# FINAL REPORT – Reading Course on Scientific & High-Performance Computing

## Questionnaire

1. From the list of topics covered:

```
[ N | I ]  Scientific Computing â€" Motivation
[ sn | si ]  Introduction/Review of C++
[ N | I ]  Modular Programming
[ N | I ]  Libraries
[ N | si ]  Make
[ sn | I ]  Version Control
[ sn | si ]  Debugging
[ sn | I ]  Unit Testing
[ N | I ]  File I/O: self-describing formats, netCDF, HDF5, ...
[ N | I ]  Numerics
[ N | I ]  Numerical Linear Algebra
[ sn | si ]  ODEs
[ N | I ]  Molecular Dynamics, N-body simulations
[ sn | si ]  PDEs
[ N | I ]  FFT
[ N | si ]  Random numbers
[ N | si ]  Monte Carlo
[ N | I ]  Profiling
[ N | I ]  Optimization
[ N | I ]  Supercomputing/HPC
[ N | I ]  Shared-memory programming, OMP
[ N | I ]  Distributed-memory programming, MPI
[ N | I ]  Other parallel approaches, technologies
[ N | I ] Introduction to Quantum Computing(kets, circuits, PennyLane...)
[ N | I ] XANDU(Concepts, projects, etc)
[  |  ]
[  |  ]
[  |  ]
```

**Please add any topic(s) that we may have covered and is not included in the list above.**

1.a) Please describe which topics, from the ones covered in the material were:

- new **(N)**,
- not new but some elements were novel **(sn)**,
- already known **(K)**, i.e. not new.

1.b) Also indicate whether these were:

- interesting **(I)**,

- somehow interesting **(si)**,
- not quite appealing **(NA)**.

2. Please indicate which topics you consider should have been covered with *more* or *less* details?

    - After learning about it and doing my own research, I became quite interested in FFT, so knowing a little bit more about it would have given me a lot more ideas for the topic. I also liked the OMP, GNU Parallel, Parallel workflows, Parallel Shared-Memory, and HPC topics. I believe these were the most important of all the subjects we studied because they were the ones we wanted to use the other topics the most in our work. However, due to time constraints, I do understand that we couldn't go into great detail on these topics. Quantum computing would have been a cool branch to check out in this course, I did some on the side with the workshop as well look at XANDU path in learning.
    - As we required to grasp the procedure and the nature of each topic in order to have a better knowledge of more complex ideas in parallel and HPC, I believed that all of the topics presented were sufficiently in-depth that none of them actually needed less explanation.

10. Please provide any comments about the provided *lectures notes* on "Practical Scientific Computing" by R.Van Zon and M.Ponce.

    - I actually really liked the *lecture notes* since they offered a rapid way to learn knowledge in a short period of time and practically served as a progression chart for what each topic in this course is aiming to cover. Tina and I identified a few minor mistakes, mostly in the grammar and spelling, but none of them indicated that the knowledge about the subject was incorrect as all topics were explained with precise detail and accuracy. I also found that the code examples gave a better idea of exactly how it can be seen in code.

11. If this would have been a *curricular* course offered to CMS students, please provide comments, recommendations or suggestions in how to improve, change, and/or adjust the material covered.

    - I think this would be a fantastic *curricular* course for CMS students since it teaches them how to be more efficient in code, how to code more cleverly, and how to think about new ideas that are relevant to the development of the future of technology. I do believe that having problems or examples from the lecture notes and from Marcelo's examples truly helps to put the knowledge into practice and to be ready for HPC and parallel work. As a result, in order to put the advice or proposal to the test, each topic of the week should have a problem or assignment. The NVIDIA seminar or the SCINET workshops, in my opinion, allow students to examine pertinent topics during this time, and I think it's a nice adaption to the course to have relevent tech company seminars and workshops that involve topics relevant in the sphere of this course to analyse and other topics.

11. i Which topics/courses would you say would have been useful to know before in order to take the best advantage of this course?

    - Although learning C++ and using a teach cluster would be ideal, this course covers much of it and gives students opportunity to work in the environment, so these skills shouldn't be necessary but do provide the course with its primary benefit. Additionally, knowing about file IO, make, modularity, and utilizing libraries would be advantageous because they would help you understand how to do most jobs and concepts practically.

# Report

i. Write a report describing the exercises that you solved, including which parts you found the most challenging ones for each of them and how did you tackle them.

Please indicate the location (URL) of the repository where you worked on these problems.

We have learnt a lot of information during the course that will help us advance our understanding of parallel computing, high performance computing, and scientific computing. We were given a realistic set of questions for each new piece of a subject to attempt to apply our knowledge and perceive it from a practical standpoint. This was a brilliant concept since it allowed us to use what we had learnt to have complete confidence or a greater comprehension of what we had read in relation to the material at hand.

1. Problem set 1 Ex 3 - (https://github.com/blade100a/CSCD92/tree/main/PS1)

- I chose to focus on arrays in the first set of problems (Exercise 3), where the objective was to dynamically generate 100,000 integers and fill an array with a formula so that we could later count the occurrences of the numbers -8 to 9 in the formula. We would print the values and deallocate the data once we had located the occurrences. This course's first assignment was a terrific approach to start understanding how C++ operates and how arrays function. I don't think I had any fundamental concerns with how to tackle this topic because it appeared simple and similar to what we had learned from CSCB07-B09. The count was, in my opinion, the most difficult aspect of this work, and a nested for loop made it simple to observe the occurrences from the formula.

