The Epsilon Wave

Curve fitting uses partial derivatives of your fit function with respect to the fit coefficients in order to find the gradient of the chi-square surface. It then solves a linearized estimate of the chi-square surface to find the next estimate of the solution (the minimum in the chi-square surface).

Since Igor doesn't know the mathematical equation for your fit function, it must approximate derivatives numerically using finite differences. That is, a model value is calculated at the present estimate of the fit coefficients, then each coefficient is perturbed by a small amount, and the derivative is calculated from the difference. This small perturbation, which is different for each coefficient, is called “epsilon”.

You can specify the epsilon value for each coefficient by providing an epsilon wave to the **CurveFit** or **FuncFit** operations using the /E flag. If you do not provide an epsilon wave, Igor determines epsilon for each coefficient using these rules:

If your coefficient wave is single precision (which is not recommended),

```
if coef[i] == 0
   eps[i] = 1e-4
else
   eps[i] = 1e-4*coef[i]
```

If your coefficient wave is double precision,

```
if coef[i] == 0
   eps[i] = 1e-10
else
   eps[i] = 1e-10*coef[i]
```

In the Curve Fitting dialog, when you are using a user-defined fitting function, you can select an epsilon wave from the Epsilon Wave menu. When you select _New Wave_, or if you select an existing wave from the menu, Igor adds a column to the Coefficients list where you can edit the epsilon values.

There are a couple of reasons for explicitly setting epsilon. One is if your fit function is insensitive to a coefficient. That is, perturbing the coefficient makes a very small change in the model value. Sometimes the dependence of the model is so small that floating-point truncation results in no change in the model value. You have to set epsilon to a sufficiently large value that the model actually changes when the perturbation is applied.

You also need to set epsilon is when the model is discrete or noisy. This can happen if the model involves a table look-up or a series solution of some sort. In the case of table look-up, epsilon needs to be large enough to make sure that you get two distinct values out of the table.

In the case of a series solution, you have to stop summing terms in the series at some point. If the truncation of the series results in less than full floating-point resolution of the series, you need to make sure epsilon is large enough that the change in the model is larger than the resolution of the series. A series might include something like a numerical solution of an ODE, using the **IntegrateODE** operation. It could also involve **FindRoots** or **Optimize**, each of which gives you an approximate result. Since these operations run faster if you don't demand high precision, there may be a strong incentive to decrease the accuracy of the computation, and that may in turn lead to a need for an epsilon wave.

Curve Fitting References

An explanation of the Levenberg-Marquardt nonlinear least squares optimization can be found in Chapter 14.4 of:

Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C*, 2nd ed., 994 pp., Cambridge University Press, New York, 1992.

The algorithm used for applying constraints is given in:

Shrager, Richard, Quadratic Programming for Nonlinear Regression, *Communications of the ACM*, *15*, 41-45, 1972.

The method is described in gory mathematical detail in:

Shrager, Richard, Nonlinear Regression With Linear Constraints: An Extension of the Magnified Diagonal Method, *Journal of the Association for Computing Machinery*, *17*, 446-452, 1970.

References for the ODRPACK95 package used for orthogonal distance regression:

Boggs, P.T., R.H. Byrd, and R.B. Schnabel, A Stable and Efficient Algorithm for Nonlinear Orthogonal Distance Regression, *SIAM Journal of Scientific and Statistical Computing*, *8*, 052-1078, 1987.

Boggs, P.T., R.H. Byrd, J.R. Donaldson, and R.B. Schnabel, Algorithm 676 - ODRPACK: Software for Weighted Orthogonal Distance Regression, *ACM Transactions on Mathematical Software*, *15*, 348-364, 1989

Boggs, P.T., J.R. Donaldson, R.B. Schnabel and C.H. Spiegelman, A Computational Examination of Orthogonal Distance Regression, *Journal of Econometrics*, *38*, 69-201, 1988.

An exhaustive, but difficult to read source for nonlinear curve fitting:

Seber, G.A.F, and C.J. Wild, *Nonlinear Regression*, John Wiley & Sons, 1989.

A discussion of the assumptions and approximations involved in calculating confidence bands for nonlinear functions can be found in the beginning sections of Chapter 5.

General books on curve fitting and statistics:

Draper, N., and H. Smith, *Applied Regression Analysis*, John Wiley & Sons, 1966.

Box, G.E.P., W.G. Hunter, and J.S. Hunter, *Statistics for Experimenters*, John Wiley & Sons, 1978.