View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by AJOL - African Journals Online

Special Conference Edition November, 2018
http://dx.doi.org/10.4314/bajopas.v11i1.80S
Bayero Journal of Pure and Applied Sciences:11(1): 502 - 515
ISSN: 2006 - 6996

Matrix-Norm Approach of Computing Levenberg-Marquardt Regularization Parameter for Nonlinear Equations

[1]Yau Balarabe Musa and [2] M. Y. Waziri

[1]Department of Mathematics and computer Sciences, Faculty of Natural and Applied Science, Sule Lamido University, Kafin Hausa, Jigawa, Nigeria.
balarabemusa.yau@jsu.edu.ng

[2] Department of Mathematical Sciences, Faculty of physical Science ,Bayero University Kano, Kano, Nigeria.

Abstract

In this paper, we present Levenberg-Marquardt method for solving nonlinear systems of equations. Here, both the objective function and the symmetric Jacobian matrix are assumed to be Lipchitz continuous. The regularization parameter is derived using Matrix-Norm approach. Numerical performance on some benchmark problems that demonstrates the effectiveness and efficiency of our approach are reported and have shown that the proposed algorithm is very promising.

Mathematics Subject Classification: 65H10, 65K05, 65F22, 65F35.
keywords: Nonlinear system of equations. Levenberg-Marquardt method. Regularization. Matrix-norm. Global convergence.

1 Introduction.

In this paper, we consider the problem of finding solution to the nonlinear equation

$$F(x) = 0 \tag{1.1}$$

where

$$F : \mathbb{R}^n \longrightarrow \mathbb{R}^n \tag{1.2}$$

is continuously differentiable function.

I.e $F = (f_1, f_2, f_3, ..., f_n)^T$ and $x = (x_1, x_2, x_3, ..., x_n)$. The Jacobian matrix, $J(x) = F'(x)$, $\forall\ x \in \mathbb{R}^n$ and is denoted as J_k which is also assumed to be symmetric and Lipschitz continuous.

The most efficient procedure for solving (1.1) is purely iterative (Amini and Rostami, 2015; Fan et al., 2005; Karas et al., 2016; Yamashita and Fukushima, 2001; Qi et al., 2016). Many algorithms have been used for solving (1.1). For instance, Newton's method, Gauss-Newton's, Trust Region Method and Quasi-Newton's method (Bidabadi, 2014; Broyden, 1967; Li and Fukshima, 1999; Solodov et al.,1998). As (1.1) is nonlinear, it may have no solution. In this paper, we assume that the solution of (1.1) exits. It is well known that the Levenberg-marquardt LM method is one of the most important and efficient methods for solving the nonlinear system of equations (Amini et al., 2015; Brown, 1971; J. Fan, 2012; Fan and Pan, 2006; He and Fan, 2015; Li, 2014). Recently, the LM method turned out to be a valuable principle for obtaining fast convergence to a solution of nonlinear system if the Jacobian matrix is Lipschitz continuous and nonsingular at the solution (Amini and Rostami, 2015).
The LM method is a classical method for solving nonlinear system of equations.The LM direction d_k, is computed at each iteration as

$$d_k = -(J_k^T J_k + \mu_k I)^{-1} J_k^T F_k \tag{1.3}$$

where, μ_k is called Levenberg-Marquadrt regularization parameter and I is an $n \times n$ identity matrix of the Jacobian.
The LM parameter μ_k, is introduced to overcome the difficulty when $J_k^T J_k$ is singular or very close to singularity (Amini et al., 2015; Fan, 2015; Karas et al., 2016; Li, 2014). By choosing a suitable parameter μ_k, the method acts like the gradient descent method whenever the current iteration is far from a solution x^*, and behaves similar to the Gauss-Newton method if the current iteration is close to x^* (AMasoud, 2018). The parameter μ_k is updated in every iteration. The notion of (local) error bound usually plays a key role in establishing the rate of convergence of the sequence of iterations generated by a given algorithm. This condition guarantees that the distance from the current iteration x_k to the solution set denoted by $\text{dist}(x_k, x^*) = \inf_{y \in x^*} ||x_k - y||$, is less than the value of a residual function $\phi : \mathbb{R}^n \to \mathbb{R}+$ at that point $(\phi(xk))$ (Masoud, 2018). For many decades, alot among researchers use various approaches for computing the regularization parameter for Levenberg-Marquadrt. It is vital to mention that Fan and Yuan,2005; Proposed (LM) parameter $\mu_k = ||F_k||^\sigma$ and obtained an algorithm that has quadratic convergence .

