



A Cholesky-Factor-Based Levenberg-Marquardt Method for Solving Nonlinear Systems of Equations

^{1,*}Ya'u Balarabe Musa, ²Muhammad Yusuf Waziri and ³Hassan Muhammad

¹Department of Mathematics, Sule Lamido University Kafin hausu, Jigawa State.

²Department of Mathematical Sciences, Bayero University Kano, Kano state.

³Department of Applied Mathematics, Federal University of Technology Babura, Jigawa State.

*Corresponding author's email: yaubmusa.mth@slu.edu.ng

ABSTRACT

Received: 6 Oct. 2026

Accepted: 18 Oct. 2026

Published: 20 Oct. 2026

We present a Levenberg-Marquardt (LM) algorithm for solving nonlinear system of equations. Nonlinear systems of equations are central to numerous scientific and engineering applications, yet solving them efficiently remains computationally challenging. To address limitations in existing Levenberg-Marquardt (LM) parameter selection strategies, this paper presents an efficient LM algorithm (ACLMM) designed to reduce computational cost. In the proposed scheme, the regularization parameter was derived analytically using established result in combination with the maximum diagonal entry of the Cholesky factor matrix. The proposed method was evaluated on seven standard benchmark problems and compared against two existing LM methods the Corrected Levenberg-Marquardt (CLM) and Algebraic Rule for Levenberg-Marquardt Parameter (ARCLM) methods using iteration counts and CPU time, across varying problem dimensions. Numerical results demonstrated that the ACLMM algorithm generally exhibited competitive or improved performance across the tested problems. Notably, for large-scale instances such as Problem 1 ($n = 1000$), ACLMM reduced the iteration count to 8 compared to 23 for CLM and 14 for ARCLM while maintaining small residual norms. Overall, the incorporation of Cholesky factor diagonal entries into the regularization parameter provides an effective and numerically stable alternative for solving nonlinear systems.

Keywords: Levenberg-Marquardt method; Nonlinear systems; Cholesky factorization; Numerical analysis.

INTRODUCTION

Nonlinear system of equations play a crucial role in a wide range of scientific and engineering applications. Solving these systems accurately and efficiently remains a significant computational challenge.

We consider solving nonlinear system of equations.

$$F(x) = 0, \quad (1)$$

Where $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$, is continuously differentiable function and $x = (x_1, x_2, x_3, \dots, x_n)^T$. We denote the solution set of equation (1) by X^* and assume that X^* is non empty. The famous method known for solving (1) is Newton Method, which uses the search direction $d_k^N = -J_k^{-1}F_k$ in every iteration, where $F_k = F(x_k)$, and $J_k = F'(x_k)$. The Newton method has certain disadvantages among which include, the Jacobian matrix J_k can be singular. This is the reason why the LM comes in, to overcome this shortcomings.

The Levenberg-Marquardt (LM) algorithm is among the widely used iterative techniques to solve nonlinear system of equations and nonlinear least-squares problems. The algorithm introduces a regularization parameter to stabilize convergence, particularly when the Jacobian matrix is ill-conditioned or nearly singular (Amini *et al.*, 2015; Fan, 2015; Karas *et al.*, 2016). The Levenberg-Marquardt method is a classical method for solving nonlinear system of equations. The trial step d_k is computed at each iteration as

$$d_k = -(J_k^T F_k + \mu_k I)^{-1} J_k^T F_k. \quad (2)$$

Numerous modifications of the Levenberg-Marquardt (LM) algorithm have been developed, primarily focusing on alternative strategies for selecting the regularization parameter. Fan and Pan

(2006) introduced a self-adaptive LM method where the parameter is defined as

$$\mu_k = \|F_k\|^\delta, \quad \delta \in (0, 1). \quad (3)$$

They established superlinear convergence under this framework and demonstrated strong numerical performance. Fan (2012) later extended this approach by setting

$$\mu_k = \lambda_k \|F_k\|^\sigma, \quad \lambda_k > 0, \sigma \in (0, 2], \quad (4)$$

which improved computational efficiency by significantly reducing the frequency of Jacobian evaluations.

