Exploring the Quantum Singular Value Transformation

Balint Pato

2021-12-08

Abstract

One of the major pain points in the field of quantum computing is the lack of new algorithms besides the three main types: quantum simulation, unstructured search (Grover's algorithm), and the hidden subgroup problem (underlying factoring, Simon's algorithm, etc.). Quantum Singular Value Transformation (QSVT)[1], a relatively new, promising framework offers a new perspective on quantum algorithms. Isaac Chuang calls it the "Grand unification of quantum algorithms" [7], as all the primordial quantum algorithms can be expressed using this framework. The hope is that QSVT will inspire new avenues and tools of algorithm research. In this project we explore QSVT, through numerical simulations and report on the results and difficulties along the way.

I Background

If we look at the quantum algorithms with proven speedups compared to best known classical algorithms on the Quantum Algorithm Zoo [4], we can see that the majority of them belong to the following main categories:

- Abelian Hidden Subgroup Problem - this is where Shor's algorithm for factoring falls
- Quantum Search - Grover's search and derivations
- Quantum Simulation - Hamiltonian simulation
- Linear Algebra problems - for example matrix inversion, HHL

.

Quantum Singular Value Transformation (QSVT) [1] is a framework that can encompass all of these algorithms. This exciting development is likely to create new avenues in algorithmic research. Martyn et al [7] wrote a pedagogical review of all these aspects. Here we report simulation results and challenges with parts of this paper.

The structure of this project is as follows: each section is a self-contained "project", it contains a brief summary of the theoretical introduction (given that [7] has most of the theory), the numerical experiments and results. Section I reproduces the results for single qubit experiments creating an inspiring connection to quantum signal processing and a multi-qubit experiment for oblivious amplitude amplification. Section II examines Hamiltonian simulations within the QSVT framework. Section III is an experiment to express a QAOA MaxCut experiment using QSVT. Finally, Section IV for a concluding remark, it is a discussion of the project.

II Introduction

The Quantum Singular Value Transformation [1] is an elegant framework to provide transformations of unitaries.

The *singular value decomposition* of a matrix $A \in \mathbb{C}^{m\times n}$ is defined as the product of two unitary matrices $W \in \mathbb{C}^{m\times m}, V \in \mathbb{C}^{n\times n}$, and a diagonal matrix $\Sigma \in \mathbb{R}^{m\times n}$, so that $A = W\Sigma V^{\dagger}$. The diagonal elements Σ_{ii} are the singular values of A. QSVT provides a way to embed certain polynomial transformations of the singular values of a matrix in a quantum computer. Informally, we'll embed the matrix A of interest in a unitary by "block encoding" and transform the unitary by interleaving it with certain processing operations to get the transformed version of $Poly(A)$, which has the same singular vectors but some polynomial transformation of its singular values:

$$\begin{pmatrix} \text{A} & \cdot \\ \cdot & \cdot \end{pmatrix} \rightarrow \begin{pmatrix} \text{Poly(A)} & \cdot \\ \cdot & \cdot \end{pmatrix} \tag{1}$$

This is surprisingly powerful, as a wide range of quantum algorithms can be expressed in this framework.

QSVT was built on top of many ideas and formalisms over the last decade. The final goal towards we'll build is to explore some of these fundamental building blocks, namely Quantum Signal Processing, Qubitization and Quantum Eigenvalue Transformation to motivate the discovery of QSVT.

II.1 Single qubit QSP

Theory

We start with Quantum Signal Processing (QSP) introduced by Low, Yoder and Chuang [6]. QSP's origins are rooted in the field of quantum control, where the experimentalist can mitigate certain types of control errors via interleaving the control pulses with sequence with pre-defined pulse sequences (like spin echo, BB1, TODO other examples from class). One particular example we'll look at here is the BB1 [8] sequence, which is of great interest in NMR spectroscopy as it leads to better image contrast.

