



THE UNIVERSITY OF
**WESTERN
AUSTRALIA**

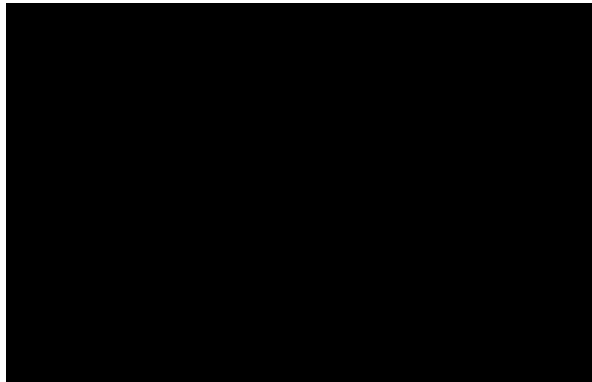
Continuous-time quantum walks: simulation and application

Joshua Izaac

BSc Physics (Hons)

Submitted: August 1, 2017

This thesis is presented for the degree of Doctor of Philosophy
of The University of Western Australia
School of Physics



Thesis Declaration

I, Joshua Izaac, certify that:

This thesis has been substantially accomplished during enrolment in the degree.

This thesis does not contain material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution.

No part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of The University of Western Australia and where applicable, any partner institution responsible for the joint-award of this degree.

This thesis does not contain any material previously published or written by another person, except where due reference has been made in the text.

The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

The work described in this thesis was funded by the Hackett Foundation, the National Natural Science Foundation of China (Grants No. 11474049 and No. 11674056), NS-FJS (Grant No. BK20160024), and the Scientific Research Foundation of the Graduate School of Southeast University (Grant No. YBJJ1623). Computational work was supported through the use of computing facilities provided by iVEC (the Western Australian Supercomputing Center: Partner Merit Allocation Project 772).

This thesis contains published work and/or work prepared for publication, some of which has been co-authored.

Signature:



Date:01/08/2017.....

Abstract

Quantum walks, the quantum analogy of the classical random walk, exhibit markedly different properties compared to their classical counterpart; taking advantage of quantum superposition and quantum correlations, they propagate across graph structures quadratically faster. They have since become an important tool in the field of quantum computation and quantum information theory, finding use in modelling complex quantum dynamical systems and providing a universal method of quantum computation. In this thesis, we consider continuous-time quantum walks across two areas of interest: efficient numerical simulation of the continuous-time quantum walk, and applications of continuous-time quantum walks to graph isomorphism and network centrality.

In Part I, we introduce the classical and quantum walks, as well as key graph theory concepts that will underpin later results. Then, in Part II, we begin by considering methods of efficiently simulating the matrix exponential using distributed memory computation. The results of our analysis leads to the presentation of the software package *pyCTQW*, which allows us to simulate multi-particle continuous-time quantum walks on arbitrary graphs more efficiently through the use of high performance computation. We follow with an analysis of fermionic continuous-time quantum walks, and show that by considering the set of symmetrized vertex states, the multi-particle fermionic quantum walk can be implemented by considering a single-particle quantum walk on a much reduced-vertex weighted graph.

We then consider quantum walk-based algorithms and applications in Part III, beginning with graph isomorphism; using the Bose-Mesner algebra, we verify that the single-particle continuous-time quantum walk cannot distinguish non-isomorphic strongly regular graphs. Further, we show that modifying the quantum walk to include a vertex-dependent phase factor does not provide any additional distinguishing power. Finally, we turn our attention to network centrality algorithms, and propose a centrality measure based on the continuous-time quantum walk. Through a statistical analysis on randomly generated graphs, we show that our proposed measure is highly correlated with the classical eigenvector centrality, and suggest that it provides an extension of the eigenvector centrality to the quantum realm. We then detail an experimental implementation of this centrality scheme on a 4-vertex star graph; to our knowledge, the first physical quantum centrality implementation. Next, we extend the continuous-time quantum walk to select directed networks via the use of the PT-symmetry framework, without any increase in the state space or resources required — unlike approaches such as the quantum stochastic walk or the Szegedy quantum walk. Lastly, we use this framework to augment our quantum centrality scheme to include directed networks, and show via statistical analysis that the centrality scheme continues to correlate well with classical measures.

Acknowledgements

First and foremost, I would like to thank my supervisors Jingbo Wang and Jie Pan. Jingbo, you have been an outstanding supervisor and mentor these past five years, and your constant support has enabled me to achieve the monumental task of compiling this body of work – especially during the few moments throughout my degree where I was sure my research was unravelling before my eyes! I would also like to thank my graduate research coordinator Paul Abbott, whose constant insight and guidance in all matters bureaucratic, grammatical, and mathematical saved me from a few near-scrapes, and for inspiring a whole chapter in this thesis with an informal post-meeting comment one late Friday afternoon.

Many thanks to our numerous collaborators, including Peng Xue, Anu Mahasinghe, Xiaosong Ma, Xiang Zhan, Jian Li, and Paul Abbott, whose input, discussions, and experimental know-how have greatly enriched this research. In addition, I'd like to acknowledge the Graduate Research School, Hackett Foundation, and The University of Western Australia for their financial support — without which, I would not have been able to visit their institutions and instigate these collaborations in the first place.

To Michael, Mitchell, and Kooper, thank you for keeping the office interesting on those long days of research. Thank you as well to Tania — visiting a rural university is much more exciting when you have someone willing to hitchhike cross-island with you. I'd like to especially thank Thomas for always being there to bounce ideas off (both research epiphanies and failures), and for introducing me to Satsuki takeaway. Our endless conversations both challenged me and lifted the quality of my research, so thank you. To my other PhD buddy Prue, thank you for keeping me sane over the last four years, including the numerous venting sessions — abandoning civilization to drive the Nullarbor on very little planning was a stroke of genius.

Thank you, Will, for the endless cups of tea, and the strangely productive writing sessions. You endured my slow descent into madness, and your country escape helped rejuvenate me when deadlines threatened to topple me.

Last, but certainly not least, I would like to thank my family, for all their love and support throughout my studies. Kath, thank you for the well-timed distractions and for always making me laugh. Thank you Mum for the constant support, even while having to put up with the neuroses archetypal of someone willing to enter graduate school on their own accord.

This research was supported by an Australian Government Research Training Program (RTP) Scholarship.

Statement of Contribution

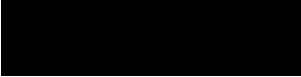
This thesis contains work that has been published and/or prepared for publication.

*J. A. Izaac and J. B. Wang, “pyCTQW: a continuous-time quantum walk simulator on distributed memory computers”, Computer Physics Communications **186**, 81–92 (2015).*

Location in thesis: chapter 4

Location of published manuscript in appendix: Appendix C.3


Student contribution to work: J. A. Izaac was the principal author of this article. J. A. Izaac performed the literature survey, devised the numerical simulation code, conducted the parallel scaling and speedup analysis, and implemented the plotting and graphical features. J. B. Wang provided overall guidance, supervision, and editing of the draft manuscript.

Coordinating supervisor signature: . **Date:** 01/08/2017....

J. A. Izaac and J. B. Wang, “Systematic dimensionality reduction for continuous-time quantum walks of interacting fermions”, Manuscript submitted for publication, 2017.

Location in thesis: chapter 5

Student contribution to work: J. A. Izaac was the principal author of this article. J. A. Izaac performed the literature survey, devised the formalism, algorithms, and numerical simulation code, conducted the numerical scaling analysis, and implemented the plotting and graphical features. J. B. Wang initiated the project and provided overall guidance, supervision, and editing of the draft manuscript.

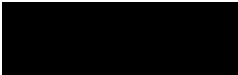
Coordinating supervisor signature: . **Date:** 01/08/2017....

A. Mahasinghe, J. A. Izaac, J. B. Wang, and J. K. Wijerathna, "Phase-modified CTQW unable to distinguish strongly regular graphs efficiently", Journal of Physics A: Mathematical and Theoretical **48**, 265301 (2015).

Location in thesis: chapter 6

Location of published manuscript in appendix: Appendix C.4

Student contribution to work: A. Mahasinghe was the principal author of this article. A. Mahasinghe devised the Bose-Mesner algebra method of analysing graph isomorphism algorithms, and used it to prove that a single particle CTQW with phase cannot distinguish non-isomorphic strongly regular graphs. J. A. Izaac performed the literature review, assisted in constructing the Bose-Mesner algebra proofs and phase-modified CTQW algorithms, and in drafting the manuscript. J. B. Wang initiated the project. J. B. Wang and J. K. Wijerathna provided overall guidance, supervision, and editing of the draft manuscript.

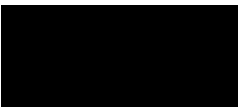
Coordinating supervisor signature:  **Date:** 01/08/2017.....

J. A. Izaac, X. Zhan, Z. Bian, K. Wang, J. Li, J. B. Wang, and P. Xue, "Centrality measure based on continuous-time quantum walks and experimental realization", Physical Review A **95**, 032318 (2017).

Location in thesis: chapter 7 and chapter 8

Location of published manuscript in appendix: Appendix C.6

Student contribution to work: J. A. Izaac was the principal author of this article. J. A. Izaac performed the initial literature survey, devised the centrality algorithm, performed the numerical simulations, carried out the statistical analysis, and drafted the manuscript. X. Zhan, Z. Bian, K. Wang, and J. Li performed the experiment, generated figures associated with the experimental results and set-up, and performed data analysis. X. Zhan and P. Xue drafted the experimental section of the manuscript. J. B. Wang initiated the project. J. B. Wang and P. Xue provided overall guidance, supervision, and editing of the draft manuscript.

Coordinating supervisor signature:  **Date:** 01/08/2017.....

J. A. Izaac, J. B. Wang, P. C. Abbott, and X. S. Ma, “Quantum centrality testing on directed graphs via PT-symmetric quantum walks”, arXiv:1607.02673v3 [quant-ph] (2017).

Location in thesis: chapter 9

Student contribution to work: J. A. Izaac was the principal author of this article. J. A. Izaac performed the initial literature survey, proposed and devised the pseudo-Hermitian CTQW framework, provided examples, analysed alternative interpretations of this framework, considered how it extends to multiple particles and interdependent networks, utilized the pseudo-Hermitian CTQW to construct a centrality algorithm on directed network, performed the numerical simulations and statistical analysis, and drafted the manuscript. J. B. Wang provided overall guidance, supervision, and editing of the draft manuscript. P. C. Abbott provided the inspiration behind this project, and feedback during editing of the manuscript. X. S. Ma provided feedback during editing of the manuscript with regards to future experimental implementation.

Coordinating supervisor signature: [redacted] **Date:** 01/08/2017

I, Jingbo Wang, certify that the student statements regarding their contribution to each of the works listed above are correct.

Coord. supervisor signature: [redacted] **Student signature:** [redacted]

Date: 01/08/2017 **Date:** 01/08/2017

List of Publications

The following list contains all work published, submitted, presented, and planned for publication throughout the candidature of my PhD.

Published Articles

- [1] Z. J. Li, J. A. Izaac, and J. B. Wang, “Position-defect-induced reflection, trapping, transmission, and resonance in quantum walks”, *Physical Review A* **87**, 012314 (2013).
- [2] J. A. Izaac, J. B. Wang, and Z. J. Li, “Continuous-time quantum walks with defects and disorder”, *Physical Review A* **88**, 042334 (2013).
- [3] J. A. Izaac and J. B. Wang, “pyCTQW: a continuous-time quantum walk simulator on distributed memory computers”, *Computer Physics Communications* **186**, 81–92 (2015).
- [4] A. Mahasinghe, J. A. Izaac, J. B. Wang, and J. K. Wijerathna, “Phase-modified CTQW unable to distinguish strongly regular graphs efficiently”, *Journal of Physics A: Mathematical and Theoretical* **48**, 265301 (2015).
- [5] S. S. Zhou, T. Loke, J. A. Izaac, and J. B. Wang, “Quantum fourier transform in computational basis”, *Quantum Information Processing* **16**, 82 (2017).
- [6] J. A. Izaac, X. Zhan, Z. Bian, K. Wang, J. Li, J. B. Wang, and P. Xue, “Centrality measure based on continuous-time quantum walks and experimental realization”, *Physical Review A* **95**, 032318 (2017).
- [7] A. C. H. Hurst, J. A. Izaac, F. Altaf, V. Baltz, and P. J. Metaxas, “Reconfigurable magnetic domain wall pinning using vortex-generated magnetic fields”, *Applied Physics Letters* **110**, 182404 (2017).

Submitted for publication

- [1] J. A. Izaac, J. B. Wang, P. C. Abbott, and X. S. Ma, “Quantum centrality testing on directed graphs via PT-symmetric quantum walks”, arXiv:1607.02673v3 [quant-ph] (2017).
- [2] J. A. Izaac and J. B. Wang, “Systematic dimensionality reduction for continuous-time quantum walks of interacting fermions”, Manuscript submitted for publication, 2017.

Selected conference papers

- [1] J. A. Izaac, X. Zhan, Z. Bian, K. Wang, J. Li, J. B. Wang, and P. Xue. A continuous-time quantum walk-based centrality measure, and its experimental realization. Talk presented at *The International Conference on Computational Modeling and Simulation*, Colombo, Sri Lanka. May 2017
- [2] J. A. Izaac, J. B. Wang, P. C. Abbott, and X. S. Ma. Quantum centrality testing on directed graphs via PT-symmetric quantum walks. Poster presented at *PHHQP16: Progress in Quantum Physics with Non-Hermitian Operators*, Kyoto, Japan. August 2016
- [3] J. A. Izaac and J. B. Wang. Pseudo-Hermitian continuous-time quantum walks on directed graphs. Talk presented at the *Complexity of Quantum Information and Computation Workshop*, Tsinghua Sanya International Mathematics Forum, Sanya, China. March 2016

Online publications

- [1] J. A. Izaac, “People take more risks when wearing helmets, potentially negating safety benefits”, *Science* (2016) [10.1126/science.aae0200](https://doi.org/10.1126/science.aae0200).
- [2] J. A. Izaac, *PT-symmetric quantum walks and centrality testing on directed graphs*, Wolfram Demonstrations Project, July 2016.

Contents

Abstract	ii
Acknowledgements	iii
Statement of Contribution	iv
List of Publications	vii
Contents	ix
List of Figures	xiii
List of Tables	xv
1 Introduction	1
I	BACKGROUND
2 Graph theory	8
2.1 Graph theory and definitions	8
2.1.1 Undirected graphs	8
2.1.2 Weighted graphs	18
2.1.3 Directed graphs	20
2.2 Important graph classes	21
2.2.1 Complete graphs	22
2.2.2 Vertex transitive graphs	22
2.2.3 Strongly connected graphs	23
2.2.4 Random graphs	24
3 Classical and quantum walks	28
3.1 Classical random walks	28
3.1.1 Discrete-time random walk	28
3.1.2 Continuous-time random walk	30
3.2 Quantum walks	30
3.2.1 Discrete-time quantum walks	31
3.2.2 Continuous-time quantum walks	33
3.2.3 Multiple-particle CTQWs	34

II	SIMULATION	37
4	Efficient numerical simulation	38
4.1	Introduction	39
4.2	Matrix exponential methods	40
4.2.1	Padé approximation	41
4.2.2	Scaling and squaring method	42
4.2.3	Krylov subspace methods	42
4.2.4	Chebyshev approximation	44
4.3	Software package	46
4.3.1	Dependencies and installation	46
4.3.2	Fortran library <code>libctqwMPI</code>	47
4.3.3	Python module <code>pyCTQW.MPI</code>	48
4.3.4	<code>pyCTQW</code> framework	51
4.4	Worked example	51
4.5	Scaling and performance	58
4.5.1	Library comparison	58
4.5.2	Parallel scaling	59
4.6	Conclusions	60
4.7	Acknowledgements	63
5	Multiple fermion simulation	64
5.1	Introduction	64
5.2	Indistinguishable particles	66
5.3	The antisymmetrized CTQW Hamiltonian	67
5.3.1	Example: two fermion walk on a 4-vertex star graph	70
5.4	Constructing the reduced fermionic adjacency matrix	73
5.5	Interacting fermions	75
5.6	Computational resources and analysis	80
5.7	Conclusion	80
III	APPLICATION	85
6	CTQW and graph isomorphism	86
6.1	Introduction	86
6.2	Graphs and matrix algebras	88
6.2.1	Bose-Mesner algebra	88

6.2.2	Strongly Regular Graphs	91
6.3	Quantum walk-based graph isomorphism algorithms	93
6.4	Single particle CTQW with phase	95
6.4.1	No phase	96
6.4.2	SV-phase addition	97
6.4.3	AV-phase addition	101
6.5	Multi-particle CTQW with interactions	103
6.5.1	GI framework and functions	104
6.6	Conclusion	105
7	Network centrality	108
7.1	Introduction	108
7.2	Classical centrality measures	109
7.2.1	Degree centrality	110
7.2.2	Eigenvector centrality	110
7.2.3	PageRank	111
7.2.4	Random walk centrality	112
7.3	Quantum centrality measures	113
8	CTQW centrality on undirected graphs	116
8.1	Introduction	116
8.2	CTQW-based centrality measure	117
8.3	Statistical analysis	120
8.3.1	Correlation to classical measures	120
8.3.2	Agreement on top-ranked vertices	126
8.4	Experimental realization	128
8.5	Conclusion	133
8.6	Acknowledgments	134
9	CTQW centrality on directed graphs	135
9.1	Introduction	135
9.2	CTQWs on directed graphs	137
9.3	PT-symmetry	138
9.4	Pseudo-Hermitian continuous-time quantum walks	142
9.4.1	3-vertex directed graph	145
9.4.2	Alternative interpretation	147
9.4.3	Multi-particle η -CTQW	149
9.4.4	Interdependent networks	150

9.5	Centrality testing	153
9.5.1	Proposed quantum scheme and examples	153
9.5.2	Random directed networks	158
9.5.3	Statistical analysis	161
9.6	Conclusion	165
9.7	Acknowledgments	166
10	Conclusion	167
	References	171
	Appendices	185
A	Linear algebra	186
A.1	Similar matrices have the same eigenvalues	186
A.2	An $n \times n$ matrix is diagonalizable iff it has n linearly independent eigenvectors .	187
A.3	All diagonalisable matrices have a biorthonormal eigenbasis	188
A.4	The redefined pseudo-Hermitian inner product preserves the η -CTQW norm . . .	190
B	Wolfram Demonstration	191
B.1	PT-symmetric Quantum Walks and Centrality Testing on Directed Graphs	192
C	Published articles	197
C.1	Position-defect-induced reflection, trapping, transmission, and resonance in quantum walks	198
C.2	Continuous-time quantum walks with defects and disorder	207
C.3	<i>pyCTQW</i> : A continuous-time quantum walks simulator on distributed memory computers	216
C.4	Phase-modified CTQW unable to distinguish strongly regular graphs efficiently .	229
C.5	Quantum Fourier transform in computational basis	242
C.6	Centrality measure based on continuous-time quantum walks and experimental realization	261
C.7	Reconfigurable magnetic domain wall pinning using vortex-generated magnetic fields	272

List of Figures

2.1	Two isomorphic 4-vertex undirected graphs	11
2.2	The undirected graph G_1 from Fig. 2.1 with a random orientation applied	13
2.3	Erdos-Renyi graph $G(25, 0.3)$ and degree distribution	26
2.4	Scale-free graph $N = 40, m = 2$, and degree distribution	27
4.1	Absolute value of the Chebyshev coefficients when calculating e^{At} for a 3×3 matrix A	45
4.2	3D and 2D output from pyCTQW of marginal probability of a two particle 3-cayley CTQW	56
4.3	Probability distribution of particle 1 vs time, plotted for various graph nodes.	57
4.4	Entanglement (represented by the Von Neumann entropy) of the two particle CTQW propagation over a 3-Cayley tree vs. time	57
4.5	CPU wall time for a two particle non-interacting CTQW on a finite line of N nodes propagated for t seconds, compared for a variety of software packages/algorithms.	59
4.6	Scaling behaviour of the Chebyshev and Krylov methods for simulating the CTQW on the finite line as number of vertices and number of processors is varied	61
4.7	Plots of speedup and efficiency vs number of processors for both the Chebyshev and Krylov methods when simulating a two particle interacting CTQW	62
5.1	4-vertex star graph	70
5.2	The cartesian product of the star graph with itself, as well as the reduced anti-symmetric 2-fermion star graph.	72
5.3	The reduced fermionic graph \tilde{G} representing a P -fermion CTQW on a 4-vertex star graph with nearest-neighbour interactions.	82
5.4	The reduced fermionic graph \tilde{G} representing a P -fermion CTQW on a 5-vertex cycle graph with nearest-neighbour interactions.	83
5.5	Time-scaling complexity for the reduced fermionic graph and Kronecker sum Hamiltonian construction	84
5.6	Qubits required to simulate a P -fermion CTQW walking on a graph of N vertices	84
8.1	CTQW probability on a 4-vertex star graph, starting from an equal superposition state.	119
8.2	Centrality measure comparison for a random Erdos-Renyi graph $G(20, 0.3)$	120
8.3	The CTQW vs. the PageRank/eigenvector centrality averaged over an ensemble of scale-free and Erdos-Renyi graphs	125
8.4	Chart showing the Jaccard set similarity between the CTQW centrality and classical PageRank/eigenvector centrality averaged over an ensemble of scale-free and Erdos-Renyi graphs	126

8.5	The quantum circuit for implementing the 4×4 unitary transformation U on a two-qubit system.	129
8.6	Conceptual experimental setup of the CTQW centrality on the 4-vertex star graph	130
8.7	Practical experimental setup of the CTQW centrality on the 4-vertex star graph .	130
8.8	Experimental photon probability distributions after eight unitary transformations	131
9.1	Non-unitary and pseudo Hermitian CTQW probability over time on a 3-vertex directed graph from initial superposition	146
9.2	Weighted graph equivalent of the η -CTQW	149
9.3	8-vertex pseudo-Hermitian interdependent network	153
9.4	4-vertex pseudo-Hermitian graph	155
9.5	Centrality measure comparison for a 7-vertex interdependent graph	157
9.6	Centrality measure comparison for a random Erdos-Renyi graph $G(15, 0.3)$ with bidirectional edges	158
9.7	Centrality measure comparison for a random Erdos-Renyi graph $G(20, 0.3)$ with unidirectional edges	159
9.8	Centrality measure comparison for a random 20-vertex in-degree scale-free graph	160
9.9	Centrality measure comparison for a random 100-vertex out-degree scale-free graph	161
9.10	PageRank, non-unitary CTQW, and η -CTQW centrality averaged over an ensemble of Erdos-Renyi and scale-free directed graphs	163
9.11	Jaccard set similarity between the PageRank and η -CTQW centrality ranks for various ensembles of Erdos-Renyi and scale-free graphs	164

List of Tables

4.1	Summary of parallel CTQW subroutines available in the Fortran <code>libctqwMPI</code> library	49
4.2	Summary of parallel CTQW classes available in <code>pyCTQW.MPI</code>	52
4.3	<code>pyCTQW.MPI</code> eigensolver keyword arguments	54
6.1	Frequency of all possible coefficients in the expansion of the selected vertex SV CTQW unitary operator for two non-isomorphic in the SRG(16,6,2,2) family.	102
6.2	Summary of parallel GI methods available in <code>pyCTQW.MPI.GraphISO</code> . For more details and required arguments, please refer to the online documentation.	105
8.1	Vigna's tau correlation coefficient for various centrality measures averaged over an ensemble of Erdos-Renyi graphs	123
8.2	The simplified sets of wave plates with indicated setting angles for realization of eight 4×4 unitary transformation	132
8.3	The setting angles of HWP ₁ and HWP ₂ for realization of the eight 4×4 unitary transformations	133
9.1	Centrality ranking of the vertices of 3-vertex graph Fig. 9.1(c), using the classical PageRank method, the non-norm conserving pseudo-Hermitian CTQW, and the probability conserving pseudo-Hermitian CTQW (η -CTQW)	155
9.2	Centrality ranking of the 4 vertices of graph Fig. 9.4, using the classical PageRank method, the non-probability conserving pseudo-Hermitian CTQW, and the probability conserving pseudo-Hermitian CTQW (η -CTQW)	155
9.3	Vigna's τ rank correlation coefficient compared for various classical (PageRank and eigenvector) and quantum (non-unitary CTQW and pseudo-Hermitian η -CTQW) centrality measures applied to the 7-vertex interdependent network in Fig. 9.5.	157

CHAPTER 1

Introduction

Since the seminal paper by Aharonov et al. [1], quantum walks have become a fundamental tool in quantum information theory, allowing us to bridge the often more esoteric world of quantum computation and algorithms with real life network theory and dynamical quantum modelling applications. The quantum analogue of the classical random walk, the astonishing abilities of the quantum walk are due, in part, to its markedly differing properties. Harnessing inherent quantum effects such as superposition, quantum coherence, and entanglement, the quantum walk propagates across structures quadratically faster than its classical counterpart, with propagation ballistic rather than diffusive [2, 3]. They also exhibit wave-like nature around defects and impurities, such as interference, resonance trapping and transmission, and can take advantage of multi-particle quantum entanglement [4–7]. As a result of their drastically different propagation properties, there has been an abundance of quantum walk-related research in the last decade, with quantum walk formulations motivating the creation of quantum algorithms that are faster and more efficient than their classical analogues [8, 9] — examples include network search and centrality [10–14], graph isomorphism analysis [5, 6, 15–18], and quantum chemistry simulation [19–24]. Furthermore, quantum walks on graphs have been proven to provide methods of universal computation [25, 26], allowing quantum walk-based systems to play a crucial role in the race to develop a quantum computer.

In order to ascertain an understanding of the advantage provided by the quantum walk to the field of quantum computation compared to the well established quantum circuit model, it is advantageous to briefly ‘retrace one’s steps’ and consider the classical random walk. The random walk performs random ‘steps’ between various states of the system, with the probability of transition between states independent of the past history of the walker — that is, the random walk is *memoryless*, and is referred to as a stochastic Markovian process. First coined by Karl Pearson in a 1905 letter to *Nature* [27, 28], the random walk has since found prestige in its ability to model countless systems within numerous fields, including physics (Brownian motion and diffusion [29]), biology (population dynamics [30,

31], genetic drift [32, 33], neuroscience [34]), economics (stock market modelling [35–37]), and network theory (graph isomorphism [38], network centrality [39, 40]), to name a few. Whilst the quantum walk is no longer a Markovian process (its time-evolution is instead governed by the time-reversible Schrödinger equation, and is now restricted to *symmetric* transitions between states due to the unitarity condition imposed by quantum mechanics), it retains the ability to naturally model systems that can be described as a collection of states with direction and history-independent transition probabilities. As a result, performing a quantum walk may allow us to extract useful information from the system, whilst taking advantage of the potential quantum speedup over classical algorithms. This has the potential to greatly increase the speed at which we can analyse underlying network structures.

With the ongoing demand for increasingly accurate models of complex biochemical processes that border the quantum regime (and thus exhibit some form of quantum coherence), this ability has further led to the investigation of quantum walkers as a viable modelling candidate [41]. So far, early research has explored their potential in analysing energy transport in biological systems [19–24], highlighting the growing versatility of quantum walks whilst providing new insights into the natural world. Most significantly, however, as quantum walks on graphs constitute a universal system of quantum computation [25, 42], any quantum walk-based algorithm or model can be recast in the quantum circuit model, and vice versa [13, 43–45]. The resulting quantum circuit reformulations can take advantage of the power afforded by constantly growing advances in theoretical and experimental quantum computation — consequently, this places the quantum walk in a commanding position, providing a new paradigm for the development of quantum algorithms. Quantum walks have therefore become an integral apparatus in linking network analysis and modelling with the problem solving potential of quantum computation. Furthermore, in recent years outside the theoreticians’ office, physical implementations of quantum walkers have been demonstrated experimentally, with approaches including the use of waveguides and photonics [46–48] and ion lattices [49–51]; paving the way for the construction of devices capable of running these quantum algorithms.

Note that, like the classical case, quantum walks are divided by two distinct approaches — the discrete-time quantum walk (DTQW), which introduces spin-states and a quantum coin operation with discrete time-evolution operators, and the continuous-time quantum walk (CTQW), which evolves the walker continuously in time [52]. Due to the enlarged Hilbert space and extra spin degrees of freedom of the DTQW, the relationship between these two formulations is inherently non-trivial; regardless, an equivalency has been explored using both a limiting approach [53] and percolation theory [54]. In this thesis, we

will be primarily concerned with the CTQW and its applications.

We begin by considering methods of efficient numerical simulation of CTQW. With the accelerating number of quantum walk algorithms and applications (especially those involving multiple walkers on increasingly large and complex graphs) and the current lack of readily available programmable quantum computers, the ability to efficiently simulate and analyse CTQW behaviour using classical computation becomes essential. As the time-evolution unitary operator is determined by the solution to the Schrödinger equation, this is analogous to requiring an efficient method of implementing the matrix exponential-vector product. Common approaches include Krylov-subspace techniques [55, 56] and the Padé approximation [57, 58]; the Chebyshev series method is another possible approach, but so far has largely been confined to the field of computational quantum chemistry [59–64]. Further complicating matters, the computational resources required to simulate multi-particle quantum walks increases exponentially with the number of interacting particles in the system. To address these issues, in chapter 4 we present a detailed investigation into the efficient simulation of the CTQW matrix exponential, with emphasis on the choice of algorithm and method of implementation. We then introduce *pyCTQW*, a distributed memory software package written in Fortran with a Python interface. By taking advantage of the impressive potential provided by modern High Performance Computing (HPC) platforms, *pyCTQW* allows efficient simulation of large multi-particle CTQW-based algorithms and systems. Moreover, we consider an additional approach to reducing computational resources required to simulate CTQWs of multiple interacting fermions in chapter 5; in particular, by removing ‘redundant’ and forbidden quantum states from consideration, we are able to significantly reduce the original graph structure of the walk. As a result, as the number of interacting fermions in the system increases, the computational resources required no longer increases exponentially.

Utilising these tools and methods developed for efficient classical simulation allows us to turn our focus towards CTQW-based applications and algorithms, and in particular, graph isomorphism and network centrality. An important open problem in mathematics and computer science, the graph isomorphism problem is concerned with efficiently determining if two differing graph labellings represent the different underlying graph structures. While its exact computational complexity remains a long standing question, a polynomial-time algorithm for determining graph isomorphism is conjectured to exist [65], and if found, would lead to dramatic advancement in fields such as cheminformatics [66]. The classical random walk has previously been considered as a candidate in developing an efficient graph isomorphism algorithm [38], and over the past two decades, the quantum speedup afforded by the quantum walk has motivated a dizzying array of quantum walk-based graph isomor-

phism algorithms proposals, both using the DTQW [15, 17, 67–69] as well as the CTQW [5, 6, 17, 70] — differing mainly in the number of particles involved, the presence of interactions, localised perturbations, and how the graph structures are compared. As of yet, however, no proposal has managed to solve the graph isomorphism problem completely, with suggested algorithms failing to distinguish various classes of large, highly symmetric and highly similar graphs. It has been shown that the DTQW has consistently higher distinguishing power, with the single-particle DTQW able to distinguish non-isomorphic graphs that are indistinguishable by single-particle CTQW algorithms, with distinguishing power increasing as the number of walkers increases. Furthermore, it has been proven that non-interacting multiple particle CTQWs are unable to distinguish particular families of non-isomorphic graphs [5, 6, 17]. In order to circumvent the increase in computational resources required by multiple particles, a perturbed single particle DTQW algorithm has been proposed allowing an increase in distinguishing power [16]. However, the CTQW would allow for an even more drastic reduction in computational resources, due to the lack of a quantum coin operation and reduced Hilbert space; it is therefore natural to ask, would a single-particle perturbed CTQW similarly allow for an increase in graph distinguishability? In chapter 6, we prove using the Bose-Mesner algebra that this is *not* the case — a single-particle CTQW perturbed in an analogous fashion to the perturbed DTQW algorithm *fails* to distinguish non-isomorphic graph families distinguished by the DTQW.

Thus, when searching for potential CTQW-based quantum algorithms, it is constructive to consider cases where the CTQW produces results on par or bettering the DTQW, allowing potential physical implementations a significant reduction in required computational resources. One such application we may consider is that of network centrality; the process of determining the nodes of a network more ‘central’ or ‘important’ to the overall network structure [71]. An immensely important topic in the field of network analysis and graph theory, the ability to determine the most central network nodes efficiently has wide-ranging applications across a range of interdisciplinary fields, due to the large number of physical systems that can be modelled as networks. For example, ranking websites for search engines [40], determining best practices for organisational management [72–74], analysing grooming networks in macaques [75], and finding active sites in protein molecules [76] all rely on the use of network centrality algorithms. Furthermore, a great number of network centrality algorithms are designed by utilizing classical random walks over graphs; including PageRank, arguably the most ‘famous’ of all centrality algorithms due to its role in the rise of the Google search engine. As such, quantum walks provide a natural method for translating these classical algorithms into the quantum realm, taking

advantage of the potential exponential speedups offered by quantum computation over the best known classical methods.

Indeed, in recent years several quantum centrality measures have been proposed — ranging from quantizations of the aforementioned classical measures to wholly new proposals — formulated using either the DTQW [12, 77] or the Quantum Stochastic Walk (QSW) [13, 78, 79], an approach that combines the CTQW with environmental dephasing. In chapter 8, we instead propose a quantum centrality measure based on the CTQW, and show via a large-scale statistical analysis that our proposed scheme is highly correlated with the classical eigenvector centrality, potentially providing a quantum analogue to the eigenvector centrality. Furthermore, due to a reduction in resources required compared to previous quantum centrality algorithms, we are able to experimentally demonstrate our quantum centrality scheme via linear optics, thus reporting the first successful physical implementation of a quantum centrality algorithm.

Unfortunately, the above quantum centrality algorithm suffers from its inability to consider non-Hermitian *directed* network structures; that is, transitions between states with a non-zero possibility of transition in one direction, but zero possibility of transition in the other. This arises due to the unitarity condition imposed by the quantum evolution of the CTQW, and is not a barrier in classical random walks — indeed, the PageRank algorithm, itself based on classical random walks, works equally well on both directed and undirected networks — a necessity for applications such as web analysis where hyperlinks are decidedly one-sided in nature. Consequently, various workarounds have been proposed for dealing with this non-unitary behaviour, for example Szegedy quantum walks [13, 77, 80] and the aforementioned QSW [13, 78]. However, both of these approaches have significant limitations when considering experimental implementation — each require a significantly expanded Hilbert space, whilst muting the effect of quantum behaviour (due to environmental dephasing).

A solution to this issue arises in the form of PT-symmetry, the observation that a system Hamiltonian need not require Hermiticity to ensure unitary time-evolution, but simply exhibit invariance under a combination of any linear operator \mathcal{P} and non-linear operator \mathcal{T} [81–85]. Building on the preliminary results of earlier investigations into PT-symmetric CTQWs [86], we therefore present in chapter 9 an independent and rigorous framework for PT-symmetric CTQWs, allowing the propagation of CTQWs on PT-symmetric directed graphs, whilst preserving the spectral information of the directed graph. Furthermore, we extend these results to interdependent networks, and show that this formalism is equivalent to an undirected, yet weighted, complete graph with self loops, providing a structural

interpretation that may aid in the design of simple physical implementation. As a result, we are able to generalize the CTQW centrality scheme to include PT-symmetric directed graphs, and we present a statistical analysis highlighting that, on directed graphs, this quantum centrality scheme continues to strongly agree with classical centrality measures such as PageRank.

This thesis is structured as a collection of papers (either published in a peer-reviewed journal or in the process of submission), each one reformatted to remove unnecessary repetition and to adhere to a common style. In Part I, we introduce key graph theory concepts in chapter 2, before introducing the classical and quantum walk in chapter 3. In Part II, we focus on efficient numerical simulation of CTQWs, with emphasis on classical computational tools such as matrix exponential algorithms and high performance computing in chapter 4, where the software package *pyCTQW* is detailed. We change track in chapter 5, detailing methods in which multi-fermion CTQWs may be simulated with substantial reduction in the computational resources required. This includes calculating the reduced antisymmetrised graph representation of the original graph under study, removing redundant and forbidden graph states.

Finally, in Part III, we consider CTQW algorithms and applications. In chapter 6, we prove via the Bose-Mesner algebra that the single-particle perturbed CTQW is unable to distinguish non-isomorphic pairs of strongly regular graphs belonging to the same family, before introducing concepts fundamental to network centrality in chapter 7. A CTQW-based network centrality algorithm is proposed in chapter 8, and an in-depth statistical analysis over an ensemble of random graphs is presented, highlighting the high correlation between our proposed quantum algorithm and the classical eigenvector centrality. We then present the results of a proof-of-concept physical implementation of the proposed CTQW quantum centrality scheme on a 4-vertex star graph, via linear optics. Then, in chapter 9, we introduce the field of PT-symmetry, and present a PT-symmetric CTQW framework that allows unitary time-evolution on directed graph structures. We show how this framework generalises for interdependent networks, and how it may be interpreted in terms of undirected and weighted graphs. Lastly, we extend the CTQW centrality scheme proposed in the previous chapter to PT-symmetric directed graphs, and provide a detailed statistical analysis on ensembles of random graphs, showing general strong agreement with classical centrality measures. The thesis then concludes in chapter 10.

PART I

BACKGROUND

CHAPTER 2

Graph theory

In this chapter, we will provide an overview of basic graph theory concepts, which will be required when analyzing quantum walks over graph structures in subsequent chapters.

2.1 Graph theory and definitions

Whilst quantum walks were initially introduced as propagating quantum particles in a discretised one-dimensional or two-dimensional position space [1], their analogy to classical random walks soon led to them being defined over *any* arbitrary graph structure [52, 87]. That is, the orthogonal position states are now represented by *vertices*, with transitions between states indicated by (undirected or directed) *edges*. As the quantum walk Hamiltonian now encodes general graph structure, the theorems and properties found in graph theory provide a useful foundation for analysing and characterising the quantum walk.

In this section, we start with an introduction to concepts common in graph theory, before applying these results in subsequent sections and chapters to quantum walks.

2.1.1 Undirected graphs

Consider an arbitrary undirected graph, denoted $G(V, E)$, composed of vertices (or nodes) $V = \{v_1, v_2, \dots, v_N\}$ and edges $E = \{e_1, \dots, e_M\}$ where $E \subseteq V \times V$. The graph consists of $|V| = N$ vertices, and the unordered pair $e_k = (v_i, v_j)$ represents an undirected edge connecting vertices v_i and v_j . The vertices are said to be *adjacent* when $(v_i, v_j) \in E$; this is denoted as $v_i \sim v_j$. Note that, without loss of generality, we will be assuming no isolated vertices — every vertex will have at least one other adjacent node. Using this as a starting point, we define some important graph properties and definitions below.

Definition 2.1 (Adjacency matrix). The adjacency matrix of an undirected graph $G(V, E)$ is an $N \times N$ matrix which fully describes the graph structure; its elements are given by

$$A_{ij} = \begin{cases} 1, & (v_i, v_j) \in E \\ 0, & (v_i, v_j) \notin E. \end{cases} \quad (2.1)$$

Using the adjacency matrix, we can also define the concept of vertex degree:

Definition 2.2 (Degree). The degree of a vertex v_i , denoted $d_i = d(v_i)$ where $d : V \rightarrow \mathbb{N}_0$, is the number of edges incident on vertex v_i . This can be calculated by taking the sum of the i th row of the adjacency matrix:

$$d_i \equiv d(v_i) = \sum_k A_{ik}. \quad (2.2)$$

Definition 2.3 (Degree matrix). The degree matrix D of an undirected graph $G(V, E)$ is an $N \times N$ diagonal matrix composed of the vertex degrees:

$$D = \text{diag}(d(v_1), d(v_2), \dots, d(v_N)) \quad \text{or equivalently} \quad D_{ij} = \delta_{ij} \sum_k A_{ik}. \quad (2.3)$$

Definition 2.4 (Degree regular graph). A graph $G(V, E)$ is denoted k -degree regular if every vertex v_i has degree k ; $d(v_i) = k \quad \forall v_i$. The degree matrix is then given by $D = kI$.

