

# Google Summer of Code Application

Trace Estimation of Matrix on Analytic Functions  
such as Matrix Inverse and Log-Determinant

Akshay Jain

## Contents

1	Abstract	1
2	About Me	1
3	Introduction	2
4	Summary of Proposed Work	2
4.1	Stretch Goals . . . . .	3
5	Relevant Contributions and Experience	3
6	Tasks Timeline	4
7	Post-GSoC	5

# 1 Abstract

This project will be concerned with creating a Trace Estimation package for Julia language. The package will include stochastic trace estimation methods such as *Hutchinson*, *Chebyshev Polynomial*, *Lanczos Quadrature* and *Gauss-Radau Quadrature* methods, with strict bounds, for some symmetric positive definite or positive semi-definite matrix  $A$  on some analytic function  $f(A)$ . This package will also include *Diagonal Approximation* and *Hierarchical Probing* methods especially for calculating trace of matrix inverse of large sparse matrix by generating an approximate inverse (i.e, preconditioner) (via *Sparse Approximate Inverse* technique). I will also explore the implementation of *Hadamard Matrices*, *mutually unbiased bases* and some machine learning techniques for random vector sampling. The methods developed during the course of this project will be rigorously tested against actual values, documented and continuously updated on GitHub.

# 2 About Me

**Name** Akshay Jain

**Location** Gandhinagar, India (UTC+5:30)

**University** Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT), Gandhinagar

**Degree** Masters in Information Technology

**Email** zen4akshay@gmail.com

**GitHub** <https://github.com/luca-aki>

**GSoC Organization** The Julia Language

**GSoC Mentors** Mohamed Tarek and Simon Byrne

Hello! I am a highly motivated first year M.Sc. Information Technology student from India. I have a Bachelor degree in Mathematics with a minor in Physics. My work interest lies in Computational Mathematics, Data Science and Artificial Intelligence. I have a strong academic background in Linear Algebra, Numerical Analysis, Real Analysis and Statistics along with an aptitude for reading science journals and research papers. I am efficient at writing code in C/C++, Java, R and Julia. I have been using Julia for practicing data analysis for a while and recently started contributing to it. I am excited to work with and contribute to the Julia community.

### 3 Introduction

Many scientific applications require computation of trace of matrix functions, for instance, calculating *Spectral Density* in solid state physics, *Log-determinant* in machine learning, *Estrada Index* in computational biology and chemical graph theory. Along with that, *trace of matrix inverse* is employed in multitude of applications, such as in Lattice Quantum Chromodynamics.

For a symmetric real matrix  $A \in \mathbb{R}^{n \times n}$  with eigen-decomposition  $A = U\Lambda U^T$ , where  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ , trace of matrix function  $f(A)$  is given by

$$\text{tr}(f(A)) = \sum_{i=1}^n f(\lambda_i)$$

Although trivial, it is a computationally expensive task to calculate complete eigen-decomposition of a large matrix. Since 1989, several algorithms have been proposed that tackle this problem by estimating the trace of matrix without doing the complete eigen-decomposition or solving  $f(A)$ . Many of these algorithms including Hutchinson, Hybrid Chebysev Polynomial and Stochastic Lanczos Quadrature use Monte Carlo simulations ( $\frac{1}{M} \sum_{i=1}^M \mathbf{z}_i^T A \mathbf{z}_i$ , where  $\mathbf{z}_i$  are random sample vectors) to estimate trace of matrix function. These algorithms are very efficient at finding trace of matrix function including matrix inverse for symmetric positive definite matrices with low condition number. Detailed read on trace estimation applications by Ubaru and Saad[1].

