

Adaptive Modified Levenberg–Marquardt Algorithm for Nonlinear Least Squares: Analysis and Applications

Wang Shan, Igor Nikiforov

Abstract—This paper proposes an Adaptive Modified Levenberg–Marquardt (AMLML) algorithm for nonlinear least-squares problems. AMLML combines curvature-aware diagonal damping with reduction-ratio-based step acceptance and damping adjustment. It is compared with classical LM and Trust-Region Reflective (TRF) methods on exponential, power-law, logistic, and queueing-inspired models under different initialization and noise conditions. The results show comparable performance under favorable conditions, while AMLML provides greater robustness under poor initialization, strong parameter coupling, and ill-conditioning. In the most difficult cases, its success-rate advantage reaches 37 percentage points over LM and 11 percentage points over TRF. AMLML also performs reliably in near-singular conditions, although it generally requires greater computational effort and does not guarantee global convergence.

Keywords—Nonlinear least squares, Levenberg–Marquardt algorithm, adaptive diagonal damping, Cloud computing

I. INTRODUCTION

Nonlinear least-squares (NLS) problems arise widely in nonlinear regression, system identification, parameter estimation, and scientific data fitting [1]–[3]. They are also important in computing-system performance modeling, where model parameters are estimated from workload, utilization, response-time, and resource-consumption data [4], [5]. In heterogeneous computing environments, accurate performance models can further support scheduling and resource-allocation decisions [6]. Queueing theory provides a particularly useful basis for describing the rapid growth of delay when system utilization approaches service capacity [5], [24].

The Levenberg–Marquardt (LM) algorithm is one of the most widely used methods for solving NLS problems [7]–[10]. By combining the Gauss–Newton direction with a damping term, LM generally provides fast convergence when the initial estimate is sufficiently accurate and the Jacobian is well conditioned [9]–[12]. However, its performance may deteriorate under poor initialization, strong parameter coupling, unequal parameter sensitivities, ill-conditioning, or near-singular behavior [1], [10]–[12]. In these cases, scalar or insufficiently scaled damping may suppress useful directions while failing to stabilize highly sensitive ones.

To address these difficulties, this paper proposes an Adaptive Modified Levenberg–Marquardt (AMLML) algorithm that combines curvature-aware diagonal damping

with reduction-ratio-based damping adjustment. The diagonal damping matrix is constructed from the approximate Hessian to regulate different parameter directions, while the reduction ratio determines step acceptance and the subsequent update of the damping parameter. The contribution therefore lies in the specific combination of diagonal regularization, adaptive damping adjustment, and failure-handling mechanisms rather than in diagonal scaling alone.

AMLML is compared with classical LM and the Trust-Region Reflective (TRF) method using exponential saturation, power-law, logistic, and queueing-inspired models. The experiments consider different initialization and noise conditions and evaluate convergence success, residual error, parameter-estimation accuracy, runtime, and computational evaluations. The results indicate that AMLML improves practical robustness in several ill-conditioned and near-singular cases, although it generally requires greater computational effort and, like other local methods, does not guarantee global convergence.

The main contributions are as follows:

1. An AMLML method combining curvature-aware diagonal damping with adaptive damping adjustment.
2. A systematic comparison of AMLML, LM, and TRF under different initialization and noise conditions.
3. Validation on nonlinear models with different degrees of parameter coupling and conditioning.
4. An analysis of convergence, parameter degeneracy, conditioning, and failure modes.

The remainder of this paper is organized as follows. Section II reviews related work. Section III presents AMLML. Section IV describes the experiments and results. Section V discusses the findings and limitations, and Section VI concludes the paper.

II. RELATED WORK ON LEVENBERG–MARQUARDT-BASED NONLINEAR OPTIMIZATION METHODS

Gauss–Newton and LM are fundamental methods for nonlinear least-squares optimization [7]–[12]. Gauss–Newton approximates the Hessian using and can converge rapidly near a solution, but its performance depends on the accuracy and rank of the Jacobian [10]–[13]. LM improves numerical stability by adding a regularization term to the approximate Hessian [7]–[9]. Subsequent studies have examined convergence conditions, damping-parameter selection, rank deficiency, and implementation strategies for LM-type methods [14]–[16].

Trust-region methods provide another important framework for improving robustness by restricting each step to a region in which the local model is considered reliable

[11], [17], [18]. Coleman and Li developed an interior trust-region approach for bound-constrained optimization [19], while Branch et al. proposed a subspace method suitable for large-scale bound-constrained problems [20]. These studies form the methodological basis of the TRF algorithm used in the present comparison.

Scaling is also important when model parameters have different magnitudes or sensitivities. Classical LM implementations already incorporate diagonal or implicit scaling ideas [8], [9], while later work has investigated alternative scaling matrices for structured nonlinear least-squares problems [21]. From a geometric perspective, unequal sensitivities and narrow parameter directions can produce highly anisotropic objective-function landscapes, making optimization difficult [22], [23].

Building on these studies, AMLM uses the diagonal of the approximate Hessian to construct parameter-specific damping and combines it with a reduction-ratio mechanism for accepting steps and adjusting regularization. The method is intended to improve robustness when parameter directions are unevenly scaled or locally ill-conditioned, while retaining the computational structure of LM.

III. PROPOSED ADAPTIVE MODIFIED LEVENBERG–MARQUARDT METHOD

A. Problem Formulation

We consider the nonlinear least squares problem of the form:

$$\min_{\theta} F(\theta) = \frac{1}{2} \|r(\theta)\|^2 = \frac{1}{2} \sum_{i=1}^m r_i^2(\theta)$$

where $r(\theta) \in \mathbb{R}^m$ denotes the residual vector and $\theta \in \mathbb{R}^n$ is the parameter vector to be estimated. The gradient of the objective function is given by:

$$g = \nabla F(\theta) = J^T r$$

and the Gauss–Newton approximation of the Hessian matrix is:

$$H \approx J^T J$$

where $J = \frac{\partial r}{\partial \theta}$ is the Jacobian matrix.

However, due to its complete reliance on local linearization, the Gauss–Newton method may produce excessively large parameter updates when the Jacobian is poorly conditioned or the model exhibits strong nonlinearity, leading to numerical instability or even divergence. Therefore, in practical applications, it is often necessary to introduce a damping mechanism to enhance robustness.

B. Classical Levenberg–Marquardt Method

In the Levenberg–Marquardt algorithm, the introduced term λI can be interpreted as an l_2 -norm regularization on the parameter increment. Here, the identity matrix I ensures that all parameter directions are equally damped, thereby improving numerical stability and preventing excessively large updates in directions with low curvature. Consequently, the LM method can be viewed as an adaptive interpolation strategy between the Gauss–Newton method and gradient descent.

The classical Levenberg–Marquardt (LM) method computes the parameter update $\Delta\theta$ by solving:

$$(J^T J + \lambda I) \Delta\theta = -J^T r,$$

where $\lambda > 0$ is a damping parameter and I is the identity matrix. This formulation can be interpreted as an

interpolation between the Gauss–Newton method ($\lambda \rightarrow 0$), and gradient descent ($\lambda \rightarrow \infty$).

While LM is robust in many practical problems, the use of the isotropic damping matrix I introduces a key limitation: all parameter directions are treated equally, regardless of their sensitivity or curvature, regardless of their sensitivity or curvature. As a result, LM may exhibit inefficient updates or convergence issues in problems with strong parameter coupling or heterogeneous scaling.

C. Proposed Safeguarded Diagonal Damping

To account for differences in parameter sensitivity and scaling, the proposed AMLM method employs a safeguarded diagonal damping matrix. At iteration k , let

$$r_k = r(\theta_k), J_k = J(\theta_k),$$

and define

$$H_k = J_k^T J_k, g_k = J_k^T r_k.$$

Instead of using the identity matrix in the damping term, a diagonal scaling matrix is constructed from the diagonal elements of the Gauss–Newton matrix:

$$D_k = \text{diag}(d_{1,k}, d_{2,k}, \dots, d_{n,k}),$$

where

$$d_{i,k} = \max\{(H_k)_{ii}, \delta \max(1, \|H_k\|_{\infty})\}, i = 1, \dots, n,$$

and $\delta > 0$ is a small safeguard parameter.

The safeguard prevents zero or extremely small diagonal elements and therefore ensures that

$$D_k > 0.$$

The AMLM step p_k is computed by solving

$$(H_k + \lambda_k D_k) p_k = -g_k,$$

or, equivalently,

$$(J_k^T J_k + \lambda_k D_k) p_k = -J_k^T r_k,$$

where $\lambda_k > 0$ is the damping parameter.

