



*Tutorials and worked examples for simulation,  
curve fitting, statistical analysis, and plotting.*  
<http://www.simfit.org.uk>

Simulating a data set that is exact to computer precision for differential equations is the first step in simulation, so that random error can be added retrospectively to simulate experimental results.

Choosing the [A/Z] option from the SIMFIT main menu is used to open the program **makdat**, which then allows you to create such almost-exact data from a library of models, or from a user-supplied model.

Before attempting to simulate a differential equation it is essential to understand these issues.

1. Choosing the correct model
2. Choosing sensible parameters (including initial conditions)
3. Choosing a reasonable range for the independent variable
4. Choosing a sensible number of divisions of the  $X$  axis
5. Viewing the current simulation
6. Saving the simulated data to a file

Program **makdat** has access to the particular library that is contained in either `w_models.dll` for 32-bit applications, or `x64_models.dll` for 64-bit applications, and there are numerous versions of these where the standard library has been augmented for special user requirements. The standard version has models for a single differential equation, but for systems of  $n$  differential equations program **deqsol** must be used.

## 1. Choosing a model

From the interface to the SIMFIT library choose the Michaelis-Menten substrate depletion model which has the following form

$$\frac{dy}{dx} = \frac{-p_2 y}{p_1 + y}.$$

Of course this model can actually be integrated and would usually have the following interpretation

$$\frac{dS}{dt} = \frac{-V_{max} S}{K_m + S}.$$

## 2. Choosing sensible parameters

There is an additional complication with differential equations as the the initial conditions are also parameters, and the SIMFIT convention is that with a system of  $n$  equations the last  $n$  parameters must be the initial conditions. Note that the default parameters for this model are as follows.

$$\begin{aligned} p_1 &= 1 \\ p_2 &= 1 \\ p_3 &= 1 \\ &= S(0) \end{aligned}$$

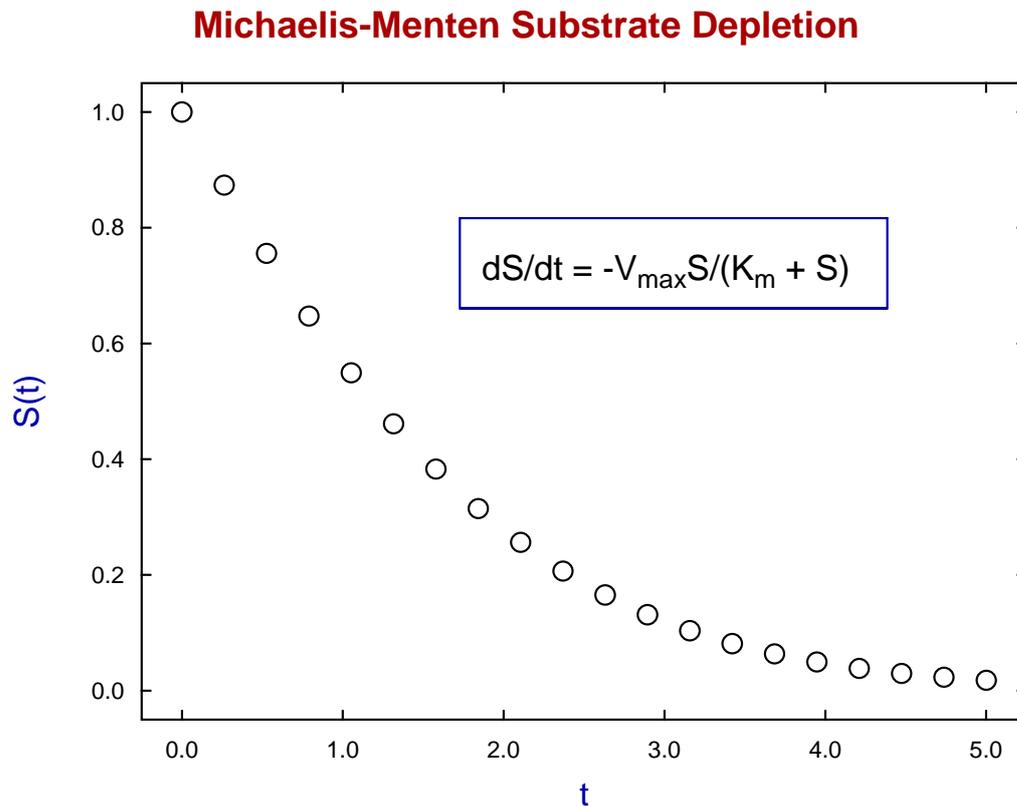
so that, with this model, parameter  $p_3 > 0$  is the substrate concentration at time  $t = 0$ .