(Fan and Pan, 2006), proved that if the parameter is choosen as $\mu_k = ||F_k||^\delta$, for $\delta \in (0,2]$, under local error bound condition, then the convergent order of the LM algorithm is $\min\{1+\delta, 2\}$. (C. Ma and L. Jiang, 2007); came up with parameter as $\mu_k = \theta||F_k|| + (1-\theta)||J_kF_k||$ as a convex combination of the above two parameters. Where $\theta \in [0,1]$. (Fan and Pan,2009); proposed their parameter as $\mu_k = \xi_k\rho(x_k)$, where ξ_k is updated by Trust- region technique, $\rho(x_k) = min\{\tilde{\rho}(x_k), 1\}$ and $\tilde{\rho} : \mathbb{R}^n \to \mathbb{R}_+$ is a positive function with $\tilde{\rho}(x_k) = O(||F_k||^\eta)$, for $\eta \in [0,1]$. J.Fan, 2012; introduced a Modified Levenberg-Marquardt method (MLM) with cubic convergence where LM parameter was choosen as $\mu_k = \lambda_k||F_k||^\delta$ with $\lambda > 0$. (Karas et al 2016); choosed LM parameter as $\min\{\mu_k^+, \mu_k^-\}]$ where

$$\mu_k^- = \frac{L_k}{4}(2||F_k|| + \sqrt{4||F_k||^2 + ||P_k(F_k||^2}), \qquad \mu_k^+ = \frac{2+\sqrt{5}}{4}L_k||F_k||. \qquad (1.4)$$

where P_k is the projection onto the range of the matrix J_k. Musa and Waziri obtained a globally convergent algorithm by using $\mu_k = \delta_k L_k\{\frac{\rho(Q_k)}{\rho(J_k)}\}^2$, where, $Q_k = J_k^T J_k + \mu I_n$, $\mu > 0$, $L_k > 0$, $\delta_k = \frac{1}{k^k}$ for $k \geq 1$, $\rho(Q_k)$ and $\rho(J_k)$ are the spectral radii of the matrix Q_k and J_k respectively.

2 Technical results.

For any $\mu > 0$,. **Theorem 2.1**
$||d|| = ||(A^TA + \mu I)^{-1}A^Tb|| \leq \frac{1}{2\sqrt{\mu}}||P(F(x))|| \leq \frac{1}{2\sqrt{\mu}}||F(x)||$. where P is the orthogonal projection onto the range of A **Proposition 2.2** For any induced matrix norm and a nonsingular matrix A, then,
$||A^{-1}|| \geq ||A||^{-1}$
where
$||A|| = \max_{||x||=1}\{||Ax||\}$ and $||A^{-1}|| = \max_{||x||=1}\{||A^{-1}x||\}$

Here, we consider the LM direction d in Theorem 2.1 and by equating $A = J_k$, we have
$d = -(J_k^TJ_k + \mu I)^{-1}J_k^TF(x_k)$

$$||d|| = || - (J_k^TJ_k + \mu I)^{-1}J_k^TF(x_k)|| \qquad (2.1)$$

$$\leq ||(J_k^TJ_k + \mu I)^{-1}||||J_k^T||||F(x_k)|| \qquad (2.2)$$

But by proposition 2.2, we have

$$\frac{||J_k^T||||F(x_k)||}{||J_k^T J_k + \mu I||} \leq ||(J_k^T J_k + \mu I)^{-1}||||J_k^T||||F(x_k)|| \tag{2.3}$$

But J_k is symmetric, hence Hermitian, thus $J_k^T J_k = J_k^2$.

$$\frac{||J_k^T||||F(x_k)||}{||J_k^2|| + \mu||I||} \leq \frac{||J_k^T||||F(x)||}{||J_k^T J_k + \mu I||} \tag{2.4}$$

