



Extreme Computing for Extreme Adaptive Optics

| | |
|----------------|--|
| Item Type | Conference Paper |
| Authors | Ltaief, Hatem;Sukkari, Dalal E.;Guyon, Oliver;Keyes, David E. |
| Citation | Ltaief H, Sukkari D, Guyon O, Keyes D (2018) Extreme Computing for Extreme Adaptive Optics. Proceedings of the Platform for Advanced Scientific Computing Conference on - PASC '18. Available: http://dx.doi.org/10.1145/3218176.3218225 . |
| Eprint version | Post-print |
| DOI | 10.1145/3218176.3218225 |
| Publisher | Association for Computing Machinery (ACM) |
| Journal | Proceedings of the Platform for Advanced Scientific Computing Conference on - PASC '18 |
| Rights | Archived with thanks to Proceedings of the Platform for Advanced Scientific Computing Conference on - PASC '18 |
| Download date | 2024-04-09 07:29:36 |
| Link to Item | http://hdl.handle.net/10754/627414 |

Extreme Computing for Extreme Adaptive Optics: the Key to Finding Life Outside our Solar System

ABSTRACT

The real-time correction of telescopic images in the search for exoplanets is highly sensitive to atmospheric aberrations. The pseudo-inverse algorithm is an efficient mathematical method to filter out these turbulences. We introduce a new partial singular value decomposition (SVD) algorithm based on QR-based Diagonally Weighted Halley (QDWH) iteration for the pseudo-inverse method of adaptive optics. The QDWH partial SVD algorithm selectively calculates the most significant singular values and their corresponding singular vectors. We develop a high performance implementation and demonstrate the numerical robustness of the QDWH-based partial SVD method. We also perform a benchmarking campaign on various generations of GPU hardware accelerators and compare against the state-of-the-art SVD implementation SGESDD from the MAGMA library. Numerical accuracy and performance results are reported using synthetic and real observational datasets from the Subaru telescope. Our implementation outperforms SGESDD by up to fivefold and fourfold performance speedups on ill-conditioned synthetic matrices and real observational datasets, respectively. The pseudo-inverse simulation code will be deployed on-sky for the Subaru telescope during observation nights scheduled early 2018.

CCS CONCEPTS

• **Mathematics of computing** → **Mathematical software performance**; • **Computing methodologies** → **Massively parallel algorithms**; • **Computer systems organization** → **Real-time system architecture**;

KEYWORDS

Computational Astronomy, Adaptive Optics, Pseudo-Inverse Calculation, GPU Computations, QDWH, Partial SVD

ACM Reference Format:

. 2018. Extreme Computing for Extreme Adaptive Optics: the Key to Finding Life Outside our Solar System. In *Proceedings of ACM Platform for Advanced Scientific Computing Conference (PASC'18)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Over the last decade, astronomers have identified thousands of exoplanets - planets orbiting stars other than our Sun. There is now solid evidence that habitable planets (rocky planets with temperate

surface temperature, able to hold liquid water) are abundant. Most of the 200 billion stars in our galaxy may host such planets. While current telescopes do not have the sensitivity to find evidence of life on nearby habitable planets, the next generation of large (~30m diameter) telescopes is entering construction and will enter operation during the 2020s decade. These larger telescopes will be capable of detecting water, oxygen, and other gases indicative of biological activity. Yet, a significant challenge remains to be solved: Earth's atmosphere blurs the image, rendering exoplanet imaging extremely difficult.

1.1 Adaptive Optics

To deliver sharp images of the sky, ground-based astronomical telescopes must overcome optical disturbances introduced by Earth's atmosphere. Without active correction of such defects, images would be blurred to approximately one arcsecond angle (1/3600 of a degree), while in the absence of optical aberrations, a telescope of diameter D imaging at wavelength λ should provide λ/D angular resolution (≈ 0.04 arcsecond in the near-infrared with current 10-m class telescopes). To recover this approximately 25-fold loss of angular resolution, adaptive optics (AO) systems [8, 9, 18, 27, 29] measure and correct for atmospheric turbulence. Any AO system must include three key elements:

- One or several **wavefront sensor(s)** (WFS) measure optical aberrations in the incoming light beam. The light source for the WFS must be located near the scene to be imaged, and can be a natural star or an artificial laser guide star if no bright star is near the area of interest. Multiple WFSs can participate to the measurement to provide correction over a wider field of view.
- One or several **deformable mirror(s)** (DM) perform the optical correction. The DM(s) induce optical aberrations that are opposite to the ones induced by atmospheric turbulence.
- A **real-time controller** (RTC) processes the WFS signals to compute DM commands. The RTC must reliably compute commands at 1-5 kHz frequency so that the AO system can keep up with fast changes in atmospheric turbulence.

AO systems are in use on most current large telescopes, routinely delivering images that are as sharp as allowed by the telescope diffraction limit λ/D .

1.2 Exoplanet Imaging, Search for Extraterrestrial Life and Extreme AO

Astronomers have long suspected that exoplanets orbit many of the stars visible in the night sky, by analogy to our solar system. The term **exoplanet** is used here to distinguish them from the planets in our solar system. While exoplanets are challenging to observe due to their small mass and size in comparison with the

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

PASC'18, July 2018, Basel, Switzerland

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

stars they orbit, technological advances have recently enabled their detection in large numbers. Several thousand exoplanets have now been identified [1], and a significant fraction of stars (from 20% to 50%) is believed to host Earth-size rocky planets with temperate surface where liquid water may exist; such planets are referred to as **habitable planets**. Despite the large and growing number of confirmed exoplanets, little is known about their potential to harbor life: habitable planets can currently only be detected indirectly from the gravitational pull they exert on their host star [6], or from the transit events when the planet passes in front of the star [13]. Characterization of exoplanet atmospheres and search for signs of biological activity will require their light to be isolated and analyzed in a spectrograph so that molecular species such as oxygen and water can be identified. Exoplanet imaging and spectroscopy are the most challenging application for AO systems: exoplanets are between five to ten orders of magnitude fainter than the stars they orbit, and the angular separation between the two objects is usually well below one arcsecond. AO systems optimized for this task are referred to as **Extreme AO** systems (ExAO) due to the extreme level of wavefront correction required. While conventional AO systems aimed at reaching the telescope diffraction limit must correct optical aberrations to a ≈ 1 radian phase residual error (corresponding to $\approx 250\text{nm}$ in the near-infrared), ExAO systems require nm-level exquisite wavefront correction of low and mid-spatial frequencies. As shown in Figure 1, ExAO systems must also employ optical coronagraphy to remove diffracted starlight, and must further reduce wavefront errors to limit how much of the bright starlight overlaps with the planet image [15]. The brightest, largest and nearest exoplanets can be imaged with current large telescopes and high performance AO systems [20, 24]: these gas giant planets are several times more massive than Jupiter (which itself is $317 \times$ Earth mass) and are seen in near-IR light owing to their strong internal heat. While habitable exoplanet imaging is widely considered out of reach of current telescopes, the next generation of 30m class telescopes, when equipped with high performance ExAO systems, will have the sensitivity to image and study habitable planets around dozens of nearby stars.

Due to the exquisite wavefront correction requirements of ExAO systems and their large number of sensing and correcting elements, the RTC is a significant challenge: every few hundred microseconds, new input measurements must be processed to compute DM(s) corrections. The computational requirements are significant due to the size of the input and output spaces (each several tens of thousands of variables), high speed (1-10 kHz) and the need for an accurate solution to be produced.