We can gauge the most important features of QSP from QSVT point of view in a single qubit setting. QSP introduces the following main tenants: we define the signal unitary $W(a), a \in [1, -1]$ (typically a rotation around the X axis), a signal processing unitary $S(\phi), \phi \in \mathbb{R}$ (typically a rotation around the Z axis), a set of $d+1$ phases $\vec{\phi} = (\phi_0, \ldots, \phi_d) \in \mathbb{R}^{d+1}$. The final interleaved unitary, the *alternating phase modulation sequence* $U_{\vec{\phi}}$ is defined as:

$$U_{\vec{\phi}} = S(\phi_0)W(a)S(\phi_1)W(a)\ldots S(\phi_d) \tag{2}$$

This unitary will have the curious transformative properties we are interested in. Note that $W(a)$ is constant throughout, and only the $S(\phi)$ members are changing. We also define the "signal basis" to be either X or Z-basis, which determines the parts of the signal unitary to be transformed per our specification, which simply just means that we'll assume that our final unitary is applied to $|0\rangle$ in case of Z (or $|+\rangle$ for X basis), and we'll look at the overlap of the final state with $|0\rangle$ (or $|+\rangle$ for X basis).

The key message of QSP is that this structure of unitaries is surprisingly rich, and affords an interesting design space, as one can pick an *almost* arbitrary polynomial (with certain properties) $Poly$ and then it is possible to find angles $\vec{\phi}$, so that $Poly(a) = \langle 0| U_{\vec{\phi}} |0\rangle$ is satisfied, or $Poly(a) = \langle +| U_{\vec{\phi}} |+\rangle$ in case of the X notation.

QSP is used in multiple equivalent "conventions", which are important to know and recognize which particular algorithm is expressed in which convention, as they can influence the results significantly. The table below summarizes the QSP conventions covered in [7].

Convention	Signal	Signal Processor
W_x convention with Z signal basis: QSP($W_x, \langle 0| \cdot |0\rangle$)	$W_x(a) = e^{i\theta/2X} = \begin{pmatrix} a & i\sqrt{1-a^2} \\ i\sqrt{1-a^2} & a \end{pmatrix}$, where $\theta = -2cos^{-1}(a)$	$S_z(\phi) = e^{i\phi Z}$
W_x convention with X signal basis: QSP($W_x, \langle +| \cdot |+\rangle$)	$W_x(a) = e^{i\theta/2X} = \begin{pmatrix} a & i\sqrt{1-a^2} \\ i\sqrt{1-a^2} & a \end{pmatrix}$, where $\theta = -2cos^{-1}(a)$	$S_z(\phi) = e^{i\phi Z}$
Reflection convention: QSP($R, \langle 0| \cdot |0\rangle$)	$R(a) = Xe^{i\theta/2Y} = \begin{pmatrix} a & \sqrt{1-a^2} \\ \sqrt{1-a^2} & -a \end{pmatrix}$, where $\theta = -2sin^{-1}(a)$	$S_z(\phi) = e^{i\phi Z}$
used when needed, the W_z convention	$W_z(w) = e^{i\theta/2Z} = \begin{pmatrix} w & 0 \\ 0 & w^{-1} \end{pmatrix}$, where $w = e^{i\theta/2}$	$S_x(\phi) = e^{i\theta X}$

We will utilize in this project QSP($W_x, \langle 0| \cdot |0\rangle$), which can be thought of the single-qubit version of (1), however it has limitations on the polynomials that are possible to implement within it. We will explore QSP($W_x, \langle +| \cdot |+\rangle$) which supposed to be more expressive. Finally, the reflection convention will be valuable for the multi-qubit amplitude amplification example, and we will study the conversion between phases of W_x and R because the former is provided by the PyQSP package, and the latter will be crucial in understanding the multi-qubit amplitude amplification example.

