Journal of Numerical Mathematics and Stochastics, **9** (1) : 80-94, 2017 © ***JNM@S***
http://www.jnmas.org/jnmas9-6.pdf Euclidean Press, LLC
Online: ISSN 2151-2302

On Computing the Regularization Parameter for the Levenberg-Marquardt Method Via the Spectral Radius Approach to Solving Systems of Nonlinear Equations

Y. B. MUSA[1], **M. Y. WAZIRI**[2], **and A. S. HALILU**[1]

[1]Department of Mathematics and Computer Sciences, Sule Lamido University, Kafin Hausa, Jigawa, Nigeria;[2]Department of Mathematical Sciences, Faculty of Science, Bayero University Kano, Kano, Nigeria, E-mail: balarabemusa.yau@jsu.edu.ng

Abstract. *In this paper, we present a Levenberg-Marquardt-type method for solving symmetric nonlinear systems of equations. The pertaining regularization parameter is computed using the spectral radius approach. Convergence results for the proposed method are also provided. A reported numerical performance of the proposed algorithm on some benchmark problems demonstrates its reliability, efficiency and extreme robustness.*

Key words : Nonlinear System of Equations, Levenberg-Marquardt Method, Regularization, Spectral Radius, Global Convergence.

AMS Subject Classifications : 65H11, 65K05, 65H12, 65H18

1. Introduction

This work deals with the system of nonlinear equations:

$$F(x) = 0, \tag{1}$$

where

$$F : \mathbb{R}^n \to \mathbb{R}^n, \tag{2}$$

i.e. $F = (f_1, f_2, f_3, \ldots, f_n)^T$ is a continuously differentiable system of functions. We denote the Jacobian by $J(x) = F'(x)$, $\forall\, x \in \mathbb{R}^n$. When iterated, it is denoted as J_k , which is also assumed to be Lipschitz continuous.

This problem is one of the cornerstones in computation and applied mathematics, with applications in physics, engineering , technology , economics , industries, etc. [1, 6, 9, 10, 14, 15]. Various methods have been widely implemented for solving (1). Typical examples are the Newton, Gauss-Newton, Trust Region, quasi-Newton methods, etc. [2, 3, 11, 16]. As (1) is nonlinear, it may have no solution at all. In this work, we assume, however, that the solution space of (1) is not empty.

The Levenberg-Marquardt method is a classical method for solving nonlinear system of equations. In this method, the trial step d_k is computed at each iteration as

$$d_k = -(J_k^T J_k + \lambda_k I)^{-1} J_k^T F_K \; . \qquad (3)$$

The Levenberg-Marquardt parameter λ_k or LM parameter is introduced to overcome the difficulty when $J_k^T J_k$ is singular or very close to singularity [1, 8, 10, 12]. The parameter λ_k is updated in every iteration. Like the Newton iterative method, it is well known that the Levenberg-Marquardt method is also of quadratic convergence, when the Jacobian matrix is nonsingular and Lipschitz continuous at the solution of (1).

Here it is vital to mention that Fan and Yuan proposed in [9] the LM parameter $\lambda_k = \|F_k\|$ and advanced a related algorithm that has quadratic convergence. These authors also proved, in [7], that if this parameter is chosen as $\lambda_k = \|F_k\|^\delta$, $\delta \in (0,2]$, under a local error bound condition, then the convergence order of the LM algorithm would be $\min\{1+\delta, 2\}$.

Fan also introduced in [5] a Modified Levenberg-Marquardt method (MLM), with cubic convergence, where the LM parameter was chosen as $\lambda_k = \mu_k \|F_k\|^\delta$ with $\mu > 0$. Then Karas and Santos chose the LM parameter, in [10], as $\min\{\mu_k^+, \mu_k^-\}$ when

$$\mu_k^- = \frac{L_k}{4}\left(2\|F_k\| + \sqrt{4\|F_k\|^2 + \|P_k(F_k)\|^2}\right), \qquad \mu_k^+ = \frac{2+\sqrt{5}}{4} L_k \|F_k\|, \qquad (4)$$

where P_k is the projection onto the range of the J_k matrix.

In the present work, we employ the spectral radius approach for computation of a novel efficient and reliable LM parameter. Our proposed method exhibits remarkable advantages over the ones presented in [8, 10].

The paper is organized as follows: in section 2, some preliminaries are presented. Section 3 contains the algorithm for the proposed method. Convergence results are presented in section 4. In Section 5, we report some numerical results to compare the new algorithm with that of [10] and [8]. Then a final remark is presented in the last section.

2. Technical Results

Definition 2.1. (Spectral radius) Let A be an $n \times n$ matrix. The spectral radius of A, denoted as $\rho(A)$, is defined as $\rho(A) = max\{|\lambda_i| \,/ \lambda \in \sigma(A)\}$, where $\sigma(A)$ is the set of all eigenvalues of the matrix A.

