# 14 Matrix Inversion

May 17, 2021

**The HHL algorithm underlies many quantum machine learning protocols, but it is a highly nontrivial algorithm with lots of conditions. In this notebook, we implement the algorithm to gain a better understanding of how it works and when it works efficiently. The notebook is derived from the computational appendix of the paper [Bayesian Deep Learning on a Quantum Computer]. We restrict our attention to inverting a $2 \times 2$ matrix, following Pan et al.'s implementation(J. Pan and Du., 2014) of the algorithm.**

## A classical approach to solving the linear equations

We will start with an approach for solving the set of linear equations $Ax = b$ in the classical domain, and then make the transition to the quantum version.
Let $\{\nu_r\}$ be the eigenvectors of the matrix $A$ such:

$$A\nu_r = \lambda_r \nu_r, r \in \{1, 2, \dots, R\}$$

For non-singular $A$, i.e. none of the $\nu_r = 0$, we can write for the inverse $A^{-1}$:

$$A^{-1}\nu_r = 1/\lambda_r \nu_r, r \in \{1, 2, \dots, R\}$$

We can decompose any vector $z$ into $A$'s eigenvectors

$$z = \sum_{r=1}^{R} \left(\nu_r^T, z\right) \nu_r$$

Applying $A$ to $z$ yields:

$$Az = \sum_{r=1}^{R} \left(\nu_r^T, z\right) A\nu_r \tag{1}$$

$$= \sum_{r=1}^{R} \lambda_r \left(\nu_r^T, x\right) \nu_r \tag{2}$$

Likewise:

$$A^{-1}z = \sum_{r=1}^{R} \left(\nu_r^T, z\right) A^{-1}\nu_r \tag{2}$$

$$= \sum_{r=1}^{R} 1/\lambda_r \left(\nu_r^T, z\right) \nu_r \tag{3}$$

So in order to solve for $x$ in $Ax = b$ we substitute:

$$x = A^{-1}b \tag{3}$$

$$= \sum_{r=1}^{R} 1/\lambda_r \left(\nu_r^T, b\right) \nu_r \tag{4}$$

The quantum matrix inversion is based on the above expression. It finds the eigenvalues $\lambda_r$ and creates a representation of the term $\left(\nu_r^T, b\right) \nu_r$, for which we do not need an explicit expression. If we manage to get hold of the $\lambda_r' s$ and convert these into $1/\lambda_r$, then we can immediately compute eq. 4 In order to make the transition to the quantum algorithm we need one more identity.
When $A\nu_r = \lambda_r \nu_r$ then it is easily shown by Taylor expansion of the matrix exponent that:

$$e^A \nu_r = e^{\lambda_r} \nu_r$$

Hence, eqn (2) can be rewritten as:

$$e^A z = \sum_{r=1}^{R} \left(\nu_r^T, x\right) e^A \nu_r \tag{4}$$

$$= \sum_{r=1}^{R} e^{\lambda_r} \left(\nu_r^T, x\right) \nu_r \tag{5}$$

And more specifically

$$e^{2\pi i A} z = \sum_{r=1}^{R} \left(\nu_r^T, x\right) e^{2\pi i A} \nu_r \tag{5}$$

$$= \sum_{r=1}^{R} e^{2\pi i \lambda_r} \left(\nu_r^T, x\right) \nu_r \tag{6}$$

If A is Hermitian then $e^{2\pi i A}$ is a Unitary matrix, U, and we can write

$$Uz = \sum_{r=1}^{R} e^{2\pi i \lambda_r} \left(\nu_r^T, x\right) \nu_r$$

If $A$ is not Hermitian, then by a simple transformation we can create $A' = \begin{pmatrix} 0 & A^\dagger \\ A & 0 \end{pmatrix}$, which is Hermitian, and solve:

$$\begin{pmatrix} 0 & A^\dagger \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix},$$

## Quantum matrix inversion in 6 steps

We'll introduce the algorithm here in 6 short steps, that will be expanded and interspersed with the code that implements the algorithm