As the adjacency matrix for an undirected graph is real and symmetric (the edge represented by the unordered pair (v_i, v_j) is equivalent to the edge represented by unordered pair (v_j, v_i) , and thus $A_{ij} = A_{ji}$), it therefore satisfies Hermiticity ($A^\dagger = A$) and has a complete orthonormal eigenbasis with real eigenvalues.

Corollary 2.5. For a k -degree regular graph $G(V, E)$ with adjacency matrix A ,

$$AJ = JA = kJ \quad (2.4)$$

where J is the all one's matrix. That is, the $(1, 1, \dots, 1)$ is an eigenvector of A with eigenvalue k .

An important property of the adjacency matrix is that it allows us to analyze the *walks* present in a graph — an important characteristic that provides information regarding network flow.

Definition 2.6 (Walk). A walk of length ℓ between vertices v_i and v_j is a sequence of $\ell + 1$ adjacent vertices of the form $v_i \sim v_{s_1} \sim v_{s_2} \cdots \sim v_{s_{\ell-1}} \sim v_j$. Vertices may be repeated

along the sequence. If $v_i = v_j$, this is referred to as a *closed* walk, otherwise it is referred to as an *open* walk.

Definition 2.7 (Path). If a walk between vertices v_i and v_j has no repeating vertices between v_i and v_j , it is referred to as a path. If $v_i = v_j$, it is a *closed* path.

Consider the ij th element of A^ℓ , the ℓ th matrix power of the adjacency matrix:

$$(A^\ell)_{ij} = \sum_{s_1} \sum_{s_2} \cdots \sum_{s_{\ell-1}} A_{is_1} A_{s_1 s_2} \cdots A_{s_{\ell-1} s_j}. \quad (2.5)$$

For the summand to be non-zero, the only possibility is $A_{is_1} = A_{s_1 s_2} = \cdots = A_{s_{\ell-1} s_j} = 1$; this occurs only when $v_i \sim v_{s_1}$, $v_{s_1} \sim v_{s_2}$, and so on. As we are summing over all possible intermediate adjacent vertices s_m , we can therefore see that $(A^\ell)_{ij}$ provides the number of walks of length ℓ emanating from vertex v_i and ending at vertex v_j .

The definition of paths and walks allows us to further define several graph *distance* measures.

Definition 2.8 (Distance). The distance between two vertices v_i and v_j , denoted $\text{dist}(v_i, v_j)$, is defined as the length of the *shortest path* connecting the two vertices.

Definition 2.9 (Diameter). The diameter \mathcal{D} of a graph $G(V, E)$ is given by $\max(\{\text{dist}(v_i, v_j) : v_i, v_j \in V\})$; i.e. it is the ‘longest distance’ contained in the graph connecting any two vertices.

It should be noted that whilst an adjacency matrix fully describes the graph G , it is *not* unique; a relabelling of the vertices (which does not affect the underlying graph structure) results in a simultaneous permutation of the rows and columns of A . If two such graphs are identical under a vertex relabelling, they are referred to as *isomorphic*.

Definition 2.10 (Graph isomorphism). Two graphs G and H are isomorphic if there exists a bijection $f : V(G) \rightarrow V(H)$ between the two sets of vertices such that any two vertices u and v are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H .

Theorem 2.11. *Two graphs $G_1(V, E)$ and $G_2(V, E)$ are isomorphic if and only if there exists a permutation matrix P such that their respective adjacency matrices, A_1 and A_2 , are related via*

$$A_2 = P^{-1} A_1 P. \quad (2.6)$$

For example, consider the two graphs displayed in Fig. 2.1 — even though they are iden-

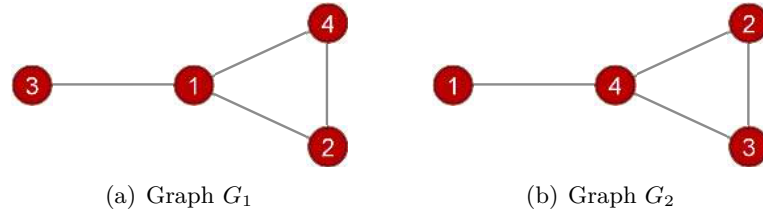


Figure 2.1: A 4-vertex undirected graph with two different vertex labellings. The graphs in (a) and (b) are isomorphic; to transform from (a) to (b), simply make the vertex label permutation $(1, 2, 3, 4) \rightarrow (4, 3, 1, 2)$.

tical, differing only in vertex labelling, Eq. 2.1 will provide two different adjacency matrices:

$$A_1 = A(G_1) = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \quad A_2 = A(G_2) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}. \quad (2.7)$$

It can easily be seen that $A_2 = PA_1P^{-1}$, where

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (2.8)$$

is the permutation matrix representing the vertex relabelling. Note that, as two isomorphic graphs have adjacency matrices related via a similarity transform, they share the same eigenvalues, determinant, and trace — these properties are isomorphic invariants. Furthermore, the eigenvectors corresponding to each eigenvalue will be preserved under vertex labelling permutation.

As an alternative to the adjacency matrix, we may also represent the graph structure using what is known as an *incidence matrix*.

Definition 2.12 (Incidence matrix). The incidence matrix of an undirected graph $G(V, E)$ is a $|V| \times |E|$ matrix M such that

$$M_{ij} = \begin{cases} 2, & \text{if } e_j = (v_i, v_i) \in E \\ 1, & \text{if } e_j = (v_i, v_k) \in E \text{ for some } k \in [1, N] \\ 0, & \text{otherwise.} \end{cases} \quad (2.9)$$

Unlike the adjacency matrix, which indicates vertex adjacency, the incidence matrix describes vertex-edge incidence; in other words, $M_{ij} = 2$ if edge e_j is a self-loop incident on vertex v_i , $M_{ij} = 1$ if edge e_j is incident on vertex v_i , and $M_{ij} = 0$ if edge e_j is *not* incident on vertex v_i . Therefore, the incidence matrix indicates which edges of the graph are incident on which vertices. Since every edge is incident on only two vertices, the oriented incidence matrix as defined by Eq. 2.9 has the property

$$\sum_{k=1}^N M_{kj} = 2. \quad (2.10)$$

For example, consider graph G_1 in Fig. 2.1(a); constructing the incidence matrix as per above, we have

$$M = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad (2.11)$$

where the rows correspond to vertices in V , the columns correspond to edges in E , and a non-zero element represents incidence between the respective vertex and edge.

A variant of the incidence matrix, known as the *oriented incidence matrix*, assigns a random direction to each edge — in effect, the edge $e_k = (v_i, v_j)$ becomes an *ordered pair*, with the edge e_k incident away from vertex v_i and towards vertex v_j . As such, the definition of the incidence matrix above is now modified to take into account this random graph orientation:

Definition 2.13 (Oriented incidence matrix). The oriented incidence matrix of an undirected graph $G(V, E)$ is a $|V| \times |E|$ matrix \tilde{M} such that

$$\tilde{M}_{ij} = \begin{cases} 2, & \text{if } e_j = (v_i, v_i) \in E \\ 1, & \text{if } e_j = (v_i, v_k) \in E \text{ for some } k \in [1, N] \\ -1, & \text{if } e_j = (v_k, v_i) \in E \text{ for some } k \in [1, N] \\ 0, & \text{otherwise.} \end{cases} \quad (2.12)$$

Following this definition, it can be seen that the column sum of an oriented incident matrix now sums to 0 rather than 2. To illustrate, consider graph G_1 from Fig. 2.1 again; choosing a random orientation for each of the edges (Fig. 2.2) and constructing the oriented

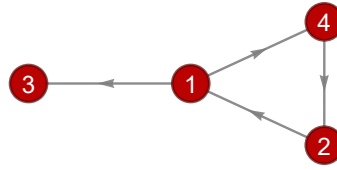


Figure 2.2: The undirected graph G_1 from Fig. 2.1 with a random orientation applied
incidence matrix,

$$\tilde{M} = \begin{bmatrix} 1 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}. \quad (2.13)$$

From hereon, the term incidence matrix and symbol M will refer to the *oriented* incidence matrix; the (unoriented) incidence matrix can then be denoted $|M|$.

Alongside the adjacency matrix and the incidence matrix, we introduce one final matrix of importance in graph theory; the (combinatorial) graph Laplacian or Laplacian matrix.

Definition 2.14 (Graph Laplacian). For an undirected graph $G(V, E)$ with adjacency matrix A and corresponding degree matrix $D = \sum_k A_{ik}$, the (combinatorial) graph Laplacian matrix is defined by

$$L = D - A \Leftrightarrow L_{ij} = \delta_{ij} \sum_k A_{ik} - A_{ij}. \quad (2.14)$$

Corollary 2.15. *The Laplacian is real and Hermitian: $L^\dagger = L$.*

As a result, it has a complete orthonormal eigenbasis $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\}$ with respective real eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_N\}$. It is therefore diagonalizable by the unitary matrix $Q_{ij} = (\mathbf{v}_i)_j$, resulting in $\Lambda = Q^T L Q$ where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$.

Lemma 2.16. *The Laplacian is positive semi-definite. As a consequence, it follows that the eigenvalues of L are real and non-negative:*

$$0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N. \quad (2.15)$$

Proof. Consider the quadratic form of the Laplacian, $\mathbf{x}^T L \mathbf{x} = \sum_i \sum_j x_i L_{ij} x_j$, where $\mathbf{x} \in \mathbb{R}^N \setminus \{0\}$ is an arbitrary column vector. Substituting in $L_{ij} = \delta_{ij} \sum_k A_{ik} - A_{ij}$ and

rearranging:

$$\begin{aligned}
\mathbf{x}^T L \mathbf{x} &= \sum_i \sum_j x_i L_{ij} x_j = \sum_i \sum_j x_i \left(\delta_{ij} \sum_k A_{ik} - A_{ij} \right) x_j \\
&= \sum_i \sum_j x_i^2 A_{ij} - \sum_i \sum_j x_i x_j A_{ij} \\
&= \frac{1}{2} \left(\sum_i \sum_j x_i^2 A_{ij} + \sum_i \sum_j x_j^2 A_{ji} \right) \\
&\quad - \frac{1}{2} \left(\sum_i \sum_j x_i x_j A_{ij} + \sum_i \sum_j x_i x_j A_{ji} \right) \\
&= \frac{1}{2} \sum_i \sum_j A_{ij} (x_i - x_j)^2 \\
&= \sum_{(v_i, v_j) \in E} (x_i - x_j)^2. \tag{2.16}
\end{aligned}$$

Since $(x_i - x_j)^2 \geq 0$ for all $(v_i, v_j) \in E$, it follows that $\mathbf{x}^T L \mathbf{x} \geq 0$ for all \mathbf{x} , and therefore L is positive semi-definite. \square

From the definition of the Laplacian, it can readily be seen that the diagonal elements of the Laplacian correspond to the vertex degrees, whilst the row and column sums of the Laplacian sum to zero,

$$\sum_k L_{ik} = \sum_k L_{ki} = 0. \tag{2.17}$$

This leads us to our next corollary:

Corollary 2.17. *The nullspace of L is given by the all one's vector $\mathbf{1} = (1, 1, \dots, 1)$, with respective eigenvalue $\lambda_1 = 0$.*

In general, the eigenspectrum of the Laplacian and the adjacency matrix have a highly non-trivial relationship; knowing properties of the eigenspectrum of one will not allow deduction of the equivalent properties for the other. However, for a degree regular graph G , the eigenspectrum of the adjacency matrix and the Laplacian are related via a linear transformation.

Lemma 2.18. *For a k -degree regular graph $G(V, E)$, the adjacency matrix A and graph Laplacian L share the same orthonormal eigenbasis \mathbf{v}_i ,*

$$A \mathbf{v}_i = \mu_i \mathbf{v}_i \quad \text{and} \quad L \mathbf{v}_i = \lambda_i \mathbf{v}_i, \quad i = 1, 2, \dots, N, \tag{2.18}$$

whilst the eigenvalues μ_i of the adjacency matrix are related to the eigenvalues λ_i of the Laplacian by

$$\lambda_i = k - \mu_i. \quad (2.19)$$

Proof. For a k -degree regular graph, we have $D = kI$, and thus $L = kI - A$. Recall that, as L is Hermitian, it is diagonalizable by $Q_{ij} = (\mathbf{v}_i)_j$, producing diagonal matrix $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$. Since D and A commute, they are simultaneously diagonalizable:

$$\Lambda = Q^T L Q = Q^T (kI - A) Q = kI - Q^T A Q = kI - M, \quad (2.20)$$

where $M = \text{diag}(\mu_1, \dots, \mu_N)$. □

Finally, it should also be noted that the Laplacian can also be constructed via the incidence matrix of the graph:

Lemma 2.19. *For an undirected graph $G(V, E)$ with oriented incidence matrix M ,*

$$L = M M^T. \quad (2.21)$$

Note that this result is independent of the random orientation chosen to compute M .

In the literature, the graph Laplacian is known by a dizzying variety of names; a quick scan will reveal papers referring to the same construction as the Laplacian matrix, discrete Laplacian, combinatorial Laplacian, graph Laplacian, Kirchoff matrix¹, amongst others. To see how the term ‘Laplacian’, more commonly associated with the Laplacian operator of vector calculus, came to be affiliated with this graph-theory matrix, let’s discretise the Laplacian operator over a 1D infinite grid.

In one-dimension, the Laplacian operator acting on a function $\phi(x)$ is given by

$$\nabla^2 \phi(x) = \frac{\partial^2}{\partial x^2} \phi(x). \quad (2.22)$$

Using the method of central finite differences, we can approximate this to second order by

$$\nabla^2 \phi(x) \approx \frac{(\phi(x+h) - \phi(x)) + (\phi(x-h) - \phi(x))}{h^2} + O(h^2), \quad (2.23)$$

¹Some readers may recognise the name Kirchoff from Kirchoff’s circuit laws. His name has also come to be attached to the Laplacian matrix, due to its use in Kirchoff’s matrix tree theorem [88].

where $h \ll 1$. We now discretise the spatial coordinate x over a finite grid of size N (x_1, x_2, \dots, x_N where $x_{i+1} - x_i = h$); this allows us to discretise the function $\phi(x)$ over the grid, $\phi_i = \phi_{x_i}$. The *discretised one-dimensional Laplacian* now takes the form

$$\nabla^2 \phi_i = \frac{(\phi_{i+1} - \phi_i) + (\phi_{i-1} - \phi_i)}{h^2} = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{h^2}, \quad (2.24)$$

or, in matrix notation,

$$\nabla^2 \phi = -\frac{1}{h^2} \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 \end{bmatrix} \phi. \quad (2.25)$$

The matrix in Eq. 2.25 is identical to the (combinatorial) graph Laplacian of a finite line of N vertices — thus, if we set $h = 1$ (signifying a distance of 1 between adjacent nodes in the graph), interpret ϕ as a function over the set of vertices ($\phi : V \rightarrow \mathbb{R}$), and let $\phi = (\phi(v_1), \dots, \phi(v_N))$, we see that

$$\nabla^2 = -L, \quad (2.26)$$

i.e. the negative graph Laplacian of a one-dimensional finite line is equal to the one-dimensional discrete Laplacian operator.

We can expand this result to any *arbitrary* undirected graph $G(V, E)$ as follows.

Theorem 2.20. *Consider the graph $G(V, E)$. Let $\phi : V \rightarrow \mathbb{R}^N$ denote a function over the vertex set V , with ϕ_i symbolizing the i th component of vector $\phi = (\phi(v_1), \dots, \phi(v_N))$. The action of the Laplacian operator acting on the vertex function ϕ_i is given by*

$$\nabla^2 \phi_i = \sum_{(v_i, v_j) \in E} (\phi_j - \phi_i) = - \sum_j L_{ij} \phi_j. \quad (2.27)$$

Proof. It can be seen that this is a generalization of Eq. 2.24 to account for all edges emanating from vertex i , and can be interpreted as summing the difference in ϕ across all vertices adjacent to v_i . Now, we know that $A_{ij} = 0$ if $v_i \not\sim v_j$ and 1 if $v_i \sim v_j$; thus, substituting this result in and rearranging:

$$\nabla^2 \phi_i = \sum_{j=1}^N A_{ij} (\phi_j - \phi_i) = \sum_{j=1}^N A_{ij} \phi_j - \phi_i \sum_{k=1}^N A_{ik}$$

$$\begin{aligned}
&= \sum_{j=1}^N A_{ij} \phi_j - \sum_{j=1}^N \phi_j \delta_{ij} \sum_{k=1}^N A_{ik} \\
&= \sum_{j=1}^N \left(A_{ij} - \delta_{ij} \sum_{k=1}^N A_{ik} \right) \phi_j \\
&= - \sum_{j=1}^N L_{ij} \phi_j. \tag{2.28}
\end{aligned}$$

Writing this in vector form, $\nabla^2 \phi = -L\phi$, and we can see that the negative graph Laplacian of an undirected graph $G(V, E)$ is equivalent to the discrete Laplacian operator acting on the graph. \square

In addition to the graph Laplacian discussed above, normalized variations also exist in the literature. In our case, we will only be concerned with the so-called *random walk normalized Laplacian*.

Definition 2.21 (Normalized graph Laplacian). For an undirected graph $G(V, E)$, the random walk normalized graph Laplacian is defined by

$$\tilde{L} = (D - A)D^{-1} \Leftrightarrow \tilde{L}_{ij} = \begin{cases} 1, & i = j \\ -\frac{1}{d(j)}, & (v_i, v_j) \in E \\ 0, & (v_i, v_j) \notin E. \end{cases} \tag{2.29}$$

That is, the normalized Laplacian can be constructed by post-multiplying the combinatorial graph Laplacian by the inverse of the degree matrix². Since we are only considering graphs with no isolated vertices ($d(v_i) = 0$), the diagonal matrix D will contain no zeros along the diagonal, ensuring $|D| \neq 0$ and that \tilde{L} is well-defined.

As might be guessed from its name, the random walk normalized Laplacian is used extensively when studying classical random walks, as it is inextricably linked to the random walk transition matrix. This property allows it to act as a stochastic transition matrix for continuous-time random walks, which we will cover in detail in chapter 3.

Note that, in a similar fashion to Eq. 2.27 we can also write the normalized Laplacian as an

²We adopt this convention due to our use of column probability vectors in subsequent section. If we were instead to use row probability vectors, we would define the random walk normalized Laplacian by $\tilde{L} = D^{-1}L$ to ensure row stochasticity for random walks.

operator acting on a function $\phi : V \rightarrow \mathbb{R}$ defined over the vertex set, with $\phi_i = \phi(v_i)$:

$$\tilde{\nabla}^2 \phi_i = - \sum_{ij} \tilde{L}_{ij} \phi_j = \sum_{(v_i, v_j) \in E} \left(\frac{1}{d_j} \phi_j - \frac{1}{d_i} \phi_i \right), \quad (2.30)$$

i.e. the normalized Laplacian acting on ϕ_i measures the sum of the differences in ϕ , divided by vertex degree, across all vertices adjacent to v_i .

From here onwards, to avoid confusion, the terms ‘graph Laplacian’, ‘discrete Laplacian’, ‘Laplacian matrix’, and simply ‘Laplacian’ will be used to refer to the un-normalized combinatorial graph Laplacian L .

2.1.2 Weighted graphs

Weighted undirected graphs are an extension of the framework detailed above, this time with real-valued weights permitted to be assigned to each edge in the graph:

Definition 2.22 (Weighted graph). A weighted, undirected graph $G(V, E, W)$ is composed of a set of vertices $V = \{v_1, v_2, \dots, v_N\}$, edge set $E = \{e_1, e_2, \dots, e_M\} \subseteq V \times V$, and edge weights $W = \{w_1, w_2, \dots, w_M\} \subseteq V \times V$. The unordered pair $e_k = (v_i, v_j)$ represents an undirected edge between v_i and v_j , whilst $w_k = w(v_i, v_j)$, $w_k \in \mathbb{R}$ represents a non-zero weight placed on the corresponding edge e_k .

Note that this is a generalization of the previous section — letting $W = \{1, 1, \dots, 1\}$ in Def. 2.22 reduces us back to the case of the unweighted, undirected graph. As such, most of the concepts and properties we saw can be extended to the weighted graph quite trivially, with only a few major subtleties relevant to this work extended upon³. In particular, the definition for the adjacency, Laplacian, and incidence matrices remains the same for weighted graphs, and are simply complemented by the addition of the weighted adjacency matrix and weighted Laplacians, and the concept of weighted degree.

Definition 2.23 (Weighted adjacency matrix). The weighted adjacency matrix of an undirected graph $G(V, E, W)$ is an $N \times N$ matrix which fully describes the graph structure;

³Although it is possible to introduce graph theory concepts starting from weighted graphs, and later consider unweighted graphs as a special case, in general this is not done. This is mostly due to a mix of historical convention, and the fact that various important theorems (e.g. Perron-Frobenius theorem) can be applied to adjacency matrices composed of solely ones and zeros.

its elements are given by

$$A_{ij}^{(w)} = \begin{cases} w_k, & e_k = (v_i, v_j) \in E \\ 0, & (v_i, v_j) \notin E. \end{cases} \quad (2.31)$$

Analogously to the (unweighted) adjacency matrix, we can then also define the concept of weighted vertex degree:

Definition 2.24 (Weighted degree). The weighted degree of a vertex v_i , denoted $d_i^{(w)} = d^{(w)}(v_i)$ where $d^{(w)} : V \rightarrow \mathbb{N}_0$, is the sum of the edge weights incident on vertex v_i . This can be calculated by taking the sum of the i th row of the weighted adjacency matrix:

$$d_i^{(w)} = d^{(w)}(v_i) = \sum_k A_{ik}^{(w)}. \quad (2.32)$$

Definition 2.25 (Weighted degree matrix). The weighted degree matrix $D^{(w)}$ of a graph $G(V, E, W)$ is an $N \times N$ diagonal matrix composed of the weighted vertex degrees:

$$D^{(w)} = \text{diag}(d^{(w)}(v_1), d^{(w)}(v_2), \dots, d^{(w)}(v_N)) = \delta_{ij} \sum_k A_{ik}^{(w)}. \quad (2.33)$$

The weighted Laplacian can now be defined in a similar fashion to how the unweighted graph Laplacian is calculated:

Definition 2.26 (Graph Laplacian). For a weighted graph $G(V, E, W)$ with weighted adjacency matrix $A^{(w)}$ and corresponding weighted degree matrix $D^{(w)} = \sum_k A_{ik}^{(w)}$, the weighted graph Laplacian matrix is defined by

$$L^{(w)} = D^{(w)} - A^{(w)} \Leftrightarrow L_{ij}^{(w)} = \delta_{ij} \sum_k A_{ik}^{(w)} - A_{ij}^{(w)}. \quad (2.34)$$

We now state, without proof, the following well-established relationship between the oriented incidence matrix of a weighted graph, and its corresponding weighted Laplacian:

Lemma 2.27. For a weighted graph $G(V, E, W)$ with oriented incidence matrix M ,

$$L^{(w)} = MWM^T, \quad (2.35)$$

where $W = \text{diag}(W)$ is a diagonal matrix containing the edge weights.

Finally, as an aside and for some physical intuition, note that the action of the weighted Laplacian on a function defined over a graph's vertex set can be interpreted as a discrete, weighted, Laplacian operator:

Theorem 2.28. Consider the weighted graph $G(V, E, W)$. Let $\phi : V \rightarrow \mathbb{R}^N$ denote a function over the vertex set V , with edge weight $w_k = w(v_i, v_j)$ corresponding to edge $e_k = (v_i, v_j)$. Let ϕ_i symbolize the i th component of vector $\phi = (\phi(v_1), \dots, \phi(v_N))$. The action of the weighted Laplacian matrix acting on the vertex function ϕ_i is given by

$$\sum_j L_{ij}^{(w)} \phi_j = - \sum_{(v_i, v_j) \in E} w(v_i, v_j) (\phi_j - \phi_i). \quad (2.36)$$

2.1.3 Directed graphs

So far, we have considered undirected graphs — both weighted and unweighted. In this section, we will now introduce the concept of *directed* graphs, and we will later see that this category poses a significant challenge to the quantum walk framework.

Definition 2.29 (Directed graph). A directed graph $G(V, E)$ is composed of a set of vertices $V = \{v_1, v_2, \dots, v_N\}$ and edge set $E = \{e_1, e_2, \dots, e_M\} \subseteq V \times V$, with the *ordered pair* $e_k = (v_i, v_j)$ representing a directed edge incident away from v_i and toward v_j .

Now that we are representing edges as ordered pairs of vertices, we see that the direction of the edge is now vital to describing the graph structure; $(v_i, v_j) \neq (v_j, v_i)$. Calculating the adjacency matrix of the directed graph (Eq. 2.1), it follows that the adjacency matrix of a directed graph is no longer symmetric — A_{ij} is not necessarily equal to A_{ji} — and thus A is no longer an Hermitian matrix. As such, it cannot be guaranteed that A has a complete orthonormal eigenbasis with real eigenvalues.

Along with the now non-symmetric adjacency matrix, directed graphs also inherit a couple of subtleties with regards to vertex degree; it is now possible to define both vertex *in-degree* and *out-degree*.

Definition 2.30 (In-degree). The in-degree of vertex v_i , denoted $d_i^- = d^-(v_i)$ where $d^- : V \rightarrow \mathbb{N}_0$, is the number of edges incident toward vertex v_i . It can be calculated from the row sums of the adjacency matrix:

$$d_i^- \equiv d^-(v_i) = \sum_k A_{ik}. \quad (2.37)$$

Definition 2.31 (Out-degree). The out-degree of vertex v_i , denoted $d_i^+ = d^+(v_i)$ where

$d^+ : V \rightarrow \mathbb{N}_0$, is the number of edges incident away from vertex v_i . It can be calculated from the column sums of the adjacency matrix:

$$d_i^+ \equiv d^+(v_i) = \sum_k A_{ki}. \quad (2.38)$$

Corollary 2.32. *The total degree $d_i = d(v_i)$ of vertex v_i is given by $d_i = d_i^+ + d_i^-$.*

For a directed graph, the construction of the incidence matrix M remains the same as in the oriented incidence matrix of an undirected graph (Eq. 2.12), with the addendum that we no longer need to randomly assign direction to the already-directed edges.

However, when we try and extend the Laplacian matrix to the directed graph, we start to run into several issues. For example, lets naïvely assume our previous undirected Laplacian matrix definitions extend without modification to the directed graph case:

$$L = D - A \Leftrightarrow L_{ij} = \delta_{ij} \sum_k A_{ik} - A_{ij} \Leftrightarrow L = MM^T. \quad (2.39)$$

Instantly, we'll notice ambiguities and false-equivalences. Do we choose D to be the diagonal matrix of total vertex degrees, in-degrees, or out-degrees? Kirchoff's matrix tree theorem would imply a diagonal matrix of in-degrees [88, 89], but calculating $L = MM^T$ produces the *total* vertex degree along the diagonal. Further muddling the picture, the directed graph Laplacian is not as well-studied as its undirected counterpart; studies constructing the directed Laplacian using vertex out-degree [90], in-degree [91], and total degree [92] coexist alongside one-another in the literature.

Thus, in order to follow conventions laid out in prior quantum walk studies [86], in this manuscript we will adopt the convention of utilizing *in-degrees* along the diagonal of the Laplacian.

Definition 2.33 (Directed graph Laplacian). For a directed graph $G(V, E)$ with adjacency matrix A , the directed graph Laplacian matrix is defined by

$$L_{ij} = \delta_{ij} \sum_k A_{ik} - A_{ij}. \quad (2.40)$$

2.2 Important graph classes

We have already seen that graphs may exhibit degree regularity (Def. 2.4) — the set of graphs satisfying this property form the class of degree regular graphs, and as a result

of this shared property, further characteristics common to this class may be deduced or proven. In a similar vein, numerous other classes of graphs all sharing unique characteristics exist. In this section, let us introduce a couple of useful graph classes, properties, and theorems that will be used throughout this thesis.

2.2.1 Complete graphs

The complete graphs are the class of graphs where every possible pair of vertices are adjacent.

Definition 2.34 (Complete graph). The complete undirected graph K_N is an undirected graph of N vertices $V = \{v_1, v_2, \dots, v_N\}$ and $N(N-1)/2$ edges $E = \{e_1, e_2, \dots, e_{N(N-1)/2}\}$ such that $v_i \sim v_j$ for all pairs of $v_i, v_j \in V$.

The total number of edges in the complete graph can be easily shown by considering that the edge set consists of every possible unordered pair of vertices; $|E| = \binom{N}{2} = N(N-1)/2$. Moreover, by the above definition, complete graphs are also members of the degree-regular class of graphs, with each vertex having degree $d = N - 1$.

Corollary 2.35. *The (unweighted) adjacency matrix of the complete graph K_N is given by $A = J - I$.*

Furthermore, complete graphs all satisfy a property known as vertex-transitivity.

2.2.2 Vertex transitive graphs

Definition 2.36 (Vertex transitivity). Let $G(V, E)$ be an undirected graph with vertex set $V = \{v_1, v_2, \dots, v_N\}$. It is vertex transitive if, for every pair of vertices (v_i, v_j) , there exists a function over the vertex set $f : V \rightarrow V$ such that $f(v_i) = v_x$ and $f(v_j) = v_y$.

Note that, under this definition, f is an automorphism of the graph vertex set, simply relabelling them in such a way as to preserve the vertex adjacency of G under the new labelling. In other words, every vertex in a vertex transitive graph has the same local environment; we cannot distinguish vertices by considering their adjacent neighbours. As a consequence, it can be seen that vertex transitive graphs must necessarily be degree-regular.

2.2.3 Strongly connected graphs

The complete graph, as defined earlier, is also an example of a strongly connected graph; that is, every pair of vertices are connected by a walk of arbitrary length.

Definition 2.37 (Strongly connected graph). A graph $G(V, E)$ is said to be strongly connected if, for every pair of vertices v_i and v_j , there exists a walk of length $k \in \mathbb{N}$ in both directions between v_i and v_j .

Recall Eq. 2.5; we can thus reformulate this condition in terms of the adjacency matrix. The adjacency matrix of a strongly connected graph is an *irreducible matrix*.

Definition 2.38 (Irreducible matrix). An adjacency matrix A for a graph $G(V, E)$ is irreducible if any of the following equivalent conditions are met:

- For every $i, j \in [1, N]$ there exists an integer $k \in \mathbb{N}$ such that $(A^k)_{ij} > 0$,
- There exists no permutation matrix P such that PAP^{-1} is block upper triangular matrix,
- Graph G is strongly connected.

Corollary 2.39. For a graph $G(V, E)$ with adjacency matrix A , the following are equivalent:

- Graph G is strongly connected
- A is irreducible

When a graph is strongly connected, it allows us to make an important observation regarding the spectrum of its adjacency matrix, due to the Perron-Frobenius theorem.

Theorem 2.40 (Perron-Frobenius theorem for irreducible matrices). Let $G(V, E)$ be a strongly connected graph $G(V, E)$, with corresponding irreducible adjacency matrix A and eigenvalues $\{\lambda_i\}$. Then,

1. There exists an eigenvalue λ_P of A that is real and positive;
2. $\lambda_P \geq |\lambda_i|$ for all $\lambda_i \neq \lambda_P$ — λ_P is referred to as the principal eigenvalue;
3. λ_i is a simple eigenvalue, i.e. it has a multiplicity of 1;
4. The eigenvector \mathbf{v}_P corresponding to λ_P is real, unique, and strictly positive, i.e. $(\mathbf{v}_P)_j > 0$ for all j .

However, note that point 2. above uses a greater than or equal to sign, implying that cases may exist where *multiple* principal eigenvalues λ_P exist. In order to guarantee that the Perron-Frobenius theorem ensures a unique λ_P , an additional property is required — we require the matrix concerned to be *primitive*.

Definition 2.41 (Primitive matrix). An adjacency matrix A for a graph $G(V, E)$ is primitive if any of the following equivalent conditions are met:

- there exists an integer $k \in \mathbb{N}_0$ such that every element of A^k is positive,
- Every vertex in G is connected to any other by a path of length at least k ,
- the graph G has at least one odd length closed path.

Corollary 2.42. *Every primitive adjacency matrix A is also irreducible.*

Corollary 2.43. *If A is an irreducible matrix, then $A + I$ is necessarily irreducible and primitive.*

Applying the Perron-Frobenius theorem to primitive matrices now ensures a unique principal eigenvalue.

Theorem 2.44 (Perron-Frobenius theorem for primitive matrices). *Let $G(V, E)$ be a strongly connected graph $G(V, E)$, with corresponding primitive adjacency matrix A and eigenvalues $\{\lambda_i\}$. Then,*

1. *There exists a unique principal eigenvalue λ_P of A that is real and positive;*
2. *$\lambda_P > |\lambda_i|$ for all $\lambda_i \neq \lambda_P$;*
3. *λ_i is a simple eigenvalue, i.e. it has a multiplicity of 1;*
4. *The eigenvector \mathbf{v}_P corresponding to λ_P is real, unique, and strictly positive, i.e. $(\mathbf{v}_P)_j > 0$ for all j .*

Henceforth, unless explicitly stated and without loss of generality, it should be assumed that all graphs discussed are strongly connected.

2.2.4 Random graphs

Sometimes, when analysing functions of a graph vertex set $\phi : V \rightarrow \mathbb{R}^N$, we would like to be able to determine the ‘typical’ behaviour of such a function. This can be difficult to do when working with specific network examples — the subconscious choices made when picking or designing specific graph structures may lead to the presentation of results

showing atypical properties of function ϕ . As such, it is critical to analyse the performance of the graph function ϕ over an ensemble of *randomly generated graphs* of vertex size N [93]. Graphs belonging to this random ensemble must conform to $|V| = N$, with the existence of edges between vertices described by a random variable or random process. Note that, as a result of random variable or process chosen to generate the edge set E , the random graphs can be grouped into *classes* which all share particular properties and characteristics; when performing a statistical analysis over random graph ensembles, care must be taken to choose classes of random graphs that accurately model the system under consideration. In this section, we will briefly introduce two classes of random graphs that will be heavily used in this thesis.

The study of random graphs was initiated by Erdős and Rényi [94] in their influential 1959 paper, and as a result, in some fields the term ‘random graph’ remains synonymous with the Erdős-Rényi random graph. A random Erdős-Rényi graph, denoted $G(N, p)$, is comprised of N vertices with edges randomly distributed via a Bernoulli distribution with probability p [95]; subsequently, this class of random graph is sometimes referred to as a *binomial random graph*. For such a network, the expected vertex degree distribution $P(k)$ (the fraction of vertices with degree k) is binomial in form,

$$P(d) \sim \binom{N-1}{d} p^d (1-p)^{N-d-1}, \quad (2.41)$$

resulting in most vertices with degree close to np , the mean number of connections [94, 96]. An example of a randomly generated Erdős-Rényi random graph $G(50, 0.3)$ is shown in Fig. 2.3, alongside a plot of its vertex degree distribution.

In the last two decades, it was noticed that quite a few real-world networks did not exhibit the binomial degree distribution observed in Erdős-Rényi graphs. Instead, they were characterised by a degree distribution with a small number of very highly connected nodes, and a long ‘tail’ indicating a large number of vertices with a limited number of connections. Known as scale-free networks, they exhibit a power law degree distribution of the form $P(d) \sim d^{-\gamma}$, with the few very highly connected vertices known as ‘hubs’. As such, this makes them well suited to modelling a wide array of physical systems and networks with similar characteristics, for example power grids, the World Wide Web, social networks, and biochemical molecules [97, 98].

In order to generate random scale-free networks, one such model commonly used is the Barabási-Albert algorithm [99, 100]. A method of preferential attachment, at each time-step, a vertex with m directed edges is introduced to the system, and preferentially at-

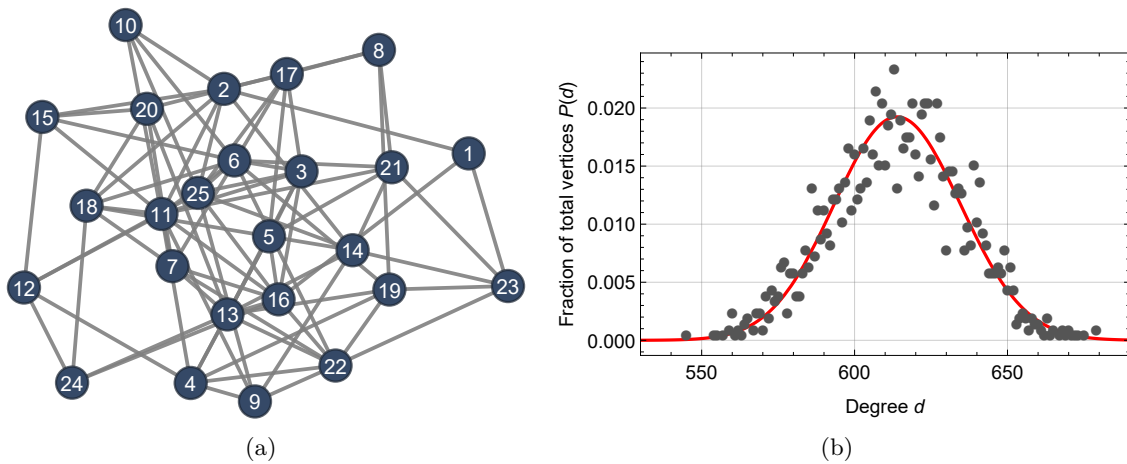


Figure 2.3: **Left:** a randomly generated Erdős-Rényi graph $G(25, 0.3)$. **Right:** The degree distribution, frequency $P(d)$ vs vertex degree d , for a randomly generated Erdős-Rényi graph $G(2048, 0.3)$ (black). The red line represents the expected degree distribution $P(d)$.

tached to existing vertices with higher degrees (with the probability of being connected to vertex v_i given by $p_i = d_i / \sum_j d_j$). This process continues until we have a graph containing the required number of vertices N ; as a result, we may characterize graphs generated from the Barabási-Albert algorithm via the parameters N and m . For the Barabási-Albert algorithm, the expected degree distribution is therefore given by

$$P(d) \sim \frac{2m(m+1)}{d(d+1)(d+2)} = \mathcal{O}(d^{-3}). \quad (2.42)$$

An example of a scale-free graph generated via the Barabási-Albert algorithm is presented in Fig. 2.4, alongside a plot of its power law degree distribution. With the basic terminology and definitions of graph theory and various graph classes established, we can now turn our attention to classical random (and later, quantum!) walks over graph structures.

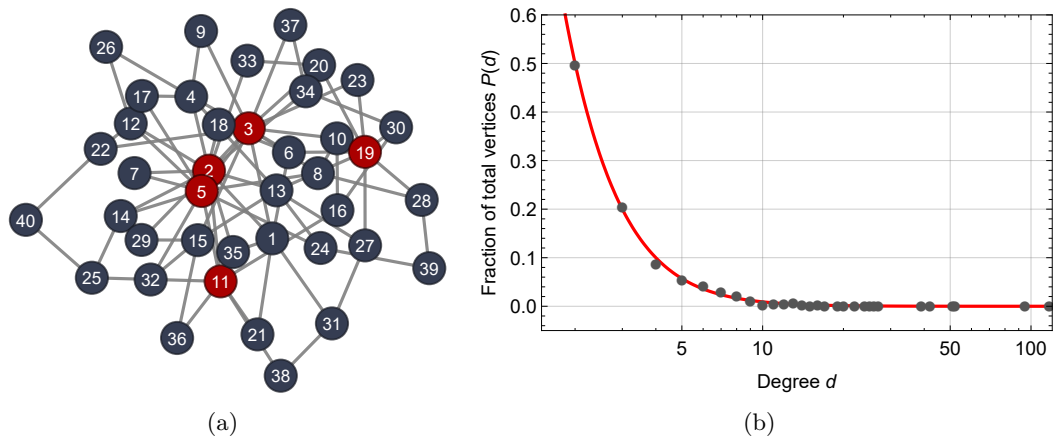


Figure 2.4: **Left:** a random scale-free graph generated via the Barabási-Albert algorithm with parameters $N = 40$, $m = 2$. **Right:** The degree distribution, frequency $P(d)$ vs vertex degree d , for a random scale-free graph generated via the Barabási-Albert algorithm with parameters $N = 1024$, $m = 2$ (black). The red line represents the expected degree distribution $P(d)$.