Lemma 2.1. *For any $A \in R^{m\times n}, b \in R^m$, $m \geq n$ and $\mu > 0$,*

$$\|(A^T A + \mu I)^{-1} A^T b\| \leq \frac{1}{2\sqrt{\mu}} \|P_{R(A)} b\|,$$

where R(A) is the range of A and $P_{R(A)}$ is the orthogonal projection onto this subspace.

Proof. Let $b' = P_{R(A)} b$ and $S = (A^T A + \mu I)^{-1} A^T b$. It is noticeable that

$$(A^T A + \mu I)s = A^T b' \; . \qquad (5)$$

Using Singular Value Decomposition (SVD) of A, we have U and V unitary matrices $m \times m$ and $n \times n$ respectively, such that $U^T AV = D, d_{i,j} = \sigma_i, i = j$ and $\sigma_i = 0$ otherwise. Note that $V^T = V^{-1}$, $U^T = U^{-1}$ and $V^T A^T AV = (U^T AV)^T U^T AV = D^T D$.

Now, multiplying (5) by V^T and using the substitutions $\tilde{s} = V^T s$, $\tilde{b} = U^T b'$ in (5), we can conclude that $D^T \tilde{b} = V^T (A^T A + \mu i) V\tilde{s} = (D^T D + \mu i)\tilde{s}$.

If $n \le m$, then, by (5), we have $\tilde{s}_i = \frac{\sigma_i}{\sigma_i^2 + \mu}\tilde{b}_i$, $i = 1, 2, \ldots, n$. Moreover, since $t/(t^2 + \mu) \le \frac{1}{(2\sqrt{\mu})}$ for $\mu > 0$ and $t \ge 0$, we have $\|\tilde{s}\| \le \frac{1}{2\sqrt{\mu}}\|b'\|$. Finally, as $\|\tilde{s} = \|s\|$, and $\|\tilde{b}\| = \|b'\|$, the conclusion of the proof is directly reached. ■

Theorem [10] **2.1**. *For any* $\mu > 0$, $\|s\| = \|(A^TA + \mu I)^{-1}A^Tb\| \le \frac{1}{2\sqrt{\mu}}\|P(F(x))\| \le \frac{1}{2\sqrt{\mu}}\|F(x)\|$, *where P stands for orthogonal projection onto the range of* $J(x)$.

For the details of the proof for this theorem, we refer the reader to lemma 2.1 and [10].

Theorem 2.2. *Let A be an* $n \times n$ *square matrix , and let* $\|.\|$ *be a consistent norm on* $R^{n\times n}$. *Then,* $\rho(A) \le \|A\|$.

Proof. Let V be a vector norm consisten with the matrix norm $\|.\|$ and let $\sigma(A)$ be the spectrum of the matrix A. If (λ, x) is an eigenpair of A with $V(x) = 1$. Then,

$$\|A\| = \|A\|V(x) \ge V(Ax) = V(\lambda x) + |\lambda|V(x) = |\lambda|. \tag{6}$$

Since λ is an arbitrary eigenvalue of A, we have $\|A\| \ge \max_{\lambda\in\sigma(A)} \{|\lambda|\} = \rho(A)$ and thus,

$\rho(A) \le \|A\|$. ■

2.1. **Our choice of μ**

Considering the LM directions in theorem 2.1 and if it is equated to d, we have

$$d = (A^TA + \mu I)^{-1}A^TF(x),$$

$$\begin{aligned}\|d\| &= \|-(A^TA + \mu I)^{-1}A^TF(x)\| \\ &\le \|(A^TA + \mu I)^{-1}\|\|A^TF(x)\| \\ &\le \frac{\|A^TF(x)\|}{\|A^TA+\mu I\|} \le \frac{\|A^T\|\|F(x)\|}{\|A^TA+\mu I\|}.\end{aligned} \tag{7}$$

From the above equation and lemma 2.1, theorems 2.1 and 2.2, we have

$$\frac{\|A^T\|\|F(x)\|}{\|A^TA+\mu I\|} \le \frac{1}{2\sqrt{\mu}}\|F(x)\| . \tag{8}$$

Clearly,

$$\mu \ge \frac{1}{4}[\frac{\rho(Q)}{\rho(A)}]^2, \tag{9}$$

where $Q = A^TA + \mu I$.

Here, our choice for μ is

$$\mu = \delta_k\frac{L}{4}[\frac{\rho(Q)}{\rho(A)}]^2, \tag{10}$$

where $\delta = \frac{1}{k^k}$, $L > 0$

3. Algorithm

Now we propose a new Levenberg-Marquadrt algorithm denoted as (SRLM) in which the regularization parameter is computed via the spectral radius approach.

