Project Description

1 Overview

Scientists in a wide range of fields are increasingly making use of randomized Krylov subspace methods (KSMs) for computing quantities involving matrix functions. Applications of such algorithms include (but are certainly not limited to) studying properties of quantum materials in physics and chemistry [EFLSV02; WWAF06; GPS15; GCR15; LSY16; CMFCLK17; CGMLFR18; VFAPP20; JWWLMDR21; CC22], probing the behavior of neural networks in machine learning [GKX19; Pap19; GWG19; YGKM20], finding triangle counts and other graph motifs in network science [Avr10; DBB19; BB20], designing better public transit in urban planning [BS22; WSMB21], and load balancing modern parallel eigensolvers in high performance computing [Pol09; LXES19].

However, despite the growing prominence of randomized Krylov subspace methods for tasks involving matrix functions, there are a number of theoretical and practical hurdles which are preventing such algorithms from being used to their full potential. These can be roughly categorized as arising from one or both of the following larger barriers:

Barrier 1. Lack of sufficient cross-disciplinary knowledge transfer between those who study these algorithms and those who use these algorithms

Barrier 2. Limited theoretical understanding of the behavior of (randomized) KSMs in modern computing environments

While neither barrier precludes the use of randomized KSMs for matrix functions in application areas, each barrier represents a concrete roadblock in the practical effectiveness of such methods. Thus, eliminating, or at least reducing, these barriers will have a significant impact on the advancement of computational science.

This grant proposal, titled *Randomized Krylov subspace methods for matrix functions*, describes a research program aimed at reducing the aforementioned barriers via the following objectives:

Objective 1. Develop randomized methods for low-rank approximation of matrix functions, traces and partial traces of matrix functions, and spectral density estimates which work in practice.

Objective 2. Prove theoretical guarantees about existing algorithms and the above-mentioned algorithms, with an emphasis on a posteriori bounds and stopping criteria.

Objective 3. Facilitate the use of such algorithms and bounds in application areas.

Mentorship of undergraduate researchers is a key aspect of this proposal, and a number of the research directions described are designed to address the objectives above while simultaneously being suitable for the inclusion of undergraduate researchers.

1.1 Organization of proposal

The proposal is centered around three research topics: (i) low-rank approximation, (ii) trace and spectrum approximation, and (iii) partial trace approximation. For each topic, we provide context for the current state of the art as well as describe how Barriers 1 and 2 listed above are limiting further progress. We then provide details on specific research directions which align with Objectives 1, 2, and 3 with the aims of reducing the aforementioned barriers. Finally , the broader impacts and intellectual merit of the overall proposal are addressed, as is the plan for supporting the training and development of undergraduate students, particularly female students and students from racial minority groups underrepresented in computational mathematics in the United States.

2 Background

For clarity of exposition, throughout this proposal $\mathbf{H}$ will be a $n \times n$ Hermitian matrix. Thus, $\mathbf{H}$ admits an eigenvalue decomposition

$$\mathbf{H} = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^*, \tag{1}$$

where $\Lambda := \{\lambda_i\}_{i=1}^n$ are the eigenvalues of and $\{\mathbf{u}_i\}_{i=1}^n$ are the (orthonormal) eigenvectors. Typically, $\mathbf{H}$ will be sparse and of very large dimension (n on the order of millions for computations on laptops/desktops, and n or the order of billions or trillions for computations on clusters and supercomputers).

A matrix function transforms the eigenvalues of a Hermitian matrix according to some scalar function, while leaving the eigenvectors untouched. Specifically, the matrix function $f(\mathbf{H})$, induced by $f : \mathbb{R} \to \mathbb{R}$ and $\mathbf{H}$, is defined as

$$f(\mathbf{H}) := \sum_{i=1}^{n} f(\lambda_i) \mathbf{u}_i \mathbf{u}_i^*. \tag{2}$$

Perhaps the most well known example of a matrix function is the matrix inverse $\mathbf{H}^{-1}$, which corresponds to the inverse function $f(x) = x^{-1}$. Other common matrix functions including the matrix sign, logarithm, exponential, square root, and inverse square root, each of which has many applications throughout the mathematical sciences [Hig08]. Throughout, we will use $\mathbf{A}$ to represent a $n \times n$ matrix of interest, and typically $\mathbf{A} := f(\mathbf{H})$.

It is important to note that even if we have $\mathbf{H}$ in hand, in all of the applications of interest to this proposal, it is prohibitively expensive to compute $f(\mathbf{H})$ itself. In fact, while $\mathbf{H}$ may be sparse, $f(\mathbf{H})$ is typically is dense and therefore too large to store, even if it were somehow possible to compute! As such, avoiding forming $f(\mathbf{H})$ explicitly is a key aspect of the algorithms related to this proposal. As a concrete example, when $n = 2^{20} \approx 10^6$, storing a dense $n \times n$ matrix of 64 bit floating point numbers would require over *8.7 terabytes* of memory! While this is too large even for many desktop computers, the algorithms discussed in this proposal can easily handle even larger problems on laptops.

2.1 Matrix-free algorithms

Matrix-free algorithms access a matrix of interest only through matrix-vector products (as opposed to having random access to the entries of the matrix). While random access can be simulated through matrix-vector products, using matrix-vector products to access a single entry is inefficient. For many linear-algebraic tasks involving a generic matrix $\mathbf{A}$ (e.g. low-rank approximation or trace estimation), there are well-known matrix-free algorithms which are highly successful in practice. In the particular case $\mathbf{A} = f(\mathbf{H})$, matrix-vector products with $\mathbf{A}$ are commonly approximated by Krylov subspace methods in a black-box manner. This is described in the next section. A major theme of this proposal is looking into this black-box in order to seek out efficiencies that would otherwise be missed.

2.2 Krylov subspace methods

Given a vector $\mathbf{v}$, Krylov subspace methods for matrix functions build an approximation to $f(\mathbf{H})\mathbf{v}$ or $\mathbf{v}^* f(\mathbf{H})\mathbf{v}$ from the Krylov subspace $\mathcal{K}_{q+1}(\mathbf{H}, \mathbf{v}) = \text{span}\{\mathbf{v}, \mathbf{H}\mathbf{v}, \ldots, \mathbf{H}^q\mathbf{v}\}$. This can be done in a matrix-free manner using just q matrix-vector products with $\mathbf{H}$. When $\mathbf{H}$ is symmetric, the Lanczos algorithm produces an orthonormal basis $\mathbf{Q}$ for Krylov subspace and a tridiagonal matrix $\mathbf{T}$ containing the coefficients for a three-term recurrence satisfied by the basis vectors. These matrices can then be used to approximate quantities such as $f(\mathbf{H})\mathbf{v}$ and $\mathbf{v}^* f(\mathbf{H})\mathbf{v}$ *without ever forming* $f(\mathbf{H})$ *itself*. The most common algorithm for doing so is called the Lanczos method for matrix function approximation (Lanczos-FA) [DK89; Saa92].

