

Estimation of the Parameters of the Reiber's Hyperbolic Function with the Levenberg-Marquardt Algorithm

Piotr Lewczuk

Lab for Clinical Neurochemistry, Department of Psychiatry, Universitätsklinikum Erlangen, Erlangen, Germany
Email: Piotr.Lewczuk@uk-erlangen.de

How to cite this paper: Lewczuk, P. (2025) Estimation of the Parameters of the Reiber's Hyperbolic Function with the Levenberg-Marquardt Algorithm. *Open Journal of Statistics*, 15, 391-396.
<https://doi.org/10.4236/ojs.2025.155021>

Received: August 18, 2025

Accepted: September 25, 2025

Published: September 28, 2025

Copyright © 2025 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

A hyperbolic model for the diffusion of proteins through the blood-cerebrospinal fluid (CSF) barrier revolutionized clinical neurochemistry thirty years ago. The regression curves were informally parametrized based on physiologically-driven constraints. The current paper readdresses this issue with numerical optimization for unconstrained non-linear regression, implementing the Levenberg-Marquardt Algorithm (LMA). Astonishingly similar estimates are obtained, which reconfirms the concepts of H. Reiber proposed in 1990s. The LMA is discussed in the context of other optimization algorithms.

Keywords

Cerebrospinal Fluid, Blood-CSF Barrier, Numerical Optimization, Levenberg-Marquardt Algorithm

1. Introduction

Thirty years ago, Hansotto Reiber proposed a hyperbolic model for the diffusion of proteins through the blood-cerebrospinal fluid (CSF) barrier [1]. His model revolutionized the CSF analysis and provided solid foundations for modern clinical neurochemistry. The model was theoretically derived from the laws of diffusion (the Fick's laws, see [2]) and empirically confirmed on large-scale patient cohorts, but the parametrization of the hyperbolic functions was originally performed with laborious manual iterations due to important, physiologically-driven constraints that have to be taken into account but are difficult to formalize. Details of this procedure are discussed in [3]. In the current paper, I show how to apply a numerical optimization algorithm to find those parameters. I need to emphasize that I am not attempting to readdress theoretical derivations of the blood-CSF

barrier diffusion. I am merely trying to explain the mathematical background for its optimization. Of course, virtually every statistical package today offers fast and reliable methods for non-linear regression; however, I believe that it is of paramount importance that neuroscientists interested in the CSF protein analysis understand the process of non-linear optimization and not only use commands built into a statistical software.

2. Materials and Methods

Consider function

$$f(x; a, b, c) = \frac{a}{b} \sqrt{x^2 + b^2} - c, \quad x > 0 \quad (1)$$

parametrized by three positive real numbers, a , b , and c , which we group into a vector, $\theta = (a, b, c)^T$. The first-order derivatives of this function with respect to the three parameters are:

$$\begin{aligned} \frac{\partial f(\theta; x)}{\partial a} &= \frac{1}{b} \sqrt{x^2 + b^2} \\ \frac{\partial f(\theta; x)}{\partial b} &= -ax^2 \left(b^2 \sqrt{x^2 + b^2} \right)^{-1} \\ \frac{\partial f(\theta; x)}{\partial c} &= -1. \end{aligned} \quad (2)$$

Now, we assume a study with N paired observations, $(x_i, y_i), i = 1, \dots, N$, with the vector of the explanatory variables, $\mathbf{x}$ (e.g. albumin quotients, Q_{Alb}), and the vector of the corresponding outcome variables, $\mathbf{y}$ (e.g. IgG quotients, Q_{IgG}), where a quotient is a dimensionless quantity obtained by division of the concentration of a given protein in the CSF by its concentration in the serum, $Q_x = \frac{[X_{CSF}]}{[X_{Serum}]}$. Our goal is to find the vector of the parameters, $\hat{\theta}$, minimizing

Euclidean norm of the vector of the residuals, $\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}$ or, stated equivalently, minimizing the sum of the squared vertical distances between the empirical observations, $\mathbf{y}$, and the model predictions, $\hat{\mathbf{y}}$. This is an unconstrained optimization problem in non-linear regression¹. Formally:

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}_+^3} \|\mathbf{r}\|_2. \quad (3)$$

This can be achieved, for example, with the Levenberg-Marquardt Algorithm (LMA) [4] [5].