Algorithm 3.1. Input: $x_0 \in \mathrm{R}$, $\beta \in (0,1), \eta \in [0,1), L_0 > 0, \delta > 0$ and $\sigma \geq 0$ with $L_0 \geq \sigma$
1. $k \leftarrow 0$
2. while $J_k^T F_k \neq 0$ do, where $F_k = F(x_k), J_k = J(x_k)$
3. compute $\rho(J_k) = max|\lambda_i|$ where λ_i are the eigenvalues of the matrix
   J_k and $\rho(J_k)$ is its spectral radius
4. set $Q = J_k^T J_k + \mu I, \mu > 0$
5. $\rho(Q) = max|\lambda_i|$, where λ_i are the eigenvalues of the matrix Q and $\rho(Q)$ is its spectral radius
6. $\mu_k = (\delta_k L_k) / 4\left[\dfrac{\rho(Q)}{\rho(J_k)}\right]^2$ where $L_k > 0$ and $\delta_k = 1/k^k$ for $k \geq 1$
7. compute $d_k = -(J_k^T J_k + \mu_k I)^{-1} J_k^T F_k$
8. $t \leftarrow 1$
9. while $\|F(x_k + td_k)\|^2 > \|F_k\|^2 + \beta t \langle d_k, J_k^T F_k \rangle$ do
10. $t \leftarrow t/2$
11. end while
12. $t_k = t$
13. $z_k = x_k + (t_k + \frac{1}{2}) d_k$
14. compute $F_{Z_k} = F(z_k), J_{Z_k} = J(z_k)$
15. set $x_{k+1} = z_k - (J_{Z_k}^T J_{Z_k} + \mu_k I)^{-1} J_{Z_k}^T F_{Z_k}$;
16. if $t_k < 1$ then
17. $L_{k+1} = 2L_k$
18. else
19. $Ared = \|F_{X_k}\|^2 - \|F_{X_k+1}\|^2$
20. $Pred = \|F_{X_k}\|^2 - \|F_{X_k} + J_{X_k} d_k\|^2 - \mu_k \|d_k\|^2 = -\langle d_k, J_{Z_k} F_{X_k} \rangle$
21. If $Ared > \eta \, Pred$ then
22. $L_{k+1} = max\left\{ \dfrac{L_k}{2}, \sigma \right\}$
23. else
24. $L_{k+1} = L_k$
25. end if
26. end if
27. $k \leftarrow k+1$
28. end while

3.1. **Remarks**

(i) The iteration begins with $k = l - 1$, and ends with $k = l$ if $J_k^T F_k \neq 0$.
(ii) If the iteration does not end at iteration $k+1$, then μ_k , d_k are well defined and this direction is a descent one for the square norn $\|F(.)\|^2$. Therefore the Armijo line search in step 7-10 of our algorithm 1 has finite termination. It means that, algorithm 1 is well defined and it either terminates with $J_k^T F_k = 0$ or it generates the infinite sequences $(x_k), (d_k), (t_k), (\mu_k), L_k)$.
(iii) δL plays the role of the Lipschitz- constant of J(x).

(iv) Though the spectral radius approach is a new concept for computing the LM parameter, we observe that $\mu_k > 0$ for both $\rho(Q)$ and $\rho(J_k) > 0$.
It should be noted that we assume that algorithm 3.1, with the above inputs, does not stop at step 2 and it generates $(x_k),(d_k),(t_k),(\mu_k),L_k),$ as (infinite) sequences.

Proposition 3.1. *If $L_k \geq L$, then $t_k = 1$ and $L_{k+1} = max\{ L_k/2, \delta\}$.*

Proof. Suppose that $L_k \geq L$. Then from the definition of μ_k and the stated assumption, the first inequality of the proposition follows from that of the parameter in the definition of d_k, theorem 2.1 with $\mu = \mu_k$, $x = x_k$, $s = d_k$, and steps 7-11. The second inequality comes from the first one and from steps 17-27. ■

Proposition 3.2. *For all k,*

$$\delta \leq L_k \leq max\{L_0, 2L\}, \tag{11}$$

and for infinitely many k, $t_k = 1$.

Proof. Now, since $L_0 \geq \delta$ and $L_{k+1} \geq max\{L_k/2, \delta\}$ for all k, the first inequality in (10) also holds for all k. The second inequality could be proved by induction with varying k. The inequality holds trivially if $k = 0$. Assume that $k \neq 0$ and it holds for some k.

Steps 17-27 of the algorithm 3.1 imply that if $t_k = 1$, then $L_{k+1} = L_k$ or $L_{k+1} = max\{\delta, L_k/2\} \leq L_k$ and in all the cases, the inequality holds for $k+1$. If $t_k < 1$, from proposition 3.1, it implies that $L_k < L$ and so $L_{k+1} = 2L_k \leq 2L$. Therefore, the inequality holds for $k+1$; hence the proof of the first part is complete. For the proof of the second part of the proposition, we suppose that $t_k < 1$ for any $k \geq k_0$.
Then, $L_k + 2^{k-k_0} L_{k_o}, k = k_0, k_0+1, k_0+2, \ldots$ This is a contradiction with (11). Consider then proposition 3.2 and step 5 of the algorithm 3.1, to write

$$\delta \|F(x_{k+1})\| \leq \mu_k \leq \frac{2+\sqrt{5}}{4} max\{L_0, 2L\} \|F(x_k)\|, \tag{12}$$

for all k. ■

Proposition 3.3. *For each k,*

$$\|F(x_{k+1})\|^2 \leq \|F(x_k)\|^2 + \beta t_k \langle J_k^T F_k, d_k \rangle$$

$$\leq \|F(x_k)\|^2 - \beta t_k \frac{\|J_k^T F_k\|^2}{\|J_k\|^2 + \mu_k} . \tag{13}$$