In some settings it is desirable to approximate $f(\mathbf{H})\mathbf{V}$ or $\mathbf{V}^* f(\mathbf{H})\mathbf{V}$, where $\mathbf{V}$ is a matrix. This can clearly be done using by approximating $f(\mathbf{H})\mathbf{v}_i$ for each column $\mathbf{v}_i$ of $\mathbf{V}$. However, there are more efficient block

algorithms which work over the block Krylov subspace $\mathcal{K}_{q+1}(\mathbf{H}, \mathbf{V}) = \text{span}\{\mathbf{V}, \mathbf{HV}, \ldots, \mathbf{H}^q\mathbf{V}\}$, where the span is over the columns of the constituent matrices. The analogous algorithm to Lanczos-FA, based on the block-Lanczos algorithm is called block-Lanczos-FA.

We remark that when $\mathbf{H}$ is non-Hermitian, there is a recent line of work on randomizing Krylov subspace methods for approximating $f(\mathbf{H})\mathbf{v}$ to reduce orthogonalization costs. These include modifying methods for linear systems such as GMRES and FOM [NT21; BG22; TGB23] and methods for general matrix functions like Arnoldi-FA [CKN22; GS22]. This proposal focuses on primarily on the Hermitian case as well as the use of randomness at a much higher level. It is therefore distinct (yet complimentary) to such methods.

3 Low rank approximation

The first topic of focus is low-rank approximation of matrix functions. Given a $n \times n$ matrix $\mathbf{A}$ and a positive integer k, the task of low-rank approximation is to find a rank k matrix $\tilde{\mathbf{A}}$ so that $\tilde{\mathbf{A}} \approx \mathbf{A}$. When the error is measured in any unitarily invariant norm (e.g. the spectral norm or Frobenius norm), the optimal solution to this problem is the truncated rank-k singular value decomposition (SVD). This proposal targets the case $\mathbf{A} = f(\mathbf{H})$.

3.1 Existing sketching-based algorithms

Computing the SVD of $\mathbf{A}$ is expensive (even if we know $\mathbf{A}$). A number of randomized algorithms based on a technique called *sketching* have been used to great success [HMT11; MT20a; TW23]. At a high level, sketching involves efficiently obtaining a simplified $n \times \ell$ matrix $\mathbf{Y}$ containing key information about $\mathbf{A}$ (analogous to how a sketch is a simplified representation of a scene). Using the information in $\mathbf{Y}$, a $n \times k$ orthonormal matrix $\mathbf{Q}$ (ideally aligned with the dominant subspace of $\mathbf{A}$) is obtained. Low-rank approximations such as $\tilde{\mathbf{A}} = \mathbf{Q}(\mathbf{Q}^*\mathbf{A})$ or $\tilde{\mathbf{A}} = \mathbf{Q}(\mathbf{Q}^*\mathbf{AQ})\mathbf{Q}^*$ can then be obtained [HMT11; MT20a; TW23].

The choice of $\mathbf{Y}$ is clearly important. The most basic choice is $\mathbf{Y} = \mathbf{AS}$, where $\mathbf{S}$ is a $n \times \ell$ random matrix whose entries are chosen from a suitable distribution. This is computationally efficient as $\mathbf{Y}$ can be computed using just ℓ matrix-vector products. That such an approach might work is intuitive: when we multiply $\mathbf{A}$ with a random matrix $\mathbf{S}$, then $\mathbf{Y} = \mathbf{AS}$ is most aligned with the singular vectors of $\mathbf{A}$ corresponding to the largest singular values. Of course, the decay of the singular values impacts the quality of the resulting low-rank approximation, and slow decay can result in poor approximations. To alleviate slow decay, it is often suggested to use the sketch $\mathbf{Y} = \mathbf{A}^q\mathbf{S}$. This approach, called *subspace iteration*, works because the singular values of $\mathbf{A}^q$ decay more quickly than those of $\mathbf{A}$. Such an approach now requires $q\ell$ matrix-vector products with $\mathbf{A}$. One can instead use the sketching matrix $\mathbf{Y} = [\mathbf{S}, \mathbf{AS}, \ldots, \mathbf{A}^q\mathbf{S}]$ which can also be computed using $q\ell$ matrix-vector products but is much larger and has provable and easily observable benefits [MM15; TW23]. This is called *block Krylov iteration*.

3.2 Barriers

The above approaches access $\mathbf{A}$ using just matrix-vector products. When $\mathbf{A} = f(\mathbf{H})$, products $\mathbf{AS} = f(\mathbf{H})\mathbf{S}$ are commonly approximated via a Krylov subspace method. In this case, the natural primitive is matrix-vector products with $\mathbf{H}$, rather than $f(\mathbf{H})$. Despite its obvious importance, this setting is under-explored, and is ripe for practical and theoretical improvements. This is arguably due to two factors. First, one of the largest applications of low-rank approximation of matrix functions is to trace estimation (see section 4), and the use of low-rank approximation in trace estimation was not popularized until recently. Second, the design and analysis of KSMs for matrix functions has historically been pursued by classical numerical linear algebrists, and those interested in randomized methods are less familiar with methods for matrix functions.

3.3 Proposed approaches

RD 1: Theoretical analysis of Krylov-aware algorithms (O1, O2). When $\mathbf{A} = f(\mathbf{H})$, rather than apply $\mathbf{A}$ to $\mathbf{S}$ using a black-box KSMs, efficiencies can be gained by taking a careful look into the black box. While conceptually simple, to the best of our knowledge, this observation has only started to be explored in earnest in recent years [CH23, PK22, PCM23].

In [PK22], it is proven that for operator monotone functions (e.g. $f(x) = \sqrt{x}$) it makes sense to use a sketch $\mathbf{Y} = \mathbf{HS}$ rather than $\mathbf{Y} = f(\mathbf{H})\mathbf{S}$, even if the costs of computing the two sketches are equal. Intuitively, this is because the the singular values of $\mathbf{H}$ decay more quickly than those of $\mathbf{A} = f(\mathbf{H})$ and therefore more information about the dominant eigenspace of $\mathbf{A}$ (which is the same as the dominant eigenspace of $\mathbf{H}$) is obtained from the sketch involving $\mathbf{H}$.