### 4 Summary of Proposed Work

I would like to propose and develop a Trace Estimation (*TraceEstimation.jl*) package for Julia language. It will be a standalone package, which will work in consonance with other Julia packages. This package will contain several stochastic trace estimation algorithms such as *Hutchinson* method[2], *Chebyshev Polynomial* method[3], *Lanczos Quadrature* method[4] and *Gauss-Radau Quadrature* method[5]. These methods will be available to estimate the trace of a matrix function  $f(A)$  without explicitly calculating the function on the matrix or doing eigen-decomposition. They will also include variance analysis, bounds for number of iterations, and strict bounds[6] for the estimated trace along with the resulting expected value. All of these methods will be implemented keeping abstraction in mind allowing any *type* of symmetric positive definite or positive semi-definite matrix, for instance dense, sparse, cu, etc, of any *eltype*, on some user supplied analytic function  $f$ , to be processed. (Note: In case of positive semi-definite matrix, with some eigenvalues equal to zero, it is not possible to evaluate some functions like matrix inverse.)

I also intend to develop a *Chebyshev-Hutchinson* hybrid algorithm that combines accuracy of modified Chebysev method[7] with Hutchinson method's Monte Carlo simulations to achieve faster convergence rates for large condition number matrices. For further improvements in accuracy, I will incorporate

*Hadamard* matrices[8] for selection of sample vectors  $\mathbf{z}_i$  in calculation of  $\mathbf{z}_i^T \mathbf{A} \mathbf{z}_i$  for Hutchinson method. Similarly, the accuracy of quadrature methods can be increased with *mutually unbiased bases* technique[9] to sample vectors. This technique helps reduce *RMS* error over large sample sizes.

TraceEstimation.jl package will also contain a non-stochastic *Diagonal Approximation* algorithm[10] especially for estimating trace of matrix inverse for large sparse matrix. This method will generate or accept a preconditioner that will be an approximate inverse of the matrix. The diagonal of this preconditioner will help interpolate the trace of matrix inverse. Furthermore, with a few modifications in the approximation stage of preconditioner, this algorithm will be able to estimate trace of any given analytic function  $f$  on a large sparse matrix. A Monte Carlo approach to *Sparse Approximate Inverse Matrix (SPAI)*[11] is also proposed. This will act as a fast stochastic method to generate preconditioner for the Diagonal Approximation routine.

Each tool developed will go through a rigorous testing process against the actual values, and the unit-tests will be made available with the package. Finally, the package will be released for Julia 1.1 and the latest nightly release, with documented code and a full usage documentation. I will also include a chart showing accuracy of different trace estimators for different kind of matrices.

## 4.1 Stretch Goals

Once I am done with implementing, testing and documenting the above proposed tools, I will head on to implement *Hierarchical Probing* algorithm[12] for  $\text{tr}(\mathbf{A}^{-1})$  for very large sparse matrices used in LQCD. This probing technique effectively exploits the structure of given sparse matrix and attains significant speedup over stochastic methods.

Finally, if time permits, I would love to explore and implement a trace estimation tool that uses machine learning for probing sample vectors. It is demonstrated that precision of trace estimation using  $O(10)$  probing vectors is similar to that of  $O(10000)$  random noise vectors[13].

## 5 Relevant Contributions and Experience

Following are the contributions directly related to this project:

1. **TraceEstimation.jl** (<https://github.com/luca-aki/TraceEstimation.jl>) This is the main repository where I will publish the latest work on the project.
2. **Hutchinson Estimator and Diagonal Approximation** (<https://github.com/luca-aki/TraceEstimation.jl/pull/5>) An implementation of Hutchinson trace estimation method and a related Diagonal Approximation algorithm for  $\text{tr}(\mathbf{A}^{-1})$ .
3. **Lanczos Quadrature** (<https://github.com/luca-aki/TraceEstimation.jl/pull/6>) Implementation of Lanczos iterative method and a basic Stochastic

Lanczos Quadrature method for  $tr(f(A))$ .

Following contributions are not directly related to this project:

1. **TestFunctions.jl** (<https://github.com/luca-aki/TestFunctions.jl>) A Julia package for optimization test functions that I work on my leisure time as a hobby and to learn more about optimization, testing, and plotting in Julia.

