

NAME

simplex-1994, simplex, simpfit – Nelder-Mead simplex minimization for parameter estimates with standard deviations

SYNOPSIS

simpfit [no options] [< inputfile | stdin] [> outputfile | stdout]

DESCRIPTION

Simpfit minimizes a function by adjusting the variable parameters according to the value returned by a user-coded subroutine. There are no restrictions on the form of the function, except that a value must be computable given the input parameters. Derivatives are not calculated. Singularities can be tolerated. Bounds tests can be incorporated simply.

The Nelder-Mead (1965) algorithm is a procedure for systematically shrinking or otherwise adjusting the shape of a polygon, the simplex. The simplex is constructed in parameter space, with vertices that are arbitrary except they describe a volume that includes the minimum point. Shrinking is carried out so that the function minimum remains within the simplex. Shrinking is continued until the vertices are so close together, satisfying an exit test, that the mean of the vertices defines with sufficient accuracy the parameter values of the function minimum. Sometimes the procedure can locate the minimum even if the starting simplex does not include it. If the function surface is sufficiently rough, the procedure fails through being trapped within a local minimum. The Nelder-Mead method appears to be more robust than gradient methods, but slower to converge when close to the minimum.

Standard deviations of the parameters are calculated by fitting a quadratic to the function surface at the minimum and then using the properties of the quadratic to calculate the variance-covariance matrix. The strategy is given by Nelder and Mead. The quadratic approximation at the current simplex minimum can be used to define a better estimate of the minimum and to construct a new simplex based upon it. Alternation between iterations of simplex shrinking and quadratic approximation can lead to faster approach to the best estimate of the minimum.

The Nelder-Mead algorithm can be used to fit a non-linear model to data. In this case, the function minimized is the least-squares test for quality of fit, the weighted sum of squares of the residuals. A model function explicitly relating a dependent variable to one or more independent variables and variable parameters is fit to a set of data points (observed values of dependent and independent variables). The simplex search locates the least squares minimum, giving estimates of the best-fit values of the parameters of the model function. Standard deviations of the parameters are estimated by quadratic approximation of the least squares surface near the minimum.

CODING

In order to minimize a function, one routine has to be coded specifically for it. **func()** calculates the function value for the simplex vertex (the set of parameter values) passed to it. Generally, only one or a few lines of code have to be rewritten. Look at code for one of the test functions (`powell.c` and `ldhfit.c`). Bounds on parameters can be implemented by a test that returns an excessively large function value. When fitting data, a second routine, **fdatprint()**, should be recoded to display calculated and input values of the independent and dependent variables, associated weights, etc. You are on your own here. The minimizer knows nothing about the data, except that it should read data in if told to, and call `fdatprint()` to display results, when appropriate.

INPUT AND OUTPUT

The input has the following structure:

one-line title;

lines following give control constants, the starting simplex, bounds, and for data fitting, the data array; the order of entry is hard-wired into the code; see a sample data file for structure. the one-word descriptors must be present in the data file; the data format is otherwise free-form;

comments at the end of the file are not read by the program

Please look at the demonstration data file (`simpfit.dat.template`) for information on what control values should be supplied.

Output is not extensive. It is meant to be self-documenting. For details, please consult the source code and the Nelder and Mead (1965) publication.

FILES

The file `ldhfit.c` is an example of coding for fitting a model function to data. The file `ldhfit.dat` is a sample input file, having typical values of the control variables, and showing the form of the simplex array (vertices by row with parameter values in columns) and data array (data points by row, with variables and weights, etc, in columns).

The files `powell.c` and `powell.dat` show coding and input for minimization of one of the "dataless" test functions that was considered by Nelder and Mead.

The file `simpfit.dat.template` explains the input.

REFERENCE:

A simplex method for function minimization.

J.A. Nelder and R. Mead (1965). *Computer J.* 7, 308.

AUTHOR

J. A. Rupley, Tucson, Arizona
rupley@u.arizona.edu

BUGS

There are limits, set in `simpdefs.h`, on the number of parameters, number of data points, and number of variables per data point.

The interface is inhuman.