Since $H_k = J_k^T J_k$ is positive semidefinite, $\lambda_k > 0$, and $D_k > 0$, the matrix

$$H_k + \lambda_k D_k$$

is positive definite. Consequently, the AMLM step is uniquely defined even when $J_k^T J_k$ is singular or nearly singular.

The diagonal scaling introduces parameter-dependent regularization. Parameter directions associated with larger diagonal elements of H_k receive stronger absolute damping, whereas directions with smaller local sensitivities receive correspondingly scaled damping. This mechanism reduces the influence of differences in parameter magnitude and sensitivity on the computation of the optimization step.

Diagonal scaling based on the Gauss–Newton matrix has previously been considered in Marquardt-type methods. Therefore, the contribution of the proposed AMLM method does not consist solely in replacing the identity matrix by $\text{diag}(J_k^T J_k)$. Its contribution lies in combining safeguarded curvature-dependent diagonal scaling with reduction-ratio-based step acceptance and adaptive damping updates for poorly initialized and near-singular nonlinear least-squares problems.

D. Adaptive Damping Parameter Update

The quality of the trial step is evaluated by comparing the actual reduction in the objective function with the reduction predicted by the local quadratic model.

The Gauss–Newton quadratic model is defined as

$$m_k(p) = F(\theta_k) + g_k^T p + \frac{1}{2} p^T H_k p.$$

The predicted reduction is

$$\text{pred}_k = m_k(0) - m_k(p_k) = -g_k^\top p_k - \frac{1}{2} p_k^\top H_k p_k.$$

Because the AMLM step satisfies

$$(H_k + \lambda_k D_k) p_k = -g_k,$$

the predicted reduction can equivalently be written as

$$\text{pred}_k = \frac{1}{2} p_k^\top (\lambda_k D_k p_k - g_k).$$

The actual reduction is

$$\text{ared}_k = F(\theta_k) - F(\theta_k + p_k).$$

The reduction ratio is therefore defined as

$$\rho_k = \frac{\text{ared}_k}{\text{pred}_k}.$$

A trial step is accepted only when it produces a sufficient actual reduction:

$$\theta_{k+1} = \begin{cases} \theta_k + p_k, & \rho_k \geq \eta_0, \\ \theta_k, & \rho_k < \eta_0. \end{cases}$$

The damping parameter is updated according to

$$\lambda_{k+1} = \begin{cases} \max(\lambda_{\min}, \gamma_{\text{dec}} \lambda_k), & \rho_k \geq \eta_2, \\ \lambda_k, & \eta_1 \leq \rho_k < \eta_2, \\ \min(\lambda_{\max}, \gamma_{\text{inc}} \lambda_k), & \rho_k < \eta_1, \end{cases}$$

where

$$0 \leq \eta_0 < \eta_1 < \eta_2 < 1, 0 < \gamma_{\text{dec}} < 1 < \gamma_{\text{inc}}.$$

When ρ_k is large, the local quadratic model provides an accurate prediction and the damping parameter is decreased. When ρ_k is small or negative, the trial step is rejected or treated as unreliable, and the damping parameter is increased.

E. Complete AMLM Algorithm

The complete AMLM procedure is summarized in Algorithm 1. At each iteration, the algorithm constructs a safeguarded diagonal scaling matrix, computes a damped Gauss–Newton step, and evaluates the agreement between the actual and predicted reductions. A trial step is accepted only if it produces a sufficient reduction in the objective function. Otherwise, the current parameter vector is retained and the damping parameter is increased.

The algorithm terminates when at least one of the gradient, step-size, or objective-reduction criteria is satisfied. A run is classified as unsuccessful if the maximum computational budget is exhausted, a nonfinite numerical value is encountered, or the damped linear system cannot be solved.

Algorithm 1. Adaptive Modified Levenberg–Marquardt Algorithm

Input: Initial parameter vector θ_0 ; initial damping parameter $\lambda_0 > 0$; safeguard parameter $\delta > 0$; thresholds η_0, η_1, η_2 ; damping factors γ_{dec} and γ_{inc} ; tolerances $\varepsilon_g, \varepsilon_x, \varepsilon_f$; maximum number of iterations $K_{\max}$.

Output: Estimated parameter vector $\hat{\theta}$; final objective value $F(\hat{\theta})$; convergence status.

Set $k = 0$, $\theta_k = \theta_0$, and $\lambda_k = \lambda_0$.

Evaluate the residual vector and objective function:

$$r_k = r(\theta_k), F_k = \frac{1}{2} \|r_k\|_2^2.$$

While $k < K_{\max}$, perform the following steps:

3.1. Evaluate the Jacobian matrix:

$$J_k = J(\theta_k).$$

3.2. Compute the Gauss–Newton matrix and gradient:

$$H_k = J_k^\top J_k, g_k = J_k^\top r_k.$$

3.3. Check the gradient termination condition. If

$$\|g_k\|_\infty \leq \varepsilon_g,$$

terminate successfully.

3.4. Construct the safeguarded diagonal matrix:

$$D_k = \text{diag}(d_{1,k}, d_{2,k}, \dots, d_{n,k}),$$

where

$$d_{i,k} = \max\{(H_k)_{ii}, \delta \max(1, \|H_k\|_\infty)\}.$$

3.5. Compute the AMLM trial step p_k by solving

$$(H_k + \lambda_k D_k) p_k = -g_k.$$

If the linear system cannot be solved or p_k contains NaN or Inf, classify the run as unsuccessful and terminate.

3.6. Check the step-size termination condition. If

$$\|p_k\|_2 \leq \varepsilon_x (\|\theta_k\|_2 + \varepsilon_x),$$

terminate successfully.

3.7. Calculate the predicted reduction:

$$\text{pred}_k = -g_k^\top p_k - \frac{1}{2} p_k^\top H_k p_k.$$

Equivalently,

$$\text{pred}_k = \frac{1}{2} p_k^\top (\lambda_k D_k p_k - g_k).$$

3.8. If $\text{pred}_k \leq 0$ or is nonfinite, reject the trial step, set

$$\theta_{k+1} = \theta_k,$$

and increase the damping parameter:

$$\lambda_{k+1} = \min(\lambda_{\max}, \gamma_{\text{inc}} \lambda_k).$$

Then set $k \leftarrow k + 1$ and continue to the next iteration.

3.9. Compute the trial parameter vector:

$$\theta_k^{\text{trial}} = \theta_k + p_k.$$

Evaluate

$$r_k^{\text{trial}} = r(\theta_k^{\text{trial}})$$

and

$$F_k^{\text{trial}} = \frac{1}{2} \|r_k^{\text{trial}}\|_2^2.$$

3.10. Calculate the actual reduction:

$$\text{ared}_k = F_k - F_k^{\text{trial}}.$$

3.11. Calculate the reduction ratio:

$$\rho_k = \frac{\text{ared}_k}{\text{pred}_k}.$$

3.12. Accept or reject the trial step:

$$\theta_{k+1} = \begin{cases} \theta_k^{\text{trial}}, & \rho_k \geq \eta_0, \\ \theta_k, & \rho_k < \eta_0. \end{cases}$$

If the trial step is accepted, set

$$r_{k+1} = r_k^{\text{trial}}, F_{k+1} = F_k^{\text{trial}}.$$

If the trial step is rejected, set

$$r_{k+1} = r_k, F_{k+1} = F_k.$$

3.13. Update the damping parameter:

$$\lambda_{k+1} = \begin{cases} \max(\lambda_{\min}, \gamma_{\text{dec}} \lambda_k), & \rho_k \geq \eta_2, \\ \lambda_k, & \eta_1 \leq \rho_k < \eta_2, \\ \min(\lambda_{\max}, \gamma_{\text{inc}} \lambda_k), & \rho_k < \eta_1. \end{cases}$$

3.14. If the step was accepted, check the relative objective-reduction criterion:

$$|F_k - F_{k+1}| \leq \varepsilon_f (F_k + \varepsilon_f).$$

If this condition is satisfied, terminate successfully.

3.15. Set

$$k \leftarrow k + 1$$

and continue.

If $k = K_{\max}$, terminate and classify the run as unsuccessful because the iteration budget has been exhausted.

Return

$$\hat{\theta} = \theta_k$$

together with the objective value, iteration count,

evaluation count and convergence status.

The safeguard in D_k ensures that the damped normal-equation matrix remains positive definite for every $\lambda_k > 0$. The rejection mechanism prevents an unsuccessful trial point from replacing the current iterate. Increasing λ_k after an unsuccessful step makes the subsequent step more conservative, whereas decreasing λ_k after a successful step allows the method to approach the Gauss–Newton regime near a solution.

