On damping parameters of Levenberg-Marquardt algorithm for nonlinear least square problems

A O Umar¹,², I M Sulaiman¹, M Mamat¹, *, M Y Waziri³, N Zamri¹

¹Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Kuala Terengganu, 21300, Terengganu, Malaysia.
²Department of Mathematics, College of Agriculture Zuru, Kebbi State, Nigeria.
³Department of Mathematics, Faculty of Physical Sciences, Bayero University, Kano, Nigeria.

Corresponding author: must@unisza.edu.my

Abstract. The Levenberg-Marquardt (LM) algorithm is a widely used method for solving problems related to nonlinear least squares. The method depends on a nonlinear parameter known as self-scaling parameter that affects the performance of the algorithm. In this paper we examine the effect of various choice of parameters and of relaxing the line search. Numerical results obtained are used to compare the performance using standard test problems which show that the proposed alternatives are promising.

1. Introduction

The nonlinear unconstrained least squares problem can be classified as a special case of the general optimization problem in $\mathbb{R}^n$ designed to find a global minimizer of the sum of square of the $m$ nonlinear function

\[
\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} \sum_{i=1}^{m} f_i^2(x) = \frac{1}{2} F(x)^T F(x) = \frac{1}{2} \|F(x)\|^2
\]

(1)

where $F: \mathbb{R}^n \to \mathbb{R}^m$ ($m > n$), $f_i: \mathbb{R}^n \to \mathbb{R}$ and $F$ is a function with continuous second partial derivatives. Problem (1) frequently arises in applications including imaging problems, data assimilation, experimental design, optimal control, and curve-fitting [1-9]. In solving these types of problems, numerous researches aimed at developing specialized algorithms that attempt to exploit the nonlinear least-square objective structure.

Most algorithms for solving (1) are based on the recursion formula

\[
x_{k+1} = x_k + \alpha_k d_k
\]

(2)

where $\alpha_k$ is a positive step-size parameter calculated via a suitable line search and $d_k$ is the search direction determined by

\[(J(x)^T J(x) + B(x))^{-1} d_k = -J(x)^T F(x)\]

(3)

Equation (3) is motivated by the fact that the function is quadratic and it represents the general form of nonlinear least squares algorithm. The gradient of the objective function is given as $f(x)^T F(x)$ and...
\[ J(x) = \nabla F(x)^T \] is the Jacobian of the \( F \) at \( x \). The Hessian, \( \nabla^2 f(x) \) is given as \( f(x)^T J(x) + B(x) \) where \( B(x) = \sum_{i=1}^{m} f_i(x) \nabla^2 f_i(x) \).

The solution to (1) was suggested by Levenberg [10] and Marquardt [11] using the following iterative sequence
\[ x_{k+1} = x_k - (J_k^T J_k + \mu_k I)^{-1} g_k \tag{4} \]
where \( g_k = J_k^T f_k \) and \( \mu_k \) is a positive regularization or damping parameter that is specifically added to remedy the possibility of the Jacobian matrix in becoming rank-deficient. Most of the algorithms developed to improve (4) involves parameter \( \mu_k \) selection by nonlinear least square problems.

Levenberg [10] choose the scaling parameter \( \mu_k \) as \( \mu_k = I \) (identity matrix) or \( \mu_k = \sqrt{\text{trace}(J_k^T J_k)} \) (the square root of the diagonal of \( J_k^T J_k \)). Marquardt [11] used \( \mu_k = 1 \) for the scaling matrix. This scaling has been extensively employed as a procedure for solving ill-conditioned linear least square problems. Also, Osborne [12] used \( \mu_k = f_k^T f_k \) as the scaling parameter and Fletcher [13] uses safeguarded quadratic interpolation to increase \( \mu_k \) by choosing \( \mu_k = (J_k^T J_k)^{-1} \) or \( \mu_k = 1 / \| J_k^T J_k \|_\infty \) or \( \mu_k = 1 / \text{trace}(J_k^T J_k)^{-1} \). In this paper, we introduced a matrix norm of \( J_k^T J_k \) as an alternative self-scaling parameter for use in LM algorithm for (1). We derive the motivation because most of the damping parameters used are matrix representation. Hence, various matrix norm parameters are considered that were not used previously to determine the solution of (1). The proposed scaling parameters are extended to solve a set of nonlinear algebraic equations.