To do so, first we define the Jacobian $N \times 3$ matrix, $\mathbf{J}$, such that its i -th row contains the three derivatives defined in Equation (2) evaluated at the i -th explanatory variable, x_i . We also define a dumping parameter, λ . After initialization of the vector of the model parameters and the dumping parameter, in a $(k+1)$ -

¹Stringently spoken, since $\hat{\mathbf{y}}$ has to be positive for all $x > 0$, this is a constrained problem, requiring $a > c$; however, since empirical observations in studies this model is developed for are always strictly positive, we may omit this constrain, only checking if the estimates indeed obey $\hat{a} > \hat{c}$.

th iteration of the algorithm the vector of the parameters is updated by:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \mathbf{h}^{(k)}, \quad (4)$$

where

$$\mathbf{h} = \left(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I} \right)^{-1} \mathbf{J}^T \mathbf{r}, \quad (5)$$

where $\mathbf{I}$ is the identity matrix (for notational simplicity, the superscripts referring to the iteration step were omitted). In each step of the algorithm, the dumping parameter, λ , is adjusted as follows. First, the error is evaluated at the new vector of the parameters. If the error has increased, the update is rejected and the dumping parameter is increased by some factor (say, 100). This is repeated until the error decreases below the error of the previous step. If the error has decreased as the result of the update, the updated parameters are accepted and the dumping parameter is decreased (say, by factor 100). The algorithm proceeds until convergence, which somehow reflects lack of further significant improvements (for example, when the norm of the updating vector, $\mathbf{h}$, gets below some predefined small positive number, say 10^{-12}).

3. Results

The algorithm was implemented in R (4.3.2)². It was first tested with one hundred repetitions of simulated datasets with 10,000 paired observations each, set such way that a heteroscedastic Normal random noise (with a constant CV of 10%) was added to the deterministic hyperbolic function. The empirical bias, variation, and the Mean Square Error (all multiplied by one thousand for better readability) are reported in **Table 1**.

Table 1. Results of the simulation study; the true values were: $a = 0.2$, $b = 0.1$, and $c = 0.05$.

Parameter	Mean	$10^3 \times \text{Bias}$	$10^3 \times \text{Var}$	$10^3 \times \text{MSE}$
$\hat{a}$	0.1998	-0.1589	0.0337	0.0337
$\hat{b}$	0.0999	-0.0909	0.0061	0.0061
$\hat{c}$	0.0499	-0.0932	0.0250	0.0251

With those results, clearly indicating that the estimates are characterized with very low bias and variance, the algorithm was applied on the real-life dataset kindly provided by H. Reiber. This dataset overlaps with the set used in his paper in 1994 to optimize the regression curve in the diagnostically most relevant region of $Q_{Alb} < 0.02$. Briefly, the samples were collected from patients aged between 0.5 and 75 years routinely diagnosed for neurologic conditions. Only subjects without intrathecal synthesis of immunoglobulins and without intracerebral haemorrhage were included in the study; CSF samples with blood contamination were excluded.

²The code is available at https://github.com/LewczukPiotr/Hyperbolic_Function.

The concentrations of the proteins were measured by automated nephelometry. All further details are given in [1]. With the arbitrarily chosen initial values $a^{(0)} = b^{(0)} = c^{(0)} = \lambda^{(0)} = 0.001$, the algorithm converged after 11 iterations and delivered the following estimates:

$$\begin{aligned}\hat{a} &= 0.0019425 \pm 0.0001714 \\ \hat{b} &= 0.0031210 \pm 0.0002414 \\ \hat{c} &= 0.0013078 \pm 0.0001258,\end{aligned}\tag{6}$$

with the standard errors resulting from well-known good approximation of the covariance matrix by $\frac{RSS}{N-p+1}(\mathbf{J}^T \mathbf{J})^{-1}$ at the convergence [6]. The same estimates were obtained with different initial values, confirming robustness of the algorithm. It needs to be emphasized that the estimates published in the original paper [1] are astonishingly similar to those obtained here. Figure 1 presents the empirical data points and the fitted hyperbolic curve.

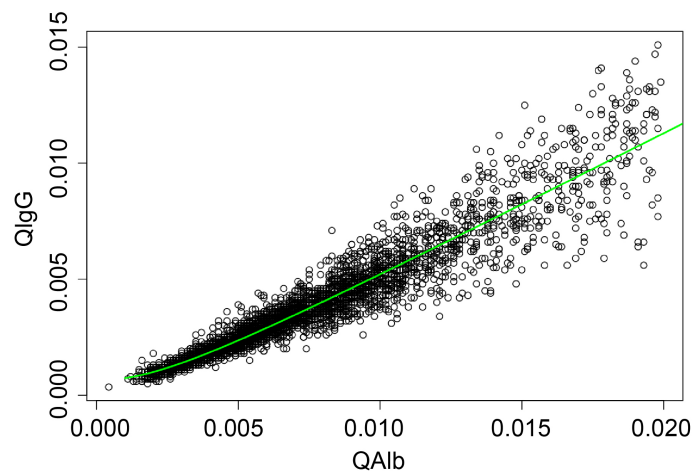


Figure 1. Empirical data from 4010 patients and the fitted hyperbolic curve.