An independent yet complimentary observation was made by the PI in [CH23] in which a so-called "Krylov-aware" method for low-rank approximation is introduced. In particular, in order to approximate $f(\mathbf{H})\mathbf{S}$ via a Krylov subspace method, the Krylov subspace $\mathcal{K}_{q+1}(\mathbf{H}, \mathbf{S}) = \text{span}\{\mathbf{S}, \mathbf{HS}, \ldots, \mathbf{H}^q\mathbf{S}\}$ (for some $q > 0$) is constructed. Thus, rather than using a sketch $\mathbf{Y} \approx f(\mathbf{H})\mathbf{S}$ with ℓ columns, a sketch $\mathbf{Y} = [\mathbf{S}, \mathbf{HS}, \ldots, \mathbf{H}^q\mathbf{S}]$ with $(q+1)\ell$ columns can be obtained for the same number of matrix-vector products with $\mathbf{H}$. The latter sketch contains strictly more information than the former. Many low-rank approximation algorithms then require $f(\mathbf{H})\mathbf{Y}$ to be computed, and such a product can be approximated from $\mathcal{K}_p(\mathbf{H}, \mathbf{Y})$ (for some $p > 0$). If $\mathbf{Y}$ has more columns, computing $\mathcal{K}_k(\mathbf{H}, \mathbf{Y})$ is ostensibly more expensive if $\mathbf{Y} = [\mathbf{S}, \mathbf{HS}, \ldots, \mathbf{H}^q\mathbf{S}]$ than if $\mathbf{Y} \approx f(\mathbf{H})\mathbf{S}$. However, in the special case $\mathbf{Y} = [\mathbf{S}, \mathbf{HS}, \ldots, \mathbf{H}^q\mathbf{S}]$, it is not hard to see that $\mathcal{K}_p(\mathbf{H}, \mathbf{Y}) = \mathcal{K}_{p+q}(\mathbf{S})$. In other words, $f(\mathbf{H})\mathbf{Y}$ can be approximated from $\mathcal{K}_p(\mathbf{H}, \mathbf{Y})$ using the same number of products regardless of which of the sketches was originally used.

There are a number of concrete theoretical and practical concerns with such a method. First, we might address the question: *how should parameters p and q be set?* This was not studied in detail in [CH23], as the method for low-rank approximation was used as a subroutine in a larger trace estimation algorithm in which sufficient lower bounds for p and q were essentially known. In subsequent work, the PI and collaborators derived theoretical guarantees for the output of Krylov-aware algorithms in terms of the best low-rank approximation [PCM23]. However, the resulting bounds are stated in terms of certain polynomial minimization problems which are not easily solved explicitly. Deriving more explicit bounds would provide a clearer picture about the precise situations in which Krylov-aware methods provide the most bennefit.

Second, while theoretical a priori error bounds provide information about how well algorithms work on various types of inputs, but are typically not very useful for use as stopping criteria. Thus, it is also reasonable to ask: *can we derive practical a posteriori bounds suitable for use as stopping critera?* The PI has developed a posteriori bounds for Lanczos-based methods for matrix functions [CTU21, CGMM22, CC22], but such approaches do not consider the aspect of randomization. To address this, approaches to a posteriori error estimates for low-rank approximation [ET22, TW23] must also be incorporated. When the two are combined, potential for further efficiencies should be explored.

RD 2: Memory efficient algorithms (O1,O2). A potential limitation of the Krylov-aware algorithm suggested in [CH23] is the large memory required to store a large sketch $\mathbf{Y} = [\mathbf{S}, \mathbf{HS}, \ldots, \mathbf{H}^q\mathbf{S}]$. There are a number of potential approaches to addressing this issue. It is reasonable to maintain a basis for some smaller subspace of the Krylov subspace $\mathcal{K}_{q+1}(\mathbf{H}, \mathbf{S})$, and such an approach was studied breifly in [CH23]. This naturally raises the question of how to choose such a subspace. In particular, it is desirable that the sketching space aligns with the top subspace of $\mathbf{A} = f(\mathbf{H})$. Algorithms such as Implicitly Restarted (Block) Arnoldi/Lanczos serve as a natural starting point for obtaining such a subspace [Sor92, CRS94, LS96, WS00, Leh01]. The convergence of such methods depends on the choice of *acceleration polynomial*, but we are unaware of theoretical guarantees similar in strength to those for low-rank approximation. We will study whether there exist simple choices of acceleration polynomial, reasonably related to the choices used in practice, which have provable convergence guarantees.

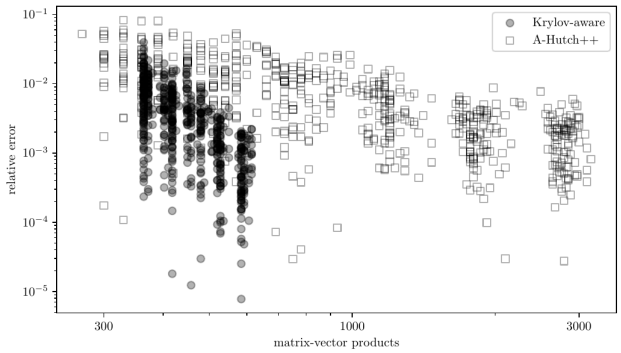

Figure 1: Performance of the Krylov-aware algorithm in [CH23] compared to the previous state-of-the-art trace estimation algorithm for approximating $\mathrm{tr}(\exp(\mathbf{A}))$ (Estrada index) for the Roget's Thesaurus graph. The PI's algorithm attains an order of magnitude reduction in costs. This proposal would support the development of similarly powerful randomized algorithms for key linear algebra tasks.

There are complimentary open questions relating to memory efficient algorithms, even for the most simple case $\mathbf{A} = \mathbf{H}$. For instance, while possible in theory using a two-pass approach [Bor00, EFLSV02, FS08a], it is not clear how to stably implement block Krylov iteration [MM15, TW23] without storing the entire block Krylov subspace in the presence of non-deterministic computation (as is typical on supercomputers [ICDG15]). Addressing this question by developing a *practical and robust* algorithm would lead to state of the art methods for low-rank approximation. Addressing this problem, and problem for general $\mathbf{A} = f(\mathbf{H})$ are complimentary, and neither appears strictly easier than the other. Thus, they will be pursued in parallel.

RD 3: Stability of Sketching under inexact products (O1, O2). It is also interesting to consider how classical sketching algorithms behave in the presence of *large* noise. Suppose, for some $\epsilon > 0$, we have a black-box which takes in an arbitrary vector $\mathbf{x}$ and outputs a vector $\mathbf{y}$ satisfying $\|\mathbf{A}\mathbf{x} - \mathbf{y}\| \leq \epsilon\|\mathbf{A}\|\|\mathbf{x}\|$. Such a guarantee is standard for important settings such as errors due to floating point roundoff or due to inexact applications of $\mathbf{A} = f(\mathbf{H})$ via Krylov susbpace methods.

In the past several years, a number of works have addressed the questions *how does nonzero ϵ impact sketching algorithms?* Algorithm analyzed include variants of the Nyströmm method [Nak20, CD22] and the randomized SVD and subspace iteration [CHP22]. These analyses focus mostly on small ϵ associated with rounding errors ($\epsilon \approx 10^{-16}$ for double precision to $\epsilon = 10^{-4}$ for single precision). We will focus on analyses which target large ϵ of similar magnitude to the desired accuracy of the low-rank approximation.