CHAPTER 3

Classical and quantum walks

In this chapter, we will provide an overview and introduction to the quantum walk framework, and discuss its relationship to the well-established classical random walk.

3.1 Classical random walks

Classical random walks over graphs are examples of Markovian processes — by letting each vertex in the graph represent a state of the system, and the edge representing the rate of transition between states, the graph can be interpreted as a Markov chain. Each step in the random walk is then determined solely by the current location of the walker; i.e. the random walk is *memoryless*. In the study of Markovian processes, random walks have proven invaluable — modelling physical processes such as Brownian motion, forming the basis of computational Monte Carlo techniques, and providing methods of network analysis. In this section, we will introduce both the discrete and continuous-time random walks, and discuss a few of their properties.

3.1.1 Discrete-time random walk

Definition 3.1 (Discrete-time random walk). Consider an strongly connected graph $G(V, E)$, composed of vertices $j \in V$ and edges $(i, j) \in E$ with $|V| = N$. A discrete-time random walk (DTRW) over G is a stochastic Markovian process that evolves as follows,

$$\mathbf{P}^{(n+1)} = T\mathbf{P}^{(n)}, \quad (3.1)$$

where T is the transition matrix, and $\mathbf{P}^{(n)} \in \mathbb{R}^N$ the state vector, constrained such that $\sum_{i=1}^N \mathbf{P}_i^{(n)} = 1$ and $\mathbf{P}_i^{(n)} \geq 0 \forall i$. The i th element of the state vector, $\mathbf{P}_i^{(n)}$, represents

the probability of finding the walker at vertex v_i at time-step n . As per convention, the transition matrix is normally taken to be

$$T = AD^{-1}, \quad (3.2)$$

where $D_{ij} = \delta_{ij} \sum_k A_{ik}$ is the diagonal degree matrix containing the vertex degrees of the graph. This ensures that T is *stochastic* ($\sum_k T_{ik} = 1$), preserving the probability of the walker.

A DTRW with a transition matrix T that is irreducible and primitive ($(T^t)_{ij} > 0$ for some $t \in \mathbb{N}$) always permits a limiting probability distribution of the walker,

$$\lim_{t \rightarrow \infty} T^t \mathbf{P}^{(n)} = \boldsymbol{\pi}, \quad (3.3)$$

which satisfies the steady-state equation $T\boldsymbol{\pi} = \boldsymbol{\pi}$. Thus, the limiting-distribution is simply the eigenvector of T with eigenvalue $\lambda = 1$, and by the Perron-Frobenius theorem must be unique. It is trivial to show that the limiting distribution is therefore proportional to the vertex degree, and given by

$$\boldsymbol{\pi}_j = \frac{D_{jj}}{\text{Tr}(D)} = \frac{\sum_i A_{ij}}{\sum_i \sum_j A_{ij}}. \quad (3.4)$$

Note that in cases where a graph only has even-length closed loops (i.e. T is not primitive), the walker will only occupy sites an even distance from the initial state at even time-steps — causing the walker to never converge to its limiting distribution $\boldsymbol{\pi}$ (the limiting distribution need not be unique, either). In such cases, it is useful to redefine the DTRW so that at every time-step, the walker only has a ϵ probability of moving as per the transition matrix:

$$\begin{aligned} \mathbf{P}^{(n+1)} &= \epsilon T \mathbf{P}^{(n)} + (1 - \epsilon) \mathbf{P}^{(n)} \\ &= \epsilon(T - I) \mathbf{P}^{(n)} + \mathbf{P}^{(n)}. \end{aligned} \quad (3.5)$$

This is known as the *lazy random walk*, and is sufficient to break the periodicity and ensure convergence to the limiting probability distribution $\boldsymbol{\pi}$.

3.1.2 Continuous-time random walk

We can interpret each time-step of the lazy random walk as corresponding to a time of ϵ [101]. Thus, by rearranging this equation and taking the limit $\epsilon \rightarrow 0$, we arrive at the Master equation,

$$\lim_{\epsilon \rightarrow 0} \frac{\mathbf{P}^{(n+\epsilon)} - \mathbf{P}^{(n)}}{\epsilon} = -(I - T)\mathbf{P}^{(n)}, \quad (3.6)$$

a stochastic Markovian process governing the time evolution of the *continuous-time random walk* (CTRW).

Definition 3.2 (Continuous-time random walk). Consider a strongly connected graph $G(V, E)$. A continuous-time random walk over G is a stochastic Markovian process that evolves as per the master equation,

$$\frac{d}{dt}\mathbf{P}(t) = -L\mathbf{P}(t), \quad (3.7)$$

with solution $\mathbf{P}(t) = e^{Tt}\mathbf{P}(0)$, where L is the normalized graph Laplacian

$$L = I - T = (D - A)D^{-1}, \quad (3.8)$$

such that e^{-Lt} is stochastic and the walk is probability conserving.

For the CTRW, the steady-state limiting probability distribution,

$$\boldsymbol{\pi} = \lim_{t \rightarrow \infty} e^{-Lt}\mathbf{P}(0) \quad (3.9)$$

must satisfy the equation $e^{-Lt}\boldsymbol{\pi} = \boldsymbol{\pi}$. After expanding the matrix exponential as a Taylor series, it can be seen that this is equivalent to $L\boldsymbol{\pi} = \mathbf{0}$; i.e. $\boldsymbol{\pi}$ is the nullspace of L . Note that since $L = I - T$, $L\boldsymbol{\pi} = (I - T)\boldsymbol{\pi} = \mathbf{0} \Rightarrow T\boldsymbol{\pi} = \boldsymbol{\pi}$, and thus the CTRW limiting distribution and the DTRW limiting distribution (Eq. 3.4) are identical.

3.2 Quantum walks

When it comes to ‘quantizing’ the classical random walks described previously, there are two well-established formulations — the discrete-time quantum walk (DTQW) and the continuous-time quantum walk. Furthermore, there are two major distinct models for the DTQW, the coined quantum walk (the oldest and most established quantum walk

model) and the Szegedy quantum walk. Here, we will briefly introduce the DTQW, before continuing on to the CTQW, the principal focus of this thesis.

3.2.1 Discrete-time quantum walks

The quantum analogue of the discrete-time random walk, the discrete-time quantum walk (DTQW) on undirected graph $G(V, E)$ differs in that it takes place in the expanded Hilbert space

$$\mathcal{H} = \mathcal{H}_P \otimes \mathcal{H}_C, \quad (3.10)$$

where $\mathcal{H}_P = \{|v_i\rangle : i = 1, 2, \dots, N\}$ is the position Hilbert space comprising of the orthonormal vertex states, and \mathcal{H}_C the coin Hilbert space, comprising of the ‘coin’ states. For each vertex state $|v_i\rangle$, there are d_i associated coin states $\{|c_j\rangle : j = 1, 2, \dots, d_i\}$, with each representing the outgoing edges from vertex v_i . Thus, the orthonormal basis for the coined quantum walk is given by

$$|v_i, c_j\rangle \equiv |v_i\rangle \otimes |c_j\rangle \in \mathcal{H}_P \otimes \mathcal{H}_C. \quad (3.11)$$

At each discrete time-step of the DTQW propagation, we perform two unitary operations on the current state of the walker: a *coin operation* acting on \mathcal{H}_C , and a *shifting operation*, acting on \mathcal{H}_P . The coin operation first acts to ‘mix’ the probability amplitudes located at each vertex v_i , by use of the local unitary coin operator C_i as follows:

$$|v_i\rangle \otimes C_i |c_j\rangle = \sum_{k=1}^{d_i} \alpha_k |v_i, c_k\rangle, \quad \sum_{k=1}^{d_i} |\alpha_k|^2 = 1, \quad (3.12)$$

i.e. C_i can be represented as a $d_i \times d_i$ unitary matrix acting on the set of coin states for vertex v_i . Once the coin operations have been performed on all vertices, we then apply the shifting operator S ,

$$S |v_i, c_j\rangle = |v_j, c_i\rangle, \quad (3.13)$$

which acts to swap two adjacent coin states along an edge. For example, if a graph contains the undirected edge (v_i, v_j) , and $|v_i, c_j\rangle$ represents the coin state at vertex v_i on edge (v_i, v_j) , then the action of the shifting operator results in the coin state $|v_j, c_i\rangle$ — the coin state at vertex v_j , on edge (v_i, v_j) . Based on this definition, we see that the shifting operator is an involutory matrix, $S^2 = I$, with dimension $\sum_i d_i \times \sum_i d_i$. Common local

coin operators on vertices with degree d include the Grover coin,

$$\left(C_G^{(d)}\right)_{ij} = \frac{2}{d} - \delta_{ij}, \quad (3.14)$$

and the DFT (discrete Fourier transform) coin,

$$\left(C_{DFT}^{(d)}\right)_{ij} = \frac{1}{\sqrt{d}} e^{2\pi i j / d}. \quad (3.15)$$

The *global* coin operator is the block diagonal matrix $\mathcal{C} = \text{diag}(C_1, C_2, \dots, C_N)$ with dimension $\sum_i d_i \times \sum_i d_i$, acting on the entire Hilbert space,

$$\mathcal{C} |v_i, c_j\rangle = |v_i\rangle \otimes C_i |c_j\rangle; \quad (3.16)$$

this reduces to $\mathcal{C} = I \otimes C^{(d)}$ if the graph is d -degree regular with the same $d \times d$ local coin operator $C^{(d)}$ acting on every vertex. As such, the unitary time-evolution operator for each discrete time-step in the coined DTQW may be given by

$$U = S \cdot \mathcal{C} \quad (3.17)$$

and thus the time-evolution from initial state $|\psi(0)\rangle$ to time t is given by

$$|\psi(t)\rangle = U^t |\psi(0)\rangle. \quad (3.18)$$

The DTQW time-evolution unitary operator can alternatively be expressed on the space of all directed edges of the graph; i.e. the space formed by all ordered tuples of vertices in the graph. For example, in the case of a d -degree Grover coin applied to a d -degree regular graph, the DTQW unitary time-evolution operator is expressible as follows:

$$U_{ij,kl} = S \cdot \mathcal{C} = \left[S \cdot \left(I \otimes C_G^{(d)} \right) \right]_{ij,kl} = \begin{cases} \frac{2}{d_j} - \delta_{il}, & j = k \\ 0, & j \neq k, \end{cases} \quad (3.19)$$

where $(v_i, v_j), (v_k, v_l) \in E$ and d_j denotes the degree of the j th vertex. This form, which is equivalent to the above definition, has been adopted by Emms et al. [68] and Smith [102].

An alternative framework for the DTQW is the eponymous Szegedy quantum walk, introduced by Szegedy [80]. In this approach, the quantum walk is formulated by directly quantizing classical Markov chains with stochastic transition matrix T (Eq. 3.2), through

the use of unitary reflection and swap operators. We don't describe the Szegedy walk here (Szegedy [80] offers a good introduction to the Szegedy DTQW formalism), but we note that the Szegedy walk can *always* be written in the form of a coined quantum walk on a complete graph with self-loops (i.e. an N^2 Hilbert space).

3.2.2 Continuous-time quantum walks

Consider a graph $G(V, E)$ with vertex set $V = \{v_1, v_2, \dots, v_N\}$ and edge set $E = \{e_1, e_2, \dots, e_M\}$, with corresponding adjacency matrix A and Laplacian L . The continuous-time quantum walk (CTQW) on graph G is the quantum analogue of the CTRW, with the main differentiation being the method of propagation. Unlike the CTRW, which propagates as per the Markovian master equations, the CTQW instead has its time evolution governed by the Schrödinger equation [52].

Definition 3.3 (Continuous-time quantum walk). The continuous-time quantum walk on graph $G(V, E)$ evolves as per the Schrödinger equation,

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H |\psi(t)\rangle, \quad (3.20)$$

where

- H is the system Hamiltonian, encoding the discrete structure of the underlying graph G ;
- $|\psi(t)\rangle = \sum_j \alpha_j(t) |j\rangle$ is the complex-valued state vector; and
- the set $\{|j\rangle : j = 1, 2, \dots, N\}$, $\langle j|j'\rangle = \delta_{jj'}$ represents the orthogonal vertex basis states corresponding to graph vertex set $V = \{v_1, v_2, \dots, v_j\}$.

Note that we sometimes use the notation $|v_j\rangle$ in lieu of the shorthand $|j\rangle$, especially in cases where we must distinguish between multiple walkers inhabiting the same vertex state, or other statespaces such as the momentum space (or coin space for the DTQW).

We use atomic units from hereon, and thus set $\hbar = m = e = 1$ — as a result, the general solution to the system is

$$|\psi(t)\rangle = U(t) |\psi(0)\rangle = e^{-iHt} |\psi(0)\rangle. \quad (3.21)$$

Consistent with standard quantum formalism, $\alpha_j(t) = \langle j|\psi(t)\rangle \in \mathbb{C}$ is the probability amplitude, and $|\alpha_j(t)|^2$ the corresponding probability, of the walker being found at vertex

j after time t . Unlike the classical CTRW, the CTQW gains properties characteristic of quantum systems — including time reversibility (hence, no limiting state) and superposition, allowing propagation through networks quadratically faster than its classical counterpart [8, 52]. However, the CTQW is no longer a stochastic process, but rather deterministic; the probabilistic nature of the walk comes from measuring the quantum state, rather than the walk’s dynamics [103]. Further, note that unlike discrete-time formulations of the quantum walk, in which probability amplitudes can only transition between adjacent (or ‘local’) vertices at each time-step, the CTQW is a *global* process. That is, it is possible for probability amplitude to transition to non-adjacent vertices in the strongly connected graph at each infinitesimal time-step Δt .

It is important to note that there are two competing conventions for the CTQW Hamiltonian that are ubiquitous in the field; the adjacency matrix ($H = \gamma A$) and the (combinatorial) Laplacian ($H = \gamma(D - A)$) [104], where γ represents the hopping rate per edge per unit time. Both provide similar dynamics, with each being preferred for particular applications — the adjacency matrix for simplicity in quantum computation calculations, and the Laplacian for its discrete approximation to the kinetic energy operator of quantum mechanics. There is one exception, however, to this general rule of thumb; if the CTQW is performed on a degree-regular graph with degree k , then the Laplacian becomes $L = D - A = kI - A$, and thus the time evolution operator is

$$U(t) = e^{-iLt} = e^{-i(kI-A)t} = e^{-ikt} e^{iAt}, \quad (3.22)$$

i.e. the dynamics of the CTQW on a degree regular graph with $H = L$ is identical to that of $H = A$, up to a global phase factor and change in direction.

In the papers presented in this thesis, we will explain each subsequent choice of the CTQW Hamiltonian, and the physical reasons that led to that choice. Nevertheless, for undirected graphs, both L and A are Hermitian matrices. It therefore follows that regardless of the choice of the system Hamiltonian, the time evolution operator $U = e^{-iHt}$ is unitary ($UU^\dagger = I$), guaranteeing that the norm of $|\psi(t)\rangle$ is conserved under a continuous-time quantum walk, as required by the Born rule of quantum mechanics [105].

3.2.3 Multiple-particle CTQWs

If we wish to extend the quantum walk to simulate P distinguishable particles on graph G , the Hamiltonian of the system is expanded to act on a N^P Hilbert space $\underbrace{\mathcal{H} \otimes \mathcal{H} \otimes \cdots \otimes \mathcal{H}}_P$,

as follows [5, 6, 42, 102]:

$$H^{(P)} = H_1 \oplus H_2 \oplus \cdots \oplus H_P + \Gamma, \quad (3.23)$$

where H_j is the free-particle Hamiltonian of the j th particle on graph G , Γ represents a potential interaction between the particles acting on the entire Hilbert space, and \oplus is the tensor or Kronecker sum defined via the Kronecker product¹,

$$A_{n \times n} \oplus B_{m \times m} = A_{n \times n} \otimes I_{m \times m} + I_{n \times n} \otimes B_{m \times m}. \quad (3.24)$$

Note that, as each particle is walking over the same graph structure, the free particle Hamiltonians are *identical* ($H_1 = H_2 = \cdots = H_P$), allowing this to be rewritten as

$$H^{(P)} = H^{\oplus P} + \Gamma, \quad (3.25)$$

where we use the shorthand

$$H^{\oplus P} = \underbrace{H \oplus H \oplus \cdots \oplus H}_P. \quad (3.26)$$

The unitary time-evolution operator is then provided by [106]

$$U(t) = e^{-iH^{(P)}t}, \quad (3.27)$$

and acts on the composite state vector

$$|\psi(t)\rangle = \sum_{j_1} \sum_{j_2} \cdots \sum_{j_P} \alpha_{j_1, j_2, \dots, j_P}(t) |j_1\rangle^{(1)} \otimes |j_2\rangle^{(2)} \otimes \cdots \otimes |j_P\rangle^{(P)} \quad (3.28)$$

such that $U(t')|\psi(t)\rangle = |\psi(t+t')\rangle$. Here, superscript (i) indicates a vertex state in the i th subspace, and, like the single-particle case, we require $\sum_{j_1} \sum_{j_2} \cdots \sum_{j_P} |\alpha_{j_1, j_2, \dots, j_P}(t)|^2 = 1$. Note that, due to the unwieldy nature of the above equation, it is common form to see the shorthand

$$|j_1, j_2, \dots, j_P\rangle \equiv |j_1\rangle^{(1)} \otimes |j_2\rangle^{(2)} \otimes \cdots \otimes |j_P\rangle^{(P)}. \quad (3.29)$$

In the case of no interactions, we can drop the Γ term – this allows us to separate the system time evolution operator into tensor products of the individual particle time evolu-

¹Note that the Kronecker sum is distinct from the matrix direct sum, which is also commonly denoted using the symbol \oplus .

tion operators, as the free particle Hamiltonians act separately on each particle's Hilbert space:

$$U(t) = e^{-iH^{\oplus P}t} = e^{-iH_1 \oplus H_2 \oplus \dots \oplus H_P t} = e^{-iH_1 t} \otimes \dots \otimes e^{-iH_P t}. \quad (3.30)$$

Recall that for the CTQW, H is commonly chosen to be either the adjacency matrix or Laplacian matrix of graph G . It is well known that the Kronecker sum of adjacency matrix A with itself, $A \oplus A$ is equivalent to the cartesian graph product of G with itself, $G \square G$ [107]. Furthermore, using the definition of the Laplacian given in Eq. 2.14, it can be easily seen that $L(A \oplus A) = L(A) \oplus L(A)$. Thus, regardless of our choice for the Hamiltonian, the simulation of a multiparticle quantum walk on graph $G(V, E)$ with $|V| = N$ vertices is *equivalent* to simulating a single particle quantum walk on graph

$$G^{\square P} = \underbrace{G \square G \square \dots \square G}_P, \quad (3.31)$$

where \square represents the cartesian product of graphs, and $G^{\square P}$ contains N^P vertices. Accordingly, this provides an avenue for classical numerical simulation of the multi-particle CTQW, an essential process for analysing multi-particle CTQW-based algorithms or systems that cannot be solved analytically.

In Part II, we turn our focus to methods of classical simulation of the CTQW and, in particular, high performance computing techniques and multi-fermion CTQWs.

PART II

SIMULATION

CHAPTER 4

Efficient numerical simulation

*This chapter is based on the paper by J. A. Izaac and J. B. Wang, “pyCTQW: a continuous-time quantum walk simulator on distributed memory computers”, Computer Physics Communications **186**, 81–92 (2015). with minimal modification to match the formatting and notation of this thesis.*

With the growing number of quantum walk applications — especially those in the biochemical sciences (which can involve numerous interacting walkers on increasingly large and complex graphs) — the ability to efficiently simulate and analyse quantum walk behaviour becomes essential. To tackle this problem, we introduce *pyCTQW*, a distributed memory software package with an object-oriented Python interface. By carefully analysing various methods of numerical CTQW simulation, *pyCTQW* allows efficient simulation of large multi-particle CTQW (continuous-time quantum walk)-based systems, whilst taking advantage of the huge potential provided by modern High Performance Computing (HPC) platforms.

Developed using Fortran, the Python module interfaces directly with the Fortran library via F2PY [109]. This allows *pyCTQW* to take advantage of the many scientific and data visualisation Python libraries available in the much more user-friendly Python environment, whilst avoiding the additional overhead that would be caused by implementing the core algorithms in an interpreted language. As such, *pyCTQW* is a valuable tool for the study, data visualisation and analysis of CTQW based systems.

In this chapter, we present an introduction to the Python and Fortran interfaces of *pyCTQW*, discuss various numerical methods of calculating the matrix exponential, and demonstrate the performance behaviour of *pyCTQW* on a distributed memory cluster. In particular, the Chebyshev and Krylov-subspace methods for calculating the quantum walk propagation are discussed, as well as methods for visualisation and data analysis.

4.1 Introduction

In the simulation of continuous-time quantum walk based systems, often the most computationally intensive process is the propagation algorithm itself; that is, the calculation of the evolved state at time t from an initial state $|\psi(0)\rangle$

$$|\psi(t)\rangle = \langle j|e^{-iHt}|\psi(0)\rangle, \quad (4.1)$$

due to the need to efficiently and accurately calculate a matrix exponential-vector dot product. In the case of single walkers or small graphs, fast and efficient matrix exponential calculation is readily available using modern numerical packages and toolkits such as *Mathematica*, *Matlab*, and *SciPy*.

Alternatively, consider a CTQW composed of P simultaneous interacting walkers on an arbitrary graph $G(V, E)$ of $|V| = N$ vertices. As per Eq. 3.23, in order to construct the system Hamiltonian, we take the Kronecker sum of the Hamiltonian over all p single walker subspaces, and add on the interaction term:

$$\begin{aligned} H^{(P)} &= H_1 \oplus H_2 \oplus \cdots \oplus H_P + \Gamma_{int} \\ &= \left(H^{(1)} \otimes I_N^{(2)} \otimes \cdots \otimes I_N^{(P)} \right) + \cdots + \left(I_N^{(1)} \otimes \cdots \otimes I_N^{(P-1)} \otimes H^{(P)} \right) + \Gamma_{int}, \end{aligned} \quad (4.2)$$

where $H^{(i)}$ is the single walker $N \times N$ Hamiltonian of the i th particle on graph G , $I_N^{(i)}$ represents the $N \times N$ identity matrix acting on particle i , and Γ_{int} is the interaction term acting on an arbitrary number of particles. In the development of *pyCTQW*, we were mainly concerned with on-site interactions between all walkers,

$$\Gamma_{int}(\alpha) = \frac{1}{2}\alpha \sum_{\substack{i, i'=1 \\ i \neq i'}}^P \sum_{j=1}^N (|j\rangle \langle j|)^{(i)} \otimes (|j\rangle \langle j|)^{(i')}, \quad (4.3)$$

where $\alpha \in \mathbb{R}$ is the interaction constant, and superscript (i) indicates the operator acts on the i th particles subspace. In the case of no interactions (i.e. $\alpha = 0$), the above definition of the multi-particle Hamiltonian (Eq. 4.2) results in a separable time evolution operator (see Eq. 3.30); however this is not possible when interactions are present. In this work, the Hamiltonian matrix is taken as the same for all particles, and is simply given by the graph Laplacian,

$$H^{(i)} = L(G) \quad \forall i \in \{1, \dots, n\}, \quad (4.4)$$

where the superscript simply defines the *subspace* the Hamiltonian acts on.

The important thing to note from the above multi-particle Hamiltonian is that for P simultaneous walkers on a graph containing N nodes, the system Hamiltonian will have dimensions $N^P \times N^P$ – significantly larger than the Hamiltonian for a single walker. Thus for systems requiring multiple walkers on large graphs, calculating the propagation using the tools mentioned above can potentially take an excessive amount of time, warranting an investigation into alternate algorithms and HPC solutions.

This chapter is structured as follows. We begin by introducing the numerical algorithms commonly used to solve CTQW-based systems in Sec. 4.2. The *pyCTQW* software package is then detailed in Sec. 4.3, with particular attention given to the structure, usage, and installation, followed by specific examples presented in Sec. 4.4, and computational performance discussed in Sec. 4.5. Finally, our conclusions are provided in Sec. 4.6.

4.2 Matrix exponential methods

Although published almost 35 years ago, ‘*Nineteen Dubious Ways to Compute the Exponential of a Matrix*’ by Moler and Van Loan [110] remains a heavily influential review of the various methods of approximating the matrix exponential, even necessitating an updated revision in 2003 [56]. Of the various methods described by Molar and Van Loan, today the two most pervasive include the squaring and scaling method (primarily for dense matrices) and Krylov subspace methods (for large sparse matrices) – both commonly used in conjunction with the Padé approximation, a high order series approximation.

An alternative algorithm that has recently found traction in the field of computational physics and quantum chemistry is the so-called Chebyshev series approximation, which takes its name from the Chebyshev polynomials that occur in the series expansion [59–64]. A huge part of what makes the Chebyshev expansion so attractive is the use of Bessel J zero functions as series coefficients, leading to exceptionally fast convergence without sacrificing a high level of accuracy. In fact, recent research has shown the Chebyshev approximation is an efficient alternative to both the scaling and squaring method [111] and Krylov subspace methods [55], which holds significant promise in parallel computation.

The matrix exponential approximations mentioned here will be expanded on below, with the background theory, advantages and implementations summarised (for a more detailed summary, refer to Molar and Van Loan [56]).

4.2.1 Padé approximation

The Padé approximant uses a rational function of power series in order to approximate the matrix exponential; this is commonly defined as

$$e^A \approx R_{pq}(A) = [D_{pq}(A)]^{-1} N_{pq}(A), \quad (4.5)$$

where

$$N_{pq}(A) = \sum_{j=0}^p \frac{(p+q-j)!p!}{(p+q)!(p-j)!} \left(\frac{A^j}{j!} \right) \quad (4.6)$$

and

$$D_{pq}(A) = \sum_{j=0}^q \frac{(p+q-j)!q!}{(p+q)!(q-j)!} \left(\frac{(-A)^j}{j!} \right). \quad (4.7)$$

Note that for $q = 0$, $p > 0$, this reduces to the Taylor series expansion of e^A ; the Padé approximant simply matches the Taylor series to order $\mathcal{O}(A^{p+q})$. Rather than implementing the above definition directly, however, in practice we typically choose the diagonal Padé approximants $p = q$, as these result in similar accuracy for significantly reduced computational cost. Eq. 4.5 then reduces to

$$e^A \approx R_{pp}(A) = [N_{pp}(-A)]^{-1} N_{pp}(A), \quad (4.8)$$

with error given by

$$e^A - R_{pp}(A) = \frac{(-1)^p (p!)^2}{(2p)!(2p+1)!} A^{2p+1} + \mathcal{O}(A^{2p+2}). \quad (4.9)$$

The main advantage of the Padé approximation over other series expansions is that for matrices where $\|A\|_2 \lesssim 1$, matrix exponentials can be computed to the same level of accuracy with up to half the computation. However, there are important downsides – firstly, a reasonable level of accuracy can only be achieved for $\|A\|_2 \lesssim 1$; issues also arise for large p , increasing the risk that the denominator is ill-conditioned for inversion (although this is highly dependent on the eigenvalue spread of A). Furthermore, the evaluation of the power series $N_{pp}(A)$ requires dense matrix arithmetic, foregoing any speed-ups/memory reductions that may have arisen by manipulating sparse matrices.

4.2.2 Scaling and squaring method

In order to apply the Padé approximation to matrices of all norms, an iterative process known as the scaling and squaring method is used, which exploits the property $e^A = (e^{A/s})^s$ to reduce the matrix norm of A before finding the Padé approximant. Thus, the adjusted Padé approximant can be written as

$$e^A \approx e^{A+E} = R_{pp}(A/2^s)^{2^s}, \quad (4.10)$$

where $s \in \mathbb{N}$, and E is a matrix signifying the deviation from the exact value when calculating the matrix exponential of e^A using the Padé approximation. The following inequality is satisfied:

$$\frac{\|E\|}{\|A\|} \leq \frac{2^{3-2p}(p!)^2}{(2p)!(2p+1)!}, \quad (4.11)$$

where the parameters p and s are chosen such that $\|E\|/\|A\| \leq \epsilon$ (here ϵ is a specified parameter determining the error tolerance), and $p + s$ is *minimised* (as the floating point operations per second for an $n \times n$ matrix is of the order $(p + s + 1/3)n^3$). Very generally, $p = 6$ or 7 is optimal for single precision, whereas $p = 17$ is optimal for double precision. For example, *MATLAB*'s `expm` function uses $p = 6$ and $\max\{s : \|A/2^s\|_\infty \leq 0.5\}$.

This method remains one of the most effective methods of numerically calculating the matrix exponential of dense matrices [57, 58], and as such is widely used in modern numerical packages (such as *MATLAB*'s `expm`, *Mathematica*'s `MatrixExp`, *SciPy*'s `linalg.expm`, and *Expokit*'s Fortran and Matlab libraries).

4.2.3 Krylov subspace methods

Over the last 35 years, an alternative method has arisen which enables us to harness the computational advantages afforded by large sparse matrices. This is the eponymous Krylov subspace method, which provides an effective scheme for efficiently calculating the matrix-vector product $e^{At}\mathbf{v}$ – avoiding the need to calculate and store a large intermediate matrix. Here A is a $n \times n$ matrix, t is a scaling parameter, and $\mathbf{v} \in \mathbb{R}^n$ is an n -element vector. Whilst not providing the matrix exponential directly, this matrix-vector product arises often in the study of quantum systems, and as such is frequently the desired computational end-product – this is generally the case in CTQWs, where transition amplitudes $\langle j | \hat{U}(t) | \psi(0) \rangle = \langle j | e^{-iHt} | \psi(0) \rangle$ are desired.

This scheme takes its name from the process of approximating an n -dimensional problem (in this case $e^{At}\mathbf{v}$) onto a smaller m -dimensional *Krylov subspace*,

$$\mathcal{K}_m(At, \mathbf{v}) = \text{span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \dots, A^m\mathbf{v}\}, \quad \forall t \in \mathbb{C}, \quad (4.12)$$

whereby dense matrix methods become adequately efficient to solve the reduced problem. For example, in the case of $e^{At}\mathbf{v}$, we first use the Arnoldi or Lanczos methods (successful Krylov-based iterative methods in their own right, used for calculating the eigenvalues of large sparse general/symmetric matrices respectively) to calculate the orthonormal basis set $V_m = \{\mathbf{v}_1, \dots, \mathbf{v}_m\} \in \mathbb{R}^{n \times m}$ from $\mathcal{K}_m(A, \mathbf{v})$. A byproduct of this calculation is the upper Hessenberg matrix $H_m \in \mathbb{R}^{m \times m}$, which is the projection of $A_n \in \mathbb{R}^{n \times n}$ onto V_m (i.e. $V_m^T A_n V_m = H_m$). Note that H_m approximates the original sparse matrix A_n ,

$$AV_m = V_m H_m + h_{m+1,m} \mathbf{v}_{m+1} \hat{\mathbf{e}}_m^T \approx V_m H_m, \quad (4.13)$$

allowing the large sparse matrix exponentiation-vector product problem to be approximated via

$$e^{At}\mathbf{v} \approx \beta V_m e^{tH_m} \hat{\mathbf{e}}_1, \quad (4.14)$$

since $\mathbf{v} = \beta \mathbf{v}_1$ for some constant β . This can then be tackled via dense matrix methods.

Krylov subspace methods are often combined with the squaring and scaling method when calculating the matrix exponential of a large sparse matrix, due to the latter's efficiency in computing the resulting dense matrix exponential. Moreover, the ability to achieve a high level of accuracy with a relatively small value of m , typically around 40, combined with the fact that successive time-steps can be computed by utilising previously constructed Krylov subspace with little-to-no extra cost, have resulted in the Krylov method becoming a leading approach for large sparse matrices. Today, Krylov matrix exponential methods are available as part of *Mathematica* (using the `MatrixExp[A, v]` function), *Expokit* [112] (a collection of Fortran and Matlab libraries designed for working with Markovian processes), and SLEPc/slepc4py (a HPC implementation with C, Fortran and Python bindings), amongst others.

4.2.4 Chebyshev approximation

Not as common as the Padé and Krylov methods, but rapidly gaining traction in fields such as high performance computing and quantum mechanics, is the Chebyshev series expansion to the matrix exponential;

$$e^{tA} = e^{(\lambda_{max} + \lambda_{min})t/2} \left[J_0(\alpha)\phi_0(\tilde{A}) + 2 \sum_{n=1}^{\infty} i^n J_n(\alpha)\phi_n(\tilde{A}) \right], \quad (4.15)$$

where $\lambda_{max}, \lambda_{min} \in \mathbb{C}$ are the eigenvalues of an $n \times n$ matrix A with largest and smallest real parts respectively, $\alpha = i(\lambda_{min} - \lambda_{max})t/2$, and $\phi_n(\tilde{A})$ are the Chebyshev polynomials, satisfying the recurrence relations

$$\phi_0(\tilde{A}) = I, \quad (4.16a)$$

$$\phi_1(\tilde{A}) = \tilde{A}, \quad (4.16b)$$

$$\phi_n(\tilde{A}) = 2\tilde{A}\phi_{n-1}(\tilde{A}) - \phi_{n-2}(\tilde{A}). \quad (4.16c)$$

Note that for maximal convergence, we require normalisation such that $\lambda \in [-1, 1]$; thus

$$\tilde{A} = \frac{2A - (\lambda_{max} + \lambda_{min})I}{\lambda_{max} - \lambda_{min}}. \quad (4.17)$$

As with the Krylov method, we can calculate the matrix-vector product $e^{At}v$ directly using the Chebyshev approximation, by post-multiplying the Chebyshev series expansion with v . This allows us to take full advantage of sparse matrix libraries when calculating the *sparse matrix* \times *dense vector* terms in the summation, whilst avoiding the extra memory usage required to store intermediate matrix terms. Another attractive property of the Chebyshev expansion is the use of Bessel function zeros as series coefficients, as $J_n(\alpha) \approx 0$ for $n > |\alpha|$, allowing for fast convergence and significantly high accuracy after only $|\alpha| \propto t$ terms (see Fig. 4.1). Whilst various definitions exist for the relative error of the Chebyshev expansion [55], for simplicity it is sufficient to instruct the series to be truncated when the condition

$$|2J_n(\alpha)| \leq \epsilon, \quad (4.18)$$

is satisfied for a specified tolerance $\epsilon > 0$, as subsequent terms can be considered negligible. In *pyCTQW*, we nominally choose $\epsilon = 10^{-18}$.

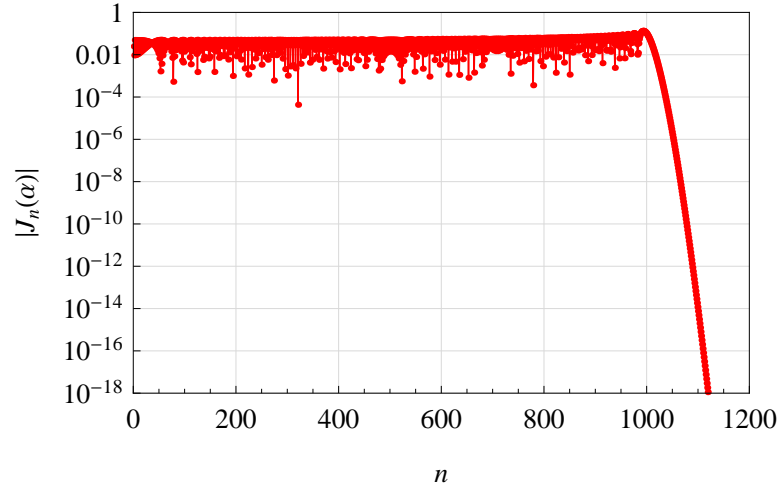


Figure 4.1: Absolute value of the Chebyshev coefficients $|J_n(\alpha)|$ for $|\alpha| = 1000$, when calculating e^{At} for a 3×3 matrix A . Note that $|\alpha|$ is proportional to the numerical range of matrix A and time-step t .

As with the Padé approximation, the Chebyshev approximation is only convergent for matrices where $\|A\| < 1$, thus necessitating the scaling procedure $A \rightarrow \tilde{A}$ detailed above. In practice only λ_{max} need be calculated, since the Laplacian for a undirected, defect-free graph is positive-semidefinite, ensuring $\lambda_{min} = 0$, whilst the calculation of λ_{max} is often computationally much less demanding.

However, when working with large finite graphs in the presence of disorder, or infinite graphs, the density of the eigenvalue spectrum around $\lambda = 0$ can lead to time-consuming eigenvalue calculations, resulting in a significant bottleneck when simulating the quantum walk propagation. This issue can be avoided in principle by implementing the scaling and squaring method, as described in Sec. 4.2.2, as it does not require the calculation of eigenvalues:

$$e^{tA} = \left[J_0(\alpha)\phi_0\left(\frac{tA}{2^s}\right) + 2 \sum_{n=1}^{\infty} i^n J_n(\alpha)\phi_n\left(\frac{tA}{2^s}\right) \right]^s. \quad (4.19)$$

Unfortunately, this approach requires the use of dense matrix operators and functions, and also loses the ability to compute the matrix-vector product $e^{tA}\mathbf{v}$ directly from a matrix-vector product series, removing many of the properties that make the Chebyshev series expansion so appealing.

As such, we have elected to implement the eigenvalue scaling method in *pyCTQW* as one of the two possible propagator methods (alongside SLEPC's built-in Krylov methods). In order to alleviate the computational expense of calculating λ_{min} or λ_{max} for infinite

graphs or graphs containing defects, the user may choose to enter a value for $\hat{\lambda}_{min}$ or $\hat{\lambda}_{max}$ directly – bypassing the eigenvalue calculation – or fallback to the Krylov method. If entering eigenvalue estimates, these should be such that $0 \leq \hat{\lambda}_{min} \leq \lambda_{min} < \lambda_{max} \leq \hat{\lambda}_{max}$, ensuring that the numerical range of \tilde{A} remains within $[-1, 1]$. Note that estimation of the extreme eigenvalues outside the actual numerical range produces the same level of accuracy at the expense of a larger value of $|\alpha|$, which would then require some extra matrix-vector terms in the series expansion to ensure accuracy. Nonetheless, the Chebyshev expansion has been shown to exhibit very low sensitivity to eigenvalue estimates [55], allowing this ‘semi-empirical’ implementation to continue to be an effective alternate to the Krylov method in specific circumstances.

Although the Chebyshev matrix exponentiation method is starting to show great promise, especially in fields of quantum chemistry and physics, the more established Krylov techniques continue to dominate in computational applications – today, *Expokit*[112] is the only well known package providing the Chebyshev methods (albeit, only for dense matrices). By implementing distributed memory Chebyshev methods *in addition* to Krylov methods, *pyCTQW* will be able to provide fast and efficient CTQW propagation methods to work with multiple quantum walkers.

4.3 Software package

The *pyCTQW* package provides both a Python-based object-orientated framework (`pyCTQW.MPI`) as well as a library (`libctqwMPI`) containing various Fortran subroutines, which enable the user to simulate and explore various user-defined CTQW systems in a HPC environment. In the latter case, this is performed via the manipulation of graph-constrained quantum walk objects, whereas in the former case a more sequential paradigm is required where the user calls various subroutines on matrix or vector objects.