1.3 Extreme Computing Needs

One of the major components of the AO framework consists in computing the pseudo-inverse of the response matrix (RM), a linear calibration generated during the observational data acquisition. The pseudo-inverse algorithm is composed of dense linear algebra operations based on compute-intensive numerical kernels. This class of algorithms usually maps well onto manycore architectures, such as GPU hardware accelerators. The main idea consists in extracting the most significant singular values with their associated singular vectors and calculating the explicit pseudo-inverse of the matrix

from this subspectrum. The challenges for the pseudo-inverse are twofold: numerical and computational. The numerical challenge resides in dealing with the rectangular shape of the control matrix (CM), which may engender numerical instabilities if a spectral decomposition, i.e., symmetric eigensolver, is employed. On the other hand, the computational challenge lies in the high algorithmic complexity of the pseudo-inverse: although being not real-time, it should still be able to keep up with the overall throughput of the AO framework.

We propose a novel high performance implementation of the partial SVD decomposition using an extension of the QR-based dynamically weighted Halley algorithm (QDWH) [25]. This new algorithm translates the original matrix problem into a reduced problem size containing only the singular values/vectors of interest. The QDWH-based partial SVD relies on GPU-friendly numerical kernels, which expose more parallelism than the standard bidiagonal-based SVD solver [14]. We benchmark our new pseudo-inverse implementation against the SVD routine with divide and conquer solver (i.e., SGESDD) from the MAGMA library [4] on various generations of GPU hardware accelerators using synthetic (randomly generated matrices) as well as matrices coming from real datasets generated from the Subaru Telescope.

The paper is organized as follows. Section 2 outlines general background on ExAO framework to support current and next generation of telescopes, and exposes the computational challenge to be addressed. Section 3 presents related work on the AO technique and the computational framework. Section 4 highlights our contributions. Section 5 recalls algorithmic variants of the pseudo-inverse method and introduces the new QDWH-based partial SVD algorithm. The details of the various high performance implementations are given in Section 6. Section 7 provides the performance results of various pseudo-inverse approaches on GPUs and compares them against existing state-of-the-art implementations. We conclude in Section 8.

2 EXTREME AO'S EXTREME COMPUTING CHALLENGE

The ExAO system must perform real-time computation of a DM solution from WFS input data. As described in §2.1, this consists of a matrix-vector multiplication (MVM), which is well within the computing capabilities of modern manycore hardware: its optimization is not discussed in this paper. Derivation of the control matrix to perform this MVM operation, described in §2.2, is considerably more challenging: this is the challenge we are addressing in this paper. As shown in §2.3, it relies on computing the pseudo-inverse of large matrices constructed from WFS measurements.

Current (2018) implementations of SGESDD deployed on off-the-shelf GPU hardware fall approximately one order of magnitude short of full-scale derivation of predictive control and sensor fusion control laws on ExAO systems deployed on current large telescope. The controller optimization scope must therefore be reduced by, for example, selecting a subset of the control modes, processing a subset of the accumulated WFS data, or selecting a subset of the available sensors. The performance gap is yet another of magnitude wider for 30-m class telescopes.

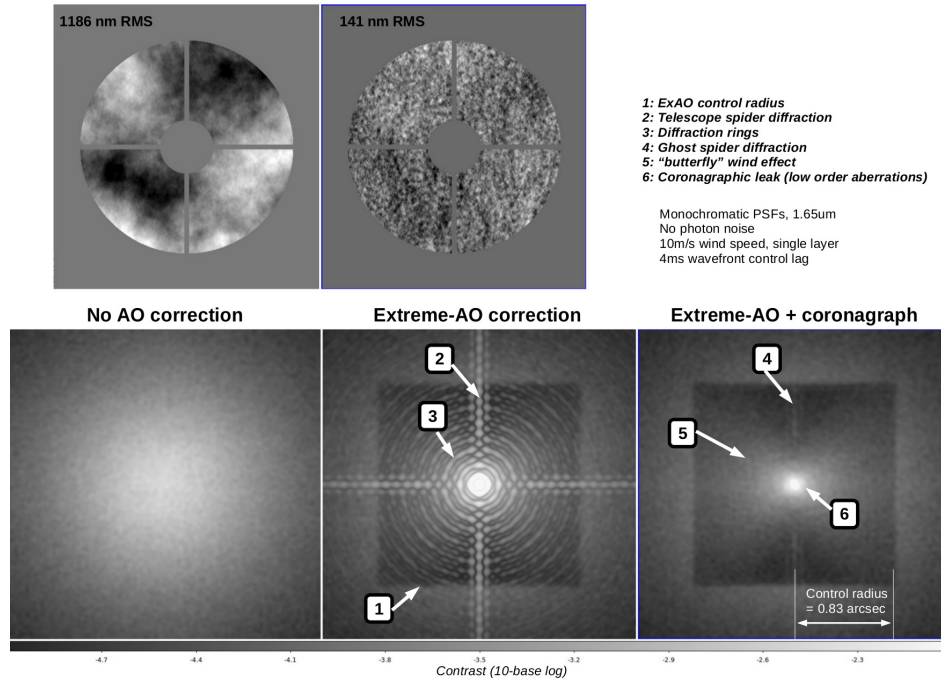


Figure 1: Image formation in an ExAO system optimized for exoplanet imaging. In this numerical simulation, a bright star is imaged in near-infrared light ($\lambda = 1.65\mu\text{m}$ with a 8-m diameter telescope). Without AO correction (left), Earth’s atmosphere creates more than one μm of optical aberrations, and the corresponding image (labeled “No AO correction”) is an approximately one arcsecond wide spot. With ExAO correction (center), the aberrations are reduced to 141 nm, and the image (labeled “Extreme AO correction”) shows that starlight is concentrated in a central diffraction-limited spot. Starlight due to diffraction (components 2 and 3) can be suppressed optically with a coronagraph. Residual starlight in the final image (labeled “Extreme AO + coronagraph”) is entirely due to uncorrected optical aberrations (components 4, 5, and 6). The ExAO correction performance defines the ability to image faint planets near the star.

2.1 Real-time computations

The conventional AO control scheme is to derive from each WFS measurement the corresponding DM solution. The relationship between WFS and DM is generally assumed to be linear, so this step is a MVM. The input vector (WFS measurement) is multiplied by the CM to estimate the output DM vector that would generate the measured WFS signal. By sending a DM command opposite to this output vector, the optical aberrations measured by the WFS are canceled. Figure 2 shows two more advanced control schemes, which are essential to ExAO systems performance.

2.1.1 Predictive Control. The conventional AO control scheme suffers from time lag: corrections applied to the DM are slightly outdated due to hardware and software delays, including WFS camera exposure, readout and data transfer time, computing time, and DM response time. Together, these delays typically add to one millisecond. In addition to hardware and software lag, conventional AO controllers perform some time-averaging of WFS measurements to reduce measurement noise due to shot noise and WFS readout noise. The combined effect is a 1 to 5 ms effective time delay in the correction, and a corresponding error in the AO correction. In ExAO imaging, this error term needs to be reduced by use of **predictive control**: the last N measurements should be optimally

used to estimate the optical aberrations at the time of correction. As shown in Figure 2, predictive control can be implemented by adopting the last N WFS measurements as the input to the correction. The corresponding control matrix is then N times larger than in conventional AO control.

2.1.2 Sensor Fusion. ExAO’s need for high accuracy measurement of optical aberrations requires multiple sensors to be deployed, and their signals to be optimally combined. The multi-sensor approach can measure small wavelength-dependent changes in optical aberrations, and reduces the null measurement space to the intersection of individual WFSs null spaces. Figure 2, bottom panel, shows a control scheme combining prediction and sensor fusion, assuming that each of the K sensors has the same number of elements n . The control matrix size can be significantly larger than in the two previous examples.

2.2 Inferring Control Matrices from Measurements

In each of the cases presented, the CM computation is a non-real-time task that must be performed before operating the AO system, and may need to be updated while the system is running.

