Quantum Algorithm for Linear Systems of Equations

Alfred Nguyen
Department of Computer Science
Technical University of Munich
05748 Garching, Bavaria
alfred.nguyen@outlook.com

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Ease of Use

A. Maintaining the Integrity of the Specifications

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

II. Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—LaTeX will do that for you.

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as "3.5-inch disk drive".
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: "Wb/m^2" or "webers per square meter", not "webers/m^2". Spell out units when they appear in text: ". . . a few henries", not ". . . a few H".
- Use a zero before decimal points: "0.25", not ".25". Use "cm^3", not "cc".)

C. Equations

Number equations consecutively. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

$$a + b = \gamma \tag{1}$$

Be sure that the symbols in your equation have been defined before or immediately following

D. LaTeX-Specific Advice

Please use "soft" (e.g., `\eqref{Eq}`) cross references instead of "hard" references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don't use the `{eqnarray}` equation environment. Use `{align}` or `{IEEEeqnarray}` instead. The `{eqnarray}` environment leaves unsightly spaces around relation symbols.

III. Template

Please note that the `{subequations}` environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you've discovered a new method of counting.

BibTeX does not work by magic. It doesn't get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

IV. Discussion

In this section, we evaluate the performance of the HHL algorithm. The most common known method to solve a system of linear equations is Gaussian Elimination. The time complexity of the Gaussian is $\mathcal{O}(N^3)$, which is drastically slower than other classical methods. As we are only interested in an estimate of an expectation value $|x\rangle M |x\rangle$, we can take a look of similar classical methods, considering other constraints.

In the following we will compare the HHL algorithm, to the classical conjugate gradient descent algorithm, focusing on time complexity and other factors that impact its efficiency.

A. Conjugate Gradient Descent

The classical conjugate gradient descent algorithm is very common method for solving linear systems of equations. It uses the conjugate direction method, which can locate the minimum of a function. By iteratively looking for solution vectors, the procedure converges toweards the solution of the linear system. This will provide us a suitable benchmark for analyzing the efficiency of the HHL algorihtm. Not only is it one of the fastest classical algorithms, it has very similar constraints set and calculates similar results (eg. $\vec{x}^\dagger M \vec{x}$).

B. Time Complexity

TABLE I
Time complexity comparison

Conjugate Gradien Descent	**HHL Algorithm**
$\mathcal{O}(\kappa s \log\left(\frac{1}{\epsilon}\right) N)$	$\mathcal{O}\left(\frac{\kappa^2 s^2}{\epsilon}\log N\right)$
$\Rightarrow \mathcal{O}(N)$	$\Rightarrow \mathcal{O}(log(N))$

As shown in the table, the HHL algorithm offers an exponential improvement compared to the conjugate gradient descent method. Conjugate gradient descent. Conjugate gradient descrent achieves a time complexity of $\mathcal{O}(N)$, whereas the HHL algorihtm runs in $\mathcal{O}(log(N))$.

Now we will take a more detailed look a the other factors involved in the timecomplexity, where:

- s is the s-sparsness
- κ is the condition number
- ϵ is the accuracy

s is the s-sparsness κ is the condition number ϵ is the accuracy

The sparsness s, describes to number of non-zero entries per row (e.g. a 2-sparse matrix only contains 2 non-zero entries per row). The condition number κ, describes how sensitiv the output of a function is on the error of the input. That means, a function is well conditioned if the output of a function does not change a lot for bigger errors in the input. Lastly, ϵ describes the accuracy of our desired solution. We can observer, that the sparsity s and condition number κ are quadratic in the HHL algorihtm, whereas in the conjugate gradient descent algorithm, both factors operate linearly. Furthermore, the accuracy ϵ is exponentially worse in the HHL algorithm. Thus, in terms of sparsity, accuracy and and condition number, the HHL algorithm a worse runtime over the conjugate gradient descent algorithm. If these prefactors where not worse than in the classical solution, this would have bizarre implications. One can show that if the dependency is better or equal than in the classical solution, one could solve *NP-Complete* problems in exponential faster times on quatum computers, which seems unlikely.

All in all, these constraints are important to consider when choosing applications, as they have a great effect on the runtime.