4. Discussion and Conclusions

The function in Equation (1) is linear in a and c but non-linear in b , which makes the problem of its optimization non-linear. Several numerical optimization methods are available for fitting a non-linear model to a set of data [7]. For example, the Gradient Descent (GD) solves Equation (3) by iteratively updating the parameters of the model in the “downhill” direction of the objective function (the one to be minimized). The update in Equation (5) is replaced by:

$$\mathbf{h}_{GD} = \alpha \mathbf{J}^T \mathbf{r},\tag{7}$$

with the positive real number α defining the length of the step in the (steepest-)descent direction. The method is simple and computationally cheap, as it requires only the Jacobian matrix. However, this conceptual and computational simplicity comes at the cost of a large number of iterative steps required for con-

vergence. In the current setting, a simplistic GD with non-adaptive $\alpha = 10^{-5}$ needed about 100,000 iterations to converge and turned out very sensitive to the initial values. The Gauss-Newton Algorithm (GNA) relies on the assumption that the objective function is approximately quadratic near the optimal solution. Information on this (quadratic) curvature enters the algorithm as the second-order differential equations matrix, the Hessian matrix, of which $\mathbf{J}^T \mathbf{J}$ is often a good approximation. This results in a rapid local convergence. The update becomes:

$$\mathbf{h}_{\text{GNA}} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{r}. \quad (8)$$

The LMA varies the updates between the two optimization methods, taking advantages of both, by adaptation of the dumping parameter, λ , in each iteration. Small values make the LMA approach the GNA (as the term $\lambda \mathbf{I}$ becomes negligible compared to $\mathbf{J}^T \mathbf{J}$). For large values of λ , the LMA mimics the GD method (with $\alpha \approx 1/\lambda$). The intuition is that if the error increases we are likely far from the minimum we are looking for; therefore, we need to increase the dumping parameter in order to turn the algorithm into a simple GD. If the error decreases, on the other hand, we are probably getting closer to the minimum and we should emphasize the information on the local curvature of the optimization function.

Hyperbolic functions can be easily, yet erroneously, mixed with a linear function. Indeed, for very small, compared to x , positive b the term under the square root becomes $\sqrt{x^2 + b^2} \approx x$ and the hyperbolic function converges to a “linear approximation”, $\tilde{f}(x) = \frac{a}{b}x - c$. This, however, leads to an obvious problem, since for positive c , the linear function will become negative for small x , which is physiologically impossible. Further, it might be tempting to use statistical properties of the regression curve (e.g. the confidence or the prediction limits) to establish diagnostic references for the disease-related interpretations. This is not appropriate, because the confidence bands of a regression curve do not reflect the biological variability, they only show the region where the “true” curve is expected to lie. Instead, following Reiber’s idea, the upper limit of the Q_{IgG} can be defined by finding parameters of a hyperbolic function fitted to the largest empirical outcomes (stratified according to Q_{Aib}), again following the same principle as it is outlined above for the average curve. Finally, it needs to be emphasized that the diffusion model proposed by H. Reiber applies equally well to the whole range of $Q_{\text{Aib}} < 0.5$ (whereas the values beyond that limit are never seen in practice) and to other proteins (IgA, IgM, prothrombin), too, and the corresponding hyperbolic curves can be analogously parametrized with the LMA outlined here.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Reiber, H. (1994) Flow Rate of Cerebrospinal Fluid (CSF)—A Concept Common to Normal Blood-CSF Barrier Function and to Dysfunction in Neurological Diseases. *Journal of the Neurological Sciences*, **122**, 189-203. [https://doi.org/10.1016/0022-510x\(94\)90298-4](https://doi.org/10.1016/0022-510x(94)90298-4)
- [2] Crank, J. (1975) *The Mathematics of Diffusion*. 2nd Edition, Oxford University Press.
- [3] Reiber, H., Zeman, D., Kušnierová, P., Mundwiler, E. and Bernasconi, L. (2019) Diagnostic Relevance of Free Light Chains in Cerebrospinal Fluid—The Hyperbolic Reference Range for Reliable Data Interpretation in Quotient Diagrams. *Clinica Chimica Acta*, **497**, 153-162. <https://doi.org/10.1016/j.cca.2019.07.027>
- [4] Levenberg, K. (1944) A Method for the Solution of Certain Non-Linear Problems in Least Squares. *Quarterly of Applied Mathematics*, **2**, 164-168. <https://doi.org/10.1090/qam/10666>
- [5] Marquardt, D.W. (1963) An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, **11**, 431-441. <https://doi.org/10.1137/0111030>
- [6] Gavin, H. (2024) The Levenberg-Marquardt Algorithm for Nonlinear Least Squares Curve-Fitting Problems. <https://people.duke.edu/~hpgavin/lm.pdf>
- [7] Nocedal, J. and Wright, S.J. (2006) *Numerical Optimization*. 2nd Edition, Springer.