For some applications, it is important to understand the matrix $\mathbf{Q}$ itself. In [CH23] the PI and collaborator obtained a bound for the difference between the orthogonal projectors onto the column spans of $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$, where $\mathbf{Y} = \mathbf{AS}$ and $\tilde{\mathbf{Y}} = \mathbf{AS} + \mathbf{E}$, where $\mathbf{E}$ accounts for inexact matrix-vector products satisfying a guarantee like that stated above. However, the resulting bound was unsatisfying in that it depended on the condition number of $\mathbf{A}$ which can be large or even infinite even in cases where sketching (with inexact products) is observed to work well. A natural target would be bounds depending on the condition number of the best rank k approximation to $\mathbf{A}$ or on the gap between the k and $(k+1)$-st singular values of $\mathbf{A}$.

It also makes sense to ask: *can we design algorithms which are more robust to inexact products than existing algorithms?* In particular, can we design algorithms which take advantage of our knowledge about how error arises in different settings (e.g. as due to roundoff, due to discretization error in a PDE solver, due to statistical noise, etc.)? Similarly, we may consider sketching algorithms for settings in which ϵ is only large because a small number of the entries of the matrix-vector product are corrupted; e.g. due to soft faults

on a high performance machine [HH11; AGGRZ13].

4 Trace and spectrum estimation

The aim of trace approximation is to approximate the trace $\operatorname{tr}(\mathbf{A}) = \sum_i A_{i,i}$ of a square matrix $\mathbf{A}$, where $A_{i,j}$ is the (i,j)-entry of $\mathbf{A}$. If $\mathbf{A}$ is known explicitly this task is trivial, but in other situations, for instance $\mathbf{A} = f(\mathbf{H})$, the task is more difficult.

4.1 Existing trace and spectrum algorithms and bounds

Matrix-free trace estimation. A large amount of past work has been done in the setting where $\mathbf{A}$ can only be accessed via matrix-vector products. In situations where some error is tolerable, randomization can be introduced to significantly decrease the number of matrix-vector products required [Gir87; Ski89; Hut89; MMMW21]. The simplest estimator is the quadratic trace estimator,

$$\mathbf{v}^*(\mathbf{A}\mathbf{v}), \qquad \mathbf{v} \text{ has independent standard normal entries.}^{1} \tag{3}$$

This estimator is unbiased and has variance $2\|\mathbf{A}\|_{\mathsf{F}}^2$. The variance can be improved to $2\|\mathbf{A}\|_{\mathsf{F}}^2/m$ by averaging m independent copies of $\mathbf{v}^*(\mathbf{A}\mathbf{v})$, so the number of copies (and therefore matrix-vector products with $\mathbf{A}$) needed scales as $m = O(\epsilon^{-2})$, where ϵ is the target accuracy.

If $\tilde{\mathbf{A}}$ is a low-rank approximation to $\mathbf{A}$, the exact trace of $\tilde{\mathbf{A}}$ can be computed cheaply and the estimator eq. (3) applied to the remainder $\mathbf{A} - \tilde{\mathbf{A}}$ [Gir87; WWAF06; WLKSG16; GSO17; Lin16; MT20b; MMMW21; PK22; ETW23]. Assuming $\|\mathbf{A} - \tilde{\mathbf{A}}\|_{\mathsf{F}}^2 \leq \|\mathbf{A}\|_{\mathsf{F}}^2$, this approach has a reduced variance. If sketching used to compute the low-rank approximation $\tilde{\mathbf{A}}$, it is possible to improve the number of matrix-vector products with $\mathbf{A}$ to $O(\epsilon^{-1})$ [MMMW21]. Perhaps the most well-known algorithm for doing so is called Hutch++ [MMMW21], and a number of practical improvements to Hutch++ enjoying the same theoretical guarantee have been studied [PK22; ETW23].

Spectral sums and spectrum approximation. Many of the applications of implicit trace estimation algorithms involved estimating $\operatorname{tr}(f(\mathbf{H}))$, which is often called a *spectral sum*. Merging implicit trace estimation and KSMs for approximating $f(\mathbf{H})\mathbf{v}$ immediately yields algorithms for approximating spectral sums, and such algorithms have been studied for nearly as long as implicit trace estimation [Ski89; BFG96; MT20a].

The *spectral density* (also called the density of states in Physics) for $\mathbf{H}$ is defined by

$$\rho(x) = \frac{1}{n}\sum_{i=1}^{n} \delta_{\lambda_i}(x), \tag{4}$$

where δ_λ is a Dirac delta mass located at λ. It is important to note that the spectral sum $\operatorname{tr}(f(\mathbf{H}))$ can be written

$$\operatorname{tr}(f(\mathbf{H})) = \sum_{i=1}^{n} f(\lambda_i) = \int f(x)\rho(x)\mathrm{d}x. \tag{5}$$

Thus, algorithms for approximating the spectral density immediately give algorithms for approximating the trace of a matrix function. As the spectral density quantity encodes all of the eigenvalues of $\mathbf{H}$ it cannot be computed exactly without obtaining all of the eigenvalues of $\mathbf{H}$ (expensive!). However, often the overall "shape" of the spectral density is what is actually important in many cases, so a natural goal is to obtain a coarse approximation to $\rho(x)$. If this approximate accurately integrates functions, then it can be used to approximate spectral sums.

[1] Other common choices of distribution include independent ± 1 entries, or drawing $\mathbf{v}$ from the uniform distribution on the hypersphere. In the case of ± 1 entries, this is commonly called Hutchinson's trace estimator.

There are a number of quadrature based methods for approximating spectral densities (and therefore spectral sums). The most well-known are the Kernel Polynomial Method (KPM) [Ski89; SRVK96; WWAF06] and Stochastic Lanczos Quadrature (SLQ) [BFG96]. These algorithms combine quadratic trace estimation eq. (3) with KSMs in order to estimate the moments of $\rho(x)$ to obtain quadrature approximations. A number of theoretical analyses which aim to balance the number of random vectors and the degree of the Krylov subspace [HMAS17; UCS17; DBB19; CTU21; BKM22; CGT23].

4.2 Barriers

Application area scientists are often unacquainted with the subtleties of Lanczos-based methods. This results in algorithmic design choices which result in less efficient and less reliable algorithms. Conversely, numerical analysts have seemingly missed a number of important algorithms and analyses from application literature, and this has resulted in significant time being spent re-inventing the wheel.

Misconceptions about the Lanczos algorithm. The Lanczos algorithm is unstable; the Lanczos vectors and symmetric tridiagonal matrix produced in finite precision arithmetic may be completely different than what would have been obtained in exact arithmetic. Because of this, there has been a widespread hesitance towards Lanczos-based approaches for problems involving matrix functions, at least without the use of computationally expensive reorthogonalization techniques [JP94; SRVK96; ADEL03; WWAF06; UCS17; GWG19]. However, there is actually a lot of work showing that the Lanczos algorithm is still effective for many tasks, even in finite precision arithmetic [Pai70; Pai72; Pai76; Pai80; Gre89; DK91; Kni96; MMS18].

