# MATH 440A/540A Parallel Scientific Computing
# Assignment 1
# Last Submission Date: March 2 (8:00am)

---

This assignment must be entirely your own work. If any part of your assignment has been copied, then your mark may be reduced to zero. Any consultation (given/taken) must be acknowledged. Late assignments will be penalized at the rate of five marks per day. In fact, you should aim to submit the assignment several days before the due date because extensions will not be granted for reasons such as busy Sayers Lab or computer breakdowns.

Submit all Assignment1 computer program files (including a makefile) and a readme.txt file (containing details required to reproduce your reported output) and any PDF/WORD document of your reports (both of the form `xxxx_*.*` : replace `xxxx` with your last name) to `mganesh@mines.edu` and to the grader `breyes@mymail.mines.edu`.
Submit written/printed copies (of your reports, tables, figures, codes etc.) to your lecturer.

---

Let $N$ be a positive even integer. Let $A_N$ and $B_N$ be (given) $N \times N$ matrices (see below). For $i, j = 1, \cdots, N$, let $[A_N B_N](i,j)$ denote the $(i,j)$-th entry of the product $A_N$ and $B_N$. Let $C_N$ be an $N \times N$ matrix with entries, for $i, j = 1, \cdots N$,

$$C_N(i,j) = \begin{cases} [A_N B_N](i,j) & \text{if} \quad |[A_N B_N](i,j)| \le 10^{300}, \\ i^j & \text{if} \quad |[A_N B_N](i,j)| > 10^{300} \quad \text{or} \quad [A_N B_N](i,j) \quad \text{is} \quad \texttt{NaN} \end{cases}.$$

Let $A_N$ be the $N \times N$ (ill-conditioned) Hilbert matrix with entries

$$A_N(i,j) = 1/(i+j-1), \qquad i, j = 1, \cdots N$$

Let $B_N$ be the $N \times N$ matrix with entries, for $i, j = 1, \cdots N$,

$$B_N(i,j) = \begin{cases} b_{ij} & \text{if} \quad |b_{ij}| \le 10^{100} \\ \widetilde{b}_{ij} & \text{if} \quad |b_{ij}| > 10^{100}, \end{cases},$$

where

$$b_{ij} = (-1)^{i+j}(i+j-1)\begin{pmatrix} N+i-1 \\ N-j \end{pmatrix}\begin{pmatrix} N+j-1 \\ N-i \end{pmatrix}\left[\begin{pmatrix} i+j-2 \\ i-1 \end{pmatrix}\right]^2, \tag{1}$$

$$\widetilde{b}_{ij} = (-1)^{i+j}(i+j-1)^2\left[\begin{Bmatrix} N+i-1 \\ N-j \end{Bmatrix}\right]^3\left[\begin{Bmatrix} N+j-1 \\ N-i \end{Bmatrix}\right]^4\left[\begin{Bmatrix} i+j-2 \\ i-1 \end{Bmatrix}\right]^5. \tag{2}$$

In (1)–(2), we used the binomial coefficient and a non-standard notation:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha!/[\beta!(\alpha-\beta)!], \qquad\qquad \begin{Bmatrix} \alpha \\ \beta \end{Bmatrix} = \sin(\alpha)/[\cos(\beta)\tan(\alpha+\beta)].$$

For a given N, let $count(N)$ denote the total number of times the definition (2) is used in setting up the matrix $B_N$

1. Write a sequential program (in F90+/C/C++) to compute the matrices $A_N, B_N$, and $C_N$, for a given even integer parameter $N$.

   (Use loops to compute the entries of the product matrix $A_N B_N$ and avoid any built-in matrix multiplication routines, such as `MATMUL`.)

   Let the CPU time to run the code (from start to finish) for this question, for a given $N$, using a single core be denoted by $T_{1,Q1}(N)$. (Here 1 in the suffix corresponds to running the code on one processing core and Q1 corresponds to the first question.)

   Your program should print the values of $N$, $C_N(N/2, N/2)$, $C_N(N, N)$, $count(N)$, and $T_{1,Q1}(N)$ (with appropriate format and details, to understand the output).
   As a reference for checking your code, note that for $N = 200$ case,
   $C_N(N/2, N/2) = 2.96767264E + 26$ and $C_N(N, N) = 8.11560702E + 12$.

   For small values of $N$ (say $N = 2, 4, 8$) using the output of $A_N B_N$, do you observe any possible connection between $A_N$ and $B_N$. (Write this connection in your report.)

2. For $N = 100 * 2^k, k = 0, 1, 2, 3, 4, 5$, run the code in Q1 in any Sayers Lab machine and, using the output, in a 5-column tabular format (with appropriate headings), tabulate the six values of $N$, $C_N(N/2, N/2)$, $C_N(N, N)$, $count(N)$, $T_{1,Q1}(N)$.

3. a) Write a sequential program (in F90+/C/C++) to first compute, for a given even integer parameter $N$. the matrices $A_N, B_N$ and save $A_N$ and $B_N$ in a hard-disk.

   Let the CPU time to run the code (from start to finish) for this question, for a given $N$, using a single core be denoted by $T_{1,Q3a}(N)$,

   Your program should print the values of $N$, and $T_{1,Q3a}(N)$ (with appropriate details, to understand the output).

   b) Write a sequential program (in F90+/C/C++) to read $A_N$ and $B_N$ from the hard-disk and then compute the matrix $C_N$.

