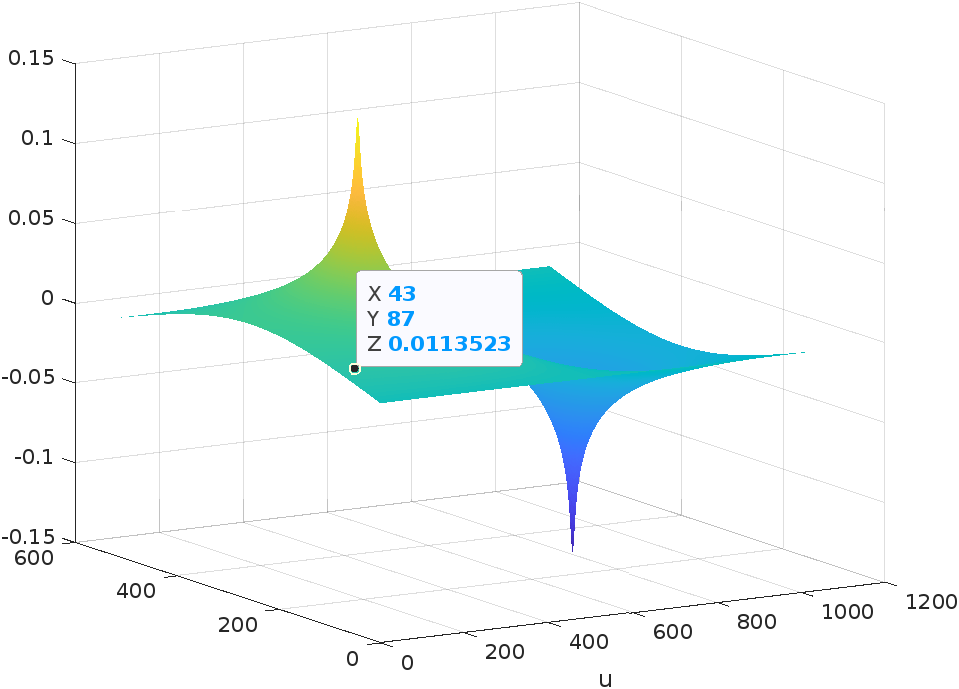
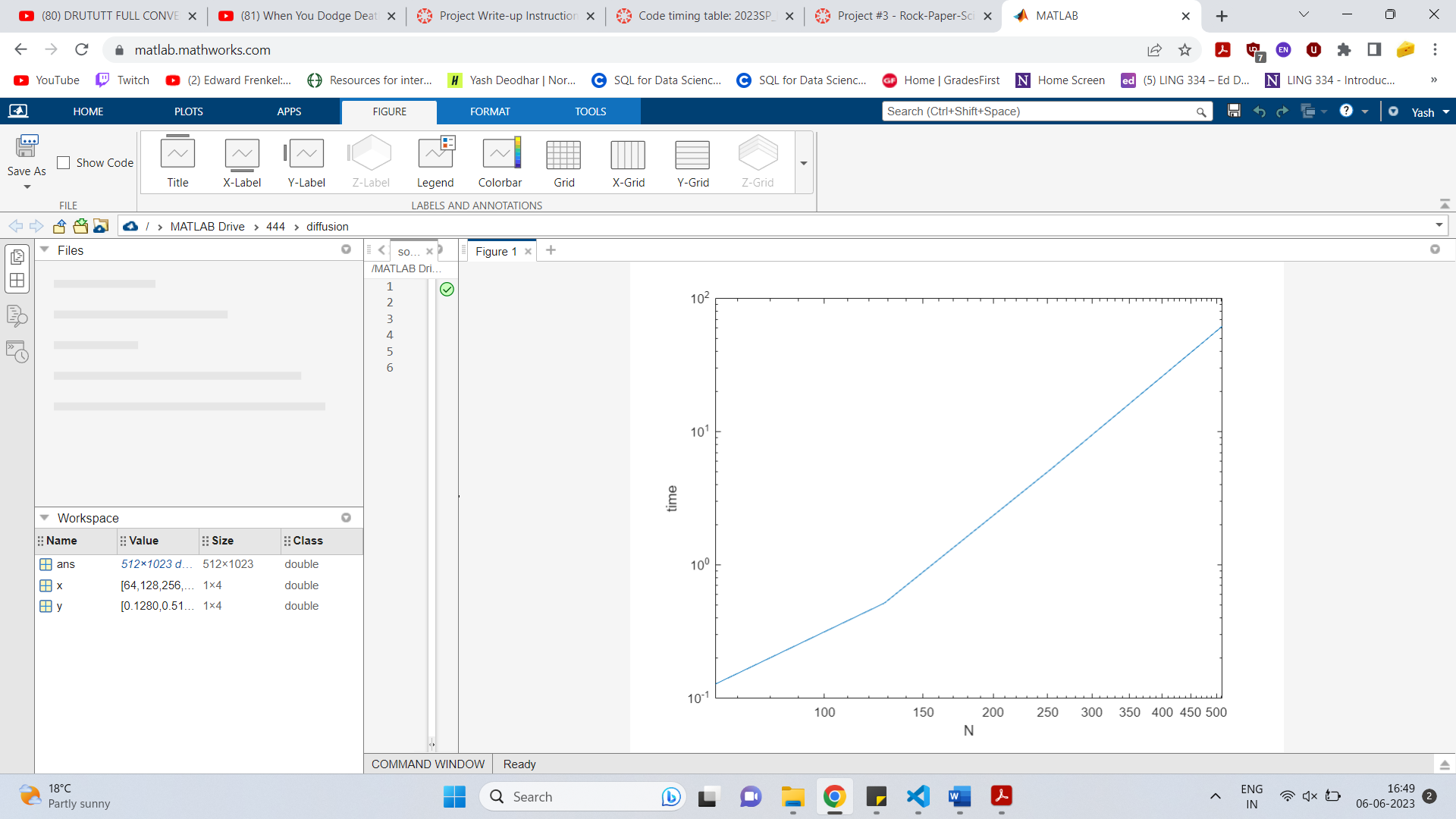
* Makefile command: make (command inside: nvcc -o diffusion diffusion.cu -lm)
* Output for N = 512, omega = 1.95, tolerance = 1e-9, max iterations = 30000:



* N vs time plot:



* Time comparison:

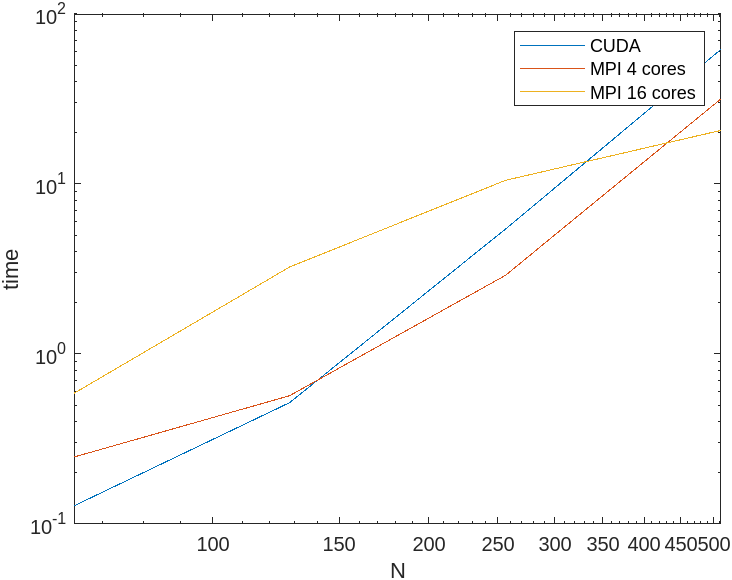
Chopps time: 2.4 seconds

Last year’s class average: 63.3 seconds

My time: 61.8 seconds

I could have made my code a lot faster if I used a kernel to transpose the matrix in the end instead of doing it serially. I could also have thought of a better strategy to implement the reduction kernel.

* CUDA and MPI comparison:



As can be seen from the figure, for low values of N, CUDA outperforms MPI but if you’re using 4 cores MPI becomes faster than CUDA just after N = 128. Furthermore, using 16 cores proves to be a better choice after about N = 300.

However, these results should be taken with a pinch of salt because I believe that my CUDA program had a lot more room for improvement than my MPI program and hence it could have much lower time values if implemented correctly.