The algorithm also returns an explicit status code. In the experiments, successful termination corresponds to satisfaction of at least one prescribed convergence criterion. Reaching the maximum iteration or evaluation budget, encountering nonfinite values, or failing to solve the damped linear system is treated as an unsuccessful run rather than being silently removed from the statistical analysis.

F. Theoretical Analysis

1) Assumptions

The theoretical analysis is based on the following standard assumptions.

Assumption 1. The residual function $r(\theta)$ is continuously differentiable on an open set containing the level set

$$\mathcal{L}(\theta_0) = \{\theta: F(\theta) \leq F(\theta_0)\}.$$

Assumption 2. The level set $\mathcal{L}(\theta_0)$ is bounded and the objective function F is bounded below.

Assumption 3. The Jacobian is Lipschitz continuous on the level set:

$$\|J(\theta) - J(z)\| \leq L_J \|\theta - z\|.$$

Assumption 4. The safeguarded scaling matrices satisfy

$$0 < d_{\min} I \leq D_k \leq d_{\max} I.$$

2) Positive Definiteness and Descent Property

Proposition 1. If $\lambda_k > 0$ and $D_k > 0$, then the matrix

$$B_k = H_k + \lambda_k D_k$$

is symmetric positive definite. Consequently, the AMLM step is uniquely defined and is a descent direction whenever $g_k \neq 0$.

Proof. For any nonzero vector v ,

$$v^T B_k v = v^T J_k^T J_k v + \lambda_k v^T D_k v = \|J_k v\|^2 + \lambda_k v^T D_k v > 0.$$

Therefore, B_k is positive definite and nonsingular. Furthermore,

$$g_k^T p_k = -p_k^T B_k p_k < 0$$

for $p_k \neq 0$. Hence, p_k is a descent direction.

3) Influence on the Condition Number

To analyze the effect of diagonal damping, define the scaled Gauss–Newton matrix

$$\tilde{H}_k = D_k^{-1/2} H_k D_k^{-1/2}.$$

Using the transformed variable

$$s_k = D_k^{1/2} p_k,$$

the AMLM system becomes

$$(\tilde{H}_k + \lambda_k I) s_k = -D_k^{-1/2} g_k.$$

Let $\mu_{\min,k}$ and $\mu_{\max,k}$ denote the smallest and largest eigenvalues of $\tilde{H}_k$. Since $\tilde{H}_k$ is positive semidefinite,

$$\kappa_2(\tilde{H}_k + \lambda_k I) = \frac{\mu_{\max,k} + \lambda_k}{\mu_{\min,k} + \lambda_k}.$$

If $\mu_{\max,k} > \mu_{\min,k}$, then

$$\frac{\partial}{\partial \lambda_k} \kappa_2(\tilde{H}_k + \lambda_k I) = \frac{\mu_{\min,k} - \mu_{\max,k}}{(\mu_{\min,k} + \lambda_k)^2} < 0.$$

Thus, increasing the damping parameter monotonically reduces the condition number of the scaled normal equations. Moreover,

$$\lim_{\lambda_k \rightarrow \infty} \kappa_2(\tilde{H}_k + \lambda_k I) = 1.$$

For the original unscaled system, the following bound holds:

$$\kappa_2(H_k + \lambda_k D_k) \leq \kappa_2(D_k) \kappa_2(\tilde{H}_k + \lambda_k I).$$

Therefore, the diagonal damping improves the conditioning of the scaled system, while the conditioning of the original system also depends on the variation among the diagonal elements of D_k . This distinction is important because diagonal damping does not necessarily reduce the unscaled condition number in every problem.

4) Convergence Properties

Theorem 1. Suppose that Assumptions 1–4 hold. Assume that unsuccessful trial steps are rejected, that λ_k is increased after unsuccessful steps, and that every accepted step satisfies

$$F(\theta_k) - F(\theta_{k+1}) \geq \eta_0 \text{pred}_k, \eta_0 > 0.$$

Then the sequence of accepted objective values is nonincreasing, and

$$\liminf_{k \rightarrow \infty} \|\nabla F(\theta_k)\| = 0.$$

Hence, every accumulation point at which the Jacobian and gradient are continuous is a first-order stationary point.

Proof sketch. Accepted steps produce a sufficient decrease in F , whereas rejected steps leave the parameter vector unchanged and increase the damping parameter. Since F is bounded below, the sum of the accepted reductions is finite. If the gradient norm were bounded away from zero, the positive definiteness of $H_k + \lambda_k D_k$, together with the boundedness assumptions, would imply a uniformly positive predicted reduction for sufficiently damped steps. The sufficient-decrease condition would then force an infinite total reduction in F , contradicting its lower boundedness. Consequently, the gradient norm must approach zero along a subsequence.

5) Local Convergence Rate

Theorem 2. Let θ^* be a stationary solution, and suppose that $J(\theta^*)$ has full column rank. If J is Lipschitz continuous in a neighborhood of θ^* , $\lambda_k \rightarrow 0$, and full AMLM steps are eventually accepted, then the iterates converge locally to θ^* .

For a general nonzero-residual problem, the local convergence is at least linear under the standard Gauss–Newton regularity conditions:

$$\|\theta_{k+1} - \theta^*\| \leq C \|\theta_k - \theta^*\|.$$

If

$$r(\theta^*) = 0$$

and the damping parameter decreases sufficiently rapidly, for example

$$\lambda_k = O(\|\nabla F(\theta_k)\|),$$

then the AMLM step asymptotically approaches the Gauss–Newton step and quadratic local convergence can be obtained:

$$\|\theta_{k+1} - \theta^*\| \leq C \|\theta_k - \theta^*\|^2.$$

Therefore, diagonal damping improves stability away from the solution without changing the asymptotic Gauss–Newton behavior when the damping parameter tends to zero.

IV. EXPERIMENTAL EVALUATION

A. Experimental Setup

To evaluate the performance of the proposed AMLM algorithm, we conduct a series of experiments on representative nonlinear models. The experiments are designed to assess robustness, convergence behavior, and sensitivity to initialization under different problem conditions.

1) Data Generation and Noise Conditions

Synthetic datasets were generated from predefined nonlinear models according to

$$y_i = f(x_i; \theta_{\text{true}}) + \varepsilon_i, \quad i = 1, \dots, m,$$

where θ_{true} denotes the ground-truth parameter vector and

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2).$$

Four noise levels were considered:

$$\sigma \in 0.01, 0.05, 0.1$$

The condition $\sigma = 0$ represents noise-free observations, whereas $\sigma = 0.01, 0.05$, and 0.1 represent low-, medium-, and high-noise conditions, respectively.

For every stochastic trial, a new noise vector was generated using a predetermined random seed. The same generated dataset was subsequently supplied to all compared algorithms. This paired design ensures that differences between algorithms are not caused by differences in the generated observations.

The random seeds used in the experiments were

$$s_j = s_0 + j, \quad j = 1, \dots, 100,$$

where $s_0 = 42$ is the initial seed. All input values, ground-truth parameters and noise realizations were stored to ensure reproducibility.

2) Initialization Conditions and Paired Trials

Three initialization categories were considered to evaluate the sensitivity of the algorithms to the starting point:

- good initialization, denoted by I_1 , with parameters close to the ground-truth values;
- moderate initialization, denoted by I_2 , with parameters moderately displaced from the ground truth;
- poor initialization, denoted by I_3 , with parameters substantially displaced from the ground truth.

The central initial parameter vectors used for each model are reported in Table 1.

Table 1. Ground-truth and initial parameter vectors

Model	True parameters	Good I_1	Moderate I_2	Poor I_3
Exponential	a=2.5 b=1.3 c=0.2	a=2.0 b=1.0 c=0.1	a=5.8 b=7.3 c=7.2	a=8.0 b=0.05 c=-2.5
Power-law	a=2.0 b=1.5 c=0.5	a=2.5 b=1.7 c=0.2	a=4.0 b=2.0 c=1.0	a=4.5 b=-2 c=1.2
Logistic	a=5.0 b=2.0 c=2.5	a=5.5 b=1.5 c=1.5	a=10.0 b=5.5 c=5.5	a=8.0 b=0.1 c=-2.5

For noisy conditions, 100 independently generated noise realizations were used for every combination of model, initialization category and noise level. The same noise realization and the same initial vector were supplied to all compared algorithms.

In the noise-free condition, the dataset and initial vector were deterministic. Therefore, each algorithm was executed once for every model and initialization category. Repeated deterministic executions were not treated as independent experimental observations.

3) Compared Algorithms

The following nonlinear least-squares algorithms were included in the comparison:

- classical Levenberg–Marquardt with identity-based damping, denoted by LM;
- the Trust-Region Reflective method, denoted by TRF;
- the proposed Adaptive Modified Levenberg–Marquardt method, denoted by AMLM.

