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

Once a correct model has been chosen and accurate data consistent with this model and extending over a wide range have been obtained, the next most important item is to select parameter starting estimates close to the expected values, and also to provide sensible upper and lower parameter limits.

The simple SimFIT curve-fitting programs, such as **mmfit**, obtain starting estimates and limits as follows.

1. Scale the data to order unity.
2. Obtain slopes and/or horizontal asymptotes by linear regression to the early and late data.
3. Because data for simple curve-fitting programs must be in non-decreasing order, fits to extreme data points can usually be used to estimate parameters that become important at the extreme data regions.
4. As each of the simple curve-fitting programs is designed for a particular model, then starting estimates and limits can usually be guessed.
5. In addition, random searches around these estimates, but constrained by the parameter limits, can be used to refine such values.

However, as **qnfit** and **deqsol** are for expert users they must perform such tasks themselves. There is nothing to stop expert users from scaling the data before input to **qnfit** and submitting data in random order, etc., but the following facts cannot be avoided.

- The starting estimates must be reasonably close to the probable parameters.
- If this condition is not met then a meaningful solution is unlikely to be found as nonlinear regression only seeks to find a local minimum.
- The upper and lower limits must not be too wide or too narrow.
- If these conditions are not met then parameters will either not be able to move sufficiently freely to allow the gradient vector and Hessian to be estimated, or they will simply stick at boundaries.

Such effects will be immediately obvious by the following situations.

- Warnings about convergence problems and bad fitting will be displayed.
- Parameter standard errors will be large.
- The fit to data will be unsatisfactory.

The obvious action is to try new starting estimates and limits, possibly with a random search, until a good fit results. Then subsequent data files can have such values appended.

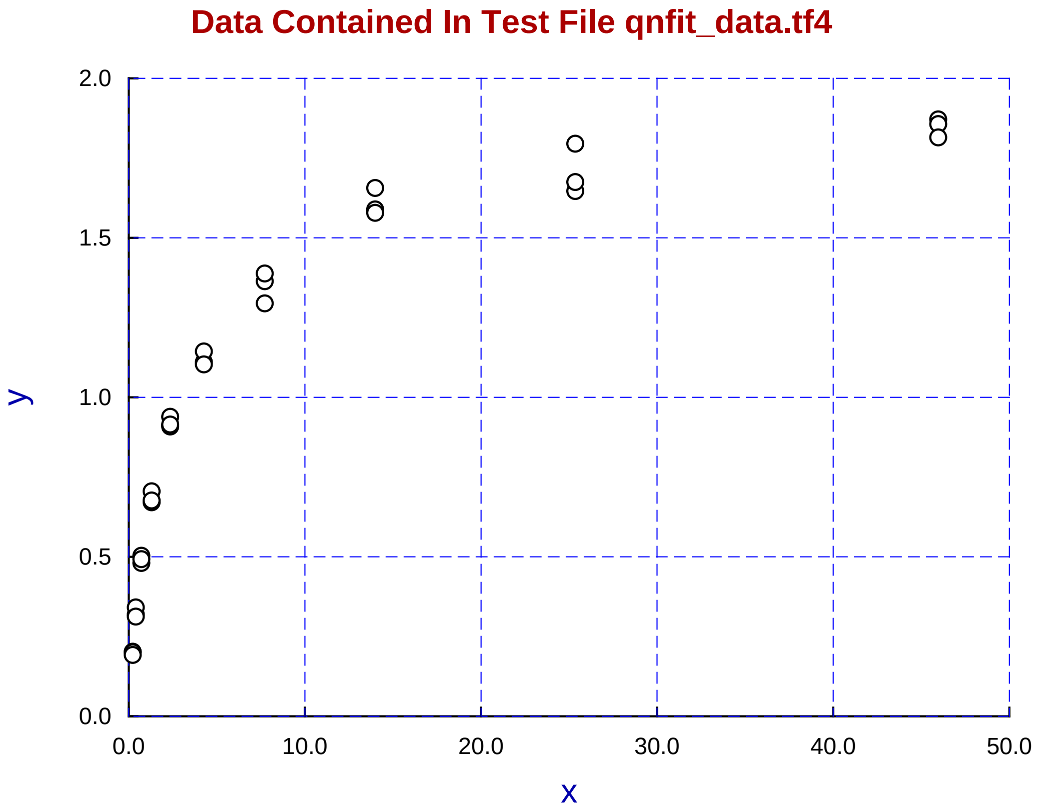
Some typical examples will be given under the assumption that users will supply their models as ASCII text files prepared using **usermod**, and will always append starting parameters and limits to the data files and use the EXPERT mode of running **qnfit**.

Example 4: Fitting the two-site Michaelis-Menten model

The initial steady-state for a mixture of two one-site enzymes or for two independent sites on one enzyme can be modeled by this expression

$$v(s) = \frac{Vmax_1 S}{Km_1 + S} + \frac{Vmax_2 S}{Km_2 + S}$$

where S is the (constant) substrate concentration. First of all use **simplot** to plot the tests data file `qnfit_data.tf4` to obtain the following graph.



