Stochastic Levenberg-Marquardt for Solving Optimization Problems on Hardware Accelerators

Y. Hong*, E. Bergou*, N. Doucet†, H. Zhang†, J. Cranney†, H. Ltaief*, D. Gratadour†, F. Rigaut†, and D. Keyes*

*Extreme Computing Research Center
Division of Computer, Electrical, and Mathematical Sciences and Engineering
King Abdullah University of Science and Technology, KSA
Email: {Yuxi.Hong, Houcine.Bergou, Hatem.Ltaief, David.Keyes}@kaust.edu.sa

†Research School of Astronomy & Astrophysics
College of Science
Australian National University, Australia
Email: {Nicolas.Doucet, Hao.Zhang1, Jesse.Cranney, Damien.Gratadour, Francois.Rigaut}@anu.edu.au

***Abstract*—We present a new Stochastic Levenberg-Marquardt (SLM) algorithm for efficiently solving large-scale nonlinear least-squares optimization problems. The SLM method incorporates stochasticity into the traditional Levenberg-Marquardt (LM) method. While the traditional LM operates on the full objective function, SLM randomly evaluates part of the objective to compute the corresponding derivatives and function values. Hence, SLM reduces the algorithmic complexity per iteration and speeds up the overall time to solution, while maintaining the numerical robustness of second-order methods. We assess the SLM method on standard datasets from `LIBSVM`, as well as on a large-scale optimization problem found in ground-based astronomy applications, and in particular adaptive optics systems of the next generation of instruments for the European Very Large and Extremely Large Telescopes. We implement SLM and deploy it on a shared-memory system equipped with multiple GPU hardware accelerators. We demonstrate the performance superiority of the SLM method over not only the traditional LM algorithm but also the state-of-the-art first-order methods. SLM finishes optimization process in less than 1 second on large datasets from the adaptive optics application, where LM and other methods require more than a few minutes. This enables to identify of the system parameters (e.g., atmospheric turbulence and wind speed) and to capture their evolution required during a night of observations with a close to real-time throughput.**

***Index Terms*—Optimization problems, Stochastic Levenberg-Marquardt, GPU Computing, Adaptive optics, Computational astronomy, Real-time processing;**

I. Introduction

Large-scale optimization is at the core of machine learning and deep learning. It is one of the most challenging missions in the machine learning community. There are many sub-sampling methods that can be combined with traditional optimization algorithm, e.g., gradient descent, to get a new variant of a classical method, namely stochastic gradient descent or minibatch SGD. An analysis of sub-sampling method for Newton's method is given in [1]. However, somewhat surprisingly, we have found in the literature no stochastic version of the classical Levenberg-Marquardt (LM) method.

In this paper, we consider a new Stochastic Levenberg-Marquardt algorithm that exploits randomization to shrink the size of the active workingset, i.e., a model that uses at a time only a random subset of the sum of the original function (minibatch). The gradient vector and Hessian matrix that are used in each iteration are calculated from this subset. We test its validity in an astronomy application, namely the identification of atmospheric turbulence parameters [2] for use in Adaptive Optics (AO) controllers [3], applicable to the new generation of Extremely Large Telescopes (ELT).

Our goal of low-power real-time performance for adaptive optics in extremely large telescopes is met through scaling to multiple GPUs a minibatch version of the state-of-the-art algorithm in the application domain, LM. This is an elementary example of high practical value of an active theme in computing: the convergence of machine learning and high performance computing that is driven by expanding problem size. The architectural trend of packing more compute power in devices whose memory capacity and memory bandwidth per processing unit cannot keep up puts an algorithmic premium on shrinking the workingsets, so that they reside higher in the memory hierarchy, closer to the processing elements, thus increasing the arithmetic intensity and shrinking the costs of time and energy of moving data.

The improved performance achieved by using the novel Stochastic Levenberg-Marquardt (SLM) method provides a means for astronomers to overcome the prohibitive dimensionality found when identifying turbulence parameters at the scale of ELT. The traditional LM method has proven useful in existing AO schemes (as is discussed in Section V), however its relevance to the controller is limited due to its burdensome execution time relative to the temporal evolution of the parameters identified. By adopting the SLM method on dedicated multi-GPU hardware, a speedup of approximately 50-fold is achieved, which brings the time-scale of the identification step from the order of minutes down to seconds.

We compare the novel SLM method to traditional LM, as well as state-of-the-art first-order methods in the astronomy application. These comparisons include time-to-solution analysis of each method, as well as testing the output of each method in end-to-end numerical AO simulation software to verify the impact of the choice of optimization scheme.

We summarize our contribution as follows:

- We present Stochastic LM method and the convergence theory.
- We show a significant acceleration in astronomy application which is close to realtime throughput.
- We compare LM/SLM with first-order methods in an astronomy application and demonstrate that SLM achieves sufficient quality and speed.

The remainder of the paper is organized as follows: Section II states the problem, presents the state-of-the-art, and details related work. Section III presents our convergence analysis for SLM with fixed regularization. Section IV introduces software that deploys SLM on accelerator hardware. Section V introduces the application of SLM in astronomy. We investigate the performance and acceleration of SLM with traditional LM method and other state-of-the-art first-order methods in this application. Section VI describes the datasets we are using to validate the SLM algorithm. Section VII analyzes the acceleration of SLM and its comparison with other optimizers, as well as the experimental results for the astronomy application. Finally, we provide our conclusions in Section VIII.

II. Problem Statement and Related Work

Problem Statement

We consider the following nonlinear least squares problem

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{2}\|F(x)\|^2 = \frac{1}{2}\sum_{i=1}^{n} F_i(x)^2, \tag{1}$$

where $f:\ \mathbb{R}^d \to \mathbb{R}$ and $F_i:\ \mathbb{R}^d \to \mathbb{R}$, for $i = 1,\dots,n$ are assumed twice continuously differentiable. Here and in the rest of the paper, $\|\cdot\|$ denotes the ℓ_2 norm. I denotes the Identity matrix of dimension n.

A. State-Of-The-Art on Levenberg-Marquardt methods

The Gauss-Newton method and its globally convergent variants are often the methods of choice to tackle nonlinear least-squares problems. The Gauss-Newton method is an iterative procedure for solving problem (1) by iteratively solving a linearized least squares subproblem of the following form:

$$\min_{s\in\mathbb{R}^d} \frac{1}{2}\|F(x) + Js\|^2, \tag{2}$$

where J is the Jacobian of the function F x. This subproblem has a unique local solution if J has full column rank, in which case the solution of the subproblem is a descent direction for the function f. Subproblem (2) is not well-posed in the case of rank deficiency of the Jacobian function J. Furthermore, the Gauss-Newton method may not be globally convergent, in the sense that its solution depends on the starting point for the minimization. The LM method [4, 5, 6] was developed to deal with the rank deficiency of J and to provide a globalization strategy for Gauss-Newton (in the sense of independence of the starting point of the minimization). At each iteration it adds a regularization term to Subproblem (2). Subproblem (2) becomes then

$$\min_{s\in\mathbb{R}^d} \frac{1}{2}\|F(x) + Js\|^2 + \frac{1}{2}\mu^2\|s\|^2, \tag{3}$$

where $\mu > 0$ is an appropriately chosen regularization parameter. Several strategies have been developed to update this parameter over the optimization process. This regularization parameter is updated at every iteration and indirectly controls the size of the step, making Gauss–Newton globally convergent (i.e., convergent to a stationary point independently of the starting point). Indeed, this added regularization seeks to determine when the Gauss-Newton step is applicable, in which case the regularization parameter is set to small value (or even to zero in some LM variants) or when it should be large and that the resulting step is small and thus making a slower but safer gradient step.