   Let the CPU time to run the code (from start to finish) for this question, for a given $N$, using a single core be denoted by $T_{1,Q3b}(N)$.

   This program should print the values $N$, $C_N(N/2, N/2)$ and $C_N(N, N)$, and $T_{1,Q3b}(N)$. (with appropriate details, to understand the output).

   Thus, for a given $N$, the total CPU time for Q3 (a and b) is
   $T_{1,Q3}(N) = T_{1,Q3a}(N) + T_{1,Q3b}(N)$.

4. For $N = 100 * 2^k, k = 0, 1, 2, 3, 4, 5$, run the codes in Q3 in any Sayers Lab machine and, using the output, in a 7-column tabular format (with appropriate headings), tabulate the six values of $N$, $C_N(N/2, N/2)$, $C_N(N, N)$, $count(N)$, $T_{1,Q3a}(N)$, $T_{1,Q3b}(N)$, $T_{1,Q3}(N)$.

5. Write a report, discussing the tabulated values in Q2 and Q4.
   Try to run the codes in Q1 and Q2, for $N = 100 * 2^k, k = 6, 7, 8, 9, 10$ and explain the reason why you may or may not be able to run this code with some/all of these parameters. (Add any of these additional results in your tables.)

   `For the next question, it is useful to first think of (or write) a sample` $4{\times}4$ `matrix and sub-dividing this matrix into 4 block sub-matrices of equal size` $2{\times}2$. `It may also be useful to think of the` $4{\times}4$ `matrix as a product of two other` $4 \times 4$ `matrices (say` $M_1, M_2$`).`

   `Write down the minimal number (and entries) of row and columns of` $M_1, M_2$ `you require to create each of the four` $2{\times}2$ `sub-matrices. Think of asking four of your friends to setup the four sub-matrices (at the same time) and then you communicate with your friends to collect the four matrices and set up the product matrix` $M_1M_2$.

6. For $N \geq 100 * 2^5$ (and divisible by 4) does your report in Q5 suggest that it is efficient to create four sub-matrices of $C_N$ first (*each of size* $N/2 \times N/2$, *using only appropriate* $N/2$ *rows of* $A_N$ *and* $N/2$ *columns of* $B_N$) using four different computers in Sayers Lab (*at the same time*) than using all four cores in the computer you used for Q5? Justify your answer.

   For a given integer $N$ divisible by 4, write down details required to create the four block sub-matrices $C_{N,i}$, $i = 1, 2, 3, 4$ each of size $N/2 \times N/2$.

   i) For a given parameter $i = 1, 2, 3, 4$, implement the block algorithm (in F90+/C/C++), by allocating and computing only essential elements of $A_N$ and $B_N$ (depending on the value of $i$) to setup and write $C_{N,i}$ in a hard-disk (shared by all the Sayers Lab computers).
   Let the CPU time for running the code from start to compute the matrix $C_{N,i}$ be denoted by $T_{1,Q6comp,i}(N)$, and the CPU time for writing $C_{N,i}$ (after setting-up) to a hard-disk be denoted by $T_{1,Q6write,i}(N)$, for $i = 1, 2, 3, 4$.
   For a given choice of $i$, your program should print the values of $N$, $T_{1,Q6comp,i}(N)$, and $T_{1,Q6write,i}(N)$. (with appropriate details, to understand the output).

   ii) Write a sequential program (in F90+/C/C++) to read the four sub-matrices $C_{N,i}, i = 1, 2, 3, 4$ from the hard-disk and setup the full $N \times N$ matrix $C_N$. Let the total CPU time required to run this code be $T_{1,Q6read,set}(N)$.

   For a given $N$, your program should print the values of $N$, $C_N(N/2, N/2)$, $C_N(N, N)$. $count(N)$, and $T_{1,Q6read,set}(N)$ (with appropriate details, to understand the output).

   Because of the assumption of running the code for $i = 1, 2, 3, 4$ *at the same time*, for a given $N$, approximately, we take the walltime for running Q6 codes, using four cores, as

$$T_{4,Q6}(N) = T_{1,Q6comp}(N) + T_{1,Q6write}(N) + T_{1,Q6read,set}(N),$$

where $T_{1,Q6comp}(N) = \max\{T_{1,Q6comp,i}(N) : i = 1,2,3,4\}$, and
$T_{1,Q6write}(N) = \max\{T_{1,Q6write,i}(N) : i = 1,2,3,4\}$ .

7. For $N = 100*2^k, k = 0,1,2,3,4,5$, run the codes in Q6 in any Sayers Lab machine and, using the output, in a 7-column tabular format (with appropriate headings), tabulate the six values of $N$, $C_N(N/2, N/2)$, $C_N(N, N)$, $count(N)$, $T_{1,Q6comp}(N)$, $T_{1,Q6write}(N)$, $T_{1,Q6read,set}(N)$, and $T_{4,Q6}(N)$.
(Try and run Q6 codes for $N = 100 * 2^k, k = 6,7,8,9,10$ and explain the reason why you may or may not be able to add these additional results in your table.)

8. Write your conclusions in your report, based on the three tabulated values (in Q2, Q4, and Q7) and any figures you may produce using the $N$ values and CPU time (in seconds).