The rest parts of this paper is structured as follows. Section 2 presents the description of nonlinear square minimization model. In section 3, we present the method under consideration in which the scaling parameter is used, the algorithms, and the various types of parameters proposed. And in section 4, we give some of the test problems considered and the corresponding results. Finally, in section 5, we present the discussion and conclusion.

2. The general methods (description of the methods)
Assuming that \( F \) is the function to be minimized, and knowing that both line search and trust region methods involve the iterative minimization of a quadratic local model
\[ Q(p) = \nabla F(x_k)^T p + \frac{1}{2} p^T H_k p \]

Then from (1), and assuming that \( f(x) = \frac{1}{2} F(x)^T F(x) \), we determine the derivative of \( F(x) \) which is simply the Jacobian matrix \( f(x) \in \mathbb{F}^{m \times n} \).

\[ \nabla f(x) = \sum_{i=1}^{m} f_i(x) \nabla f_i(x) = J(x)^T F(x) \tag{5} \]
\[ \nabla^2 f(x) = \sum_{i=1}^{m} (\nabla^2 f_i(x)). \nabla f_i(x)^T + f_i(x) \nabla^2 f_i(x) \tag{6} \]
\[ = J(x)^T J(x) + S(x) \]

where \( S(x) = \sum_{i=1}^{m} f_i(x) \nabla^2 f_i(x) \). Considering the Newton direction
\[ H(x) d = -g(x) \]
and replacing \(-g(x)\) with (2.1) and \(H(x)\) with (6) and assuming that \(\sum_{i=1}^{m} f_i(x) \nabla^2 f_i(x) = 0\) we arrive at

\[
(J(x)^T J(x))d_k = -J(x)^T F(x) \tag{7}
\]

which is widely known as the Gauss-newton method. The difficulty in solving (1) as observed in Gauss-Newton method (7) arises when \(J_k^T J_k\) is singular, or equivalently when \(J\) has linearly dependent columns, because then (5) don’t have a unique minimizer.

Similarly, replacing \(-g(x)\) with (5) and \(H(x)\) with (6) and also approximating \(S(x) \approx \lambda I\) we get

\[
x_{k+1} = x_k - (J_k^T J_k + \mu I)^{-1} g_k \tag{8}
\]

which is called the Levenberg-Marquardt method. However, the issue with (8) is the choice of \(\mu\) [10-13].

We present some useful definitions as follows.

**Definition 2.1**
A norm is a function which assigns a positive length or size to all vectors in a vector space, other than the zero vector.

**Definition 2.2**
The Euclidean norm of a matrix \(A\), denoted as \(\|A\|\), is defined as

\[
\|A\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}^2}
\]

Notation in all cases \(\|\cdot\|\) refers to the Euclidean norm \(l_2\) vector norm or to the induced operator norm.

**Definition 2.3** [14]
Let \(M_n(R)\) denote the real \(n \times n\) matrices, \(I \in M_n(R)\) the identity matrix, \(D \in M_n(R)\) the set of diagonal matrices \(D = \text{diag}(d_1, d_2, ..., d_n)\) with \(\max |d_k| = 1\) and \(e_k \in R^n\) the vector \((0,0,...,1_k,0,...,0)^T\). A matrix norm on \(M_n(R)\) is a non-negative real valued function \(\|\|\) defined on \(M_n(R)\) satisfying:

- \(\|A\| = 0\) if and only if \(A = 0\)
- \(\|\lambda A\| = |\lambda| \|A\|\) for \(A \in M_n(R)\) and \(\lambda \in R^n\)
- \(\|A + B\| \leq \|A\| + \|B\|\) for \(A, B \in M_n(R)\).

3. **Levenberg-Marquardt method (LM)**
A popular and efficient method for solving (1) and (3) is the Levenberg-Marquardt (LM) method [10, 11]. This algorithm gives a method that depends nonlinearly on a damping parameter \(\mu_k\) and uses a search direction that is a solution of the linear set of equations defined as

\[
(J(x)^T J(x) + \mu_k I)d_k = -J(x)^T F(x) \tag{9}
\]

where \(I\) is an identity matrix and the scalar \(\mu_k\) controls both direction of \(d_k\) and the magnitude. The direction is given by

\[
d_k = -(J_k^T J_k + \mu_k I)^{-1} g_k \tag{10}
\]
where \( f = f(x) \), \( F = F(x) \), \( g_k = j_k^TF_k \) and \( \mu_k \) is a positive regularization or damping parameter that is added specifically to remedy the possibility of the Jacobian matrix being rank-deficient and has the following effects:

a) for all \( \mu > 0 \) the coefficient matrix is positive definite, thereby ensuring that \( d_k \) is a descent direction

\[
d_k^TF'(x) = d_k^T(j_k^TF_k) = -d_k^T(j_k^T(j_k^Tj_k + \mu_k I)d_k < 0
\]