LM and TRF were implemented using `scipy.optimize.least_squares`, whereas AMLM was implemented using the update procedure described in Algorithm 1.

All algorithms received identical residual functions, Jacobian matrices, datasets and initial parameter vectors. The same stopping tolerances and maximum residual-evaluation budget were used whenever supported by the corresponding implementation.

No parameter bounds were imposed in the main comparison. Consequently, TRF was evaluated as an unconstrained trust-region method rather than being provided with additional information unavailable to LM and AMLM.

4) Algorithm Parameters and Stopping Criteria

To ensure a fair comparison, all algorithms were evaluated using the same convergence tolerances and the same maximum number of residual-function evaluations whenever these settings were supported by the corresponding implementation.

Three stopping criteria were considered: the gradient criterion, the step-size criterion and the objective-reduction criterion.

The gradient criterion was defined as

$$\|g_k\|_\infty = \|J_k^T r_k\|_\infty \leq \varepsilon_g.$$

The step-size criterion was defined as

$$\|p_k\|_2 \leq \varepsilon_x (\|\theta_k\|_2 + \varepsilon_x).$$

The objective-reduction criterion was defined as

$$|F(\theta_{k+1}) - F(\theta_k)| \leq \varepsilon_f (F(\theta_k) + \varepsilon_f).$$

A run was regarded as successfully converged when at least one of these criteria was satisfied and all returned parameters and objective values were finite. Reaching the maximum computational budget without satisfying a convergence criterion was treated as an unsuccessful run.

The gradient, step-size and objective-function tolerances were set to

$$\varepsilon_g = 10^{-8}, \quad \varepsilon_x = 10^{-8}, \quad \varepsilon_f = 10^{-8}.$$

The maximum number of residual-function evaluations was set to

$$N_{\text{fev}, \text{max}} = 2000.$$

For AMLM, the initial damping parameter was set to

$$\lambda_0 = 10^{-3}.$$

The safeguarded diagonal matrix was constructed using

$$\delta = 10^{-12}.$$

The step-acceptance and damping-update thresholds were

$$\eta_0 = 10^{-4}, \quad \eta_1 = 0.25, \quad \eta_2 = 0.75.$$

After a highly successful step, the damping parameter was reduced using

$$\gamma_{\text{dec}} = 0.5,$$

whereas after an unsuccessful step it was increased using

$$\gamma_{\text{inc}} = 2.$$

To avoid numerical underflow or overflow, the damping parameter was restricted to

$$\lambda_{\text{min}} = 10^{-15}, \lambda_{\text{max}} = 10^{15}.$$

The complete parameter settings are summarized in Table 2.

Table 2. Algorithm parameters used in the experiments

Parameter	LM	TRF	AMLM
Gradient tolerance (ϵ_g)	(10 ⁻⁸)	(10 ⁻⁸)	(10 ⁻⁸)
Step tolerance (ϵ_s)	(10 ⁻⁸)	(10 ⁻⁸)	(10 ⁻⁸)
Objective tolerance (ϵ_f)	(10 ⁻⁸)	(10 ⁻⁸)	(10 ⁻⁸)
Maximum residual evaluations ($N_{\text{res,max}}$)	2000	2000	2000
Initial damping (λ_0)	Implementation default	—	(10 ⁻³)
Diagonal safeguard (δ)	—	—	(10 ⁻¹²)
Acceptance threshold (η_0)	—	—	(10 ⁻⁴)
Lower update threshold (η_l)	—	—	0.25
Upper update threshold (η_u)	—	—	0.75
Damping decrease factor (γ_{dec})	—	—	0.5
Damping increase factor (γ_{inc})	—	—	2
Minimum damping (λ_{min})	—	—	(10 ⁻¹⁵)
Maximum damping (λ_{max})	—	—	(10 ¹⁵)

5) Failure Detection and Treatment of Unsuccessful Runs

A run was classified as unsuccessful if at least one of the following events occurred:

- the maximum iteration or residual-evaluation budget was reached without satisfying a convergence criterion;
- the parameter vector, residual vector, Jacobian matrix or objective value contained NaN or Inf;
- the damped linear system could not be solved;
- the parameter vector entered an invalid region of the model;
- the parameter norm exceeded the numerical threshold θ_{max} ;
- the objective function exceeded the numerical threshold F_{max} .

For the queueing-inspired model

$$y = \frac{a}{b - x} + c,$$

a run was additionally classified as unsuccessful if

$$\min_i |b - x_i| < \epsilon_{\text{sing}},$$

because the model and its Jacobian become numerically unstable when the denominator approaches zero.

Unsuccessful runs were retained in the experimental record and included in the calculation of the convergence success rate:

$$R_{\text{success}} = \frac{N_{\text{successful}}}{N_{\text{total}}}.$$

RSS, parameter-estimation error and iteration statistics were calculated over successfully converged runs and were accompanied by the corresponding number of successful runs. Unsuccessful runs were not silently removed or replaced.

Successfully completed runs with unusually large parameter errors, residuals or runtimes were retained. Because the resulting distributions may be skewed, medians and interquartile ranges were reported in addition to arithmetic means.

6) Evaluation Metrics and Statistical Analysis

The algorithms were evaluated in terms of accuracy, convergence reliability, computational efficiency and numerical conditioning.

The residual sum of squares was calculated as

$$\text{RSS} = \sum_{i=1}^m [y_i - f(x_i; \theta)]^2.$$

To compare parameter recovery across models and parameters with different scales, the relative parameter error was defined as

$$E_{\theta} = \frac{\|\hat{\theta} - \theta_{\text{true}}\|_2}{\|\theta_{\text{true}}\|_2 + \epsilon_{\theta}},$$

where $\epsilon_{\theta} > 0$ prevents division by zero.

Convergence reliability was evaluated using the success rate and the distribution of failure reasons. Computational efficiency was evaluated using runtime, iteration count, residual-function evaluations and Jacobian evaluations.

The total evaluation count was defined as

$$N_{\text{eval}} = N_{\text{res}} + N_{\text{jac}},$$

where N_{res} and N_{jac} denote the numbers of residual-function and Jacobian evaluations, respectively.

Numerical conditioning was assessed using

$$\kappa_2(H_k), \kappa_2(H_k + \lambda_k I),$$

and

$$\kappa_2(D_k^{-1/2} H_k D_k^{-1/2} + \lambda_k I).$$

For repeated stochastic trials, results were summarized using the mean, median, standard deviation and interquartile range. Because all algorithms were evaluated using the same datasets and initial vectors, algorithm comparisons were performed using paired observations. The number of successful and unsuccessful runs was reported separately for every experimental condition.

The complete stochastic experimental design included three initialization categories, three nonzero noise levels and 100 paired trials for each model:

$$3 \times 3 \times 100 = 900$$

stochastic trials per algorithm and per model. The noise-free condition was analyzed separately using either deterministic runs or initialization perturbations, as specified above.

B. Multi-Model Validation

To evaluate whether AMLM generalizes across nonlinear least-squares problems with different structural properties, three representative models were considered: exponential saturation, power-law, and logistic models. These models exhibit different degrees of parameter coupling, curvature variation, and sensitivity to initialization.

AMLM was compared with classical LM and the Trust-Region Reflective (TRF) method under three initialization levels and four noise levels, $\sigma \in \{0.01, 0.05, 0.1\}$. For each stochastic condition, 100 paired trials were performed using identical datasets, noise realizations, and initial parameter vectors for all three algorithms.

The comparison considered convergence success rate, RSS, relative parameter error, runtime, and computational evaluations. Failed runs—including numerical overflow, invalid model values, and termination without satisfying the

convergence criteria—were retained in the success-rate calculation. Accuracy and efficiency statistics were computed only over successfully converged runs, with the number of successful runs reported for each condition. Parameter trajectories and fitted curves were additionally examined to identify cases in which numerical convergence did not correspond to correct parameter recovery or nonlinear structure.

- **Exponential Saturation Model**

The exponential saturation model is defined as $y(x; \theta) = a(1 - e^{-bx}) + c$, $\theta^* = [2.5, 1.3, 0.2]^T$. Synthetic observations were generated as $y_i^{obs} = a^*(1 - e^{-b^*x_i}) + c^* + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ where $\sigma \in \{0, 0.01, 0.05, 0.10\}$.

LM, TRF, and AMLM were evaluated using good, moderate, and poor initial estimates. For each combination of initialization and noise level, 100 independent runs were performed.