Building upon Fan's formulation, Amini and Rostami (2015) proposed a hybrid step strategy incorporating both a standard and an approximate LM step per iteration, with the parameter defined as

$$\mu_k = \lambda_k \|F_k\|^\delta, \quad (5)$$

where $\delta = 1/\|F_k\|$ for $\|F_k\| \geq 1$, and $\delta = 1 + 1/k$ otherwise. Under a local error bound condition, their trust-region framework achieved global and cubic convergence.

For symmetric systems of nonlinear equations, Musa *et al.* (2017) derived a spectral-radius-based parameter given by

$$\mu_k = \frac{\delta_k L_k}{4} \left[\frac{\rho(Q)}{\rho(J_k)} \right]^2, \quad L_k > 0, \delta_k = k^{-k}, \quad (6)$$

proving its convergence properties.

Alternatively, Karas *et al.* (2016) proposed an explicit algebraic parameter selection scheme using

$$\mu_k = \min\{\mu_k^+, \mu_k^-\}, \quad (7)$$

where

$$\mu_k^- = \frac{L_k}{4} \left(2\|F_k\| + \sqrt{4\|F_k\|^2 + \|P_k(F_k)\|^2} \right) \quad \text{and} \quad \mu_k^+ = \frac{2 + \sqrt{5}}{4} L_k \|F_k\|, \quad (8)$$

with P_k representing the projection onto the range of J_k . They verified that all accumulation points generated by their sequence are stationary.

More recently, Musa and Waziri (2018) introduced a matrix-norm approach to compute the regularization parameter as

$$\mu_k = \frac{\delta_k L_k \|J_k^2\|}{2\|F_k\|}, \quad L_k > 0, \delta_k = k^{-k}. \quad (9)$$

Their formulation yields a moderate LM step size that accelerates convergence toward the solution, demonstrating strong numerical efficiency and global convergence. In a related study, Zeng and Zhou (2022) improved the convergence results for an efficient Levenberg-Marquardt (LM) method incorporating the regularization parameter

$$\lambda_k = \frac{\mu_k \|F_k\|}{1 + \|F_k\|}. \quad (10)$$

Global and superlinear convergence were established under Hölderian continuity and Hölderian local error bound conditions, which weaken the traditional assumptions of Lipschitz continuity and standard local error bounds, respectively. Numerical experiments confirmed the robustness of the algorithm for singular systems satisfying these conditions.

To enhance computational performance, Han and Rui (2024) introduced an adaptive LM parameter defined by

$$\lambda_k = \begin{cases} \mu_k \left(\theta \frac{\|F_k\|^\delta}{1 + \|F_k\|^\delta} + (1 - \theta) \|F_k\|^\delta \right), & \text{if } \|F_k\| \leq 1, \\ \mu_k \left(\theta \frac{\|F_k\|^\delta}{1 + \|F_k\|^\delta} + (1 - \theta) \|F_k\|^{-\delta} \right), & \text{otherwise,} \end{cases} \quad (11)$$

where $\theta \in [0, 1]$, $\delta \in [1, 2]$, and μ_k is updated dynamically using a trust-region strategy. When the iterate sequence $\{x_k\}$ approaches a solution, $\|F_k\| \rightarrow 0$, rendering $\lambda_k = \mu_k \|F_k\|^\delta$ for $\delta = 1$. Conversely, far from the solution set, the dominant term ensures $\lambda_k = \mu_k \theta$. This mechanism regulates parameter magnitude to prevent arbitrarily small trial steps, thereby accelerating convergence.

Concurrently, Zhu et al. (2024) designed an accelerated adaptive two-step LM algorithm. Their approach constructed the LM parameter using a convex combination of normalized residual and gradient norms:

$$\lambda_k = \alpha \frac{\|F_k\|}{1 + \|F_k\|} + (1 - \alpha) \frac{\|J_k^\top F_k\|}{1 + \|J_k^\top F_k\|}, \quad \alpha \in (0, 1). \quad (12)$$

Additionally, they introduced a dynamic update criterion for the upper bound of the approximate step size and proved convergence under Hölderian conditions.