As a cosequence, the sequence$(\|(x_k)\|)$ is strictly decreasing and

$$\sum_{k=0}^{\infty} \beta t_k \frac{\|J_k^T F_k\|^2}{\|J_k\|^2 + \mu_k} \leq \|F(x_0)\|^2.$$

Proof. The first inequality follows from the stopping condition for the armijo line search (steps 7-10). Considering the definition of s_k and μ_k, and the fact that $\mu_k > 0$, leads to

$$-\langle J_k^T F_k, d_k \rangle = \langle J_k^T F_k, (J_k^T J_k + \mu_k I)^{-1} J_k^T F_k \rangle \geq \frac{\|J_k^T F_k\|^2}{\|J_k^T J_k\| + \mu_k},$$

which implies trivially the second inequality. Moreover, the last statement of the proposition follows directly from (12). ■

4. Convergence Analysis

The following results facilitate the convergence analysis of of algorithm 3.1.

Proposition 4.1. *If the sequence (x_k) is bounded, then it has a stationary accumulation point.*

Proof. by proposition 3.2, $t_k = 1$ for infinitely many k. Since (x_k) is bounded, there exists a subsequence (x_{k_j}) convergent to $\tilde{x}$, such that $t_{K_j} = 1$ for all j.
Thus, going by proposition 3.3,

$$\sum_{k=0}^{\infty} \beta \frac{\|J_{kj}^T F_{kj}\|^2}{\|J_{kj}\|^2 + \mu_{kj}} \leq \|F(x_0)\|^2.$$

In addition, Since F and J are continuous, $\|F(x_k)\|$, $\|J_k\|$ and μ_k are bounded.
Hence, the sequence $(J_{k_j}^T F_{k_j})$ converges to 0. Then use continuity of both F and J to conclude that $J(\tilde{x})^T F(\tilde{x}) = 0$. ■

Next we can prove that the step length is bounded away from zero.

Proposition 4.2. *If* $\delta > 0$, *then,*

$$t_k \geq \frac{8\delta^2/L^2}{1+16\delta/L}$$

for all k.

Proof. For any $t \in [0,1]$ the following holds.

$$\frac{L^2}{4} t^3 \|d_k\|^2 + Lt\, \|F(x_k)\| - \mu_k$$

$$\leq t\, (\frac{L^2}{4} \|d_k\|^2 + L\|F(x_k)\|) - \mu_k$$

$$\leq t\, (\frac{L^2}{16\,\mu_k} \|F(x_k\|^2 + L\|F(x_k)|$$

$$\leq t(\frac{L^2}{16\,\delta_k} \|F(x_k\, \|^2 + L\|F(x_k)\|) - \sigma\|F(x_k)\| = \delta\|F(x_k)\|\, [t(\frac{L^2}{16\,\delta^2} + \frac{L}{\delta}) - 1]. \quad (14)$$

The bound of t gives the first inequality, while the second from theorem 2.4 and the third from the fact that $\mu_k \geq \sigma\|F(x_k)\|$, as stated in (11).
By inequality (13) and Lemma 2.3 in [10], we have that if

$$0 \leq t \leq \frac{1}{\frac{L^2}{16\delta^2} + \frac{L}{\delta}} = \frac{8\delta^2/L^2}{1+16\delta/L},$$

then,

$$\|F(x_k + td_k)\|^2 \leq \|F(x_k)\|^2 + t\langle d_k, J_k^T F_k\rangle.$$

Hence, the result follows from steps 7-10 of algorithm 3.1. ■

Proposition 4.3. *If $\delta > 0$, then all accumulation points of the sequence (x_k) are stationary for the function $\|F(x)\|^2$.*

Proof. Suppose that (x_{k_j}) converges to some $\tilde{x}$. From propositions 3.3 and 4.2, we have that

$$\sum_{j=1}^{\infty} \frac{\|J_{kj}^T F_{kj}\|^2}{\|J_{kj}\|^2 + \mu_{kj}} < \infty.$$

It follows from (11) and the continuity of F and $J(x)$ that $\|J_{k_j}\|^2 + \mu_{k_j}$ is bounded. Therefore, $J_{k_j} F_{k_j}$ converges to 0. ■