Re-inventing the wheel. What we are calling a quadratic trace estimator is often called the *Hutchinson's trace estimator*, especially when $\mathbf{v}$ is chosen uniformly from the set of vectors with entries $\pm n^{-1/2}$. However, [Hut89] was not the first use of quadratic trace estimators for the task of approximating the trace of an implicit matrix; [Hut89] itself cites [Gir87] which addresses the same task by using samples of $\mathbf{v}$ drawn uniformly from the unit hypersphere. Algorithms based on the use of random vectors back at least to the mid 1970s [ABKS75; WW76; WW77; RV89].

In fact, such estimators are a special case of the concept of *typicality* in quantum physics. Typicality has its origins in work of Schrödinger [Sch27] and von Neumann [Neu29] from the late 1920s but was dismissed and/or forgotten until a resurgence in the mid 2000s [GMM09; GLTZ06; PSW06; Rei07]; see [GLMTZ10] for a historical overview and discussion in a modern context and [JWWLMDR21] for a review of algorithms based on typicality.

Likewise, while the first tail bounds for quadratic trace estimators are typically attributed to [AT11; RKA14], quadratic trace estimators were analyzed before either of these papers. For instance, [Rei07] provides tail bounds based on Chebyshev's inequality for quadratic trace estimators used for the specific purpose of estimating the trace of a symmetric matrix. Sub-Gaussian concentration inequalities for quadratic trace estimators, similar to those in [AT11; RKA14] are derived in [PSW06] using Levy's Lemma, a general result about concentration of measure [Led01]; see also [Gog10, Theorem 2.2.2].

4.3 Proposed approaches

Algorithms for spectrum and spectral sum approximation are relatively mature compared to algorithms for the other topics discussed in this proposal. Even so, there are a number of important problems relating to the use of these algorithms in practice which we target in this proposal. These have significant potential to advance basic science due to the widespread use of the related algorithms.

RD 4: practical a posteriori error estimates (O2,O3). The focus of this research direction is on KPM and SLQ, as these are the most fundamental and widely used algorithms for spectrum and spectral sum approximation. Both methods have two primary sources of error (i) statistical error due to averaging m independent and identically distributed random variables, and (ii) approximation error due to the use of a

Krylov subspace method. These sources of error should be balanced, and a number of existing work provides a priori bounds for these two sources of error [HMAS17; UCS17; DBB19; CTU21; BKM22; CGT23].

However, while a priori bounds provide insight into the types of problems on which a given algorithm may be suitable, they are rarely useful for estimating the actual error of the algorithm when it is run. As such, a posteriori error bounds and/or estimates suitable for use as stopping criteria are needed to make algorithms practical.

We will combine statistical variance estimation techniques such as the jackknife method and bootstrapping [Joh01] with fine-grained error bounds and estimates for Lanczos-based methods [FS08b; FS09; ITS09; FKLR13; FGS14; FS15; CGMM22; XC23]. To facilitate ease of use, these will be integrated into a python program which can easily be integrated into existing codes. We note that error estimates will also naturally lead to estimates for variance reduced estimators which apply eq. (3) to a remainder term [WWAF06; MT20b; MMMW21; PK22; ETW23] including the Krylov-aware based methods introduced in [CH23].

In addition, despite being one of the aspects of this proposal with the largest potential for benefiting basic science, this research direction is particularly suitable for collaboration with undergraduate students.

RD 5: practical/extensible software for KPM and SLQ (O3). To address the lack of software designed specifically for spectrum and spectral sum approximation, we will develop a general purpose Python library for fundamental tasks relating to spectrum and spectral sum approximation based on KPM and SLQ. There are presently a number of one-off packages which make use of KPM approximations, particularly in condensed matter physics [GWAW14; RFC20; MMF20]. There are also some one-off packages for SLQ, particularly in machine learning [Pap19; YGKM20]. However, these packages are focused on specific applications, and therefore the implementations of the KPM/SLQ are fairly specific to the application at hand.

As far as we are aware, all existing KPM implementations make use of a Chebyshev recurrence on an interval containing the spectrum of $\mathbf{H}$. If this interval is not chosen correctly the algorithm may fail to converge, so there is typically a pre-processing step in which the extremal eigenvalues of $\mathbf{H}$ are estimated. This step results in extra computation. In [Che23a], the PI introduced implementation of the KPM based on the Lanczos algorithm which avoids the need for hyperparameters such as the interval of approximation to be selected a priori. Moreover, this Lanczos-based approach makes it feasible to test out many different choices of parameters, such as the orthogonal polynomial family, which can improve the qualitative features of the approximation.

The PI has developed a basic library `spectral_density` [Che23b] which implements the spectrum aware KPM from [Che23a]. The library takes as input the tridiagonal matrix output by the Lanczos algorithm and outputs various spectrum and spectral sum approximations via KPM and SLQ. This proposal will support extending the functionality of the `spectral_density`, primarily by the inclusion of automatic error estimates. Such estimates will be developed as part of research direction 4, and our proposed approach is described more carefully there.

RD 6: dissemination of theory on finite precision behavior (O3). As described above, some of the largest barriers relating to trace estimation arise from a lack of cross-disciplinary knowledge transfer. Addressing misconceptions and lack of knowledge about the Lanczos algorithm in finite precision arithmetic has the potential to significantly benefit application areas, particularly computational quantum physics. In fact, even within the numerical analysis community, the behavior of Lanczos in finite precision arithmetic is not fully understood, and what is known, is not widely known. For example, despite arguably being one of the most important stability analyses of the Lanczos algorithm (especially for tasks relating to trace estimation), [Kni96] has only 7 total citations before 2022.

The PI has already made incremental progress towards this goal. In [Che23a], published in the Journal of Chemical Physics, the PI argued that the Lanczos algorithm could be used to stably implement the KPM. While there are a number of practical benefits to this approach, one of the main aims of the paper was to try and transfer the key ideas from [Kni96] to the Physics community. A more detailed explication for a general

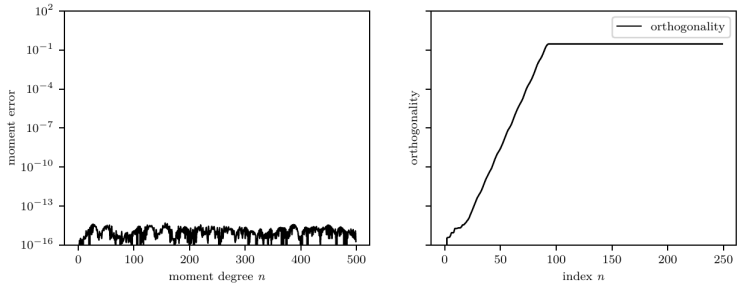

Figure 2: Comparison of error computed by Lanczos with and without reorthogonalization (left) and the loss of orthogonality without reorthogonalization (right). This illustrates that despite the massive loss of orthogonality, the Lanczos algorithm can still be used to accurately obtain the moment information used by algorithms such as the KPM [WWAF06; Che23a]. This proposal would support the dissemination of this behavior to communities outside of physics ensuring they can make the most efficient use of algorithms.

audience is also found in [CTU22]. In addition, the PI has raised this point in discussions and seminar talks with computational physicists.