Since,

$$\frac{||J_k^T||||F(x_k)||}{||J_k^2|| + \mu||I||} = \frac{||J_k^T||||F(x_k)||}{||J_k^2|| + \mu(\sum_1^n 1^2)^{1/2}} \tag{2.5}$$

$$= \frac{||J_k^T||||F(x_k)||}{||J_k^2|| + \mu\sqrt{n}}$$

By theorem 2.1,

$$\frac{||J_k^T||||F(x)||}{||J_k^2|| + \mu\sqrt{n}} \leq \frac{1}{2\sqrt{\mu}}||F(x)|| \tag{2.6}$$

Also,

$$\frac{||J_k^T||}{||J_k^2|| + \mu\sqrt{n}} \leq \frac{1}{2\mu} \leq \frac{1}{2\sqrt{\mu}}.$$

Hence, either

$$\frac{||J_k^T||}{||J_k^2|| + \mu\sqrt{n}} \leq \frac{1}{2\mu} \leq \frac{1}{2\sqrt{\mu}}$$

Or

$$\frac{1}{2\mu} \leq \frac{||J_k^T||}{||J_k^2|| + \mu\sqrt{n}} \leq \frac{1}{2\sqrt{\mu}}$$

Supposition 1: Let

$$\frac{||J_k^T||}{||J_k^2|| + \mu\sqrt{n}} \leq \frac{1}{2\mu}$$

It implies that

$$\mu \geq \frac{\rho(J_k)^2}{2\rho(J_k) - \sqrt{n}} \geq \frac{\rho(J_k)^2}{2\rho(J_k) + \sqrt{n}/n}. \tag{2.7}$$

$$\mu \leq \frac{||J_k^2||}{2||J_k|| - \frac{\sqrt{n}}{n}}, \tag{2.8}$$

for $||J_k|| > \frac{\sqrt{n}}{2n}$.

where n is the dimension of the square $n \times n$ matrix J_k.
Supposition 2: let

$$\frac{1}{2\mu} \leq \frac{||J_k^T||}{||J_k^2|| + \mu\sqrt{n}}$$

This implies that

$$\mu \geq \frac{||J_k^2||}{2||J_k|| - \frac{\sqrt{n}}{n}}, \tag{2.9}$$

for $||J_k|| > \frac{\sqrt{n}}{2n}$
From the two suppositions, we have

$$\mu \leq \frac{||J_k^2||}{2||J_k|| - \frac{\sqrt{n}}{n}}. \tag{2.10}$$

And

$$\mu \geq \frac{||J_k^2||}{2||J_k|| - \frac{\sqrt{n}}{n}}. \tag{2.11}$$

Since

$$\frac{||J_k^2||}{2||J_k|| + \frac{\sqrt{n}}{n}} \leq \frac{||J_k^2||}{2||J_k|| - \frac{\sqrt{n}}{n}}, \forall n \tag{2.12}$$

Since $||F_k|| \leq ||J_k||$, for all k, we proposed to choose our μ as

$$\mu_k = \frac{\delta_k L_k ||J_k^2||}{2||F_k||} \tag{2.13}$$

$L_k > 0$ and $\delta_k = 1/k^k$ for $k \geq 1$

3 Algorithm (MNLM)

Input: $x_0 \in \mathbb{R}$, $\beta \in (0, 1), \eta \in [0, 1), L_0 > 0$, $\delta > 0$ 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 $||J_k||_F = \sqrt{tr(J_k^2)}$, where,$tr(J_k^2)$, is the trace of the square of the matrix J_k,
4. Set $\mu_k = \frac{\delta_k L_k ||J_k^2||}{2||F_k||}$, $\delta_k = 1/k^k$ for $k \geq 1$ $||F_k|| > 0$
5. Compute $d_k = -(J_k^T J_k + \mu_k I)^{-1} J_k^T F_k$
6. $t \leftarrow 1$