Proposition 4.4. *If $\delta > 0$ and (x_k) is bounded, then*
$$\min_{i=1,2,3,\ldots,k} \|J(x_i)F(x_i)\| = O(\frac{1}{\sqrt{k}}).$$

Proof. We define
$M = sup_k\{\|J_k\|^2 + \delta L max\{L_0, 2L\}\|F_k\|\}$.
Then, by (11) and propositions 3.3 and 4.2, for any k, we have
$$k\frac{\beta}{M}\left(\frac{8\,\delta^2/L^2}{1+16\,\delta/L}\right) min\, \|J_k^T F_k\|^2 \le \sum_{i=1}^{k} \beta t_i \ \frac{\|J_i^T F_i\|^2}{\|J_i\|^2 + \mu_i} \le \|F(x_0)\|^2,$$
and the result follows. ■

5. Numerical Results

In this section, we report on some numerical results of our proposed method. The performance of the algorithm 3.1 was tested on certain bench-mark problems in comparison to two other LM methods. The algorithms were coded in MATLAB 7.10.0 (R2010a) and run on a personal computer with a 3.0GHZ CPU processor.

The results are listed in Table 1-2 where different initial points were considered. We adopted almost all the parameters used in [10] and the remaining ones are : $L_0 = 20, \epsilon = 10^{-4}, \eta = 1, \beta = 10^{-4}$ and $\delta = 10^{-8}$.
We say that the method found a solution if
$$N^* = \|J_k^T F_k\| \le 10^{-5}. \qquad (15)$$
The meanings of the columns in Tables 1-2 are stated as follows:
n: the dimension of the problem;
Iter: the total number of iterations;
#Fun: number of function evaluations;
cpu: the cpu time in seconds;
N^*: denotes the stopping criterion.
We denote failure of the algorithm by –, and this migh occur due to low memory or $N^* > 10^{-5}$.

The proposed method and the other two metods were tested on some benchmark problems. Problems 2-6 below are derived from [13], problem 1 is a modified form of problem 1 of [13], while problem 7 is taken out of [4].

Problem 1:

$$F(x) = \begin{pmatrix} 3 & -1 & & & \\ -1 & 3 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 3 \end{pmatrix} x + (e_1^x - 1, \ldots, e_n^x - 1)^T.$$

Problem 2:

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (sinx_1 - 1, \ldots, sinx_n - 1)^T.$$

Problem 3:

$$F_1(x) = x_1(x_1^2 + x_2^2) -1,$$
$$F_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2),$$
$$F_n(x) = x_n(x_{n-1}^2 + x_n^2).$$
$$i = 2, 3, \ldots, n-1.$$

Problem 4:

$$F_{3i-2}(x) = x_{3i} - 2x_{3i-1} - x_{3i}^2 - 1,$$
$$F_{3i-1}(x) = x_{3i-2}\, x_{3i-2}\, x_{3i} - x_{3i-2}^2 + x_{3i-1}^2 - 2,$$
$$F_{3i}(x) = e^{-x_{3i-2}} - e^{-x_{3i-1}}.$$
$$i = 1, \ldots, \frac{n}{3}.$$

Problem 5:

$$F_i(x) = (1 - x_i^2) + x_i(1 + x_i\, x_{n-2}\, x_{n-1} x_n) - 2.$$
$$i = 2, 3, \ldots, n.$$

Problem 6:

$$F_1(x) = x_1^2 - 3x_1 + 1 + cos(x_1 - x_2),$$
$$F_i(x) = x_1^2 - 3x_i + 1 + cos(x_i - x_{i-1}),$$
$$i = 1, 2, \ldots, n.$$

Problem [4] 7:

$$F_i(x) = e_i^x - 1,\ i = 1, 2, 3, \ldots, n \text{ and } x_0 = (0.02, 0.02, 0.02, \ldots, 0.02)^T.$$

Problem 8:

$$F_i(x) = x_i^2 - 4,\ i = 1, 2, 3, \ldots, n \text{ and } x_0 = (-1, -1, -1, \ldots, -1)^T.$$

Table 1 : Numerical CLM Results for problems 1 - 7

Problems	*n*	# Iter	# Fun	N^*	cpu	Exist
1	10	8	9	3.51E-17	0.5478	1
	100	18	32	6.14E-11	1.0321	1
	500	20	38	9.61E-07	35.8419	1
	1000	23	54	8.85E-10	204.6444	0
2	10	10	22	3.43E-23	0.7946	1
	100	12	25	1.31E-24	0.8178	1
	500	15	31	1.18E-24	15.9103	1
	1000	22	34	3.30E-24	48.0528	1
3	10	8	11	4.63E-17	0.0209	1
	100	20	38	8.38E-08	1.0979	1
	500	74	151	-	115.145	3
	1000	84	170	-	1154.436	3
4	10	17	26	-	0.2651	3
	100	17	26	-	0.7987	3
	500	18	27	-	22.128	3
	1000	18	27	-	183.4044	3
5	10	5	6	5.92E-31	0.011	1
	100	15	4	9.11E-17	0.1916	1
	500	17	4	4.20E-24	3.2699	1
	1000	19	4	7.71E-27	26.7783	1
6	10	8	11	4.63E-17	0.0209	1
	100	20	38	8.38E-08	1.0979	1
	500	74	151	-	115.145	3
	1000	84	170	-	1154.436	3
7	10	5	6	5.92E-31	0.011	1
	100	15	5	9.11E-17	0.1916	1

