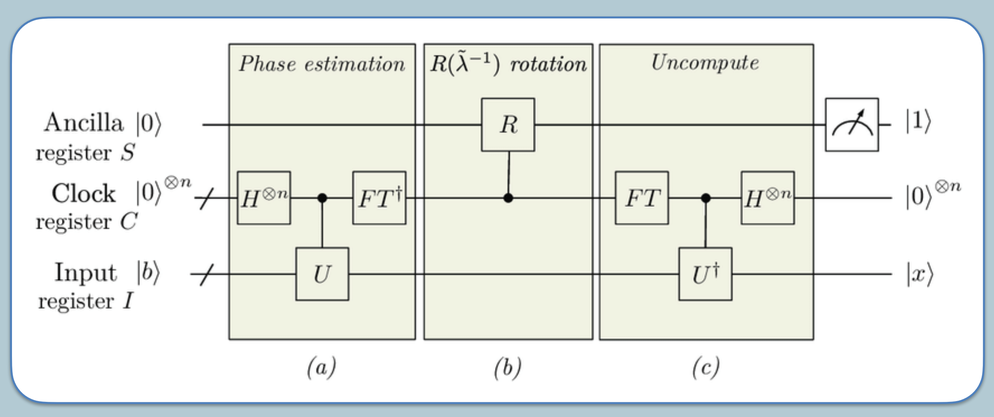
Solving linear systems with HHL quantum algorithm on GPUs

Dhruv Sood[1], Manish Modani[2], Nilmani Mathur[1], Vikram Tripathi[1], Andreas Hehn[3]
[1]Department of Theoretical Physics, Tata Institute of Fundamental Research, Mumbai;
[2]NVIDIA India; [3]NVIDIA Switzerland.

Introduction

- Linear Systems of equations are ubiquitous in all fields of science and often arise as intermediary steps in studies of physical systems. Classical methods for solving such problems take time that scales polynomially with the problem size.
- The HHL algorithm [1] is a quantum algorithm that is able to solve such systems in time that scales logarithmically with the size. Below is a schematic diagram of the quantum circuit corresponding to it.

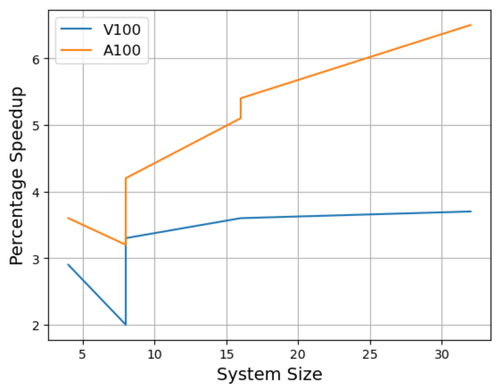


- The HHL algorithm employs Quantum Phase Estimation and the Quantum Fourier Transform to estimate the eigenphases of a hermitian matrix *A*. These are then employed to invert the matrix and solve the titular problem *A*x=b.
- The algorithm can be employed solve large linear systems and hence investigate physical systems more efficiently once satisfactory quantum hardware is available. We aim to use this and other such algorithms for studying quantum many body systems, which are often challenging to study via classical means.
- Further, NVIDIA's cuQuantum SDK allows us to speed up such simulations using GPUs. It has been demonstrated to provide significant speedups to routine like Shor's Algorithm, the Quantum Fourier Transform, and Sycamore [2,3]. We believe that cuQuantum will allow us to exploit classical hardware for quantum simulation and be quantum ready when the requisite quantum hardware is available.
- The HHL algorithm is implemented with cuQuantum with the goal of performance optimisation and making it scalable to higher system sizes and number of qubits.
- In this work, we present initial results obtained from the implementation of the HHL algorithm and the speedups gained by GPU enablement.

A Test : Diagonal Matrices

We begin with systems where the problem matrix is diagonal with entries corresponding to binary fractions upto a multiplicative constant. Such matrices were used to verify that the implementation was satisfactory. We have tested the HHL algorithm for various matrix sizes and complexities. All runs were conducted first using only CPU capabilities, and then using V100 (32 GB) and A100 (80 GB) NVIDIA GPUs. The table and plot below shows the times taken in executing HHL for such matrices.

System Size	Total Qubits	Speedup with V100 GPUs (%)	Speedup with A100 GPUs (%)
4	8	2.9	3.6
8	8	2.0	3.2
8	10	3.3	4.2
16	10	3.6	5.1
16	12	3.6	5.4
32	12	3.7	6.5



We checked the results of the simulations by calculating the projection of the obtained result upon the exact solutions. This fidelity was found to average at 0.87 over all conducted runs.

