Simfit

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

Constrained nonlinear regression involves minimizing an objective function such as a scaled weighted sum of squares

$$\frac{WSSQ}{NDOF} = \frac{1}{n-m}\sum_{i=1}^{n} w_i(y_i - f(x_i,\Theta))^2$$

with respect to a m dimensional parameter vector Θ, where there are n observations y_i, and a model $f(x,\Theta)$, along with parameter starting estimates and limits, and specified weights w_i. The aim is to use the parameter estimates and confidence limits to asses the value of model parameters that have a physical interpretation, such as diffusion constants, chemical reaction rate constants, or growth rates, etc. Of course the observations y, independent variables x, functions f, and weights w would often be vector quantities. The main SimFIT programs to perform this type of optimization are **qnfit** providing quasi-Newton and other methods, and **deqsol** for systems of differential equations, and numerous considerations must be understood and several conditions must be satisfied before these iterative techniques can obtain sensible solution points. Some of these these are now summarized.

1. **The model**

   The model can be used from a default library of models, but it is normally anticipated that users would define their own specialized model, and create a model file using program **usermod**. It will be obvious that the model should be parsimonious, using only the minimum number of parameters, and where every parameter has a scientific interpretation. Often data can be normalized by users before curve-fitting if this reduces the number of parameters that need to be estimated. For instance, normalizing observations so that $f(0) = 0$, or $f(\infty) = 1$, or, especially in the case of differential equations, so that initial conditions do not need to be estimated.

2. **The data**

   The data must be extensive in the sense that $n >> m$, and with a high signal to noise ratio, but they should also cover a range where the effect of all parameters can be assessed. For instance, with exponential decay the range should extend beyond the longest half-life, and, where growth data or ligand binding data approach a horizontal asymptote, the experimental data should clearly be starting to look asymptotic.

3. **The weights**

   The variance of experimental observations almost always increases as the absolute value of the observations increase, and even though the expectation will often be zero, the distribution will have longer tails, more like a Cauchy than a Gaussian distribution. Now the theory required for the analysis of goodness of fit and calculation of statistics to estimate parameter reliability and perform model discrimination depends on the principle of maximum likelihood, which assumes a linear model with uncorrelated normally distributed error. This means that either $w_i = 1$ if the variance is constant, or $w_i = 1/s_i^2$ otherwise where the standard deviations s_i are known exactly. So several approaches are possible.

   - Assume constant variance and set all $s_i = 1$. This leads to fitting being dominated by large observations, and hence the parameters contributing to the large values will be estimated more accurately than those only contributing to small values.
   - Assume constant relative error and assume that standard deviation is proportional to the absolute value of the observation. This can lead to the opposite effect to assuming constant variance, i.e. biasing the fit towards small values.
   - Assume that s_i is a defined function of the observations or the best-fit function values. This requires an assumption about the functional dependency of variance and, if the best-fit model is used rather than observations, then weighting changes as iterations proceed.

- Estimate the error variance independently. This is undoubtedly the best method as long as at least five replicates are determined at each independent variable setting and, if possible, a smoothing technique is used to determine a reliable model for the change in variance as a function of the value of observations.

4. **The starting estimates and limits**

   Constrained nonlinear optimization is an iterative technique that attempts to find a local minimum given parameter starting estimates and parameter limits. So naturally it is important that the starting estimates are close to the true values and the limits are not so wide that parameters can stray into unlikely regions of parameter space. SimFIT programs **qnfit** and **deqsol** also use the starting estimates to normalize the internal parameters to order unity, as calculation of maxim descent vectors and augmented Lagrangians will be most accurate if all internal parameters are of order unity.

Success and Failure

If all the above criteria are met then convergence to a minimum should be achieved so that $WSSQ$ will be approximately chi-squared distributed with $NDOF = n - m$. Then the objective function should be of order unity with reasonable parameter estimates and satisfactory goodness of fit analysis.

On the other hand, if the conditions are not met then failure will occur with appropriate error messages. In such dubious cases you should switch on the options to evaluate the parameter covariance matrix, the condition number of the Hessian, and study tables of residuals and residuals plots. Note that, if the objective functions is too small or too large on entry due to an inappropriate model, poor quality data, incorrect weighting, or unrealistic starting estimates, then the routine will not be able to estimate the gradient vector and exit will occur without fitting.

In order to become familiar with program **qnfit** some very simple examples will be given next to illustrate the standard way to proceed. That is:

1. select the model type required, e.g. one function of one variable;
2. input a data set composed in the EXPERT mode with starting estimates and limits appended;
3. read in the model file, e.g. created using program **usermod**; then
4. proceed to fitting.