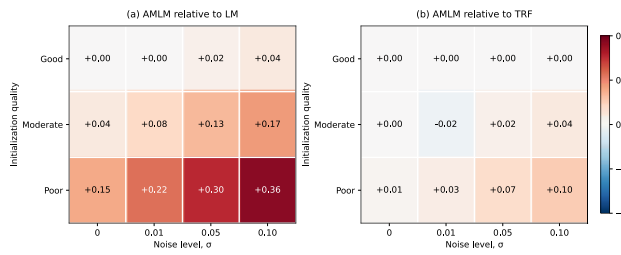
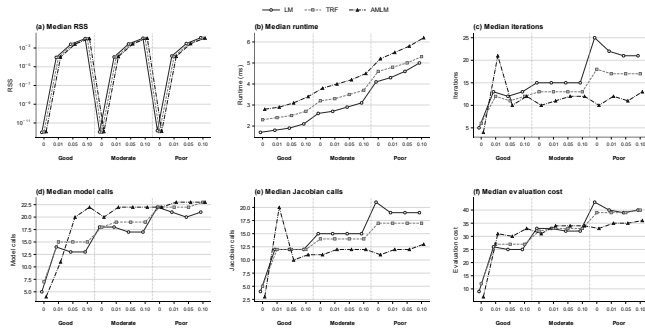


Figure 1 Difference in convergence success rate between AMLM and LM or TRF for the exponential saturation model

As shown in Figure 1, the three algorithms have similar success rates under good initialization. The advantage of



AMLM increases as the initial estimate deteriorates and the noise level rises. Under poor initialization and $\sigma = 0.10$, its success-rate advantage reaches 0.36 over LM and 0.1 over TRF. Thus, TRF and AMLM are more robust than LM, while AMLM performs best under the most difficult conditions.

Figure 2 Accuracy and computational characteristics of LM, TRF, and AMLM for the exponential saturation model

Figure 2 shows that the three methods obtain similar RSS values. LM has the shortest runtime, whereas AMLM generally requires more model evaluations because of its trial-step acceptance procedure. However, under poor initialization, AMLM requires fewer iterations and Jacobian evaluations and achieves the lowest total evaluation cost. Therefore, its additional runtime is compensated by improved robustness and reduced dependence on expensive Jacobian recomputation.

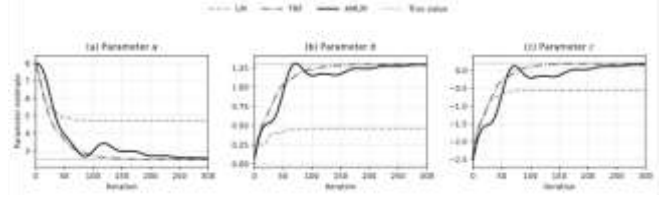


Figure 3 Parameter trajectories of LM, TRF, and AMLM under poor initialization

For the poor initial estimate $\theta_0 = [8.0, 0.05, -2.5]^T$, LM converges to an incorrect parameter combination, whereas TRF and AMLM recover the true solution. AMLM exhibits temporary nonmonotonic oscillations caused by adaptive adjustment of the damping parameters but ultimately converges to the true parameter vector.

Another important phenomenon is observed when the initial estimate deviates substantially from the true parameters, for example, $\theta_0 = [5.0, -0.1, 1.0]^T$. As shown in Figure 4, LM consistently converges to an incorrect linear-like solution, almost independently of the noise level.

This behavior is explained by the local degeneracy of the exponential model. When b approaches zero,

$$1 - e^{-bx} \approx bx$$

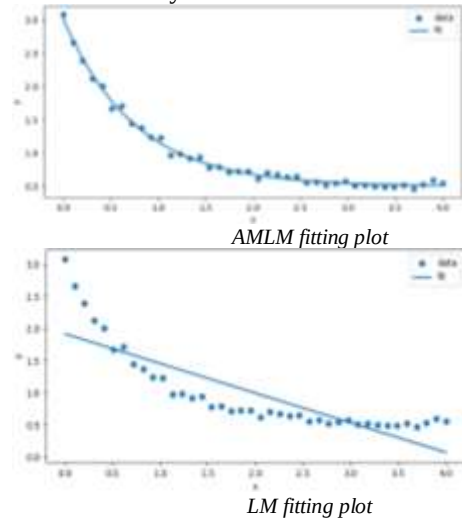
and the model becomes approximately

$$y(x; \theta) \approx abx + c$$

In this region, the individual effects of a and b become difficult to distinguish. Consequently, the Jacobian-based LM update may be misled by an inaccurate local approximation and converge to an incorrect parameter combination without recovering the nonlinear saturation structure.

In contrast, AMLM successfully recovers the correct nonlinear structure. Its adaptive diagonal damping applies different regularization strengths to different parameter directions, thereby preventing misleading local curvature information from dominating the update. This result demonstrates that AMLM is less sensitive to poor initialization and is better able to escape degenerate parameter regions.

Overall, classical LM remains efficient under favorable initialization, while TRF provides improved stability at a moderate computational cost. AMLM has the greatest runtime overhead but offers the highest robustness under poor initialization, elevated noise, and strongly nonuniform parameter sensitivity.



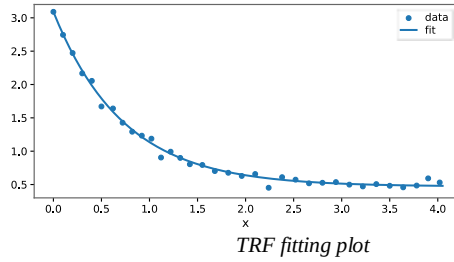


Figure 4 Fitting results of AMLM, LM and TRF under poor initialization for the exponential model

- Power-Law Model

The second experiment considers the power-law model $y(x; \theta) = ax^b + c$, $\theta^* = [2.0, 1.5, 0.5]^T$

This model is sensitive to the exponent b , especially when the initial estimate is negative or far from its true value. LM, TRF, and AMLM were evaluated under good, moderate, and poor initialization at four noise levels:

$$\sigma \in \{0, 0.01, 0.05, 0.10\}$$

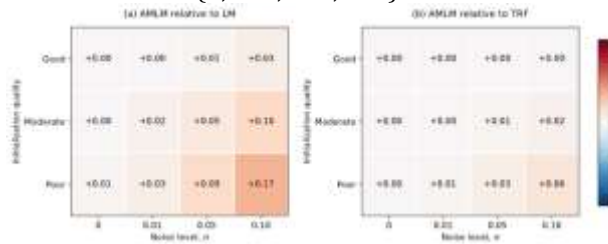


Figure 5 Difference in convergence success rate between AMLM and LM or TRF for the power-law model

As shown in Figure 5, all three algorithms have similar success rates under favorable conditions. The advantage of AMLM increases with noise and initialization error. Under poor initialization and $\sigma = 0.1$, its success-rate advantage reaches 0.17 over LM and 0.06 over TRF. AMLM therefore provides the greatest benefit when exponent estimation becomes difficult.

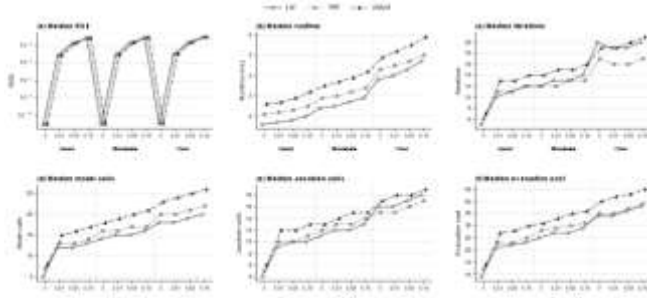


Figure 6 Accuracy and computational characteristics of LM, TRF, and AMLM for the power-law model

Figure 6 shows that the three algorithms obtain similar RSS values. LM has the shortest runtime and the lowest computational cost, whereas AMLM requires more iterations, model evaluations, and Jacobian evaluations. Thus, the improved robustness of AMLM is accompanied by additional computational overhead. TRF generally provides an intermediate compromise between efficiency and stability.

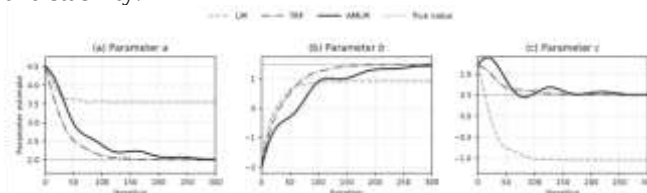


Figure 7 Parameter trajectories of LM, TRF, and AMLM under poor initialization

For the poor initial estimate

$$\theta_0 = [4.5, -2.0, 1.2]^T$$

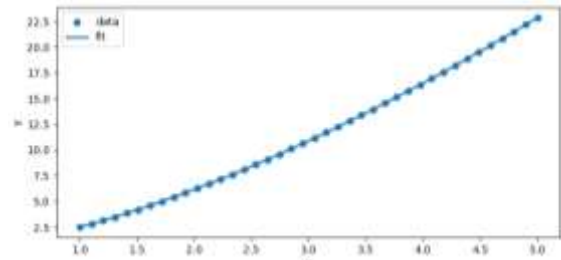
LM converges to an incorrect parameter combination, with approximately $a = 3.55$, $b = 0.9$, and $c = -1.05$. TRF rapidly approaches the true values, whereas AMLM converges more gradually and exhibits damped oscillations. Nevertheless, AMLM ultimately recovers the correct parameter vector.