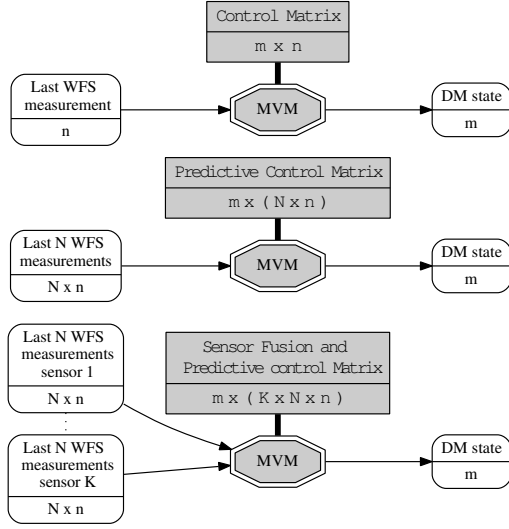


Figure 2: Three possible AO control loop real-time implementations: conventional AO control (top), predictive control (center) and sensor fusion + predictive control (bottom). Vector and matrix sizes are shown below names.

In conventional AO control, the CM is computed as the pseudo-inverse of the system response matrix (RM) describing the linear response between DM commands and WFS measurements. The RM is acquired by issuing commands to the DM and measuring the corresponding WFS signal changes. The pseudo-inverse is most often computed using the SVD to allow control modes selection, and reject modes for which the WFS response is weak.

In predictive control and sensor fusion, the linear relationships between WFS measurements are usually poorly constrained, and cannot be measured directly. For example, the linear relationship between past and present optical aberrations is a function of wind speed at multiple altitudes above the telescope and the contribution of each atmosphere layer to the overall measured wavefront. This information is generally not accessible in real-time, so the predictive control matrix must be inferred from the WFS measurements.

The problem can be translated into a **supervised machine learning** challenge operating in a continuum multi-dimensional space where input vectors are WFS measurements (including history for predictive control and spanning multiple WFSs for sensor fusion) and output vectors are the optical aberrations without time lag. Both input and output vectors of the training set are acquired by the WFS(s). We propose to investigate this problem using an analytical approach, although more sophisticated statistical approaches (based on the maximum likelihood estimation) for the prediction may be more suitable [3], thanks to the natural linear variation of the atmospheric turbulence. This is beyond the scope of this paper and will be reported in the future.

2.3 The SVD-Based Pseudo-Inverse Approach

Thanks to the high measurement cadence, a large training set can be acquired on a timescale shorter than the temporal evolution

of underlying linear dependencies (e.g., a change in wind speed and direction): at 2 kHz measurement cadence, $l = 1.2$ million samples are collected in 10 min. In a typical ExAO system, upward of $n = 1,000$ modes are measured by the main WFS. The predictive control challenge is a supervised machine learning problem, aimed at finding the n -by- Nn prediction matrix X that maps input vectors a containing the N last WFS measurements to the m -sized output space consisting of future value of the WFS coefficients. The input vector a and output vector b are therefore of sizes Nn and $m = n$ respectively. The optimal prediction matrix X minimizes the euclidian distance $\|Xa - b\|$ across the training set consisting of l pairs (a, b) . By writing the l training pairs (a, b) as a Nn -by- l **data matrix** A , and a n -by- l **output matrix** B , the problem requires solving a linear system $XA = B$ where the solution X , i.e., the prediction matrix, is a $(n \approx 1000)$ -by- $(Nn \approx 20,000)$ matrix. The linear system must be solved in a few minutes to keep up with incoming data rates. Computation requirements become even larger with sensor fusion, as described in Section 2.1.2: the dimension of the input space grows linearly with the number K of WFSs.

Computing the solution $X = BA^+$ with a SVD-based pseudo-inverse allows to filter out the noise from the empirical measurement data structures A , by extracting the most significant singular values and their corresponding singular vectors.

3 RELATED WORK

Computing the pseudo-inverse is a critical computational phase of the AO framework. There are several algorithmic variants to calculate it, mostly based on two-sided transformations, i.e., the symmetric eigensolver (SEVD) and the SVD. These two solvers rely on block algorithms and use a one-stage approach, as implemented in LAPACK [5]. They basically reduce the dense matrix into condensed structures, i.e., tridiagonal and bidiagonal forms, for the SEVD and SVD solvers, respectively.

Both algorithms have been leveraged during the last two decades by breaking the one-stage reduction into a two-stage approach, which casts most of the computations in terms of compute-intensive Level 3 BLAS. The dense matrix is first reduced to band tridiagonal/bidiagonal form. Then, the extra off-diagonal elements are chased down at the bottom of the matrix during a bulge chasing procedure, until the required condensed forms appear. The corresponding solver carries on with the computation and extracts the pairs of eigen/singular values/vectors from the condensed forms, followed by successive back transformations generated during the two-stage reduction, to get the final SEVD and SVD of the original dense matrix.

The two-stage approach has been originally introduced in [10] on shared-memory systems using multithreaded BLAS. The two-stage approach for SEVD and SVD has been further optimized on multi-core x86 architectures [16, 17, 22, 23], by relying on tile algorithms using the PLASMA library [4], associated with a dynamic runtime system to enable asynchronous fine-grained task executions. As opposed to two-stage SVD algorithm, the two-stage SEVD method has been eventually ported on distributed-memory systems and on systems equipped with hardware accelerators, as implemented in the ELPA library [7] and in the MAGMA library [4], respectively.

Although the SEVD-based pseudo-inverse has demonstrated significant performance in the context of AO simulation [2, 12], there is a fundamental numerical bottleneck with this approach. Indeed, the pseudo-inverse requires to make the outer product of the original dense matrix with its transpose, from which the SEVD solver can be applied. This outer product squares the condition number of the matrix and may create numerical instabilities. Another statistical-based approach has been investigated recently [11, 21], which consists into explicitly generating the covariance matrix of the telemetry matrix and to take into account the noise from the system. Although highly efficient, the covariance matrix generation may be time-consuming, due to its expensive memory accesses.

Calculating the pseudo-inverse using the SVD has not been considered for AO in computational astronomy, due to its high algorithmic complexity. Based on the QR-based dynamically weighted Halley algorithm (QDWH), a novel algorithm for computing the SVD has been introduced in [25]. Although highly parallel with GPU-friendly numerical kernels [28], its number of floating-point operations is even higher than the standard SVD algorithm. An extension of the aforementioned QDWH-based SVD permits to calculate only the most significant singular values and corresponding singular vectors, reducing the overall algorithmic complexity, while efficiently computing the pseudo-inverse. This QDWH-based partial SVD represents the crux of the paper.

4 CONTRIBUTIONS

The paper contributions are threefold. First, we introduce a new QDWH-based partial SVD algorithm for the pseudo-inverse method, which calculates the most significant singular values and their corresponding singular vectors. Second, we develop a high performance implementation and present the numerical robustness of the QDWH-based partial SVD method on synthetic and real datasets. Last but not least, we perform a benchmarking campaign on various generations of GPU hardware accelerators and compare against the state-of-the-art MAGMA library [4]. The resulting pseudo-inverse simulation code will be deployed on-sky for the Subaru Telescope during the next observation night, scheduled early this year.

5 THE PSEUDO-INVERSE ALGORITHM

This section describes the various approaches for calculating the pseudo-inverse of a general dense matrix and presents their algorithmic complexities.

