104B CP1 Group 6 Report

Who wrote it and who took what role and what is it about?

Group 6's project is taking the charge on comparing Chat GPT's performance on a few desirable methods from either Math 104a or 104b, and evaluating the performance and coding style between a human and a LLM (Large Language Model). These methods include the Jacobi iterative method for solving *linear* systems of equations and the gradient descent method for optimization problems.

The report was written by *Reese Karo* and with the help of everyone in the group, and the coders that will be comparing their style of code for their favorite method with the Artificial intelligence's code are *Emerson Domingo*, *John Lain*, *Sean Nguyen*, and *Zhenyuan Ni*. There was no specific leader of the group since everyone shared the same tasks of writing code for their desired method. Since there was no presentation for our group on this project, we did not need a member creating and delivering a presentation file.

What is the main point, conclusion, or story? Can you contrast what one can reasonably expect and what really happened?

The main idea behind this project was to compare performance between machine and human coders. Learning math in the classroom is important to get the main ideas, but learning to code in math is a whole different level of learning. So, our vision was to choose our favorite methods coming from 104a and 104b which included: Jacobi iterative method, gradient descent method, and the LU decomposition method. We want to test and see how close in comparison the AI's code is in comparison to a human writer. Everyone knows how quickly these LLMs can code up an algorithm, but we would like to see if they perform accurately compared to human-written code. Therefore our main goal is to compare the performance of a machine that can write the code in less than a minute to ourselves as programmers and mathematicians.

One conclusion drawn from the use of GPT 4 and GPT 3.5 was when working with the Runge Kutta 4th order method or, RK4 method. We see that GPT 4 would break up the Runge Kutta method into separate helper functions, one for the stepping, and to complete the iteration and save the results. The RK4 method under the 3.5 model, which roughly took 10-15 seconds to write, returned a more explicit code which could be easier to understand for the common coder. Thus the speed of the 3.5 model and the explicitness may be easier for most people. Another point to add is the use of plotting. Both models plotted, however the 3.5 model used a 3d plotting system using Matplotlibs 3d projection in the line: "ax = fig.add_subplot(111, projection = '3d)'". The GPT 4 model used a simple 2d plotter to help display the results, which made it harder to understand and conclude results.

What is the evidence or context of the point or conclusion? In particular, what prompts did you use when using AI tools?

The fact that the code written by GPT performed as well as code written by humans in some examples is evidence that AI is a powerful tool for algorithm/code writing.

Some examples of the prompts used:

- Can you create a gradient descent method and make this code more efficient (with code included)?

The code before "# multivariable example" is the one offered by ChatGPT, and the code after this comment is a quick test by myself. This trial shows that everyone can use ChatGPT as an algorithm searcher to find a suitable method to solve real world problems, and use a quick example to check if the function offered is reliable. For example, this question about local minimums takes an important role in the neural network model Backpropagation process. Whenever facing a problem, ChatGPT is a good way to locate a method to solve the problem.

And for Jacobi Method, we asked chat GPT 3.5:

- "Write me an algorithm in python for the jacobi method of solving matrices"

Similarly for the RK4 method taught in 104A to help with solving ODE's analytically,

- Can you build me a script that can perform rk4 method with the Rossler and Chen attractor with these initial conditions: initial conditions: Rossler: x0, y0, z0 = 0, 1, 0 Chen: x1,y1,z1 = 5,10,10 x2,y2,z2 = -7,-5,-10

What is the limitation of what you did?

The main limitation of this project was coming up with a relevant idea that is interesting and unique within the given time frame of the project. We had many considerations of different projects to pursue. With Chat GPT becoming more mainstream, even with non-computer science oriented students, we knew it would be an interesting and current topic to pursue. And so, with careful consideration of the time given, we decided to move forward with the simpler project of comparing code and performance and user readability between the machine and the human.

Any advice for those who want to do similar experiments?

Chat GPT, Copilot, AI Agents and LLMs in general are very powerful tools that are revolutionizing tech and everyday work across every industry. It has allowed us to cut down on time in writing and reading with shorter paragraph summaries of long essays and literature, repetitive coding tasks, and help with creativity for certain topics. What these LLM's have struggled to work on is mathematics due to the structure of the LLMs.

Is it well organized?

The project is stored in a repository under the ownership of Reese Karo, and contains files with each coder's personal file to write, and a main notebook containing every method, we markdown comments to supply the reader with supplementary information about the code. The methodology behind the organization follows as human method #1, same method #1 but by the machine, and then comparing and contrasting with comments in the jupyter notebook file.

Reflect on the whole process. What do you want to make note of to become a better programmer/presenter/documenter/leader?

It was interesting to work on programming with a small group, which is different from what us students have normally experienced in our math classes. A lot of the time, coding is pretty isolating, with one person in front of a screen working on their program alone. However, I think this way of creating code is more similar to what people in the professional world experience, as they work on big projects, like the AI tools we researched in our project. Furthermore, it was more beneficial to us as students, as well, because we were able to bring all of our knowledge from our different specialties and passions to the table, and overall grow as programmers and mathematicians from the process.

I think brainstorming ideas as soon as the project instructions are released would have helped our group. I think giving yourself and the group as much time as possible to work on the project helps out in the long run and with the quality of the project. Coding is a process that you may have to step away from and return back with a fresh mind, in order to get past coding blocks. I also think that in terms of becoming a better coder, Chat GPT is a good resource to use in terms of efficiency and quality. We don't believe it's a useful tool for just strictly coding and teaching coding, but to aid in the performance/quality of a coder who has already been coding and wants to speed up the process of their project or their own codes.

Secondly, to help with the process of a project like this, communication is the main key to being successful. When communication lacks in big group projects, it can make it hard to achieve anything or make any progress. However the communication takes time and builds upon itself. Using collaborational tools in these projects such as Github and Google docs/slides, allows everyone to be on the same page.

Lastly, in terms of becoming a better programmer, I think it is important to remember that there will always be mistakes in the process of achieving your outcome. Some of us definitely struggled when we were trying to create a program for these different mathematical concepts that didn't work seamlessly. So, it is important to note that there is never a "perfect" code, as there are so many different ways to create something. However, when going through a roadblock in your coding process, it is important to keep an open mind, and allow yourself to use resources like Chat GPT, or other people's ideas to learn other's perspectives on the subject matter. Not only will this help you fix bugs and such, it will also help you gain a deeper understanding of the concepts, and become a better programmer, in general.