(11)

b) for large values of \( \mu \), we get

\[
d_k \approx -\frac{1}{\mu}g = -\frac{1}{\mu}F'(x)
\]

that is, a little step in the direction of steepest descent. This is suitable in instances when the current iterative point is far from the solution.

c) if the value of \( \mu \) is small, then eqn(7) = eqn(10), for \( x \) close to \( x^* \), and thus, a good step in the final iteration stages.

Clearly, the \( \mu_k \) influences both the step-size and the direction. The choice of the starting value of \( \mu \) ought to be associated to the element size in \( A_0 = J(x_0)^TJ(x_0) \) e.g by allowing \( A_0 = \tau \max\{a_{ii}\} \) [15]. Since a scalar matrix \( \mu_k \) is added, the problem of choosing the \( \{\mu_k\} \) arises in such a way as to retain the Gauss-Newton method’s local convergence properties [16]. Note that the local convergence depends on the magnitude of \( f(x) \) relative to the nonlinearity of \( F \) and the spectrum of \( J(x)^TJ(x) \) [16] that is when the residuals are large. The convergence analysis is reported by [12, 17, 18].

The major drawback in the execution of the LM algorithm is an effective stratagem for controlling the size of \( \mu_k \) at every iterative point to enable it to be efficient for a broad spectrum of problems [19]. Similarly, the choice of \( \mu_k \) also affects the step-size. For further references recent approaches for nonlinear problems and applications, please see [20-22].

In order to address the problem and knowing that many variants of the LM method have been implied via numerous strategies to select \( \mu \). In this paper, we consider three different parameters for \( \mu_k \) for solving nonlinear least square problems. The anticipation is to improve on the performance of the scaling parameter in (5). \( \mu_k \) is chosen as follows and referred to as P1, P2 and P3.

\[
\mu_k = \frac{\|j_k\|^2}{\|J_k\|^2} \quad \text{P1}
\]

(12)

\[
\mu_k = \frac{\|j_k^TJ_k\|^2}{\|J_k^TJ_k\|^2} \quad \text{P2}
\]

(13)

\[
\mu_k = \|j_k^Tj_k\| \quad \text{P3}
\]

(14)

and the performance are compared. We now generate an iterative sequence \( \{x_n\} \) using the algorithm defined below.

**Algorithm:** Levenberg Marquarts (LM)

Step 1. Initialize

Step 2. Determine the starting point \( x_0 \) and for \( k = 0,1,2 \ldots \)

Step 3. Calculate the starting Jacobian matrix \( J_k \) and \( \mu_k \)
Step 4. Compute $[\frac{\partial}{\partial x_k} f_k + \mu_k I] S_k = -\frac{\partial^2}{\partial x_k^2} F_k$ where $I$ is an identity matrix
Step 5. Set $x_{k+1} = x_k$ and $f_{k+1} = f_k$
Step 6. Repeat step 3 to step 5 with $k = k + 1$ until $\epsilon \leq 10^{-5}$ are achieved.

4. Numerical results
In this section, numerical performance for various damping parameters and new proposed parameters are compared for problems of nonlinear least squares. The comparison is based on the number of iterations and execution (CPU) time. We also considered $\epsilon = 10^{-5}$ and the gradient value as stopping criteria. The algorithms are coded on MATLAB R2015a having the capacity of double precision. Below are the problems considered in the computation.

**Problem 1.** Extended Rosenbrock function [23]
- a) $c = 100$, $n = 2$, and $m = 2$
- b) $f(x) = \sum_{i=1}^{n/2} c(x_{2i-1} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2$
- c) Minimum value: $f^* = 0$ at $x^* = (1,1)$
- d) Initial guess: $x^0 = (-1.2, 1.0)$

**Problem 2.** Extended Freudenstein & Roth functions [24]
- a) $n = 2$ and $m = 3$
- b) $f(x) = \sum_{i=1}^{n/2} (-13 + x_{2i-1} + ((5 - x_{2i})x_{2i} - 2)x_{2i})^2 + (-29 + x_{2i-1} + ((x_{2i} + 1)x_{2i} - 14)x_{2i})^2$
- c) $f^* = 0$ at $x^* = (5,4)$ is the minimum value:
- d) Initial guess: $x^0 = (0.5, -2)$