7. while $||F(x_k + td_k)||^2 > ||F_k||^2 + \beta t \langle d_k, J_k^T F_k \rangle$ do 8. $t \leftarrow t/2$
9. end while
10. $t_k = t$
11. $z_k = x_k + (t_k + \frac{1}{2})d_k$
12. Compute $F_{z_k} = F(z_k), J_{z_k} = J(z_k)$
13. set $x_{k+1} = z_k - (J_{z_k}^T J_{z_k} + \mu_k I)^{-1} J_{z_k}^T F_{z_k};$
14. if $t_k < 1$ then
15. $L_{k+1} = 2L_k$
16. else
17. $Ared = ||F_{x_k}||^2 - ||F_{x_k+1}||^2$
18. $Pred = ||F_{x_k}||^2 - ||F_{x_k} + J_{x_k} d_k||^2 - \mu_k ||d_k||^2 = -\langle d_k, J_{z_k} F_{x_k} \rangle$
19. If $Ared > \eta Pred$ then 20. $L_{k+1} = max\{\frac{L_k}{2}, \sigma\}$
21. else
22. $L_{k+1} = L_k$
23. end if
24. end if
25. $k \leftarrow k + 1$
26. end while.

4 Numerical results

In this section, we report some numerical results of our proposed method. The performance of the Algorithm was tested on certain bench-mark problems in comparison to two other LM methods. The Algorithms were coded in MATLAB 7.10.0 (R2014a) 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 (Karas et al., 2016) and the remaining ones are stated as follows: $L_0 = 20, \epsilon = 10^{-4}, \eta = 1, \beta = 10^{-4}$ and $\sigma = 10^{-8}$.
We say that the method found a solution if

$$||J_k^T F_k|| \leq 10^{-5} \tag{4.1}$$

The meanings of the columns in Tables 1-2 are stated as folows:
n: the dimension of the problem;
Iter: The total number of iterations;
#Fun: Number of function evaluations;

cpu: the cpu time in seconds;
Stop*:Denotes the stopping criterion.
moreover, if Exist is 1, it implies that the strategy converges and otherwise diverges.

4.1 Result discussion

The results corresponding to the solve problems are represented in the performance profiles of Figure1, 2 and 3, for the number of iterations, cputime and function evaluation.The outcomes of the three strategies, Corrected LM, denoted as (CLC) by (He and Fan, 2015). Algebraic rule of computing LM Parameter by (Karas, 2016) and our proposed method, i.e Matrix Norn Approch of Computing LM parameter denoted as (MNLM) are displayed for each problem respectively.
It is also very known that some variations of the CPU time may occur from one excution of an algorithm to the other, we run eight times and consider the average CPU time of the last six runs, where the first and last CPU times are discarded. Problems 5 and 6 were not solved by CLM and 4 at higher dimension.
Similarly, problem 4 was not solved by both ARCLM and our proposed method at higher dimension It is moreover clear from Tables 1 and 2 and Figures 1 and 2 that our proposed method solves about 73% of the total tested problems with the fewest number of iterations, cpu time 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 proposed method greatly out performed both CLM and ARCLM with regard to number of iterations, cputime and function evaluations.
Problems 2- 6 below are deduced from (Waziri and Sabiu ,2015), while problem 1 is a modified form of problem 1 of ((Waziri and Sabiu ,2015), and 7 is sourced from (Darvish and Shin, 2011) .

Problem1 : $F_1(x) = x_1(x_1^2 + x_2^2) - 1$,
$F_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2)$,
$F_n(x) = x_n(x_{n-1}^2 + x_n^2)$.
$i = 2, 3, ..., n - 1$.

Problem 2: $F(x) = \begin{pmatrix} 3 & -1 & & & \\ -1 & 3 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 3 \end{pmatrix} x + (e_1^x - 1, ..., e_n^x - 1)^T.$

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

Problem 4: $F(x) = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + (sinx_1 - 1, ..., sinx_n - 1)^T.$

Problem 5: $F_i(x) = (1 - x_i^2) + x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2.$
$i = 1, 2, ..., n.$

Problem 6: $F_1(x) = x_1^2 - 3x_1 + 1 + cos(x_1 - x_2),$
$F_i(x) = x_1^2 - 3x_i + 1 + cos(x_i - x_{i-1}), i = 1, 2, ..., n.$

Problem 7. (Darvish and Shin,2011)
$F_i(x) = e_i^x - 1$, $i = 1, 2, 3, ..., n$
and
$x_0 = (0.02, 0.02, 0.02, ..., 0.02)^T$

Table 1: Numerical Results for CLM, ARCLM and MNLM on problems 1-5, Special Conference Edition November, 2018

