

## LA-UR-14-24879

Approved for public release; distribution is unlimited.

Title: Comments on Different techniques for finding best-fit parameters

Author(s): Fenimore, Edward E.  
Triplett, Laurie A.

Intended for: document past work

Issued: 2014-07-01

---

**Disclaimer:**

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

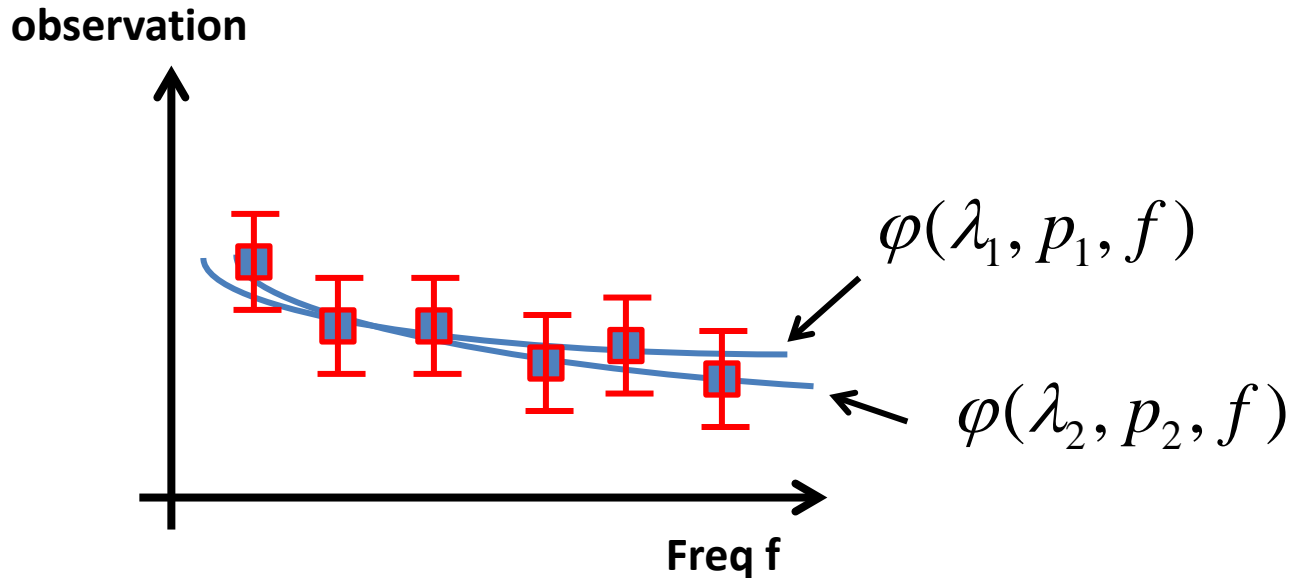
# **Comments on different techniques for finding best-fit parameters**

**Ed Fenimore, Laurie Triplett  
Los Alamos National Laboratory  
LA-UR xx-xxxx**

# Abstract

A common data analysis problem is to find best-fit parameters through chi-square minimization. Levenberg-Marquardt is an often used system that depends on gradients and converges when successive iterations do not change chi-square more than a specified amount. We point out in cases where the sought-after parameter weakly affects the fit and cases where the overall scale factor is a parameter, that a Golden Search technique can often do better. The Golden Search converges when the best-fit point is within a specified range and that range can be made arbitrarily small. It does not depend on the value of chi-square.

# A typical problem: fit 2 parameters to N data points



$$\chi^2 = \sum_1^N \frac{\{o_i - \varphi(\lambda, p, f_i)\}^2}{\sigma_1^2}$$

**Eq 1**

**Consider situation where  $\varphi(\lambda, p, f)$  is weakly dependent on  $p$  and  $\lambda$  is a scale factor. That is**