**Problem 3:** Powell singular function [25]
- a) $n = 4$, $m = 4$
- b) $f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 + x_4)^4$
- c) $f^* = 0$ at $x^* = (0, 0, 0, 0)$ is the minimum value:
- d) Initial guess: $x^0 = (3, -1, 0, 1)$

**Problem 4:** Jennrich and Sampson function [26]
- a) $n = 2, m \geq n$
- b) $f_i(x) = 2 + 2l - (\exp[ix_1] + \exp[ix_2])$
- c) $x_0 = (0.3, 0.4)$
- d) $f = 124.362 .. at x_1 = x_2 = 0.2578 .. for m = 10$

**Problem 5.** Box three-dimensional function [27]
- a) $n = 3, m \geq n$ variable
- b) $f_1(x) = \exp[-t_1x_1] - \exp[-t_1x_2] - x_3(\exp[-t_1] - \exp[-10t_1])$; where $t_i = (0.1)i$
- c) $x_0 = (0, 10, 20)$
- d) $f = 0$ at $(1,10,1), (10,1,-1)$
and wherever $x_1 = x_2$ and $x_3 = 0$

**Problem 6.** Brown and Dennis function [16]
- a) $n = 4, m \geq n$ variable
- b) $f_1(x) = \exp[-t_1x_1] + \exp[-t_1x_2] + x_3 + x_4\sin(t_i) - \cos(t_i))^2$; where $t_i = i/5$
- c) $x_0 = (25.5, 5, -1)$
- d) $f = 85822.2 .. if m = 20$

**Problem 7.** Kowalik and Osborne function [28]
a) \( n = 4, m = 11 \)

b) \( f_1(x) = y_i - \frac{x_i (u_i^2 + u_i x_2)}{(u_i^2 + u_i x_2 + x_i)} \)

where

| \( i \) | \( y_i \) | \( u_i \) | \( i \) | \( y_i \) | \( u_i \) |
|-------|-------|-------|-------|-------|-------|
| 1     | 0.1957 | 4.0000 | 7     | 0.0456 | 0.1250 |
| 2     | 0.1947 | 2.0000 | 8     | 0.0342 | 0.1000 |
| 3     | 0.1735 | 1.0000 | 9     | 0.0323 | 0.0833 |
| 4     | 0.1600 | 0.5000 | 10    | 0.0235 | 0.0714 |
| 5     | 0.0844 | 0.2500 | 11    | 0.0246 | 0.0625 |
| 6     | 0.0627 | 0.1670 |

c) \( x_0 = (0.25, 0.39, 0.415, 0.39) \)

d) \( f = 3.07505 \ldots 10^{-4} \)

\[ f = 1.02734 \ldots 10^{-3} \] at \( +\infty, -14.07 \ldots, -\infty, -\infty \)

**Problem 8.** Helical Valley function [13]

a) \( n = 3, m \geq n \) variable

b) \( f_1(x) = 100\left[ (x_3 - 10\theta(x_1, x_2))^2 + [r(x_1, x_2) - 1]^2 \right] + x_2^2 \)

where \( \theta(x_1, x_2) = \begin{cases} 
    (2\pi)^{-1}\arctan^{x_2/x_1} & \text{for } x_1 > 0 \\
    \left( \frac{1}{2} + (2\pi)^{-1}\arctan^{x_2/x_1} \right) & \text{for } x_1 < 0
\end{cases} \)

\( (x_1, x_2) = (x_1^2 + x_2^2)^{1/2} \)

c) \( x_0 = (1, -1, 0) \)

**Table 1:** A comparison of various parameter \( \mu_k \) in terms of Number of iterations.

| Problems | P1 | P2 | P3 | LEV | MARQ | FLETCHER |
|----------|----|----|----|-----|------|----------|
| 1        | 13 | 14 | 14 | 13  | 14   | 14       |
| 2        | 65 | 169| 169| 65  | 169  | 169      |
| 3        | 17 | 18 | 18 | 17  | 18   | 18       |
| 4        | 190| 20 | 20 | 190 | 20   | 20       |
| 5        | 93 | 329| 329| 93  | 329  | 329      |
| 6        | 5  | 5  | 5  | 5   | 5    | 5        |
| 7        | 12 | 12 | 12 | 12  | 12   | 12       |
| 8        | 0.055307 | 0.107589 | 0.064859 | 0.062254 | 0.116669 | 0.652948 |