In this section, installation and dependencies will be addressed briefly, after which usage of both the Python and Fortran components of *pyCTQW* will be discussed in detail.

4.3.1 Dependencies and installation

As the ability to readily take advantage of modern HPC systems is one of the major design goals of this package, the decision was made to utilize the MPI-based PETSc [113–115] and SLEPc [116, 117] scalable linear algebra libraries. This allowed rapid parallelisation whilst

also taking advantage of the optimised data structures and matrix operations provided by PETSc and SLEPc.

Furthermore, in addition to requiring PETSc and SLEPc for parallel computation, the *pyCTQW* Python module also makes use of a wide variety of scientific and data visualisation libraries, including NumPy, SciPy, Matplotlib and NetworkX. This provides a rich user environment, and builds on the maturity and efficiency of the aforementioned libraries. It should be noted that whilst the Python module requires the *petsc4py* Python extension in order for the user to directly manipulate PETSc data structures and access MPI attributes (such as `rank`), the quantum walking calculations are called directly from the Fortran library, allowing the module to exploit the speed gain afforded by Fortran.

Once these dependencies are met, the Python module can be installed from the source code by running the following terminal command:

```
python setup.py install
```

whilst the Fortran library can be compiled via the included makefile:

```
make fortran
```

For additional installation help and compilations options, for example building *pyCTQW* as a shared library, the user should refer to the online *pyCTQW* documentation [7].

4.3.2 Fortran library `libctqwMPI`

Of the two interfaces available as part of the software package, the Python module is designed for an easier end-user experience, and thus is much more full featured. However, the underlying Fortran library may still be used, and is described briefly below.

To call functions and subroutines from the included `libctqwMPI` Fortran library, the main Fortran program should have the following basic structure:

```
program main
  ! load libctqwMPI module
  use ctqwMPI

  ! PETSc headers
  #include <finclude/petsc.h>

  PetscErrorCode :: ierr
  PetscMPIInt    :: rank
```

```

! initialize SLEPc and PETSc
call PetscInitialize(PETSC_NULL_CHARACTER,ierr)
call MPI_Comm_rank(PETSC_COMM_WORLD,rank,ierr)

! program code here

! finalise PETSc
call PetscFinalize(ierr)
end program main

```

Note that this structure is influenced by typical PETSc and SLEPc Fortran programs, due to the dependence on these aforementioned libraries. As such, we can take advantage of PETSc system routines, e.g. allowing command line arguments to be easily added, as well as code profiling capabilities (i.e. using PETSc `LogStages` to create a computational summary of parallel code execution). For more details, refer to the PETSc and SLEPc documentations [114, 118].

Once the `libctqwMPI` module has been included and PETSc and SLEPc have been properly initialised, CTQW subroutines can then be called directly. For example, to calculate the propagation of Hamiltonian H from state `psi0` to `psi` after time `t`,

```
call qw_cheby(psi0,psi,t,H,Emin,Emax,rank,n)
```

where `psi0`, `psi` and `H` are properly declared/allocated PETSc vectors/matrices, `t`, `Emin` and `Emax` are PETSc reals, and `rank`, `n` are PETSc integers. For a summary of available parallel CTQW subroutines, see Tab. 4.1.

Once the Fortran program is complete, it can be compiled and linked against `libctqwMPI` using the supplied makefile template, and then executed by running

```
mpirun -np X <program> [options]
```

where `X` is the number of MPI nodes to use, and `options` refer to either user defined or inbuilt PETSc/SLEPc command line arguments used to modify program function. For further information and additional details regarding the Fortran interface, the user is advised to refer to the online documentation [7].

4.3.3 Python module `pyCTQW.MPI`

Once `pyCTQW.MPI` is installed, calculations involving CTQWs can be called in several ways; either via an executable Python script, or as an interactive session, for example, using the

Table 4.1: Summary of parallel CTQW subroutines available in the Fortran `libctqwMPI` library. For more details and required arguments, refer to the online documentation.

<i>Hamiltonian subroutines</i>	
<code>importadjtoh</code>	Import an adjacency matrix from a file, and create a PETSc Hamiltonian matrix
<code>adjtoh</code>	Convert an adjacency array to a PETSc Hamiltonian matrix
<code>hamiltonian_p<N>_line</code>	Create the Hamiltonian matrix representing $N = 1, 2$ or 3 walkers on an infinite line
<i>Statespace subroutines</i>	
<code>p<N>_init</code>	Initialise the statespace of an $N = 1, 2$ or 3 particle CTQW on specified graph nodes
<code>marginal<N></code>	Calculate the marginal probability of particle number $p \in N$ for an $N = \{1, 2, 3\}$ particle CTQW
<i>Quantum walk propagation</i>	
<code>min_max_eigs</code>	Calculate the minimum and maximum eigenvalues of a PETSc matrix using SLEPc
<code>qw_krylov</code>	Propagate the quantum walk for time t using Krylov subspace methods
<code>qw_cheby</code>	Propagate the quantum walk for time t using the Chebyshev series expansion
<code>entanglement</code>	Calculates the 2 particle Von Neumann entropy $S = -\sum_i \lambda_i \log_2 \lambda_i$, where λ_i are the eigenvalues of the reduced density matrix $\rho_2 = \text{Tr}_1(\psi(t)\rangle \langle \psi(t))$

iPython or iPython-Notebook environments. The syntax in both cases is identical, with slight differences in execution. Here, we will be concerned mainly with executable Python scripts, required by most HPC job schedulers.

Initialisation

In order to set-up the Python environment correctly, we must first initialize the PETSc environment, and import the `pyCTQW.MPI` module:

```
#!/usr/bin/env python2.7
import sys, petsc4py
petsc4py.init(sys.argv)
from petsc4py import PETSc
import pyCTQW.MPI
```

User defined options

PETSc can be used to create command line arguments for the script; this is useful in HPC environments where multiple jobs are to be submitted for a parameter exploration. For

example, the following code creates two command line options, `-t` and `-N`, with default values of 100 and 20 respectively:

```
OptDB = PETSc.Options()
N = OptDB.getInt(`N`, 100)
t = OptDB.getReal(`t`, 20)
```

When running the complete parallelised simulation, the options created are invoked as follows,

```
mpirun -np X <script>.py [options]
```

where `X` is the number of MPI compute nodes/processes. In the case of the latter example,

```
mpirun -np X <script>.py -N 5 -t=`0.3`
```

would alter the two user defined variables as indicated. Furthermore, most PETSc and SLEPc subroutines accept command line options which modify their settings; for instance, when using the SLEPc EPS eigensolver, the eigensolver type to be used can be changed dynamically by passing an `eps_type` option:

```
mpirun -np 2 <program> -eps_type=`lapack`
```

For more details on built-in PETSc/SLEPc command line options, refer to the PETSc and SLEPc documentation [114, 118].

Rank and local operations

When running on multiple nodes, sometimes only specific nodes are required to perform a specific calculation or operation, for instance, the I/O operations where all nodes already have the same information. Using PETSc, the rank (i.e. the MPI process number) can be determined for each process, and conditional statements are used to control which node performs the I/O operation:

```
rank = PETSc.Comm.Get_rank(PETSc.COMM_WORLD)
if rank == 0:
    print `1P Line\n`
```

Caution is advised, however – the above should only be utilised for user defined functions and processes, as all of the methods and functions available in `pyCTQW.MPI` are designed to work globally on all processes, and should not be created or called on a subset of all available nodes; doing so may result in unresponsive simulations. Moreover, most objects in `pyCTQW.MPI` contain I/O methods (e.g. `pyCTQW.MPI.Graph.exportState()`),

alternatively PETSc I/O methods are also available (e.g. `PETSc.Vec.view()`); these are global over all nodes as mentioned above and should be used over custom I/O methods when possible.

Code profiling

PETSc also allows for easy code profiling by supplying the command line option `-log_summary` when executing your script. This functionality is built-in to `pyCTQW.MPI` with log stages created automatically for available methods, for example, when creating the Hamiltonian or initial statespace, finding the eigenvalues, or CTQW propagation, etc. This provides a wealth of important computational information, such as memory usage, communication, and timing data, allowing the user to fine-tune simulation parameters and solvers for ultimate efficiency.

Custom log stages may also be included using the following template:

```
stage1 = PETSc.Log.Stage('First Stage Title')
stage1.push()
# place stage 1 functions/operations here
stage1.pop()
```

4.3.4 pyCTQW framework

The `pyCTQW.MPI` Python module provides an object-oriented framework for simulating CTQW systems, which involves the creation of CTQW objects storing various data structures (such as Hamiltonians), which can be manipulated via built-in methods and functions (see Tab. 4.2).

The number of available methods and attributes is too large to detail in full here. Instead, a two particle CTQW on a Cayley tree will be discussed in the next section to provide a feel of the framework and its structure. For further details, refer to the *pyCTQW* online documentation [7].

4.4 Worked example

In this example, a two particle continuous-time quantum walk will be performed on a 3-Cayley tree using `pyCTQW.MPI`. After initialising PETSc and `petsc4py`, the *pyCTQW*

Table 4.2: Summary of parallel CTQW classes available in `pyCTQW.MPI`. For more details and required arguments, please refer to the online documentation.

<i>CTQW submodule</i>	
<code>ctqw.Hamiltonian</code>	Contains methods for initializing, creating and manipulating Hamiltonian matrices
<code>ctqw.EigSolver</code>	Contains methods for setting up and solving the for the boundary eigenvalues of a distributed PETSc matrix
<i>Quantum walks on arbitrary graphs</i>	
<code>Graph</code>	1 particle CTQW on a user-defined graph
<code>Graph2P</code>	2 particle CTQW on a user-defined graph
<code>Graph3P</code>	3 particle CTQW on a user-defined graph
<i>Quantum walks on infinite lines</i>	
<code>Line</code>	1 particle CTQW on an infinite line
<code>Line2P</code>	2 particle CTQW on an infinite line
<code>Line3P</code>	3 particle CTQW on an infinite line
<i>Graph isomorphism</i>	
<code>GraphISO</code>	A graph isomorphism solver, containing functions for creating graph certificates and checking isomorphism of adjacency matrices (see Sec. 6.5)

module is imported and a `Graph2P` object created, indicating that we will be working on a graph containing $n = 10$ nodes:

```
walk = pyCTQW.MPI.Graph2P(10)
```

Next, we import a 10×10 adjacency matrix of the 3-Cayley tree from a text file¹ with elements delimited by either tabs or spaces²:

```
d = [3,4]
amp = [2.0,1.5]
walk.createH(`graphs/cayley/3-cayley.txt`, `txt`,
            d=d, amp=amp, layout=`spring`, interaction=0.5)
```

In this example, defects are introduced at vertices 3 and 4 of the graph, with amplitudes 2 and 1.5 respectively, and the particles interact with strength 0.5 when co-located at the same vertex; this results in a system Hamiltonian of the form

$$H = H^{(1)} \oplus H^{(1)} + \Gamma_{int}(0.5), \quad (4.20)$$

where

$$H^{(1)} = L_{3C} + 2|3\rangle\langle 3| + 1.5|4\rangle\langle 4|$$

¹Importing the adjacency matrix from a 2D PETSc binary array, specified by filetype `'bin'`, is also supported.

²Alternative delimiters can also be passed using the `delimiter` keyword.

$$= \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 3 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 5 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 2.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

is the 1-particle Hamiltonian and L_{3C} is the Laplacian matrix of the 3-Cayley tree. Note that the graph layout is being stored in ‘spring’ form, which is the default if not specified with alternative layout options including circle, spectral and random. After executing the above code, a Hamiltonian object `walk.H` is produced, allowing the Hamiltonian matrix and its eigenvalues to be accessed.

Now that the system Hamiltonian has been calculated, the initial statespace can be populated. For example, the initial state is chosen to be $|\psi(0)\rangle = \frac{1}{\sqrt{2}}(|0,1\rangle + i|1,1\rangle)$, which is implemented as follows:

```
init_state = [[0,1,1.0/numpy.sqrt(2.0)],
              [1,1,1.0j/numpy.sqrt(2.0)]]
walk.createInitState(init_state)
```

Note that the initial state is passed to the CTQW object in the form of an $n \times 3$ array with format `[[x1,y1,amp1],[x2,y2,amp2],...]`, which generalizes an $n \times p$ array for a p -particle quantum walk. In more complex cases, an initial statespace can be imported from an $n \times n$ 2D array in text format, or an n^2 element PETSc binary vector, via `PETSc.Graph2P.importInitState()`.

The next process involves setting the eigensolver properties, which are then passed to the `EigSolver` object contained in our `walk` object:

```
walk.EigSolver.setEigSolver(tol=1.e-2,
                             verbose=False,emin_estimate=0.)
```

It is important to note that, since the minimum eigenvalue of a finite graph is necessarily 0, calculation of the minimum eigenvalue is avoided automatically when *pyCTQW* detects a finite graph. To reduce computational time further, `emax_estimate`

may also be provided, however this *must* be larger than the *actual* maximum eigenvalue for Chebyshev convergence to be ensured (with convergence time then increasing as $|\text{emax_estimate} - \text{emin_estimate}|$ increases). For additional keywords that can be passed to the eigensolver, see Tab. 4.3.

Table 4.3: pyCTQW.MPI eigensolver keyword arguments

<code>esolver</code> (<i>str</i>)	The default eigensolver algorithm to use. <ul style="list-style-type: none"> • <code>`krylovschur'</code> (<i>default</i>) – Krylov-Schur • <code>`arnoldi'</code> – Arnoldi Method • <code>`lanczos'</code> – Lanczos Method • <code>`power'</code> – Power/Rayleigh Quotient Iteration • <code>`gd'</code> – Generalized Davidson • <code>`jd'</code> – Jacobi-Davidson, • <code>`lapack'</code> – LAPACK eigensolver subroutines • <code>`arpack'</code> – ARPACK subroutines
<code>workType</code> (<i>str</i>)	The eigensolver worktype (either <code>`ncv'</code> or <code>`mpd'</code>). The default is to let SLEPc decide.
<code>workSize</code> (<i>int</i>)	Sets the work size if <code>workType</code> is set
<code>tol</code> (<i>float</i>)	Tolerance of the eigensolver (default is to let SLEPc decide)
<code>maxIt</code> (<i>int</i>)	maximum number of iterations of the eigensolver (default is to let SLEPc decide)
<code>verbose</code> (<i>bool</i>)	If <code>True</code> , writes eigensolver information to the console
<code>emax_estimate</code> (<i>float</i>)	Override the calculation of the graphs maximum eigenvalue (must be larger than or equal to the <i>actual</i> maximum eigenvalue for Chebyshev convergence)
<code>emin_estimate</code> (<i>float</i>)	Override the calculation of the graphs minimum eigenvalue (must be smaller than or equal to the <i>actual</i> minimum eigenvalue for Chebyshev convergence)

Everything is now set up, allowing propagation to begin. The following commands will allow a record of the calculated probability at specified nodes and/or the multipartite entanglement^{3,4} to be stored over time:

```
# create a probability handle for nodes 0-4,9:
walk.watch([0,1,2,3,4,9])
# create a handle to watch the entanglement
walk.watch(None, watchtype='entanglement',
            verbose=False, esolver='lapack')
```

Finally, we allow the CTQW to propagate over the 3-Cayley tree for $t = 5$, using time-steps of $\Delta t = 0.01$:

³as entanglement is a global measurement, there is a large amount of node communication which may increase overall computation time.

⁴the entanglement watchtype uses an eigenvalue solver, and so accepts `EigSolver` keywords detailed in Tab. 4.3.


```
for dt in np.arange(0.01,t+0.01,0.01):
    walk.propagate(dt,method='chebyshev')
```

In this example, we use the Chebyshev propagator, but the Krylov propagator (which is slightly slower) may be used instead. It should also be noted that, as the watch handles are updated every time propagation occurs, the small time-step allows transient data to be collected at a reasonable resolution. However, if the end result of the CTQW propagation is all that is required, it is equally valid to remove the loop and use a time-step $\Delta t = t$, without any loss of accuracy and with some saving in computation time.

Now that the simulation is complete, raw data can be exported and visualisations can be produced, as shown by the following example:

```
# p1 and p2 marginal probabilities at t=5
walk.plot('plot.png')

# 3D graph showing of marginal probability at t=5
walk.plotGraph(output='plot_3D.png')

# p1 and p2 probability over time for node 1
walk.plotNode('plot_node1.png',1)
# p2 probability over all watched nodes
walk.plotNodes('plot_nodes_particle2.png',p=2)

# plot the entanglement vs. time
walk.plotEntanglement('entanglement.png')

# export the partial trace as a text file
walk.exportPartialTrace('rhoX.txt','txt',p=1)
# export the final state to a PETSc binary file
walk.exportState('output_state.bin','bin')
```

A selection of these plots is reproduced in Fig. 4.2, Fig. 4.3 and Fig. 4.4.

Finally, once the CTQW object is no longer needed, memory can be freed by destroying it together with all associated matrices and vectors:

```
walk.destroy()
```

Now that the simulation script is complete, it can be made executable and run on distributed memory systems using `mpirun`.

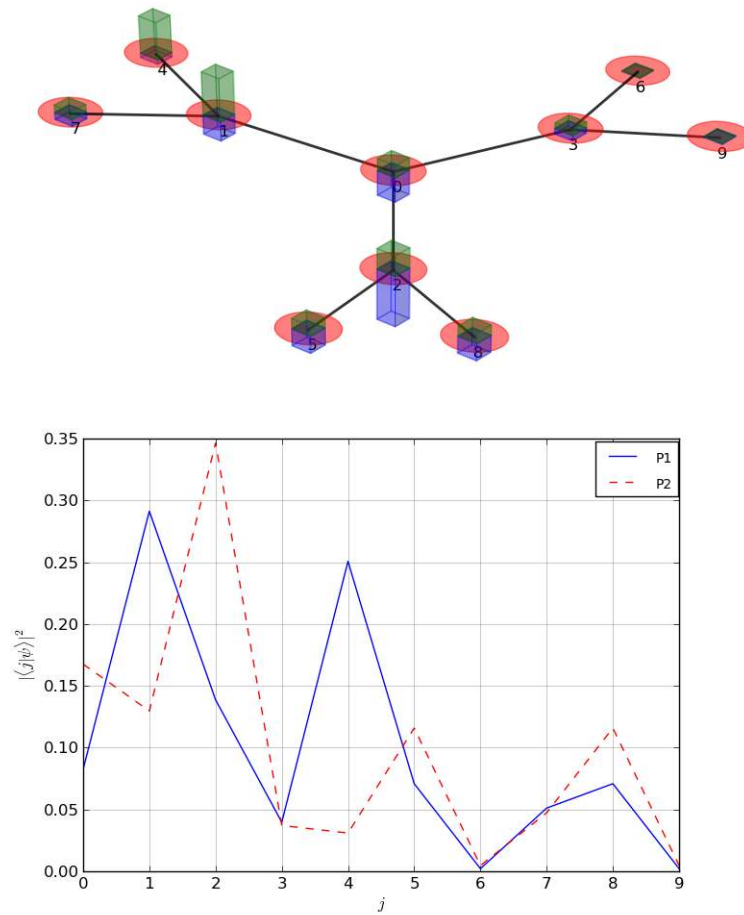


Figure 4.2: Marginal probability of particles 1 and 2 from the worked example after CTQW propagation over a 3-Cayley tree, as plotted over a 3D representation of the graph (top) and vs. node number (bottom).

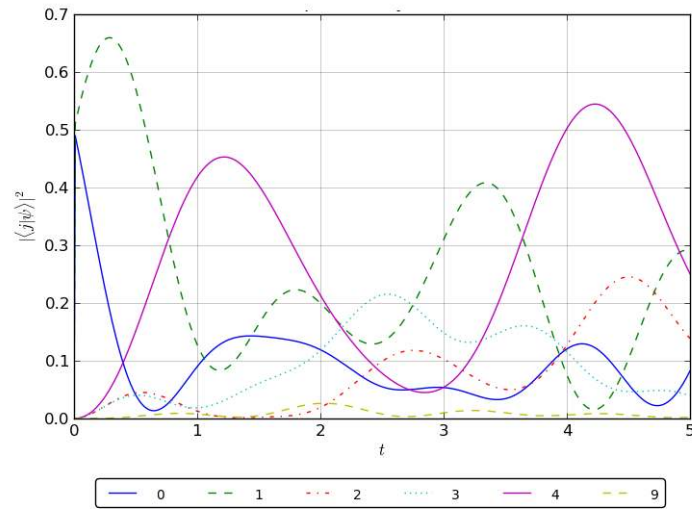


Figure 4.3: Probability distribution of particle 1 vs time, plotted for various graph nodes.

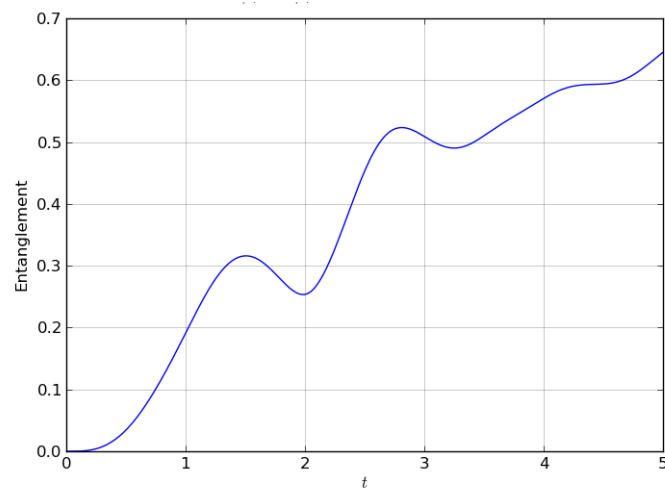


Figure 4.4: Entanglement (represented by the Von Neumann entropy) of the two particle CTQW propagation over a 3-Cayley tree vs. time

4.5 Scaling and performance

In order to effectively determine the performance of the Chebyshev and Krylov exponential methods used in *pyCTQW*, the following study was undertaken comparing the aforementioned algorithms across various implementations. Furthermore, as one of the main design points of *pyCTQW* is to provide a framework for under-the-hood parallelization of CTQW-based simulations, the parallel scaling behaviour of *pyCTQW* was also explored for a variety of input problem sets.

Note that in the following performance study, all computations were completed on a Xeon X5650 based SGI cluster with an InfiniBand low-latency interconnect.

4.5.1 Library comparison

For this comparison, the problem set chosen is that of a two-particle non-interacting CTQW along a finite line of length N , with vertices labelled $| -N/2 + 1 \rangle, \dots, |0\rangle, \dots, |N/2\rangle$. This system can be made sufficiently large and complex, and involves the calculation of an $N \times N$ matrix exponential, allowing performance differences to be clearly visible as N scales whilst reducing Hamiltonian construction overheads. To add complexity into the system, defects are placed at vertices $|3\rangle$ and $|4\rangle$ of amplitudes 2 and 1.5, respectively. Propagation is calculated for $t = 10$ from initial state

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}} (|0\rangle \otimes |0\rangle + i|1\rangle \otimes |0\rangle).$$

Fig. 4.5 shows how the resulting wall time of the system scales as N increases, with the propagation simulated using the Chebyshev algorithm implemented via Intel Fortran/LAPACK, Python/SciPy, and *pyCTQW* (using both a single process and 240 MPI processes on 40 6-core Intel Xeon X5650s). Also plotted for comparison is the same propagation calculated using Mathematica's in-built `MatrixExp[]` function, using the Krylov method and sparse matrices.

It can immediately be seen that *pyCTQW* outperforms all other methods, with the exception of Intel Fortran/LAPACK for $N \lesssim 100$. However, the dense matrix methods available in LAPACK quickly prove unsustainable compared to the other sparse matrix methods for $N \gtrsim 1000$. Considering the remaining implementations, whilst the scaling of SciPy's sparse matrix Chebyshev algorithm scales similarly to a single process of *pyCTQW* for large N (indicating perhaps overhead related to the interpretive nature of Python), the

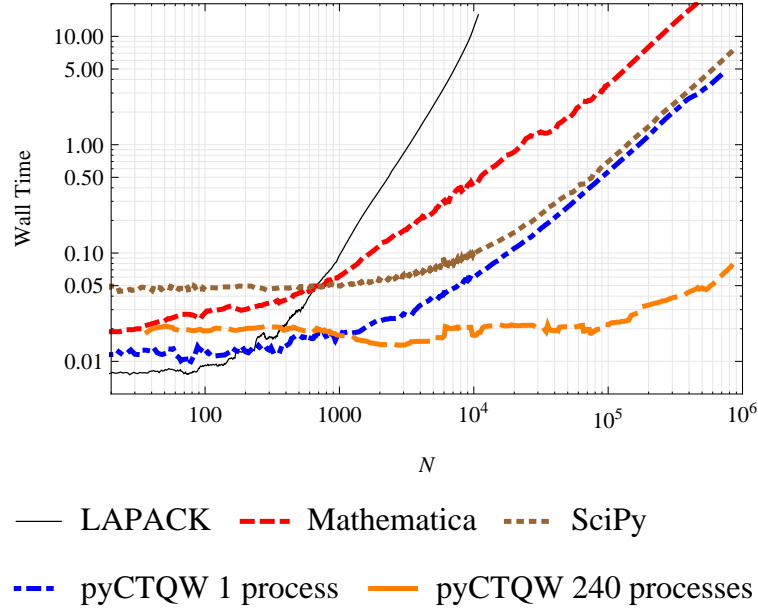


Figure 4.5: The CPU wall time for a two particle non-interacting CTQW on a finite line of N nodes, propagated for $t = 10$ using a variety of different software libraries/packages as shown by the legend. All simulations are computed using the Chebyshev algorithm, with the exception of *Mathematica* where the in-built `MatrixExp[]` function is used.

small N behaviour is almost an order of magnitude slower.

Thus, coupled with the two orders of magnitude speedup exhibited by *pyCTQW* when parallelized for $N \sim 10^6$, it can be seen that *pyCTQW* easily outperforms various other implementations for large CTQW systems, whilst still remaining competitive for N small.

4.5.2 Parallel scaling

To analyze the scalability of *pyCTQW*, both one-particle and two-particle CTQWs on a finite line of N vertices were computed using p MPI threads. In both cases, the CTQW was propagated for $t = 5$ with defect $\Gamma = 2|0\rangle\langle 0| + 1.5|1\rangle\langle 1|$. With one particle, propagation commenced from initial state

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle),$$

whereas when two-particles were used, propagation commenced from initial state

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}} (|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$$

with an interaction term of $\Gamma_{int}(1)$.

Fig. 4.6 highlights the scaling results for the one-particle problem set. Similar to what was seen earlier, the Chebyshev algorithm is approximately one order of magnitude faster than the Krylov algorithm, however both algorithms exhibit analogous scaling behaviour ($\sim O(1/p)$).

The two-particle problem set results are plotted in Fig. 4.7, this time with parallel speedup (defined as $S_p = T_1/T_p$, where T_p is the wall time for p MPI threads) and parallel efficiency (defined as $E_p = \frac{T_1}{pT_p} = S_p/p$) displayed. Here, it can be seen that both the Krylov and Chebyshev algorithms demonstrate impressive speedup, with little difference between them, although the Chebyshev appears slightly more advantageous. The default SLEPc eigenvalue solver, in contrast, shows a marked drop in efficiency as the number of processors increases, indicating significant communication overhead. This has the potential to affect the *pyCTQW* simulation time when using the Chebyshev algorithm in the case where the maximum eigenvalues cannot be efficiently calculated.

An accuracy comparison of the Chebyshev and Krylov algorithms has been previously presented by Bergamaschi and Vianello [55], and so has not been reproduced here. Of particular note are their findings that the relative error of the Chebyshev approximation is either comparable or significantly reduced compared to the relative error of the Krylov algorithm, with all errors rising with t . Furthermore, a low sensitivity to eigenvalue estimation on the convergence of the Chebyshev series was also demonstrated, justifying the avoidance of costly eigenvalue solvers that in some cases may result in bottlenecks.

4.6 Conclusions

In this chapter, we presented a distributed memory software framework, with an object-oriented Python interface *pyCTQW*, that allows the efficient simulation of CTQW-based systems. This package also provides easy methods of visualisation and data analysis. We also detailed the Chebyshev series expansion for the calculation of the matrix exponential, and provided performance data indicating the resulting high degree of speedup achieved over other well established algorithms, such as Krylov-subspace methods. As such, the *pyCTQW* package will serve as an important software tool going forth for many of the numerical simulations presented in this thesis.

With future versions of *pyCTQW*, features considered for inclusion include open system quantum walks using superoperators, and extending the allowable number of simultaneous

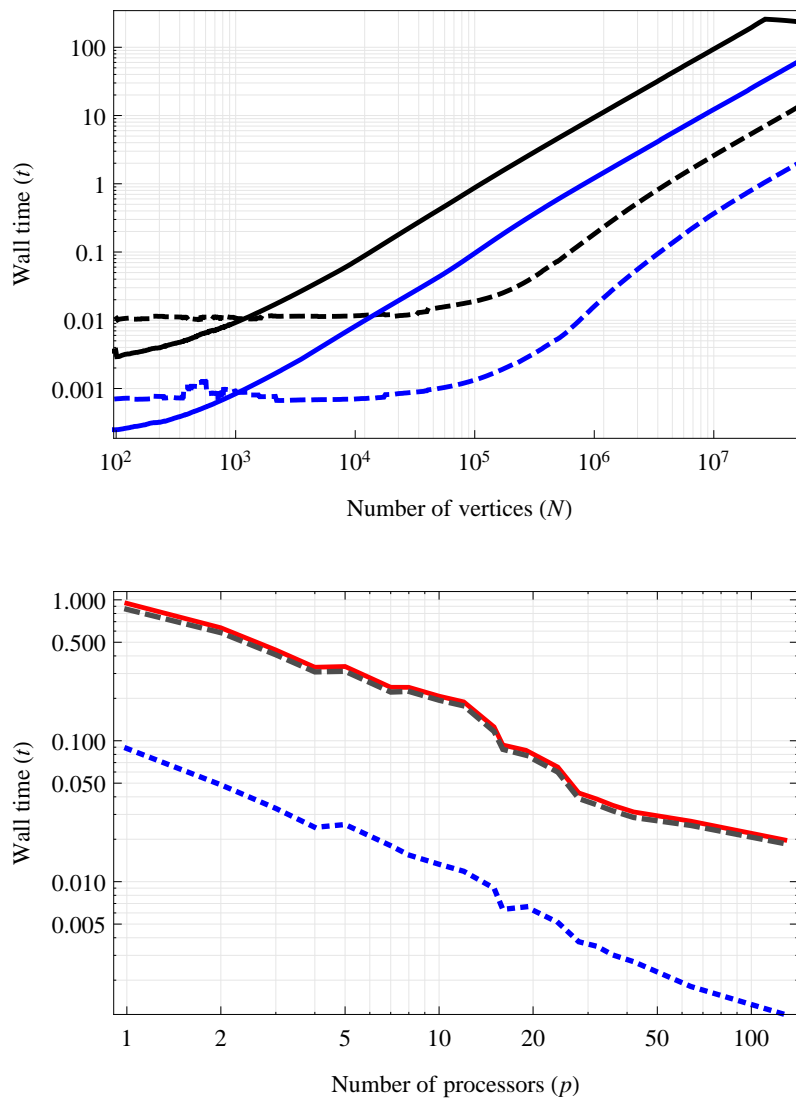


Figure 4.6: Top: The CPU wall time for a one particle CTQW on a finite line of N nodes, propagated for $t = 5$ using the Krylov (black) and Chebyshev (blue) algorithms for 1 MPI thread (solid line) and 128 MPI threads (dashed line). Bottom: Strong scaling behaviour for problem size $N = 10^5$ shown for the Krylov (black, dashed), Chebyshev (blue, dotted) and Krylov-Schur eigenvalue solver (red, solid).

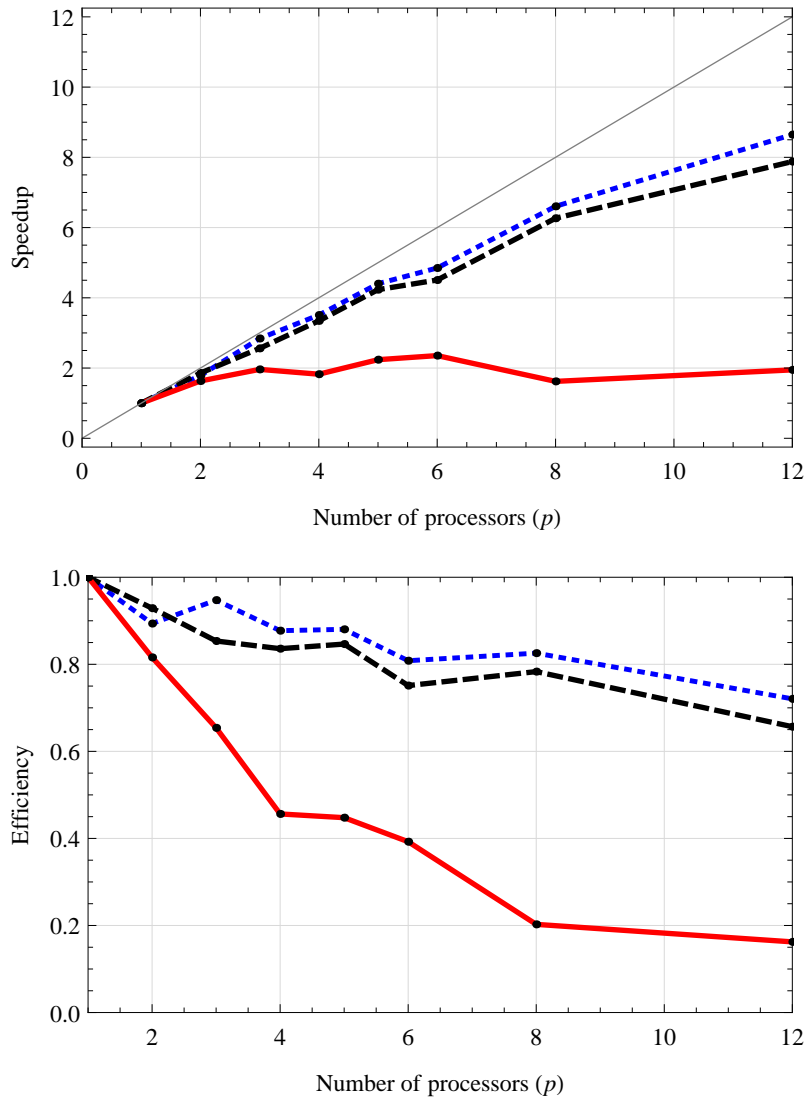


Figure 4.7: The speedup (top) and efficiency (bottom) for a two particle interacting CTQW on a finite line of $N = 150$ nodes, propagated for $t = 5$ using the Krylov (black, dashed) and Chebyshev (blue, dotted) algorithms, with the Krylov–Schur eigenvalue solver also shown (red, solid). The solid gray diagonal line indicates ideal speedup.

quantum walkers. Furthermore, additional code profiling and testing is likely to result in optimisations and increased performance.

4.7 Acknowledgements

This work was supported through the use of computing facilities provided by iVEC (the Western Australian Supercomputing Center: Partner Merit Allocation Project 772). J. A. Izaac would also like to acknowledge valuable discussions with Chris Harris and Chris Bording at iVEC@UWA, and would like to thank the Hackett foundation and The University of Western Australia for financial support.

CHAPTER 5

Multiple fermion simulation

This chapter is based on the manuscript by J. A. Izaac and J. B. Wang, “Systematic dimensionality reduction for continuous-time quantum walks of interacting fermions”, with minimal modification to match the formatting, notation, and terminology of this thesis. The paper has been submitted for publication in Physical Review E.

As noted in the previous chapter, if we wish to extend the quantum walk to simulate P distinguishable particles on graph G , the Hamiltonian of the system is expanded to act on a N^P Hilbert space — in effect, simulating the multi-particle CTQW on graph G via a single particle CTQW propagating on the cartesian graph product $G^{\square P}$. The properties of the cartesian graph product have been well studied [107], and classical simulation of multi-particle CTQWs are common in the literature [119, 120]. However, the above approach is generally applied as is when simulating indistinguishable particles as well, with the particle statistics then applied to the propagated N^P state vector to determine walker probabilities. In this chapter, we address the question: how can we modify the underlying graph structure G in order to simulate multiple indistinguishable CTQWs with a reduction in the size of the state space?

5.1 Introduction

Whilst single-particle CTQWs have historically formed the basis of many theoretical and experimental results and applications, this is rapidly changing; interacting multi-particle CTQWs have been shown to provide a method of universal computation [42], and have been considered in applications such as graph isomorphism algorithms [5, 6]. The dynamics of multiple interacting indistinguishable CTQWs have also been investigated on one-dimensional optical lattices [121] as well as cycle graphs [120]. Furthermore, multi-fermion CTQWs have been experimentally realised via 2D and 3D arrayed waveguides [122], taking advantage of non-classical correlations to probe underlying network structure to a greater degree than possible with a single particle [119].

When simulating non-interacting multi-particle CTQWs either classically or numerically, it is common practice to evolve the time-evolution operator $U(t)$ for the single-particle case, before calculating the multi-particle time-evolution operator via application of the Kronecker product ($U(t) \otimes U(t) \otimes \cdots \otimes U(t)$; see Eq. 3.30). If the particles are indistinguishable, we may then project the resulting state vector onto the subset of allowed symmetrised or antisymmetrised states, for bosons and fermions respectively. However, in the case where interactions are present between the particles, the time-evolution operator is no longer separable; for P particles walking on an N vertex graph G , the N^P -dimensional system Hamiltonian must first be constructed and simulated over time, a computationally expensive process for large graphs and/or numerous particles. Instead, by projecting the CTQW graph Hamiltonian onto the symmetrised/antisymmetrised subspace *prior* to calculating the time-evolution operator, the ultimate size of the unitary operator may be reduced, leading to a reduction in resources required to simulate the system.

In fact, this basic idea has been employed in various forms in the study of CTQW-based algorithms. In their analysis of two non-interacting fermionic CTQWs on graph G with corresponding adjacency matrix A , Gamble et al. [5] noted that, using various commutation relations, it is possible to express the adjacency matrix of $G \square G$, $A \oplus A$, in the basis of allowed fermionic states,

$$\left(\frac{\langle i, j | - \langle j, i |}{\sqrt{2}} \right) (A \oplus A) \left(\frac{|k, \ell\rangle - |\ell, k\rangle}{\sqrt{2}} \right) = \delta_{ik} A_{\ell j} + \delta_{j\ell} A_{ik} - \delta_{i\ell} A_{jk} - \delta_{jk} A_{i\ell}. \quad (5.1)$$

Here, $|i, j\rangle = |i\rangle \otimes |j\rangle$ is an orthonormal basis state representing particle 1 and particle 2 localised at vertices v_i and v_j respectively, and $\{\frac{1}{\sqrt{2}}(|i, j\rangle - |j, i\rangle) : i, j = 1, \dots, N, i < j\}$ the set of antisymmetrised states. This then enabled the authors to apply the Bose-Mesner algebra to examine the distinguishing ability of two non-interacting fermions on strongly regular graphs; concluding that the non-interacting fermionic CTQW does not provide the basis for a solution to the graph isomorphism problem (more on this in chapter 6).

Attempting to solve a very similar problem regarding single-particle CTQWs, it was recognised by Novo et al. [123] that certain graphs possess symmetry, that, in effect, restrict the CTQW dynamics to a subspace of the N -dimensional Hilbert space spanned by the vertices in the graph. As a consequence, the Hamiltonian need not be described by the $N \times N$ adjacency matrix; the CTQW can instead be simulated via the matrix exponentiation of a smaller, *reduced* Hamiltonian corresponding to a weighted graph on $N' \leq N$ vertices. This result was then utilised to analyse problems in CTQW-based quantum search and quantum transport algorithms.