$$\varphi(\lambda, p, f) = \lambda \varphi'(p, f) \quad \text{Eq 2}$$

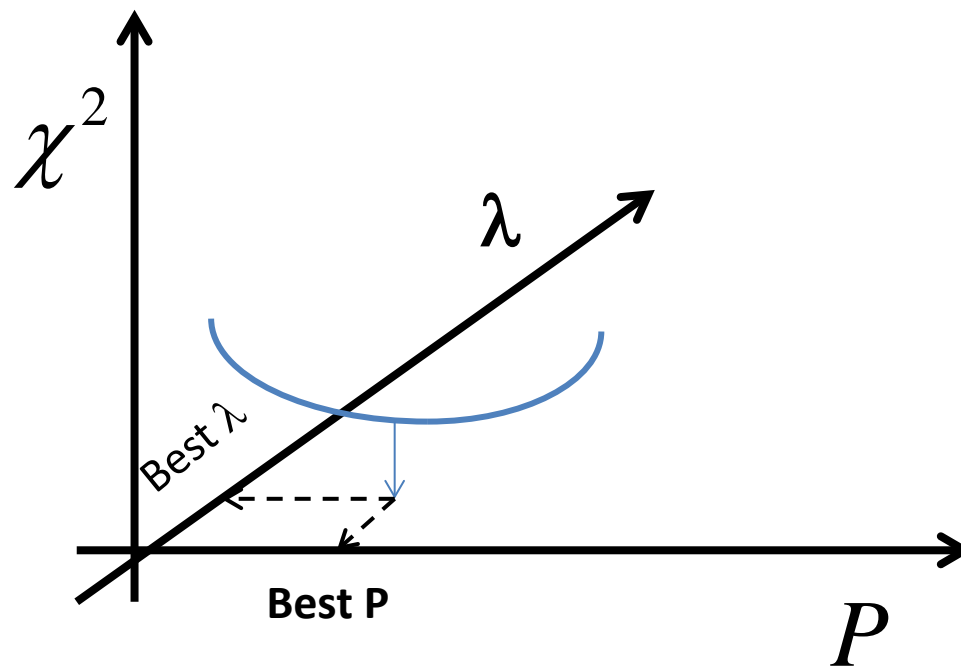
**The weak dependency on  $p$  means that the  $\chi^2$  might be a long shallow surface.**

**A common method for fitting parameters to data uses the Levenberg-Marquardt method to find the  $\chi^2$  minimum.**

**If the L-M method fits to two parameters (i. e.,  $\lambda$  and  $p$ ), it needs the partial derivatives  $\delta\phi/\delta\lambda$  and  $\delta\phi/\delta p$ .**

**The convergence criteria is usually that successive iterations of  $\chi^2$  are not changing much and/or that the reduced  $\chi^2$  is less than  $\sim 1$ .**

Use gradients to find minimum in 2-D  
until changes in  $\chi^2$  are small





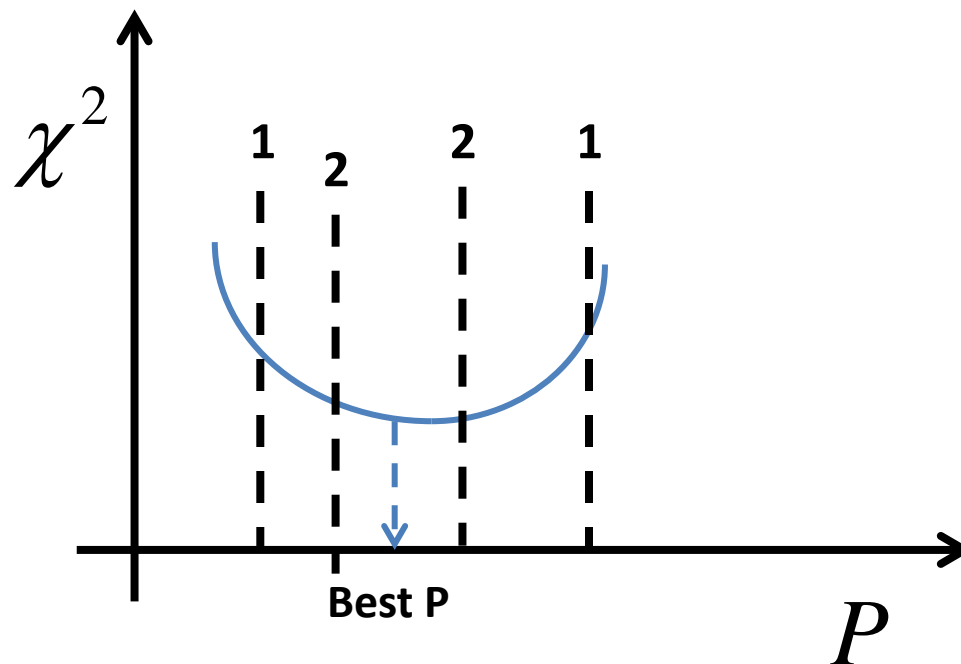
**An alternative method: “Golden search” to minimize  $\chi^2$  in 1-D by finding the best  $\lambda$  analytically for any value  $p$**