From this graph we can see from the asymptotic behavior as $S \to \infty$ that $Vmax_1 + Vmax_2 \approx 2$, while from the half-saturation point Km_1 and Km_2 are between 1 and 10. It should also be observed that, because of symmetry in the model, the pairs $(Vmax_1, Km_1)$ and $(Vmax_2, Km_2)$ are arbitrary and can be interchanged.

To emphasize: the usual and recommended way to use **qnfit** is to read in the data as a SimFIT data file with starting estimates and limits appended, and then to supply the model required as a ASCII text file prepared using **usermod**. So, to demonstrate this procedure, the next sequence should be used.

- Open **qnfit** from the [A/Z] button on the main SimFIT menu, choose to input data for one function of one variable, and then, from the [Demo] button, select `qnfit_data.tf4` which has the following starting estimates and limits appended.

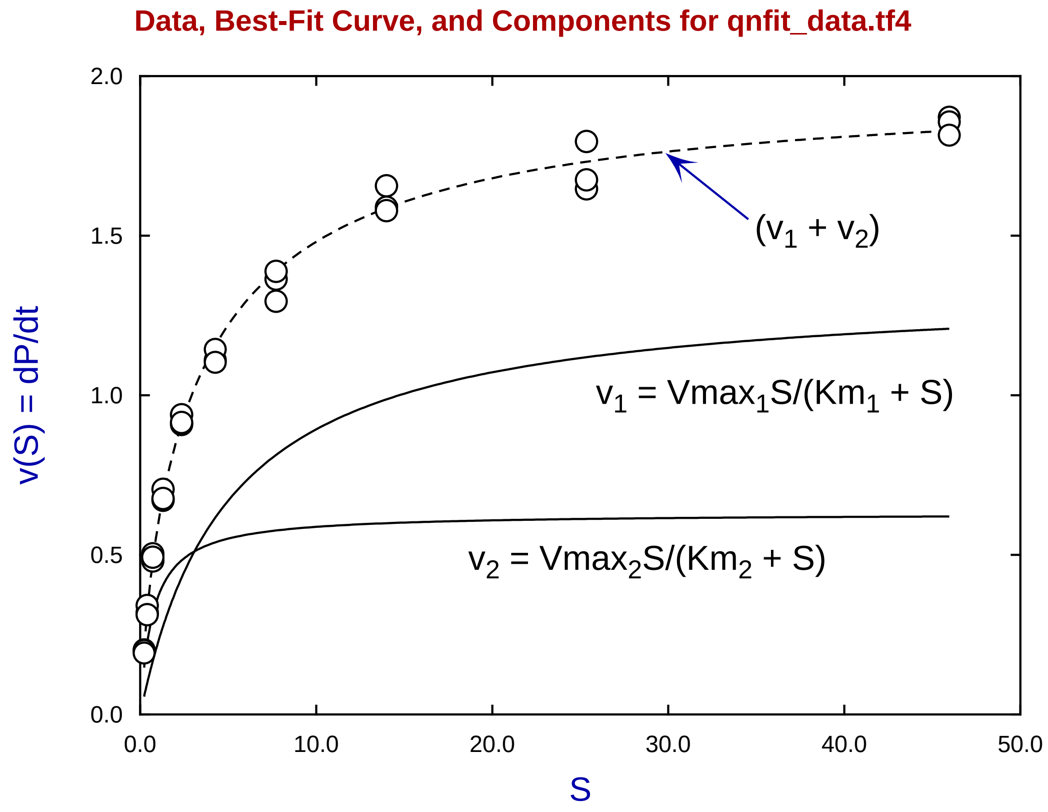
```
begin{limits}
 0.1   0.8   5.0
 0.1   1.2   5.0
 0.1   1.0  15.0
 0.1   8.0  15.0
end{limits}
```

- Then choose to input the corresponding ASCII text model file `qnfit_model.tf4` which defines the model in the following way, where using $f(1) = A + B$ shows how the model could easily be extended to three or more isoenzymes.

```
%
Sum of two independent Michaelis-Menten active sites
v(S) = Vmax_1*S/(Km_1 + S) + Vmax_2*S/(Km_2 + S) where
p(1) = Vmax_1, p(2) = Vmax_2, p(3) = Km_1, p(4) = Km_2
1 equation
1 variable
4 parameters
%
begin{expression}
A = p(1)x/[p(3) + x]
B = p(2)x/[p(4) + x]
f(1) = A + B
end{expression}
%
```

The parameter estimates and confidence limits from fitting are displayed in the next table followed by a plot of data, best-fit curve and sub-models contributing to the overall fit.

Number	Low-Limit	High-Limit	Value	Standard Error	Lower95%cl	Upper95%cl	*p*
1	0.1	5.0	1.02901	0.13352	0.7546	1.30346	0.0000
2	0.1	5.0	1.02542	0.10378	0.8121	1.23873	0.0000
3	0.1	15.0	1.04328	0.11698	0.8028	1.28372	0.0000
4	0.1	15.0	9.74605	2.86518	3.8566	15.6355	0.0022



The plot was constructed by re-fitting using the corresponding built-in **qnfit** library model which is recognized as a linear combination of sub-models, but this does not happen with user-defined model files so the separate components have to be plotted independently then overlayed upon the usual best-fit display.

