the duration to the entire experiment, and entering the equation as $ex1 = input$.

When the model is solved, the contents of $q1$ will increase to $1.0e5$, and then the infusion will switch on and off several times to maintain the concentration in $q1$ at approximately that level. A sample output is shown in Figure 2.

Fitting (Optimization)

time	q1
0.000	0.000
13.333	47824.792
26.667	78151.623
40.000	100614.642
53.333	99976.623
66.667	99986.861
80.000	99991.485
93.333	99972.798
106.667	100010.337
120.000	99995.190
133.333	99999.426
146.667	99993.329
160.000	100031.439
173.333	99998.374
186.667	99998.608
200.000	99998.159
213.333	100031.822
226.667	99998.387
240.000	99998.810
253.333	99997.929
266.667	100030.633
280.000	100010.660
293.333	99996.686
306.667	99997.507
320.000	99995.663
333.333	99998.935
346.667	99998.506
360.000	99997.902
373.333	99996.177
386.667	99996.542
399.333	99996.764
400.000	100038.240

Figure 2

This type of approximation can be safely used anywhere within SAAM II when a conditional switch is required.

As an aside, the SAAM II *if* statement shown at the top is fully implemented in both the Compartmental and Numerical applications but its full functionality is turned off to prevent it from being used improperly in implicit cases where it will work but will produce incorrect results.

Q: What's the easiest way to specify covariates in study files?

A: When there are a number of subjects being analyzed using the same basic model, certain attributes may vary among the subjects, such as weight. An exogenous input, for example, might be specified as the amount of the bolus divided by the weight of the subject. Assuming that the magnitude of the bolus is constant, the least efficient method of

Fitting (Optimization)

specifying the input would be to actually calculate the value of *bolus / weight* for each subject and to enter this value into the **Exogenous Input** window in each respective study file.

A slightly better method would be to define the input by the equation, $ex1 = 1.4e6 / weight$, in the **Exogenous Input** window, and then write an equation for each subject, such as $weight = 43.6$, in the **Equations** window. While this is an improvement, it still stores subject information in both the Model and Data.

The best method is to define the input by equation in the **Exogenous Input** window, such as $ex1 = 1.4e6 / weight$, but to use the data constant feature in SAAM II for the weight. As explained in the Manual and Help, a data constant has the same effect as declaring one value for a data element at all experiment times. For this example, a separate line would be added to the data like:

```
CONST weight 43.6
```

This is also equivalent to using the equation, $weight = 43.6$, in the model. The advantage to this method is that it allows the model to remain the same for each subject and relegates the differences to the data. This makes entering the specific information for each subject faster and more efficient.

Note that the amount of the bolus could also be stored as a data constant if it varies among the subjects.

Q: How can I get smoother curves on my plots?

A: The plot detail depends on the number and location of the times at which the solution is calculated. The solution times, in turn, are dependent upon the data points, and the number of calculation intervals specified. The first rule is that every time listed in the Data window creates a solution point. The second rule is that the interval between solution points is never larger than the length of the experiment divided by the number of calculation intervals specified. If, for example, the experiment is 100 minutes long and the calculation intervals setting is 20, the application insures that the solution times are never more than $100 / 20 = 5$ minutes apart. If there are any gaps in the data larger than 5 minutes, the application will insert sufficient solution points to reduce the gap to 5 minutes or less.

The calculation intervals are set at the top of the **Computational Settings** window (to open it click **Settings** on the **Compute** menu). Notice that the value may be set anywhere from 1 to 200. Using the maximum setting provides the greatest amount of detail but also requires

Fitting (Optimization)

the most time for computations. For smaller models, the additional time required to Solve or Fit may be insignificant, but for larger models it may be significant.

When one or more variables are plotted in SAAM II, the curve is drawn by connecting adjacent solution values via straight line segments. If the slope changes gradually or the points are very close together, the curve will appear quite smooth. If, however, the slope is changing rapidly and/or the points are farther apart, the curve may look jagged. Figure 1 shows a plot where the calculations intervals setting is one, so the only solution points are at the data points. Notice that the curve is very rough since the data points are not very close together.