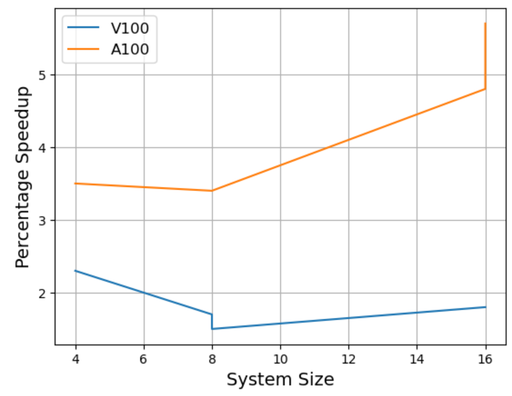
This serves as a demonstration that the HHL algorithm has been implemented correctly and accelerated with cuQuantum. We now move on to matrices where the eigenvalues are not binary fractions and see how the algorithm fares there.

Irrational Eigenvalues

After establishing the viability of the implementation and demonstrating the small but visible speedup achieved using cuQuantum, we moved onto testing it on diagonal matrices where the eigenvalues are generated randomly.

This is expected to interfere with the phase estimation routine in the HHL algorithm, hence increasing the circuit depth and the time taken by the simulation. A table and plot detailing the times taken in such runs is presented next.

System Size	Total Qubits	Speedup with V100 GPUs (%)	Speedup with A100 GPUs (%)
4	8	2.3	3.5
8	8	1.7	3.4
8	10	1.5	3.4
16	10	1.8	4.8
16	12	1.8	5.7



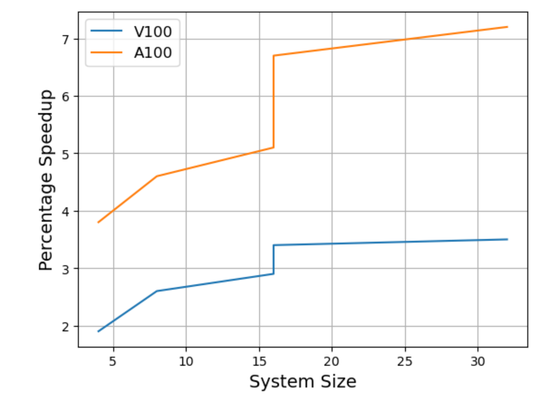
The table above shows that cuQuantum is able to speed up the simulation by up to 5.7%. We further observe that this speedup increases as we increase the size of the problem, which is due to the fact that at lower qubits the problem does not utilise all available cores.

On an average we find a speedup of 2.42% using V100 GPUs and 4.52% using A100 GPUs. Another characteristic evident from the date is that while the speedup attained via V100 GPUs appears to stagnate, the A100 speedup appears to still be rising. We are exploring this further for higher system size and number of qubits.

Randomised Matrices

We now move on to working with matrices where the eigenvalues and elements are generated randomly. In this scenario, a diagonal matrix with random entries is generated and then subjected to a random rotation to make it non-sparse.
In such runs the circuit depth is much higher and thus the simulation takes considerable time. The times obtained for these runs are tabulated and plotted below.

System Size	Total Qubits	Speedup with V100 GPUs (%)	Speedup with A100 GPUs (%)
4	8	1.9	3.8
8	10	2.6	4.6
16	10	2.9	5.1
16	12	3.4	6.7
32	12	3.5	7.2



Again we see that a significant speedup is achieved in GPU-enable runs. Indeed, the average speedups of 2.86% for V100 and 5.48% for A100 are both higher their counterparts in the previous section. This is likely due to the increased complexity of the matrix and the higher circuit depth demands more system resources and a GPU-enabled run is better at coping with that. We aim to pursue this effect as well in future work. The scaling of the speedup still follows the same pattern as above, and can be attributed to the same reasons.

These results demonstrate that one can try and solve linear systems corresponding to realistic problems with the HHL algorithm. Here we have demonstrated that using up to 12 qubits. We believe it is possible to further increase the system size and optimise both the implementation of the HHL algorithm and the compatibility with cuQuantum to obtain better speeds. We have noticed that a significant amount of time is spent during preprocessing (including transpilation) which restricts the scalability. However, we are actively working on remedying this. Once that is achieved, we plan to apply this to studying physical models related to quantum many body physics.

Conclusions

- We have implemented the HHL algorithm successfully for diagonal, diagonal with irrational eigenvalues and randomly generated hermitian positive definite matrices.
- We have demonstrated speedups for HHL algorithm using V100 and A100 GPUs.
- We have found that the percentage speedups increase with system size.
- We plan to optimise it further to extend it's compatibility to dense matrices that possess highly entangled eigensystems and to scale it for larger systems.
- With this, we envisage to study entangled quantum many body systems in the near future, which are otherwise difficult to investigate.

References

1. Harrow, A. W., Hassidim, A., & Lloyd, S. (2009). Quantum Algorithm for Linear Systems of Equations. Phys. Rev. Lett., 103150502.
2. M. Modani, A. Banerjee and A. Das, "Performance analysis of quantum algorithms on Param Siddhi-AI system," 2022 International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT), Pune, India, 2022.
3. Modani, M., Banerjee, A., Das, A. (2023). Multi-GPU-Enabled Quantum Circuit Simulations on HPC-AI System. In: Sharma, N., Goje, A., Chakrabarti, A., Bruckstein, A.M. (eds) Data Management, Analytics and Innovation. ICDMAI 2023. Lecture Notes in Networks and Systems, vol 662. Springer, Singapore.