For applications including supply chain optimization, Li et al. (2023) proposed an adaptive parameter scheme given by

$$\lambda_{k+1} = \frac{\mu_{k+1} \|F_{k+1}\|}{1 + \|F_{k+1}\|}, \quad (13)$$

coupled with a piecewise acceptance rule based on the ratio of actual to predicted reduction. This mechanism mitigates stagnation in local minima during unsuccessful iterations. Under local error bound conditions, global and quadratic convergence rates were established for both symmetric and asymmetric systems.

Furthermore, Wang and Fan (2020) investigated the local convergence properties of the LM method under Hölderian conditions. By assuming Hölderian continuity of the Jacobian and Hölderian local error bounds, they derived explicit local convergence rates, demonstrating that superlinear or quadratic convergence is preserved under these relaxed conditions for specific choices of the regularization parameter.

Despite these advancements, balancing low computational overhead with rapid local convergence remains a central challenge in optimization, especially for large-scale ill-conditioned problems.

Existing parameters often require expensive norm evaluations or fail to dynamically adjust trial steps near solutions.

The aim of this study is to develop and numerically evaluate a Cholesky-factor-based Levenberg–Marquardt iterative method (ACLMM) with a new regularization parameter for efficiently solving nonlinear systems of equations. By exploiting the main diagonal entries of the Cholesky factor matrix, the proposed approach provides an effective regularization mechanism with minimal additional computational overhead.

The novel contributions of this paper are summarized as follows:

- We derive a new adaptive regularization parameter based on established results and the maximum diagonal entry of the Cholesky factor matrix.
- We establish an efficient algorithmic framework (ACLMM) that incorporates adaptive step updates to enhance stability.
- We perform extensive numerical experiments on standard benchmark problems up to $n = 1000$ dimensions, demonstrating competitive iteration counts and reduced CPU time compared with established methods.

PRELIMINARIES

1. Positive definite Matrix : A matrix M is Positive definite if and only if for any given non zero vector x , the quadratic form $x^T M x > 0$
2. Cholesky Factor : Is a lower triangular matrix with positive diagonal entries

SOME ESTABLISHED RESULTS

Theorem 1. If a matrix A is positive definite, then A is nonsingular (invertable)

Theorem 2. Let M be any $n \times n$ nonsingular Matrix, and let $A = M^T M$. Then, A is positive definite

Theorem 3. Let L be an $n \times n$ square matrix, and let $\|\cdot\|$ be a norm on $\mathbb{R}^{n \times n}$. Then, $\rho(L) \leq \|L\|$.

Theorem 4. (Cholesky Decomposition Theorem) Let A be a positive definite matrix. Then, A can be decompose in exactly one way into a product $A = M^T M$ (CD) such that M is lower triangular and has all main diagonal entries a_{ii} positive. M is called the Cholesky factor of A .

Theorem 5. The eigenvalue of any triangular matrix (upper triangular, lower triangular) are exactly the entries on its main diagonal

Theorem 6. $\|d_k\| = \|(A^T A + \mu I)^{-1} A^T x\| \leq \frac{1}{2\sqrt{\mu}} \|P(F(x))\| \leq \frac{1}{2\sqrt{\mu}} \|F(x)\|$, where P is the orthogonal projection onto the range of matrix A .

Technical Results

In this section, we present the theoretical foundation for the proposed regularization parameter μ_k . Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a continuously differentiable function, $F_k = F(x_k)$, and $J_k = F'(x_k)$.