5.1 The Pseudo-Inverse Based on the Symmetric Eigenvalue Decomposition

The pseudo-inverse of a general dense matrix A using the SEVD can be described as in Algorithm 1. The general matrix is first symmetrized by multiplying it by its transpose. The eigenvalue decomposition of the resulting symmetric matrix is then performed, from which only the eigenpairs above a certain threshold are saved. The pseudo-inverse can then be computed by forming the diagonal and the orthogonal matrices containing the selected eigenvalues and eigenvectors, respectively. The pseudo-inverse is finally obtained by effectively multiplying both matrices, while taking advantage of their nice numerical properties, e.g., the inverse of an orthogonal matrix is equal to its transpose.

Algorithm 1 Pseudo-Inverse using the spectral decomposition.

- 1: Compute $S = A^T A$
 - 2: Solve $Sx = \lambda x$
 - 3: Extract the eigenpairs for which $\lambda > threshold$
 - 4: Calculate the pseudo-inverse $S^+ = X^T \Lambda^{-1} X$
-

The main challenge with this approach is its numerical robustness. Indeed, the condition number of the symmetric matrix S is equal to the square of the condition number of the original matrix A . Although some applications may be resilient to it, this approach is usually not pursued as numerical instabilities may emerge and eventually disturb the eigensolver employed during the diagonalization procedure.

From an algorithmic complexity perspective, the bulk of the computation occurs at the symmetric eigenvalue decomposition. As indicated in Section 3, one-stage or two-stage tridiagonal reductions correspond to the first computational phase for the symmetric eigenvalue decomposition, regardless if a single or a subset of eigenpairs is needed. This reduction phase may typically account up to 50% of the total elapsed time, when all eigenpairs are needed. Once the condensed form has been achieved, there exist then symmetric eigensolvers (e.g., based on the multiple relatively robust representations), which are capable of extracting only the eigenpairs of interests within a user-defined range or interval. The back transformation has still to be carried on to retrieve the spectral decomposition of the original dense matrix.

5.2 The Pseudo-Inverse Based on the standard SVD

The pseudo-inverse of a general dense matrix A using the SVD can be described as in Algorithm 2. This approach does not suffer from numerical sensitivity but may be time-consuming, due to the high number of operations. In case the matrix is rectangular, an initial QR or LQ factorization is applied to annihilate the extra rows and columns, respectively. The first computational phase, which reduces the dense matrix into bidiagonal form, may cost twice as much as the tridiagonal reduction. Moreover, similarly to the symmetric eigensolver for the pseudo-inverse in Section 5.1, this reduction to condensed form is required, regardless of how many singular values/vectors are needed. In fact, the SVD solver based on the QR algorithm or the divide-and-conquer method cannot calculate a subset of the singular values/vectors. Therefore, the whole spectrum of the singular values/vectors has first to be determined and later filtered out using the threshold parameter, set by the application. The back transformation is then necessary to compute the singular values/vectors of the original matrix A , and here again, this expensive computational stage is mandatory, regardless if a single or several singular values/vectors have been requested.

Algorithm 2 Pseudo-Inverse using the standard SVD.

- 1: Compute $A = U \Sigma V^T$
 - 2: Extract the singular values/vectors for which $\sigma > threshold$
 - 3: Calculate the pseudo-inverse $A^+ = \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}^T$
-

5.3 The Pseudo-Inverse Based on the QDWH-based partial SVD

We herein present an approach, which combines the numerical robustness and the low arithmetic complexity to calculate the pseudo-inverse of a dense matrix A . By extending the QR-based dynamically weighted Halley algorithm (QDWH) [25] for the SVD, this new SVD-based pseudo-inverse inherently provides the flexibility to calculate only a subset of the singular values/vectors spectrum, while ensuring backward numerical stability. To our knowledge, this is perhaps the only dense SVD solver, which provides this versatility toward computing only a partial SVD within a user-defined range or interval. This QDWH-based partial SVD algorithm permits to project the original matrix spectrum into a reduced matrix problem containing only the desired singular values/vectors. Therefore, it allows to remove all together the incompressible costs required by the two aforementioned methods during the reduction phase to condensed form. In case the original matrix problem is rectangular, similarly to the pseudo-inverse based on the standard SVD, an initial QR or LQ factorization is applied to annihilate the extra rows and columns, respectively.

Algorithm 3 presents the pseudo-inverse method for a general dense matrix A using the QDWH-based partial SVD. The iterative QDWH procedure used in the partial SVD calculation is a lightweight variant of the original QDWH method, i.e., performing less iterations and without estimations of the matrix condition number and second norm, to calculate the polar decomposition $A = U_p H$. It requires only a few Cholesky or QR-based iterations to get the polar factor U_p containing the desired singular values/vectors. In fact, QDWH enables to isolate the subspectrum of interest from the overall spectrum. This is done through a tunable numerical user-defined *threshold*. The *threshold* is an application-dependent parameter, which physically corresponds to the lower bound of the wanted singular values. In fact, it is a tunable parameter in the sense that it infers the actual number of wanted singular values. Therefore, there is a direct link between the percentage of the wanted singular values and the threshold Li . A QR factorization is then applied on $U_p + Id$ to reveal the index of the first diagonal elements of R in absolute value, which is below the *threshold*, also referred to as the null space. This index then determines the orthogonal vectors to extract from Q , from which the projected size of the reduced matrix problem is calculated. Once the smaller matrix problem is generated, a standard SVD solver can be applied to get the full set of singular values/vectors. This full set of singular values/vectors of the reduced matrix problem relates to the subset of desired singular values/vectors from the original matrix problem. The desired singular values and left singular vectors are identical for both matrix problems, while the desired right singular vectors of the original matrix problem needs to accumulate \tilde{Q} with the right singular vectors of the reduced matrix problem. This pseudo-inverse approach presents many advantages. It relies on one-sided transformations, i.e., Cholesky and QR factorizations, which expose much more parallelism than the previous two pseudo-inverse methods based on two-sided transformations. It also focusses the computational power only on the desired spectrum, without wasting resources by *over-solving* for the whole spectrum. Under extremely ill-conditioned matrices, the method may however raise numerical

issues, in case a larger spectrum of the singular values/vectors are needed. These issues may be fixed by enforcing QR-based QDWH iterations (instead of Cholesky), at the expense of increasing the algorithmic complexity, and therefore, the overall elapsed time.

Algorithm 3 Pseudo-Inverse using the QDWH-Based Partial SVD.

- 1: Compute the polar decomposition $A = U_p H$ using QDWH
 - 2: Calculate $[Q \ R] = QR(U_p + Id)$
 - 3: Find the index $ind = \min(find(abs(diag(R)) < threshold))$
 - 4: Extract $\tilde{Q} = Q(:, ind : end)$
 - 5: Reduce the original matrix problem $\tilde{A} = A \times \tilde{Q}$
 - 6: Compute the SVD of the reduced matrix problem $\tilde{A} = U \Sigma \tilde{V}^T$
 - 7: Compute the right singular vectors $V = \tilde{Q}^T \times \tilde{V}$
 - 8: Calculate the pseudo-inverse $A^+ = V \Sigma^{-1} U^T$
-

5.4 Algorithmic Complexity

Without loss of generality, Table 1 reports the algorithmic complexity for square matrices of size Nn (i.e., the number of predictive filter steps times the number of modes measured by the WFS) based on the three pseudo-inverse approaches discussed in this section. The pseudo-inverse based on the SEVD is cheaper than the SVD-based but may encounter numerical issues, especially for ill-conditioned matrices. The algorithmic complexity of the pseudo-inverse based on the QDWH-based partial SVD depends on the number of QDWH Cholesky-based iterations (typically two or three) and the size s of selected singular values/vectors. Assuming $s \ll Nn$ and three iterations to get the polar factor from QDWH, the total number of operations is $14Nn^3$, about 66% of the pseudo-inverse based on the standard SVD. The actual SVD solver occurs now only on the reduced problem matrix of size s .

| | SEVD-based | Standard SVD-based | QDWH-based partial SVD |
|------------------------|------------|--------------------|--|
| Algorithmic complexity | $9Nn^3$ | $22Nn^3$ | QDWH: $(4+1/3)Nn^3 \times \#it_{Chol}$ QR and GEMM: $4/3Nn^3 + 2sNn^2 + 2Nns^2$ SVD: $22s^3$ |

Table 1: Algorithmic complexity for various pseudo-inverse algorithms.