Variants of LM algorithm have been studied also when the Gauss-Newton subproblem model is replaced by a random model that is only accurate with a given probability [7, 8]. They have also been applied to problems where the objective value is subject to noise [9, 10]. In this paper, we consider a stochastic LM variant that handles particular random models, i.e., models which uses only a random subset of the sum of the classical model. To the best of our knowledge this is the first paper to consider this variant of LM.

B. Levenberg-Marquardt method in Astronomy Application

Tomographic AO requires an estimation of the atmospheric conditions, which can be represented through a limited number of parameters, assuming a multi-layer turbulence model. The Learn and Apply method [2] employed herein and tested on real telescope [11], makes use of the LM method to identify the necessary parameters.

Depending on the AO system configuration, the matrices involved can be quite large (up to 100K $\times$ 100K). Furthermore the number of parameters to be estimated is proportional to the number of turbulent layers (up to 40) in the atmosphere model. Problems of such dimension typically require on the order of a few minutes [12] which can hinder their applicability to the near-real-time requirements of the application.

III. Stochastic Levenberg-Marquardt with Fixed Regularization

We present here a convergence analysis for the stochastic LM Method. We consider a particular version of SLM in which the regularization parameter is kept fix over the iterations. Algorithm 1 describes the variant considered, which can be shown to be globally convergent with the classical complexity bound known in the literature for stochastic algorithms like SGD and its variants.

We use the following notation in the rest of the paper $f^k = f(x^k)$, $F^k = F(x^k)$, $J^k = S^k \nabla F(x^k)$, $g^k = \nabla f(x^k) = \nabla F(x^k)^T F^k$, $\tilde{g}^k = \nabla F(x^k)^T S^k F^k$, where $S^k \in \mathbb{R}^{n,n}$ is a diagonal matrix with ones and zeros randomly distributed over the diagonal scaled by the number of non zero elements divided by n. We note that the scaling used in the definition of the stochastic matrix S^k makes its expectation equals to Identity matrix I.

Remark 1. Our analysis can be easily generalized to cover more distributions from which to choose S^k. However, for the sake of simplicity and ease of readability of our proofs we limit ourselves to the above-mentioned distribution.

Algorithm 1 Stochastic Levenberg-Marquardt Algorithm with fixed regularization

Initialization: : choose x^0

1: **for** $k = 1, 2, ..., K$ **do**

2:
$$x^{k+1} = x^k - (J^{k\top} J^k + \mu I)^{-1} \tilde{g}^k \tag{4}$$

3: **end for**

We first state the general assumptions (several of which are classical ones) that we use for the convergence analysis of Algorithm 1.

Assumption 1. *(Lower bound) The function f is lower bounded; that is, there exists an $f^\star \in \mathbb{R}$ such that $f(x) \geq f^\star$, for all x.*

Assumption 2. *($\mathcal{L}$-smoothness) The function f is $\mathcal{L}$ smooth if its gradient is $\mathcal{L}$-Lipschitz continuous, that is, for all $x, y \in \mathbb{R}^d$,*

$$f(x) \leq f(y) + \nabla f(y)^\top (x - y) + \frac{\mathcal{L}}{2} \|x - y\|^2.$$

Assumption 3. *(Boundness of stochastic gradient) The stochastic gradient is bounded by $G > 0$, that is,*

$$\mathbb{E}\left[\|\tilde{g}^k\|\right] \leq G.$$

Note that the latter inequality implies (using Jensen's inequality)

$$\mathbb{E}\left[\|\tilde{g}^k\|\right]^2 \leq G^2.$$

Assumption 4. *The function F Jacobian is bounded by $\kappa_J > 0$, that is,*

$$\mathbb{E}\left[\|\nabla F(x^k)\|\right] \leq \kappa_J.$$

The latter Assumption implies

$$\mathbb{E}\left[\|J^k\|\right] \leq \kappa_J, \text{ independently from } S^k.$$

Lemma 1. *(Unbiasedness of stochastic gradient) The stochastic gradient is unbiased. That is,*

$$\mathbb{E}\left[\tilde{g}^k | x^k\right] = g^k = \nabla f(x^k).$$

Proof. Direct from the fact that $\mathbb{E}\left[S^k\right] = I$. □

Lemma 2.

$$\left(J^{k\top} J^k + \mu I\right)^{-1} = \frac{I}{\mu} - \frac{1}{\mu^2} J^{k\top} \left(\frac{1}{\mu} J^k J^{k\top} + I\right)^{-1} J^k$$

Proof. Direct by application of the Sherman-Morrison-Woodbury formula. □

Lemma 3. *Let Assumptions 4 and 3 hold, then*

$$\mathbb{E}\left[\left\|\nabla f^k \frac{1}{\mu^2} J^{k\top} \left(\frac{1}{\mu} J^k J^{k\top} + I\right)^{-1} J^k \tilde{g}^k\right\| | x^k\right] \leq \kappa_J^2 G^2.$$

Proof. see Appendix B. □

Lemma 4. *Let Assumption 3 hold, then*

$$\mathbb{E}\left[\left\|(J^{k\top} J^k + \mu I)^{-1} \tilde{g}^k\right\|^2 | x^k\right] \leq \frac{G^2}{\mu^2}$$

Proof.

$$\begin{aligned}\left\|(J^{k\top} J^k + \mu I)^{-1} \tilde{g}^k\right\|^2 &\leq \left\|(J^{k\top} J^k + \mu I)^{-1}\right\|^2 \left\|\tilde{g}^k\right\|^2 \\ &\leq \frac{\left\|\tilde{g}^k\right\|^2}{\mu^2}.\end{aligned}$$

Now, by taking the conditional expectation and using Assumption 3 we get the desired result. □

Theorem 1. *Let Assumptions 1, 2, 3 and 4 hold. Let $K > 0$, $\mu_0 > 0$ and $\mu = \mu_0 \sqrt{K+1}$, then*

$$\frac{\sum_{k=0}^K \mathbb{E}\|\nabla f^k\|^2}{K+1} \leq \frac{C}{\sqrt{K+1}},$$

where

$$C := \mu_0 \left(f^0 - f^*\right) + \frac{2\kappa_J^2 G^2 + \mathcal{L} G^2}{2\mu_0}.$$