At iteration k , consider the symmetric matrix $W_k = J_k^T J_k + \mu_{k-1} I$, where $\mu_0 > 0$ is a given initial parameter. Since W_k is symmetric and positive definite, by Theorem 4, there exists a unique Cholesky factorization such that

$$W_k = L_k L_k^T, \quad (14)$$

where $L_k \in \mathbb{R}^{n \times n}$ is a lower triangular matrix with strictly positive diagonal entries $l_{ii} > 0$

Spectral Bound on the Cholesky Factor Matrix

By Theorem 5, the eigenvalues of a triangular matrix are precisely its main diagonal entries l_{ii} ($i = 1, 2, \dots, n$). Defining the maximum diagonal entry as:

$$\alpha_{\max} = \max_{1 \leq i \leq n} (l_{ii}), \quad (15)$$

it follows that the spectral radius of L_k is $\rho(L_k) = \alpha_{\max}$.

Furthermore, by Theorem 3, for any induced matrix norm $\|\cdot\|$, the spectral radius is bounded above by the norm.

$$\rho(L_k) \leq \|L_k\| \implies \alpha_{\max} \leq \|L_k\|. \quad (16)$$

Dividing both sides by $\|L_k\| > 0$, we obtain the normalized structural ratio:

$$\frac{\alpha_{\max}}{\|L_k\|} \leq 1. \quad (17)$$

Step-Size Bound and Parameter Derivation

From Theorem 6, the trial step d_k generated by the Levenberg-Marquardt step satisfies the norm inequality

$$\|d_k\| \leq \frac{\|F_k\|}{2\sqrt{\mu_k}}. \quad (18)$$

Rearranging this inequality to isolate the parameter μ_k yields an upper bound:

$$\sqrt{\mu_k} \leq \frac{\|F_k\|}{2\|d_k\|} \implies \mu_k \leq \left(\frac{\|F_k\|}{2\|d_k\|} \right)^2. \quad (19)$$

To avoid calculating or estimating the unknown step norm $\|d_k\|$ prior to solving the linear system and to incorporate the local structural information of W_k , we scale the upper bound using the normalized ratio $\frac{\alpha_{\max}}{\|L_k\|} \leq 1$.

Additionally, to ensure rapid local convergence near the solution set X^* while maintaining numerical stability far from X^* , we incorporate the dynamic forcing sequence $\delta_k = k^{-k}$ for $k = (1, 2, 3, \dots)$.

Substituting these components, our proposed adaptive Levenberg-Marquardt regularization parameter is defined as:

$$\mu_k = \delta_k \left(\frac{\alpha_{\max}}{2\|L_k\|} \right)^2, \quad \text{where } \delta_k = \frac{1}{k^k}, \quad k = (1, 2, 3, \dots). \quad (20)$$

Proposed Algorithm (ACLMM)

In this section, we present the formal algorithmic framework for the proposed work, named *A Cholesky-Factor-Based Levenberg-Marquardt Method* (ACLMM), for solving nonlinear systems of equations.

Numerical Experiments

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. That is, Corrected Levenberg-Marquardt (CLM) and Algebraic Rule of Computing Levenberg-Marquardt Parameter (ARCLM) respectively. The comparison is based on number of iterations and CPU time.

The Algorithms were coded in MATLAB 7.10.0 (R2020a) and run on a personal computer with a 3.0GHZ CPU processor. The results are listed in Table 1 where different initial points were considered. We say that the method found a solution if

$N^* = \|J_k^T F_k\| \leq \epsilon$ The meaning of the columns in table 1 are stated as follows:

n : the dimension of the problem;

$\#Iter$: the total number of iterations;

$\#Fun$: the number of function evaluations. cpu : the cpu time in seconds,

N^* : stopping criterion.

Moreover, if Exist is 1, it implies that the strategy converges and otherwise diverges.

We denote failure of the algorithm by -, and it might occur due to low memory or $N^* > 10^{-5}$

Performance profile of CLM, ARCLM and ACLMM methods

Below is the figure indicating the performances of our new method (ANLMM) in comparison to (CLM and ARCLM). The comparison was conducted in terms of number of iterations.