To formally state the QSP theorem for QSP($W_x, \langle 0| \cdot |0\rangle$): TODO: rewrite for <0|0>

Theorem II.1. *QSP($W_x, \langle 0| \cdot |0\rangle$). A QSP phase sequence $\vec{\phi} \in \mathbb{R}^{d+1}$ exists such that*

$$Poly(a) = \langle +| e^{i\phi_0 Z} \prod_{k=1}^{d} W_x(a) e^{i\phi_k Z} |+\rangle \tag{3}$$

for $a \in [-1, 1]$ and for any real polynomial Poly $\in \mathbb{R}[a]$ if and only if the following conditions hold:

(i) $\deg(Poly) \leq d$

(ii) *Poly has parity $d \mod 2$*

(iii) $\forall a \in [-1, 1], |Poly(a)| \leq 1$

Where the parity of a polynomial $P(x)$ is defined as *even* when it only has non-zero coefficients for even powers of x, and *odd*, when it only has non-zero coefficients for even powers of x. The product is exactly the product we discussed in equation (2). Due to the X convention, we are moving a bit away from the "block-encoding" picture here, the element of the unitary that is transformed is the sum of all four elements as can be seen from this simple equation:

$$\langle +| \begin{pmatrix} a & b \\ c & d \end{pmatrix} |+\rangle = \frac{a+b+c+d}{2} \tag{4}$$

$$\tag{5}$$

It is also worth noting that there are theorems exactly stating the how the off-diagonal elements are transformed by a polynomial $Q(x)$, but we will ignore that angle for the moment for simplicity.

As we would like to transform Wx phases into R phases, we'll need a precise way to do that, which we will discuss explicitly in the following two theorems, as they are not fully covered or explained thoroughly in the papers.

Theorem II.2. *The reflection convention is equivalent to the W_x conventions and for a given phase sequence $\vec{\phi} \in \mathbb{R}^{d+1}$ in the W_x convention, we can find the corresponding $\vec{\phi'} \in \mathbb{R}^{d+1}$ by setting $\phi'_0 = \phi_0 - \frac{\pi}{4}, \phi'_d = \phi_d - \frac{\pi}{4}, \phi'_k = \phi_k - \frac{\pi}{2}, k \in [1, d-1]$, and then the corresponding alternating phase modulation sequence $U_{\phi'}$, is related as:*

$$U_{\vec{\phi}} = i^d U_{\vec{\phi'}}$$

Proof. The reflection convention is equivalent to the W_x conventions, by the following relationship:

$$W_x(a) = ie^{-i\frac{\pi}{4}Z} R(a) e^{-i\frac{\pi}{4}Z}. \tag{6}$$

Thus, given a phase sequence $\vec{\phi} \in \mathbb{R}^{d+1}$ in the W_x convention, we can find $\vec{\phi'} \in \mathbb{R}^{d+1}$ to bring our transformation to the same format. Using Eq.(6), it is easy to see that we'll need to tweak the two phases at the end by by the $\phi/4$ phase, and all the others by $\phi/2$, i.e. Then we have:

$$e^{i\phi_0 Z} \prod_{k=1}^{d} W_x(a) e^{i\phi_k Z} = e^{i\phi_0 Z} \prod_{k=1}^{d} ie^{-i\frac{\pi}{4}Z} R(a) e^{-i\frac{\pi}{4}Z} e^{i\phi_k Z} = \tag{7}$$

$$= i^d e^{i(\phi_0 - \frac{\pi}{4})Z} \left(\prod_{k=1}^{d-1} R(a) e^{-i\frac{\pi}{4}Z} e^{i\phi_k Z} e^{-i\frac{\pi}{4}Z} \right) R(a) e^{-i\frac{\pi}{4}Z} e^{i\phi_d Z} = \tag{8}$$

$$= i^d e^{i\phi'_0 Z} \prod_{k=1}^{d} R(a) e^{i\phi'_k Z}. \tag{9}$$

□

Note that, seemingly, there are miscalculations in both papers we are reviewing in this context with regards to this result: equation (A5) in [7], and the proof of Corollary 8 in the Arxiv pre-print [2], which is missing from the published version in [1]. They both account for the i phase using $ie^{i\phi_k Z} = e^{i\phi_k + \frac{\pi}{2}Z}$ which is not true in general. However, this argument works when we are in the Z basis and only looking at the $\langle 0| U |0\rangle$ element's transformation. This is what the next theorem is for, which will be very useful for the amplitude amplification case.