Proof. see Appendix A. □

Corollary 1. *Let Assumptions 1, 2, 3 and 4 hold. Let $K > 0$, $\mu_0 > 0$ and $\mu = \mu_0 \sqrt{K+1}$, then*

$$\min_{k \in \{0,...,K\}} \mathbb{E}[\|\nabla f^k\|^2] \leq \mathcal{O}\left(\frac{1}{\sqrt{K+1}}\right).$$

Proof. Direct from Theorem 1. □

This corollary states that the minimum over iterations of the square of the function gradient norm, in expectation, is of the order of $\mathcal{O}\left(\frac{1}{\sqrt{K+1}}\right)$. This is the classical complexity bound known in the literature for the stochastic methods, like SGD and its variants. One can also derive the convergence of the algorithm to a stationary point by increasing the regularization parameter. We note that our convergence analysis can be extended to convex and strongly convex cases. Due to space limitations we skip this analysis.

IV. Implementation Details

We introduce the software stack and programming model used for the implementation of SLM, with CUDA kernels for the stochastic Hessian and function values. The proposed algorithm is implemented in the Supervisor module with HIgh Performance Software (SHIPS), a software platform dedicated to the soft real-time component of tomographic AO. The most challenging component of the astronomy application is the huge number of data points required to compute during each epoch. The most time consuming steps per iteration are computation of $f(x)$, the gradient vector, and the Hessian matrix, as listed in lines 5 and 8, respectively in VI-C, in the Algo 2. The implementation in [12] uses CUDA programming and multiple GPUs to accelerate the process. We reimplement two kernels with the ability to handle the case of stochasticity. Our kernels use a random index array to point CUDA threads to the data we select in the current iteration. In order to avoid sampling from a large array around 200K $\times$ 200K, we form an index groups over the total set of elements. Each index group contains 1024 consecutive indices. We also implemented a multiple GPU version for acceleration.

V. Astronomy Application

A. Challenges in Ground-Based Astronomy

When the aperture of a telescope is sufficiently large, its image quality becomes limited by the astronomical "seeing condition" as a result of atmospheric turbulence (i.e., *seeing limited*) as opposed to being limited by the resolving power of the telescope aperture (i.e., *diffraction limited*). A wide range of AO techniques can be applied to mitigate this effect by controlling a deformable mirror in real time to correct for observed optical wavefront aberrations.

The next generation of AO systems, which will be supplied to both state-of-the-art 10m class telescopes (such as the European Very Large Telescope – VLT) and future giant telescopes such as the 40m class ELT, boast high density Wavefront Sensors (WFS) and Deformable Mirrors (DM) in a high complexity system design together with a strong requirement on the system final performance. This combination puts severe constraints on computational capabilities to be deployed on the telescope.

One of the most important steps in AO operation is estimating the atmospheric turbulence profile, and it has fundamental benefits if it can be done in real time. Given the rapid evolving nature of the atmospheric turbulence, this estimation should be performed in seconds, and this can be achieved with the use of the SLM method.

B. Atmospheric Turbulence

Though continuously distributed over altitude, the atmospheric turbulence can be modelled by n_L vertically discrete thin layers. The input dataset used in this experiment is the measurement covariance matrix calculated from recent WFS data in the AO system. The dimensions of this matrix become larger as the AO system becomes more complicated.

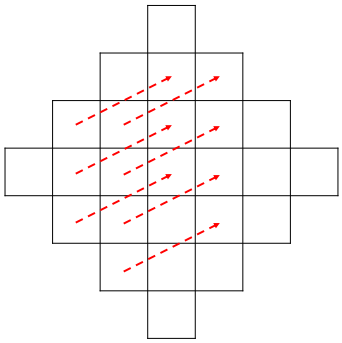


Figure 1: Sub-aperture arrangement for a SHWFS with 7 sub-apertures in each dimension.

As an illustration, a typical Shack-Hartmann WFS consists of a grid of sub-apertures that divides the light from the telescope pupil into sub-regions, every sub-aperture measures the local average wavefront slope in one sub-region in (x, y) direction. For example, an ELT scale AO instruments with 6 WFSs and each of those with 80 sub-apertures in each dimension will have a covariance matrix with dimensions of approximately 100K $\times$ 100K.

Analytically, the covariance matrix can be regarded as the sum of the variances of wavefront distortions caused by each layer of turbulence, the statistical properties of which can be derived from the von Karman model [13] based on several assumptions [14, 15].

The $(m, n)^{\text{th}}$ entry of covariance matrix is then given by

$$C_{s,mn} = \sum_{i=1}^{n_L} F\{C_{n,i}^2, v_{w,i}, \theta_{w,i}, r_i(m, n)\}, \tag{5}$$

where $C_{s,mn}$ is the slope space covariance value; i is the index of turbulence layer which counts up to n_L; $C_{n,i}^2$ is the relative turbulence strength of ith layer; $v_{w,i}$, $\theta_{w,i}$ are the wind speed and wind direction of ith layer respectively; and $r_i(m, n)$ is the distance between the m^{th} and n^{th} sub-aperture projected on the i^{th} layer.

In AO systems, $r_i(m, n)$ can be pre-calculated from the system configuration (i.e., number of subapertures and pointing directions of the WFS which should be fixed for a given system or observation), and the main target of atmosphere profiling is to estimate the C_n^2, v_w and θ_w for each layer. In a simple case when only C_n^2 is considered, the function $F\{C_{n,i}^2\}$ is linear which leads to a linear least-squares solution, however the same function becomes nonlinear when also estimating the wind profile.

Under most circumstances, the input covariance has redundancy due to the geometrical distribution of sub-apertures as shown in Fig 1. All sub-aperture pairs marked with red arrows have the same spatial distance thus same value in covariance, but appear in different entries in the covariance matrix. Such redundancy makes the SLM method appealing.

C. Optimizers in Turbulence Identification

In this paper, we consider the case of the MAVIS instrument for the VLT [16]. MAVIS will be equipped with multiple high-

order (40 $\times$ 40 sub-apertures) WFSs in order to achieve a high level of atmospheric turbulence compensation in order to perform nearly diffraction limited observations in the visible on the VLT. The total number of entries in the covariance matrix (i.e., n from Eq 1) is approximately 4×10^8. This falls in the range of *big data* optimization. We consider a 10-layer turbulence profile (i.e., $n_L = 10$ from Eq 5). Beyond the size of the workload itself, the ability to update the turbulence parameters at a regular rate (every second or less) is critical to ensure the stability of the AO performance and be able to work in the visible range. Vidal et al [2] propose and demonstrated the Levenberg-Marquardt method for this process, but the drawback of the traditional Levenberg-Marquardt method is that it can not utilize the stochasticity to randomly sample a subset of the dataset to accelerate the optimization process.