In both cases it was shown that by utilising the underlying symmetry of the system to map the CTQW on a graph to a smaller, reduced-vertex graph structure, analysis of the system under study, as well as the dynamics of the walk, can be more intuitively understood and more efficiently simulated. In this chapter, we show that a similar approach can be applied to P -particle interacting fermionic CTQWs, and provide methods of directly calculating the antisymmetrised reduced adjacency matrices. As a result, as the number of interacting fermions increases, the computational resources required no longer increases exponentially. Accordingly, this may lead to methods of rapid numerical and experimental implementation, as well as simpler methods of analytic study.

This chapter is structure as follows. In Sec. 5.2, we introduce notation and definitions describing multiple indistinguishable particles on a graph structure. We then detail in Sec. 5.3 the antisymmetrised CTQW Hamiltonian and construction of the antisymmetrised reduced graph, and provide an explicit example in the case of the two-fermion CTQW on a 4-vertex star graph. This framework is then extended to include multiple interacting fermions in Sec. 5.5. Finally, Sec. 5.6 discusses the resources required to classically simulate the reduced graph, before we provide concluding remarks in Sec. 5.7.

5.2 Indistinguishable particles

Let

$$|j_1, j_2, \dots, j_P\rangle \equiv |j_1\rangle \otimes |j_2\rangle \otimes \dots \otimes |j_P\rangle \in \mathcal{H}^{N^P} \quad (5.2)$$

denote the state of the system described by P particles located on vertices v_{j_1} to v_{j_P} respectively, where $j_i \in [1, N]$ for all $i = 1, \dots, P$. By construction, this set of states are orthonormal:

$$\langle j'_1, j'_2, \dots, j'_P | j_1, j_2, \dots, j_P \rangle = \langle j'_1 | j_1 \rangle \langle j'_2 | j_2 \rangle \dots \langle j'_P | j_P \rangle = \delta_{j'_1 j_1} \delta_{j'_2 j_2} \dots \delta_{j'_P j_P}, \quad (5.3)$$

and thus the operator

$$\hat{x}_k = \sum_{j_1=1}^N \dots \sum_{j_P=1}^N j_k |j_1, j_2, \dots, j_P\rangle \langle j_1, j_2, \dots, j_P|, \quad 1 \leq k \leq P \quad (5.4)$$

returns the vertex location of the k th particle:

$$\hat{x}_k |j_1, j_2, \dots, j_k, \dots, j_P\rangle = j_k |j_1, j_2, \dots, j_k, \dots, j_P\rangle. \quad (5.5)$$

If we then evolve the system for time t from an initial state $|\psi_0\rangle$, the resulting probability of P distinguishable particles being localised on a particular set of vertices $\{j_1, j_2, \dots, j_P\}$ is given by

$$\mathbb{P}_{j_1, \dots, j_P}(t) = |\langle j_1, j_2, \dots, j_P | U(t) | \psi_0 \rangle|^2. \quad (5.6)$$

However, if the particles are instead *indistinguishable*, then we can no longer distinguish between states where different particles occupy the same sets of vertices. The following therefore represent *equivalent states*,

$$|j_1, j_2, \dots, j_P\rangle \equiv |j_{\sigma(1)}, j_{\sigma(2)}, \dots, j_{\sigma(P)}\rangle \quad (5.7)$$

where σ represents a permutation acting on P elements $\{1, 2, \dots, P\}$.

In general, the two classes of identical particles we will be considering are bosons and fermions, exhibiting symmetric and antisymmetric statistics respectively. These statistics will need to be taken into account when performing measurements of the two-particle system described above. For example, in the two-particle case $P = 2$, the probability of finding the system localised at vertices v_i and v_j is given by

$$\mathbb{P}_{i,j}(t) = \begin{cases} \frac{1}{2} |\langle i, j | U(t) | \psi_0 \rangle + (-1)^s \langle j, i | U(t) | \psi_0 \rangle|^2, & i \neq j \\ |(1 + (-1)^s) \langle i, j | U(t) | \psi_0 \rangle|^2, & i = j \end{cases} \quad (5.8)$$

where $s = 0$ for bosons, 1 for fermions.

A significant drawback to simulating an identical multi-particle CTQW in this fashion is that the measurement statistics are only applied *after* the time-evolution has been calculated – requiring the computation of an $N^P \times N^P$ operator in order to calculate the time-evolution of significantly fewer than N^P states. One way we can avoid this issue is by symmetrising/anti-symmetrising the graph G itself, and removing the redundant states in the simulation.

5.3 The antisymmetrized CTQW Hamiltonian

Fermion statistics requires antisymmetrized states – that is, only states that are antisymmetric under the particle exchange operator $\hat{P}_{x,y}$ are allowed; $\hat{P}_{x,y} |\psi\rangle = -|\psi\rangle$. Here, the

particle exchange operator acts to swap the locations of particles x and y , as follows:

$$\hat{P}_{x,y} |j_1, \dots, j_x, \dots, j_y, \dots, j_P\rangle = |j_1, \dots, j_y, \dots, j_x, \dots, j_P\rangle. \quad (5.9)$$

Note that this property cannot hold if $j_x = j_y$ for any x and y – thus two or more fermions cannot be in the same state/be located at the same vertex.

A state vector in our N^P Hilbert space can be made antisymmetric by the antisymmetrizing operator \mathcal{A} . This acts in the following fashion,

$$\mathcal{A} |j_1, \dots, j_P\rangle = \frac{1}{P!} \sum_{\sigma \in S_P} \text{sgn}(\sigma) |j_{\sigma(1)}, \dots, j_{\sigma(P)}\rangle, \quad (5.10)$$

where σ is a member of the permutation group S_P , representing all permutations acting on $\{1, 2, \dots, P\}$, and $\text{sgn}(\sigma) = \pm 1$ is the *sign* or *signature* of the particular permutation σ — providing a value of 1 if σ there are an even number of particle transpositions, and -1 otherwise. Alternatively, we may write this as a Slater determinant,

$$\mathcal{A} |j_1, \dots, j_P\rangle = \frac{1}{P!} \begin{vmatrix} |j_1\rangle_1 & \cdots & |j_1\rangle_P \\ \vdots & \ddots & \vdots \\ |j_P\rangle_1 & \cdots & |j_P\rangle_P \end{vmatrix}, \quad (5.11)$$

where $|\rangle_p$ denotes the state of the p th particle.

Several properties to note about the antisymmetrizing operator:

1. \mathcal{A} is **Hermitian**: $\mathcal{A}^\dagger = \mathcal{A}$. Thus, \mathcal{A} has real eigenvalues and a complete set of orthonormal eigenvectors.
2. \mathcal{A} is **singular**: \mathcal{A}^{-1} does not exist. This follows from the observation that \mathcal{A} , if applied to a symmetrised state $|\psi^S\rangle$, gives 0; $\mathcal{A}|\psi^S\rangle = 0$. This indicates that \mathcal{A} has a non-trivial nullspace, and thus \mathcal{A} is singular or non-invertible. Therefore, if we use \mathcal{A} to antisymmetrize an arbitrary state, *we can never recover the initial state* – the symmetric components are gone for good.
3. \mathcal{A} is **idempotent**: $\mathcal{A}^n = \mathcal{A}$. Its eigenvalues are either 0 or 1.
4. $[\mathcal{A}, H] = 0$: since \mathcal{A} commutes with the Hamiltonian¹, antisymmetry is a constant of motion. That is, if the CTQW is initialised in an antisymmetric state, it will remain in an antisymmetric state.

¹This follows from the well-known property that \mathcal{A} commutes with *any* Hermitian operator [124].

5. Since \mathcal{A} is idempotent, its trace will always be an integer, and equal to the rank.

Thus, from Eq. 5.10, it can be seen that $\text{Tr}(\mathcal{A}) = \text{rank}(\mathcal{A}) = \binom{N}{P}$.

As fermion statistics excludes two or more particles from coexisting in the same state, without any loss of generality let's consider the subset of ordered states (using lexicographical ordering) in our orthonormal N^P Hilbert space, where no more than one particle is located on any vertex:

$$|\phi_n\rangle \in \{|j_1, j_2, \dots, j_P\rangle : j_1 < j_2 < \dots < j_P\}. \quad (5.12)$$

Here, n denotes the n th element of this subset, and

$$n = 1, 2, \dots, \binom{N}{P}. \quad (5.13)$$

By antisymmetrizing these states, we generate a complete basis set of antisymmetrized states, denoted $|F_n\rangle$,

$$|F_n\rangle = \sqrt{P!} \mathcal{A} |\phi_n\rangle, \quad n = 1, 2, \dots, \binom{N}{P}, \quad (5.14)$$

where the $\sqrt{P!}$ is required for normalisation. Note that

$$\mathcal{A} |F_n\rangle = \sqrt{P!} \mathcal{A}^2 |\phi_n\rangle = \sqrt{P!} \mathcal{A} |\phi_n\rangle = |F_n\rangle. \quad (5.15)$$

This follows from the fact that \mathcal{A} is idempotent; the states $|F_n\rangle$ are already antisymmetrized, so further applications of \mathcal{A} have no effect. Furthermore, as $\text{rank}(\mathcal{A}) = \binom{N}{P}$, and we have $\binom{N}{P}$ states $|F_n\rangle$, we can conclude that the set of basis states $|F_n\rangle$ represent the orthonormal eigenvectors of \mathcal{A} with eigenvalue 1. Thus

$$\mathcal{A} = \sum_{n=1}^{\binom{N}{P}} |F_n\rangle \langle F_n|. \quad (5.16)$$

Note that, since \mathcal{A} is singular, the right hand side does *not* span the entire Hilbert space — this can be easily seen from the basis restriction in Eq. 5.12. In the case of two particles ($P = 2$), this expression for \mathcal{A} simplifies down to

$$\mathcal{A} = \frac{1}{2} \left(I - \sum_{i=1}^N \sum_{j=1}^N |j, i\rangle \langle i, j| \right), \quad (5.17)$$

which agrees with the form of the antisymmetrised operator $(I - S)/2$ used by Gamble

et al. [5].

We now project the P -particle Hamiltonian $H^{\oplus P}$ onto our antisymmetrized basis:

$$\tilde{H}_{ij} = \langle F_i | H^{\oplus P} | F_j \rangle = P! \langle \phi_i | (\mathcal{A} H^{\oplus P} \mathcal{A}) | \phi_j \rangle, \quad i, j = 1, 2, \dots, \binom{N}{P}. \quad (5.18)$$

Therefore, rather than simulate a P -particle fermionic CTQW via a single particle on a N^P vertex graph $G^{\square P}$ (as we must do when working with distinguishable particles), we can instead simulate the dynamics via a single particle CTQW on a $\binom{N}{P}$ vertex graph $\tilde{G} = \mathcal{A} G^{\square P} \mathcal{A}$, where vertex v_i corresponds to antisymmetrised state $|F_i\rangle$.

5.3.1 Example: two fermion walk on a 4-vertex star graph

Let's consider a quick example where we have two fermions ($P = 2$) walking on a 4-vertex star graph G , shown in Fig. 5.1.

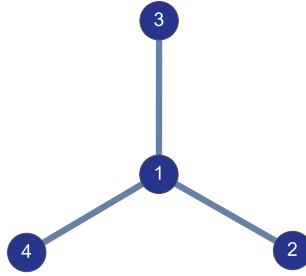


Figure 5.1: 4-vertex star graph

The adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad (5.19)$$

and the allowed states $|\phi_n\rangle$ are

$$|\phi_n\rangle \in \{|1, 2\rangle, |1, 3\rangle, |1, 4\rangle, |2, 3\rangle, |2, 4\rangle, |3, 4\rangle\}_n \quad (5.20)$$

(we have $\binom{4}{2} = 6$ states, as expected). Applying the antisymmetrizing operator as per

Eq. 5.11 provides us with the resulting fermionic basis states:

$$|F_n\rangle = \sqrt{2}\mathcal{A}|\phi_n\rangle = \left\{ \frac{1}{\sqrt{2}}(|1, 2\rangle - |2, 1\rangle), \frac{1}{\sqrt{2}}(|1, 3\rangle - |3, 1\rangle), \frac{1}{\sqrt{2}}(|1, 4\rangle - |4, 1\rangle), \right. \\ \left. \frac{1}{\sqrt{2}}(|2, 3\rangle - |3, 2\rangle), \frac{1}{\sqrt{2}}(|2, 4\rangle - |4, 2\rangle), \frac{1}{\sqrt{2}}(|3, 4\rangle - |4, 3\rangle) \right\}_n. \quad (5.21)$$

From here, it is straightforward to project $A^{\oplus 2} = A \oplus A$ onto the new antisymmetric basis states $|F_n\rangle$. One such way is to note that the basis states for $A \oplus A$, can be written as a unit vector:

$$|i, j\rangle \rightarrow \hat{e}_{2(i-1)+j} \in \mathbb{R}^{N^2}. \quad (5.22)$$

Using this representation, the set of fermionic basis states given by Eq. 5.21 become the columns of our change of basis matrix, $M \in \mathbb{R}^{N^2 \times \binom{N}{P}}$,

$$M_{ij} = |F_j\rangle_i. \quad (5.23)$$

For this particular example, where $N = 4$ and $P = 2$, this gives

$$M^\dagger = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}. \quad (5.24)$$

Therefore, the reduced fermion adjacency matrix is given by

$$\tilde{A} = M^\dagger(A \oplus A)M = \begin{bmatrix} 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad (5.25)$$

and a simulation requiring a 16×16 time-evolution operator has been reduced to a 6×6 time-evolution operator², and it can be readily verified that this gives the same result as Eq. 5.1. Thus, the 2-fermion CTQW on the 4-vertex star graph is *equivalent* to a single

²Note that, as M is a change of basis transformation, $M^\dagger M = I$. Furthermore, we have the result

particle walk on a 6-vertex weighted cycle graph, as shown in Fig. 5.2(b).

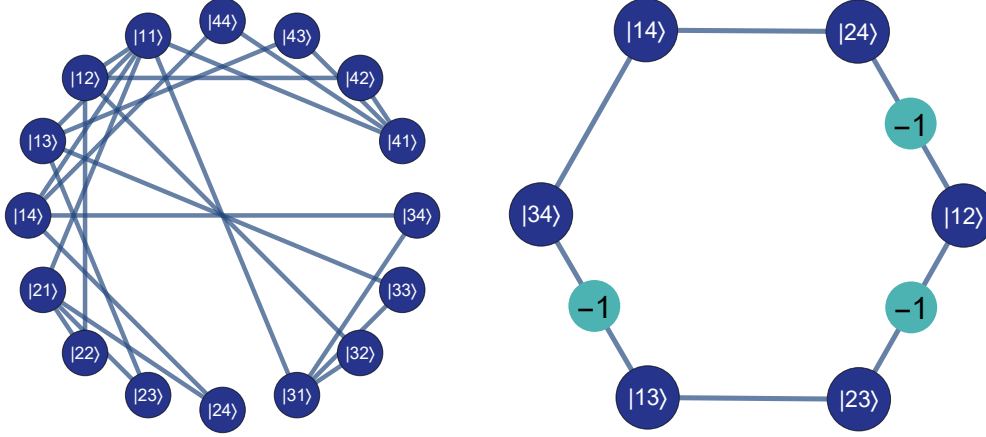


Figure 5.2: Left: The cartesian product of the star graph with itself, $G \square G$. Right: the reduced antisymmetric 2-fermion star graph \tilde{G} . The edges $(|12\rangle, |23\rangle)$, $(|12\rangle, |24\rangle)$, and $(|13\rangle, |34\rangle)$, have an edge weighting of -1 due to particle exchange.

Before we continue, a slight aside; it turns out that by choosing a different labelling for our fermionic states, we can minimise the number of negative values occurring in the reduced antisymmetrised adjacency matrix. Indeed, repeating the above process with the following set of allowed states,

$$|\phi'_n\rangle = \{|1, 2\rangle, |3, 2\rangle, |3, 1\rangle, |3, 4\rangle, |1, 4\rangle, |2, 4\rangle\}_n, \quad (5.26)$$

produces the reduced fermionic adjacency matrix

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (5.27)$$

Unfortunately, there does not appear a systematic way to determine the proper basis to minimise the number of weighted edges, without checking all 2^N possible choices of bases [5]. As such, we will continue to assume lexicographical ordering of allowed fermionic basis states from hereon.

$MM^\dagger = \mathcal{A}$. Thus, we are implicitly antisymmetrizing the system:

$$\tilde{A}^{(2)} = M^\dagger(A \oplus A)M = (M^\dagger M)M^\dagger(A \oplus A)M = M^\dagger \mathcal{A}(A \oplus A)M.$$

5.4 Constructing the reduced fermionic adjacency matrix

In the previous example, note that we calculated $A^{\oplus P}$ before performing the fermionic change of basis operation; i.e. we still needed to compute a $N^P \times N^P$ operator during the computation. However, by studying the antisymmetrized graph $\mathcal{A}G^{\square P}\mathcal{A}$, it is possible to define the P -fermion adjacency matrix directly, without the need to calculate the Kronecker sum intermediate. To begin with, we consider the two-fermion case.

Theorem 5.1 (Two-fermion reduced adjacency matrix). *Let $G(V, E)$ be an unweighted, undirected graph of N vertices and $N \times N$ adjacency matrix A . The fermionic basis is given by $|F_i\rangle = \sqrt{2}\mathcal{A}|\phi_i\rangle$, where $|\phi_i\rangle \in \{|r, s\rangle : r < s\}$. The reduced antisymmetrised graph $\tilde{G}(V', E', W')$ for two fermions walking on graph $G(V, E)$ has adjacency matrix given by*

$$\tilde{A}_{ij} = \langle F_i | A^{\oplus 2} | F_j \rangle = \begin{cases} 1, & f_{ik} = f_{jk}, f_{i\ell} \sim f_{j\ell} \\ -1, & f_{ik} = f_{j\ell}, f_{i\ell} \sim f_{jk} \\ 0, & \text{otherwise,} \end{cases} \quad (5.28)$$

where $f_{ik} = \langle \phi_i | \hat{x}_k | \phi_i \rangle$, $k = 1, 2$, $\ell = 1, 2$, and $k \neq \ell$. Note that \tilde{G} is a weighted graph with vertex set size $|V'| = \binom{N}{2} = N(N-1)/2$.

In other words, if we have two states $|\phi_i\rangle = |r, s\rangle$ and $|\phi_j\rangle = |r', s'\rangle$, where $r < s$ and $r' < s'$, then:

- $\tilde{A}_{ij}^{(2)} = 1$ if and only if, comparing the two states, one particle remains at the same vertex, whilst the other particle inhabits adjacent vertices ($v_r = v'_r, v_s \sim v'_s$ **or** $v_r \sim v'_r, v_s = v'_s$);
- $\tilde{A}_{ij}^{(2)} = -1$ if and only if, comparing the two states, *different* particles are at the same vertex, whilst the remaining inhabited vertices are adjacent ($v_r = v'_s, v'_r \sim v_s$ **or** $v_r \sim v'_s, v'_r = v_s$) — there has been a relative particle exchange between the two states.

Thus, it can be seen that Thm. 5.1 is equivalent to Eq. 5.1. By rewriting the above using set notation, it is easy to recognise that we are, in essence, working with the *symmetric difference* of the two states $|\phi_i\rangle$ and $|\phi_j\rangle$.

Definition 5.2 (Symmetric difference). The symmetric difference between the states $|\phi_i\rangle = |j_1, j_2, \dots, j_P\rangle$ and $|\phi_j\rangle = |j'_1, j'_2, \dots, j'_P\rangle$ is the unordered set given by

$$|\phi_i\rangle \ominus |\phi_j\rangle = \{j_1, j_2, \dots, j_P\} \ominus \{j'_1, j'_2, \dots, j'_P\}$$

$$= (\{j_1, \dots, j_P\} \setminus \{j'_1, \dots, j'_P\}) \cup (\{j'_1, \dots, j'_P\} \setminus \{j_1, \dots, j_P\}), \quad (5.29)$$

where \setminus is the set complement, and \cup the set union.

Once calculated, we are then checking to see if this symmetric difference is an element of the edgeset of G . If this is the case, then we assign $\tilde{A}_{ij} = 1$ if the state $|\phi_i\rangle$ after traversing the edge $|\phi_i\rangle \ominus |\phi_j\rangle$ can be sorted in lexicographical ordering with an even number of transpositions (i.e. it has a signature of 1); otherwise, $\tilde{A}_{ij} = -1$. Note that, for convenience, from hereon we will use the notation $|\psi\rangle \rightarrow (i, j)$ to indicate that the state $|\psi\rangle$ has traversed the graph edge (v_i, v_j) , and thus all particles located at vertex v_i are now located at vertex v_j .

This process may still appear slightly esoteric, but can be clarified with a quick example. Consider the 4-vertex star graph $G(V, E)$ in Fig. 5.1, and in particular, the two-particle fermionic basis states $|\phi_1\rangle = |1, 2\rangle$ and $|\phi_4\rangle = |2, 3\rangle$. Taking the symmetric difference of these two states, we get

$$|\phi_1\rangle \ominus |\phi_4\rangle = |1, 2\rangle \ominus |2, 3\rangle = (1, 3). \quad (5.30)$$

Now, we can see that the ordered pair $(1, 3)$ is an edge in the 4-vertex star graph; $(1, 3) \in E$. Thus, we know that $\tilde{A}_{14} = \tilde{A}_{41} = \pm 1$. To determine the sign, we pick one of the starting basis states $|\phi_1\rangle$ or $|\phi_4\rangle$, and allow the quantum particles to traverse the edge $|\phi_1\rangle \ominus |\phi_4\rangle$; e.g. $|\phi_1\rangle \rightarrow |\phi_1\rangle \ominus |\phi_4\rangle = |1, 2\rangle \rightarrow (1, 3) = |3, 2\rangle$. Calculating the signature of this state with respect to the lexicographical ordering, we find $\text{sgn}(|3, 2\rangle) = -1$, and thus $\tilde{A}_{14} = \tilde{A}_{41} = -1$. Repeating this process for all $|\phi_i\rangle$ and $|\phi_j\rangle$ allows us to construct the antisymmetrised adjacency matrix \tilde{A}_{ij} for the 4-vertex star graph (Eq. 5.25).

With this insight, we can generalise this procedure for P particles.

Theorem 5.3 (P -fermion reduced adjacency matrix). *Let $G(V, E)$ be an unweighted, undirected graph of N vertices, with corresponding adjacency matrix A . The fermionic basis is given by $|F_i\rangle = \sqrt{P!} \mathcal{A} |\phi_i\rangle$, where $|\phi_i\rangle \in \{ |j_1, j_2, \dots, j_P\rangle : j_1 < j_2 < \dots < j_P \}$.*

The reduced antisymmetrised graph $\tilde{G}(V', E', W')$ for P fermions walking on graph $G(V, E)$ has adjacency matrix given by

$$\tilde{A}_{ij} = \langle F_i | A^{\oplus P} | F_j \rangle = \begin{cases} \sigma_{ij}, & |\phi_i\rangle \ominus |\phi_j\rangle \in E \\ 0, & |\phi_i\rangle \ominus |\phi_j\rangle \notin E, \end{cases} \quad (5.31)$$

where $\sigma_{ij} = \text{sgn}(|\phi_i\rangle \rightarrow |\phi_i\rangle \ominus |\phi_j\rangle)$. \tilde{G} is a weighted graph, with vertex set size $|V'| = \binom{N}{P}$.

Using this definition to construct the reduced fermionic adjacency matrix for a given graph G , one possibility is to perform two loops over all $i, j = 1, 2, \dots, \binom{N}{P}$, requiring $\binom{N}{P}^2$ iterations. However, the number of loop iterations required can be significantly reduced by instead iterating over all *edges* in the original graph G — reducing the number of iterations required to

$$|E| \binom{N-2}{P-1} \leq N^2 \binom{N-2}{P-1} < \binom{N}{P}^2. \quad (5.32)$$

The algorithm required to generate the P -particle reduced fermion adjacency matrix of graph G by iterating over all edges is outlined in algorithm 5.1, and a functional implementation using *Mathematica* is provided in Program 5.1.

5.5 Interacting fermions

Using this reduced-vertex model, it is relatively simple to include interactions, i.e. modification of the multi-particle Hamiltonian to include a coupling factor Γ that acts on the entire N^P Hilbert space:

$$H = H^{\oplus P} + \Gamma. \quad (5.33)$$

Hereafter, we will use the CTQW convention $H = A$ for simplicity; however, all results derived follow through similarly for the case where $H = L$.

Let us begin by considering distinguishable particles; once we have established our formalism, we can generalise to interacting fermions. Whilst interactions may be chosen arbitrarily, they are generally selected such that any interactions between sets of particles depends only on their pairwise distances on the graph structure [102]. Furthermore, we often restrict ourselves to forms of interactions that conform to observable physical systems; such as purely local interactions (the on-site or Bose-Hubbard model) or nearest-neighbour interactions [42]. On-site interactions occur when two particles inhabit the same vertex; for example, in the two particle case, this can be written as

$$\Gamma = \alpha \sum_{i=1}^N |i, i\rangle \langle i, i|, \quad (5.34)$$

where α is a parameter determining the strength of the interaction. As onsite interactions do not occur in the case of fermions, due to the Pauli-exclusion principle, they will not be considered further.

Algorithm 5.1: Algorithm for constructing the reduced fermionic adjacency matrix, by iterating over all edges in the original graph. In the following pseudo-code, square brackets represent matrices, and curly brackets represent unordered sets.

Function *ReducedFermionAdj* (A, P)
Input : adjacency matrix A , and integer $1 \leq P \leq N$
Output: reduced fermionic adjacency matrix

```

 $N = \text{length}(A)$ ;
 $M = \text{binomial}(N, P)$ ;
edges =  $\{(i, j) : i, j = 1, 2, \dots, N, A_{ij} = 1\}$ ;
adj =  $[0 : i, j = 1, 2, \dots, M]$ ;
for  $(i, j)$  in edges do
    symdiff =  $\{1, 2, \dots, N\} \setminus \{i, j\}$ ;
    for  $k$  in {subsets of symdiff of length  $P - 1$ } do
        s1 = sorted(append( $k, i$ ));
        s2 = sorted(append( $k, j$ ));
        s1E = replaceall(s1,  $i \rightarrow j$ );
        adjfindIndex(s1, N), findIndex(s2, N)} = signature(s1E);
    end
end
return adj

```

Function *findIndex* ($state, N$)
Input : an allowed fermionic state using lexicographical ordering (e.g. $[1, 3]$ represents $(|1, 3\rangle - |3, 1\rangle)/\sqrt{2}$), and the size of the Hilbert space N
Output: the index number of the lexicographically ordered fermionic state

```

 $P = \text{length}(state)$ ;
ind = 1;
iold = 1;
for  $i$  in state do
    ind += binomial( $N, P$ ) - binomial( $N - i + iold, P$ );
     $N -= i - iold + 1$ ;
     $P -= 1$ ;
    iold =  $i + 1$ ;
end
return ind

```

```

In[1]:= ReducedFermionAdj[A_, p_] := Module[{n = Length[A], states, m, edges},
  n = Length[A];
  m = Binomial[n, p];
  edges = Most[ArrayRules[A]][[All, 1]];
  SparseArray[
    Flatten@Table[
      states = Sort /@ {Append#[i[[1]]], Append#[i[[2]]]} &
      /@ Subsets[Complement[Range[n], i], {p - 1}];
      Thread[{findIndex#[i[[1]], n], findIndex#[i[[2]], n]} &
      /@ states -> (Times@@Signature)
      /@ (states[[All,1]] /. {i[[1]] -> i[[2]]}),
      {i, edges}],
    {m, m}]

findIndex[state_, n_] := findIndex[state, n] =
Module[{p = Length[state], ind = 1, iold = 1, ntmp = n},
  Do[ind += Binomial[ntmp, p] - Binomial[ntmp - i + iold, p];
  ntmp -= i - iold + 1; p -= 1; iold = i + 1, {i, state}];
  Return[ind]]

```

Program 5.1: *Mathematica* implementation of the algorithm described in algorithm 5.1, the reduced fermion adjacency matrix. Here, the input A is an undirected, unweighted $N \times N$ adjacency matrix, and $2 \leq p \leq N$ the number of fermions to be simulated. The output is a sparse $\binom{N}{p}$ -dimensional matrix.

Nearest-neighbour interactions occur when particles are located in adjacent vertices; that is, vertices connected by an edge. In the two particle case, this has the following form:

$$\Gamma = \alpha \sum_{(i,j) \in E(G)} (|i, j\rangle \langle i, j| + |j, i\rangle \langle j, i|). \quad (5.35)$$

To generalise this to P particles, we need to first define the *number operator* \hat{n}_i . Applying the number operator \hat{n}_i to a separable state,

$$\hat{n}_i |j_1, j_2, \dots, j_P\rangle = n_i |j_1, j_2, \dots, j_P\rangle, \quad (5.36)$$

results in the eigenvalue $n_i \in \mathbb{N}_0$, a measurement of the number of distinguishable particles occupying vertex i . The number operator can therefore be constructed as follows. Let $|j_i\rangle_m$ denote the sum of all possible P -particle states for the m th particle located at vertex i :

$$|j_i\rangle_m = \sum_{k_1=1}^N \cdots \sum_{k_{P-1}=1}^N |j_{k_1}, \dots, j_{k_{m-1}}, j_i, j_{k_{m+1}}, \dots, j_{k_{P-1}}\rangle. \quad (5.37)$$

i.e. $|j\rangle_m$ acts non-trivially *only* on the location register for the m th distinguishable particle [42]. The particle number operator is then given by

$$\hat{n}_i = \sum_{m=1}^P |j_i\rangle_m \langle j_i|_m, \quad (5.38)$$

allowing us to write the generalised nearest-neighbour interaction term for P particles:

$$\Gamma_{nn} = \beta \sum_{(u,v) \in E(G)} \hat{n}_u \hat{n}_v \quad (5.39)$$

(in the case $P = 1$, it can be easily checked that $\Gamma = 0$). While the above formulation applies to distinguishable particles, this can easily be generalised to fermions by applying the number operator \hat{n}_i to the set of antisymmetrised basis states $|F_n\rangle$.

Noting that the nearest-neighbour interaction is necessarily diagonal, we may construct the reduced fermion interaction operator via a loop iterating over all $\binom{N}{P}$ allowed states and all resulting $\binom{P}{2}$ pairwise combinations, resulting in

$$\frac{1}{2}P(P-1) \binom{N}{P} \quad (5.40)$$

loop iterations.

Algorithm 5.2: Algorithm for constructing the reduced fermionic interaction matrix, by iterating over all allowed fermionic states. In the following pseudo-code, square brackets represent matrices, and curly brackets represent unordered sets.

```

Function ReducedFermionInt ( $A, P$ )
  Input : adjacency matrix  $A$ , and integer  $2 \leq P \leq N$ 
  Output: reduced fermionic interaction matrix
   $N = \text{length}(A)$ ;
   $M = \text{binomial}(N, P)$ ;
   $\Gamma = [ ]$ ;
  for states in {subsets of  $[1, 2, \dots, N]$  of length  $P$ } do
    sum_of_int = 0;
    for  $(i, j)$  in {subsets of states of length 2} do
      sum_of_int +=  $A_{i,j}$ ;
    end
     $\Gamma = \text{append}(\Gamma, \text{sum\_of\_int})$ ;
  end
  return  $\Gamma$ 

```

```
In[2]:= ReducedFermionInt[A_, p_] := Module[{n=Length[A]},
  SparseArray[Band[{1,1}] -> Total@Transpose[
    A[[Sequence@@#]] &
    /@ Subsets[#, {2}] &
    /@ Subsets[Range[n], {p}]]] ]
```

Program 5.2: *Mathematica* implementation of the algorithm described in algorithm 5.2, the reduced fermion interaction operator. Here, the input A is an undirected, unweighted $N \times N$ adjacency matrix, and $2 \leq p \leq N$ the number of fermions to be simulated. The output is a sparse $\binom{N}{p}$ -dimensional diagonal matrix.

Using algorithm 5.1 and algorithm 5.2, we can now consider the resulting reduced graph structure of a P -fermion CTQW on graph G with nearest neighbour interactions. The reduced fermionic graphs representing a P -fermion CTQW with nearest-neighbour interactions on a 4-vertex star graph and a 5-vertex cycle graph are shown in Fig. 5.3 and Fig. 5.4 respectively. Firstly, note the significant reduction in graph size; each reduced graph contains $\binom{N}{P}$ vertices, compared to $G^{\square P}$, which contains N^P vertices. Furthermore, it can be seen that the reduced graphs corresponding to P and $P' = N - P$ both share the same underlying graph structure, simply differing in the location of edge weighting, and, if $\alpha \neq 0$, the number of self-loops per vertex. This result may lead to new techniques of analysis when investigating algorithms based on CTQWs of multiple fermions.

5.6 Computational resources and analysis

Whilst care has been taken in algorithm 5.1 and algorithm 5.2 to reduce the time taken in generating the reduced fermionic graphs, our reliance on high-level *Mathematica* functions in Program 5.1 and Program 5.2 (such as `Binomial[N,P]` and `Subsets[Range[N],{P}]`) obscures the time-scaling complexity of the provided code fragments. Without knowing how these functions are implemented and their corresponding scaling, it is difficult to deduce for certain the analytic scaling behaviour of the overall algorithm. Thus, in order to compare the time complexity of the Kronecker sum method and reduced fermion method of constructing the P -fermion Hamiltonian of CTQW on graph G with nearest-neighbour interactions, a short numerical analysis was undertaken.

The results of the numerical scaling analysis, performed using *Mathematica* 11.0.1.0 with an Intel i7-3770 processor and 16 GB of available RAM, can be seen in Fig. 5.5. Here, the two algorithms under consideration were both applied to the construction of the system Hamiltonian of a P -fermion CTQW on a randomly generated Erdős-Rényi graph $G(N, 0.3)$ with nearest neighbour interactions. It can be seen that both the Kronecker sum and the reduced fermionic graph approach scale as expected; $\mathcal{O}(N^P)$ for the Kronecker sum, and $\mathcal{O}\left(P(P-1)\binom{N}{P}\right)$ for the reduced fermionic graph — with the reduced fermionic graph algorithm significantly faster for $P \gtrsim 5$. Furthermore, as this numerical complexity analysis does not take into account CTQW propagation (i.e. computation of the matrix exponential e^{-iHt}), the overall computational cost of the Kronecker sum approach is likely to far exceed that of the reduced graph approach. Unfortunately, due to memory constraints, we were unable to generate Kronecker sum results for $P > 5$ when $N = 15$, and $P > 4$ when $N = 20$; generation of sparse matrices of dimension larger than 20^5 proving too large for our computational set-up. This supports the need for this reduced fermionic graph framework — indeed, using this framework we were able to simulate a P -fermion CTQW on a 20 vertex graph with interactions for *all* possible values of P .

5.7 Conclusion

Therefore, we see one of main advantages of the reduced fermionic graph approach is in reducing the required *computational resources*; i.e. the classical memory need to perform the computation. For example, approaching the simulation of multiple-fermions as per Sec. 5.3 requires the initial computation of $A^{\oplus P}$, an $N^P \times N^P$ matrix, before performing the change of basis transformation $MA^{\oplus P}M^T$. Computing this directly, and assuming no

optimization/sparse matrix techniques are used, a memory space of size $\mathcal{O}(N^{2P})$ is required — unmanageable with the RAM available on typical desktop computers for graphs of 30 vertices and more than 3 or 4 fermions [125].

By instead constructing the reduced fermionic graph directly, as outlined in this chapter, the required computational resources are significantly reduced, and are bound by $\binom{N}{P}^2$ where

$$\binom{N}{P} = \frac{N!}{P!(N-P)!} = \frac{1}{P!} \prod_{j=0}^{P-1} (N-j). \quad (5.41)$$

An even greater computational advantage may be found if simulated via quantum computation, as simulating the reduced fermionic graph structure would require only $\log_2 \binom{N}{P}$ qubits, rather than P qubits, as shown in Fig. 5.6. This could also potentially lead to methods of reducing resources required in experimental design, especially if issues surround generation of multiple interacting fermions; in fact, performing the reduced fermionic algorithm in ‘reverse’ — mapping a multi-particle CTQW on a graph to a single particle CTQW on a larger, more complex graph — may allow experimentalists to probe large, complex network structures by simply scaling up the number of particles injected into the system. As such, the techniques detailed in this chapter have potential applications across classical simulation, quantum computation, experimental design, and the analysis of CTQW-based quantum algorithms. In the following part, we will study several potential applications of the CTQW, including one in which multi-particle CTQWs have played an increasingly greater role — the graph isomorphism problem.

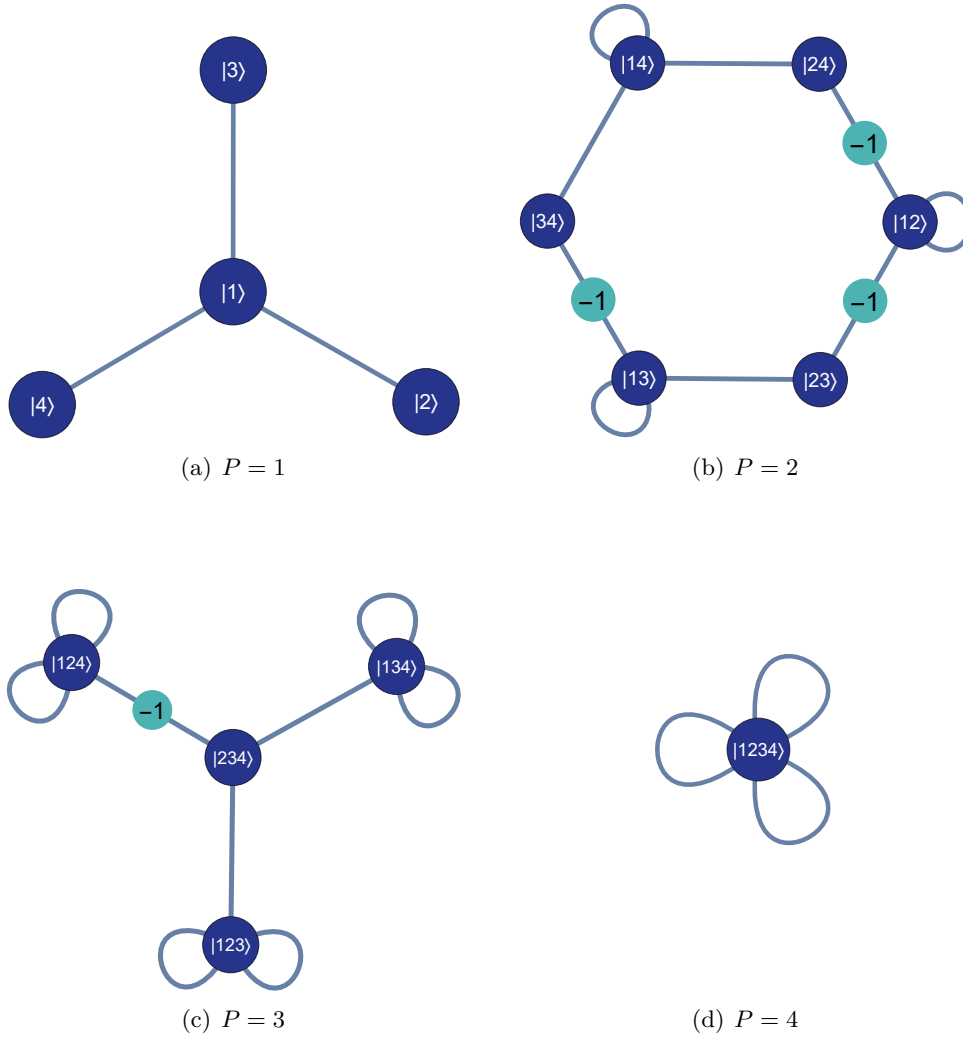


Figure 5.3: The reduced fermionic graph \tilde{G} representing a P -fermion CTQW on a 4-vertex star graph with nearest-neighbour interactions. Edges marked ‘-1’ have an edge weighting of -1, whilst unlabelled edges have an edge weighting of 1. Self loops represent nearest-neighbour interactions of strength α and have an edge weighting of α .