		CLM					ARCLM					MNLM				
Problem	n	#iter	#Fun	cpu	Stop*	Exist	#iter	#Fun	cpu	Stop*	Exist	#iter	#Fun	cpu	Stop*	Exist
Problem 1	10	50	52	1.0023	7.59E-12	1	47	48	0.9886	9.59E-10	1	6	7	0.0289	6.13E-05	1
	100	160	164	2.4352	4.86E-11	1	156	157	1.9665	2.86E-09	1	7	8	0.2009	7.55E-09	1
	300	280	282	21.0124	9.21E-11	1	274	275	12.0861	4.21E-09	1	7	8	1.8141	2.51E-08	1
	500	360	362	52.1232	3.13E-11	1	357	358	43.1883	5.13E-08	1	7	8	3.9958	4.42E-08	1
	1000	423	425	412.6543	6.60E-09	1	413	414	311.9831	6.60E-09	1	7	8	24.5345	9.60E-08	1
Problem 2	10	12	14	0.6754	5.81E-17	1	8	9	0.2555	5.81E-17	1	5	6	0.0339	7.68E-09	1
	100	26	28	2.08	2.16E-23	1	23	24	1.08	2.16E-23	1	6	7	0.4262	1.19E-12	1
	300	39	41	3.6492	5.59E-21	1	36	37	2.7492	5.59E-21	1	6	7	1.4349	4.43E-12	1
	500	51	53	10.4552	1.68E-22	1	45	46	8.3552	1.68E-22	1	6	7	3.5053	1.57E-10	1
	1000	69	72	59.0291	1.56E-19	1	60	61	49.6291	1.56E-19	1	6	7	17.7409	1.55E-08	1
Problem 3	10	19	21	0.5153	1.16E-24	1	11	12	0.0153	1.16E-24	1	5	6	0.015	1.64E-09	1
	100	24	24	0.765	6.12E-22	1	19	20	0.265	6.12E-22	1	6	7	0.2027	5.40E-13	1
	300	31	33	2.0969	2.48E-22	1	27	28	1.2969	2.48E-22	1	6	7	0.7811	3.5883-12	1
	500	39	41	4.6834	3.7924-18	1	32	33	3.7834	3.7924-18	1	6	7	2.0596	2.18E-11	1
	1000	52	54	34.7355	2.27E-22	1	42	43	24.6355	2.27E-22	1	6	7	10.694	1.50E-10	1
Problem 4	10	29	31	0.2343	2.05E-21	1	22	23	0.0343	2.05E-21	1	4	5	0.018	4.25E+02	1
	100	72	74	1.0227	4.63E-05	1	62	63	0.8227	4.63E-05	1	5	6	0.2694	1.23E-02	1
	300	107	109	5.3188	7.52E-01	1	94	95	4.3188	7.52E-01	3	5	6	1.2575	2.45E-02	1
	500	116	118	22.6643	4.29E+01	3	97	98	11.6643	4.29E+01	3	5	6	4.0052	1.10E-02	3
	1000	123	125	54.1644	2.5518+03	3	98	99	44.8644	2.5518+03	3	5	6	66.9562	3.92E+00	3
Problem 5	10	31	33	0.3375	8.08E7	3	23	24	0.0375	8.08E-11	1	7	8	0.024	4.66E-08	1
	100	52	54	1.5374	1.55E+10	3	40	41	0.6374	1.55E-10	1	8	9	0.0455	3.02E-09	1
	300	61	63	3.9368	1.53E+10	3	58	59	2.7368	1.53E-10	1	8	9	1.1176	8.42E-09	1
	500	79	81	10.2907	2.01E+10	3	70	71	8.2907	2.01E-10	1	8	9	3.225	1.21E-08	1
	1000	103	105	69.782	2.15E+10	3	93	94	56.782	2.15E-10	1	8	9	17.4412	2.11E-08	1
Problem 6	10	18	20	0.4736	2.51E+3	3	8	10	0.0736	2.51E-19	1	8	9	0.0406	1.46E-07	1
	100	21	23	0.9799	1.22E+5	3	12	14	0.3799	1.22E-19	1	8	9	0.408	2.63E-09	1
	300	28	30	3.4147	3.66E+6	3	19	20	1.3147	3.66E-17	1	10	11	0.4617	4.32E-07	1
	500	31	33	23.2535	5.02E+07	3	20	21	13.2535	5.02E-19	1	13	14	7.9561	7.33E-08	1
	1000	36	38	36.706	1.99E+08	3	22	23	15.706	1.99E-17	1	15	16	28.0288	1.42E-05	1
Problem 7	10	29	31	0.5163	3.73E-16	1	21	22	0.0163	3.78E-16	1	6	7	0.0284	2.06E-09	1
	100	32	34	0.9333	1.03E-17	1	26	26	0.5333	9.30E-18	1	7	8	0.4704	3.85E-11	1
	300	36	38	3.9066	1.47E-16	1	30	31	2.0066	2.47E-16	1	7	8	3.1142	3.93E-11	1
	500	39	41	6.8976	3.50E-19	1	33	34	4.4976	3.52E-19	1	7	8	12.213	3.96E-11	1
	1000	44	46	44.8615	7.33E-21	1	38	39	24.5615	5.33E-21	1	7	8	79.6555	3.93E-11	1

