

Project 5: General Matrix Multiplication

The BLAS (Basic Linear Algebra Subprograms) are routines/functions that provide standard building blocks for performing basic vector and matrix operations. There are 3 levels of routines. General matrix multiplication (GEMM) is one of the most frequently used functions in BLAS libraries. The definition of GEMM in a BLAS library is $C \leftarrow \alpha AB + \beta C$, where A and B are two matrices of the same type and the same size, α and β are two scalars.

Please implement GEMM for **double** matrices in C or C++. Please put all your efforts into efficiency.

Requirements

1. The interface of the function should be consistent with the OpenBLAS library, and it is `cblas_dgemm()` in the file <https://github.com/xianyi/OpenBLAS/blob/develop/cblas.h>. The example to call `cblas_dgemm()` is at <https://gist.github.com/xianyi/6930656>.
2. Test the performance using some matrices of different sizes. You can generate some matrices with random values to test your function.
3. Compare your implementation with the `cblas_dgemm()` in OpenBLAS. The results of your implementation should be the same or very similar to those of OpenBLAS.
4. Try to improve the speed of your implementation, and make it close to the speed of OpenBLAS.
5. [Optional] Test your program on X86 and ARM platforms, and describe the differences.

The Report

1. The submitted report should contain the design, implementation, and evaluation of the function.
2. Presentation of the project highlighting the key features and results.

Rules:

1. The project report and the source code must be submitted before the deadline. Any submission after the deadline (even by 1 second) will result in **a score of 0**. The deadline is 23:59 on May 28.
2. The files should be submitted as report.pdf, xxx.cpp, yyy.cpp zzz.h. The files should **NOT** be compressed into one.
3. The score will depend on the quality of both the source code and the report. The report should be easy to understand and provide a clear description of the project, especially the highlights.
4. Attention should be paid to code style. Adequate time is given for code to be written correctly and with good style. Deductions will be made for poor code style. Code style guides, such as

the Google C++ Style Guide (<http://google.github.io/styleguide/cppguide.html>), can be used as a reference.