Next Step: Prerequisites for the Upcoming Lecture

I've decided to forgo the idea of assigning you as reviewers for another document (just this once).

On Wednesday, we will begin our programming segment. To ensure a smooth start, it's crucial that everyone has the capability to develop and run programs on AlmaLinux 9.

Programming languages useful for numerical computing can be broadly categorized into two classes: compiled languages and interpreted languages. We'll delve deeper into this distinction in future lectures. For now, I'd like each of you to select:

1. One compiled language
2. One interpreted language

These will be the languages you'll primarily use throughout the course.

To prepare for Wednesday's lecture, please:

1. Ensure your AlmaLinux 9 environment is fully set up and operational.
2. Research and choose your preferred compiled and interpreted languages.
3. If possible, install the necessary tools and environments for your chosen languages on your AlmaLinux 9 system.

If you encounter any difficulties or have questions about this preparation, please don't hesitate to reach out.

Suggestions for Choosing Your Languages:

Compiled Languages (select one from):

- C
- C++
- Fortran

My Opinions:

1. For Those Interested in High-Performance Computing (HPC):
   Fortran, in my opinion, excels in numerical computing, especially for HPC. It's simple yet powerful, but its widespread use is limited to specific fields such as climate science, geology, and astrophysics - areas that heavily rely on HPC. If you learn c is very easy to learn and use Fortran.

2. Regarding C++:
   While powerful, C++ can be challenging for beginners when used to its full potential. If you're considering C++, I'd still advise starting with C, as C++ can be viewed as an extension of C.

My Recommendations for Compiled Languages:

I strongly recommend C. It's an excellent foundation for understanding core programming concepts and is widely used across various fields. Moreover, programs written in C are easily integrated with other languages, both compiled and interpreted. This makes C an excellent choice for broad applicability.

Getting Started with C:
For those choosing C, I recommend this guide to help you begin: https://github.com/tgmattso/CompSciForPhys/blob/main/learningC.c

Interpreted Languages (select one from):

- Python
- Julia
- Mathematica
- Matlab

Recommendations for Interpreted Languages:

1. If You're Already Familiar:
   If you're already proficient in one of these languages, I encourage you to choose a different one. Expanding your skill set can broaden your perspective and enhance your problem-solving abilities.
2. For Those New to Interpreted Languages:
   Each of these languages has its strengths and use cases in numerical computing. Here are some considerations:

a) Julia:
My personal recommendation is Julia. Despite being relatively new, it was designed to combine the speed of compiled languages with the ease of use similar to Matlab. It's specifically tailored for numerical computing, making it an excellent choice for our course.

b) Mathematica:
Another strong contender is Mathematica. While not as fast as Julia, it offers a robust set of numerical features and excels in symbolic algebra. Its integrated environment can be

particularly useful for certain types of problems. But still, Julia could be better to be learned during this course than later by yourself (Mathematica has very good tutorials!)

c) Python:
Known for its versatility and extensive libraries, Python is widely used in scientific computing, data analysis, and machine learning. But, to be speed competitive with other languages it needs to heavy rely on external libraries and compiled language wrapping (this could exponentially increase its interpretability and debugging complexity)

After choosing a compiled and an interpreted language now is the problem you want to solve:

Given the following input:

a = 3 -> a real-valued scalar number

x = (1,1,1,1,1, ..., 1) -> a real-valued vector of dimension N = 20

y = (4,4,4,4,4, ..., 4) -> a real-valued vector of dimension N = 20

write a computer program that calculates the following scalar product: z = a*x + y and print its value on a text file.

I know very well that some of you have never programmed, and for some other of you this is an easy task, however for the next lecture is important to have this in mind:

1) Probably you have to update your Markdown guide for docker to be able to compile and run this program [search information for DockerFile, have a look at: https://stackoverflow.com/questions/19585028/i-lose-my-data-when-the-container-exits and https://phoenixnap.com/kb/how-to-commit-changes-to-docker-image#:~:text=Docker%20images%20are%20immutable%2C%20i.e.,to%20a%20new%20Docker%20image.]
2) Don't be shy, ask help to your colleagues, look at their Markdown and use AI! Use Discord channel #oracle to ask for help, me or another of your colleagues could reply!

Don't worry if you are not able to finish this task, if you feel lost or if you think everything is too messy, in the next lecture we will review all together the tasks (however, please try anyway to do it, it is important at least to try!) Remember: making mistakes is the best way to learn (and this is true for everything, not only programming).