Considering the data redundancy mentioned before, we propose that stochastic algorithms should behave comparably to the traditional LM method. In order to verify this, we compare three state-of-the-art first-order algorithms and the SLM method with traditional LM method. The three first-order methods are stochastic gradient descent (SGD), Adam, and SGD with momentum (Momentum).

The LM implementation in Chapter 3.2 of [17] can be interpreted as a variant of the LM method with an adaptive μ. Similarly, we extend Algorithm 1 to handle an adaptive μ, resulting in the SLM variant proposed in Algorithm 2.

Algorithm 2 SLM Method in astronomy application with adaptive μ

Initialization: : $k = 0$; $x = x_0$; $kmax = 50$; $\mu = \tau \max\{a_{i,i}\}$ $\epsilon_1 = \epsilon_2 = 1 \times 10^{-6}$; found = $\|g\|_\infty \leqslant \epsilon_1$;
while not found and $k <$ kmax **do**
  $k := k + 1$;
  solve $((J^k)^\top J^k + \mu I)h_{lm} = -\tilde{g}^k$;
  found = $\|h_{lm}\| \leqslant \epsilon_2(\|x\| + \epsilon_2)$; {//no progress}
  $\rho = \frac{f(x) - f(x_{new})}{\frac{1}{2}h_{lm}^\top(\mu h_{lm} - \tilde{g}^k)}$
  if $\rho > 0$ **then**
    $x_{new} = x + h_{lm}$;
    update J^k and $\tilde{g}^k$
    found = $\|g\|_\infty \leqslant \epsilon_1$; {//small gradient norm}
    $\mu = \mu \max(1/3, 1 - (2\rho - 1)^3)$;
    $\nu = 2$;
  else
    keep using same J^k and $\tilde{g}^k$ in next round.
    $\mu = \mu\nu$; $\nu = 2\nu$;
  end if
end while
return x

VI. Datasets Description

We use three datasets to validate the performance of SLM: a simple exponential function, a standard machine learning dataset from the LIBSVM library, and an atmosphere model function from an astronomy application. The first two datasets are convex problems of relatively small size, to show the behavior of SLM in well understood settings. The atmospheric turbulence profiling dataset is derived from a proposed AO instrument for the 8-m Very Large Telescope (VLT), which is non-convex and larger in size.

A. Synthetic function Dataset

The first dataset is a simple exponential function, where $f(x) = e^{ax^2+bx+c}, a = 1, b = 2, c = 3$. We generate data by choosing $x_i \in \frac{t_i}{1000}, t_i = 1, 2, ..., 1000$, then we generate function values using x_i. The initial values are chosen uniformly from $[0, 1]$. There are total 1000 data points in this dataset.

B. Madelon Dataset

The second dataset is Madelon from LIBSVM library. This is a standard machine learning regression problem dataset. We apply logistic regression on this problem with the objective function given by $f_i(x) = \log\left(1 + \exp(-b_i a_i^\top x)\right)$, where $a_1, ..., a_n \in \mathbb{R}^d$, $b_1, ..., b_n \in \mathbb{R}$. There are 2000 samples in the dataset and $d = 500$.

C. Astronomy Dataset

The dataset used in this experiment is a covariance matrix with its size at approximately 20K $\times$ 20K elements. Every entry in this matrix is considered as one input data point for the algorithm. As mentioned in Section V, the matrix is generated from a 10-layer atmosphere profile averaged from experimental data using a pseudo-analytical model. Readers may refer to [2, 13] for more details with respect to the system model while here we focus on its most salient features. The layers are vertically distributed within $[0, 14000]$m altitude range and the parameters to be estimated are the relative intensity, wind speed and wind direction for each turbulence layer. The system is typically nonconvex, and the total number of parameters to identify in this case is $d = 30$.

VII. Experimental Results

A. System and Environment Settings

We test on three NVIDIA DGX systems, each with two sockets of 20-core Intel(R) Xeon(R) Broadwell CPU E5-2698 v4 @ 2.20GHz, that are connected to eight GPUs through PCIe (10GB/s). They differ in the GPU they host. The first one has eight Nvidia P100s interconnected with 20GB/s NVLink. The second one has eight Nvidia V100s, We also test on an NVIDIA DGX system node equipped with 8 A100 GPUs. We use double precision for the whole computation. A single P100 GPU has a theoretical peak performance of 5.3 TFlop/s in double precision, whereas a V100 GPU performance goes as high as 7.8 TFlop/s, and an A100 peaks at 19.5 Tflop/s.

B. Experiments for synthetic function and Madelon

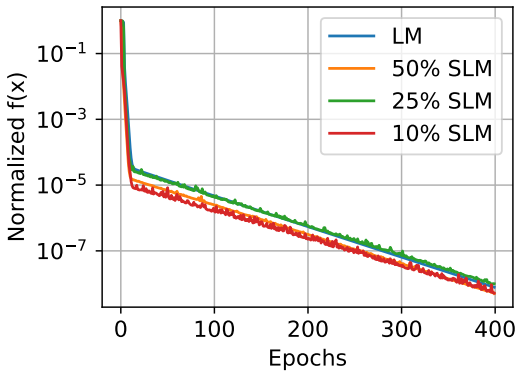
In this section, we show the results of three experiments. In experiments for synthetic function and Madelon, we use normalized function value (or Chi-Square value χ^2) and gradient norm as criteria to see how SLM converges in relation to the standard LM method. We sample points from the

datasets using a uniform distribution. We examine the SLM algorithm in different scenarios for fraction of data examined. We plot only the criteria by epoch since these two datasets are small. We let the standard algorithms run 400 epochs and for a fair comparison, e.g., for SLM using 50% data, we additional 800 iterations. We compare Algorithm 1 with different data fraction (50%, 25%, 10% of the data) and the standard Levenberg-Marquardt method. All methods are initialized from a random vector. We compare the function value $f(x)$ and the gradient norm $\|g\|_\infty$ of different methods. We report the convergence rate of different algorithms as a function of epoch.

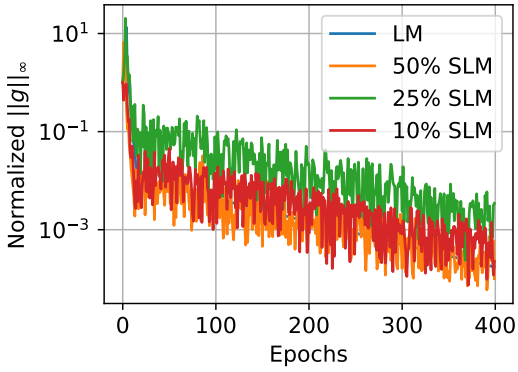
In synthetic function dataset, the results for function value in Fig 2a and gradient norm in Fig 2b show that all the methods have decreasing norms. The SLM methods have oscillating gradient norms within a decreasing trend. SLM using 10% of the data has a larger gradient norm than LM method by two orders of magnitude; however, all are decreasing.