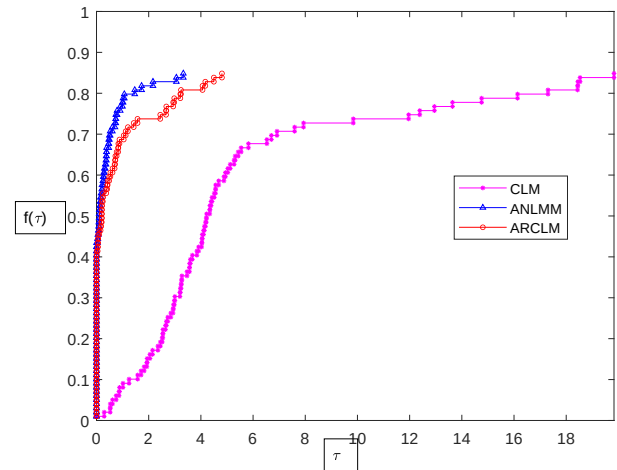


Figure 1: Performance profiles of CLM, ARCLM, and ACLMM with respect to the number of iterations for Problems 1–7.

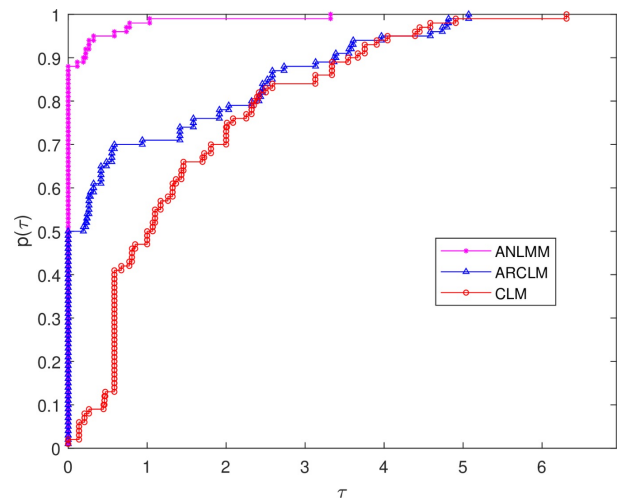


Figure 2: Performance profiles of CLM, ARCLM, and ACLMM with respect to CPU time for Problems 1–7.

RESULTS DISCUSSION

The numerical results in Table 1 demonstrate that the performance of the proposed algorithm (ACLMM) relative to CLM and ARCLM is problem-dependent.

On test problems with large dimensions ($n = 500, 1000$), ACLMM exhibits substantial advantages in iteration counts and convergence speed for Problems 1, 2, 3, and 6. For example, in Problem 1 ($n = 1000$), ACLMM converges in only 8 iterations compared with 23 for CLM and 14 for ARCLM while achieving a smaller residual norm. Similarly, for Problem 2 across all tested dimensions ($n = 10$ to $n = 1000$), ACLMM consistently converges in 5 iterations, whereas CLM requires up to 22 iterations. For