6 HIGH PERFORMANCE GPU-BASED IMPLEMENTATION

We describe the high performance GPU-based implementation of the pseudo-inverse of a dense matrix A of size n using the QDWH-based partial SVD algorithm, described in Section 5.3. The pseudo-code is highlighted in Algorithm 4 and relies on the MAGMA [4] and NVIDIA cuBLAS [26] libraries. The code is mostly composed of dense linear algebra kernels rich in compute-intensive Level 3 BLAS operations, which makes it GPU-friendly. It can therefore achieve a decent percentage of the system's theoretical peak performance. Our implementation is GPU-resident, i.e., the memory footprint of the code fits on the GPU's main memory. MAGMA provides some support for out-of-core algorithms, in case the GPU's main memory is limited or simply the problem is too large to fit on the device. Data movement will have then to be carefully handled, since the GPU

interconnect bandwidth (i.e., PCIe bus) is typically more than an order of magnitude slower than the GPU onboard memory bandwidth. Last but not least, although the pseudo-code is written in single precision arithmetics to satisfy the computational requirements of the AO framework, it can be easily extended to other precisions for a broader application impact.

Algorithm 4 Pseudo-code of the pseudo-inverse matrix calculation based on the QDWH partial SVD using MAGMA/cuBLAS.

```

1:  $k = 1$ ,  $Li = \text{threshold}$ ,  $\text{conv} = 100$ 
2: while ( $\text{conv} \geq \sqrt[3]{\epsilon \text{ps}} \ \& \ |1 - Li| \geq \epsilon \text{ps}$ ) do
3:    $L2 = Li^2$ ,  $dd = \sqrt[3]{(4(1 - L2)/L2^2)}$ 
4:    $sqd = \sqrt{1 + dd}$ 
5:    $a1 = sqd + \sqrt{8 - 4 \times dd + 8 \times (2 - L2)/(L2 \times sqd)})/2$ 
6:    $a = \text{real}(a1)$ ,  $b = (a - 1)^2/4$ ,  $c = a + b - 1$ 
7:    $Li = Li \times (a + b \times L2)/(1 + c \times L2)$ 
8:    $B \leftarrow \text{magma\_slaset}(\text{MagmaFull}, 0.0, 1.0, B)$ 
9:    $B \leftarrow \text{cublasSgemm}(A^T, A, B)$ 
10:   $C \leftarrow \text{magma\_stranspose}(A, C)$ 
11:   $C \leftarrow \text{magma\_sposv}(\text{MagmaUpper}, B, C)$ 
12:   $U_p^k \leftarrow \text{magma\_slacpy}(\text{MagmaFull}, A, U_p^k)$ 
13:   $A \leftarrow \text{cublasSgeam}(C, U_p^k, A)$ 
14:   $U_p^{k+1} \leftarrow \text{cublasSgeam}(A, U_p^k, U_p^k)$ 
15:   $\text{conv} \leftarrow \|U_p^{k+1} - U_p^k\|_F$ 
16:   $k = k + 1$ 
17: end while
18:  $B \leftarrow \text{magma\_slaset}(\text{MagmaFull}, 0.0, 1.0, B)$ 
19:  $U_p \leftarrow \text{cublasSgemm}(A^T, A, B)$ 
20:  $U_p \leftarrow \text{magma\_sgeqrf}(U_p, \text{tau})$ 
21:  $\text{ind} = \min(\text{find}(\text{abs}(\text{diag}(U_p)) < \text{threshold}))$ 
22:  $Q \leftarrow \text{magma\_sorgqr}(U_p, \text{tau}, Q)$ 
23:  $\tilde{Q} = Q(:, \text{ind}:\text{end})$ 
24:  $\tilde{A} \leftarrow \text{cublasSgemm}(A_{\text{orig}}, \tilde{Q}, \tilde{A})$ 
25:  $[U \Sigma \tilde{V}^T] \leftarrow \text{magma\_sgesvd}(\tilde{A})$ 
26:  $V \leftarrow \text{cublasSgemm}(V, \tilde{Q}^T, V)$ 
27:  $B \leftarrow \text{cublasSgemm}(V, \Sigma^{-1}, B)$ 
28:  $A^+ \leftarrow \text{cublasSgemm}(B, U^T, A^+)$ 

```

7 EXPERIMENTAL RESULTS AND ANALYSIS

This section describes the experimental results and presents performance analysis of the GPU-based implementation for the QDWH partial SVD used in the pseudo-inverse, using synthetic matrices and real observational datasets.

7.1 Environment Settings

We perform our experiments on three GPU systems, each equipped with a different hardware accelerator generation: **K80**, **P100** and **V100** with 12GB, 16GB and 16GB of memory, respectively. The GPU **K80** host is a two-socket 14-core Intel Broadwell system with 128GB of main memory. The GPU **P100** and **V100** hosts are two-socket 16-core Intel Haswell systems, each with 128GB of main memory. We use MAGMA v2.3 and CUDA v9.0 (including cuBLAS) with GCC compilers. Our ultimate goal is to use the QDWH partial SVD algorithm to solve the machine learning predictive control challenge, described in §2.1.1 for the Subaru Coronagraphic Extreme AO (SCEAO) system [19], deployed on the 8.3m diameter Subaru Telescope. We compare our QDWH partial SVD solver against the standard SVD based on the divide-and-conquer solver, since the symmetric eigendecomposition (SEVD) may raise numerical

issues, as described in Section 5.1. All computations have been performed in single precision (SP) arithmetics. Performance results have been averaged over three successive executions. The order of the numerical accuracy has been recorded for the first run, since the orders of the subsequent runs are similar.

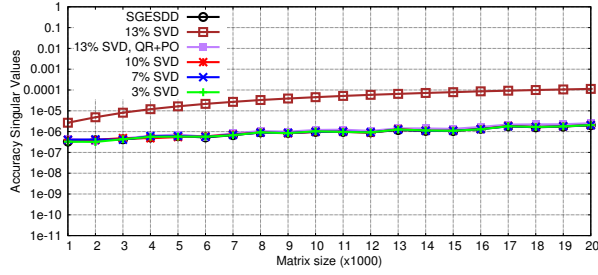
7.2 Definitions of Testing Matrices

We run the QDWH partial SVD against ill-conditioned synthetic matrices as well as matrices generated from real observational datasets. Since the QDWH partial SVD performance is sensitive to the matrix condition number due to more iterations required before convergence, ill-conditioned matrices represent the most challenging testcases for our implementation and are, therefore, the ones of interests in our herein experimental study. These matrices have been generated using the testing matrix generator SLATMS from LAPACK [5] from geometrically distributed singular values. For the testing matrices generated from real observational datasets, x and y were constructed from WFS measurements acquired on-sky on Sept 12, 2017 (UT). The WFS speed was set at $f = 2$ kHz, and $n = 1161$ modes were measured. Four data matrices A , representative of predictive control requirements, were constructed for computing $N = [1, 5, 10, 15]$ predictive filter steps from $t = 15$ second of WFS measurements. The ultimate matrix sizes are $Nn = [1161, 5805, 11610, 17415]$ by 29990, respectively.