Theorem II.3. *Using the same definitions as Theorem II.2, in case of Z-basis quantum signal processing, we can find $\vec{\phi'} \in \mathbb{R}^d$ (i.e. one less element) reflection phases, that can be expressed as $\phi'_0 = \phi_0 + \phi_d + (d-1)\frac{\pi}{2}, \phi'_k = \phi_k - \frac{\pi}{2}, k \in [1, d-1]$, and then the corresponding alternating phase modulation sequence $U_{\phi'}$, is related through the final projection as:*

$$\langle 0| U_{\vec{\phi}} |0\rangle = \langle 0| e^{i(\phi'_0)} R(a) \prod_{k=1}^{d-1} e^{i\phi'_k Z} R(a) |0\rangle = Poly(a)$$

Proof. The main trick to recognize here is that $e^{i\phi Z}|0\rangle = \begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{pmatrix} |0\rangle = e^{i\phi}|0\rangle$, and similarly $\langle 0| e^{i\phi Z} = e^{i\phi}\langle 0|$, thus the first and the last signal processing terms can be brought to the front as global phases. Then from Theorem II.2 and Theorem II.1, with some rearranging we'll get the stated result:

$$Poly(a) = \langle 0| i^d e^{i\phi'_0 Z} \prod_{k=1}^{d} R(a) e^{i\phi'_k Z} |0\rangle = \tag{10}$$

$$= \langle 0| i^d e^{i(\phi_0 - \frac{\pi}{4})} R(a) \left(\prod_{k=1}^{d-1} e^{i\phi'_k Z} R(a) \right) e^{i(\phi_d - \frac{\pi}{4})} |0\rangle \tag{11}$$

$$= \langle 0| e^{id\frac{\pi}{2}} e^{i(\phi_0 - \frac{\pi}{4})} R(a) \left(\prod_{k=1}^{d-1} e^{i\phi'_k Z} R(a) \right) e^{i(\phi_d - \frac{\pi}{4})} |0\rangle \tag{12}$$

$$= \langle 0| e^{i(\phi_0 + \phi_d + (d-1)\frac{\pi}{2})} R(a) \left(\prod_{k=1}^{d-1} e^{i\phi'_k Z} R(a) \right) |0\rangle. \tag{13}$$

□

This will be useful for us, as the phases provided by the PyQSP [3] package we'll use, the W_x convention is the default, however, we'll need to transform them for the amplitude amplification example later to the reflection convention.

Numerical experiments

We are interested in reproducing the following results:

- To demonstrate the original QSP effect, we'll reproduce the effect of the BB1 sequence on a simple X rotation
- We will ensure that the Chebyshev polynomials T_1, T_2, T_3, T_4, T_5 are produced correctly. These have well defined phases for W_x: $T_1 : [0, 0], T_2 : [0, 0, 0], \ldots$, and as such provide great test cases.
- We will explore some more complex polynomials used later in the report, namely the polynomial approximation of the sign function for amplitude amplification and the fixed point search function polynomial
- Some polynomials will be produced in all the conventions we care about, namely QSP(W_x, $\langle +| \cdot |+\rangle$), QSP($W_x$, $\langle 0| \cdot |0\rangle$), QSP($R$, $\langle 0| \cdot |0\rangle$). This will help verifying the conversion logic between the reflection convention and the QSP(W_x, $\langle +| \cdot |+\rangle$) convention.

The code for the BB1 experiment can be found in the bb1.py file. The Cirq circuit for the BB1 transformed Rx rotation at theta $\pi/2$ is as follows:

```
q0: ──Rz(-0.58π)──Rx(0.5π)──Rz(1.16π)──Rx(0.5π)──Rz(0)──Rx(0.5π)──Rz(-1.16π)──Rx(0.5π)──Rz(0.58π)──Rx(0.5π)──Rz(-π)──
```

, where the line is broken for

The circuits for [0,0] are here for example:

```
QSP circuit for QSP([0, 0], conv=wx, basis=x) response
q: ──Rz(0)──Rx(-0.667π)──Rz(0)──
QSP circuit for QSP([0, 0], conv=r, basis=x) response
q: ──Rz(0.5π)──R(0.5)──Rz(0.5π)──
```

We used Cirq, design...todo...