$$\chi^2 = \sum_1^N \frac{\{o_i - \lambda \phi'(p, f_i)\}^2}{\sigma_i^2} \quad \text{Eq 3}$$

$$\frac{\delta \chi^2}{\delta \lambda} = 0 = 2 \sum_1^n \frac{\{o_i - \lambda \phi'(p, f_i)\} \phi'(p, f_i)}{\sigma_i^2} \quad \text{Eq 4}$$

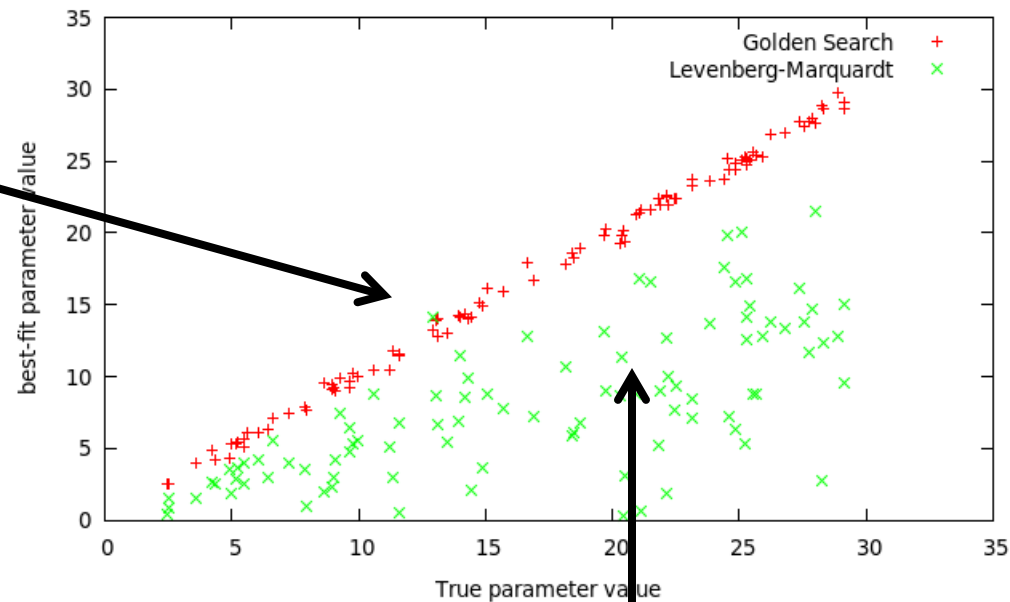
$$\lambda \sum_1^N \frac{\phi'(p, f_i) \phi'(p, f_i)}{\sigma_i^2} = \sum_1^N \frac{o_i \phi'(p, f_i)}{\sigma_i^2} \quad \text{Eq 5}$$

**Keep Splitting the 1-D range until the minimum is bracketed to some accuracy in  $p$  (no dependency on the value of  $\chi^2$ ).**



## Compare the $p$ found by analysis with the true $p$ (warning: fake data)

Golden search  
method always  
finds minimum



L-M method underestimates  $p$   
because converges early near its initial  
guess and not at the true minimum

**Because the L-M method determines  $p$  poorly, it is unlikely that  $\lambda$  could be found.**

**Since the Golden Search method can find  $p$  well, it is likely that  $\lambda$  can also be found from the data. It is given by Eq 5.**