In the Madelon dataset, the SLM methods also shows the same decreasing rate as LM methods, as is shown in Fig 3a. The normalized gradient norm results in Fig 3b shows SLM methods converge to the same level as LM methods. The behavior of SLM methods over different levels of data fraction are relatively stable, as in the synthetic function dataset.

We have not yet investigated the relationship between the percentage of data (or batch size) used and performance of the algorithm, and we cannot guarantee that the SLM method is stable in any choice of batch size. We have observed SLM to fail if we choose a very small batch size.



(a)



(b)

Figure 2: (a) $f(x)$ of synthetic function. (b) $\|g\|_\infty$ of synthetic function.

C. Experiments for Astronomy dataset

We here compare the SLM method with the classical LM method and state-of-the-art of first-order methods (SGD, Adam [18], Momentum). For the learning rate of first-order methods, we test different learning rates and choose a value close to the largest stable learning rate. Here we use $lr = 0.1$. For other parameters in the Adam and Momentum methods, we follow the classical methods of tuning for each. We use the function value and gradient norm as criteria, and we record the actual time to solution for all experiments. The reason is that the astronomy dataset is large enough to usefully measure the expected acceleration of the SLM method. Here the emphasis is on realized performance rather than convergence rate.

We are also interested in the impact of data fraction and initial conditions in the real application. Second order methods are well-known to fall into local minima. When operating on-sky, the real atmosphere parameters are unknown so in order to validate the SLM method we should investigate the choice of initial conditions here.

1) SR Results and Science Analysis: One of the most critical performance metrics of an AO system is the Strehl Ratio (SR), which can be considered as the performance of the optical system compared to one with no optical aberrations. As such a SR of 1.0 represents a perfect imaging system, and a low SR represents a worse imaging system.

In the optical wavelengths relevant to the MAVIS system, a SR of around 0.2 is considered to be impressive, and a SR less than 0.1 is considered to be underperforming. Tables I and II show that by using the SLM method to identify the atmospheric turbulence parameters, we are always able to achieve an impressive SR in simulation. The rapid plunge of the blue curves in Fig 4a and Fig 4b show that the timing requirement of approximately 1 second is also satisfied in SLM on the available hardware (see zoom insets). The remaining curves show that all of the first-order methods fail the timing requirement, although interestingly the Adam method is able to reach an impressive SR when the data fraction is sufficiently low.

Table I: SR results for Random initialization with 4 data fraction (50%, 10%, 1%, 0.1%) shown as data fraction (DF). LM's SR value is 0.203.

	SLM				SGD			
DF	0.5	0.1	0.01	0.001	0.5	0.1	0.01	0.001
SR	.203	.199	.204	.206	.005	.003	.003	.002
	Adam				Momentum			
DF	0.5	0.1	0.01	0.001	0.5	0.1	0.01	0.001
SR	.005	.064	.203	.199	.006	.033	.054	.009

2) Convergence Results: We analyze the convergence of different methods on a single P100 GPU node in terms of time to solution. We measure the performance for both zero

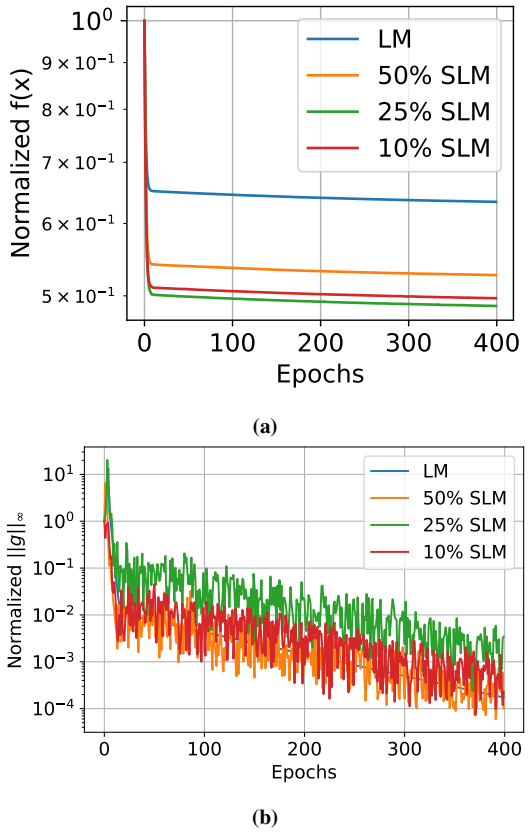


Figure 3: (a) $f(x)$ of madelon dataset. (b) $\|g\|_\infty$ of madelon dataset.

Table II: SR results for Zero initialization with 4 data fraction (50%, 10%, 1%, 0.1%), shown as data fraction (DF). LM's SR value is 0.214.

	SLM				SGD			
DF	0.5	0.1	0.01	0.001	0.5	0.1	0.01	0.001
SR	.213	.213	.214	.214	.004	.004	.004	.002
	Adam				Momentum			
DF	0.5	0.1	0.01	0.001	0.5	0.1	0.01	0.001
SR	.008	.114	.210	.211	.004	.030	.055	.007

initialization and random initialization when we use 0.001 data per iteration. All the methods start from the same points. For zero initialization, we put all C_n^2 fraction on the 1^{st} layer and all other parameters to be 0. For random initialization we generate a random number uniformly for all parameters. Considering the real application need, we set a time out (300 s). For first-order method, we set an alternative cap of the maximum number of iterations at 2000. For second-order method, we set this number to be 200. All methods stop once they meet either of the following criteria: 1. $h_{lm} \leq \epsilon(\|x\|+\epsilon)$ 2. $\|\tilde{g}^k\| \leq \epsilon$ where $\epsilon = 1e^{-8}$.

We observe that when we use 0.5, 0.1, 0.01 of the data, most of the methods time out and we do not achieve real-time results. On the other hand, the convergence behavior is similar to the case we highlight next. We focus on the case with the smallest dataset — a data fraction of 0.001.

When the data fraction equals 0.001, the function value vs timing for zero initialization is shown in Fig 4a and gradnorm vs timing for zero initialization is shown in Fig 4b. SLM finishes the optimization in about 1.25 seconds. SLM achieves a dramatic drop of $f(x)$ and gradient norm while other methods run more than 1 minute without acceptable results. LM can achieve the same convergence but it needs a lot time (exceeds 300s). Among first-order methods, Adam achieves best performance but it stills needs more than 60 seconds.