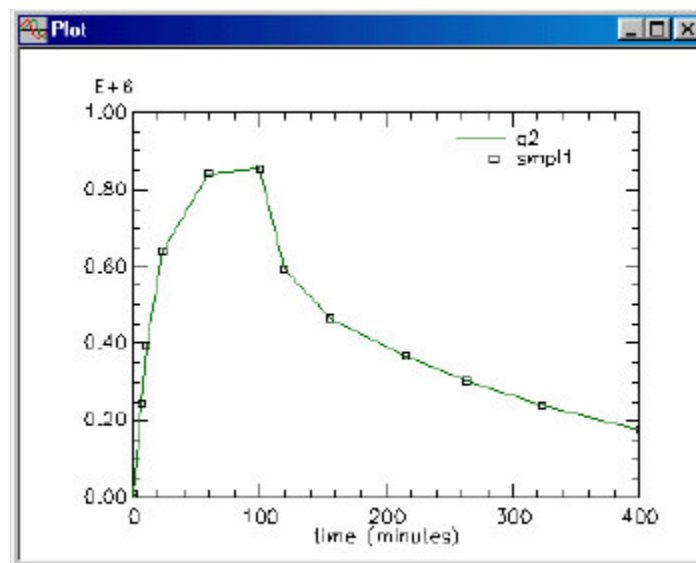


Figure 1

By changing the computational intervals setting to 100, the time between solution points is guaranteed never to exceed $400 / 100 = 4$ minutes. After making the change and solving again, the resulting plot in Figure 2 shows the curve to be much smoother.

Fitting (Optimization)

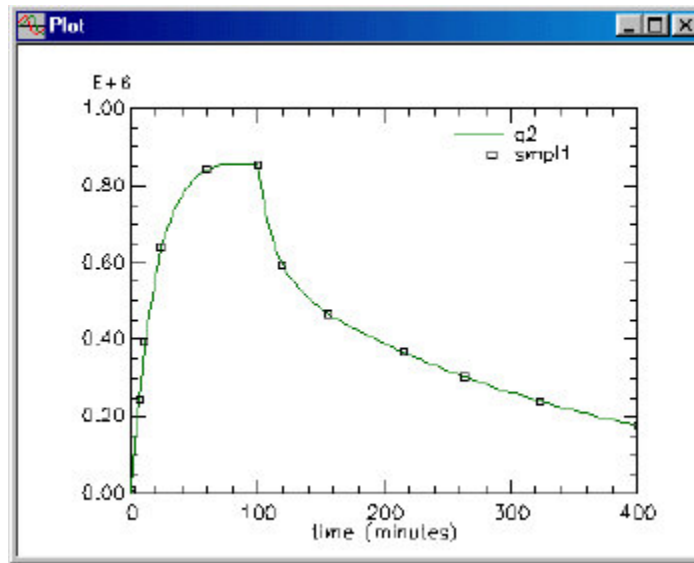


Figure 2

For a large model, the significant increase in the number of solution points could significantly increase the time required to Fit the model. There is another alternative. Since portions of the curve are fairly flat, it may not be necessary to calculate as many solution points in those areas. The area of particular interest in this example appears to be primarily from time 40 to 150. First, setting the number of calculation intervals back to 20 may be adequate before minute 40 and after minute 150 of the experiment. Since the data times also create solution points, opening the Data window and adding the data table for time only as shown below (notice that it contains only a column for time, t) will create solution points at each of those times.

Fitting (Optimization)

Data

t

40.0

45.0

50.0

55.0

60.0

65.0

70.0

75.0

80.0

85.0

90.0

100.0

105.0

110.0

115.0

120.0

125.0

130.0

135.0

140.0

END

Fitting (Optimization)

After closing the Data window and Solving the model again, the resulting curve is shown in Figure 3. Notice that the curve looks nearly as smooth as the one in Figure 2 that required many more solution points. This technique can be particularly useful where the finer detail is required over only a small portion of the curve.