Table 2 : Numerical ARCLM Results for problems 1 - 7

Problems	n	# Iter	# Fun	N^*	cpu	Exist
1	10	8	9	3.51E-17	0.2478	1
	100	12	21	6.14E-11	1.0321	1
	500	13	22	9.61E-07	17.8419	1
	1000	14	29	8.85E-10	194.6444	0
2	10	6	11	3.43E-23	0.0946	1
	100	6	7	1.31E-24	0.5178	1
	500	6	7	1.18E-24	5.9103	1
	1000	6	734	3.30E-24	48.0528	1
3	10	8	11	4.63E-17	0.0209	1
	100	20	38	8.38E-08	1.0979	1
	500	74	151	-	115.145	3
	1000	84	170	-	1154.436	3
4	10	17	26	3.10E-09	0.2651	1
	100	17	26	3.41E-011	0.7987	1
	500	18	27	1.72E-07	22.128	1
	1000	18	27	-	183.4044	3
5	10	5	6	5.92E-31	0.011	1
	100	20	38	8.38E-08	1.0979	1
	500	74	151	-	115.145	3
	1000	84	170	-	1154.436	3
6	10	8	11	4.63E-17	0.0209	1
	100	20	38	8.38E-08	1.0979	1
	500	74	151	-	115.145	3
	1000	84	170	-	1154.436	3
7	10	5	6	5.92E-31	0.011	1
	100	3	4	9.11E-17	0.1916	1

Table 3 : Numerical SRLM Results for problems 1 - 7

Problems	*n*	# Iter	# Fun	N^*	cpu	Exist
1	10	6	9	2.50E-16	0.0689	1
	100	8	11	3.74E-21	0.4653	1
	500	8	11	7.59E-09	16.4488	1
	1000	8	11	6.74E-17	142.0582	1
2	10	5	7	1.48E-17	0.0486	1
	100	5	7	9.43E-19	0.3946	1
	500	5	7	3.13E-20	11.9695	1
	1000	5	7	3.09E-20	95.8803	1
3	10	7	10	9.85E-18	0.1744	1
	100	8	11	3.74E-21	0.4653	1
	500	9	12	6.29E-21	18.6814	1
	1000	9	12	1.98E-16	154.9348	1
4	10	27	123	5.00E-13	0.589	1
	100	26	123	5.26E-10	2.2461	1
	500	26	123	2.64E-7	144.0507	1
	1000	27	128	-	255.188	3
5	10	5	8	4.42E-18	0.0216	1
	100	5	8	4.12E-19	0.3548	1
	500	2	3	8.96E-19	6.4269	1
	1000	2	3	3.58E-21	39.9825	1
6	10	7	10	9.85E-18	0.1744	1
	100	8	11	3.74E-21	0.4653	1
	500	9	12	6.29E-21	18.6814	1
	1000	9	12	1.98E-16	154.9348	1
7	10	5	8	4.42E-18	0.0216	1
	100	5	8	4.42E-18	0.0216	1

The results corresponding to the solved problems are depicted in the performance profiles of Figures 1, 2 and 3, for the number of iterations, cpu time and function evaluation.The outcomes of the three strategies, CLM [7], ACRLM [9], ARCLM and our proposed method SRLM are respectively , displayed, row by row for each problem.

It is well known that some variations of the CPU time might occur from one execution of an algorithm to the other. We run seven times and consider the average CPU time of the last six runs, where the first CPU time is discarded. Problems 4 was considered unsolved by the CLM, which also has not solved problems 3 and 6 at higher dimensions. The ARCLM clearly fails to solve problems 3, 4 and 7 at high dimensions. It is also worth noting that our proposed method apparently fails to solve problem 4 at $n = 1000$.

It is moreover clear from Tables 1, 2, 3 and Figures 1, 2 and 3 that our method solves about 69% of the total tested problems with the least number of iterations and function evaluations. Moreover, in contrast to the two other algorithms, it can also be observed that as the dimension increases, our proposed algorithm requires less cpu time to get to the approximated solution. In terms of robustness and efficiency, our method slightly outperformed both the CLM and ARCLM with regard to number of iterations, cpu time and function evaluations.

5.1. **Performance profiles**

Below are figures illustrating the performance of our new SRLM method in comparison to CLM and ARCLM. The comparison was made in terms of number of iterations, cpu-time and function evaluations.