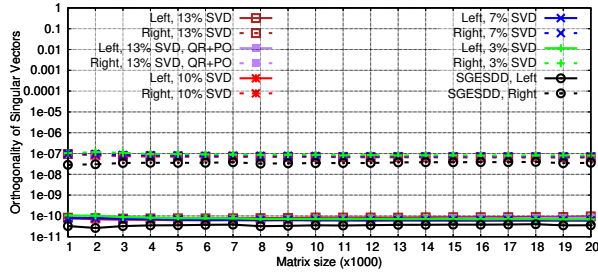
It will be also interesting to investigate advanced configurations with higher WFS frequencies and longer period of WFS measurements in the context of future Extremely Large Telescopes (ELT), which will require solving even larger problem sizes. Since GPU's main memory is a scarce resource, out-of-core algorithms (using CPU's main memory) and multiple GPUs may enable solving such large problems with additional data movements. These communications can be hidden by the large amount of available computations. Simulating envisioned hardware specifications of ELTs and software deployment on today's hardware are critical to identify algorithmic and implementation performance bottlenecks early on.

7.3 Numerical Accuracy

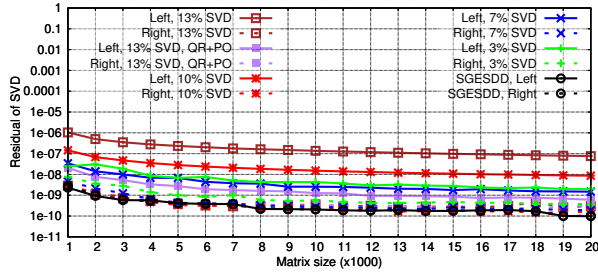
For the numerical accuracy, we look at three assessment metrics, which are typical for the SVD: the accuracy of the singular values, the orthogonality of the singular vectors and the backward stability [14], as described in Section 7.3 of [28]. Figures 3 show the numerical accuracy/robustness results of QDWH partial SVD, performed on the **K80** system, using synthetic ill-conditioned matrices. The threshold Li , introduced used in line 1 of Algorithm 4, is a tunable numerical parameter, which directly influences the number of requested singular values/vectors (i.e., 3%-7%-10%-13%) as well as the obtained accuracy/performance. Numerical results demonstrate the robustness of the QDWH partial SVD method, by getting accuracy results around the machine precision for SP computations. The accuracy starts to slightly deteriorate at 13%. As explained at the end of Section 5.3, the lost digits can be recovered by enforcing the first iteration of QDWH to rely on the QR factorization, which is much more robust numerically than the Cholesky factorization, as seen in the curves labelled "13% SVD, QR+PO" of Figures 3. If the threshold Li is even smaller so that a high percentage of singular values can be eventually selected, the



(a) Accuracy of singular values.



(b) Orthogonality of left/right singular vectors.



(c) Backward stability.

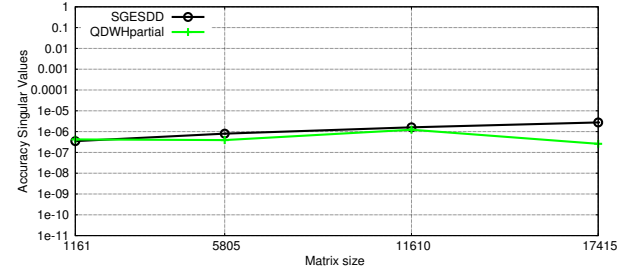
Figure 3: Assessing the numerical accuracy/robustness of QDWH partial SVD on K80 using synthetic ill-conditioned matrices. Some curves may overlap, due to same accuracy.

accuracy may further deteriorate without possible recovery, due to the increase of the condition number. The QDWH partial SVD method is typically robust when extracting less than 15%-20% of the overall spectrum. This threshold Li can be tuned with a priori knowledge on the singular value distribution.

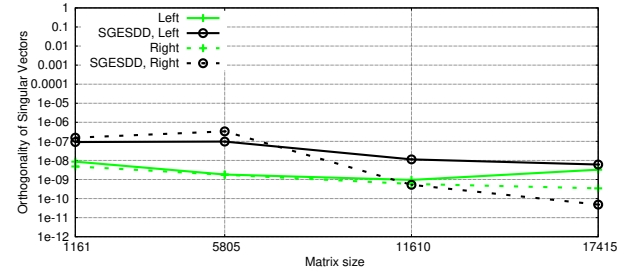
Figures 4 provide the numerical assessment using the same metrics on matrices generated from observational data. We set the threshold so that around 10% of the most significant singular values/vectors are retained in order to fulfill the SCE_{AO} specifications. The obtained accuracy results are at the level of machine precision for SP arithmetics, which highlight the robustness of QDWH partial SVD.

7.4 Performance Results

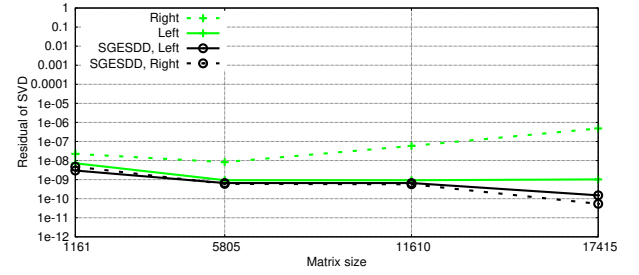
Figures 5 and 6 highlight the performance comparisons of QDWH partial SVD against SGESDD using two performance metrics: time to solution (in seconds) and floating-point operations per seconds



(a) Accuracy of singular values.



(b) Orthogonality of left/right singular vectors.



(c) Backward stability.

Figure 4: Assessing the numerical accuracy/robustness of QDWH partial SVD on K80 using real observational datasets.

(Gflop/s), respectively. While the former is commonly reported, the latter allows to assess the performance of the actual implementation on a given hardware. In particular, Figures 5a, 5b and 5c show the elapsed time in seconds using synthetic ill-conditioned matrices on the **K80**, **P100** and **V100** systems, respectively. Our QDWH partial SVD is much faster than SGESDD, achieving up to threefold, fourfold and fivefold performance speedups on aforementioned systems. When the required spectrum increases, QR-based QDWH iterations are favored to ensure proper numerical accuracy and engender time performance penalties, but still remain faster than SGESDD.

Figure 5d draws the elapsed time on real observational datasets for the matrix sizes described in Section 7.2. Our QDWH partial SVD outperforms SGESDD by more than fourfold performance speedup. Furthermore, Figures 6 show up to 1.8 Gflop/s, 7 Gflop/s and 9 Gflop/s performance, which corresponds to 45%, 75%, and 65% of