This experiment demonstrates that LM remains the most efficient method under favorable conditions, while TRF and AMLM are less sensitive to poor initialization. AMLM provides the highest success rate in difficult cases, although TRF achieves comparable stability with lower computational cost.

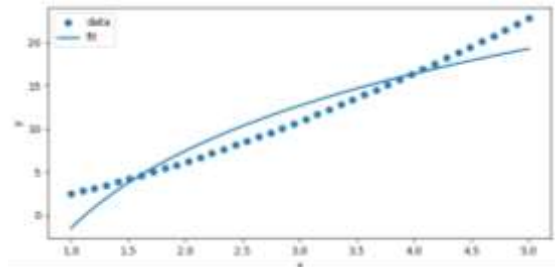
A second noteworthy phenomenon is observed for the initial estimate

$$\theta_0 = [4.5, -2.0, 1.2]^T.$$

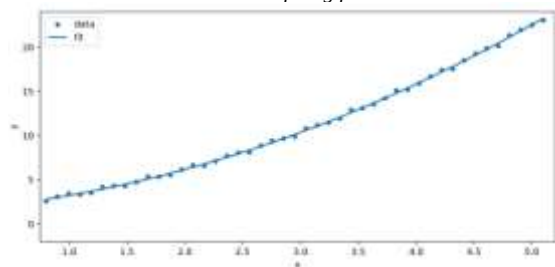
As shown in Figure 8, LM converges to a solution that deviates substantially from the true nonlinear trend. Although the fitted curve is smooth, it fails to reproduce the curvature of the underlying power-law relationship, particularly at larger values of x . In contrast, AMLM successfully recovers the nonlinear structure and closely matches the observed data.



AMLM fitting plot



LM fitting plot



TRF fitting plot

Figure 8 Fitting results of AMLM, LM and TRF under poor initialization for the power-law model

This behavior is associated with the strong coupling and unequal sensitivities of the power-law parameters. Under poor initialization, the standard LM step may be dominated by certain parameter directions and converge to a stable but inaccurate parameter combination. The adaptive diagonal damping of AMLM regulates each direction separately, thereby reducing parameter imbalance and enabling recovery of the correct curvature.

This observation further demonstrates that a smooth fitted curve and numerical convergence do not necessarily indicate correct parameter identification. AMLM is more robust in preserving the nonlinear structure of the power-law model under poor initialization.

- Logistic (Sigmoid) Model

The third experiment considers the logistic model

$$y(x; \theta) = \frac{a}{1 + e^{-b(x-c)}}, \quad \theta^* = [5.0, 2.0, 2.5]^T$$

Parameter a determines the upper asymptote, b controls the growth rate, and c defines the inflection point. Because these parameters are strongly coupled, poor initialization may produce an inaccurate sigmoid shape and an ill-conditioned Jacobian. LM, TRF, and AMLM were evaluated under good, moderate, and poor initialization at $\sigma \in \{0, 0.01, 0.05, 0.10\}$

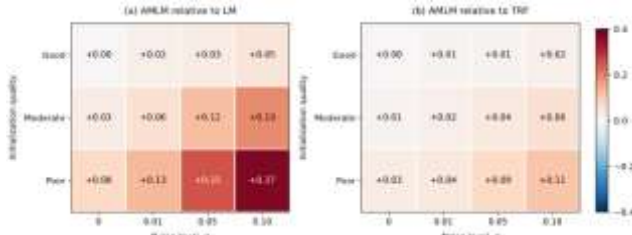


Figure 9 Difference in convergence success rate between AMLM and LM or TRF for the logistic model

Figure 9 shows that the algorithms have similar success rates under good initialization. The advantage of AMLM increases with initialization error and noise. Under poor initialization and $\sigma = 0.1$, AMLM exceeds LM by 0.37 and TRF by 0.11. Thus, AMLM provides the greatest robustness under the most difficult conditions, while TRF also considerably improves upon classical LM.

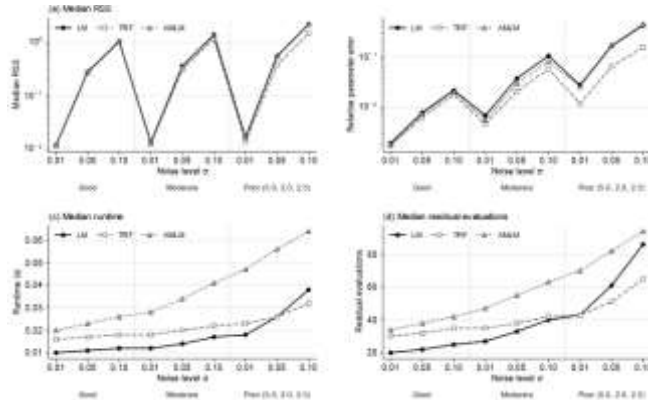


Figure 10 Accuracy and computational characteristics of LM, TRF, and AMLM for the logistic model

As shown in Figure 10, the median RSS values of the three methods are generally similar. Under moderate and poor initialization, AMLM produces the lowest relative parameter error, indicating more accurate recovery of the model parameters. However, this robustness is accompanied by the longest runtime and the largest number of residual evaluations. LM is computationally cheaper, whereas TRF provides an intermediate compromise between efficiency and robustness.

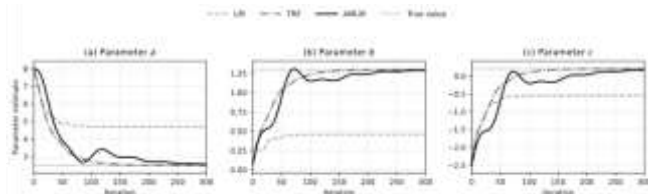


Figure 11 Parameter trajectories of LM, TRF, and AMLM for the logistic model under poor initialization

For the poor initial estimate

$$\theta_0 = [8.0, 0.1, -2.5]^T$$

LM converges to an incorrect parameter combination and does not recover the true sigmoid structure. TRF approaches the true parameters relatively quickly. AMLM follows a longer, nonmonotonic trajectory, particularly for a , but its oscillations gradually decay and all parameters ultimately converge to their true values.

This behavior results from the strong coupling among the asymptote, growth-rate, and inflection-point parameters. Under poor initialization, the standard LM step may be dominated by an inaccurate local approximation. The adaptive diagonal damping of AMLM regulates the parameter directions separately, thereby improving stability and enabling recovery of the correct nonlinear structure.

Overall, LM remains efficient under favorable initialization, TRF offers a good balance between cost and stability, and AMLM provides the highest robustness under poor initialization and elevated noise.

C. Application-Oriented Validation Using a Queueing-Inspired Model

To evaluate AMLM in an application-oriented setting, the following queueing-inspired nonlinear model is considered:

$$y(x; \theta) = \frac{a}{b-x} + c, \quad x < b$$

where x represents resource utilization, y denotes system delay, b characterizes effective service capacity, a controls the rate of delay growth, and c represents the baseline delay. Motivated by queueing theory, the model describes the rapid increase in response time as system load approaches capacity, a behavior commonly observed in cloud workloads involving disk I/O and network resources.

A critical numerical condition occurs when b approaches the input range. As $b - x \rightarrow 0$, both the model response and its derivatives increase sharply:

$$\frac{\partial y}{\partial a} = \frac{1}{b-x}, \quad \frac{\partial y}{\partial b} = -\frac{a}{(b-x)^2}, \quad \frac{\partial y}{\partial c} = 1$$

Consequently, the Jacobian becomes ill-conditioned and the parameter directions exhibit strongly unequal sensitivities.

As shown in Figure 12(a), AMLM accurately reproduces the nonlinear increase in the observed delay, including the high-curvature region. In contrast, LM fails to recover the underlying relationship and produces an almost constant fitted curve with a sharp local disturbance, as illustrated in Figure 12(b). This failure is caused by the extreme Jacobian sensitivity near the singular boundary, which results in unreliable parameter updates.

TRF also recovers the overall nonlinear trend, as shown in Figure 12(c), and performs substantially better than LM. However, AMLM provides the closest agreement with the observations in the considered experiment.

