
Subject: Re: minimization method(amoeba subroutine)
Posted by [Russell Ryan](#) on Thu, 23 May 2013 17:55:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thursday, May 23, 2013 1:59:44 AM UTC-4, gunvi...@gmail.com wrote:

> Hello everyone,

>

> Can anyone please explain me about minimization method. I recently read in a paper "The minimization was done using down hill simplex method(Nelder and Mead(1965)), So I checked in idl which routine will do this and found that amoeba subroutine will do this. But I couldnt understand the concept behind this. Could any of you explain me what exactly this 'amoeba subroutine' does and how to use it.

>

> Thanking you in advance

Mats and Craig have good answers. But from a practical side, the amoeba algorithm has pros and cons.

PROs:

only need to evaluate the function (ie. not any derivatives).

quickly finds a local minimum with an intuitive approach (faster because you don't compute derivatives).

NOTE: If you're fitting an analytic function where you can compute the derivative, then I'd recommend using an algorithm that is capable of using this information (such as POWELL or MPFIT). There's no reason *NOT* to use this info.

CONs:

might be faster by using derivatives (such as POWELL or LMFIT/MPFIT --- like the note above) is not a global minimizer.

does not give confidence intervals, uncertainties, or PDFs. Maybe you want error bars too?

All of these algorithms are tricky to use because they don't (necessarily) find the global minimum, so you ought to embed such algorithms in fancy things (like simulated annealing or so on). To use the amoeba.pro implmentation, you need to build a "simplex" or set the ranges. This is easy to do for some functions, but hard for others (for example, say some part of parameter space is out of bounds for some value of another parameter. Suppose you're fitting x,y,z to some function. Now $x > 0$ if $y < 0$, but $-\infty < x < \infty$ for $y > 0$. What are the bounds now?). I've never found much use in amoeba, it's clunky and using derivatives helps a lot --- even if they're finite difference derivatives. Since CW already posted, you ought to look at his code: mpfit. For every application I've needed non-linear fitting, this seems to work much better and has lots more control over the fitting (such as the constraining parameters).