4.2 Performance Profile

Below are the figures indicating the performances of our new method (MNLM) in comparison to (CLM and ARCLM). The comparison was conducted in terms of number of iterations, CPU- time and function evaluation .

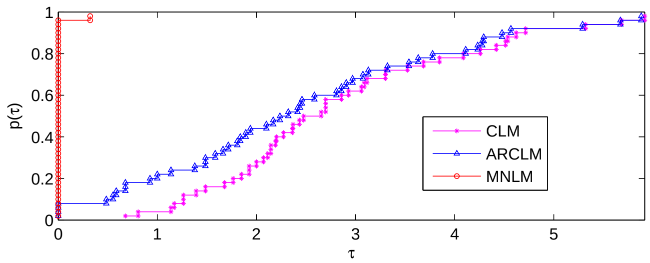

Figure 1: Performance profile of CLM , ARCLM and MNLM. methods with respect to number of iterations for problem 1-7

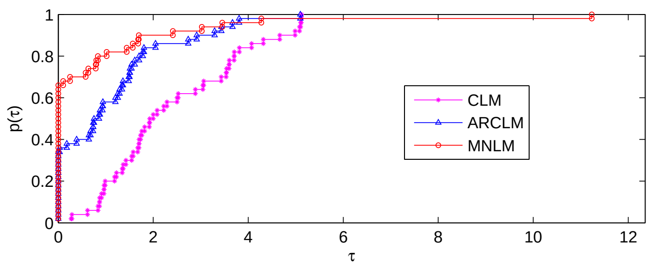

Figure 2: Performance profile of CLM , ARCLM and MNLM. methods with respect to cputime for problem 1-7

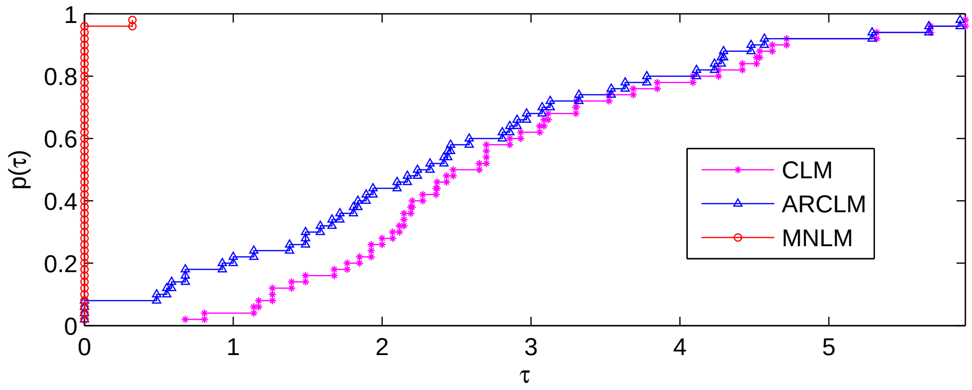

Figure 3: Performance profile of CLM , ARCLM and MNLM. methods with respect to function evaluation for problem 1-7

5 Final remarks.

We proposed a new procedure of computing Levenberg-Marquadrt regularization parameter for method of nonlinear system of equations. The matrix-norm approach has been used for derivation of the parameter and in turns produces a moderate LM step that makes the iterate move faster to the solution. From the numerical experiment conducted, the approach has shown that it is both efficient and promising.