**Table 2:** A comparison of various parameter \( \mu_k \) in terms of CPU time.

| Problems | P1 | P2 | P3 | LEV | MARQ | FLETCHER |
|----------|----|----|----|-----|------|----------|
| 1        | 0.120523 | 0.233917 | 0.388981 | 0.170719 | 0.207157 | 0.220070 |
| 2        | 0.069684 | 0.237261 | 0.082540 | 0.062810 | 0.057172 | 0.062623 |
| 3        | 0.140450 | 0.147964 | 0.180425 | 0.160564 | 0.157061 | 0.147887 |
| 4        | 0.447850 | 0.728587 | 0.681286 | 0.33266 | 0.693351 | 0.599557 |
| 5        | 0.170491 | 0.905273 | 0.937456 | 0.232035 | 1.19805 | 0.929092 |
| 6        | 0.226939 | 0.239796 | 0.234014 | 0.210587 | 0.213467 | 0.194045 |
| 7        | 8    | 0.132568 | 0.167880 | 0.201884 | 0.134270 | 0.151592 | 0.201089 |
The algorithm used for the implementation of the various damping parameters were successfully implemented using MATLAB 2015a. The computational results are discussed here. The tolerance level considered in the implementation of the algorithms is $10^{-5}$.

Problem one was initially discussed by [29] and is now frequently used as a test function. The solution in terms of number of iterations are described in Table 1 and that of the execution (CPU) time in Table 2 shows that (P1) and Levenberg parameter has the same number of iteration but with different CPU time in which P1 attains the solution faster. These parameters can be classified as belonging to the same category as compared to the parameters P2, P3, Marquardt and Fletcher. The performance of the new parameters compares favourably with [10,11,13].

Problem two is from [24]. The solution as presented in Table 1 number of iterations shows that P1 excellently agrees with Levenberg [10] parameter, but it attains the solution faster as seen from Table 2. P2, P3 can be categorized in the same class with Marquardt and Fletcher as seen in Tables 1 and 2. The results of the remaining Problems considered shares similar performance as that of problem 2 discussed earlier. The only significant observation is that in problem 6 the tolerance used is $10^{-3}$, and the performance is as stated in Table 1. Please, (see, [30-33]) for further discussion on least square methods.

5. Conclusion
In this paper, we present alternative damping parameters for solving least square problems. The parameters can be classified into two groups namely, group A and group B. It is evident from Tables 1 and Table 2 that the performance of Group A is far better that of group B, except problem 4 that proved otherwise. The computational results obtained has shown that the alternative damping parameters proves to compete favourably with existing parameters used in other methods for the type of test problems considered.

References
[1] Barz, T., Körkel, S., Wozny, G. Nonlinear ill-posed problem analysis in model-based parameter estimation and experimental design. Comps & Chemical Engineering, 77, 24-42. (2015).
[2] Cornelio, A. Regularized nonlinear least squares methods for hit position reconstruction in small gamma cameras. Applied mathematics and computation, 217(12), 5589-5595. (2011).
[3] Gratton, S., Lawless, A. S., Nichols, N. K. Approximate Gauss–Newton methods for nonlinear least squares problems. SIAM Journal on Optimization, 18(1), 106-132. (2007).
[4] Golub, G., Pereyra, V. Separable nonlinear least squares: the variable projection method and its applications. Inverse problems, 19(2), R1. (2003).
[5] Henn, S. A Levenberg–Marquardt scheme for nonlinear image registration. BIT Numerical Mathematics, 43(4), 743-759. (2003).
[6] Kim, S. J., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D. An interior-point method for large-scale $\ell_1$-regularized least squares. IEEE journal of selected topics in signal processing, 1(4), 606-617. (2007).
[7] Li, J., Ding, F., Yang, G. Maximum likelihood least squares identification method for input nonlinear finite impulse response moving average systems. Mathematical and Computer Modelling, 55(3-4), 442-450. (2012).
[8] Tang, L. M. A regularization homotopy iterative method for ill-posed nonlinear least squares problem and its application. In Applied Mechanics and Materials (Vol. 90, pp. 3268-3273). Trans Tech Publications Ltd. (2011).
[9] Sulaiman, I.M. Mamat, M. Waziri, M.Y., Mohamed, M.A., Mohamad, F.S. Solving Fuzzy Nonlinear Equation via Levenberg-Marquardt Method. Far East Journal of Mathematical Sciences. 103(10), 1547-1558. (2018)
[10] Levenberg, K. A method for the solution of certain non-linear problems in least
squares. *Quarterly of applied mathematics*, 2(2), 164-168. (1944).