This goal will be addressed with a dual approach. The PI will continue to produce exposition which can be easily understood by domain scientists. In addition, the PI will increase their efforts to present at seminars and conferences relevant to domain scientists. To facilitate this, funding for conference travel is requested.

5 Partial trace estimation

The final topic of focus is on algorithms for partial trace estimation approximation. While closely related to trace estimation, there are a number of practical and pedagogical differences which merit treating partial trace estimation as a distinct topic.

If $\mathbf{A}$ is a $dm \times dm$ matrix, then we can decompose $\mathbf{A}$ as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \mathbf{A}_{1,d} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \cdots & \mathbf{A}_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{d,1} & \mathbf{A}_{d,2} & \cdots & \mathbf{A}_{d,d} \end{bmatrix},$$

where each $\mathbf{A}_{i,j}$ is a $m \times m$ matrix. The partial trace of $\mathbf{A}$ (with respect to this partitioning) is defined as

$$\mathrm{tr}_{\mathrm{b}}(\mathbf{A}) := \begin{bmatrix} \mathrm{tr}(\mathbf{A}_{1,1}) & \mathrm{tr}(\mathbf{A}_{1,2}) & \cdots & \mathrm{tr}(\mathbf{A}_{1,d}) \\ \mathrm{tr}(\mathbf{A}_{2,1}) & \mathrm{tr}(\mathbf{A}_{2,2}) & \cdots & \mathrm{tr}(\mathbf{A}_{2,d}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{tr}(\mathbf{A}_{d,1}) & \mathrm{tr}(\mathbf{A}_{d,2}) & \cdots & \mathrm{tr}(\mathbf{A}_{d,d}) \end{bmatrix}. \tag{6}$$

In other words, we just take the trace of each of the blocks!

The partial trace arises naturally in quantum mechanics where the state of a quantum system is represented by a density matrix $\boldsymbol{\rho}$. To obtain the density matrix for a particular subsystem of the total quantum system we "trace out" the effects of the irrelevant part of the quantum system (similar to how we might integrate out irrelevant variables of a joint probability density). Mathematically, "tracing out" is done via the partial trace, where the partitioning for the partial trace is determined by the particular subsystem of interest; i.e. the density matrix for the system of interest is $\mathrm{tr}_{\mathrm{b}}(\boldsymbol{\rho})$.

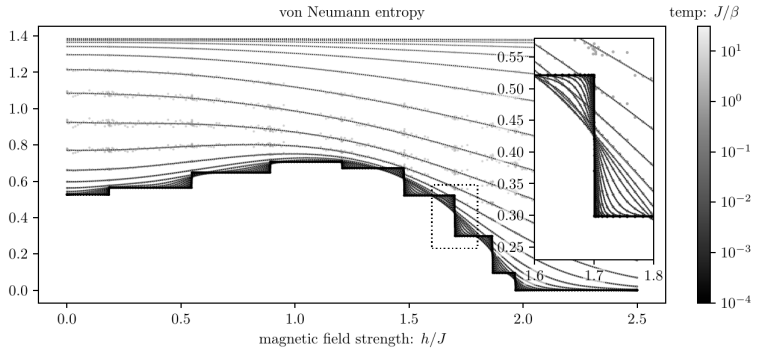

Figure 3: Von Neumann entropy of the 1D solvable spin chain computed with the algorithm in [CCLNPW23] (colored lines) compared with true solution (dotted lines). The research project culminating in [CCLNPW23] included several undergraduate students, including two female students and one Black student. This proposal seeks funding for similar projects involving students from groups underrepresented in STEM.

5.1 Barriers

Partial trace approximation faces more acute versions of of the same barriers as trace and spectrum approximation. As discussed in the next section, there are only a few algorithms for this task all of which have been introduced in the last 5 years. Moreover, there is extremely little in the way of theoretical guarantees for these algorithms since most mathematicians and computer scientists have seemingly never heard of the partial trace! It is the PI's belief that the single largest barrier to the development of partial trace algorithms is the compartmentalization of disciplines.

5.2 Past work

As with the trace, computing the partial trace is trivial when $\mathbf{A}$ is known. However, the relevant setting for quantum systems is when $\mathbf{A} = f(\mathbf{H})$.

PI Chen has recently developed an algorithm for approximating the partial trace of a matrix function [CC22]. The proposed algorithm is conceptually simple and uses the simplest implicit trace estimation algorithm to approximate each entry of the partial trace and uses a Krylov subspace method to approximate the matrix exponential. In particular, the quadratic trace estimator eq. (3) can be used to obtain an unbiased estimator for $\mathrm{tr}_{\mathrm{b}}(\mathbf{A})$,

$$(\mathbf{I}_d \otimes \mathbf{v})^* \mathbf{A} (\mathbf{I}_d \otimes \mathbf{v}) = \begin{bmatrix} \mathbf{v}^* \mathbf{A}_{1,1} \mathbf{v} & \mathbf{v}^* \mathbf{A}_{1,2} \mathbf{v} & \cdots & \mathbf{v}^* \mathbf{A}_{1,d_{\mathrm{a}}} \mathbf{v} \\ \mathbf{v}^* \mathbf{A}_{2,1} \mathbf{v} & \mathbf{v}^* \mathbf{A}_{2,2} \mathbf{v} & \cdots & \mathbf{v}^* \mathbf{A}_{2,d_{\mathrm{a}}} \mathbf{v} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}^* \mathbf{A}_{d,1} \mathbf{v} & \mathbf{v}^* \mathbf{A}_{d,2} \mathbf{v} & \cdots & \mathbf{v}^* \mathbf{A}_{d,d} \mathbf{v}. \end{bmatrix}. \tag{7}$$

Krylov subspace methods for approximating $\mathbf{V}^* f(\mathbf{H}) \mathbf{V}$ can then be used in cases $\mathbf{A} = f(\mathbf{H})$. In [CCLNPW23], variance reduction techniques are studied.

Both [CC22, CCLNPW23] focus on applications to quantum thermodynamics. In particular, open quantum systems in which a the total system is held in thermal equilibrium due to weak coupling with a "super bath" are described by a total system density

$$\boldsymbol{\rho} = \boldsymbol{\rho}(\beta) = Z(\beta)^{-1} \exp(-\beta \mathbf{H}), \qquad Z(\beta) = \mathrm{tr}(\exp(-\beta \mathbf{H})). \tag{8}$$

Thus, approximating the partial trace of $\mathbf{A} = \exp(-\beta \mathbf{H})$ for a range of β is the main computational difficultly in studying the reduced system density matrix $\mathrm{tr}_{\mathrm{b}}(\boldsymbol{\rho})$.