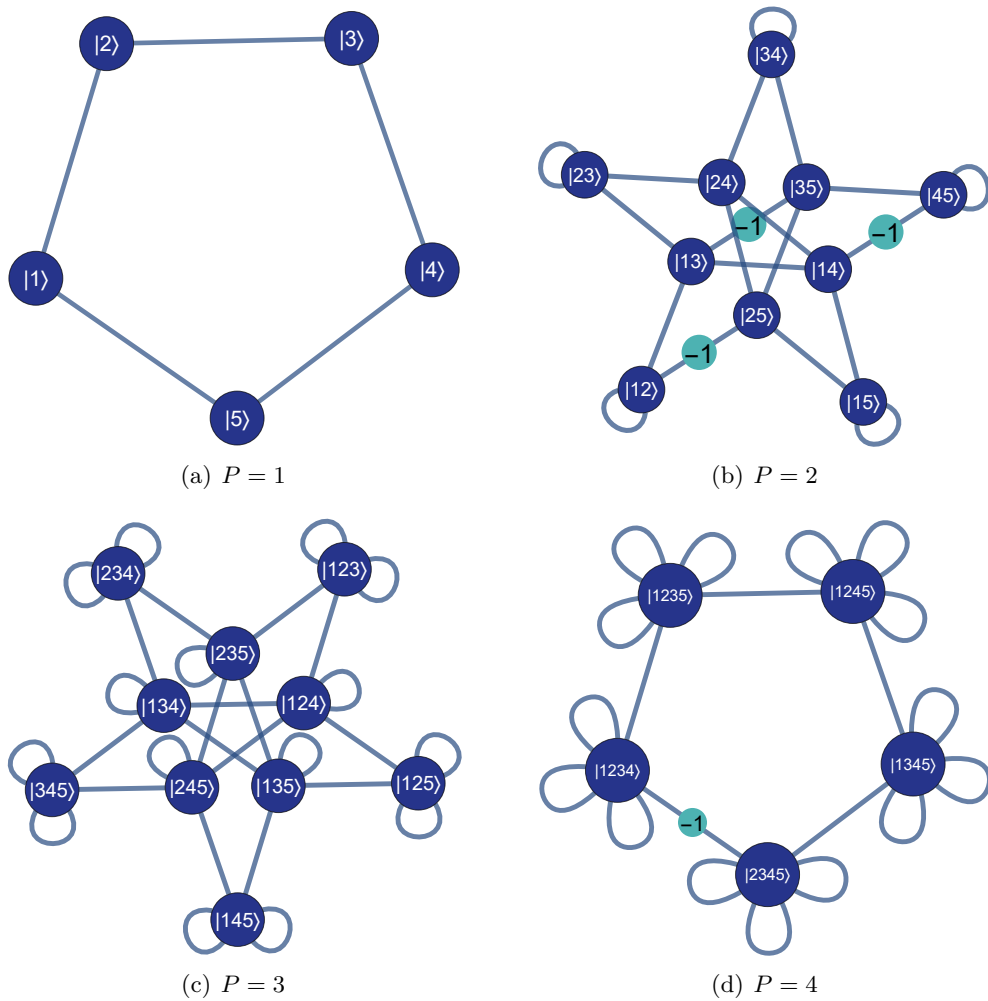


Figure 5.4: The reduced fermionic graph \tilde{G} representing a P -fermion CTQW on a 5-vertex cycle graph with nearest-neighbour interactions. Edges marked ‘-1’ have an edge weighting of -1, whilst unlabelled edges have an edge weighting of 1. Self loops represent nearest-neighbour interactions of strength α and have an edge weighting of α .

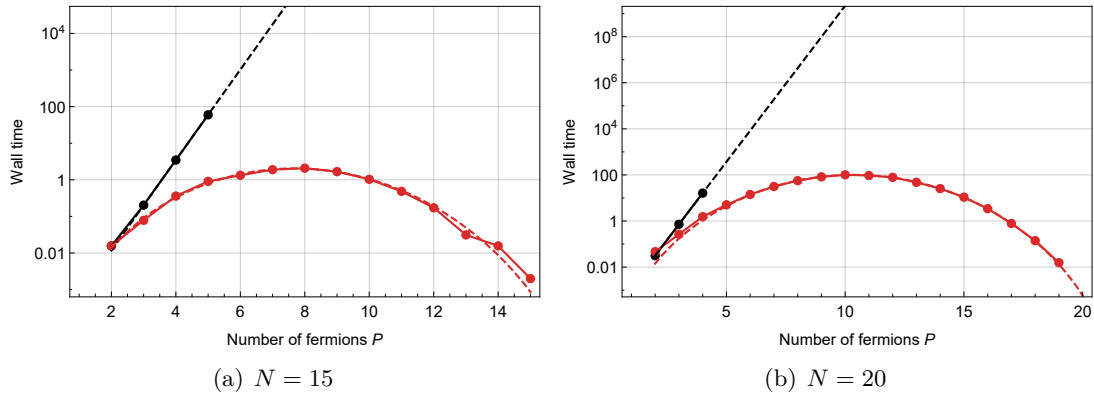


Figure 5.5: Wall time required to construct the P -fermion CTQW Hamiltonian with nearest neighbour interactions, using the Kronecker sum approach $H^{\oplus P} + \Gamma$ (black) and the reduced fermion graph approach detailed in Program 5.1 and Program 5.2 (red), for a randomly generated Erdős-Rényi graph $G(N, 0.3)$. The dashed lines represent regression fits of the form aN^P (black, dashed), and $aP(P-1)\binom{N}{P}$ (red, dashed), for constant a .

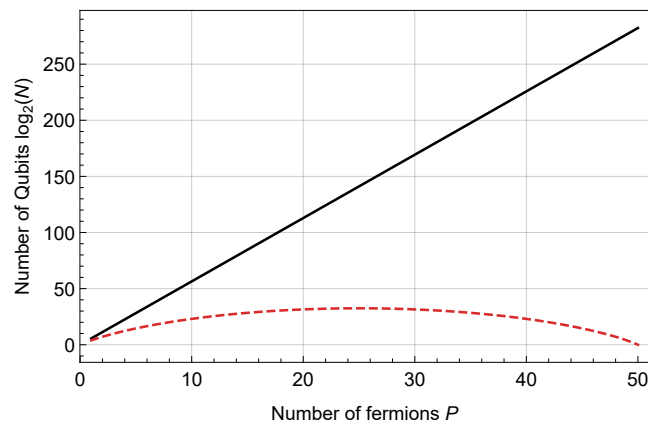


Figure 5.6: Qubits required to simulate a P -fermion CTQW walking on a graph of N vertices, by either (a) introducing additional particles to the system (black), or (b) performing a single-particle walk on the equivalent reduced fermionic graph (red, dashed).

PART III

APPLICATION

CHAPTER 6

CTQW and graph isomorphism

*This chapter is based on the paper by A. Mahasinghe, J. A. Izaac, J. B. Wang, and J. K. Wijerathna, “Phase-modified CTQW unable to distinguish strongly regular graphs efficiently”, Journal of Physics A: Mathematical and Theoretical **48**, 265301 (2015). with minimal modification to match the formatting, notation, and terminology of this thesis, and an expanded introduction.*

Over the last decade, quantum walks have received a significantly increased research focus, due in part to its applicability as a computational model in quantum computation, as well as a wide range of potential applications — including network theory, computer science, quantum information and biochemistry. In particular, the application of quantum walks in graph isomorphism testing has been especially noteworthy. Various quantum walk-based algorithms have been developed, aiming to distinguish non-isomorphic graphs with polynomial scaling, within both the discrete-time quantum walk and continuous-time quantum walk frameworks. Whilst both the single-particle DTQW and CTQW have failed to distinguish non-isomorphic strongly regular graph families (prompting the move to multi-particle GI algorithms), the single-particle DTQW has been successfully modified by the introduction of a phase factor, vastly increasing its distinguishing power. In this chapter, we prove that an analogous phase modification to the single particle CTQW does not have the same distinguishing power as its discrete-time counterpart — in particular, it cannot distinguish strongly regular graphs with the same family parameters efficiently.

6.1 Introduction

Graph isomorphism (GI) is an important open problem in mathematics and computer science, with potential applications in a wide variety of fields ranging from network theory to biochemistry — for example, efficient GI algorithms may provide the key to distinguishing molecular representations in cheminformatics [66, 126]. However, the existence of a polynomial-time algorithm for determining graph isomorphism remains a long-standing

unsolved question in computational complexity theory, with the current fastest algorithm scaling as $2^{\mathcal{O}(\sqrt{n \log n})}$ [127]. In fact, the complexity algorithm scaling of the graph isomorphism problem has yet to be definitively ascertained, though it is conjectured to exist in P and not NP-complete [65].

Over the last several years, there have been various attempts to construct a quantum walk-based algorithm to distinguish non-isomorphic graphs, by utilizing the time-evolution of the quantum walk to calculate ‘graph invariants’ or ‘graph certificates’ which, ideally, produce identical results for two graphs if and only if they are isomorphic. In particular, algorithms have been proposed based on the DTQW [15, 17, 67–69] as well as the CTQW [5, 6, 17, 70] — differing mainly in the number of particles involved, the presence of interactions, localised perturbations, and construction of the GI certificate.

In order to test the distinguishing ability of proposed quantum GI algorithms, a common benchmark has become their capacity to discern isomorphic strongly regular graphs (SRGs) with the same graph parameters. As a consequence, this has allowed analytic examination of proposed quantum walk graph certificates on SRGs; quoting Rudinger et al. [17],

If a particular certificate will always fail to distinguish two non-isomorphic SRGs, it is because all elements of a certificate, as well as their multiplicities, are functions of SRG family parameters.

It has subsequently been proven using this approach that conventional single-particle quantum walk algorithms, both discrete-time and continuous-time, cannot distinguish arbitrary non-isomorphic SRGs. To alleviate this issue, Douglas and Wang [15] developed a perturbed one-particle DTQW-based GI algorithm that was able to successfully distinguish non-isomorphic pairs of strongly regular graphs (SRGs) with up to 64 vertices. Multiparticle DTQW algorithms were further explored by Berry and Wang [67]; non-interacting and interacting two-particle DTQWs were used to successfully distinguish specific SRG families. Although lacking the ‘coin flip’ operation found in DTQWs – and thus resulting in a system with reduced degrees of freedom – two-particle interacting CTQW GI algorithms have been proposed that successfully distinguish arbitrary SRGs. Further, it has been proven by Gamble et al. [5] that a non-interacting CTQW cannot distinguish a pair of SRGs with the same family parameters, contrary to the discrete case.

Whilst the single-particle CTQW was discarded by Gamble et al. [5] and Shiau et al. [128] as it was unable to distinguish families of non-isomorphic SRGs with the same family parameters, it is natural to ask: would a *perturbed* one-particle CTQW, modified analogously

to the process used by Douglas and Wang [15] for the DTQW, result in a similar improvement in distinguishing power? This has not previously been explored in the literature, due in part to ambiguities in defining the perturbed CTQW system. In this chapter, we consider several possible methods of modifying the single-particle CTQW, and prove that none of those considered is as powerful as the phase added DTQW described in Douglas and Wang [15]. Furthermore, we also provide an explanation for the cause of the increased distinguishing power of DTQWs.

This chapter is structured as follows. In Sec. 6.2, we introduce a symmetric commutative matrix algebra known as the Bose-Mesner algebra, and apply it to the class of strongly regular graphs. We then briefly describe previously proposed quantum walk-based graph isomorphism algorithms, particularly the CTQW constructed algorithms of Gamble et al. [5], and the DTQW algorithms of Douglas and Wang [15] and Berry et al. [16], in Sec. 6.3. Next, in Sec. 6.4, we modify the single particle CTQW graph isomorphism algorithm to include vertex-dependent phase, and prove using the Bose-Mesner algebra that such an algorithm cannot distinguish two isomorphic graphs within the same strongly regular graph family. Finally, in Sec. 6.5, we detail the multi-particle interacting CTQW GI algorithm available in the *pyCTQW* software package, which utilizes distributed memory and high performance computing for efficient numerical simulation.

6.2 Graphs and matrix algebras

A cellular algebra is the algebra formed from the complex span of non-negative integer matrices with elements 0 or 1, such that the resulting algebra is closed under the Schur product (also known as Hadamard multiplication) and the Hermitian operator, and contains the matrix of all one's J [129]. Further, cellular algebras which contain the identity matrix are known as coherent algebras [130]. As a result, cellular algebras are an important tool for analysing and graph structures and networks. In this section, we will introduce a commutative coherent algebra known as the Bose-Mesner algebra, and define it for the case of strongly regular graphs. This will allow us to fully explore the distinguishing power of CTQWs over strongly regular graphs.

6.2.1 Bose-Mesner algebra

To produce the Bose-Mesner algebra of a graph, an associative and commutative matrix algebra, we first must find a way to generate a linearly dependent spanning set of symmetric

matrices that represent the graph structure, whilst remaining closed under commutative multiplication. To do so, we harness the algebraic combinatorics theory of association schemes [131].

Definition 6.1 (Graph association scheme). A graph association scheme for graph $G(V, E)$, $|V| = N$, is a set of symmetric $N \times N$ adjacency matrices $\mathcal{A} = \{A_0, A_1, \dots, A_{\mathcal{D}}\}$ such that

- (i) $A_0 = I$,
- (ii) $\sum_{\ell=0}^{\mathcal{D}} A_{\ell} = J$, and
- (iii) $A_i A_j = A_j A_i \in \text{span}(\mathcal{A})$.

Using this as a guideline, let's attempt to define an association scheme of adjacency matrices. Let $G(V, E)$ be an undirected strongly connected vertex set with diameter \mathcal{D} . We can then define the so-called *distance graphs* $G_{\ell}(V_{\ell}, E_{\ell})$, in which any two vertices $v_i, v_j \in V_{\ell}$ are adjacent in G_{ℓ} if and only if we have $\text{dist}(v_i, v_j) = \ell$ in graph G . We can extend this analogy to the adjacency matrix — the adjacency matrices A_{ℓ} are referred to as the *distance matrices* \mathcal{A}_{ℓ} of G .

Definition 6.2 (Distance matrices). For an undirected graph $G(V, E)$ with diameter \mathcal{D} , the distance matrices are given by

$$(\mathcal{A}_{\ell})_{ij} = \begin{cases} 1, & \text{dist}(v_i, v_j) = \ell \\ 0, & \text{dist}(v_i, v_j) \neq \ell, \end{cases} \quad \ell = 0, 1, \dots, \mathcal{D}. \quad (6.1)$$

As a result of this definition, it can be seen that $\mathcal{A}_0 = I$, the identity matrix, and $\mathcal{A}_1 = A$, the adjacency graph of A . Furthermore, since the graph is strongly connected, there exists a unique ℓ such that $(\mathcal{A}_{\ell})_{ij} = 1$ for each $i, j \in [0, N]$, and as a result,

$$\sum_{\ell=1}^{\mathcal{D}} \mathcal{A}_{\ell} = J. \quad (6.2)$$

This implies that the matrices \mathcal{A}_{ℓ} are *linear independent*. If we combine the properties Eq. 6.2, the existence of the identity $\mathcal{A}_0 = I$, then we may be tempted to call the set $\{\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_{\mathcal{D}}\}$ a *graph association scheme*. However, as the elements are not guaranteed to commute under matrix multiplication, this is not the case.

In order to ensure that the set of distance matrices of a strongly connected graph form an association scheme, we must restrict ourselves to a subset of graphs known as distance-

regular.

Definition 6.3 (Distance-regular graph). Let $G(V, E)$ be an undirected graph with vertex set $V = \{v_1, v_2, \dots, v_N\}$ and edge set $E = \{e_1, e_2, \dots, e_M\}$, and corresponding adjacency matrix A . The diameter of a graph $G(V, E)$ is *distance-regular* if, for any two vertices $v_x, v_y \in V$, the number of vertices at distance i from v_x ($|\{v_\ell \in V : \text{dist}(v_x, v_\ell) = i\}|$) and the number of vertices at distance j from v_y ($|\{v_\ell \in V : \text{dist}(v_y, v_\ell) = j\}|$) depends *only* upon i, j , and $\text{dist}(v_x, v_y)$.

The quantity is termed the *intersection number* of a distance-regular graph, and, as per the definition above, is defined solely in terms of the variables i, j , and ℓ .

Definition 6.4 (Intersection number). Let $G(V, E)$ be an undirected graph with diameter \mathcal{D} . For all $v_x, v_y \in V$ such that $\text{dist}(v_x, v_y) = \ell$, the intersection number is defined by

$$p_{ij}^{(\ell)} = |\{v_z \in V : \text{dist}(v_x, v_z) = i, \text{dist}(v_y, v_z) = j\}|, \quad i, j = 0, 1, \dots, \mathcal{D}. \quad (6.3)$$

The intersection number is strictly positive ($p_{ij}^{(\ell)} > 0$ for all $i, j, \ell \in [0, \mathcal{D}]$) and symmetric ($p_{ij}^{(\ell)} = p_{ji}^{(\ell)}$).

For this class of graphs, the distance matrices do indeed commute under matrix multiplication, $\mathcal{A}_i \mathcal{A}_j = \mathcal{A}_j \mathcal{A}_i$; in fact, this matrix multiplication can be written out in the distance matrix basis by the relation [132]

$$\mathcal{A}_i \mathcal{A}_j = \sum_{k=0}^{\mathcal{D}} p_{ij}^{(k)} \mathcal{A}_k. \quad (6.4)$$

Thus, we come to the following conclusion.

Theorem 6.5. *The set of distance matrices $\mathcal{A} = \{A_0, A_1, \dots, A_{\mathcal{D}}\}$ for a distance-regular graph $G(V, E)$ of diameter \mathcal{D} forms a graph association scheme.*

Now that we have a valid graph association scheme for distance-regular graphs, we may construct the Bose-Mesner matrix algebra:

Definition 6.6 (Bose-Mesner algebra). The Bose-Mesner algebra for a graph association scheme $\mathcal{A} = \{A_0, A_1, \dots, A_{\mathcal{D}}\}$ is the $\mathcal{D} + 1$ -dimensional matrix algebra generated by the complex $\text{span}(\mathcal{A})$. Matrix multiplication, as well as Schur multiplication (element-by-element multiplication $(A \circ B)_{ij} = A_{ij} B_{ij}$) are commutative.

One may note that the association scheme basis of the Bose-Mesner algebra is comprised of commuting symmetric matrices — as such, it is a well-known property of linear algebra

that they must be simultaneously diagonalizable by a unitary matrix, i.e. there exists some unitary matrix S such that $S^T \mathcal{A}_\ell S$ is a diagonal matrix. As the basis matrices are also normal ($\mathcal{A}_\ell \mathcal{A}_\ell^\dagger = \mathcal{A}_\ell^\dagger \mathcal{A}_\ell$) It follows that there exists an alternative basis for the Bose-Mesner algebra consisting of minimal orthogonal idempotent matrices [131, 133]. We use this knowledge to define a subalgebra of the Bose-Mesner algebra, known as the dual Bose-Mesner algebra [132, 134, 135].

Definition 6.7 (Dual Bose-Mesner algebra). The dual Bose-Mesner algebra with respect to vertex v_r for distance-regular graph $G(V, E)$ with diameter \mathcal{D} , is given by $\mathcal{F}^{(r)} = \text{span}(\{F_0^{(r)}, F_2^{(r)}, \dots, F_{\mathcal{D}}^{(r)}\})$, where $F_\ell^{(r)}$ is a diagonal orthogonal idempotent matrix ($F_i^{(r)} F_j^{(r)} = \delta_{ij} F_i^{(r)}$) defined by

$$\left(F_\ell^{(r)}\right)_{ii} = \begin{cases} 1, & \text{dist}(v_r, i) = \ell \\ 0, & \text{dist}(v_r, i) \neq \ell. \end{cases} \quad (6.5)$$

The basis satisfies the property $\sum_{\ell=0}^{\mathcal{D}} F_\ell^{(r)} = I$.

6.2.2 Strongly Regular Graphs

Let us now consider a subset of distance-regular graphs known as the strongly regular graphs.

Definition 6.8 (Strongly regular graph). Let $G(V, E)$ be a k -degree regular graph comprising of N vertices and adjacency matrix A that is neither a complete graph ($A \neq J - I$) nor null graph (adjacency matrix $A \neq I$). G is then said to be strongly regular with parameters $\text{srg}(N, k, \lambda, \mu)$ if

- every pair of adjacent vertices have exactly λ common neighbours,
- every pair of non-adjacent vertices have exactly μ common neighbours.

From hereon, we will use the notation $\text{SRG}(N, k, \lambda, \mu)$ to denote an arbitrary strongly regular graph with aforementioned parameters. As an aside, choose a vertex v_i in $\text{SRG}(N, k, \lambda, \mu)$; counting the number of edges incident from vertices adjacent to v_i to vertices non-adjacent to v_i , we find

$$|\{(v_x, v_y) : v_x \sim v_y, v_i \sim v_x, v_i \not\sim v_y\}| = k(k - \lambda - 1). \quad (6.6)$$

Repeating this analysis, but counting in the other direction, it is easy to see that

$$|\{(v_y, v_x) : v_y \sim v_x, v_i \sim v_x, v_i \approx v_y\}| = \mu(n - k - 1). \quad (6.7)$$

As all edges are undirected, it follows that

$$k(k - \lambda - 1) = \mu(n - k - 1), \quad (6.8)$$

and thus the parameters (N, k, λ, μ) are *not* independent.

Note that the SRG satisfies the criteria for distance-regularity. Further, the path between any two non-adjacent vertices has length 2, with the path traversing the two via any one of the μ common neighbours. This leads us to the following corollary:

Corollary 6.9. *A distance-regular graph $G(V, E)$ with diameter $\mathcal{D} = 2$ is strongly regular.*

Thus, the SRG can be described by intersection numbers as follows.

Theorem 6.10. *For a strongly regular graph $SRG(N, k, \lambda, \mu)$, the intersection numbers are given by*

$$p_{ij}^{(0)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & N - k - 1 \end{bmatrix}_{ij}, \quad p_{ij}^{(1)} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & \lambda & k - \lambda - 1 \\ 0 & k - \lambda - 1 & N - 2k + \lambda \end{bmatrix}_{ij},$$

$$p_{ij}^{(2)} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & \mu & k - \mu \\ 1 & k - \mu & N - 2k + \mu - 2 \end{bmatrix}_{ij}. \quad (6.9)$$

Let's consider walks of length 2 in a strongly regular graph more closely. Recall from Eq. 2.5 that $(A^2)_{ij}$ gives us the number of walks of length 2 emanating between vertices v_i and v_j . In the case of the $SRG(N, k, \lambda, \mu)$, it can be seen that

$$(A^2)_{ij} = \begin{cases} k, & i = j \\ \lambda, & i \neq j, v_i \sim v_j \\ \mu, & i \neq j, v_i \not\sim v_j. \end{cases} \quad (6.10)$$

It is easy to verify that this result can be written as the following matrix equation,

$$A^2 = kI + \lambda A + \mu(J - I - A), \quad (6.11)$$

and comparing this with Eq. 6.4, we see that for the SRG, we have intersection numbers $p_{11}^{(0)} = k$, $p_{11}^{(1)} = \lambda$, and $p_{11}^{(2)} = \mu$, as would be expected. Thus, we see that the Bose-Mesner algebra of an SRG is given by the spanning set $\text{span}(\{\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2\})$ where

$$\mathcal{A}_0 = I, \tag{6.12}$$

$$\mathcal{A}_1 = A, \tag{6.13}$$

$$\mathcal{A}_2 = J - I - A. \tag{6.14}$$

Likewise, the dual Bose-Mesner algebra with respect to reference vertex v_r has the basis elements $F_0^{(r)}$, $F_1^{(r)}$, and $F_2^{(r)}$, where

$$(\mathcal{F}_\ell^{(r)})_{ii} = (\mathcal{A}_\ell)_{ri}. \tag{6.15}$$

6.3 Quantum walk-based graph isomorphism algorithms

In order to utilize the quantum walk to determine graph isomorphism, most approaches attempt to make use of the unitary time-evolution of the walker to construct a graph invariant — that is, a function f over the graphs G and H such that $f(G) = f(H)$ if and only if the graphs G and H are isomorphic. Of course, if any such function were proven to exist, this would subsequently solve the graph isomorphism problem. In the study of quantum walk-based graph isomorphism algorithms, the proposed graph invariants are usually denoted *graph certificates*, keeping with the terminology used in the field of computational complexity.

Due to the unknown vertex-labelling permutation relation between two isomorphic graphs, the graph certificate is generally constructed by performing a frequency measurement on the resulting probability distribution of the evolved quantum walk — i.e. producing an ordered list. The various graph certificates proposed have been examined by Rudinger et al. [17], who notes that they form an integral part of the GI algorithm, and must be considered alongside the quantum walk propagation to form the algorithm as a whole. As a result, the choice of graph certificate is an important factor when designing or comparing GI algorithms. Commonly used graph certificates for unperturbed quantum walks include (but are not limited to):

1. The sorted list of elements of the unitary time-evolution operator $U(t)$, after prop-

agating the walker for time t :

$$\text{sort} (\{ \langle i|U(t)|j \rangle : i, j \in [1, N] \}). \quad (6.16)$$

This certificate is generally employed with CTQWs, and can be easily extended to multiple walkers [5, 6].

2. The sorted list of probabilities of finding a discrete-time quantum walker at each vertex v_i after propagating for time t from every possible coin state:

$$\text{sort} \left(\left\{ \sum_{j=1}^{d_i} | \langle v_i, c_j | U(t) | v_k, c_\ell \rangle |^2 : i, k \in [1, N], \ell \in [0, d_k] \right\} \right). \quad (6.17)$$

Whilst generally applied to DTQWs, it can also be extended to CTQWs, where $|v_i, c_j\rangle$ now corresponds to the bosonic (symmetrized) or fermionic (antisymmetrized) state of locating particle 1 and vertex v_i and particle 2 at vertex c_j [17].

3. The sorted list of probabilities of finding two non-interacting discrete-time quantum walkers at each pair of vertices v_i and $v_{i'}$, summed over all time-steps $0 \leq t \leq 2N$, and propagated from every possible bosonic edge-state:

$$\text{sort} \left(\left\{ \sum_{j=1}^{d_i} \sum_{t=0}^{2N} \left| \langle v_i^{(1)}, c_j^{(1)}, v_{i'}^{(2)}, c_{j'}^{(2)} | U(t) | \beta_{kl}^{(+)} \rangle \right|^2 : (v_k, v_\ell) \in E \right\} \right). \quad (6.18)$$

Here, $|\beta_{kl}^{(+)}\rangle = \frac{1}{\sqrt{2}} \left(|v_k^{(1)}, c_\ell^{(1)}, v_\ell^{(2)}, c_k^{(2)}\rangle + |v_\ell^{(1)}, c_k^{(1)}, v_k^{(2)}, c_\ell^{(2)}\rangle \right)$ represents the bosonic edge state where particles 1 and 2 are delocalised over edge $(v_k, v_\ell) \in E$ in a symmetric fashion; this is chosen so as to remain independent of vertex labelling, yet reduce the required number of certificate elements to compute [67].

These approaches are all aimed at solving the graph isomorphism problem in polynomial time by exploiting massive quantum parallelism — by flattening the matrix of transition amplitudes and discarding detailed structural information, they are therefore efficient (i.e. computable in polynomial time). Unfortunately, it has been shown that the three forms of the graph certificate above are *not* universal for non-interacting CTQWs or single particle DTQWs — although their distinguishing power *can* be increased by including interactions in the case of the CTQW [5, 6] or by increasing the number of non-interacting DTQWs [67].

In order to try and reduce the additional computational resources required to implement multi-particle quantum walk GI algorithms, other approaches have been examined in the

literature. For example, Emms et al. [68] successfully distinguish certain pairs of SRGs by examining the cospectrality of the unitary time-evolution operators. Alternatively, using a single-particle DTQW with a vertex dependent phase factor, Douglas and Wang [15] compared the resulting probabilities at timestep $t = N^2$, and successfully distinguished certain pairs of SRGs with the same family parameters. It is natural to ask whether the continuous counterpart of this phase addition could be successfully used for GI testing. In the following section, we will introduce several analogous vertex-dependent phase modifications to the single-particle CTQW, and show that the corresponding GI certificates (unlike the DTQW case) do not afford additional distinguishing power to the single-particle CTQW¹.

6.4 Single particle CTQW with phase

In considering the possibilities of modifying the DTQW to increase distinguishing power without extending the system to two particles, Douglas and Wang [15] successfully modified the single particle DTQW by adding a phase, distinguishing several non-isomorphic SRGs with the same family parameters. In their algorithm, a vertex is selected as the reference vertex, and a local phase is applied for transitions to all adjacent vertices. This is achieved by selectively modifying the shifting operator S ; i.e. when acting on the reference vertex v_r , a phase factor is introduced $S|v_r, c_j\rangle = e^{i\theta}|v_j, c_r\rangle$. The graph certificate is then taken as the probabilities at all pairs of vertices (v_i, v_j) of the graph, which makes it uniform up to a vertex permutation.

We consider several possible algorithms for CTQW phase additions, restricting ourselves to those which are uniform for permutations of the vertices. To construct the graph certificate, we then utilize transition amplitudes between all pairs of vertices (Eq. 6.16). It should be noted that this is equivalent to the DTQW graph certificate construction as used by Douglas and Wang [15] — naïvely assuming this follows through in continuous time, we would conclude that whenever the method used by Douglas and Wang [15] can distinguish two vertex transitive graphs (for example the first few distinct pairs of SRGs within the same family), so should the method of CTQW phase additions. Here, we use

¹Of course, if we were simply concerned with maximising the quantum walks distinguishing power, modifying the algorithm to account for the detailed structure of $\langle i|U(t)|j\rangle$ would most likely be the most advantageous approach — and may even be able to distinguish non-isomorphic graphs that produce CTQW time-evolutions with the same list of elements. However, to our knowledge, there does not exist any proposed CTQW GI algorithm that makes use of the structural information in an efficient manner, due to the unknown vertex-labelling permutation relation between two graphs. One could always test all possible permutations of rows and columns between the two $\langle i|U(t)|j\rangle$ matrices, but this would require factorial/exponential time and thus offers no advantage over classical algorithms.

the Bose-Mesner algebra of SRGs to show that the elements of the graph certificate, as well as their multiplicities, are simply functions of the SRG family parameters, and thus this is not the case.

Before we begin, a quick note regarding the Hamiltonian choice for our CTQW GI algorithm. As previously noted in Sec. 3.2.2, amongst the literature, there are several competing definitions of the Hamiltonian, the most popular including the adjacency matrix or the graph Laplacian. However, in the case of degree-regular graphs the time evolution operator using the Laplacian, e^{-iLt} , is simply proportional to e^{iAt} up to a global phase factor. Therefore, as we are working with SRGs, for simplicity and without loss of generality we will select $H = A$.

6.4.1 No phase

We will begin by proving, using the Bose-Mesner algebra, that two non-isomorphic SRGs with the same family parameters cannot be distinguished, before arriving at the explicit continuous counterpart of the phase added DTQW. Finally we consider general phase additions, and compare and contrast the two methods.

Lemma 6.11. *The ordered transition amplitude lists of a single-particle CTQW are identical for two non-isomorphic SRGs with the same family parameters (N, k, λ, μ) .*

Proof. Recall that the Hamiltonian is expressible as $H = A$. Since the basis elements of the Bose-Mesner algebra obey the relations

$$\mathcal{A}_i \mathcal{A}_j = \sum_{\ell=0}^2 p_{ij}^{(\ell)} \mathcal{A}_\ell, \quad (6.19)$$

where $\mathcal{A}_0 = I$, $\mathcal{A}_1 = A$ and $\mathcal{A}_2 = J - I - A$, it can be seen that

$$A^n = \mathcal{A}_1^n = \sum_{\ell_1=0}^2 \cdots \sum_{\ell_{n-1}=0}^2 p_{11}^{(\ell_1)} p_{1\ell_1}^{(\ell_2)} \cdots p_{1\ell_{n-2}}^{(\ell_{n-1})} \mathcal{A}_{\ell_{n-1}}, \quad (6.20)$$

i.e. powers of the adjacency matrix can be written as a linear combination of the Bose-Mesner basis, a consequence of the Bose-Mesner algebra being closed under matrix multiplication. Thus, we have $U(t) \in \mathcal{A}$, and the CTQW unitary time evolution operator can be written as

$$U(t) = e^{-iAt} = \sum_{n=0}^{\infty} \frac{1}{n!} (-it)^n A^n = \alpha \mathcal{A}_0 + \beta \mathcal{A}_1 + \gamma \mathcal{A}_2, \quad (6.21)$$

where α , β and γ are time dependent and depend *only* upon the family parameters (N, k, λ, μ) of the SRG. The graph certificate is thus given by $\{\langle i|U(t)|j\rangle : i, j \in [0, N]\}$, i.e. the list of elements of the matrix $U(t)$. We can now calculate the diagonal and off-diagonal elements of U :

- Diagonal elements (i.e. $i = j$): as $(\mathcal{A}_1)_{jj} = (\mathcal{A}_2)_{jj} = 0$ and $(\mathcal{A}_0)_{jj} = 1 \forall j$, we therefore have N diagonal elements equal to α .
- Off-diagonal elements ($i \neq j$): we have $(\mathcal{A}_0)_{ij} = 0 \forall i, j$, whilst \mathcal{A}_1 and \mathcal{A}_2 contains a mixture of zeros and ones. Since the SRG is k -degree regular, the adjacency matrix will contain k non-zero elements for each N vertex. Thus, there will be kN off-diagonal elements equal to β , and $N^2 - N - kN$ off-diagonal elements equal to γ .

Hence, $\{\langle i|U(t)|j\rangle : i, j \in [0, N]\}$ contains N elements of α , kN elements of β , and $N^2 - N - kN$ elements of γ . As α , β and γ depend only on (N, k, λ, μ) , two non-isomorphic SRGs with identical family parameters will produce the same time evolution and therefore GI certificate. Hence, two SRGs with same family parameters are not distinguished the CTQW. \square

As expected, the Bose-Mesner approach provides the same conclusion as previously shown by Gamble et al. [5].

6.4.2 SV-phase addition

The first possible option for phase addition includes adding a local phase factor at a selected vertex. We shall call this SV (selected vertex) phase addition. The Hamiltonian in this case is simply the adjacency matrix with phase added along the diagonal at a *single* vertex v_r (the reference vertex); this can be expressed as $H = A + \theta F_0$, where F_0 is the first element in the dual Bose-Mesner basis (we have dropped the superscript (r) for clarity). The resulting time evolution operator is thus $U_{SV}(t, \theta) = e^{-i(A+\theta F_0)t}$. Note that it follows that if the two graphs to be tested are vertex transitive, comparing the transition amplitudes at the vertices is sufficient to measure the strength of the SV-phase added CTQW.

Considering pairs of SRGs with same family parameters, the Shrikhande graph and the (4,4)-lattice graph provide the non-isomorphic pair of $\text{SRG}(N, k, \lambda, \mu)$ with minimum N . For these graphs, the family parameters are given by $N = 16, k = 6, \lambda = \mu = 2$, and both these graphs are vertex transitive. Therefore, the transition amplitudes will provide an

unbiased graph certificate. The effect of SV-phase added CTQW on this pair is described in Lem. 6.12.

Lemma 6.12. *The ordered transition amplitude lists of the SV-phase added CTQW are identical for two non-isomorphic SRGs with parameters $(16,6,2,2)$.*

Proof. Recall the Hamiltonian for SV-phase added CTQW is given by $A + \theta F_0$. Consider the terms in the expansion of the SV time evolution operator:

$$U_{SV} = \sum_{n=0}^{\infty} \frac{1}{n!} (-it)^n (A + \theta F_0)^n = \left[I - it(\mathcal{A}_1 + \theta F_0) + \frac{(-it)^2}{2!} (\mathcal{A}_1 + \theta F_0)^2 + \dots \right]. \quad (6.22)$$

Let us use mathematical induction on n to prove the following result,

$$(A + \theta F_0)^n = \sum_{j=0}^2 \alpha_j^{(n)} \mathcal{A}_j + \sum_{j=0}^2 \beta_j^{(n)} F_j + \sum_{j=0}^2 \sum_{k=0}^2 \gamma_{j,k}^{(n)} F_j \mathcal{A}_k + \chi^{(n)} \mathcal{A}_1 F_0 + \epsilon^{(n)} \mathcal{A}_2 F_0 + \zeta^{(n)} \mathcal{A}_1 F_0 \mathcal{A}_1 + \eta^{(n)} \mathcal{A}_2 F_0 \mathcal{A}_1, \quad (6.23)$$

where $\alpha_j^{(n)}$, $\beta_j^{(n)}$, $\gamma_{j,k}^{(n)}$, $\chi^{(n)}$, $\epsilon^{(n)}$ and $\zeta^{(n)}$ are coefficients depending only upon the family parameters of the SRG. By requiring $\alpha_1^{(1)} = 1$, $\beta_0^{(1)} = \theta$, and $\gamma_{j,k}^{(1)} = \chi^{(1)} = \epsilon^{(1)} = \zeta^{(1)} = \eta^{(1)} = 0$, it can be seen that for the case $n = 1$, Eq. 6.23 reduces to

$$A + \theta F_0 = \mathcal{A}_1 + \theta F_0. \quad (6.24)$$

Thus Eq. 6.23 holds true for $n=1$.

