Solving linear equation systems on noisy intermediate-scale quantum computers

André Mielke¹,* and Tim Ricken³

¹ Institute of Mechanics, Structural Analysis, and Dynamics, Faculty of Aerospace Engineering and Geodesy, Pfaffenwaldring 27, 70569, Stuttgart, Germany

Quantum computing promises classically unparalleled benefits for various applications. Its properties are exploited in the Harrow-Hassidim-Lloyd (HHL) algorithm that, in conjunction with quantum phase estimation, is capable of constructing quantum states that are proportional to the solution of linear equation systems and does so exponentially faster than the fastest known classical algorithms. We explore this capability by computing the nodal displacements of a 1-dimensional loaded cantilever, discretized by using the finite element method (FEM).

1 Quantum computing

1.1 Introduction

A general qubit state can be written as \(|\Psi\rangle = \alpha_1|0\rangle + \alpha_2|1\rangle\) [1], where the coefficients \(\alpha_i\) are the amplitudes of the qubit state. Akin to classical linear algebra, \(|0\rangle\) and \(|1\rangle\) are basis vectors, representing classical computing states. For multi-qubit register states, the tensor product of each qubit’s basis vectors is taken to get the basis vectors of the full system, resulting in \(2^n\) basis vectors for an \(n\)-qubit system. Hence, the available space for performing computations grows exponentially with each added qubit. A computation itself is a rotation \(U\) in the Hilbert space spanned by the basis vectors. Applying this rotation to a general \(n\)-qubit state can be expressed by applying this rotation to each basis vector [1]. This gives rise to quantum parallelism. If each basis vector encodes one datum of a set of inputs to an algorithm, applying \(U\) to \(|\Psi\rangle\) will result in a linear combination of each individual solution.

1.2 The Harrow-Hassidim-Lloyd algorithm

The exponentially increasing computing space of quantum registers can be used to solve sparse linear equation systems faster and taking up less space than classical approaches by using analog-encoding [2], where the information about the original matrices and vectors is encoded in the amplitudes of multi-qubit states. A requirement for HHL is that the matrix must be hermitian, which is automatically given for orthogonal matrices, and of dimension that is some power of 2. Both can be ensured by appropriate transformations [3].

The problem to solve is \(Ku = f\), and the goal is to construct \(|u\rangle = K^{-1}|f\rangle\). The trick is that if \(K\) is Hermitian, it can be implemented on a quantum computer as a Hamiltonian, creating the time evolution \(|u(t)\rangle = \exp(-itK^{-1})|f\rangle\), which by itself would suffice to gather some global information about the result, such as the largest displacement [3].

To get the full solution \(|u\rangle\), quantum phase estimation [4] is employed. It introduces another qubit register of the same size as \(|f\rangle\), initialized as \(|0\rangle\). The Hamiltonian here is constructed by tensoring \(K\) with the momentum operator \(\hat{p}\), which induces a translation in the second qubit register proportional to the eigenvalues \(\lambda_i\) of \(K\). The time evolution is

\[
\exp(-itK \otimes \hat{p}) |u\rangle |0\rangle = \sum_i \beta_i |\psi_i\rangle |\lambda_i t\rangle ,
\]

where \(\beta_i\) are some unknown, irrelevant amplitudes, and \(|\psi_i\rangle\) are the eigenvectors corresponding to the eigenvalues \(\lambda_i\) of \(K\). The quantum computer can now use the eigenvalues \(\lambda_i\) saved in the second quantum register to apply the phase \(U_\phi = \exp(id\lambda_i^{-1})\) to the full state, after which the first computation from equation 1 is undone by applying the inverse transformation to get

\[
\sum_i \beta_i \exp(id\lambda_i^{-1}) |\psi_i\rangle |0\rangle = \exp(idK^{-1}) |f\rangle |0\rangle \approx (1 + idK^{-1}) |f\rangle |0\rangle ,
\]

where the linearization on the right side is sound if \(d\) is sufficiently small. The relevant part \(idK^{-1}|f\rangle\) can be extracted in logarithmic time [4]. This procedure can be modified to get the extraction in \(\text{poly}(\frac{e^{2N}}{\epsilon^6} N \log N)\) [5]. A classical conjugated gradient solver for sparse matrices has complexity \(O(\frac{e^{2N}}{\epsilon^6} N \log N)\) [5], where \(s \ll N\) is the sparsity of the stiffness matrix, \(\kappa = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}\) its condition number, and \(\epsilon\) the precision. The modified HHL is much more sensitive to the sparsity and condition number of the system matrix, but exponentially faster in the input size \(N\).