In the repository: bb1.py, qsp.py TODO

Results

We successfully recreated the functions and the plots match that of the paper's and more. In Figure 1 we show the results of the BB1 sequence, where the signal unitary is implemented with the $cirq.Rx(rads = theta)$ gate, the signal is $\theta \in [-\pi, \pi]$. It is clear how the BB1 transformation "increases the stability" of the qubit $|0\rangle$ state and the sensitivity for the signal values around $\theta \approx 2\pi/3$.

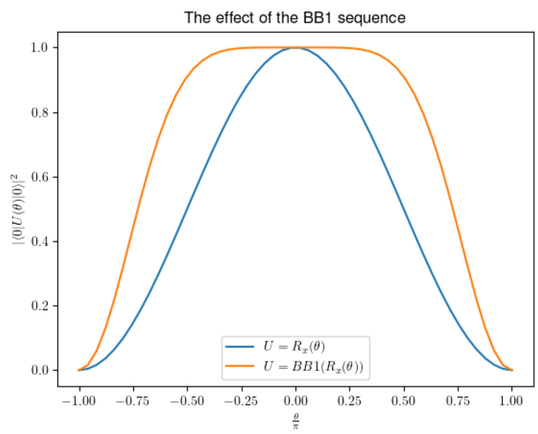
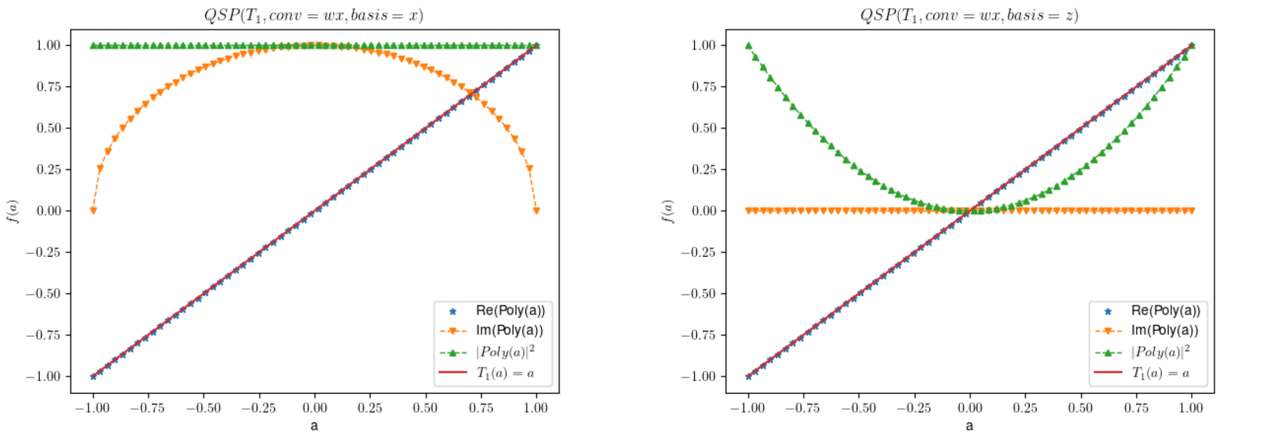


Figure 1: The effect of BB1 on R_x. Every point is the result of a simulation of a single BB1 sequence interleaved with a fixed θ rotation of $R_x(\theta)$.

Figures 2 and 3 show some of the resulting plots from the Chebyshev polynomial tests and that the real part of the QSP signal completely agrees with the target polynomials. Chebyshev polynomials have the great property that there are analytical proofs for them to work in QSP, and as such they serve as great test cases. They indicate that the logic in our single qubit QSP implementation is sound and in future development can be converted to test cases if the software package will get used in a library.