Table 1: Numerical Results for CLM, ARCLM and ACLMM on Problems 1–7

Problem	n	CLM					ARCLM					ACLMM				
		#iter	#Fun	N^*	cpu	Exist	#iter	#Fun	N^*	cpu	Exist	#iter	#Fun	N^*	cpu	Exist
Problem 1	10	8	9	3.51E-17	0.5478	1	8	9	3.51E-17	0.2478	1	6	9	2.50E-16	0.0689	1
	100	18	32	6.14E-11	1.0321	1	12	21	6.14E-11	1.0321	1	7	10	6.92E-11	0.7871	1
	500	20	38	9.61E-07	35.8419	1	13	22	9.61E-07	17.8419	1	8	11	7.59E-09	16.4488	1
	1000	23	54	8.85E-10	204.6444	0	14	29	8.85E-10	194.6444	0	8	11	6.74E-17	142.0582	1
Problem 2	10	10	22	3.43E-23	0.7946	1	6	11	3.43E-23	0.0946	1	5	7	1.48E-17	0.0486	1
	100	12	25	1.31E-24	0.8178	1	6	7	1.31E-24	0.5178	1	5	7	9.43E-19	0.3946	1
	500	15	31	1.18E-24	15.9103	1	6	7	1.18E-24	5.9103	1	5	7	3.13E-20	11.9695	1
	1000	22	34	3.30E-24	48.0528	1	6	7	3.30E-24	48.0528	1	5	7	3.09E-20	95.8803	1
Problem 3	10	8	11	4.63E-17	0.0209	1	8	11	4.63E-17	0.0209	1	7	10	9.85E-18	0.1744	1
	100	20	38	8.38E-08	1.0979	1	20	38	8.38E-08	1.0979	1	8	11	3.74E-21	0.4653	1
	500	74	151	-	115.145	3	74	151	-	115.145	3	9	12	6.29E-21	18.6814	1
	1000	84	170	-	1154.436	3	84	170	-	1154.436	3	9	12	1.98E-16	154.9348	1
Problem 4	10	17	26	-	0.2651	3	17	26	3.10E-09	0.2651	1	27	123	5.00E-13	0.589	1
	100	17	26	-	0.7987	3	17	26	3.41E-011	0.7987	1	26	123	5.26E-10	2.2461	1
	500	18	27	-	22.128	3	18	27	1.72E-07	22.128	1	26	123	2.64E-7	144.0507	1
	1000	18	27	-	183.4044	3	18	27	-	183.4044	3	27	128	-	255.188	3
Problem 5	10	5	6	5.92E-31	0.011	1	5	6	5.92E-31	0.011	1	5	8	4.42E-18	0.0216	1
	100	15	4	9.11E-17	0.1916	1	3	4	9.11E-17	0.1916	1	5	8	4.12E-19	0.3548	1
	500	17	4	4.20E-24	3.2699	1	3	4	4.20E-24	3.2699	1	2	3	8.96E-19	6.4269	1
	1000	19	4	7.71E-27	26.7783	1	3	4	7.71E-27	26.7783	1	2	3	3.58E-21	39.9825	1
Problem 6	10	8	11	4.63E-17	0.0209	1	8	11	4.63E-17	0.0209	1	7	10	9.85E-18	0.1744	1
	100	20	38	8.38E-08	1.0979	1	20	38	8.38E-08	1.0979	1	8	11	3.74E-21	0.4653	1
	500	74	151	-	115.145	3	74	151	-	115.145	3	9	12	6.29E-21	18.6814	1
	1000	84	170	-	1154.436	3	84	170	-	1154.436	3	9	12	1.98E-16	154.9348	1
Problem 7	10	5	6	5.92E-31	0.011	1	5	6	5.92E-31	0.011	1	5	8	4.42E-18	0.0216	1
	100	15	5	9.11E-17	0.1916	1	3	4	9.11E-17	0.1916	1	5	8	4.12E-19	0.3548	1
	500	17	4	4.20E-24	3.2699	1	3	4	4.20E-24	3.2699	1	2	3	8.96E-19	6.4269	1
	1000	19	4	7.71E-27	26.7783	1	3	4	7.71E-27	26.7783	1	2	3	3.58E-21	39.9825	1

Problems 3 and 6 at $n = 1000$, CLM and ARCLM fail to converge within the iteration limit (Exit status 3), whereas ACLMM successfully converges in 9 iterations with a residual norm of 1.98×10^{-16} .

However, ACLMM is not uniformly superior across all benchmark systems:

- **Problem 4:** ACLMM requires more iterations (up to 27 iterations) and significantly more function evaluations (123–128) compared to CLM and ARCLM (17–18 iterations and 26–27 function evaluations).
- **Problem 5:** Although ACLMM achieves comparable iteration counts, its recorded CPU time for higher dimensions ($n = 500$ and $n = 1000$) is higher than that of CLM and ARCLM. This additional CPU overhead is attributed to the evaluation and diagonal extraction of the Cholesky factor L_k at each iteration step.