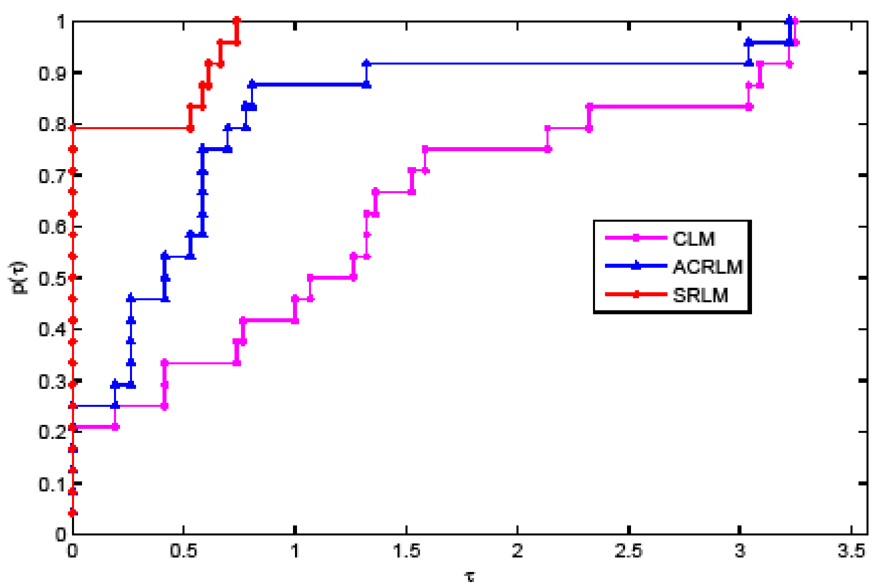

Figure 1: Performance profile for CRLM, ARCLM and SRLM methods with respect to number of iterations for problems 1-7

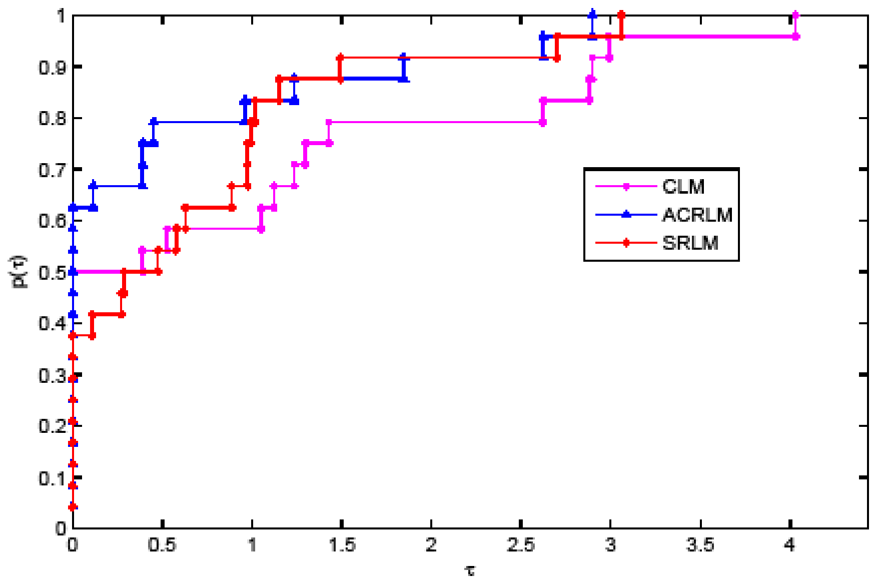

Figure 2: Performance profile for CRLM, ARCLM and SRLM methods with respect to cpu-time for problems 1-7

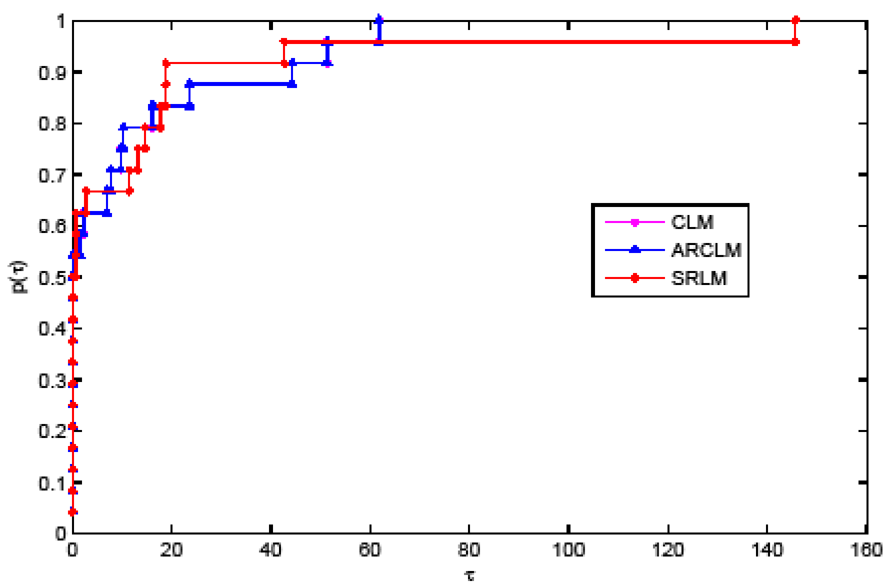

Figure 3: Performance profile for CRLM, ARCLM and SRLM methods with respect to function evaluations for problems 1-7