### 3. Choosing a reasonable range for the independent variable(s)

It is usual to advance the solution for differential equations from time zero, which means that program **makdat** will not allow starting values less than zero.

### 4. Viewing the current simulation

Once a data set has been simulated it is always possible to display a table of the independent variables and simulated values, or a plot as follows.



To illustrate the process of finding a range of  $x$  for simulation when this depends on fixed values of  $y$ , that is to find  $x = x(y)$  when there is no simple explicit expression for  $x(y)$ , consider the Von Bertalanffy growth differential equation

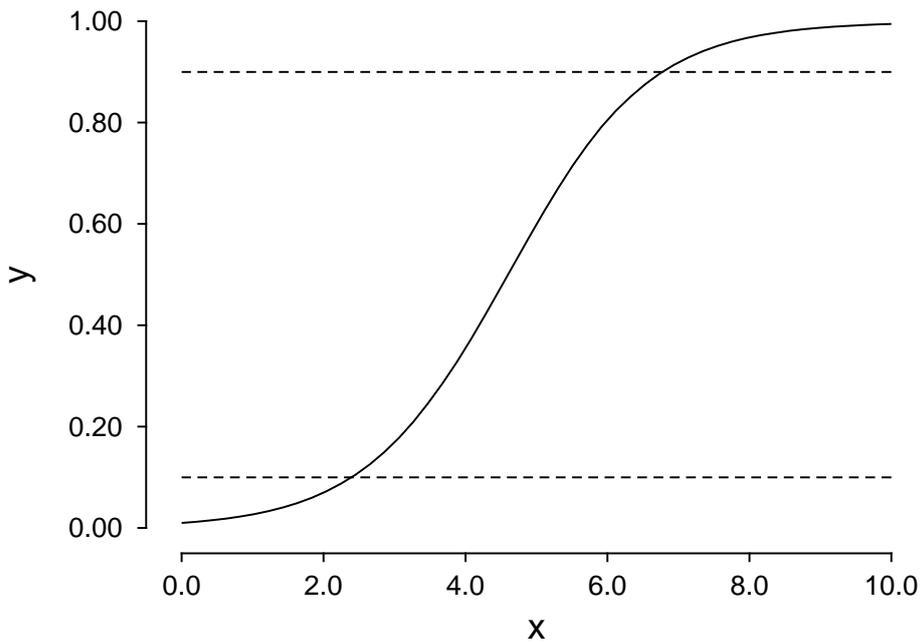
$$\frac{dy}{dx} = Ay^m - By^n,$$

where  $A > 0$ ,  $B > 0$  and  $n > m$ . After setting the parameters  $A = B = m = 1$ ,  $n = 2$ , and initial condition  $y_0 = 0.001$ , for instance, program **makdat** estimated the following results:

$X_{\text{start}} = 0$ ,  $X_{\text{stop}} = 5$ ,  $y_1 = 0.1$ ,  $x_1 = 2.3919$ ,  
 $X_{\text{start}} = 0$ ,  $X_{\text{stop}} = 9$ ,  $y_2 = 0.9$ ,  $x_2 = 6.7679$ ,

providing the roots required to simulate this equation between the limits  $y_1 = 0.1$  and  $y_2 = 0.9$ .

$$dy/dx = Ay^m - By^n$$



Note that, when attempting such root-finding calculations, **makdat** will attempt to alter the starting estimates by decreasing  $X_{\text{start}}$  and increasing  $X_{\text{stop}}$  if a root cannot be located, but it will not change the sign of these starting estimates. In the event of problems locating roots, there is no substitute for plotting the function to get some idea of the position of the roots, as shown in the previous figure.

## 5. Saving the simulated data to a file

Curve-fitting files saved will have the independent variables in the first column, then the simulated values in the penultimate column, followed by a final column of weights.

There are three reasons for saving the simulated data to a file.

- **Using the saved file for fitting**

It is assumed that normally data are simulated in order to be fitted. So a final column of weights (usually 5% of the calculated function value) will be added to the saved file to make it a curve fitting file. The last column must be left in for weighting, or else replaced by a column of 1 if unweighted fitting is required. This is easily done using program **editmt**.

- **Using the saved file for plotting**

The simulated data can be used retrospectively for plotting by program **simplot** which ignores the final column of weights.

- **Using the saved file for adding random error**

As the data written to file will be almost exact, they can be used to confirm that a model simulated with no added error can be fitted to return the correct best-fit parameters. If it is wished to add random error to simulate experimental data, the files can be input into program **adderr**, which can also overwrite the default weights by alternative weighting schemes if required.