* Corresponding author: e-mail andre.mielke@isf.uni-stuttgart.de, phone +49 711 685 69544

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
2 Cantilever example problem

Since current hardware is limited to a small number of qubits, the problem size must be small enough to fit on available devices. Here, we used IBM’s "qasm simulator" and qiskit [7] to test the problem setup. This simulator uses a noise model reflecting the noisy environment on real hardware. The results are sampled from noisy versions of the quantum circuit.

We analyze a cantilever discretized into 4 truss elements, bounded by 5 nodes, fixed at node n1 and loaded at node n5 (cf. figure 1a). After applying the boundary condition, the reduced system is described by $K \in \mathbb{R}^{4 \times 4}$ and $\bar{f} \in \mathbb{R}^{4}$, so $\log_2(4\cdot 4) + \log_2(4) + \log_2(4) = 8$ qubits are needed. The whole circuit has a depth (total number of quantum gate operations) of 110, of which 58 gates are CNOTs, which are especially noisy. We use a simple Jacobi preconditioner $P^{-1} = \delta_{ij}$ to further improve the condition number of $K$. The deformed cantilever is depicted in figure 1b, comparing the analytical and the mean quantum-computed displacements along with the standard deviation for each nodal displacement. 100 noisy runs of the circuit were performed, summarized in table 1.

|        | $u_2$ | $u_3$ | $u_4$ | $u_5$ |
|--------|-------|-------|-------|-------|
| Classical | 0.15  | 0.3   | 0.45  | 0.6   |
| Quantum  | 0.17 ± 0.05 | 0.31 ± 0.04 | 0.44 ± 0.04 | 0.59 ± 0.04 |
| Deviation | 13.3% | 3.3%  | 2.2%  | 1.7%  |

Table 1: Nodal displacements and deviation of the quantum-computed mean displacements from the reference solution.

3 Summary

Figure 1b seems to show convincing agreement of the analytical and quantum-computed solutions, table 1 shows a deviation of the quantum-computed mean from the analytical solution of $\Delta u_2 = 13.3\%$. This is a significant error, although the analytical result lies inside the standard deviation. The standard deviation is high in all cases, ranging from 6.8% up to 29.8%. In a simple system like a cantilever, these errors do not seem catastrophic for the final result as long as the displacements are small, but they will accumulate for larger systems and cause noticeable deviations from the analytical solution. The main source of these errors is likely the high number of CNOT gates. A better approach might be to use a variational quantum eigensolver [8], which is only quadratically faster than classical algorithms, but less prone to noise by depending on fewer C NOT gates.

Acknowledgements  Open access funding enabled and organized by Projekt DEAL.

References

[1] M. Ying, Artificial Intelligence 174 (2), p. 162–176 (2010).
[2] K. Mitarai, M. Kitagawa, and K. Fujii, Phys. Rev. A 99, 012301 (2019).
[3] A. W. Harrow, A. Hassidim, and S. Lloyd, Physical review letters 113 (15), 150502 (2009).
[4] A. Y. Kitaev, Russian Mathematical Surveys, 52 (6) p. 1191–1249 (1997).
[5] S. Hestenes, Journal of Research of the National Bureau of Standards, 49, p. 409–436 (1952)
[6] A. M. Childs, R. Kothari, R. D. Somma, Communications in Mathematical Physics, 356, pp. 1057–1081. (2017)
[7] H. Abraham, A. Offei, I. Y. Akhalwaya, et al., Qiskit: An Open-source Framework for Quantum Computing (2019)
[8] A. Peruzzo, et al., Nature Communications, 5, 4213 (2014)