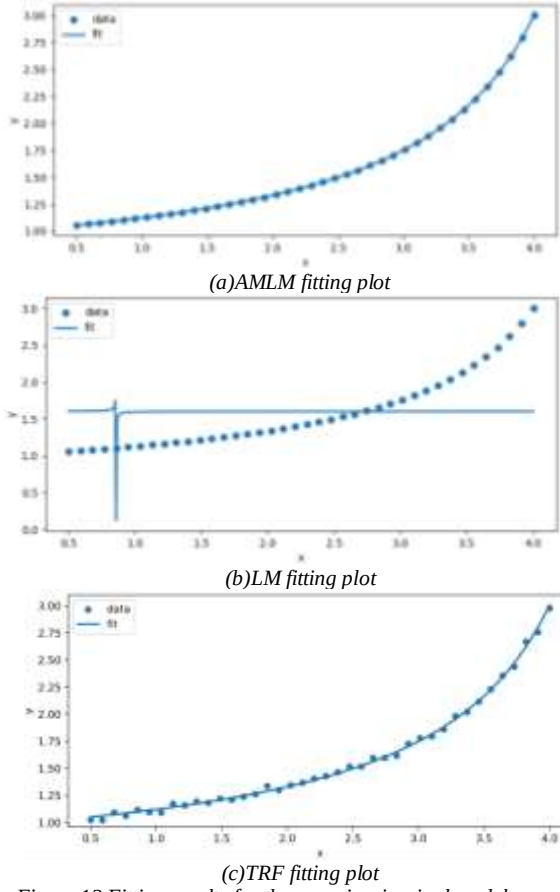


Figure 12 Fitting results for the queueing-inspired model near the singular regime

The stability of AMLM is attributed to its adaptive diagonal damping, which regulates the parameter directions according to their local sensitivities and prevents a highly sensitive direction from dominating the update. TRF achieves stability by restricting the overall step within a trust region, whereas the scalar damping used by classical LM is insufficient in this near-singular case.

These results demonstrate that AMLM improves numerical stability when the model operates near a critical region with strongly unbalanced parameter sensitivities. Both AMLM and TRF successfully recover the nonlinear queueing behavior, while LM fails under the considered experimental conditions.

D. Conditioning Analysis

To further explain the differences among the algorithms, the conditioning of the linearized system is analyzed. At iteration k , the classical LM step is obtained from

$$(J_k^T J_k + \lambda_k I) \Delta \theta_k = -J_k^T r_k$$

The numerical sensitivity of this system can be characterized by the condition number

$$\kappa(A_k) = \frac{\sigma_{\max}(A_k)}{\sigma_{\min}(A_k)}, \quad A_k = J_k^T J_k$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are the largest and smallest singular values. A large value of $\kappa(A_k)$ indicates that small errors in the residuals, Jacobian, or numerical calculations may cause large changes in the computed step.

Poor conditioning arises when the Jacobian columns have substantially different magnitudes or become nearly linearly dependent. This behavior was observed in the tested models. In the exponential saturation model, a and b become difficult to distinguish when b is close to zero. In

the power-law and logistic models, strong parameter coupling produces unequal directional sensitivities. The most severe case occurs in the queueing-inspired model near $b = x$, where the derivatives increase rapidly and the normal equations become numerically unstable.

Classical LM improves invertibility through scalar damping. If the eigenvalues of $J_k^T J_k$ are μ_i , the condition number of the damped matrix is

$$\kappa(J_k^T J_k + \lambda_k I) = \frac{\mu_{\max} + \lambda_k}{\mu_{\min} + \lambda_k}$$

Increasing λ_k therefore reduces the condition number. However, the same damping is applied to every parameter direction. A value sufficiently large to stabilize the most sensitive direction may excessively restrict progress in the remaining directions.

AMLM instead uses the system

$$(J_k^T J_k + \lambda_k D_k) \Delta \theta_k = -J_k^T r_k$$

where D_k is a positive diagonal matrix determined from the local parameter sensitivities. This direction-dependent regularization balances the scales of the parameter directions and limits updates associated with large or unreliable Jacobian components. If all diagonal elements of D_k are positive and bounded away from zero, the regularized matrix remains positive definite for $\lambda_k > 0$.

For analysis, the diagonally scaled matrix can be written as

$$\tilde{A}_k = D_k^{-1/2} J_k^T J_k D_k^{-1/2} + \lambda_k I$$

This representation shows that AMLM combines parameter scaling with scalar regularization in the transformed coordinates. When D_k adequately reflects the column scales of the Jacobian, the transformed system is better balanced and less sensitive to strongly unequal parameter magnitudes.

The experimental results are consistent with this interpretation. LM remains efficient when the Jacobian is well conditioned, but its reliability decreases under poor initialization, parameter degeneracy, and near-singular behavior. TRF limits the overall step and consequently improves stability. AMLM additionally regulates individual directions, which explains its improved robustness in the difficult exponential, power-law, logistic, and queueing-inspired experiments.

Nevertheless, diagonal damping does not guarantee a lower condition number in every iteration. If the initial estimate is extremely inaccurate, the local Jacobian may provide misleading sensitivity information, and the resulting diagonal scaling may be ineffective. Thus, AMLM reduces the practical effects of ill-conditioning in many difficult cases but does not eliminate dependence on the quality of the local approximation.

E. Convergence and Failure Analysis

The convergence behavior of LM, TRF, and AMLM depends strongly on the initial estimate, noise level, parameter coupling, and conditioning of the local Jacobian. Under favorable initialization, all three methods generally converge to comparable residual values. In this regime, classical LM is usually the most computationally efficient because its scalar damping introduces little additional overhead.

For AMLM, a trial step is computed from

$$(J_k^T J_k + \lambda_k D_k) \Delta \theta_k = -J_k^T r_k$$

where D_k is a positive diagonal damping matrix. The trial parameters are

$$\theta_k^{trial} = \theta_k + \Delta\theta_k$$

Step acceptance is determined using the ratio between the actual and predicted reduction:

$$\rho_k = \frac{F(\theta_k) - F(\theta_k + \Delta\theta_k)}{m_k(0) - m_k(\Delta\theta_k)}$$

where

$$F(\theta) = \frac{1}{2} \|r(\theta)\|_2^2$$

is the nonlinear least-squares objective and m_k is its local quadratic approximation. A step is accepted when it produces sufficient actual reduction. Successful steps allow the damping parameter to decrease, whereas unsuccessful steps increase damping and trigger a new trial step.

This mechanism explains the nonmonotonic parameter trajectories observed in several experiments. Temporary oscillations or large intermediate parameter values do not necessarily indicate final divergence. In the power-law and logistic experiments, AMLM initially exhibits pronounced oscillations but later returns to the attraction region of the correct solution. Therefore, convergence should be evaluated using the objective value, parameter error, feasibility, and final fitted structure rather than parameter monotonicity alone.

The experiments reveal four principal failure modes:

- 1) Convergence to a degenerate parameter region.

In the exponential saturation model, LM may converge to an incorrect linear-like solution when b is close to zero because

$$1 - e^{-bx} \approx bx$$

The residual may appear acceptable even though the nonlinear parameter structure is not recovered.

- 2) Incorrect curvature recovery.

In the power-law model, LM can produce a smooth fitted curve that fails to reproduce the true curvature, particularly for large values of x . This results from strong parameter coupling and unequal directional sensitivities.

- 3) Instability near a singular boundary.

For the queueing-inspired model

$$y(x; \theta) = \frac{a}{b - x} + c$$

the model and its Jacobian become highly sensitive when $b - x$ approaches zero. LM may then generate unstable steps or converge to a physically incorrect response. TRF and AMLM reduce this risk by restricting the step or applying direction-dependent regularization.

- 4) Failure under extremely poor initialization.

AMLM is not uniformly superior to LM. When the initial estimate lies far outside the attraction region of the desired solution, the Jacobian may provide misleading sensitivity information. Since D_k is constructed from this local information, AMLM may apply inappropriate directional scaling, leading to slow convergence, large oscillations, or convergence to an inaccurate parameter region.

An iteration is considered successfully converged when the objective reduction, parameter update, or gradient norm satisfies the prescribed tolerances:

$$\begin{aligned} |F_{k+1} - F_k| &\leq \varepsilon_F \\ \|\Delta\theta_k\|_2 &\leq \varepsilon_\theta (1 + \|\theta_k\|_2) \end{aligned}$$

or

$$\|J_k^T r_k\|_\infty \leq \varepsilon_g$$

Termination caused only by the maximum iteration count, numerical overflow, an invalid model value, or repeated rejection of trial steps is treated as failure. Moreover, a small residual alone is insufficient to establish successful parameter identification. The estimated parameters must also remain feasible and reproduce the correct nonlinear structure.