### Step 1 Compute the eigenvalues using the QPE primitive

We know from the previous chapter on quantum phase estimation that if we have an operator U working on one of its eigenvectors $|u\rangle$ where:

$$U|u\rangle = e^{2\pi i \theta}|u\rangle$$

then we can find $\theta$ with QPE in base vector encoding in an ancillary register as $|\theta\rangle$. Let's write this as a QPE primitive, $\hat{P}$, for this specific $U$ that works on 'empty' ancillaregister :

$$\hat{P}|00\ldots00\rangle|u\rangle = |\theta\rangle|u\rangle$$

Starting from the linear equations to solve $Ax = b$. If we choose $U$ to be equal to $e^{2\pi iA}$, then if $|u\rangle$ is an eigenvector of $A$ with eigenvalue $\lambda$, $|u\rangle$ is also an eigenvector of U. Additionally then, the phase angle $\theta$ is equal to $\lambda$. Using a shorthand $|k\rangle$ for the ancillary register $|00\ldots00\rangle$, we can thus write:

$$\hat{P}|k\rangle|u\rangle = |\lambda\rangle|u\rangle$$

Writing $|b\rangle$ as a decomposition in the eigenvectors $|u_i\rangle$ of $U$

$$\hat{P}|k\rangle|b\rangle \to \sum_i c_i|u_i\rangle|\tilde{\lambda}_i\rangle \text{ , } c_i = \langle u_i|b\rangle$$

In practice we will need to scale the problem with a factor T to avoid practical issues like the eigenvalues becoming larger than $2\pi$ causing wrap around on the unit circle and ambiguity in the solution. Effectively this means that we will work with:

$$U = e^{2\pi iA/T}$$

And correct for T at the end by classically dividing it out.

## Step 2 Compute the reciprocate eigenvalues

This step is often treated as trivial, but a good implementation will need to deal with this. Ultimately it requires a remapping of the eigenvalue register, as in

$$\sum_i c_i|u_i\rangle|\lambda_i\rangle \to \sum_i c_i|u_i\rangle|1/\lambda_i\rangle$$

.

## Step 3 Extract the eigenvalues

The eigenvalues, or in fact the reciprocate eigenvalues, are still in basis encoding, whereas we need the eigenvalues in amplitude encoding. This requires the application of a controlled rotation step:

$$\sum_i c_i|u_i\rangle|1/\lambda_i\rangle|a\rangle \to \sum_i c_i|u_i\rangle|1/\lambda_i\rangle\left(e_0|0\rangle + e_1/\lambda_i|1\rangle\right)$$

For now we write for the single bit ancillary register:

$$|a'\rangle = \xi(\lambda_i)|0\rangle + C/\lambda_i|1\rangle$$

where $C$ is a real constant,and $\xi(\lambda_i)$ is a function of the eigenvalues. As we will see $\xi(\lambda_i)$ will be removed through 'rejection sampling' at the end, and can be ignored.

## Step 4 Uncompute the reciprocate eigenvalues

Undo the mapping of the eigenvalue register from $|1/\lambda_i\rangle$ to $|\lambda_i\rangle$

$$\sum_i c_i|u_i\rangle|1/\lambda_i\rangle|a'\rangle \to \sum_i c_i|u_i\rangle|\lambda_i\rangle|a'\rangle$$

### Step 5 Uncompute the QPE operation using the inverse QPE primitive

As

$$\hat{P}|b\rangle|k\rangle = \sum_i |u_i\rangle\langle u_i|b\rangle|\lambda_i\rangle$$

Then:

$$\hat{P}|x\rangle|k\rangle = \sum_i |u_i\rangle\langle u_i| \left( \sum_{i'} 1/\lambda_{i'} |u_{i'}\rangle\langle u_{i'}|b\rangle \right) |\lambda_i\rangle \tag{6}$$

$$= \sum_i \sum_{i'} 1/\lambda_{i'} \langle u_i|u_{i'}\rangle\langle u_i|b\rangle|u_i\rangle|\lambda_i\rangle \tag{7}$$

$$= \sum_i 1/\lambda_i \langle u_i|b\rangle|u_i\rangle|\lambda_i\rangle \tag{8}$$