Aggregate Summary Statistics

Across the 28 evaluated problem instances (7 benchmark problems $\times$ 4 dimensions):

- **Overall Success Rate:** ACLMM solved 27 out of 28 instances (96.4% success rate), compared to 22/28 (78.6%) for CLM and 22/28 (78.6%) for ARCLM.
- **Average Iteration Count:** Across successfully solved instances, ACLMM averaged 8.8 iterations per problem, whereas CLM and ARCLM averaged 20.3 and 16.5 iterations, respectively.

Figures 1 and 2 present the Dolan–Moré performance profiles for iteration counts and CPU time, respectively. Figure 1 confirms that ACLMM has the highest probability $P_s(1)$ of being the top performer in terms of iteration count. Figure 2 illustrates that while ACLMM achieves higher overall reliability across all test cases, its CPU time performance advantage varies depending on matrix dimension and Cholesky factorization cost.

CONCLUSION

In this paper, we introduced a novel Levenberg–Marquardt algorithm (ACLMM) featuring an adaptive regularization param-

eter derived from the maximum diagonal entry of the Cholesky factor matrix. The proposed method was evaluated on seven standard benchmark nonlinear systems across varying dimensions ($n \in \{10, 100, 500, 1000\}$) and compared against the Classical/Corrected Levenberg–Marquardt (CLM) and Algebraic Rule Levenberg–Marquardt (ARCLM) methods.

The empirical findings indicate that the proposed parameter selection strategy significantly improves convergence speed and iteration counts for several large-scale and ill-conditioned systems, such as Problems 1, 2, 3, and 6. However, the comparative performance remains problem-dependent; for instance, ACLMM required additional function evaluations on Problem 4 and incurred higher per-iteration CPU overhead on Problem 5 due to the computation and extraction of the Cholesky factor diagonal entries.

This study has some limitations.

1. The empirical evaluation was restricted to seven benchmark systems.
2. The relative computational efficiency and iteration savings vary according to specific system structures and dimensions.

To build upon these results, future research should focus on extending and testing the ACLMM algorithm on large-scale sparse nonlinear systems, highly ill-conditioned or singular equations, and real-world scientific and engineering applications.

REFERENCES

- Amini, K., & Rostami, F. (2015). A modified two steps Levenberg–Marquardt method for nonlinear equations. *Journal of Computational and Applied Mathematics*, 288, 341–350. <https://doi.org/10.1016/j.cam.2015.04.040>
- Chen, L., & Ma, Y. (2023). A new modified Levenberg–Marquardt method for systems of nonlinear equations. *Journal of Mathematics*, 2023, Article 6043780. <https://doi.org/10.1155/2023/6043780>
- Fan, J. (2012). The modified Levenberg–Marquardt method for nonlinear equations with cubic convergence. *Mathematics of Computation*, 81(277), 447–466. <https://doi.org/10.1090/S0025-5718-2011-02496-8>

Fan, J., & Pan, J. (2006). Convergence properties of a self-adaptive Levenberg–Marquardt algorithm under local error bound condition. *Computational Optimization and Applications*, 34, 47–62. <https://doi.org/10.1007/s10589-005-3074-z>

Han, Y., & Rui, S. (2024). A new adaptive Levenberg–Marquardt method for nonlinear equations and its convergence rate under the Hölderian local error bound condition. *Symmetry*, 16(6), 674. <https://doi.org/10.3390/sym16060674>

Karas, E. W., Santos, S. A., & Svaiter, B. F. (2016). Algebraic rules for computing the regularization parameter of the Levenberg–Marquardt method. *Computational Optimization and Applications*, 65(3), 723–751. <https://doi.org/10.1007/s10589-016-9845-x>

Musa, Y. B., & Waziri, M. Y. (2018). Matrix-norm approach of computing Levenberg–Marquardt regularization parameter for nonlinear equations. *Bayero Journal of Pure and Applied Sciences*, 11(1), 502–513. <https://doi.org/10.4314/bajopas.v11i1.80s>

Musa, Y. B., Waziri, M. Y., & Halilu, A. S. (2017). On computing the regularization parameter for the Levenberg–Marquardt method via the spectral radius approach to solving systems of nonlinear equations. *Journal of Numerical Mathematics and Stochastics*, 9(1), 80–94.