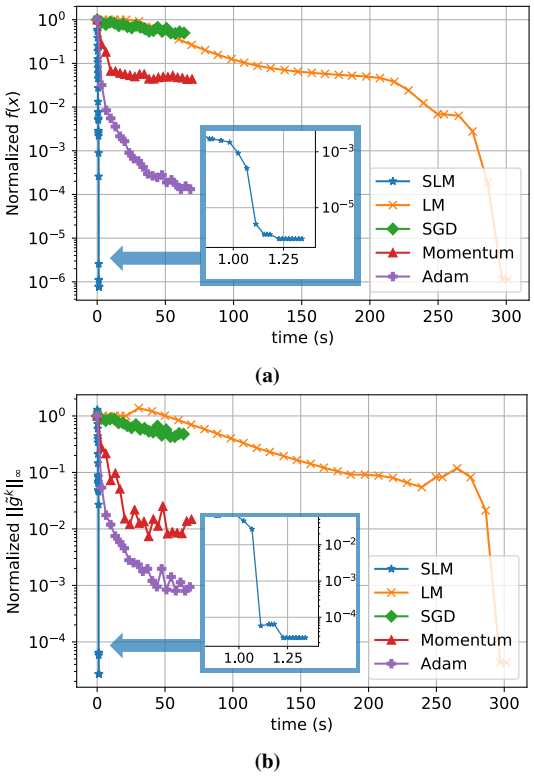


Figure 4: Zero initialization with DF=0.001 (a) $f(x)$ vs Time (b) $\|\tilde{g}^k\|_\infty$ vs Time.

From a random initialization, we observe that the SLM method still finishes the optimization nearly 1 second; see Fig 5a and Fig 5b. Among the first-order methods, Adam also behaves the best. Though it is time-consuming, Adam achieves the same level of normalized $f(x)$ and even better normalized $\|\tilde{g}^k\|_\infty$. This agrees with our results for Strehl Ratio.

3) Scalability Results: We conduct scalability experiments on our SLM implementation with multiple GPUs Since the most time-consuming part of the application is a pair of CUDA kernels for Hessian and gradient kernel ("gh kernel") and function value kernel ("fval kernel"), it is reasonable measure the strong scalability and weak scalability of these two CUDA kernels.

We use three different data fractions in the experiment in order to fully understand the scalability of the algorithm. For strong scaling we use $\frac{T_1}{N \times T_N}$ to express the efficiency. For weak scaling we use $\frac{T_1}{T_N}$ to express the efficiency. The following trend in strong scalability can be observed: when we use larger data fraction we can achieve better scaling. The

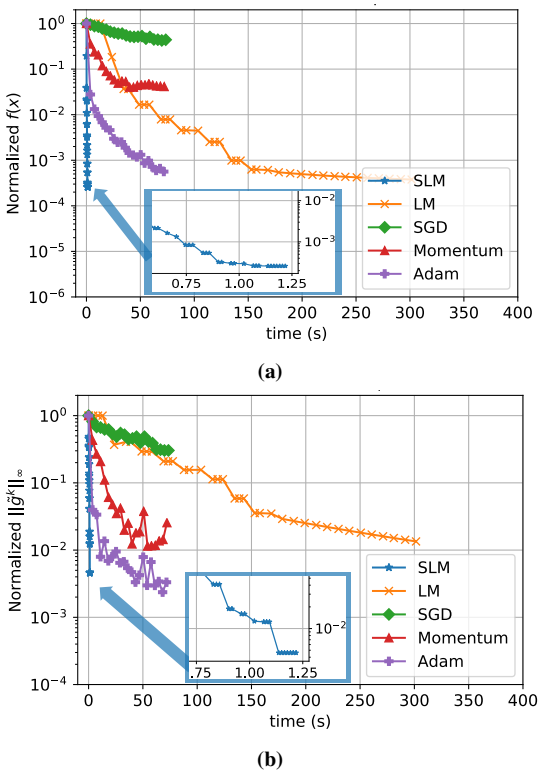


Figure 5: Random initialization with DF=0.001 (a) $f(x)$ vs Time (b) $\|\tilde{g}^k\|_\infty$ vs Time.

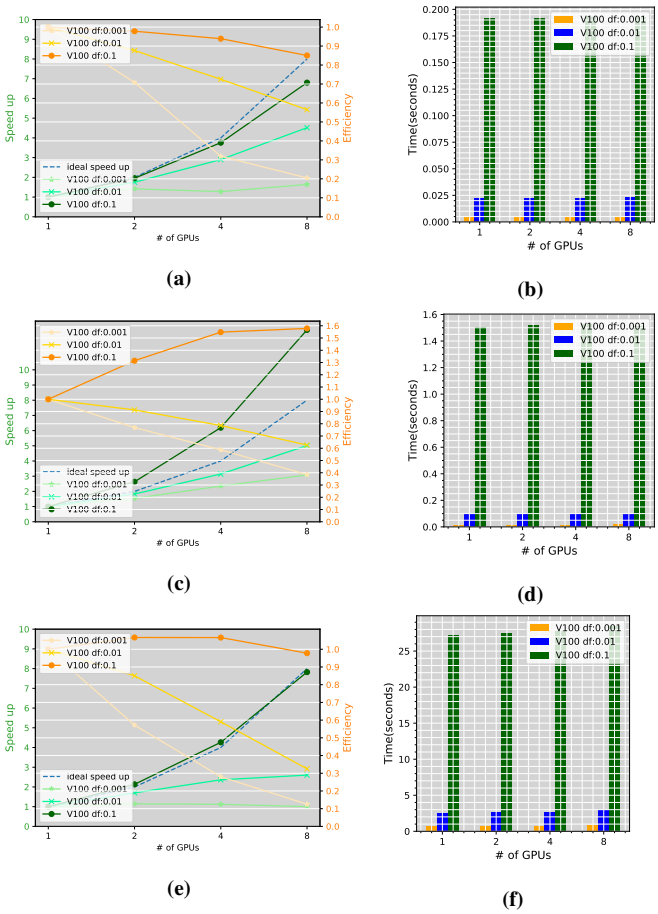


Figure 6: Strong scaling for two CUDA kernels and SLM algorithms with 8 NVIDIA V100 GPUs.

reason is that small data sizes do not fully occupy the CUDA kernels When we use more GPUs, we reduce the size hosted on each GPU. As data fraction goes up, a clear improvement is visible. Both CUDA kernels and algorithms show good weak scalability (note the vertical axis).

We observe that the gh kernel has a superunitary speed up in 6c. One of reasons is that the gh kernel needs a lot of extra registers for the storage of gradient vector. When all of the computation on a single GPU, it waits to fetch the data. In the multi-GPU case, data is evenly distributed among the GPUs so that each CUDA thread has more registers.

We did the same experiments on NVIDIA-DGX with 8 P100 GPUs and latest NVIDIA-DGX with 8 A100 GPus, we observed the same pattern, as is shown in 7 and 8.

The fastest version of all the algorithms is when using SLM with data fraction equals 0.001 running on a 4 P100 GPU. the total time for the optimization is 0.938. We successfully to finish the optimization process in less than 1 seconds.