[11] Marquardt, D. W. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2), 431-441. (1963).

[12] Osborne, M. R. Nonlinear least squares—the Levenberg algorithm revisited. *The ANZIAM Journal*, 19(3), 343-357. (1976).

[13] Fletcher, R., Powell, M. J. A rapidly convergent descent method for minimization. *The computer journal*, 6(2), 163-168. (1963).

[14] Malek-Shahmirmazdi, M. A characterization of certain classes of matrix norms. *Linear and Multilinear Algebra*, 13(2), 97-99. (1983).

[15] Madsen, K., Nielsen, H. B. Tingleff, O. Methods for non-linear least squares problems. . (2004)

[16] Brown, K. M., Dennis, J. E. Derivative free analogues of the Levenberg-Marquardt and Gauss algorithms for nonlinear least squares approximation. *Numerische Mathematik*, 18(4), 289-297. (1971).

[17] Powell, M. J. D. Convergence properties of a class of minimization algorithms. In *Nonlinear programming* 2 (pp. 1-27). Academic Press. (1975).

[18] Moré, J. J. The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis* (pp. 105-116). Springer, Berlin, Heidelberg. (1978).

[19] Branch, M. A., Grace, A. *MATLAB: Optimization Toolbox: User’s Guide*. Math Works. (1996).

[20] Mamat, M., Sulaiman, I.M., Ghazali, P.L. Sambas, A. An accelerated scheme for solving parameterized fuzzy nonlinear equations. International Journal of Advanced Science and Technology, 29(05), 248-255. (2020).

[21] Sulaiman, I.M., Mamat, M., Ghazali, P.L. Foziah, H.M. Kamfa, K., Benny, P. An Improved Shamanski Method for Solving Nonlinear Equation. Journal of Advanced Research in Dynamical and Control Systems 12(2): 591-594. (2020)

[22] Kamfa, K. Waziri, M.Y. Sulaiman, I.M. Mamat, M. Ibrahim, H.M. A Quasi-Newton Like Method via Modified Rational Approximation Model for Solving System of Nonlinear Equation. Journal of Advanced Research in Dynamical and Control Systems 12(2):1019-1026. (2020)

[23] Moré, J. J., Garbow, B. S., Hillstrom, K. E. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software (TOMS)*, 7(1), 17-41. (1981).

[24] Freudenstein, F., Roth, B. Numerical solution of systems of nonlinear equations. *Journal of the ACM (JACM)*, 10(4), 550-556. (1963).

[25] Powell, M. J. D. An iterative method for finding stationary values of a function of several variables. *The Computer Journal*, 5(2), 147-151. (1962).

[26] Jennrich, R. I., Sampson, P. F. Application of stepwise regression to non-linear estimation. *Technometrics*, 10(1), 63-72. (1968).

[27] Box, M. J. A comparison of several current optimization methods, and the use of transformations in constrained problems. *The Computer Journal*, 9(1), 67-77. (1966).

[28] Kowalik, J., Osborne, M. R. Methods for Unconstraint Optimizatton Problems. Elservmr North-Holland, New York. (1968).

[29] Rosenbrock, H. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3), 175-184. (1960).

[30] Fraley, C. *Algorithms for nonlinear least-squares problems* (No. SOL-88-16). STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB. (1988).

[31] Hebdon, M. D. An algorithm for minimization using exact second derivatives. Atomic energy Research Establishment report TP515, Harwell,England. (1973).

[32] Mohammad, H., Waziri, M. Y., Santos, S. A. A brief survey of methods for solving nonlinear least-squares problems. *Numerical Algebra, Control & Optimization*, 9(1), (2018).

[33] More, J. J. The Levenberg-Marquardt algorithm: implementation and theory, in numerical Analysis. Proceedings of the 1977 Dundee conference on numerical analysis (Berlin, Heidelberg New York, Tokyo) (G. A. Watson, ed.), Lecture notes in mathematics 630,
Springer Verlag 1978. pp. 105-116. (1977)