AMATH 483 / 583 (Roche) - Homework Set 6

Due Monday May 28, 5pm PT

May 21, 2025

Homework 7 (80 points)

1. (+20) **OpenBLAS L1, L2, L3**. We will now upgrade the performance of our BLAS efforts. Measure and plot the performance of double precision L1, L2, and L3 BLAS using the OpenBLAS library on Hyak and the functions stated here. Use appropriate vector and matrix of dimensions in column major format for problem sizes $n = 2$ to $n = 4096$, stride $n* = 2$. Let each n be measured *ntrial* times and plot the average performance for each instance versus n, $ntrial \geq 3$. Submit a single performance plot. Your plot will have 'flops' on the y-axis, or some variation of FLOPs such as MFLOPs, and the problem size on the x-axis.

   ```
   void cblas_daxpy(const int N, const double alpha, const double *X,
   const int incX, double *Y, const int incY);

   void cblas_dgemv(const enum CBLAS_ORDER order, const enum CBLAS_TRANSPOSE TransA,
   const int M, const int N, const double alpha, const double *A, const int lda,
   const double *X, const int incX, const double beta, double *Y, const int incY);

   void cblas_dgemm(const enum CBLAS_ORDER Order, const enum CBLAS_TRANSPOSE TransA,
   const enum CBLAS_TRANSPOSE TransB, const int M, const int N, const int K,
   const double alpha, const double *A, const int lda, const double *B,
   const int ldb, const double beta, double *C, const int ldc);
   ```

2. (+20) **Complex double linear system solver**. Plot both the log of the residual and the log of the normalized error ($\frac{\|b-Az\|_2}{\|A\|_\infty \|z\|_2 \epsilon_{\text{machine}}}$) versus the square matrix dimensions 16,32,64,...,8192 for the following LAPACK routine. It is supported in the OpenBLAS build on Hyak. Submit your plot, and label it accordingly.

   ```
   lapack_int LAPACKE_zgesv(int matrix_order,
                            lapack_int n,
                            lapack_int nrhs,
                            lapack_complex_double* a,
                            lapack_int lda,
                            lapack_int* ipiv,
                            lapack_complex_double* b,
                            lapack_int ldb);
   ```

   Use the following snippet code to initialize your matrices and rhs vectors and note the headers I use:

   ```
   #include <iostream>
   #include <complex>
   #include <cstdlib>
   #include <cstring>
   #include <cmath>
   #include <vector>
   #include <chrono>
   #include <limits>
   #include <cblas.h>
   #include <lapacke.h>

   ...
   int main() {
   ...
   a =(std::complex<double>*)malloc(sizeof(std::complex<double>) * ma * na);
   b = (std::complex<double>*)malloc(sizeof(std::complex<double>) * ma);
   z = (std::complex<double>*)malloc(sizeof(std::complex<double>) * na);
   ```

```
...
srand(0);
int k =0;
for (int j = 0; j < na; j++) {
    for (int i = 0; i < ma ; i++) {
        a[k] = 0.5 - (double)rand() / (double)RAND_MAX
        + std::complex<double>(0, 1)
        * (0.5 - (double)rand() / (double)RAND_MAX);
        if (i==j) a[k]*=static_cast<double>(ma);
        k++;
    }
}
srand(1);
for (int i = 0; i < ma; i++) {
    b[i] = 0.5 - (double)rand() / (double)RAND_MAX
    + std::complex<double>(0, 1)
    * (0.5 - (double)rand() / (double)RAND_MAX);
}
...
```

3. (+20) **Compare OpenBLAS to CUBLAS on HYAK**. Measure and plot the performance of double precision column major matrix multiply ($\alpha AB + \beta C \rightarrow C$) for square matrices of dimension $n = 2$ to $n = 16384$, stride $n* = 2$ for both the OpenBLAS and CUDA BLAS (CUBLAS) implementations on HYAK. Let each n be measured $ntrial$ times and plot the average performance for each case versus n, $ntrial \geq 3$. Submit your performance plot and C++ test code. Your plot will have 'flops' on the y-axis, or some variation of FLOPs such as MFLOPs, and the dimension of the matrices on the x-axis.

4. (+20) **CPU-GPU data copy speed on HYAK**. Write a C++ code to measure the data copy performance between the host CPU and GPU (H2D), and between the GPU and the host CPU (D2H). Copy 1 byte to 2GB increasing in multiples of 2. You will plot the bandwidth for both directions: (bytes per second) on the y-axis, and the buffer size in bytes on the x-axis. Submit your plot and test code.