Finally, Figure 4 demonstrates the results for the application specific functions based on the angles defined in Appendix D.1. and D.2. in [7]. One aspect that was interesting is that the PyQSP package seems random for producing the phase angles for the sign function approximating polynomial. An issue[1] on the Gihub repository has been opened to clarify the behavior.

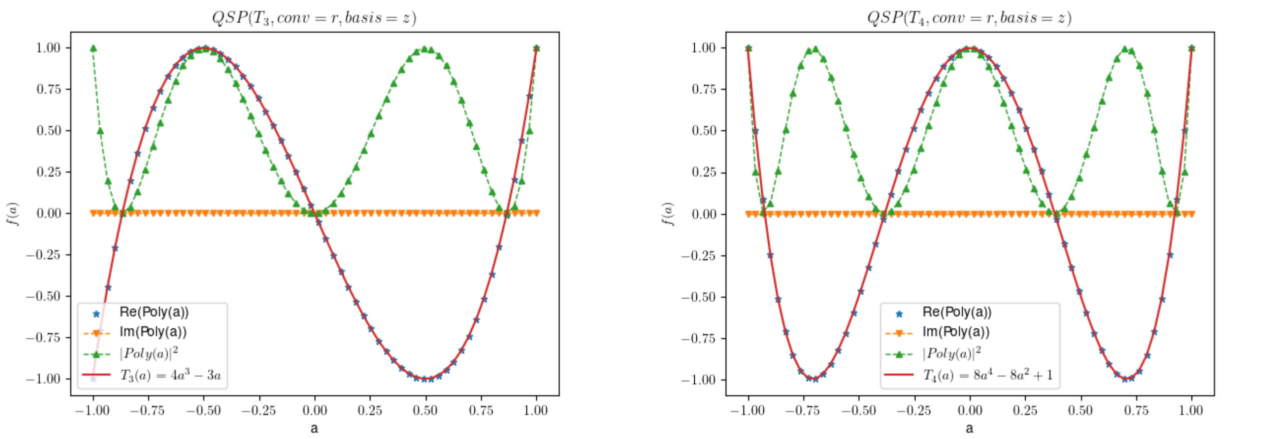
[1]https://github.com/ichuang/pyqsp/issues/1



(a) Poly(a)=a for the coefficients [0,0] for sanity checking in the X basis. The simulated points agree completely with the target polynomial.

(b) Poly(a)=a for the coefficients [0,0] for sanity checking in the Z basis. The simulated points agree completely with the target polynomial.

Figure 2: Sanity checking with the $T_1(x) = x$ Chebyshev polynomial



(a) $T_3 = 4x^3 - 3x$ Chebyshev polynomial, for the W_x signal $[0, 0, 0, 0]$, transformed to $QSP(R, \langle 0| \cdot |0\rangle)$ convention. The simulated points agree completely with the target polynomial.

(b) $T_4 = 8x^4 - 8x^2 + 1$ Chebyshev polynomial, for the W_x signal $[0, 0, 0, 0, 0]$, transformed to $QSP(R, \langle 0| \cdot |0\rangle)$ convention. The simulated points agree completely with the target polynomial.

Figure 3: Sanity checking with the higher order Chebyshev polynomials

II.2 Qubitization example

Theory

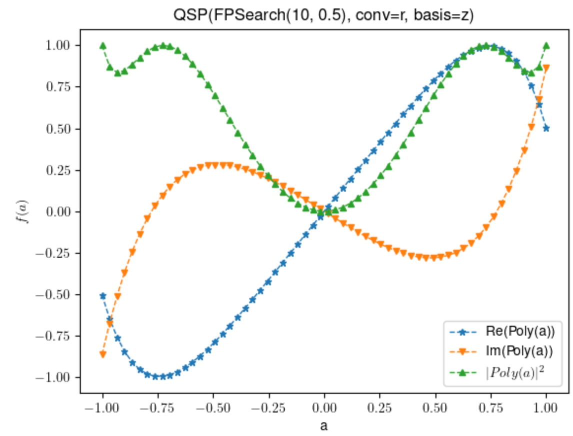
A natural way to generalize QSP to multi-qubit systems is by way of "qubitization" [5]. In problems where it is possible to identify one-qubit-like subsystems in a larger hilbert space, we can apply the QSP principles to create polynomial transformations of certain values. Specifically in this section we'll look at amplitude amplification - or "amplitude conversion", we'll pick an amplitude $a \in \mathbb{R}$ (we can assume it is real, as we can merge a potential phase into the starting state) of a unitary matrix and produce $Poly(a)$.