5.3 Proposed approaches

RD 7: Partial trace of large subsystems (O1). When d is large, it might be intractable to explicitly compute and/or store a $d \times d$ matrix such as $\boldsymbol{\rho}^* = \mathrm{tr}_{\mathrm{b}}(\boldsymbol{\rho})$. The existing methods for approximating the estimator eq. (7) with a quadratic form requires storing at least d vectors of length md. However, we might hope to obtain a compressed representation more cheaply. One approach would be to try to sketch the estimator eq. (7) to obtain a low-rank approximation. This can potentially be done using only a few vectors of length md which, in the case $d \gg m$, would result in significant storage savings over the d vectors of length md required by the estimator eq. (7).

As a starting point, we will first consider two simpler problems. First, we will develop methods to estimate quantities such as $\mathrm{tr}(\mathbf{O}\boldsymbol{\rho}^*)$, where $\mathbf{O}$ is some fixed $d \times d$ operator which we can access via matrix-vector products. Using a standard trace estimator, where $\mathbf{u}$ is a suitable random vector, we have an estimator

$$\mathbf{u}^* \mathbf{O}(\mathbf{I}_d \otimes \mathbf{v})^* f(\mathbf{H})(\mathbf{I}_d \otimes \mathbf{v})\mathbf{u},$$

which can be approximated with a Krylov subspace method. We will study the convergence of such methods for common choices of $\mathbf{O}$. Second, we will consider the task of estimating spectral properties of ρ^*, for instance the top eigenvalues or the spectral density. This is intermediate to obtaining a low-rank approximation, but still provides useful insights into properties of $\boldsymbol{\rho}^*$.

RD 8: Extension of partial trace methods to non-equilibrium systems (O1). Many critical questions in physics related to the *dynamic* (time-evolution) behavior of quantum systems. The time evolution of a quantum system with density $\boldsymbol{\rho}(t)$ and Hamiltonian $\mathbf{H}(t)$ is described by the Schrödinger equation

$$\mathrm{i}\frac{\partial \boldsymbol{\rho}(t)}{\partial t} = \mathbf{H}(t)\boldsymbol{\rho}(t) - \boldsymbol{\rho}(t)\mathbf{H}(t). \tag{9}$$

This raises the question: *how can we compute $\boldsymbol{\rho}(t)$ over some time interval $t \in [0, T]$?* In the case that $\mathbf{H}$ is time-independent it is easy to obtain the solution

$$\boldsymbol{\rho}(t) := \exp(-\mathrm{i}t\mathbf{H})\boldsymbol{\rho}(0)\exp(\mathrm{i}t\mathbf{H}). \tag{10}$$

This serves as a natural starting point towards addressing general time evolution. Even so, the development of algorithms for this more restricted setting would immediately allow important physical problems to be studied.

There are a number of existing methods for simulating the evolution eq. (10) when $\boldsymbol{\rho}(0)$ is a pure state ($\boldsymbol{\rho}(0) = \mathbf{u}\mathbf{u}^*$) via time-stepping techniques. In particular, tensor network methods such as the Density Matrix Renormalization Group (DMRG), Matrix Product States (MPS), Projected Entangled Pair States (PEPS), and time-evolving matrix product operator (TEMPO) have found widespread success in computational quantum physics, particularly for local, gapped Hamiltonians [MM95; Sch05; Orú14; SKKKL18; CCVALKG22; CSK22]. Loosely, such methods work by representing length n vectors in a basis of vectors that can each be represented using just $O(\mathrm{poly}(\log(n)))$ numbers. For systems with the right kind of structure and suitable $\boldsymbol{\rho}(0)$, these basis vectors can be chosen in such a way that only a small number of them are needed to represent every vector used in the computation. As such, these approaches can be incredibly memory efficient (requiring only $O(\mathrm{poly}(\log(n)))$ memory), and can be applied to systems where even storing a single dense vector with n entries is impossible.

The case $\boldsymbol{\rho}(0)$ is a thermal state of some system with Hamiltonian $\mathbf{H}_0$ ($\boldsymbol{\rho}(0) = \exp(-\beta\mathbf{H}_0)/\mathrm{tr}(\exp(-\beta\mathbf{H}_0))$) is also of interest. If β is very large, then $\boldsymbol{\rho}(0)$ has only a few non-negligible components, and the above techniques can be applied. However, when β is small, $\boldsymbol{\rho}(0)$ may have up to n components of roughly equal importance. In some cases, $\boldsymbol{\rho}(0)$ can still be represented in a compact tensor network format [VGRC04; KREO19], but for general Hamiltonians there is no general technique for obtaining such a compact representation. Moreover, for generic systems, the number of basis vectors required for tensor network based

approaches can limit their benefits. In such cases, we are unaware of any existing work which avoids the $O(n^2)$ storage required to store the initial thermal state $\boldsymbol{\rho}(0)$.

We propose to apply randomization to avoid this cost when the goal is to output $\mathrm{tr}_{\mathrm{b}}(\boldsymbol{\rho}(t))$. In particular, we will make use of the estimator

$$(\mathbf{I}_d \otimes \mathbf{v})^* \boldsymbol{\rho}(t)(\mathbf{I}_d \otimes \mathbf{v}) \propto (\mathbf{I}_d \otimes \mathbf{v})^* \exp(-\mathrm{i}t\mathbf{H}) \exp(-\beta \mathbf{H}_0) \exp(\mathrm{i}t\mathbf{H})(\mathbf{I}_d \otimes \mathbf{v}). \tag{11}$$

For fixed t, this can be evaluated by first approixmating $\mathbf{V}(t) = \exp(\mathrm{i}t\mathbf{H})(\mathbf{I}_d \otimes \mathbf{v})$ with a KSM, and then approximating $\mathbf{V}(t)^* \exp(-\beta \mathbf{H}_0)\mathbf{V}(t)$ with another KSM. However, it is not immediately clear how to efficiently obtain approximations for multiple times without simply repeating this process. Addressing this will be the main goal of this research direction.

6 Broader impacts

6.1 Facilitate basic science

While the topics of focus are relevant to many domain and application areas, they have all been selected due to their potential for facilitating the advancement of basic science. The objectives of this proposal are designed specifically with the goal of improving practitioner's abilities to easily use numerical methods for tasks involving matrix functions, and the specific research directions reflect concrete and tractable approaches to successful completion of these objectives.

Particular emphasis is placed on the design of applications which can be used to help explore fundamental questions in quantum physics and chemistry. Successful completion of research directions 8 or 7 would open up the possibility of numerically studying aspects of quantum systems for which no tractable methods previously existed. This would immediately allow several important open theoretical questions to be studied numerically, allowing theorists to gain further insight. Simultaneously, practitioners could use such methods to inform and verify physical experiments. Progress on research directions 1, 2, 4, 6, and 5 would improve existing methods for a number of tasks in studying the equilibrium thermodynamics of quantum systems by providing more powerful and user-friendly algorithms thereby allowing scientists to spend less time waiting for algorithms to run and more time on science.