Now, assuming it is true for n , consider the case $n + 1$. The terms of the expansion of $(A + \theta F_0)^{n+1}$ are derived by right-multiplying each term in Eq. 6.23 by \mathcal{A}_1 and F_0 respectively. Since $\{\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2\}$ form the basis of the Bose-Mesner algebra of SRGs, we can make use of the multiplication closure relation $\mathcal{A}_i \mathcal{A}_j = \sum_{\ell=0}^2 p_{ij}^{(\ell)} \mathcal{A}_\ell$. Thus, right-multiplying the first term by \mathcal{A}_1 , we get

$$\left(\sum_{j=0}^2 \alpha_j^{(n)} \mathcal{A}_j \right) \mathcal{A}_1 = \sum_{j=0}^2 \alpha_j^{(n+1)} \mathcal{A}_j, \quad (6.25)$$

where $\alpha_j^{(n+1)}$ denotes a new coefficient signifying its continued contribution to the first term in the expansion Eq. 6.23. Similarly, right-multiplying the second and third terms of

the expansion by \mathcal{A}_1 , we find that they contribute to the $F_j \mathcal{A}_k$ double summation term:

$$\left(\sum_{j=0}^2 \beta_j^{(n)} F_j \right) \mathcal{A}_1 = \sum_{j=0}^2 \gamma_{a_{j,1}}^{(n+1)} F_j \mathcal{A}_1, \quad (6.26a)$$

$$\left(\sum_{j=0}^2 \sum_{k=0}^2 \gamma_j^{(n)} F_j \mathcal{A}_k \right) \mathcal{A}_1 = \sum_{j=0}^2 \sum_{k=0}^2 \gamma_{b_{j,k}}^{(n+1)} F_j \mathcal{A}_k. \quad (6.26b)$$

where $\gamma_{a_{j,k}}^{(n+1)} + \gamma_{b_{j,k}}^{(n+1)} + \gamma_{c_{j,k}}^{(n+1)} = \gamma_{j,k}^{(n+1)}$ ($\gamma_{c_{j,k}}$, the final contribution to $\gamma_{j,k}^{(n+1)}$, will be defined later). Similarly, after right-multiplication by \mathcal{A}_1 the fourth and fifth terms can now be written in terms of the sixth and seventh terms of Eq. 6.23, respectively:

$$\left(\chi^{(n)} \mathcal{A}_1 F_0 \right) \mathcal{A}_1 = \zeta^{(n+1)} \mathcal{A}_1 F_0 \mathcal{A}_1, \quad (6.27a)$$

$$\left(\epsilon^{(n)} \mathcal{A}_2 F_0 \right) \mathcal{A}_1 = \eta^{(n+1)} \mathcal{A}_2 F_0 \mathcal{A}_1. \quad (6.27b)$$

For the case SRG(16,6,2,2), it can be shown that

$$\mathcal{A}_1 F_0 \mathcal{A}_1^2 = 2 (F_1 \mathcal{A}_1 + F_1 \mathcal{A}_2 + 2 \mathcal{A}_1 F_0 + F_1), \quad (6.28a)$$

$$\mathcal{A}_2 F_0 \mathcal{A}_1^2 = 2 (F_2 \mathcal{A}_1 + F_2 \mathcal{A}_2 + 2 \mathcal{A}_2 F_0 + F_2). \quad (6.28b)$$

Thus, multiplying the sixth term of Eq. 6.23 by \mathcal{A}_1 in the case of SRG(16,6,2,2), and expanding in terms of the basis elements,

$$(\zeta^n \mathcal{A}_1 F_0 \mathcal{A}_1) \mathcal{A}_1 = \beta_{a_1}^{(n+1)} F_1 + \gamma_{c_{1,1}}^{(n+1)} F_1 \mathcal{A}_1 + \gamma_{c_{1,2}}^{(n+1)} F_1 \mathcal{A}_2 + 2 \chi_a^{(n+1)} \mathcal{A}_1 F_0, \quad (6.29a)$$

$$(\zeta^n \mathcal{A}_2 F_0 \mathcal{A}_1) \mathcal{A}_1 = \beta_{a_2}^{(n+1)} F_2 + \gamma_{c_{2,1}}^{(n+1)} F_2 \mathcal{A}_1 + \gamma_{c_{2,2}}^{(n+1)} F_2 \mathcal{A}_2 + 2 \epsilon_a^{(n+1)} \mathcal{A}_2 F_0 \quad (6.29b)$$

We now return to the general case of an arbitrary SRG, and repeat the previous procedure — this time right-multiplying the terms in Eq. 6.23 by \mathcal{F}_1 :

$$\left(\sum_{j=0}^2 \alpha_j^{(n)} \mathcal{A}_j \right) F_0 = \beta_{a_0}^{(n+1)} F_0 + \chi_b^{(n+1)} \mathcal{A}_1 F_0 + \epsilon_b^{(n+1)} \mathcal{A}_2 F_0. \quad (6.30)$$

Now we make use of the property $F_j F_k = \delta_{jk} F_j$ of the dual Bose-Mesner algebra at the reference vertex v_r , and establish the following:

$$\left(\sum_{j=0}^2 \beta_j^{(n)} F_j \right) F_0 = \beta_{b_0}^{(n+1)} F_0. \quad (6.31)$$

Considering the three-fold products of the form $F_j \mathcal{A}_k F_l$, we can now make use of the fact

that the uv th entry of this three-fold product is expressible as

$$(F_j \mathcal{A}_k F_l)_{uv} = \begin{cases} 1, & \text{if } v_u \in G_j(x), v_v \in G_l(x), \text{ and } v_u \in G_k(v) \\ 0, & \text{otherwise,} \end{cases} \quad (6.32)$$

(where $G_j(x)$ denotes the set of vertices in the graph located at a distance j from the vertex x [136, 137]) to deduce the following:

$$F_0 \mathcal{A}_1 F_0 = F_2 \mathcal{A}_1 F_0 = F_0 \mathcal{A}_2 F_0 = F_1 \mathcal{A}_2 F_0 = O. \quad (6.33)$$

Further, since

$$(\mathcal{A}_j F_k)_{uv} = \begin{cases} 1, & \text{dist}(v_u, v_v) = j \text{ and } \text{dist}(v_v, v_x) = k \\ 0, & \text{otherwise} \end{cases} \quad (6.34)$$

we deduce that

$$F_1 \mathcal{A}_1 F_0 = \mathcal{A}_1 F_0, \quad (6.35a)$$

$$F_2 \mathcal{A}_2 F_0 = \mathcal{A}_2 F_0. \quad (6.35b)$$

Recall that $\mathcal{A}_0 = I$; it follows that

$$F_j \mathcal{A}_0 F_0 = F_j F_0 = \begin{cases} F_0, & j = 0 \\ O, & j = 1, 2. \end{cases} \quad (6.36)$$

Using this fact with Eq. 6.33 and Eq. 6.35, we establish the following:

$$\left(\sum_{j=0}^2 \sum_{k=0}^2 \gamma_{j,k}^{(n)} F_j \mathcal{A}_k \right) F_0 = \beta_{c_0}^{(n+1)} F_0 + \chi_c^{(n+1)} \mathcal{A}_1 F_0 + \epsilon_c^{(n+1)} \mathcal{A}_2 F_0. \quad (6.37)$$

Since $F_j F_k = \delta_{jk} F_j$, the terms $\mathcal{A}_1 F_0$ and $\mathcal{A}_2 F_0$ undergo only a change of coefficients when right-multiplied by F_0 . Finally, in the specific case of $\text{SRG}(16,6,2,2)$, it can be shown that

$$\mathcal{A}_1 F_0 \mathcal{A}_1 F_0 = \mathcal{A}_2 F_0 \mathcal{A}_1 F_0 = O. \quad (6.38)$$

Now, using Eq. 6.25–6.27, 6.29–6.31, and 6.37–6.38, we establish that

$$(A + \theta F_0)^{n+1} = \sum_{j=0}^2 \alpha_j^{(n+1)} \mathcal{A}_j + \sum_{j=0}^2 \beta_j^{(n+1)} F_j + \sum_{j=0}^2 \sum_{k=0}^2 \gamma_{j,k}^{(n+1)} F_j \mathcal{A}_k$$

$$+ \chi^{(n+1)} \mathcal{A}_1 F_0 + \epsilon^{(n+1)} \mathcal{A}_2 F_0 + \zeta^{(n+1)} \mathcal{A}_1 F_0 \mathcal{A}_1 + \eta^{(n+1)} \mathcal{A}_2 F_0 \mathcal{A}_1. \quad (6.39)$$

Therefore, by the principle of mathematical induction, we establish Eq. 6.23. Next, we consider the term $\mathcal{A}_2 F_0 \mathcal{A}_1$; this three-fold product is expressible as a combination of singletons, two-fold products and the three-fold product $\mathcal{A}_1 F_0 \mathcal{A}_1$:

$$\mathcal{A}_2 F_0 \mathcal{A}_1 = F_1 + \mathcal{A}_1 F_1 + \mathcal{A}_2 F_2 - F_0 \mathcal{A}_1 - \mathcal{A}_1 F_0 \mathcal{A}_1. \quad (6.40)$$

Observe that the singletons can be expressed as two-fold products, as $\mathcal{A}_0 = F_0 + F_1 + F_2 = I$. Thus, it follows that $(A + \theta F_0)^n$ is expressible as a linear combination of two-fold products and the single three-fold product $\mathcal{A}_1 F_0 \mathcal{A}_1$; and therefore we express $(-it)^n (A + \theta F_0)^n$ as follows;

$$(-it)^n (A + \theta F_0)^n = \sum_{j=0}^2 \sum_{k=0}^2 \left(\alpha_{j,k}^{(n)} \mathcal{A}_j F_k + \beta_{j,k}^{(n)} F_j \mathcal{A}_k + \gamma^{(n)} \mathcal{A}_1 F_0 \mathcal{A}_1 \right). \quad (6.41)$$

Hence, summing over all n , the SV time evolution operator is expressible as

$$U_{SV}(t, \theta) = - \sum_{j=0}^2 \sum_{k=0}^2 \left(\alpha_{j,k} \mathcal{A}_j F_k + \beta_{j,k} F_j \mathcal{A}_k + \gamma \mathcal{A}_1 F_0 \mathcal{A}_1 \right). \quad (6.42)$$

where $\alpha_{j,k}$, $\beta_{j,k}$ and γ depend on the family parameters of the SRG, and implicitly on time t and phase θ .

Now we calculate the contribution of each term in this summation. Since each matrix is comprised solely of ones and zeros, an entry of the matrix U is a summation of $\alpha_{j,k}$, $\beta_{j,k}$, and γ . The contribution of each element can be determined both combinatorially and computationally. Tab. 6.1 provides a list of all possible elements in the U_{SV} expansion of graphs in the SRG(16,6,2,2) family. As expected, since $\alpha_{j,k}$, $\beta_{j,k}$ and γ depend *only* on (N, k, λ, μ) , each element in the table occurs an equal number of times in the summation for both U_{SV} and \dot{U}_{SV} . Thus it follows that the two SRGs with parameters (16,6,2,2) are not distinguishable by the SV-phase added CTQW. \square

6.4.3 AV-phase addition

In this section, a phase θ will be added to all vertices adjacent to a selected *reference vertex*; let us refer to this method as the adjacent vertices (AV) phase added CTQW. Note that this is the continuous analogue of the perturbed DTQW GI algorithm by Douglas and

Element	Occurrences in U_{SV}	Occurrences in \hat{U}_{SV}
$\alpha_{00} + \beta_{00}$	1	1
$\alpha_{11} + \beta_{01}$	6	6
$\alpha_{22} + \beta_{02}$	9	9
$\alpha_{10} + \beta_{11}$	6	6
$\alpha_{12} + \beta_{11}$	18	18
$\alpha_{22} + \beta_{12}$	36	36
$\alpha_{02} + \beta_{20}$	9	9
$\alpha_{11} + \beta_{21}$	18	18
$\alpha_{12} + \beta_{21}$	36	36
$\alpha_{20} + \beta_{22}$	9	9
$\alpha_{21} + \beta_{22}$	36	36
$\alpha_{22} + \beta_{22}$	36	36
$\alpha_{01} + \beta_{10} + \gamma$	6	6
$\alpha_{11} + \beta_{11} + \gamma$	12	12
$\alpha_{21} + \beta_{12} + \gamma$	18	18

Table 6.1: Number of occurrences of all possible coefficients in the expansion of the selected vertex (SV) CTQW unitary operator Eq. 6.42 of two non-isomorphic graphs G and \hat{G} in the SRG(16,6,2,2) family.

Wang [15], which successfully distinguished all non-isomorphic members of the same SRG family they tested.

Mathematically, the AV-phase added CTQW can be achieved by replacing the ones in the v_r th row and column of the Hamiltonian by the phase θ . Consequently, the Hamiltonian for the AV-phase added CTQW is expressible as

$$(H_{AV})_{uv} = \begin{cases} \theta A_{uv} & \text{if } u = v_r \text{ or } v = v_r \\ A_{uv} & \text{otherwise.} \end{cases} \quad (6.43)$$

An alternative form of expressing the Hamiltonian is through the use of the elements of the Bose-Mesner algebra and its dual at the reference vertex as discussed in the previous section. This allows us to express the AV-phase added Hamiltonian via the expansion

$$H_{AV} = \mathcal{A}_1 + \theta(F_0\mathcal{A}_1 + \mathcal{A}_1F_0). \quad (6.44)$$

Since the two graphs are vertex transitive, we again take the transition amplitudes over all vertices as the graph certificate. We then prove that the AV-phase added CTQW is not as powerful as its discrete time equivalent, the phase-added DTQW used by Douglas and Wang [15]. Lem. 6.13 gives the desired result.

Lemma 6.13. *The ordered transition amplitude lists of the AV-phase added CTQW are identical for two non-isomorphic SRGs with parameters (16,6,2,2).*

Proof. Recall that the AV-Hamiltonian is expressible as

$$U_{AV}(t, \theta) = e^{-i[\mathcal{A}_1 + \theta(F_0\mathcal{A}_1 + \mathcal{A}_1F_0)]t}, \quad (6.45)$$

therefore,

$$U_{AV}(t, \theta) = I - it[\mathcal{A}_1 + \theta(F_0\mathcal{A}_1 + \mathcal{A}_1F_0)] + \frac{(-it)^2}{2!} [\mathcal{A}_1 + \theta(F_0\mathcal{A}_1 + \mathcal{A}_1F_0)]^2 + \dots \quad (6.46)$$

Similar to the SV-phase added CTQW, all the higher powers cancel out, reducing to a linear combination of singletons, two-fold products and a single three-fold product again;

$$U_{AV}(t, \theta) = \sum_{j,k=0}^2 \left(\hat{\alpha}_{j,k}\mathcal{A}_iF_j + \hat{\beta}_{j,k}F_i\mathcal{A}_j + \hat{\gamma}\mathcal{A}_1F_0\mathcal{A}_1 \right). \quad (6.47)$$

where, as before, $\hat{\alpha}_{j,k}$, $\hat{\beta}_{j,k}$ and $\hat{\gamma}$ depend on the family parameters of the SRG, and implicitly on time t and phase θ . Therefore, following the same argument as detailed in the proof of Lem. 6.12, the set of transition amplitudes and thus the graph certificates for two non-isomorphic graphs within SRG(16,6,2,2) are identical. \square

6.5 Multi-particle CTQW with interactions

Whilst we have proved that a single-particle perturbed CTQW cannot distinguish SRGs in the same family (augmenting the existing result for non-interacting multi-particle CTQWs [5]), it is currently unknown whether multi-particle CTQWs with interactions afford the same distinguishing power as DTQWs. Although not analyzed analytically nor numerically here, it is worth noting that, along with continuous-time quantum walking simulations, the software package *pyCTQW* detailed in chapter 4 also provides CTQW-based methods for determining whether two graphs G and H are isomorphic. The GI implementation used in *pyCTQW* is an adaptation of the two particle interacting DTQW algorithm of Berry et al. [16] for continuous-time quantum walks, and extended to allow up to 3 interacting quantum particles. Coupled with the parallel capability of *pyCTQW*, this functionality is designed to allow a quick and efficient quantum-walk based method of determining whether two graphs are indeed isomorphic. It should be noted that this algorithm cannot distinguish arbitrary graphs when using only a single walker [133] or two interacting walkers [5],

however the question of whether the three particle interacting CTQW can distinguish arbitrary strongly regular graphs remains unanswered.

GI certificate and algorithm

The P -particle interacting CTQW GI certificate, used to distinguish non-isomorphic graphs, is constructed by *pyCTQW* as described below:

1. Construct the P -particle CTQW Hamiltonian for the graph, with on-site interaction term $\Gamma_{int}(1)$.
2. (a) A bosonic edge state of the graph,

$$\left| \beta_{j,k}^+ \right\rangle = \frac{1}{\sqrt{2}} \left(\left| v_j^{(1)} \right\rangle \otimes \cdots \otimes \left| v_j^{(P)} \right\rangle + \left| v_k^{(1)} \right\rangle \otimes \cdots \otimes \left| v_k^{(P)} \right\rangle \right), \quad (6.48)$$

is chosen for some $j, k \in \{1, \dots, N\}$ such that $v_j \sim v_k$ (they are adjacent of the graph), and superscript (i) represents the particle number.

- (b) Propagate the P -particle interacting CTQW starting from initial state $|\psi(0)\rangle = \left| \beta_{j,k}^+ \right\rangle$ for time $t = 2N$. Store the resulting particle probabilities over the graph nodes in the form of a list; $P_{j,k} = \left\{ \left| \langle v_\ell | \hat{U}(2N) | \beta_{j,k}^+ \rangle \right|^2 : v_\ell \in V \right\}$.
- (c) Repeat step (2) for all possible bosonic edge states $\left| \beta_{j,k}^+ \right\rangle$.
3. Create a frequency table containing all resulting unique values from $\bigcup_{j,k} P_{j,k}$, along with the frequency at which each value occurs.

By comparing the graph certificates constructed for two arbitrary graphs, *pyCTQW* then attempts to determine whether the two graphs are isomorphic.

6.5.1 GI framework and functions

Similarly to the CTQW simulations, the graph isomorphism class `pyCTQW.MPI.GraphISO` is provided by the `pyCTQW.MPI` Python module, containing a variety of methods and functions that can be used to analyse arbitrary graphs.

Before calling any of the GI functions, a graph isomorphism object must be created. For example, creating a graph isomorphism object utilizing two interacting walkers propagated via the Krylov method:

```
gi = pyCTQW.MPI.GraphISO(p=2,propagator='krylov')
```

Other parameters that may be passed when creating the `GraphISO` object include `freqTol` (*float*), the tolerance to use when constructing the frequency table (the default value is 0.01; decreasing this value may result in rounding/precision errors causing fluctuations in the GI certificate), as well as `compareTol` (*float*, default 1×10^{-9}), the tolerance to use when comparing two GI certificates – two certificates are considered isomorphic if $\max|cert_1 - cert_2| < compareTol$. Also note that if the Chebyshev propagator is used, `GraphISO.setEigSolver()` may be used to set the various eigensolver properties; see Tab. 4.3.

Once initialised, various MPI enabled GI methods are available; these are briefly described in Tab. 6.2, and provide a means of efficiently calculating the GI certificate and testing for isomorphism.

Table 6.2: Summary of parallel GI methods available in `pyCTQW.MPI.GraphISO`. For more details and required arguments, please refer to the online documentation.

<code>GIcert()</code>	Generate the GI certificate of a graph.
<code>AllIsomorphicQ()</code>	Calculates whether each pair of graphs (from a folder containing adjacency matrices) are isomorphic, returning an array R with $R_{ij} = 1$ if graphs i and j are isomorphic, and $R_{ij} = 0$ otherwise.
<code>isomorphicQ()</code>	Returns <code>True</code> if two graphs are isomorphic.

One drawback of the multi-particle interacting CTQW algorithm described here is the large state space required for simulation; a 3-particle walk on a N vertex graph results in the calculation of a $N^3 \times N^3$ matrix exponential.

6.6 Conclusion

It is well known that a standard single-particle DTQW or CTQW cannot distinguish a pair of SRGs with the same family parameters, and furthermore it has been proven by Gamble et al. [5] that a two particle CTQW with no interaction cannot distinguish such a pair. In the case of the single particle DTQW, variations (such as the addition of phases at a reference vertex) lead to an increased distinguishing power, providing single-particle DTQWs with the ability to distinguish the aforementioned SRGs [15]. However, no analogous modifications had been previously explored in the case of the single particle CTQW; whilst considered unlikely, it was hitherto unknown for sure whether the continuous counterpart of a single-particle DTQW could be modified to produce a similar increase in

distinguishing ability.

In this study, it was proven that various phase-added CTQWs are unable to distinguish non-isomorphic SRGs with family parameters (16,6,2,2), the smallest such set of family parameters with a distinct disadvantage compared to the equivalent DTQW algorithm. Note that although the two-fold and singleton Bose-Mesner products derived in Thm. 6.4.2 are specific to SRG(16,6,2,2), equivalent relations continue to hold in the case of general SRG parameters – allowing the Hamiltonian to remain written as linear combinations of k -fold products. Consequently, the time evolution will persist in producing identical graph certificates for two non-isomorphic SRGs with same parameters; this inference has been supported by numerical computation.

Instead of the phase additions discussed here, one may consider a general phase addition to the single particle CTQW. The corresponding Hamiltonian of the system is expressible as $A \circ P_1 \circ P_2 \circ \dots \circ P_k$, where A is the adjacency matrix, P_i is the i th phase matrix (produced by adding $\theta_i - 1$ to specified elements of J), and \circ is the Schur product of matrices (defined as $(A \circ B)_{ij} = A_{ij}B_{ij}$). As long as this belongs to the algebra generated by the basis elements of the Bose-Mesner algebra and the dual Bose-Mesner algebra at the reference vertex v_r , the general phase-modified CTQW will be of similar form to the phase modifications described in this paper. This particular algebra of the spanning set $\mathcal{F}^{(r)} = \{\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_{\mathcal{D}}, F_0^{(r)}, F_1^{(r)}, \dots, F_{\mathcal{D}}^{(r)}\}$ was first studied by Terwilliger [136], and is therefore referred to as the Terwilliger algebra (denoted $\mathbb{T}^{(r)} = \text{span}(\mathcal{F}^{(r)})$) of the graph with respect to v_r . It should be noted that in most studies the dual Bose-Mesner algebra is analyzed in conjunction with the Bose-Mesner algebra, thus leading towards the Terwilliger algebra [132, 134, 135]. Considering the general phase addition, whenever $A \circ P_1 \circ P_2 \circ \dots \circ P_k$ does not belong to the Terwilliger algebra, the exponentiated Hamiltonians of SRGs with same parameters are expected to be distinguishable. Therefore, for a pair of SRGs to be distinguishable, the CTQW has to be phase-modified in a way that $A \circ P_1 \circ P_2 \circ \dots \circ P_k$ does not belong to the Terwilliger algebra.

In order to explain the reduced distinguishing power of the single-particle phase-added CTQW compared to the discrete case, it is worth recalling (Sec. 3.2) that the two quantum walks evolve in different spaces; even confined to a single-particle system, the unitary time-evolution of the CTQW is simply a polynomial of the $N \times N$ adjacency matrix or Laplacian of the graph, whilst the DTQW unitary time-evolution operator is a significantly more complicated expression given by

$$U(t) = (S \cdot C)^t, \quad (6.49)$$

where the shifting operator S and the global coin operator \mathcal{C} both act in the much larger Hilbert space $\mathcal{H} = \mathcal{H}_P \otimes \mathcal{H}_C$ with dimension $\sum_i d_i \times \sum_i d_i$.

Thus, it can be seen that due to the extra degrees of freedom afforded by the coin space, the DTQW evolves in a higher dimensional space compared to the CTQW, allowing it to possess extra distinguishing power. Hence, it follows that a phase addition to the DTQW Hamiltonian is not reducible, as is the case for the CTQW — the complexity contributed to the structure of the DTQW Hamiltonian by the coin operation seemingly greatly affects the time evolution of the discrete time walk.

Considering these results, we conclude that the DTQW is significantly more powerful than CTQWs for distinguishing non-isomorphic graphs, even when taking into account phase defects or modifications. Although the DTQW has its own limitations (as shown in [68] and [102]), it intrinsically has extra distinguishing power compared to the CTQW, due to its time evolution occurring in a larger dimensional Hilbert space.

However, it is worth noting that the increased distinguishing power of the DTQW comes at the cost of a larger statespace, and thus physically implementing such algorithms requires an increase in resources over the CTQW. In the next part of this thesis, we will consider the use of CTQWs in analysing network centrality — an application in which the CTQW produces results on par with the DTQW, but with significantly reduced experimental overhead.

CHAPTER 7

Network centrality

*This chapter is based on the paper by J. A. Izaac, X. Zhan, Z. Bian, K. Wang, J. Li, J. B. Wang, and P. Xue, “Centrality measure based on continuous-time quantum walks and experimental realization”, *Physical Review A* **95**, 032318 (2017)., with minimal modification to match the formatting, notation, and terminology of this thesis.*

In the study of network structure and graph theory, centrality measures are an integral tool, allowing determination and ranking of vertices deemed to be most important. Due to the large number of physical systems that can be modelled as networks, this has seen wide application across multiple disciplinary fields, including internet technology (ranking web sites for search engines [40]), industry (power distribution [138]), business (organisational management [72–74]), biology (grooming networks in macaques [75]), and biochemistry (finding active sites in proteins [76]). In this chapter, we present a detailed introduction to the theory of network centrality, and discuss some well-known random-walk based centrality measures. Furthermore, we review proposed quantum walk-based centrality measures, and consider the continuous-time quantum walks’ aptness in providing a quantum analogue of the classical eigenvector centrality.

7.1 Introduction

At its most basic, a graph centrality measure C satisfies the following properties:

- $C : G(V, E) \rightarrow \mathbb{R}^{|V|}$ is a function or algorithm that accepts a graph as input, and returns a real-valued vector over the set of vertices V .
- Higher values are provided to vertices deemed more ‘important’ or ‘central’ to the graph structure.

However, with this general definition comes several caveats. Firstly, note that no meaning has been attributed to vertices with low centrality values — this is deliberate, as noise grows successively larger for vertices beyond the topmost ranked vertices [139]. As such,

centrality measures convey very little information regarding a majority of vertices; they are solely for determining the most central nodes (ranking *all* the vertices is more the domain of *influence measures* [140]). Secondly, what constitutes ‘importance’ is subjective — it depends on the application or model to be analysed, and how information ‘flows’ throughout the network [71]. For example, information might flow predominantly through paths (a sequence of unique edges and vertices — characteristic of bacterial and viral infections [141]), trails (vertices can be revisited but each edge is only traversed once — the flow of gossip in social networks [142]), and walks (where there is no restrictions on edge and vertex sequences — for example bank note exchange in a population). Moreover, this flow can occur through serial duplication (travelling via one edge at each timestep — gift exchange) or parallel duplication (traversing multiple edges simultaneously — radio broadcasting).

Thus, it is important to apply a centrality measure that models information flow corresponding to the network under study; failure to do so may result in poor results, and even the inability to correctly interpret the results [71]. To deal with this plethora of scenarios, various classical centrality measures have been introduced: degree centrality, eigenvector centrality, betweenness centrality, closeness centrality, and PageRank, among others. Of these, degree, eigenvector and PageRank centrality are what is known as *radial parallel duplication* measures, which measure network flow via walks emanating from or terminating at particular nodes [143].

In this chapter, we begin by discussing the various categories and classical centrality measures in common use in Sec. 7.2. Secondly, in Sec. 7.3, we give an overview of recently proposed quantum walk-based centrality measure, and make the case for introduction of a CTQW-based quantum centrality measure.

7.2 Classical centrality measures

Here, we describe some well-known classical centrality measures, including degree centrality, eigenvector centrality, PageRank, and random walk centrality. It is pertinent to note that, due to the walk-based approach of these aforementioned classical measures, many of these centrality measures can be reformulated as classical random walks — we will make a note of these relationships where applicable.

7.2.1 Degree centrality

The degree centrality measure, calculated via the row-sums of the adjacency matrix,

$$C_j^{(deg)} = \frac{\deg(v_j)}{\sum_k \deg(v_k)} = \frac{\sum_i A_{ij}}{\sum_i \sum_j A_{ij}}, \quad (7.1)$$

is based on walks of length one emanating from each vertex, and is useful in cases when dealing with direct and immediate influence between nodes. Further, it can be seen that the limiting probability distribution of classical random walks (Eq. 3.4) are proportional to the node degree, allowing the degree centrality to be simulated via a Markovian process.

7.2.2 Eigenvector centrality

Eigenvector centrality, on the other hand, is a useful measure when considering long-term ‘indirect’ influence between vertices; if a vertex with low degree is adjacent to a vertex with a high number of connections, the first vertex will likewise have a high eigenvector centrality measure [71]. The simplest centrality measure to be based on the spectral properties of a graph, the eigenvector centrality is defined by $C_j^{(ev)} = \mathbf{v}_j$, where \mathbf{v} is the eigenvector of the adjacency matrix with maximum eigenvalue [144]. Also referred to as the *principal eigenvector*, it is chosen to ensure (via the Perron-Frobenius theorem, Thm. 2.40, assuming that A is irreducible) that the ranking $C_j^{(ev)}$ remains strictly positive.

It has been shown by Bonacich [145] that the eigenvector centrality is proportional to the row-sums of matrix S , $\mathbf{v}_j \propto \sum_i S_{ij}$, where

$$S = A + \frac{1}{\lambda} A^2 + \frac{1}{\lambda^2} A^3 + \dots = \sum_{n=1}^{\infty} \lambda^{1-n} A^n. \quad (7.2)$$

Combining these two equations, we see that

$$\mathbf{v}_j \propto \sum_i \sum_{n=1}^{\infty} \lambda^{1-n} (A^n)_{ij}, \quad (7.3)$$

where λ is the maximum (principal) eigenvalue. As $(A^n)_{ij}$ represents the number of walks of length n between vertices i and j , it can be seen that the eigenvector centrality performs walks of *all* lengths, weighted inversely by length, from each node.

7.2.3 PageRank

Unfortunately, the eigenvector centrality can provide ineffectual results when applied to directed graphs, for a multitude of reasons. For example, vertices not in a strongly connected component will be assigned an eigenvector vertex centrality of 0, as A is no longer irreducible and the Perron-Frobenius theorem is no longer valid. Furthermore, ‘dangling nodes’ (vertices or components with in-degree and zero out-degree) found in directed acyclic subgraphs can result in the eigenvector centrality ‘localising’ or accumulating at the affected vertices [146, 147].

To rectify these issues, a wide range of variations to the eigenvector centrality have been proposed, including PageRank and Katz centrality. Of these, PageRank is arguably the most well known spectral centrality measure — developed as a ranking algorithm for sites on the World Wide Web, it has accumulated significant prestige as the algorithm behind the Google search engine [40]. In this context, vertices represent websites, with directed edges the links between them. Designed from conception to take into account directed networks, the PageRank improves on the eigenvector centrality by modifying the adjacency matrix to ensure stochasticity and irreducibility.

To do so, the eigenvector centrality method is instead applied to the Google matrix G , defined by

$$G = \alpha E + \frac{1}{N}(1 - \alpha)J, \quad 0 \leq \alpha \leq 1, \quad (7.4)$$

where N is the number of vertices in the graph, E is the *patched adjacency matrix*, column-normalised¹ to ensure G is stochastic,

$$E_{ij} = \begin{cases} A_{ij}/\sum_k A_{kj}, & \sum_k A_{kj} \neq 0 \\ 1/N, & \sum_k A_{kj} = 0, \end{cases} \quad (7.5)$$

and J the all one’s matrix. The addition of J is to provide a small ‘random surfer effect’, i.e. a non-zero uniform probability that a walker at a particular vertex can jump to *any* other vertex, even in cases of non-adjacency. Moreover, since J is an irreducible matrix, it has the effect of ensuring that G is irreducible (the corresponding graph is strongly connected) and primitive (since all elements $G_{ij} > 0$). In practice, α is generally chosen to be 0.85, providing a good compromise between information flow via hyperlinks and the random surfer effect.

¹Although E is column-normalised, i.e. each column is normalised by the corresponding vertex out-degree, the PageRank has been shown to be highly correlated with vertex in-degree [148].

Once the Google matrix is calculated, the PageRank centrality measure is then applied by solving the eigenvector equation

$$G\mathbf{x} = \mathbf{x}, \quad (7.6)$$

as, per the Perron-Frobenius theorem, the eigenvector corresponding to the largest eigenvalue ($\lambda = 1$ for PageRank, as G is stochastic) will be strictly positive. Note that this equation is identical to that of a DTRW; thus, the PageRank can be modelled as a DTRW with G taken to be the transition matrix. Nevertheless, when $\alpha < 1$, the PageRank continues to model its centrality measure on walks of all lengths, due to the random surfer effect. To see this explicitly, it can be easily shown that in the case $0 \leq \alpha < 1$,

$$\begin{aligned} G\mathbf{x} &= \mathbf{x} \\ \Rightarrow \alpha E\mathbf{x} + \frac{1}{N}(1-\alpha)J\mathbf{x} &= \mathbf{x} \\ \Rightarrow (I - \alpha E)\mathbf{x} &= \frac{1}{N}(1-\alpha)J\mathbf{x} \end{aligned} \quad (7.7)$$

has the exact solution

$$\begin{aligned} \mathbf{x} &= (I - \alpha E)^{-1} \frac{1}{N}(1-\alpha)J\mathbf{x} = (1-\alpha) \left(\sum_{k=0}^{\infty} \alpha^k E^k \right) \sum_{j=1}^N \frac{\mathbf{e}_j}{N} \\ \therefore \mathbf{x}_i &= \frac{1}{N}(1-\alpha) \sum_{j=1}^N \left(\sum_{k=0}^{\infty} \alpha^k E^k \right)_{ij}, \end{aligned} \quad (7.8)$$

and therefore \mathbf{x} is calculated using walks of k length for all $k \in \mathbb{N}$, weighted by α^k [149].

7.2.4 Random walk centrality

The random walk centrality (RWC), unlike the centralities previously discussed, is not a radial volume based measure (counting the number of walks between each pair of nodes) but rather a radial *length* based measure, quantifying the length of the walks between nodes [143]. Alternatively, this can be interpreted as a measure of the expected time for information to arrive at a particular node; i.e., the effectiveness or speed of communication [71]. It therefore follows that the random walk centrality is not a spectral measure like the eigenvector centrality and PageRank, but rather a form of *closeness centrality*.

The RWC measure, introduced by Noh and Rieger [39] and based on a DTRW, is given

by

$$C_j^{(RWC)} = \frac{\pi_j}{\tau_j}, \quad (7.9)$$

where π is the random walk limiting distribution, and

$$\tau_j = \sum_{n=0}^{\infty} (T_{jj}^n - \pi_j) \quad (7.10)$$

is the characteristic relaxation time of vertex j . Here, T represents the DTRW transition matrix of the graph structure to be analyzed (see Sec. 3.1.1).

7.3 Quantum centrality measures

The above described walk-based centrality measures are classical in nature. However, due to the close relationship between centrality measures and classical walks, in recent years several quantum centrality measures have been proposed, ranging from quantizations of the aforementioned classical measures to wholly new proposals. For example, the Quantum PageRank utilizes the Szegedy quantum walk (a DTQW formulation [80]) to quantize the directed Markov chains encoded by the Google matrix, before taking the long-time average of the walks probability distribution [12, 13, 77]. In essence, providing a quantum analogue of PageRank centrality.

The quantum stochastic walk (QSW) is another approach, which makes use of the Lindblad master equation to introduce environmental decoherence to a CTQW [13, 78, 79]. In practice, this has the effect of creating a continuous-time walk continuum parametrized by ω , with $\omega = 0$ (no dephasing) corresponding to a purely quantum walk (CTQW) on an undirected graph, and $\omega = 1$ (complete dephasing) corresponding to a purely classical walk (CTRW) over a digraph. By restricting the domain to $0 < \omega \ll 1$, quantum dynamics and the resulting quantum speedup are retained, however the walker will eventually converge to the CTRW limiting probability distribution [103]. Similarly to Quantum PageRank, the centrality measure is then given by the time-average of the probability distribution.

Another proposed quantum centrality measure based on the CTQW utilizes the quantum Jensen-Shannon divergence (QJSD) to assign importance to the network nodes [150]; it is shown that this measure is proportional to the degree centrality of the graph, and is independent of the specific CTQW Hamiltonian chosen ($H = A$, $H = L$, or $H = \hat{L}$) and initial walker state $|\psi(0)\rangle$.

Finally, Berry and Wang [11] proposed a novel method, in which the quantum search algorithm is applied to graph structures via DTQW — the resulting frequency of successful search probability was shown to correlate with the (lazy) random walk centrality of Noh and Rieger [39]. Thus, the quantum centrality scheme of Berry and Wang [11] differs from the three previous quantum centrality schemes, as it considers the mean speed of the walker in transmitting information over the network — it is a form of *quantum closeness centrality*.

Unfortunately, when it comes to physically implementing these quantum centrality measures, we run into various issues. Regarding the DTQW and QSW formulations, due to the use of either a coin state (DTQW) or an environment (QSW), the size of the statespace must be significantly increased, taking us beyond the current experimental capability to simulate quantum graph centrality of even simple graph structures. For example, for a graph of N vertices, the Szegedy DTQW formulation used in the Quantum PageRank scheme requires a statespace of size N^2 . Physical implementation of the QJSD quantum centrality measure, on the other hand, is hindered by the lack of an existing quantum algorithm that enables the measure to be performed experimentally. As such, the ability to physically realise these quantum centrality measures is currently beyond our reach. Furthermore, only two of these proposed centrality measures permit the use of directed graphs — namely, the QSW (through the use of decoherence) and the Quantum PageRank (as the Szegedy walk ‘encodes’ information regarding directed edges into the DTQW local coin operations). The QJSD approach of Rossi et al. [150] and the DTQW approach of Berry and Wang [11] are both undefined on directed graphs.

To avoid the aforementioned shortfalls, we would like to design a quantum walk-based centrality measure that:

- (a) is physically realisable with current experimental techniques (i.e. the measure should possess a significantly smaller statespace than the QSW and Quantum PageRank, allowing for a reduction in resources);
- (b) is purely quantum in nature, unlike the QSW which reaches a classical stochastic equilibrium;
- (c) measures long-term indirect influence between vertices, to avoid capturing information that is trivially revealed by calculating the network degree like the QJSD; and
- (d) can be extended to directed graphs.

The CTQW appears uniquely placed to satisfy the first two of these goals — a CTQW on a graph containing N vertices requires a statespace of size N , and its time-evolution

as per the Schrödinger equation is purely quantum. Regarding point (c), recall Eq. 7.3 of the eigenvector centrality. It should be noted that this result allows us to draw interesting parallels with the CTQW; consider the CTQW time-evolution operator for infinitesimal time dt :

$$U(t) = e^{-iHdt} = \sum_{n=0}^{\infty} \frac{1}{n!} (-idt)^n H^n. \quad (7.11)$$

That is, like the eigenvector centrality, the CTQW performs walks of all lengths at each infinitesimal time-step dt , weighted inversely by walk length. Thus, at the very least, the CTQW may provide the means for an eigenvector-like quantum centrality measure. Finally, the well-established theory of PT-invariance provides an avenue for expanding the CTQW to directed (albeit not arbitrary) graphs.

In the following chapters, we will detail a CTQW-based centrality measure, building on the foundation of classical radial centrality measures such as eigenvector centrality, whilst allowing us to take advantage of the quantum speedup afforded over the CTRW. Utilizing a significantly smaller statespace than the QSW and DTQW, we also present the experimental results of a proof of concept physical implementation of this quantum centrality measure using quantum optics — the first successful physical demonstration of a quantum centrality algorithm. Finally, we will introduce our pseudo-Hermitian CTQW framework, which allows our CTQW-based centrality measure to be extended to select directed graphs.

CHAPTER 8

CTQW centrality on undirected graphs and experimental implementation

*This chapter is based on the paper by J. A. Izaac, X. Zhan, Z. Bian, K. Wang, J. Li, J. B. Wang, and P. Xue, “Centrality measure based on continuous-time quantum walks and experimental realization”, Physical Review A **95**, 032318 (2017). with minimal modification to match the formatting, notation, and terminology of this thesis.*

Network centrality has important implications well beyond its role in physical and information transport analysis; as such, various quantum walk-based algorithms have been proposed for measuring network vertex centrality. In this chapter, we propose a continuous-time quantum walk algorithm for determining vertex centrality, allowing us to preserve the full quantum behaviour of the walker, whilst limiting the dimension of the Hilbert space to N . Furthermore, we show that it generalizes to arbitrary graphs via a statistical analysis of randomly generated scale-free and Erdős-Rényi networks. Finally, as a proof of concept, the algorithm is detailed on a 4-vertex star graph and physically implemented via linear optics, using spatial and polarization degrees of freedoms of a single photon. We therefore report the first successful physical demonstration of a quantum centrality algorithm.

8.1 Introduction

The quantum walk framework, in providing a method of modelling network structures that doubles as a universal system of quantum computation, is uniquely placed in the quest to find quantum analogues of classical network algorithms. As such, one potential application of the quantum walk is in providing an efficient quantum algorithm for vertex centrality ranking in network analysis. Previous studies have proposed algorithms built on the standard discrete-time quantum walk [11], the Szegedy discrete-time quantum walk [12, 13, 77], or the continuous-time quantum stochastic walk (QSW) [13, 78, 79]. However, whilst comparing well to classical centrality measures, these have the distinct disadvantage

of requiring expanded Hilbert spaces (up to N^2 dimensions for a graph of N vertices), or in the case of the QSW, muting the quantum behaviour due to decoherence.

In this study, we propose an alternative quantum walk centrality algorithm based on the CTQW, allowing us to preserve the full quantum behaviour of the walker, whilst limiting the dimension of the Hilbert space to N . Furthermore, we have experimentally implemented this algorithm in the case of the 4-vertex star graph. As far as we are aware, this is the first quantum centrality measure to be physically implemented.

