Project No.	**Project Title**	
2020-01-184	Benchmarking BLAS and LAPACK implementation for use with R, Python, and Julia	
Academic Advisor		**Co-Advisor**
Dr. Jonathan Rosenblatt		-
Team Members		
Guy Barel		
guybare@post.bgu.ac.il		

Abstract:
Linear algebra libraries contain essential methods for statistical software. Machine learning and AI use numeric linear algebra operations on data sets represented as matrices and vectors, making numerical linear algebra a core component in the field. In recent years, the HPC community has attracted considerable research attention on obtaining high performance in matrix computations. The growth of database dimensions brings an urgent need for analyzing the data sets in a reasonable time. Therefore, knowing what the optimal libraries are in computing speed is crucial for any user interested in the field.

This project benchmarks and discusses the performance of representative matrix operations, in terms of execution time, for a wide range of linear algebra libraries - Basic Linear Algebra Subprograms (BLAS), Linear Algebra Package (LAPACK), and high-performance libraries (OpenBLAS, ATLAS, and MKL). This evaluation covers an extensive set of BLAS libraries combined with three programming languages, R, Python, and Julia, offering extended analysis compared to related publications. Two main parameters are examined, (i) the selection of a linear algebra library and (ii) the preferred programming language, to select the most efficient combination of both, reaching optimized performance time.

For benchmarking purposes, it is necessary to replace the underlying BLAS implementation of a programming language, where the replacement process has been proven particularly challenging. For this purpose, I use a novel framework called FlexiBLAS to exchange the BLAS implementation at run-time via an environment variable.

The obtained results show that both parameters provide significant performance differences. For larger dimension matrices, optimized BLAS implementations allow for significant performance gains compared to the standard BLAS and LAPACK implementations. Furthermore, the performances of identical libraries with different programming languages results differ. This project provides valuable experiences and insights for the HPC community and can assist future projects in optimizing numerical linear algebra operations time performances.

Keywords: BLAS, LAPACK, OpenBLAS, ATLAS, MKL, FlexiBLAS, R, Python, Julia, Linear algebra libraries, Benchmarks, High-performance computing, Numerical linear algebra