6.2 Advance cross-disciplinary collaboration

One of the barriers to the development of randomized KSMs for matrix functions identified by this proposal is the lack of cross-disciplinary knowledge transfer. This proposal is broadly aimed at mitigating this barrier, and progress on any of the proposed research direction would address this barrier. Research directions 5, 6, 7, and 8 all explicitly target this barrier.

7 Broader impacts: Undergraduate mentorship

Mentorship of undergraduate researchers is a key component of this proposal, and the PI is committed to mentoring students, particularly those from underrepresented groups. He has a demonstrated record of current and past success in doing so.

Funding for 3-4 undergraduate students as research technicians is one of the main costs in the proposed budget. It is expected that each student will serve a tenure of roughly 12 months if they begin at the start of their final year of their undergraduate studies, or 12-18 months if they begin in their penultimate year. Since mentorship of these students is an important aspect of the proposal, we have included an overview similar to what would be required for an REU supplement. The preference of the PI is to hire students as technicians (rather than support them via an REU) in order to more easily prioritize students eligible for federal work study through the internal hiring process for student employees at NYU. Based on communications with the Program Director, it is the understanding of the PI that the DMS Computational Mathematics program has no real preference of one mechanism over the other (REU vs technicians).

7.1 The Research Environment

New York University (NYU) is a world-renowned research institution with many opportunities for undergraduate researchers. In particular, the College of Arts and Sciences hosts an annual Undergraduate Research Conference (URC) and has a twice-yearly application for research funding through the Dean's Undergraduate Research Fund (DURF). Students hired with support from this proposal will be expected to take advantage of these opportunities which will help their development into independent researchers. Students will also be encouraged to make use of other institutional resources such as the Greene supercomputer, writing center, and the many seminars and events hosted by the Courant institute. Finally, due to NYU's location in the northeastern US, conferences and seminars relevant to this proposal are regularly held within a short commute of NYU. Students will be supported in attending events relevant to their research.

7.2 Nature of Student Activities

A significant difficulty in undergraduate research is finding meaningful projects which are suitable for students with relatively little background. The PI has successfully organized several research projects which are expected to result in publications of significant scientific value. It is the opinion of the PI that undergraduate research projects can and should result in an real scientific contribution, and substantial effort will be allocated to conceptualizing such projects.

Research Directions 4, 5, 7, and 8 all have potential to be partially addressed by research projects suitable for undergraduates new to research. The PI is prepared to immediately begin hiring undergraduate researchers for projects relating to these directions once funding is received. More detailed descriptions

were provided in the respective sections above. All projects involving undergraduates will contain a mix of implementing and running numerical algorithms as well as understanding their theoretical justifications. The extent to which a given project will focus on software vs. algorithm development vs. theoretical analysis will be determined based on student interest and background, as well as the overall state of the field when the students begin the project.

Students will work with 1-2 other students who begin around the same time. Having peers working on the same project gives students a chance to discuss research in a lower-stakes/less stressful setting than one-on-one with a faculty member such as myself. In addition, it provides them with additional social aspects which may benefit their overall undergraduate course of study. At the outset of a project, students will work on the same general tasks in order to gather sufficient background knowledge and become more familiar with their peers. This will take the form of reading suggested resources and implementing basic algorithms, as well as weekly group meetings to check progress and provide a chance for questions to be answered. As the students' familiarity with the project mature, students will begin to focus their time on individual goals related to their particular strengths/interests. Weekly group meetings will allow the students will share their progress and receive feedback and suggestions.

Students will be involved in all aspects of the paper preparation and publication process, allowing them to gain insight into the writing process, figure generation, peer review, etc. This will aid in their development into more independent researchers. In addition, students will prepare a presentation or poster for the URC. Students may also attend a research conference to present their results if a suitable conference occurs in the late stages of the research project.

7.3 Student and Mentor Professional Development

It is the view of the PI that mentorship should be tailored to the unique relationship between mentor and (each) mentee. The student activities described in the previous section are all designed to directly benefit the professional development of students, with a particular emphasis on advancing their capacity as independent researchers in computational mathematics.

In order to improve his own mentorship, the PI will continue to engage with training to improve the quality of their mentorship. In particular, the PI will continue to regularly attend university sponsored events, as well as events focused on student development at conferences.

To monitor the progression of students supported with funds from this proposal prepare, the PI will require that students an end of semester report each semester. The report will ask students to describe what they have done, what they have learned, identify areas of strength and for improvement, and what they hope to do in the future. The students will then meet with the PI to review the report and discuss how best the student's professional goals can be supported in the upcoming semester. While the report will allow the PI to make adjustments to their mentorship, filling out such a report also forces students to think critically about their own development, and to begin to take further ownership of their development as researchers.

Finally, the PI intends to maintain professional contact with students once they finish working with him (either due to graduate, end of a project, etc.). The PI believes that this is important in supporting the professional development of mentees, as well as tracking the success of past mentees in order to improve the way current mentees are supported.

All incoming students will go through NYU's employee on-boarding process, which includes comprehensive information on workplace expectations. In addition, the PI will make it clear, through modeling and explicit discussion, that students supported by this proposal (and in fact anyone working with the PI in any capacity) are expected to actively contribute towards making the research environment a welcoming and inclusive environment.

7.4 Student Recruitment and Selection

7.5 Project Evaluation and Reporting

The primary way in which individual student progress will be evaluated is through the end-of-semester reports and meetings described above. Successful student-involved progress on the research directions described above will be viewed as success and is expected. However, it is intended that students will still be able to benefit from their involvement with activities supported by this proposal even if they do not make significant progress on the research directions outlined above, and success will be evaluated holistically. As the number of students involved overall will be small, the PI will track participating students beyond graduation through informal professional contact.

8 Results from Prior NSF Support

8.1 NSF Graduate Research Fellowship, NSF DGE 1762114, 06/2019-05/2022, $138,000

Intellectual Merit. The fellowship resulted in 12 papers focusing primarily on Lanczos-based KSMs for tasks involving matrix functions [Che21, CTU21, CTU22, CGMM22, CGMM23, CC22, BCW22, CH23, XC23, CT23, CGT23, ACGMM23]. Beside publications in numerical analysis, these include publications in statistics [Che21], optimization [BCW22], machine learning [CTU21], and chemical physics [CC22].

Broader Impacts. A number of the works produced during the fellowship aimed specifically to address cross-disciplinary gaps in understanding [CTU21, CTU22, CC22] related directly to this proposal. In addition, the PI put considerable energy towards synergistic actives including organizing two mini-symposiums at international conferences, serving as the department's graduate student representative, organizing the department's reading group on numerical analysis, and mentoring several undergraduate students.