2. Problem set 2 Ex 2 - (https://github.com/blade100a/CSCD92/tree/main/PS2-Ex)

- We then received problem set 2, which, as I shall explain, I feel was the most difficult for me. Exercise 2 was what I chose to do because I enjoy researching matrix and vector operations. Opening a text file, creating an array to contain a matrix, and utilizing defiend methods from the LAPACK library to compute eigenvalues were the primary objectives of this exercise. Finding the appropriate LAPACK function to utilize was the difficult and time-consuming aspect for me. This was after I realised that LAPACK has a huge number of functions that can identify eigenvalues depending on the parameters. The problem that kept coming up was how to correctly use the library, without which I was unable to complete the task. In order for the code to understand where I am referring to the library and function, I had to redownload and find the library properly. The other problem was that the function I was using kept giving me errors. I was unable to fully identify the cause of this problem over time, but I think it may have been due to the wrong parameters being provided, the function not being the

best one to use, or a combination of these. I would also recommend trying the EIGEN library, which is better suited for simple matrix and vector operations. I will return back to this problem as I believe there is probably an easy fix but still have to research on it further.

3. FFT Presentation and Code - (Pres: https://github.com/blade100a/CSCD92/tree/main/FFT-Presentation, Code: https://github.com/blade100a/CSCD92/tree/main/PS2-FFT)

- After finishing our problem sets, we were instructed to conduct research on a subject related to one of the many important numerical approaches used in scientific computing. I chose the Fourier Transform because, although I had heard of it before, I had never fully understood what it was. After doing some study, I learned how powerful and useful this numerical approach is and how almost everywhere it is employed. When I decided to go one step further and try to code it myself, it was one of the hardest aspects. The major goals I set for my code were to produce a complex FFT and complex equations as a result. The design of the FFT plan presented a difficulty for me since I had to ensure that the parameters were provided and that it was carried out correctly. I was able to overcome this challenge by consulting my study and the class notes, and I discovered precisely how to develop, carry out, and then destroy the plan in this particular circumstance. It became really simple and elegant to look at after knowing exactly how to design the plan. It only required a few lines of code. Overall, it was the most enjoyable part since I got to pretend I was a researcher and really do it.

4. Assignment on Parallel Workflows - (https://github.com/blade100a/CSCD92/tree/main/PS3-Workflow-Parallel)

- The practical application of parallel computing was covered in this section of the course, when we were given code and instructed to parallelize it. In addition to learning how to execute a task on the teach cluster, we would see both serial and parallel computing and identify a significant difference in performance and computation time. Finding the precise portion of the serial code to parallelize was the difficult aspect of this task. This problem was resolved for me by parallelizing the for loop while the main work was being performed and importing the gnu-parallel module. As a result, we could clearly observe how much faster the parallel took than the serial, which took 12 to 13 minutes. This was crazy and one of the cool practical research portions we did as the parallel only took 30 SECONDS!

ii. Write a short report about what you consider is the most interesting aspects of Scientific Computing and High-Performance Computing. Also include comments on how you consider (if there is any of) these topics that could result to be useful for your career path.

- The numerical techniques and libraries, particularly FFTs, caught my attention the most when it came to scientific computing and high-performance computing since, after doing my own study, I found this methodology to be very intriguing. I adore how a method developed a long time ago is incredibly special, discovering and displaying different traits from a big pool of data. Finding out how this method is used for phones, hospitals, headphones, etc. was also fascinating. Another intriguing feature of this method is the existence of libraries that make using FFT considerably simpler than having to write code for it. This approach proves to be quite helpful for my job path since if I were to conduct any scientific computing, I have found a range of libraries to do procedures more conveniently but also effectively, making code more simpler and ideal. The overall performance and efficiency(using modular programming, testing, debugging, ...) of scientific and high-performance computing is another element since there are many ways to improve code and manage a lot of data. Having the capacity to do activities and duties at a lot faster rate is helpful in any career, thus this becomes quite handy.

iii. Consider that you are given a code which is required to be optimized. The code can take dataset and the size of the data sets affects the different metrics of the performance (i.e. running time, memory utilization, IO, communication, etc.)

a) What steps would you take in order to study and draft a performance analysis report of this code?

- Given a code that takes a dataset and the size of the data sets affects the different metrics of the performance is a problem that can be faced with large scale problems or large data to be interpeted. There are variety of steps that can be taken in order to study and draft a performance analysis report of this code. I would first, see how much of the code is IO as its the slowest portion of most code with large amounts of data. I would see how much of performance is given to just this portion if its bottlenecking, possible solutions changing data to binary format, parallelize the io, minimize number of files for more responsive, etc. After checking the elapsed and cpu time we would determine the most performance we can get out of the slowest part. I would add timing statements to measure out certain sections of the code to find areas than be optimized, modularized, etc.

iv. You are in the process of submitting an allocation request for compute usage to one of the national systems, for which you must submit an scaling analysis of the jobs you will run. For details on this process, see https://alliancecan.ca/en/services/advanced-research-computing/research-portal/resource-allocation-competitions/rac-frequently-asked-questions

a) What type of information should you submit?
   - Depending on the code, the data I would provide would be the amount of nodes, cores per job, tasks, and number of GPUs. I would do a scaling study to find the optimal number of resources (nodes, CPUs, and GPUs) for improved performance in some of these scenarios. Using some timing statements to evaluate and optimise the code's pertinent areas, running multiple times to see trends.

b) Should you also submit a strong and weak scaling? Justify.
   - I believe here I would need to submit a strong and weak scaling, the reasoning being is it provides a good indication to the job sizes and how much resouce should be requested from the national system as it provides a better level of efficiency. We know strong scaling(Amdahl's law) is fixed size with respect to a number or processors and weak scaling(Gustafson's law) is scaled size with respect to a number of processors, thus providing a better insight in requesting the number of processors needed for each job once submited. Otherwise, if we dont submit a strong and weak scaling, we wouldn't know the best efficient way of requesting our jobs especially if they are larger or smaller jobs in scale.