The experiments provide empirical evidence of improved robustness rather than a guarantee of global convergence. AMLM performs particularly well under moderate-to-poor initialization, elevated noise, unequal parameter sensitivities, and near-singular behavior. Nevertheless, extremely inaccurate initialization and misleading local Jacobian information remain important limitations. LM is preferable in well-conditioned cases, TRF offers a balanced compromise, and AMLM provides the strongest practical robustness in most of the difficult cases considered.

V. DISCUSSION

The proposed AMLM replaces scalar damping with curvature-aware diagonal damping, enabling parameter-specific regularization. Experiments on exponential saturation, power-law, logistic, and queueing-inspired models show that AMLM generally improves robustness under poor initialization, noise, unequal parameter sensitivities, and ill-conditioned Jacobians.

LM remains the most efficient method under good initialization and well-conditioned conditions. TRF provides a useful balance between stability and computational cost. AMLM is most advantageous in difficult cases where scalar damping cannot adequately control strongly unbalanced parameter directions. It successfully recovers nonlinear structures in several cases where LM converges to degenerate or inaccurate solutions. In the near-singular queueing-inspired experiment, both AMLM and TRF recover the nonlinear delay trend, whereas LM fails.

However, AMLM is not uniformly superior. It usually requires more residual evaluations and longer runtime. When initialization is extremely far from the desired parameter region, inaccurate local Jacobian information may also produce inappropriate diagonal scaling, causing oscillations, slow convergence, or inaccurate solutions. Therefore, AMLM improves practical robustness but does not guarantee global convergence.

Overall, LM is preferable for well-conditioned problems with reliable initial estimates, TRF offers a balanced alternative, and AMLM is more suitable when stability and correct parameter recovery are more important than computational cost.

The implementation and experimental scripts are available on GitHub:

https://github.com/shan5972/nonlinear/blob/main/AMLM_APP.py.

VI. CONCLUSION

This paper proposed an Adaptive Modified Levenberg–Marquardt (AMLM) algorithm with curvature-aware diagonal damping for nonlinear least-squares problems. Experiments conducted using three initialization levels, four

noise levels, and 100 runs per setting compared AMLM with classical LM and TRF.

The results show that AMLM generally provides the highest convergence robustness under poor initialization, elevated noise, unequal parameter sensitivities, and ill-conditioned conditions. It recovers the correct nonlinear structure in several cases where LM converges to inaccurate or degenerate solutions. In the near-singular queueing-inspired experiment, AMLM and TRF successfully recover the nonlinear response, whereas LM fails. Under favorable conditions, however, all three methods achieve similar accuracy, with LM offering the lowest computational cost and TRF providing a balance between efficiency and stability.

AMLM is not uniformly superior: it generally requires more evaluations and longer runtime, and extremely poor initialization may produce misleading diagonal scaling. Therefore, AMLM improves practical robustness but does not guarantee global convergence.

Future work will investigate parameter bounds, multi-start and global initialization strategies, alternative damping constructions, and applications to higher-dimensional and real-world problems. Overall, curvature-aware diagonal damping provides a useful extension of LM when reliable parameter recovery is more important than minimum computational cost.

Declaration on the Use of AI-Assisted Tools: Large language model (LLM) tools were used for language editing and stylistic refinement of the manuscript. All scientific content, experimental design and conclusions were developed and verified by the authors.

REFERENCES

- [1] D. M. Bates and D. G. Watts, *Nonlinear Regression Analysis and Its Applications*. New York, NY, USA: Wiley, 1988.
- [2] G. A. F. Seber and C. J. Wild, *Nonlinear Regression*. Hoboken, NJ, USA: Wiley, 2003.
- [3] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1999.
- [4] H. Khazaei, J. Mišić, and V. B. Mišić, "A fine-grained performance model of cloud computing centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 11, pp. 2138–2147, 2013, doi: 10.1109/TPDS.2012.280.
- [5] H. Khazaei, J. Mišić, and V. B. Mišić, "Performance analysis of cloud computing centers using queueing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 5, pp. 936–943, 2012, doi: 10.1109/TPDS.2011.199.
- [6] S. Wang and I. V. Nikiforov, "A Lyapunov-based dynamic scheduling algorithm for heterogeneous computing clusters," *Computing, Telecommunications and Control*, vol. 19, no. 1, pp. 65–79, 2026, doi: 10.18721/JCSTCS.19107.
- [7] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quart. Appl. Math.*, vol. 2, no. 2, pp. 164–168, 1944, doi: 10.1090/qam/10666.
- [8] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963, doi: 10.1137/0111030.

- [9] J. J. Moré, "The Levenberg–Marquardt algorithm: Implementation and theory," in *Numerical Analysis*, G. A. Watson, Ed. Berlin, Germany: Springer, 1978, pp. 105–116, doi: 10.1007/BFb0067700.
- [10] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia, PA, USA: SIAM, 1996.
- [11] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.
- [12] Å. Björck, *Numerical Methods for Least Squares Problems*. Philadelphia, PA, USA: SIAM, 1996.
- [13] G. H. Golub and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM J. Numer. Anal.*, vol. 10, no. 2, pp. 413–432, 1973, doi: 10.1137/0710036.
- [14] K. Madsen, H. B. Nielsen, and O. Tingleff, *Methods for Non-Linear Least Squares Problems*, 2nd ed. Lyngby, Denmark: Technical University of Denmark, 2004.
- [15] J. Fan and Y. Yuan, "On the quadratic convergence of the Levenberg–Marquardt method without nonsingularity assumption," *Computing*, vol. 74, pp. 23–39, 2005, doi: 10.1007/s00607-004-0083-1.
- [16] N. Yamashita and M. Fukushima, "On the rate of convergence of the Levenberg–Marquardt method," in *Topics in Numerical Analysis*. Vienna, Austria: Springer, 2001, pp. 239–249.
- [17] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*. Philadelphia, PA, USA: SIAM, 2000.
- [18] T. Steihaug, "The conjugate gradient method and trust regions in large scale optimization," *SIAM J. Numer. Anal.*, vol. 20, no. 3, pp. 626–637, 1983, doi: 10.1137/0720042.
- [19] T. F. Coleman and Y. Li, "An interior trust region approach for nonlinear minimization subject to bounds," *SIAM J. Optim.*, vol. 6, no. 2, pp. 418–445, 1996, doi: 10.1137/0806023.
- [20] M. A. Branch, T. F. Coleman, and Y. Li, "A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems," *SIAM J. Sci. Comput.*, vol. 21, no. 1, pp. 1–23, 1999, doi: 10.1137/S1064827595289108.
- [21] H. Schwetlick and V. Tiller, "Nonstandard scaling matrices for trust region Gauss–Newton methods," *SIAM J. Sci. Comput.*, vol. 21, no. 1, pp. 348–362, 1999, doi: 10.1137/S1064827597327739.
- [22] M. K. Transtrum, B. B. Machta, and J. P. Sethna, "Geometry of nonlinear least squares with applications to sloppy models and optimization," *Phys. Rev. E*, vol. 83, Art. no. 036701, 2011, doi: 10.1103/PhysRevE.83.036701.
- [23] M. K. Transtrum and J. P. Sethna, "Improvements to the Levenberg–Marquardt algorithm for nonlinear least-squares minimization," *arXiv:1201.5885*, 2012, doi: 10.48550/arXiv.1201.5885.
- [24] L. Kleinrock, *Queueing Systems, Volume I: Theory*. New York, NY, USA: Wiley, 1975.

Wang Shan, is currently pursuing a Ph.D. in Cloud Computing at Institute of Computer Science and Cybersecurity, Peter the Great St. Petersburg Polytechnic University, Russia. Her research interests include resource scheduling, cloud-edge collaboration, and intelligent optimization in distributed systems. And her recent work focuses on applying Lyapunov optimization theory to real-time systems and heterogeneous cluster management. (email:wangshan@mail.ru, <https://orcid.org/0000-0001-8591-9080>)

Igor Valerievich Nikiforov, PhD, Associate Professor at the Higher School of Software Engineering at Peter the Great St. Petersburg Polytechnic University. In 2011, he graduated from St. Petersburg State Polytechnic University with a degree in «Computer Science and Automated Systems Software». In 2014 he defended his dissertation for the degree of candidate of technical sciences (PhD) on the speciality 05.13.11 - «Mathematical and software support for computers, complexes and computer networks». He is the author of more than 100 scientific publications. Research interests – software engineering, simulation modeling, big data analytics, distributed computing. (email: nikiforov_iv@spbstu.ru, <https://orcid.org/0000-0003-0198-1886>)