Example 5: Fitting a double exponential model

This example involves fitting a model for B in the irreversible chemical kinetic scheme $A \xrightarrow{k_1} B \xrightarrow{k_2} C$ which, with various additional definitions, is used extensively elsewhere, for instance for absorption from the stomach and elimination from the blood in pharmacokinetics.

$$f(t) = \frac{k_1 A_0}{k_2 - k_1}\left[\exp(-k_1 t) - \exp(-k2t)\right], \quad k_1 \neq k_2$$

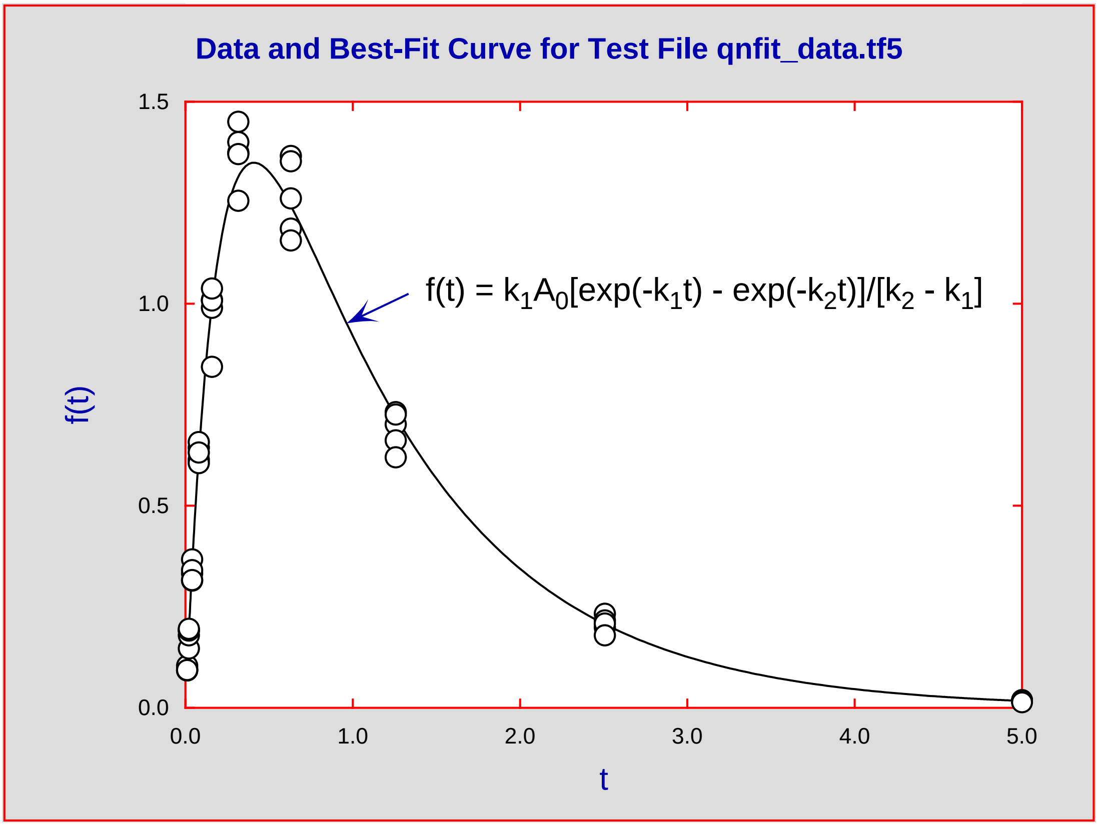
The test file is `qnfit_data.tf5` and, from viewing the data in **simplot** clearly k_1 is around unity, while $k_2 > k_1$, and $2 < A_0 < 30$ is less easy to guess, so the following starting estimates and limits are appended.

```
begin{limits}
0.0   0.5 10.0
0.0   6.0 10.0
0.0  12.0 30.0
end{limits}
```

The model file in `qnfit_model.tf5` is shown next, followed by the table of results and the plot.

```
A = p(1)p(3)/[p(2) - p(1)]
B = exp[-p(1)x] - exp[-p(2)x]
f(1) = A*B
```

Number	Low-Limit	High-Limit	Value	Standard Error	Lower95%cl	Upper95%cl	*p*
1	0.0	10.0	1.00458	0.015634	0.97313	1.03604	0.0000
2	0.0	10.0	4.85240	0.146363	4.55795	5.14684	0.0000
3	0.0	30.0	9.83258	0.224251	9.38144	10.2837	0.0000


Data and Best-Fit Curve for Test File qnfit_data.tf5
f(t)
t
f(t) = k$_1$A$_0$[exp(-k$_1$t) - exp(-k$_2$t)]/[k$_2$ - k$_1$]