References

Amini, K. andRostami, F. (2015). A modified two steps Levenberg-Marquardt method for nonlinear equations. Computational and Applied Mathematics, 40:0377-0427.DOI: 10.1016/j.cam.2015.04.040.

Bidabadi, N. (2014.). A modidied BFGS algorithm for solving symmetric nonlinear equations. Taylor and Francis Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered : Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK, 7:675-687.

Kenneth, M. B. and Dennis, J.E. Jr.(1971). Derivative free analogues of the Levenberg-Marquadrt and gauss algorithm for non-linear least- squares approximation. Numer. Math. t8, 289-297 (t972), Springer-Verlag 1972, 18:289-297.

Broyden, C.G.(1967). Quasi-newton methods and their applications to function minimisation. Journal for Mathematical Computation, 21:577-593.

Santos, S.A, Karas, E.W. and Benar Svaiter, F (2016). Algebraic rule for computing the regularization parameter of the Levenberg-Marquardt method for nonlinear equations. Computional and Optimizational Application, 65:723-751:DOI 10.1007/s10589069845.

Fan, F (2012). The modified Levenberg-Marquardt method for nonlinear equations with cubic convergence. Mathematics of Computation, 81:447-466.

Fan, J. and Pan, J. (2006). Convergence properties of a self-adaptive of Levenberg-Marquardt algorithm undwer local error bound condition. Computation Optimization Appl, 34:47-62.

Fan,J and Pan, J (2009):. A note on Levenberg-Marquardt parameter. Applied Mathematics and Computation 207(2):351-359, 207(2):351-359

Fischer (2002). A local behavior of an iterative frame work for generalized equations with nonisolated solutions. Math. Program., Ser. A, 94B(1):91-124.

Yeden, H. and Fan, B. (2015). Corrected Levenberg- Marquadrt method with nonmonotone line search for solving system of nonlinear equations. Aplied Mathematics and Comutation, 360:159-169. doi¿10.1016/j.amc.2015.03.076.

Yuan, Y.X. and Fan, J. (2005). On the quadratic convergence of the Levenberg-Marquardt method with out nonsingularity assumption for nonlinear equations. Computational and ap- plied mathematics, 74:23-39.

Li, M.and Fukshima, N. (1999.). A globally and superlinearly convergent gaus-newton based bfgs symmetric methods for solving nonlinear systems of equations. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 37:152-172. DOI: 10.1137/S0036142998335704.

Li, M. (2014). A modied two steps levenberg-marquardt method for nonlinear equations. Numerical Functional Analysis and Optimization, 35:310-322.DOI: 10.1016/j.cam.2015.04.040.

Musa, Y. B., Waziri, M. Y. and Halilu, A. S. (2017). On computing the regularization parameter for the levenberg-marquardt via spectral radius approach for solving system of nonlinear system of equation. Journal of Numerical Mathematics and Stochastics, 9(1):80-94.

Ma, C. and Jiang, L. (2007). Some research on levenberg-marquadrt method for nonlinear equations. Applied Mathematics and Computation, 184:1032-1014. DOI: 10.1016/j.amc.2006.07.004. Ronan M. T., Fleming M. A., Francisco J. A. A . and Phan, T. V. (2018). Local convergence of the levenberg-marquardt method under holder metric subregularity. 1(2):369-383.

Fukushima, M. ,and Yamashita, N. (2001). On the rate of convergence of the levenberg-marquardt method for nonlinear equations. Comput.l suppl., 15:239-249.

Liyan, Q., Xiantao X, and Liwei, Z.(2016). A parameter-self-adjusting levenbergmarquardt method for solving nonsmooth equations. Journal of Computational Mathematics, 34:317338. DOI: 10.4208/jcm.1512m2015-0333.

Solodov, M.V. and Svaiter, B.F. (1998). A globally convergent inexact newton method for systems of monotone equations. 34:355-369.
Waziri, M.Y. and Sabiu,J. (2015). Derivative -free conjugate gradient method and its global convergence for systems of nonlinear equations. International Journal of Mathematics and Mathematical Sciences, 8:64-72. doi.org/10.1155/2015/961487.