In the next fairly trivial examples note that the test data files and model files can be easily located using the [Demo] button on the file-open dialogue box.

Example 1: One function of one variable

Open SimFIT program **qnfit** then follow the next steps.

1. Choose to fit one function of one variable
2. From the file-open dialogue press [Demo] then view and open the test file
   `qnfit_data.tf1`
3. Choose to open an ASCII text model file then from the file-open dialogue press [Demo] then view and open the test file
   `qnfit_model.tf1`
4. Choose the EXPERT mode for starting estimates then fit

Note that this is simulated data for a quadratic and the EXPERT mode appended section is as follows.

```
begin{limits}
  -10  1  10
  -10  1  10
  -10  1  10
end{limits}
```

The model file defines a quadratic $f(x) = p_1 x + p_2 x^2 + p_3$ as follows.

```
%
Model for a polynomial of degree 2

f(x) = p(1)x + p(2)x^2 + p(3)

%
1 equation
1 variable
3 parameters
%
begin{expression}
f(1) = p(1)x + p(2)x^2 + p(3)
end{expression}
%
```

Now, to obtain a permanent copy of the outcome after fitting, extract the table of best-fit parameters using the [Results] then [Extract tables] options from the main SimFIT menu to import the following table into your document.

Best-fit parameters for curve-fit 1 using LBFGSB

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	p
1	-10.0	10.0	2.12035	0.0197309	2.07829	2.1624	0.0000
2	-10.0	10.0	-0.11565	0.0035714	-0.12326	-0.1080	0.0000
3	-10.0	10.0	0.10347	0.0032091	0.09663	0.1103	0.0000

For 50,90,95,99% confidence limits using [parameter value +/- $t(\alpha/2)$*std.err.]
t(0.25) = 0.691, t(0.05) = 1.753, t(0.025) = 2.131, t(0.005) = 2.947

Note that the $t_\nu(.)$ values are provided in case you want to calculate parameter confidence limits in addition to the default 95% values.

Example 2: One function of two variables

Proceeding exactly as for example one except that one function of two variables is chosen, the data file is `qnfit_data.tf2` and the model file is `qnfit_model.tf2` observe that now the data file has four columns $(x, y, g(x, y), s)$ for observations $g(x, y)$.

The appended EXPERT mode section defining the lower-limits, starting values, and upper limits for the three parameters follows

```
begin{limits}
-10 -2 10
-10  2 10
-10  4 10
end{limits
```

while the model has the next definition followed by the results from fitting.

```
%
Linear model with two variables
g(x,y) = p(1)x + p(2)y + p(3)
%
1 equation
2 variables
3 parameters
%
begin{expression}
f(1) = p(1)x + p(2)y + p(3)
end{expression}
%
```

Best-fit parameters for curve-fit 2 using LBFGSB

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	p
1	-10.0	10.0	0.96311	0.0334131	0.895683	1.03054	0.0000
2	-10.0	10.0	0.95436	0.0323055	0.889165	1.01955	0.0000
3	-10.0	10.0	1.05694	0.0359344	0.984422	1.12946	0.0000

For 50,90,95,99% confidence limits using [parameter value +/- $t(\alpha/2)$*std.err.]
t(0.25) = 0.680, t(0.05) = 1.682, t(0.025) = 2.018, t(0.005) = 2.698

Example 3: One function of three variables

This time the data file is `qnfit_data.tf3`, while the model file is `qnfit_model.tf3` defining a function of three variables thus

```
%
Linear model with three variables
h(x,y,z) = p(1)x + p(2)y + p(3)z + p(4)
%
1 equation
3 variables
4 parameters
%
begin{expression}
f(1) = p(1)x + p(2)y + p(3)z + p(4)
end{expression}
%
```

while the best-fit parameters for the model $h(x, y, z) = p_1x + p_2y + p_3z + p_4$ are displayed in the next table.

Best-fit parameters for curve-fit 3 using LBFGSB

Number	Low-Limit	High-Limit	Value	Std.Error	Lower95%cl	Upper95%cl	p
1	-10.0	10.0	1.00348	0.015231	0.97335	1.03361	0.0000
2	-10.0	10.0	0.99476	0.017071	0.96098	1.02853	0.0000
3	-10.0	10.0	0.98594	0.017159	0.95199	1.01988	0.0000
4	-10.0	10.0	-2.94533	0.055805	-3.05572	-2.83493	0.0000

For 50,90,95,99% confidence limits using [parameter value +/- $t(\alpha/2)$*std.err.]
t(0.25) = 0.676, t(0.05) = 1.657, t(0.025) = 1.978, t(0.005) = 2.614