6. Final Remarks

In this paper, we have proposed a new approach to computing the regularization parameter in the Levenberg-Marquadrt method for solving nonlinear systems of equations. This is namely the spectral radius approach which produces a moderate LM step that makes the iterates move faster towards the solution. In term of convergence, all accumulation points of the sequence generated by the algorithm are indeed stationary. From the numerical experiments conducted, the spectral radius approach has shown that it is both efficient and competitive.

6.1. **Future research challenges**

The greatest challenge ahead, is how to apply the Levenberg-Marquardt algorithm with our regularization parameter evaluation in areas of modern civil engineering, where most of the problems are inverse and nonlinear in nature. The image compression is another area which needs to be addressed by the algorithm since digital images and videos are still demanding in terms of storage space and bandwidth. In recent years, Intelligent Mobile Robots are subjects that have been receiving increasing attention. With all the existing literature on Neutral Networks, finding minima of the error functions still pose a difficult problem.

Acknowledgments
The authors are very grateful to Professor Elizabeth W. Karas (ewkaras@gmail.com) of Federal University of Parana, Curitiba, Brazil, for her great support throughout this work.

References

[1] K. Amini, and F. Rostami, A modified two steps levenberg-marquardt method for nonlinear equations, *Journal of Computational and Applied Mathematics* **288**, (2015),341-350.

[2] N. Bidabadi, A modified BFGS method for solving symmetric nonlinear equations, *Journal of Interdisciplinary Mathematics* **17**(5-6), (2014),461-469.

[3] C.G. Broyden, Quasi-Newton methods and their applications to function minimization, *Mathematics of Computation* **21**, (1967), 368-381.

[4] M. T. Darvishi, and B-C Shin, Higher-order Newton-Krylov methods to solve systems of nonlinear equations, *Journal of the Korean Society for Industrial and Applied Mathematics* **15**(1), (2011),19-30.

[5] I. Fan, The modified Levenberg-Marquardt method for nonlinear equations with cubic convergence, *Mathematics of Computation* **81**, (2012), 447-466.

[6] I. Fan, Accelerating the modified Levenberg-Marquardt method for nonlinear equations, *Mathematics of Computation* **83**, (2014), 1173-1187.

[7] J. Fan, and J. Pan, Convergence properties of a self-adaptive of Levenberg-Marquardt

algorithm under local error bound condition, *Computational Optimization and Applications* **34**(1), (2006), 47-62.

[8] Y. He, C. Ma, and B. Fan. Corrected Levenberg-Marquadrt algorithm with a nonmonotone line search for solving system of nonlinear equations, *Applied Mathematics and Computation* **260**, (2015), 159-169.

[9] J. Fan, and Y. X. Yuan, On the quadratic convergence of the Levenberg-Marquardt method with out nonsingularity assumption for nonlinear equations, *Computing* **74**, (2005), 23-39.

[10] E.W. Karas, S. A. Santos, and B. F. Svaiter, Algebraic rules for computing the regularization parameter of the Levenberg-Marquardt method, *Computational Optimization and Applications* **65**(3), (2016), 723-751.

[11] D. H. Li, and M. Fukshima, A globally and superlinearly convergent Gauss-Newton based BFGS method for symmetric nonlinear equations, *SIAM Journal of Numerical Analysis* **37**(1), (1999), 152-172.

[12] M. Kimiaei, Nonmonotone self-adaptive Levenberg-Marquardt approach for solving systems of nonlinear equations, *Numerical Functional Analysis and Optimization* **39**(1),(2018),47-66.

[13] J. Liua, and X. Wu, New three-term conjugate gradient method for solving unconstrained optimization problems, *ScienceAsia* **40**, (2014), 295–300.

[14] M. Fukushima, and N. Yamashita, On the rate of convergence of the Levenberg-Marquardt method (239-249), in *Topics in Numerical Analysis, Computing Supplementa, Computing* **15**, Springer, Berlin, 2001.

[15 L. Qi, X. Xiao, and L. Zhang, A parameter-self-adjusting Levenberg-Marquardt method for solving nonsmooth equations, *Journal of Computational Mathematics* **34**(3), (2016), 317-338.

[16] M.V. Solodov and B. F. Svaiter, A globally convergent inexact Newton method for systems of monotone equations (355-369), in *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, M. Fukushima and L. Qi (Eds), Kluwer,Boston, 1998.

Article history: Submitted March, 11, 2018; Revised May, 03, 2018; Accepted June, 06, 2018.