Problem input:

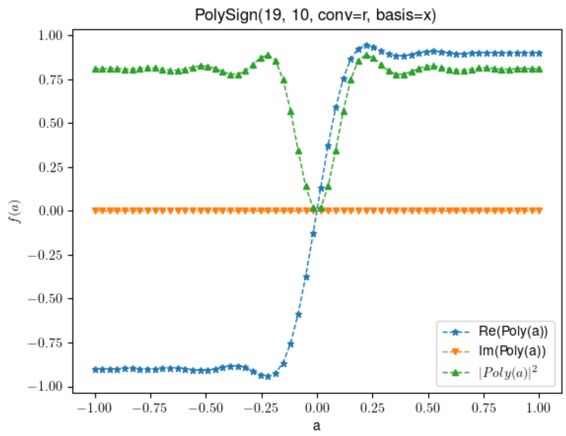
- a unitary U (which may act on some larger Hilbert space) with $a := \langle A_0| U |B_0\rangle \in \mathbb{R}$ initial value for two specific states, $|A_0\rangle$ and $|B_0\rangle$
- the inverse of our operation, $U^\dagger$
- $A_\phi = e^{i\phi\Pi_a}$, $B_\phi = e^{i\phi\Pi_b}$, where Π_a, Π_b are projectors to the subspace of $|A_0\rangle$ and $|B_0\rangle$ respectively. We choose these to be $\Pi_a = (2|A_0\rangle\langle A_0| - 1), \Pi_b = (2|B_0\rangle\langle B_0| - 1)$

Task: create a circuit Q using the elements above to create a final unitary that is $Poly(a)$.

This task, when $Poly(a) \to 1$, is known as *amplitude amplification*. It can be solved without knowing the specific initial value, a. The quantum algorithm we'll implement to solve amplitude amplification is called *fixed point amplitude amplification*. Using the idea of qubitization we'll show how this algorithm can be implemented using Theorem II.1 and the conversion theorems between W_x and R conventions.



(a) Oblivious amplitude amplification QSP response for fpsearch with degree 10 and $\delta = 0.5$. All points are simulated numerically using the $QSP(R, \langle 0| \cdot |0\rangle)$ convention.



(b) Sign function approximation using the Gauss error function erf with $d = 19$, $k = 10$. Every point is simulated numerically using the $QSP(R, \langle 0| \cdot |0\rangle)$ convention.

Figure 4: QSP response functions of application specific polynomials

The key insight is that U in the bases of $|A_0\rangle$ and $U|B_0\rangle$'s orthogonal component $|A_\perp\rangle$, and $|B_0\rangle, |B_\perp\rangle$ becomes a two qubit matrix:

$$U = \begin{array}{cc} & \begin{array}{cc} |B_0\rangle & |B_\perp\rangle \end{array} \\ \begin{array}{c} |A_0\rangle \\ |A_\perp\rangle \end{array} & \begin{bmatrix} a & \sqrt{1-a^2} \\ \sqrt{1-a^2} & -a \end{bmatrix} \end{array} \tag{14}$$

, where the following are the definitions for the perpendicular states:

$$|A_\perp\rangle = \frac{(I - |A_0\rangle\langle A_0|)\, U\, \langle B_0|}{\|\sqrt{(I - |A_0\rangle\langle A_0|)\, U\, \langle B_0|}\|} \tag{15}$$

$$|B_\perp\rangle = U^\dagger \left(a\, |A_0\rangle + \sqrt{1-a^2}\, |A_\perp\rangle \right) \tag{16}$$