VIII. Conclusion

We proposed and verified a new Stochastic Levenberg-Marquardt method for solving large-scale nonlinear least-squares optimization problems. The SLM method utilizes a subset of the dataset to estimate the gradient vector and Hessian matrix in order to reduce per iteration cost. We establish a convergence theory for a basic SLM method. We validate the quality and performance of SLM on two small datasets and a large real application dataset. In the real application, GPU hardware accelerators are effectively used to accelerate the computation.

To further validate the performance and robustness of the SLM method, we adapted a classical system identification problem found in ground-based astronomy applications. We found that an acceleration from the order of minutes down to seconds was achieved compared to the traditional LM method in this application, with no notable loss in quality of the optimization result as verified by end-to-end numerical simulations. This result has a considerable positive impact on the performance limitations of ground-based astronomy, particularly in the new generation of Extremely Large Telescopes currently being developed.

Acknowledgment

We thank Nvidia and Paris Observatory for computing resources.

References

[1] F. Roosta-Khorasani and M. W. Mahoney, "Sub-sampled Newton methods," *Mathematical Programming*, vol. 174, no. 1-2, pp. 293–326, 2019.

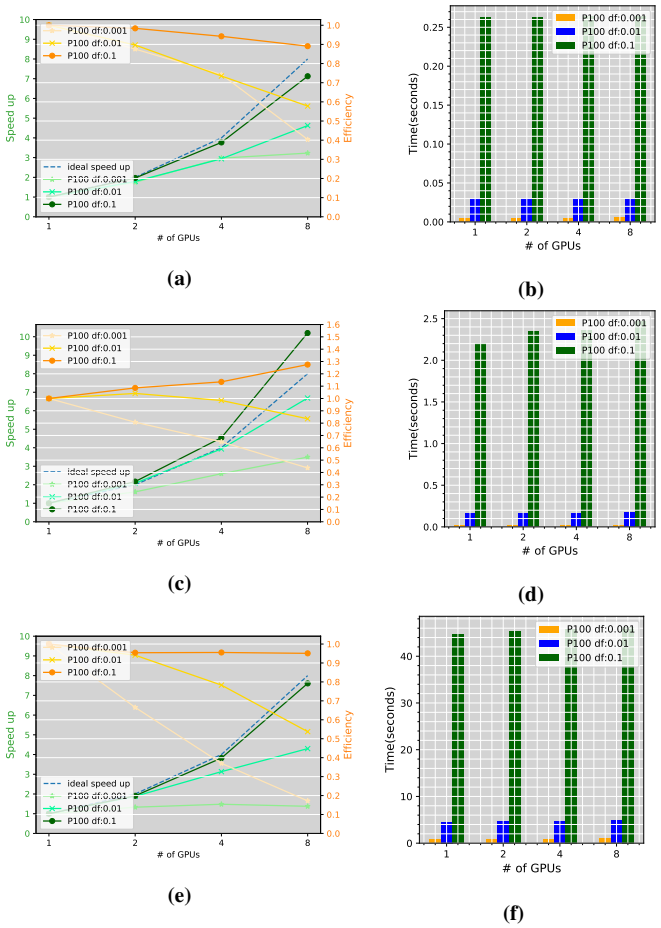


Figure 7: Strong scaling for two CUDA kernels and SLM algorithms with 8 NVIDIA P100 GPUs.

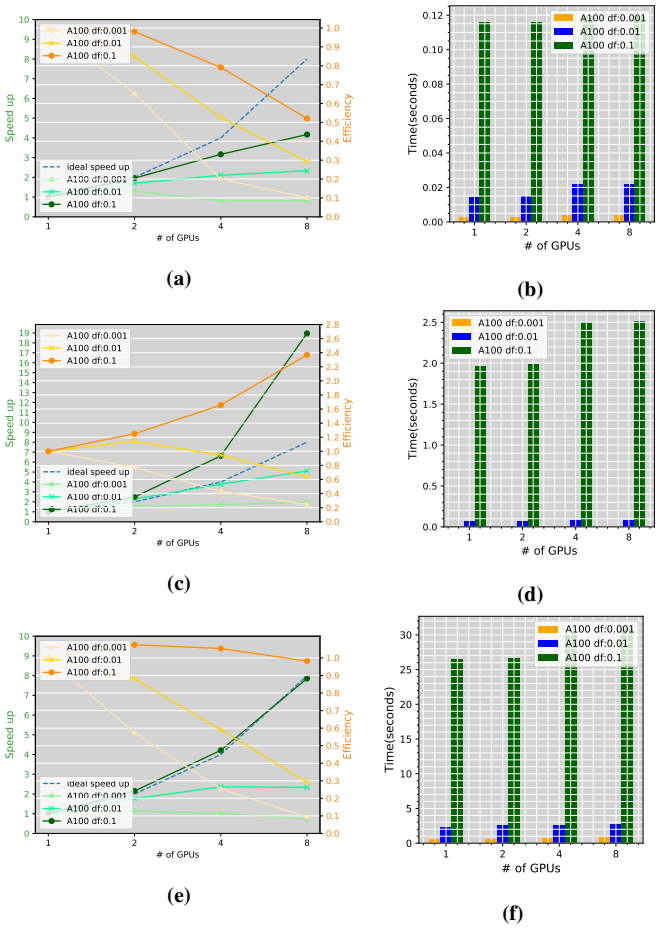


Figure 8: Strong scaling for two CUDA kernels and SLM algorithms with 8 NVIDIA P100 GPUs, x axis is # of GPU.

[2] G. R. F. Vidal, E. Gendron, "Tomography approach for multi-object adaptive optics," *J. Opt. Soc. Am. A*, vol. 27, No. 11, 2010.

[3] C. Kulcsár, H.-F. Raynaud, C. Petit, and J.-M. Conan, "Minimum variance prediction and control for adaptive optics," *Automatica*, vol. 48, no. 9, pp. 1939 – 1954, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109812002750

[4] K. Levenberg, "A method for the solution of certain problems in least squares," *Quart. Appl. Math.*, vol. 2, pp. 164–168, 1944.

[5] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math.*, vol. 11, pp. 431–441, 1963.

[6] E. Bergou, Y. Diouane, and V. Kungurtsev, "Convergence and iteration complexity analysis of a Levenberg-Marquardt algorithm for inverse problems," vol. 185, pp. 927–944, 2020.

[7] E. Bergou, S. Gratton, and L. N. Vicente, "Levenberg-Marquardt methods based on probabilistic gradient models and inexact subproblem solution, with application to data assimilation," *SIAM/ASA J. Uncertain. Quantif.*, vol. 4, pp. 924–951, 2016.

[8] E. Bergou, Y. Diouane, V. Kungurtsev, and C. W. Royer, "A stochastic Levenberg-Marquardt method using random models with application to data assimilation," 2018, arXiv:1807.02176v1.