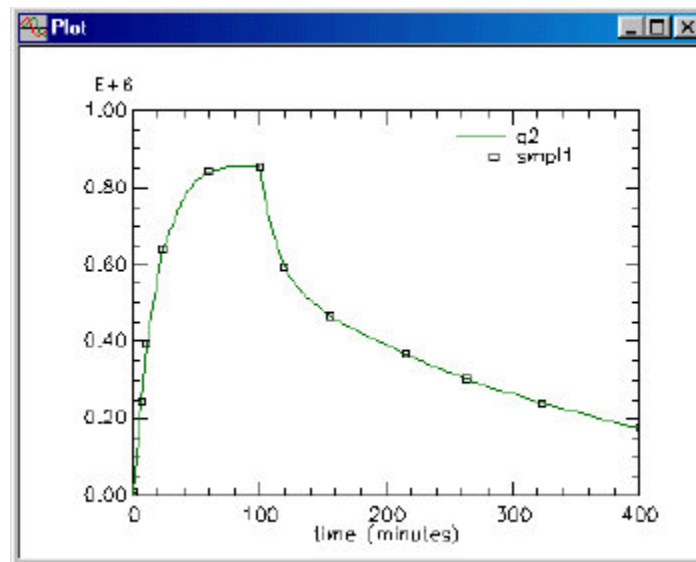


Figure 3

In general, it's best to leave the number of calculation intervals set to the default of 30 until after the model has been **Fitted**. If the plotted curve(s) are not satisfactory, then either the number of intervals can be increased for smaller models, or the last technique with time-only data points can be used if necessary to further enhance a small section of the model.

Q: How can I use Solutions when developing a model?

A: The **Solutions** feature in the SAAM II Compartmental and Numerical applications provides the capability to save the state of a model and the value(s) for selected variables following a **Solve** or **Fit**. In short, saving a solution stores the parameter values and results for selected variables. Assuming the model configuration is not modified, you can restore the parameter values at a later time. You may also compare the saved data variable values against those for the same or different variables of a subsequent solution. For more information on the mechanics of using the **Solutions** feature, see **Working with Solutions** in the **Computational Features** section in either the Help system or in the User Guide.

Fitting (Optimization)

Supposing a model has been developed that appears to satisfactorily fit the experimental data, but there are a few variations of the model that still might be worthwhile investigating (e.g. adding or deleting a compartment, adding a delay between compartments, etc.). Before modifying the model, **Fit** it and then open the **Solutions** window by clicking **Solutions** on the **Compute** menu. Enter a prefix, such as *Root*, that indicates that the current model is the basis for the subsequent variations. Click the **Add** button, select each of the samples in the **Solution Variables** list, and click the **Done** button. The saved solution will appear in the list in the left pane of the **Solutions** window. Click on it and notice that the saved samples appear in the right pane with a suffix indicating the name of the solution (e.g. *s1 {Root1}*). Close the **Solutions** window. It may also be useful to open the **Statistics** window, copy pertinent information on the **Fit** (such as the objective information) to the clipboard, and then paste it into the **Notes** window (or to another word processor) with the title *Root1* to indicate the source.

At this point you can modify the model as desired. When finished, **Fit** the new variation. To visually compare the results against the original model, open the **Plot** window and click **Plot/Table Variables...** on the **Set** menu to open the **Plot and Table Variables** dialog box. Insure that **List All Variables** is checked. In the list of variables, select the sample, the sample from *Root1*, and the associated data. Click the **Done** button. The resulting plot will show the original and new samples superimposed on the data, and may indicate whether the derivative model is an improvement. After closing the **Plot** window, open the **Statistics** window and compare the new statistics against those from *Root1* saved in the **Notes** window or elsewhere.

It may also be helpful to save the model to a study file after each modification so that it can be readily retrieved at a later time. Turning on the **Save Results to Text File** option is also a quick way to store sample values, statistics and other results after each **Fit** (in Version 1.2 or later).