Wang, H., & Fan, J. (2020). Convergence rate of the Levenberg–Marquardt method under Hölderian local error bound. *Optimization Methods and Software*, 35(4), 767–786. <https://doi.org/10.1080/10556788.2019.1694927>

Waziri, M. Y., & Sabi'u, J. (2015). A derivative-free conjugate gradient method and its global convergence for solving symmetric nonlinear equations. *International Journal of Mathematics and Mathematical Sciences*, 2015, Article 961487. <https://doi.org/10.1155/2015/961487>

Zeng, M., & Zhou, G. (2022). Improved convergence results of an efficient Levenberg–Marquardt method for nonlinear equations. *Journal of Applied Mathematics and Computing*, 68(6), 3655–3671. <https://doi.org/10.1007/s12190-021-01599-6>

Zhu, D., Yang, Y., & Cao, M. (2024). An accelerated adaptive two-step Levenberg–Marquardt method with the modified Metropolis criterion. *AIMS Mathematics*, 9(9), 24610–24635. <https://doi.org/10.3934/math.20241199>

APPENDIX

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) respectively.

Problem 1.

$$F(x) = \begin{pmatrix} 3 & -1 & & & \\ -1 & 3 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 3 \end{pmatrix} x + \begin{pmatrix} e^{x_1} - 1 \\ \vdots \\ e^{x_n} - 1 \end{pmatrix}, \quad (21)$$

with initial guess $x_0 = (1, 1, 1, \dots, 1)^T$.

Problem 2.

$$F(x) = \begin{pmatrix} 2 & -1 & & & \\ 0 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} x + \begin{pmatrix} \sin x_1 - 1 \\ \vdots \\ \sin x_n - 1 \end{pmatrix}, \quad (22)$$

with initial guess $x_0 = (1, 1, 1, \dots, 1)^T$.

Problem 3.

$$\begin{aligned} F_1(x) &= x_1(x_1^2 + x_2^2) - 1, \\ F_i(x) &= x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2), \quad i = 2, 3, \dots, n-1, \\ F_n(x) &= x_n(x_{n-1}^2 + x_n^2), \end{aligned} \quad (23)$$

with initial guess $x_0 = (1, 1, 1, \dots, 1)^T$.

Problem 4.

$$\begin{aligned} F_{3i-2}(x) &= x_{3i} - 2x_{3i-1} - x_{3i}^2 - 1, \\ F_{3i-1}(x) &= x_{3i} - 2x_{3i-2}x_{3i} - x_{3i-2}^2 + x_{3i-1}^2 - 2, \quad i = 1, \dots, \frac{n}{3}, \\ F_{3i}(x) &= e^{-x_{3i-2}} - e^{-x_{3i-1}}, \end{aligned} \quad (24)$$

with initial guess $x_0 = (1, 1, 1, \dots, 1)^T$.

Problem 5.

$$F_i(x) = (1 - x_i^2) + x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2, \quad i = 1, 2, \dots, n, \quad (25)$$

with initial guess $x_0 = (0.5, 0.5, 0.5, \dots, 0.5)^T$.

Problem 6.

$$\begin{aligned} F_1(x) &= x_1^2 - 3x_1 + 1 + \cos(x_1 - x_2), \\ F_i(x) &= x_1^2 - 3x_i + 1 + \cos(x_i - x_{i-1}), \quad i = 2, \dots, n, \end{aligned} \quad (26)$$

with initial guess $x_0 = (2, 2, 2, \dots, 2)^T$.

Problem 7. (Darvish and Shin,2011)

$$\begin{aligned} F_i(x) &= e_i^x - 1, \quad i = 1, 2, 3, \dots, n \\ \text{with initial guess } x_0 &= (0.02, 0.02, 0.02, \dots, 0.02)^T \end{aligned}$$