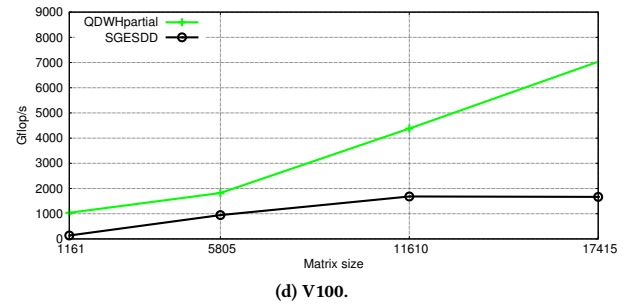
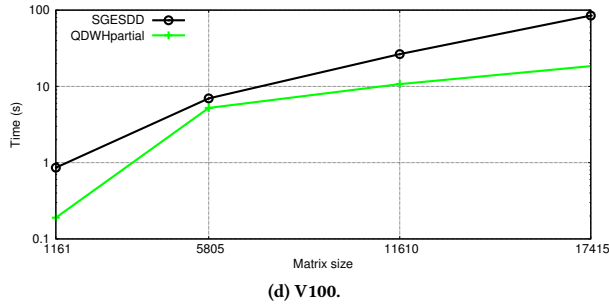
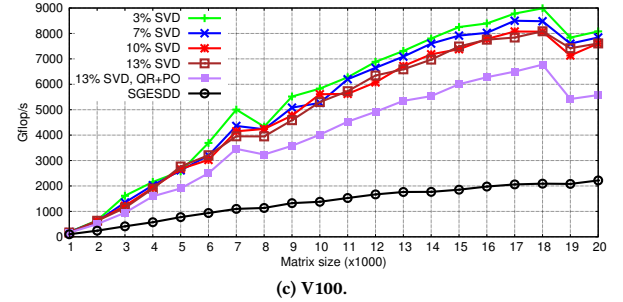
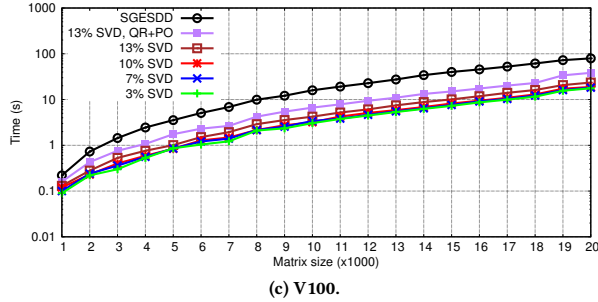
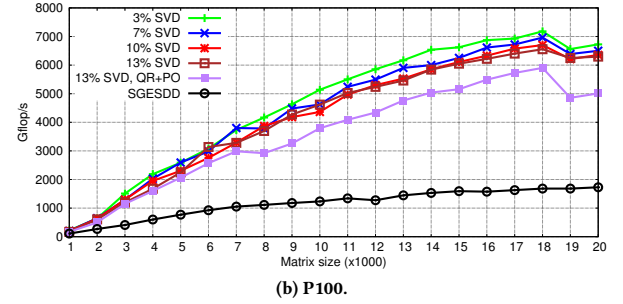
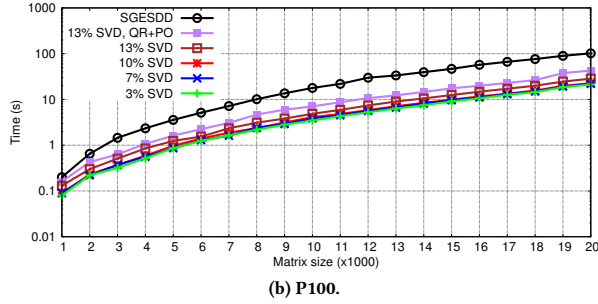
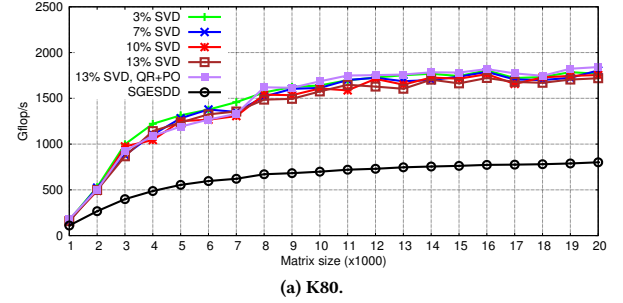
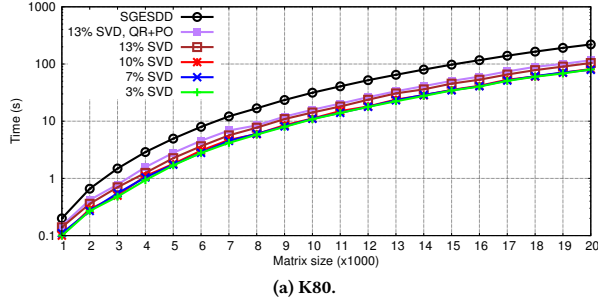


Figure 5: Performance comparisons in seconds of QDWH partial SVD using synthetic ill-conditioned matrices (a, b, c) and real observational datasets (d).

Figure 6: Performance comparisons in Gflops/s of QDWH partial SVD using synthetic ill-conditioned matrices (a, b, c) and real observational datasets (d).

the theoretical performance peak on the **K80**, **P100** and **V100** systems, respectively. Although the sustained peak performance is usually a more representative metric, the current performance assessment shows that there is still room for improvement, although our implementation is able to take advantage of the underlying

hardware resources much more than SGESDD. In fact, the performance gap between QDWH partial SVD and SGESDD widens, as the hardware technology scaling improves. This indicates that our algorithm is capable of extracting performance, and thanks to its

inherent compute-bound kernels, benefits from hardware with over-provisioned floating-point units, which will populate future supercomputers.

8 CONCLUSION AND FUTURE WORK

We have implemented a novel QDWH partial SVD algorithm to compute the pseudo-inverse of a dense matrix. We have demonstrated its numerical robustness and performance efficiency on various single GPU hardware generations with synthetic matrices and matrices generated from real observational datasets, in the context of the Subaru Coronagraphic Extreme AO (SCEAO) system [19], deployed on the 8.3m diameter Subaru Telescope. Our implementation outperforms the state-of-the-art SVD implementation SGESDD based on the divide and conquer solver from MAGMA by up to fivefold and fourfold performance speedups on synthetic ill-conditioned matrices and real observational datasets, respectively. The adaptive optics predictive control and sensor fusion approaches enabled by faster SVDs will also allow ultra-sharp visible light imaging with large telescope, and will extend the high performance Extreme AO correction to extend to fainter stars.

Future works include extension to multiple GPUs using tile algorithms associated with a dynamic runtime system to schedule various computational tasks and mitigate the data movement overhead on the underlying hardware resources. We would like also to investigate the feasibility of using least square methods as a competitive approach to the pseudo-inverse algorithm. This would require implementing the QR-based least square linear solver with an updating mechanism to prevent recomputing the entire factorization, while keeping up with the cadence of the optics hardware. Last but not least, the optical aberrations at the telescope, due to the atmospheric turbulences, may be predicted using a statistical approach instead, based on a supervised machine learning method by estimating the maximum likelihood [3].