**Experience** Along with an academic experience in Linear Algebra, Numerical Analysis and Statistics. I have also taken courses in Algorithm Design, Mathematical modeling, Complex Analysis, Data Analysis, and Machine learning. I am well versed at translating algorithms into efficient code. I have a good understanding of object oriented programming and related concepts. I also happen to have a good experience in video-game design and game intelligence programming using C#/Unity Engine.

## 6 Tasks Timeline

I will be dividing the project into four phases for matching the GSoC evaluation scheme. I will be dedicating at minimum 35-40 hours a week to this project and I will work during weekends as well.

Task	Duration
Community Bonding Period	
1. Interact with selected students and mentors. Learn about their project and tell them about mine.	1 Week
2. Gather requirements, create use cases and do research work for Trace Estimation package.	2 Weeks
Phase 1	
3. Develop and test Chebyshev Polynomial method.	2 Weeks
4. Re-factor hutchinson.jl to work with supplied functions and implement Hadamard matrices.	1 Week
5. Complete Lanczos Quadrature method with improvement in stochastic bounds.	1 Week
Phase 2	
6. Develop and test Hybrid Chebyshev-Hutchinson method.	2 Weeks
7. Develop and test Gauss-Radau Quadrature method.	1 Week
8. Finish any backlogs.	1 Week
Phase 3	
9. Develop and test Diagonal Approximation method.	1 Week
10. Develop and test SPAI for Diagonal Approximation.	1 Week
11. Finish documentation and do stress testing of TraceEstimation.jl.	2 Weeks
Work on stretch goals.	-

## 7 Post-GSoC

After completion of GSoC, I plan to complete my education and continue working with Julia community. In the future, I would like to work on Differential Equations packages, in JuliaDiffEq, (as it was one of my preference) and FluxML.

## References

- [1] Shashanka Ubaru and Yousef Saad. Applications of trace estimation techniques. In *International Conference on High Performance Computing in Science and Engineering*, pages 19–33. Springer, 2017.
- [2] MF Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.
- [3] Insu Han, Dmitry Malioutov, and Jinwoo Shin. Large-scale log-determinant computation through stochastic chebyshev expansions. In *International Conference on Machine Learning*, pages 908–917, 2015.
- [4] Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of  $\text{tr}(f(a))$  via stochastic lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017.
- [5] Zhaojun Bai, Gark Fahey, and Gene Golub. Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics*, 74(1-2):71–89, 1996.
- [6] Farbod Roosta-Khorasani and Uri Ascher. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, 15(5):1187–1212, 2015.
- [7] Gérard Meurant. Estimates of the trace of the inverse of a symmetric matrix using the modified chebyshev algorithm. *Numerical Algorithms*, 51(3):309–318, 2009.
- [8] Paraskevi Fika and Christos Koukouvinos. Stochastic estimates for the trace of functions of matrices via hadamard matrices. *Communications in Statistics-Simulation and Computation*, 46(5):3491–3503, 2017.
- [9] Jack K Fitzsimons, Michael A Osborne, Stephen J Roberts, and Joseph F Fitzsimons. Improved stochastic trace estimation using mutually unbiased bases. *arXiv preprint arXiv:1608.00117*, 2016.
- [10] Lingfei Wu, Jesse Laeuchli, Vassilis Kalantzis, Andreas Stathopoulos, and Efstratios Gallopoulos. Estimating the trace of the matrix inverse by interpolating from the diagonal of an approximate inverse. *Journal of Computational Physics*, 326:828–844, 2016.

- [11] J Straßburg and Vassil N Alexandrov. A monte carlo approach to sparse approximate inverse matrix computations. *Procedia Computer Science*, 18:2307–2316, 2013.
- [12] Andreas Stathopoulos, Jesse Laeuchli, and Kostas Orginos. Hierarchical probing for estimating the trace of the matrix inverse on toroidal lattices. *SIAM Journal on Scientific Computing*, 35(5):S299–S322, 2013.
- [13] Boram Yoon. Estimation of matrix trace using machine learning. *arXiv preprint arXiv:1606.05560*, 2016.