Hence

$$\hat{P}^{-1} \sum_i \langle u_i|b\rangle|u_i\rangle|\lambda_i\rangle|a'\rangle = \sum_i \langle u_i|b\rangle|u_i\rangle|k\rangle(\xi(\lambda_i)|0\rangle + C/\lambda_i|1\rangle) \tag{9}$$

$$= \sum_i \langle u_i|b\rangle|u_i\rangle|k\rangle\xi(\lambda_i)|0 > + \sum_i \langle u_i|b\rangle|u_i\rangle|k\rangle C/\lambda_i|1\rangle \tag{10}$$

$$\tag{11}$$

As

$$\sum_i 1/\lambda_i \langle u_i|b\rangle|u_i\rangle = |x\rangle$$

We can simplify the right hand term and write

$$\hat{P}^{-1} \sum_i \langle u_i|b\rangle|u_i\rangle|\lambda_i\rangle|a'\rangle = \sum_i \langle u_i|b\rangle|u_i\rangle|k\rangle\xi(\lambda_i)|0 > +C|x\rangle|k\rangle|1\rangle$$

We see that the right hand term is the answer that we are looking for, scaled with the known constant C. Note that the uncompute steps will return the ancillary register $|k\rangle$ back to its initial state $|00\ldots00\rangle$ and puts the solution vector $|x\rangle$ in the input register replacing $|b\rangle$ and using the same amplitude encoding.

### Step 6 Post select on the ancilla value

In the last step we perform a measurement of the ancila qubit and select only those outcomes where the ancilla $|a'\rangle = 1$. That leaves us with the correct values for $|x\rangle$, only to be corrected for $C$ and other scaling factors that have been applied in the implementation.
Figure 1 shows the overall schematics

## Bibliography

J. Pan, Y. Cao X. Yao Z. Li C. Ju H. Chen X. Peng S. Kais and J. Du. (Feb. 2014). “Experimental realization of quantum algorithm for solving linear systems of equations”. In: *Physical Review Letters* 89.89. URL: `https://arxiv.org/abs/1302.1946`.

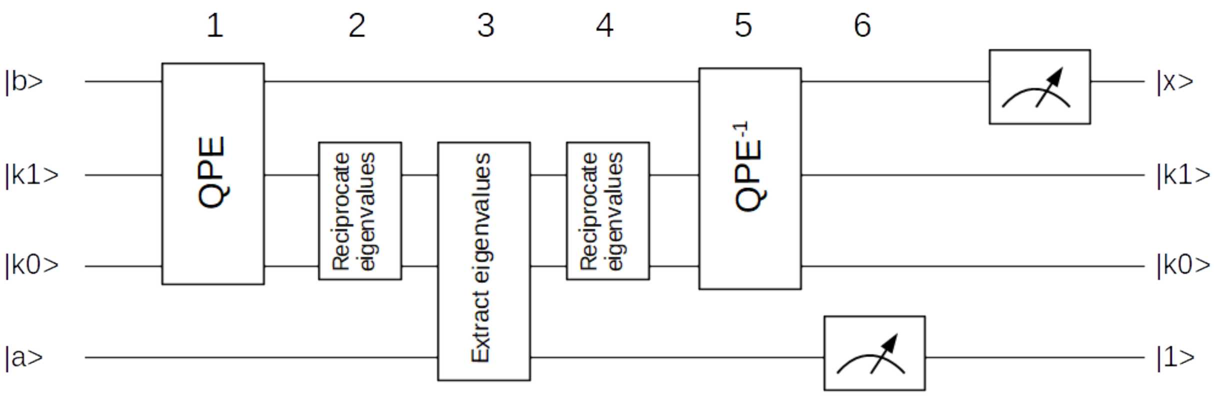


**Figure 1:** *Overall schema for quantum matrix inversion*