REFERENCES

- [1] [n. d.]. The Extrasolar Planets Encyclopaedia. <http://exoplanet.eu/>. ([n. d.]). Accessed: 2017-12-26.
- [2] A. Abdelfattah, E. Gendron, D. Gratadour, D. E. Keyes, H. Ltaief, A. Sevin, and F. Vidal. 2014. High Performance Pseudo-analytical Simulation of Multi-Object Adaptive Optics over Multi-GPU Systems. In *Euro-Par 2014 Parallel Processing*, Fernando Silva, Inês Dutra, and Vitor Santos Costa (Eds.). Lecture Notes in Computer Science, Vol. 8632. Springer International Publishing, 704–715. https://doi.org/10.1007/978-3-319-09873-9_59
- [3] S. Abdullah, H. Ltaief, Y. Sun, M. Genton, and D. E. Keyes. 2017. ExaGeoStat: A High Performance Unified Framework for Geostatistics on Manycore Systems. *Submitted to the IEEE Trans. Parallel Distrib. Syst.* (2017).
- [4] E. Agullo, J. Demmel, J. Dongarra, B. Hadri, J. Kurzak, J. Langou, H. Ltaief, P. Luszczek, and S. Tomov. 2009. Numerical Linear Algebra on Emerging Architectures: The PLASMA and MAGMA Projects. *Journal of Physics: Conference Series* 180 (2009).
- [5] E. Anderson, Z. Bai, C. Bischof, S. L. Blackford, J. W. Demmel, J. J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. C. Sorensen. 1999. *LAPACK User's Guide* (3 ed.). Society for Industrial and Applied Mathematics, Philadelphia.
- [6] G. Anglada-Escudé, P. J. Amado, J. Barnes, Z. M. Berdiñas, R. P. Butler, G. A. L. Coleman, I. de la Cueva, S. Dreizler, M. Endl, B. Giesers, et al. 2016. A Terrestrial Planet Candidate in a Temperate Orbit Around Proxima Centauri. *Nature* 536, 7617 (2016), 437–440.
- [7] T. Auckenthaler, V. Blum, H. J. Bungartz, T. Huckle, R. Johanni, L. Krämer, B. Lang, H. Lederer, and P. R. Willems. 2011. Parallel Solution of Partial Symmetric Eigenvalue Problems from Electronic Structure Calculations. *Parallel Comput.* 37, 12 (2011), 783–794. <http://dx.doi.org/10.1016/j.parco.2011.05.002>
- [8] H. W. Babcock. 1953. The Possibility of Compensating Astronomical Seeing. *Publications of the Astronomical Society of the Pacific* 65, 386 (1953), 229–236.
- [9] J. M. Beckers. 1993. Adaptive Optics for Astronomy: Principles, Performance, and Applications. *Annual review of astronomy and astrophysics* 31, 1 (1993), 13–62.
- [10] C. H. Bischof, B. Lang, and X. Sun. 2000. Algorithm 807: The SBR Toolbox—Software for Successive Band Reduction. *ACM Trans. Math. Software* 26, 4 (2000), 602–616. <https://doi.org/10.1145/365723.365736>
- [11] A. Charara, H. Ltaief, D. Gratadour, D. E. Keyes, A. Sevin, A. Abdelfattah, E. Gendron, C. Morel, and F. Vidal. 2014. Pipelining Computational Stages of the Tomographic Reconstructor for Multi-Object Adaptive Optics on a Multi-GPU System. In *International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2014, New Orleans, LA, USA, November 16-21, 2014*, Trish Damkroger and Jack J. Dongarra (Eds.). IEEE Computer Society, 262–273. <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7012142>; <http://dl.acm.org/citation.cfm?id=2683593>
- [12] É. Gendron, A. Charara, A. Abdelfattah, D. Gratadour, D. E. Keyes, H. Ltaief, C. Morel, F. Vidal, A. Sevin, and G. Rousset. 2014. A Novel Fast and Accurate Pseudo-Analytical Simulation Approach for MOAO. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Vol. 9148. Article 91486L, 6 pages. <https://doi.org/10.1117/12.2055911>
- [13] M. Gillon, A. H.M.J. Triaud, B.-O. Demory, E. Jehin, E. Agol, K. M. Deck, S. M. Lederer, J. De Wit, A. Burdanov, J. G. Ingalls, et al. 2017. Seven Temperate Terrestrial Planets Around the Nearby Ultracool Dwarf Star TRAPPIST-1. *Nature* 542, 7642 (2017), 456–460.
- [14] G. H. Golub and C. F. Van Loan. 1996. *Matrix Computations* (third ed.). Johns Hopkins University Press, Baltimore, Maryland.
- [15] O. Guyon. 2005. Limits of Adaptive Optics for High-Contrast Imaging. *The Astrophysical Journal* 629, 1 (2005), 592.
- [16] A. Haidar, H. Ltaief, and J. J. Dongarra. 2011. Parallel Reduction to Condensed Forms for Symmetric Eigenvalue Problems Using Aggregated Fine-Grained and Memory-Aware Kernels. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC '11)*. ACM, New York, NY, USA, Article 8, 11 pages. <https://doi.org/10.1145/2063384.2063394>
- [17] A. Haidar, H. Ltaief, P. Luszczek, and J. Dongarra. 2012. A Comprehensive Study of Task Coalescing for Selecting Parallelism Granularity in a Two-Stage Bidiagonal Reduction. In *IEEE 26th International Parallel and Distributed Processing Symposium*. 25–35. <https://doi.org/10.1109/IPDPS.2012.13>
- [18] J. W. Hardy. 1998. *Adaptive Optics for Astronomical Telescopes*. Vol. 16. Oxford University Press on Demand.
- [19] N. Jovanovic, F. Martinache, O. Guyon, C. Clergeon, G. Singh, T. Kudo, V. Garrel, K. Newman, D. Doughty, J. Lozi, J. Males, Y. Minowa, Y. Hayano, N. Takato, J. Morino, J. Kuhn, E. Serabyn, B. Norris, P. Tuthill, G. Schworer, P. Stewart, L. Close, E. Huby, G. Perrin, S. Lacour, L. Gauchet, S. Vievard, N. Murakami, F. Oshiyama, N. Baba, T. Matsuo, J. Nishikawa, M. Tamura, O. Lai, F. Marchis, G. Duchene, T. Kotani, and J. Woillez. 2015. The Subaru Coronagraphic Extreme Adaptive Optics System: Enabling High-Contrast Imaging on Solar-System Scales. *PASP* 127 (Sept. 2015), 890. <https://doi.org/10.1086/682989> arXiv:astro-ph.IM/1507.00017
- [20] A-M Lagrange, M Bonnefoy, G Chauvin, D Apai, D Ehrenreich, A Boccaletti, D Gratadour, D Rouan, D Mouillet, S Lacour, et al. 2010. A giant planet imaged in the disk of the young star β Pictoris. *Science* 329, 5987 (2010), 57–59.
- [21] H. Ltaief, D. Gratadour, A. Charara, and E. Gendron. 2016. Adaptive Optics Simulation for the World's Largest Telescope on Multicore Architectures with Multiple GPUs. In *Proceedings of the Platform for Advanced Scientific Computing Conference (PASC '16)*. ACM, New York, NY, USA, Article 9, 12 pages. <https://doi.org/10.1145/2929908.2929920>
- [22] H. Ltaief, P. Luszczek, and J. J. Dongarra. 2012. *Enhancing Parallelism of Tile Bidiagonal Transformation on Multicore Architectures Using Tree Reduction*. Springer Berlin Heidelberg, Berlin, Heidelberg, 661–670. https://doi.org/10.1007/978-3-642-31464-3_67
- [23] P. Luszczek, H. Ltaief, and J. J. Dongarra. 2011. Two-Stage Tridiagonal Reduction for Dense Symmetric Matrices Using Tile Algorithms on Multicore Architectures. In *Proc. 25th IEEE International Symposium on Parallel and Distributed Processing (25th IPDPS'11)*. IEEE Computer Society, Anchorage, Alaska, USA, 944–955.
- [24] C. Marois, B. Macintosh, T. Barman, B. Zuckerman, I. Song, J. Patience, D. Lafrenière, and R. Doyon. 2008. Direct Imaging of Multiple Planets Orbiting the Star HR 8799. *Science* 322, 5906 (2008), 1348–1352.
- [25] Y. Nakatsukasa and N. J. Higham. 2013. Stable and Efficient Spectral Divide and Conquer Algorithms for the Symmetric Eigenvalue Decomposition and the SVD. *SIAM J. Scientific Computing* 35, 3 (2013). <http://dx.doi.org/10.1137/120876605>
- [26] NVIDIA. 2018. The NVIDIA CUDA Basic Linear Algebra Subroutines. <https://developer.nvidia.com/cublas/>. (2018).
- [27] F. Roddier. 1999. *Adaptive Optics in Astronomy*. Cambridge university press.
- [28] D. Sukkari, H. Ltaief, and D. E. Keyes. 2016. A High Performance QDWH-SVD Solver Using Hardware Accelerators. *ACM Trans. Math. Softw* 43, 1 (2016), 6:1–6:25. <http://doi.acm.org/10.1145/2894747>
- [29] R. K. Tyson. 2015. *Principles of Adaptive Optics*. CRC press.