[9] J. J. Moré, "The Levenberg-Marquardt algorithm: implementations and theory," in *Lecture Notes in Math.*, G. A. Watson, Ed. Berlin: Springer-Verlag, 1977, vol. 360, pp. 105–116.

[10] S. Bellavia, S. Gratton, and E. Riccietti, "A Levenberg-Marquardt method for large nonlinear least-squares problems with dynamic accuracy in functions and gradients," *Numer. Math.*, vol. 140, pp. 791–824, 2018.

[11] E. Gendron, T. Morris, A. Basden, F. Vidal, D. Atkinson, U. Bitenc, T. Buey, F. Chemla, M. Cohen, C. Dickson, N. Dipper, P. Feautrier, J.-L. Gach, D. Gratadour, D. Henry, J.-M. Huet, C. Morel, S. Morris, R. Myers, J. Osborn, D. Perret, A. Reeves, G. Rousset, A. Sevin, E. Stadler, G. Talbot, S. Todd, and E. Younger, "Final two-stage MOAO on-sky demonstration with CANARY," pp. 99 090C–99 090C–17, 2016. [Online]. Available: http://dx.doi.org/10.1117/12.2231432

[12] N. Doucet, D. Gratadour, H. Ltaief, E. Gendron, A. Sevin, F. Ferreira, F. Vidal, R. Kriemann, and D. Keyes, "Efficient supervision strategy for tomographic ao systems on e-elt," 01 2017.
[13] É. Gendron, A. Charara, A. Abdelfattah, D. Gratadour, D. Keyes, H. Ltaief, C. Morel, F. Vidal, A. Sevin, and G. Rousset, "A novel fast and accurate pseudo-analytical simulation approach for MOAO," in *Adaptive Optics Systems IV*, vol. 9148. International Society for Optics and Photonics, 2014, p. 91486L.
[14] R. K. Tyson, *Principles of adaptive optics*. CRC press, 2015.
[15] L. Poyneer, M. van Dam, and J.-P. Véran, "Experimental verification of the frozen flow atmospheric turbulence assumption with use of astronomical adaptive optics telemetry," *JOSA A*, vol. 26, no. 4, pp. 833–846, 2009.
[16] F. Rigaut, D. Brodrick, G. Agapito, V. Viotto, C. Plantet, B. Salasnich, R. Mcdermid, G. Cresci, S. Ellis, M. Aliverti, S. Antoniucci, A. Balestra, A. Baruffolo, M. Bergomi, M. Bonaglia, G. Bono, L. Busoni, E. Carolo, S. Chinellato, G. D. Silva, S. Esposito, D. Fantinel, J. Farinato, T. Fusco, D. Haynes, A. Horton, G. Gausachs, J. Gilbert, D. Gratadour, D. Greggio, M. Gullieuszik, P. Haguenauer, V. Korkiakoski, D. Magrin, L. Magrini, L. Marafatto, H. Mcgregor, T. Mendel, S. Monty, B. Neichel, F. Pedichini, E. Pinna, E. Portaluri, K. Radhakrishnan, R. Ragazzoni, D. Robertson, C. Schwab, R. Sharp, M. Stangalini, S. Stroebele, E. Thorn, A. Vaccarella, D. Vassallo, S. Venkatesan, L. Waller, S. Warner, F. Zamkotsian, and H. Zhang, "Toward a conceptual design for MAVIS," in *Adaptive Optics for Extremely Large Telescopes 6*, June 2019.
[17] K. Madsen, H. B. Nielsen, and O. Tingleff, "Methods for non-linear least squares problems," 2004.
[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representation (ICLR)*, 2015.

APPENDIX

A. Proof for Theorem 1.

Proof. From the $\mathcal{L}$-smoothness of the function f we have

$$\begin{aligned} f^{k+1} &\leq f^k - \nabla f^{k^\top}(x^{k+1}-x^k) + \frac{\mathcal{L}}{2}\|x^{k+1}-x^k\|^2 \\ &\overset{\text{By eq: (4)}}{=} f^k - \nabla f^{k^\top}\left(J^{k^\top}J^k+\mu I\right)^{-1}\tilde{g}^k + \frac{\mathcal{L}}{2}\left\|(J^{k^\top}J^k+\mu I)^{-1}\tilde{g}^k\right\|^2 \\ &\overset{\text{By Lemma 2}}{=} f^k - \frac{1}{\mu}\nabla f^{k^\top}\tilde{g}^k + \frac{1}{\mu^2}\nabla f^{k^\top}J^{k^\top}\left(\frac{1}{\mu}J^kJ^{k^\top}+I\right)^{-1}J^k\tilde{g}^k + \frac{\mathcal{L}}{2}\left\|(J^{k^\top}J^k+\mu I)^{-1}\tilde{g}^k\right\|^2 \end{aligned}$$

which after taking the conditional expectation and by using Lemmas 1, 3 and 4, we get

$$\mathbb{E}[f^{k+1}|x^k] \leq f^k - \frac{1}{\mu}\|\nabla f^k\|^2 + \frac{\kappa_J^2 G^2}{\mu^2} + \frac{\mathcal{L}G^2}{2\mu^2},$$

which after taking the expectation (by using the tower property) we get

$$\mathbb{E}[f^{k+1}] \leq \mathbb{E}[f^k] - \frac{1}{\mu}\mathbb{E}[\|\nabla f^k\|^2] + \frac{2\kappa_J^2 G^2 + \mathcal{L}G^2}{2\mu}.$$

By rearranging the terms and multiplying by μ we get

$$\mathbb{E}[\|\nabla f^k\|^2] \leq \mu\left(\mathbb{E}[f^k] - \mathbb{E}[f^{k+1}]\right) + \frac{2\kappa_J^2 G^2 + \mathcal{L}G^2}{2\mu}.$$

By summing the latter inequality over k from 0 to K and dividing by $K+1$ we get

$$\frac{\sum_{k=0}^K \mathbb{E}\|\nabla f^k\|^2}{K+1} \leq \frac{\mu\left(f^0 - \mathbb{E}[f^{K+1}]\right)}{K+1} + \frac{2\kappa_J^2 G^2 + \mathcal{L}G^2}{2\mu}.$$

By changing μ in the last inequality by $\mu_0\sqrt{K+1}$, and using Assumption 1 we get the desired result. □

B. Proof for Lemma 3.

Proof.

$$\begin{aligned} &\left\|\nabla f^{k^\top} J^{k^\top}\left(\frac{1}{\mu}J^kJ^{k^\top}+I\right)^{-1}J^k\tilde{g}^k\right\| \\ &\quad\leq \|\nabla f^k\|\,\|J^k\|^2\left\|\left(\frac{1}{\mu}J^kJ^{k^\top}+I\right)^{-1}\right\|\|\tilde{g}^k\| \\ &\quad\leq \kappa_J^2 G\left\|\tilde{g}^k\right\|. \end{aligned} \tag{6}$$

Now, by taking the conditional expectation and using Assumption 3, we get the desired result. □