An intuitive way of thinking about these subspaces is by way of visualizing the two concentric Bloch spheres as in Figure **??**, borrowed from [7]. Each pair of states spans a qubit space. The rotation operators A_ϕ and B_ϕ just become rotations around the Z axis for these Bloch spheres. As U has the format above, we can then use the QSP reflection convention! We will state this by way of a formal theorem:

Theorem II.4. *Amplitude amplification. Given unitary operators U and its inverse $U^\dagger$, and operators $A_\phi = e^{i\phi(2|A_0\rangle\langle A_0| - I)}$, $B_\phi = e^{i\phi(2|B_0\rangle\langle B_0| - I)}$, for a polynomial Poly of degree d, which satisfies the properties of Theorem II.1, there exist $\vec{\phi} \in \mathbb{R}^d$ that satisfies the following for even d:*

$$\langle A_0| \left[\prod_{k=1}^{d/2} U B_{\phi_{2k-1}} U^\dagger A_{\phi_{2k}} \right] U |B_0\rangle = Poly(a) \tag{17}$$

, and for odd d (TODO!)

$$\langle B_0| \left[\prod_{k=1}^{(d-1)/2} U B_{\phi_{2k-1}} U^\dagger A_{\phi_{2k}} \right] U |B_0\rangle = Poly(a) \tag{18}$$

TODO: describe how this is from the combo of experience, Martyn / Gilyen.

Numerical methods

- CNOT embedding for a - choice of states: |2>, |3> -

Results

- it seems that by calculating the rotation sequence on arbitrary unitaries, the Cirq simulator ends up with a non-unitary matrix due to numerical precision errors - 'cirq.Circuit.unitary' gives a non-unitary matrix!

III Discussion

We managed to reproduce some of the results of the QSVT framework via numerical simulations in Cirq.

Future plans...

References

[1] *A. Gilyén, Y. Su, G. H. Low, and N. Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In* Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, *STOC 2019, pages 193–204, New York, NY, USA, June 2019. Association for Computing Machinery. ISBN 978-1-4503-6705-9. doi: 10.1145/3313276.3316366. URL* `https://doi.org/10.1145/3313276.3316366`.

[2] *A. Gilyén, Y. Su, G. H. Low, and N. Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics.* Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, *pages 193–204, June 2019. doi: 10.1145/3313276.3316366. URL* `http://arxiv.org/abs/1806.01838`. *arXiv: 1806.01838.*

[3] *ichuang. Quantum Signal Processing, Oct. 2021. URL* `https://github.com/ichuang/pyqsp`. *original-date: 2021-03-22T12:00:56Z.*

[4] *S. Jordan. Quantum Algorithm Zoo, Feb. 2021. URL* `https://quantumalgorithmzoo.org/`.

[5] *G. H. Low and I. L. Chuang. Hamiltonian Simulation by Qubitization.* Quantum, *3:163, July 2019. ISSN 2521-327X. doi: 10.22331/q-2019-07-12-163. URL* `http://arxiv.org/abs/1610.06546`. *arXiv: 1610.06546.*

[6] *G. H. Low, T. J. Yoder, and I. L. Chuang. The methodology of resonant equiangular composite quantum gates.* Physical Review X, *6(4):041067, Dec. 2016. ISSN 2160-3308. doi: 10.1103/PhysRevX.6.041067. URL* `http://arxiv.org/abs/1603.03996`. *arXiv: 1603.03996.*

[7] *J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang. A Grand Unification of Quantum Algorithms.* arXiv:2105.02859 [quant-ph], *Aug. 2021. URL* `http://arxiv.org/abs/2105.02859`. *arXiv: 2105.02859.*

[8] *S. Wimperis. Broadband, Narrowband, and Passband Composite Pulses for Use in Advanced NMR Experiments.* Journal of Magnetic Resonance, Series A, *109(2):221–231, Aug. 1994. ISSN 1064-1858. doi: 10.1006/jmra.1994.1159. URL* `https://www.sciencedirect.com/science/article/pii/S1064185884711594`.