This chapter is structured as follows. We introduce our CTQW-based quantum centrality scheme in Sec. 8.2, before presenting a thorough statistical analysis using ensembles of randomly generated graphs in Sec. 8.3, highlighting the suitability of the quantum centrality scheme for general graphs. Then, we discuss our experimental implementation via linear optics in Sec. 8.4, before finally presenting our conclusions in Sec. 8.5.

8.2 CTQW-based centrality measure

Similarly to the Quantum PageRank and the QSW centrality measures, as the time-evolution of the CTQW is determined by the Hamiltonian — and thus the underlying network structure — one method for extracting the centrality information is to simply start the walker in an equal superposition of all vertex states, $|\psi(0)\rangle = \frac{1}{\sqrt{N}} \sum_j |j\rangle$ (so as not to bias any one particular vertex), and compare the time-average probability of locating the walker at each vertex¹.

Now, convention allows for two choices for the Hamiltonian (see Sec. 3.2.2) — we may choose either the adjacency matrix A or the graph Laplacian L (given by $L_{ij} = \delta_{ij} \sum_k A_{ik} - A_{ij}$, a discrete approximation to the continuous-space Laplacian). However, the construction of the undirected graph Laplacian ensures that the equal superposition state is always an eigenvector (Thm. 2.17), resulting in a stationary time-evolution:

$$U |\psi(0)\rangle = e^{-iLt} \left(\frac{1}{N} \sum_j |j\rangle \right) = \frac{1}{N} \sum_j |j\rangle \quad \forall t. \quad (8.1)$$

As such, the Laplacian is ill-suited for such a CTQW centrality measure on undirected graphs, as it will be unable to distinguish vertices more central to the network structure.

¹Note that this is only one of many possible choices of an unbiased initial state; while we must ensure that the walker is initialised in a state of uniform probability over the entire vertex set, there is no restriction on the relative phases between vertex states. Here, for convenience, we choose the initial state where there is no relative phase difference between vertices.

This is not the case for the adjacency matrix; thus, for the remainder of this work, we will set $H = A$.

To briefly summarise, the proposed CTQW centrality scheme works as follows:

1. Prepare the quantum walker in an initial equal superposition over all vertex states: $|\psi(0)\rangle = \frac{1}{N} \sum_j |j\rangle$.
2. Propagate the walker for time $t \gg 0$: $|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle$, where $H = A$ is the graph adjacency matrix.
3. Calculate the time-average probability distribution of finding the walker at each vertex:

$$C_j^{(CTQW)} = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t |\langle j | \psi(t) \rangle|^2 dt. \quad (8.2)$$

To fully ascertain the reliability of the proposed CTQW centrality measure, we will consider both a simple example (allowing us to qualitatively assess the measures performance), as well as a rigorous statistical analysis comparing the CTQW measure to the PageRank over an ensemble of randomly generated graphs. Freeman [151], in his discussion of the canonical formulations of centrality measures, noted that the degree, closeness and betweenness centralities all attain their highest values for the central node of the star graph; Borgatti and Everett [143], in reviewing Freeman's work, suggested that this may serve as a defining characteristic of a 'proper' centrality measure. Thus, let us consider a 4-vertex star graph as an example of the proposed CTQW centrality measure.

For the 4-vertex star graph shown in Fig. 8.1(a), the adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad (8.3)$$

with the first vertex (vertex 0) the central node. In this case, the time-evolution operator is given by

$$U(t) = \frac{1}{3} \begin{bmatrix} 3c(t) & s(t) & s(t) & s(t) \\ s(t) & c(t) + 2 & c(t) - 1 & c(t) - 1 \\ s(t) & c(t) - 1 & c(t) + 2 & c(t) - 1 \\ s(t) & c(t) - 1 & c(t) - 1 & c(t) + 1 \end{bmatrix}, \quad (8.4)$$

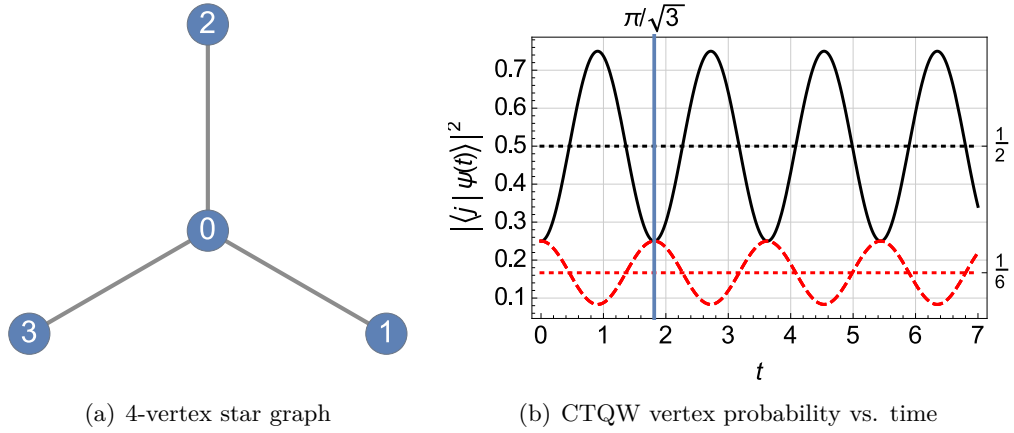


Figure 8.1: CTQW probability at vertex 0 (black, solid) and vertex 1,2,3 (red, dashed) on a 4-vertex star graph. The initial state is a equal superposition of all vertex states. The dotted lines show the respective long-time averaged probability of the respective vertices, with the blue vertical line denoting one period ($T = \pi/\sqrt{3}$).

where $c(t) = \cos(\sqrt{3}t)$ and $s(t) = -i\sqrt{3}\sin(\sqrt{3}t)$. Using this operator to propagate from an initial equal superposition of vertex states $|\psi(0)\rangle = \frac{1}{4}\sum_j |j\rangle$, the probability of locating the walker on vertex j at time t is

$$|\langle j|\psi(t)\rangle|^2 = |\langle j|U(t)|\psi(0)\rangle|^2 = \left[\frac{1}{2} - \frac{1}{4}c(2t)\right] \delta_{j0} + \left[\frac{1}{6} + \frac{1}{12}c(2t)\right] \sum_{j'=1}^3 \delta_{jj'} \quad (8.5)$$

(Fig. 8.1(b)). Noting that this probability distribution is periodic with period $T = \pi/\sqrt{3}$, the CTQW centrality measure becomes

$$C_j^{(CTQW)} = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t |\langle j|\psi(t)\rangle|^2 dt = \frac{1}{T} \int_0^T |\langle j|\psi(t)\rangle|^2 dt, \quad (8.6)$$

yielding values of $1/2$ for $j = 0$, and $1/6$ for $j = 1, 2, 3$. This fits well with what would be expected intuitively — the central vertex (vertex 0) has the highest time-averaged probability, indicating a high centrality measure, whilst the remaining vertices (1,2,3) are equivalent and have an equal and lower ranking. The proposed CTQW centrality measure therefore satisfies one of the defining properties of centrality measures; however, a detailed statistical analysis is required to properly assess its behaviour on general graphs.

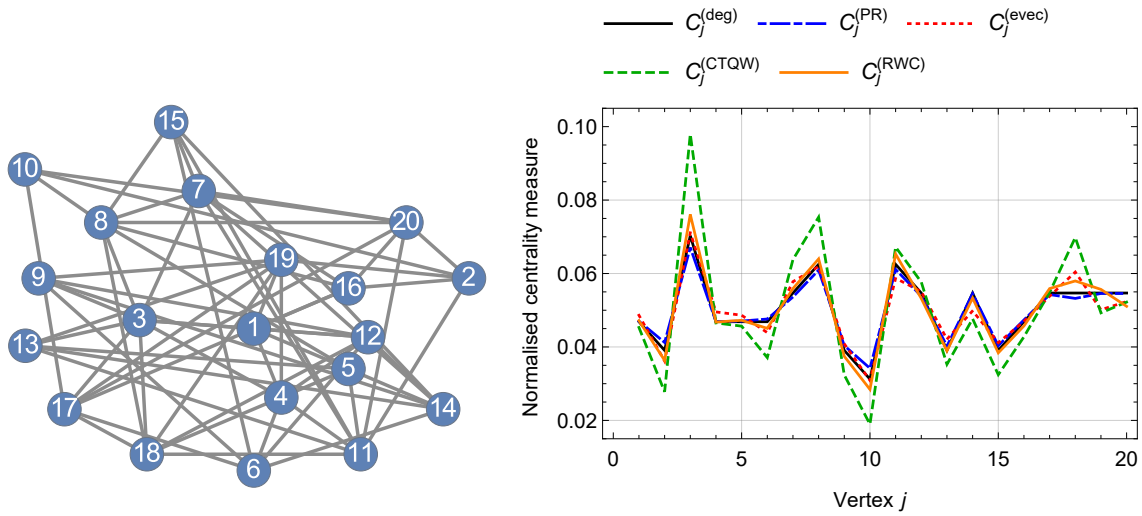


Figure 8.2: **Left:** Randomly generated Erdős-Rényi graph $G(20, 0.3)$ **Right:** Normalised vertex centrality values for vertex j on $G(20, 0.3)$. Measures shown are the degree centrality, PageRank, eigenvector centrality, CTQW centrality, and RWC centrality.

8.3 Statistical analysis

To investigate the reliability of this newly proposed quantum centrality algorithm, it is pertinent to compare its ranking results to classical algorithms on large random graphs. To do so, we consider two classes of random graphs — Erdős-Rényi networks, and scale-free networks.

8.3.1 Correlation to classical measures

Firstly, let's investigate correlation between the CTQW centrality measure and classical measures. As centrality measures only provide useful information for the top 5 or so valued vertices (with noise growing successively larger for lower ranked nodes [139]), we will consider a randomly generated 20-vertex Erdős-Rényi graph $G(20, 0.3)$, and use this as the basis of our correlation test. The graph generated and its respective vertex centrality values (calculated using degree centrality, PageRank, eigenvector centrality, CTQW centrality, and RWC centrality) are shown in Fig. 8.2. Qualitatively, it can be seen that all centrality measures strongly agree on the top-ranked vertices, with slight variations for the lower ranked vertices, as is expected.

Another method of quantifying the correlation between the various centrality methods is to calculate their rank correlation coefficients. One such metric is Kendall's rank correlation

coefficient, commonly referred to as Kendall's τ coefficient [152]. To get a rudimentary understanding of what the Kendall τ coefficient is measuring, we will take a brief foray into the realm of rank correlation coefficients, starting with the centrality measure inner product.

Definition 8.1 (Centrality measure inner product). The centrality inner product of the centrality measures $C^{(r)}$ and $C^{(s)}$ applied to graph vertex set V is defined by

$$\langle C^{(r)}, C^{(s)} \rangle = \sum_i \sum_{\substack{j \\ i < j}} \text{sgn}(C_i^{(r)} - C_j^{(r)}) \text{sgn}(C_i^{(s)} - C_j^{(s)}). \quad (8.7)$$

This has the property $\langle C^{(r)}, \alpha C^{(s)} \rangle = \langle \alpha C^{(r)}, C^{(s)} \rangle = \text{sgn}(\alpha) \langle C^{(r)}, C^{(s)} \rangle$. The norm is given by

$$\|C^{(r)}\| = \sqrt{\langle C^{(r)}, C^{(r)} \rangle},$$

with maximum value $\|C^{(r)}\| = \sqrt{N(N-1)/2}$ when all values are distinct, and $\|C^{(r)}\| = 0$ if and only if $C^{(r)}$ is constant. The centrality measure inner product also satisfies a Cauchy-Schwartz like inequality, $|\langle C^{(r)}, C^{(s)} \rangle| \leq \|C^{(r)}\| \|C^{(s)}\|$.

The Kendall's τ correlation coefficient is then defined as follows:

Definition 8.2 (Kendall's τ correlation coefficient). Let $C^{(r)} \in \mathbb{R}^N$ and $C^{(s)} \in \mathbb{R}^N$ correspond to centrality measures $C^{(r)}$ and $C^{(s)}$ applied to graph vertex set V . The Kendall's τ correlation coefficient is given by

$$\tau(C^{(r)}, C^{(s)}) = \frac{\langle C^{(r)}, C^{(s)} \rangle}{\|C^{(r)}\| \|C^{(s)}\|}. \quad (8.8)$$

As a result, $-1 \leq \tau \leq 1$, where $\tau = 1$ denotes perfect agreements between the ranked lists, $\tau = 0$ denotes no correlation, and $\tau = -1$ denotes perfect anticorrelation (i.e. one list is the reverse of the other). As a general rule of thumb, $\tau \geq 0.8$ is referred to as a very strong rank correlation, $0.6 \leq \tau < 0.8$ a strong rank correlation, $0.4 \leq \tau < 0.6$ a moderate rank correlation, $0.2 \leq \tau < 0.4$ a weak rank correlation, and $0 < \tau < 0.2$ a very weak rank correlation [153].

Note that this definition is equivalent to

$$\tau(C^{(r)}, C^{(s)}) = (C - D) / \binom{N}{2}, \quad (8.9)$$

where

- $C = \sum_i \sum_{j, i < j} [\text{sgn}(C_i^{(r)} - C_j^{(r)}) = \text{sgn}(C_i^{(s)} - C_j^{(s)})]$ is the number of *concordant*

pairs (i.e. the number of pairs of vertices (v_i, v_j) across the two lists such that *both* lists have the *same* ranking order of v_i and v_j)

- $D = \sum_i \sum_{j_{i < j}} \left[\text{sgn}(C_i^{(r)} - C_j^{(r)}) = -\text{sgn}(C_i^{(s)} - C_j^{(s)}) \right]$ is the number of *discordant pairs* (i.e. the number of pairs of vertices (v_i, v_j) across the two lists such that *both* lists have the *opposite* ranking order of v_i and v_j)

This allows the following probabilistic interpretation of Kendall's τ coefficient: $\tau(C^{(r)}, C^{(s)}) = p - (1 - p) = 2p - 1$ where $p = C / \binom{N}{2}$ is the probability of picking a concordant pair at random.

In the field of centrality analysis, Kendall's τ has become the definitive metric [154–157], by means of its ubiquity, efficient computability [158], and the fact that variants exist that take into account ties [159]. Despite this, Kendall's τ coefficient is not particularly suited towards comparing centrality measures. Measures with highly correlated top-ranked vertices may produce comparatively low τ values, as Kendall's τ equally weights all discordant pairs, regardless of where they appear in the ranking. Recently, weighted modifications have been proposed — specifically catered to comparing centrality measures — which use a hyperbolic weighting function to more heavily weight correlations of the top-ranked vertices. These include the AP (average precision) correlation [160] and Vigna's τ correlation coefficient [161]. As Vigna's τ further takes into account ties, we will apply Vigna's τ correlation coefficient to analyse the results of Fig. 8.2. We briefly describe Vigna's τ correlation coefficient here before presenting the results.

Definition 8.3 (Weighted centrality measure inner product). The weighted centrality inner product of the centrality measures $C^{(r)}$ and $C^{(s)}$ applied to graph vertex set V is defined by

$$\langle C^{(r)}, C^{(s)} \rangle_{w, \rho_{rs}} = \sum_i \sum_{\substack{j \\ i < j}} \text{sgn}(C_i^{(r)} - C_j^{(r)}) \text{sgn}(C_i^{(s)} - C_j^{(s)}) w(\rho_{rs}(i), \rho_{rs}(j)) \quad (8.10)$$

where $w(i, j) : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a weight function applied to vertices v_i and v_j , and the function $\rho_{rs} : [N] \rightarrow [N]$ is defined such that ρ_{rs} assigns a rank to each vertex depending on the centrality values given in $C^{(r)}$, with 1 the highest rank. In case of a tie in $C^{(r)}$, the tie is broken based on the ranking in $C^{(s)}$. If both cases are a tie, this doesn't matter, as it won't influence the value of τ .

Definition 8.4 (Vigna's τ correlation coefficient). Let $C^{(r)} \in \mathbb{R}^N$ and $C^{(s)} \in \mathbb{R}^N$ correspond to centrality measures $C^{(r)}$ and $C^{(s)}$ applied to graph vertex set V . The Vigna's τ correlation coefficient is given by averaging out the Kendall's τ coefficient calculated using

	Degree	PageRank	Eigenvector	CTQW	RWC
Degree	1.	0.882	0.882	0.882	0.875
PageRank	0.882	1.	0.622	0.622	0.687
Eigenvector	0.882	0.622	1.	1.	0.87
CTQW	0.882	0.622	1.	1.	0.87
RWC	0.875	0.687	0.87	0.87	1.
	Degree	PageRank	Eigenvector	CTQW	RWC
Degree	1.	0.924	0.86	0.851	0.925
PageRank	0.924	1.	0.687	0.689	0.77
Eigenvector	0.86	0.687	1.	0.95	0.877
CTQW	0.851	0.689	0.95	1.	0.861
RWC	0.925	0.77	0.877	0.861	1.

Table 8.1: Vigna’s τ coefficient comparing the vertex rankings for the labelled centrality measures, calculated for the Erdős-Rényi graph shown in Fig. 8.2 (top) and averaged over an ensemble of 200 random Erdős-Rényi graphs $G(20, 0.3)$ (bottom).

the weighted centrality inner product with both ρ_{rs} and ρ_{sr} :

$$\tau(C^{(r)}, C^{(s)}) = \frac{1}{2} \left(\frac{\langle C^{(r)}, C^{(s)} \rangle_{w, \rho_{rs}}}{\|C^{(r)}\|_{w, \rho_{rs}} \|C^{(s)}\|_{w, \rho_{rs}}} + \frac{\langle C^{(r)}, C^{(s)} \rangle_{w, \rho_{sr}}}{\|C^{(r)}\|_{w, \rho_{sr}} \|C^{(s)}\|_{w, \rho_{sr}}} \right) \quad (8.11)$$

For the calculations performed in this thesis, we choose an *additive* weight scheme, $w(i, j) = w_1(i) + w_2(j)$, as it provides a natural *top-k correlation measure* (ranks will be assigned significantly larger values after the first top k values). Following the analysis completed by Vigna [161], we select a hyperbolic weight function

$$w(i, j) = i^{-1} + j^{-1} \quad (8.12)$$

for the following reasons:

- As the rank of the vertex grows, the mass of the weight grows indefinitely
- It does not converge; a function that does converge (i.e. $1/i^2$) would make the influence of the low-rank elements vanish too quickly
- It decays sufficiently fast to discriminate extra information versus Kendall’s τ coefficient; a function that decays too slowly (i.e. logarithmic decay) would be very similar to Kendall’s τ .

Vigna’s τ correlation coefficients for Fig. 8.2 are shown in Tab. 8.1; for additional robust-

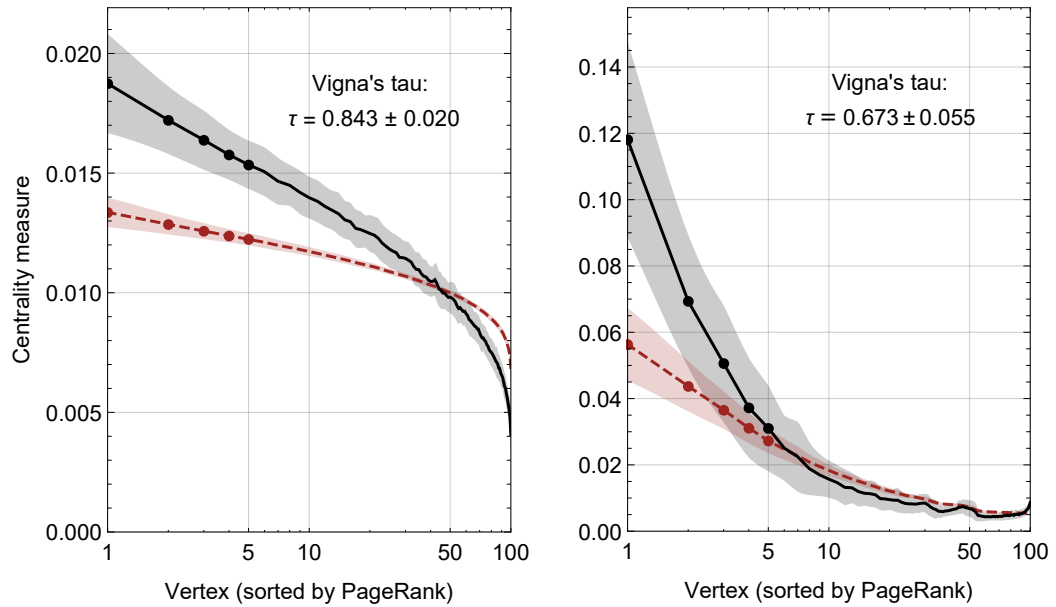
ness, this analysis is repeated and averaged over an ensemble of 200 randomly generated $G(20, 0.3)$ graphs. It can be seen that there is a very strong correlation between the CTQW centrality ranking and the eigenvector centrality ranking ($\tau = 0.95$ averaged across the ensemble). This lends credence our previous hypothesis in Sec. 7.3 — the CTQW centrality scheme appears to be ranking the graph vertices in a similar fashion to the eigenvector centrality, by considering walks of *all* lengths emanating from each vertex weighted inversely by length.

We now consider ensembles of larger Erdős-Rényi and scale-free graphs, and compare the CTQW centrality measure to the eigenvector centrality measure (its closest classical analogue), as well as the PageRank (the classical centrality measure with arguably the most impact in the last decade). This analyses will allow us to verify the behaviour of the CTQW centrality for large graphs of varying degree distributions.

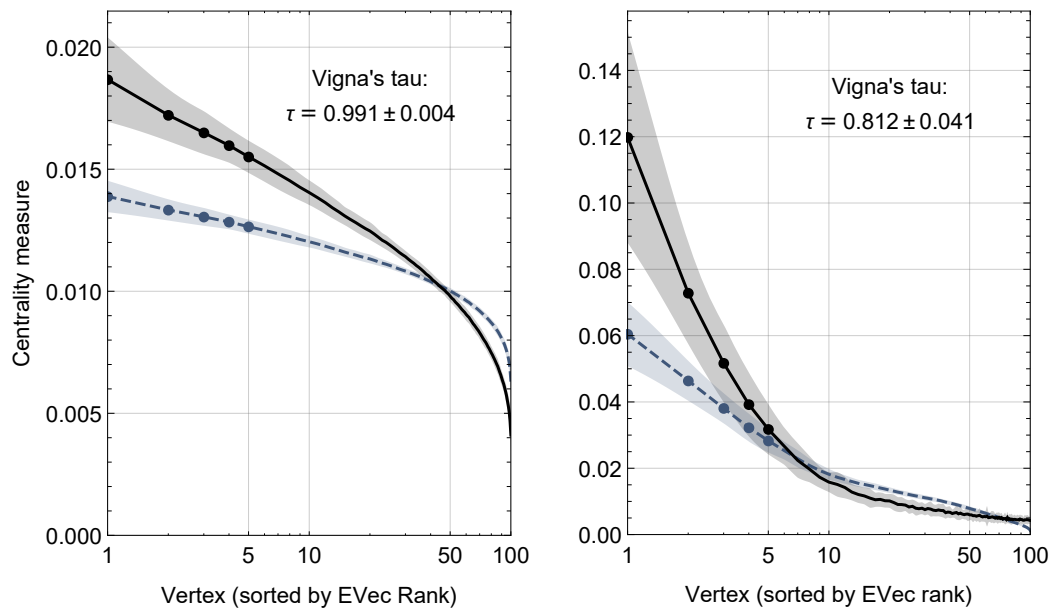
We begin by generating an ensemble of 200 Erdős-Rényi and scale-free graphs (the latter by way of the Barabási-Albert algorithm), and calculating the average PageRank, eigenvector, and CTQW centrality measures over the ensemble. These results are shown in Fig. 8.3. It can be seen that, on average, the CTQW ranking agrees with the classical algorithms on the location of the five most central vertices, whilst also following the following the same general trendline (binomial for the Erdős-Rényi, power law for the scale-free). In fact, the CTQW measure for the top 5 vertices *outperforms* that of the PageRank and eigenvector centrality, by assigning a higher centrality measure, perhaps allowing for greater distinguishability when sampled experimentally. However, it appears that this comes at the cost of larger measure variance compared to the PageRank.

Finally, examining the mean Vigna τ values for each of the ensembles, it can be observed that the results further support our conjecture that the CTQW centrality should provide a quantum eigenvector-like centrality. When compared to the eigenvector centrality, we observe a very strong correlation of ($\tau = 0.991 \pm 0.004$ for the Erdős-Rényi ensemble and $\tau = 0.812 \pm 0.041$ for the scale-free ensemble), with a strong correlation when compared to the PageRank ($\tau = 0.84 \pm 0.02$ for the Erdős-Rényi ensemble and $\tau = 0.673 \pm 0.055$ for the scale-free ensemble).

Note that Vigna's τ coefficient does a good job of informing us how *correlated* the ranked lists are, allowing us to classify the centrality measures based on how they encode information flow through the network. However, Vigna's correlation coefficient, whilst a better indicator of overall rank agreement than Kendall's τ , continues to suffer from the fact that small discrepancies in ranking of lower-ranked vertices negatively affect the coefficient value. For example, in Fig. 8.2 it can be seen that the PageRank (a radial volume



(a) The average CTQW centrality measure (black) compared to the average PageRank measure (red, dashed).



(b) The average CTQW centrality measure (black) compared to the average eigenvector centrality (blue, dashed)

Figure 8.3: The average CTQW centrality measure compared to classical centrality measures for vertices in an ensemble of 200 Erdős-Rényi graphs (left) and 200 scale-free graphs (right). The Erdős-Rényi graphs have parameters $N = 100$, $p = 0.3$. The scale-free networks are constructed via the Barabási-Albert algorithm with $N = 100$ and $m = 2$ edges added at every generation. The shaded areas represent one standard deviation from the average centralities, and the top 5 ranked vertices are shown by the markers.

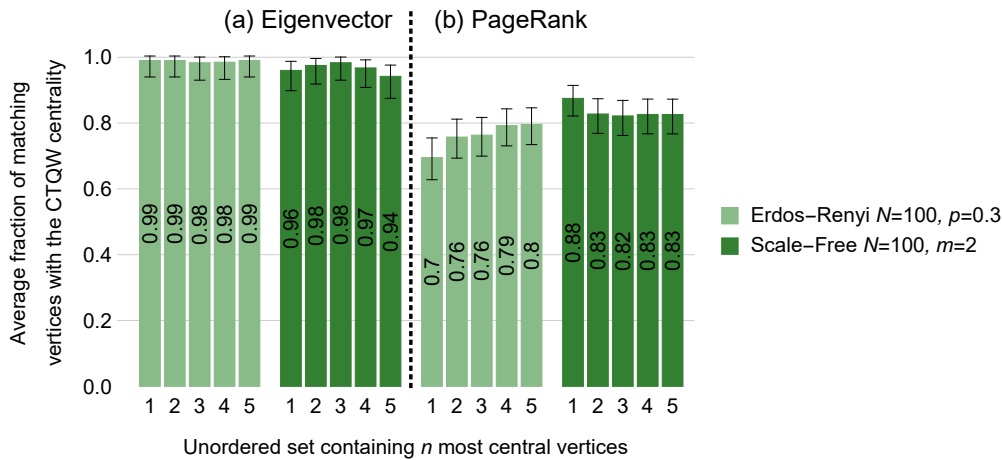


Figure 8.4: Charts showing the Jaccard set similarity between the CTQW centrality ranking and (a) the eigenvector centrality and (b) the PageRank ranking for an ensemble of 100 Erdős-Rényi and 100 scale-free graphs. Each bar represents the unordered set containing the n most central vertices as determined by the PageRank and CTQW measures, whilst the vertical axis gives the average fraction of matching vertices between the two sets. Error bars indicate the Agresti-Coull 95% confidence interval.

measure) and random walk centrality (a radial length measure) are in total agreement on the location of the top 3 most central vertices, despite exhibiting a lower correlation than other measures in agreement ($\tau = 0.687$). Thus, whilst these approaches might be useful in determining correlation between various centrality measures, they distract from the main question: how frequently do two centrality measures agree on the k top-most ranked vertices?

8.3.2 Agreement on top-ranked vertices

In order to answer this question quantitatively, we employ the Jaccard measure of set similarity [162]. This provides an indicator of how well each centrality measure is able to determine the identity of the top k highest centrality individuals. Firstly, for each graph, unordered sets containing the n most central vertices according to each measure were compared — the fraction of matching vertices providing a quantitative value for the agreement between the two measures. Finally, these were averaged over the entire ensemble, providing a general measure of the agreement between the PageRank and the CTQW centrality ranking, with uncertainty approximated by calculating the Agresti-Coull 95% confidence interval [163] — the results are presented in Fig. 8.4.

It can be seen that the eigenvector and CTQW centrality measures are in near perfect

agreement — agreeing on the most central node 99% of the time for Erdős-Rényi networks, and 96% of the time for scale-free networks. This is likely due to the strong correlation between the CTQW and eigenvector centralities noted previously, and indicates that this strong correlation continues to hold for larger graphs of varying degree distributions.

Turning our attention to the PageRank, we find a strong agreement with the CTQW measure, albeit not as strong as the eigenvector centrality; on the location of the most central vertex, they named the same vertex 88% of the time for scale-free graphs, dropping to 70% for Erdős-Rényi graphs. As the number of vertices compared increases, the agreement factors decrease slightly for the scale-free and increase slightly for the Erdős-Rényi, before both ending around 80% by the time the top 5 vertices are compared. This discrepancy might be partially explained by considering the CTQW measure variance in Fig. 8.3:

- For the Erdős-Rényi graphs, a majority of vertices have degree close to the mean, leading to the top-ranked vertices having similar centrality measures. The average CTQW centrality measure of the second and third-ranked vertices lies *within* the uncertainty region of the most central vertex; so even as the top 5 are easily distinguished, changes in their initial ordering might appear.
- For the scale-free graphs, with a small number of connected hubs, the hubs are easily distinguished by both measures. However, beyond the hubs, most vertices have similar degree due to the power law distribution — leading to small discrepancies between the measures as more vertices are ranked.

Nonetheless, our results here show that the CTQW measure proposed works excellently as a centrality measure on undirected graphs — it assigns higher values to the central node of a star graph and equal lower values to the surrounding nodes, correlates well with the classical eigenvector centrality (allowing us to posit that the CTQW measure extracts centrality in a similar fashion to the eigenvector centrality, namely via weighted walks of all lengths), and generalises to arbitrary random scale-free and Erdős-Rényi graphs. Thus, the proposed quantum scheme sufficiently determines node centrality, and in contrast to the Quantum PageRank algorithm (which requires computation of the dense Google matrix), preserves the sparse structure of the network in the Hamiltonian; a property that allows for known efficient quantum implementation [164].

In the following section, we build off this result to experimentally implement the CTQW centrality scheme on a star graph using linear optics — a proof-of-concept experiment and the first physical implementation (to our knowledge) of quantum centrality.

8.4 Experimental realization

Linear optics enables the efficient implementation of an arbitrary unitary transformation on various degrees of freedoms of a single photon. For example, any 2×2 unitary transformations on the polarizations of a photon can be realized by a set of half-wave plates (HWPs) and quarter-wave plates (QWPs) [165]. Here, we aim to devise a linear-optics realization to physically implement the previously described CTQW centrality measure on the 4-vertex star graph Fig. 8.1(a), using spatial and polarization degrees of freedoms of a single photon. This process can be summarized as follows; we first prepare a four-dimensional equal superposition quantum state $|\psi(0)\rangle = \frac{1}{2} \sum_{j=0}^3 |j\rangle$ within the single photon subspace of 4 independent modes, and then perform a 4×4 unitary transformations on the state. Finally, the probability distributions are then obtained through projective measurement on the state. The unitary transformations applied to the initial state $|\psi(0)\rangle$ are provided by $U(k\Delta t)$, where $k \in \{1, 2, \dots, 8\}$.

It is well known that arbitrary 4×4 unitary transformations can be decomposed using the cosine-sine decomposition method [166–171]. For each unitary transformation in $U(k\Delta t)$, there exist unitary matrices \mathbb{L} , \mathbb{S} , and \mathbb{R} , such that $U = \mathbb{L}\mathbb{S}\mathbb{R}$ where \mathbb{L} and \mathbb{R} are block-diagonal,

$$\mathbb{L} = \left[\begin{array}{c|c} L & 0 \\ \hline 0 & L' \end{array} \right], \mathbb{R} = \left[\begin{array}{c|c} R & 0 \\ \hline 0 & R' \end{array} \right], \quad (8.13)$$

and \mathbb{S} is an orthogonal cosine-sine matrix,

$$\mathbb{S} = \left[\begin{array}{cc|cc} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \hline -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{array} \right], \quad (8.14)$$

where L , L' , R , and R' are arbitrary 2×2 unitary transformations on two modes. This matrix \mathbb{S} can be further decomposed by a 2×2 unitary transformation

$$S = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (8.15)$$

on the subspace spanned by the modes $\{|0\rangle, |2\rangle\}$ and I on the subspace spanned by the modes $\{|1\rangle, |3\rangle\}$.

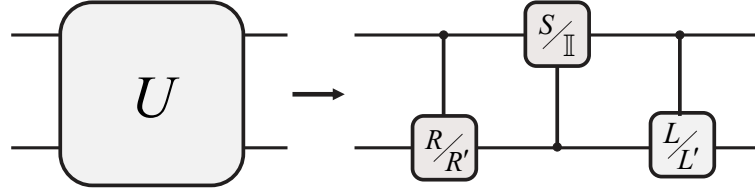


Figure 8.5: The quantum circuit for implementing the 4×4 unitary transformation U on a two-qubit system.

This decomposition method can be used to decompose any higher dimensional unitary operations into series of two dimensional unitary operations, and thus our technology can be used to realise, in principle, any dimensional unitary operations. However, it is noteworthy that the numbers of beam displacers (BDs) used to prepare a two-dimensional state and to realise the two-dimensional unitary operation are $d - 1$ and $2^d - 2$ respectively — in other words, the number of optical elements grows exponentially with the dimension of the unitary operation, and decoherence in cascaded interferometers also increases.

For convenience, we encode the four-dimensional quantum states by two-qubit states as $\{|0\rangle = |\tilde{0}\tilde{0}\rangle, |1\rangle = |\tilde{0}\tilde{1}\rangle, |2\rangle = |\tilde{1}\tilde{0}\rangle, |3\rangle = |\tilde{1}\tilde{1}\rangle\}$. The unitary transformations \mathbb{L} , \mathbb{S} , and \mathbb{R} can be rewritten as

$$\begin{aligned}\mathbb{L} &= |\tilde{0}\rangle\langle\tilde{0}| \otimes L + |\tilde{1}\rangle\langle\tilde{1}| \otimes L', \\ \mathbb{S} &= S \otimes |\tilde{0}\rangle\langle\tilde{0}| + \mathbb{I} \otimes |\tilde{1}\rangle\langle\tilde{1}|, \\ \mathbb{R} &= |\tilde{0}\rangle\langle\tilde{0}| \otimes R + |\tilde{1}\rangle\langle\tilde{1}| \otimes R'.\end{aligned}\tag{8.16}$$

The 4×4 unitary transformations $U(k\Delta t)$ can then be implemented by these three controlled two-qubit transformations in Fig. 8.5.

A schematic of our experimental setup is depicted in Fig. 8.6. Note that the two qubits are encoded via the spatial and polarization modes of a single photon; the first qubit $|\tilde{0}\rangle$ ($|\tilde{1}\rangle$) represents the upper (lower) spatial mode of the photon, whilst the second qubit $|\tilde{0}\rangle$ ($|\tilde{1}\rangle$) represents the horizontal (vertical) polarization of the photon. Polarization-degenerated photon pairs are generated by type-I spontaneous parametric down-conversion in 0.5mm-thick nonlinear- β -barium-borate crystal pumped by a 400.8nm CW diode laser with 90mW of power, with the single photon generated by triggering of the other photon. Next, interference filters are used to restrict the photon bandwidth to 3nm — the photons are in horizontal polarization after the first polarizing beam splitter.

The initial 4 vertex equal superposition state is prepared in two steps. Firstly, after passing through a half-wave plate (HWP_a) at 22.5° — which rotates the polarization of

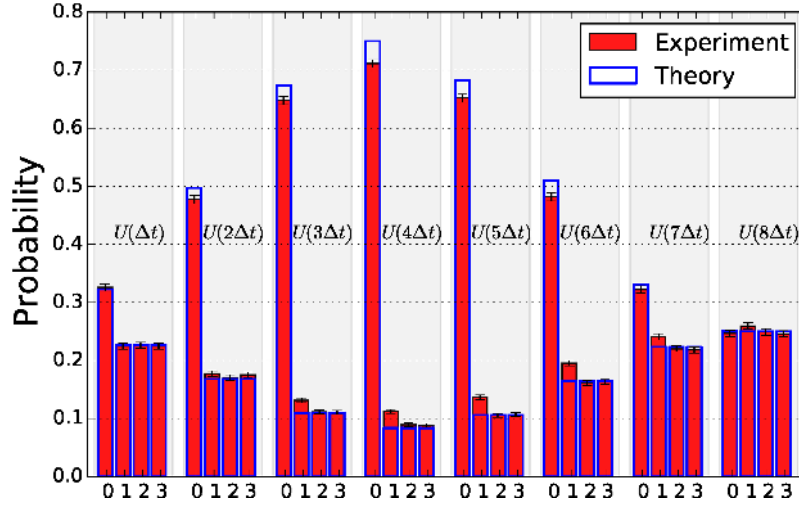


Figure 8.8: Photon probability distributions after eight unitary transformations. Red bars represent experimental results. Blue borders represent theoretical predictions. Errors are estimated via propagated Poissonian statistics. A small additional uncertainty may be present in the measurement of nodes 0 and 1, due to the photon representing the two states with different polarization but inhabiting the same spatial mode.

the photon to equal superposition of horizontal and vertical polarizations — the photons are split into two parallel paths by a birefringent calcite BD, which transmits the vertically polarized photons directly and displaces horizontally polarized photons by 3mm. Second, two HWPs (HWP_b and HWP_c), at -22.5° and 22.5° , are inserted into the upper and lower modes respectively to flip the polarizations. Thus, the state of the single photons is prepared in $|\psi(0)\rangle = \frac{1}{\sqrt{2}} \sum_{j=0}^3 |j\rangle$.

For the controlled two-qubit transformations \mathbb{L} and \mathbb{R} , the spatial mode of photons serves as the control qubit, with the polarization is the target qubit. In the upper and lower modes, the 2×2 unitary transformations L (R) and L' (R') are applied to the polarization degrees of freedom, which can be realized by a combination of QWPs and HWPs sequence inserted into the corresponding spatial mode.

Finally, for the 4×4 unitary transformation \mathbb{S} , the polarization of photons serves as the control qubit. An HWP at 45° inserted into the lower input mode first flips the polarization of photons. After the first BD, the horizontally polarized photons in both the upper and lower input modes are propagating in the same path (the middle), and as a result the transformation S is then applied by using an HWP at $\theta/2$ in the middle path; i.e. $S \otimes |\tilde{0}\rangle \langle \tilde{0}|$ is applied to states $|\tilde{0}\tilde{0}\rangle$ and $|\tilde{1}\tilde{0}\rangle$. The vertically polarized photons in the upper or lower input modes are not affected, and after the second BD remain in the upper or lower output modes; that is, $\mathbb{I} \otimes |\tilde{1}\rangle \langle \tilde{1}|$ is applied to states $|\tilde{0}\tilde{1}\rangle$ and $|\tilde{1}\tilde{1}\rangle$. Two HWPs at 45° inserted into the other two paths (the propagating paths between two BDs) are used

to flip the polarizations of photons in these paths, with the propagating modes changing after the photons pass through the following BD. After the second BD, an HWP at 45° is inserted into the upper output mode to compensate for the effect of the first HWP in the lower input mode, flipping the polarizations of photons back.

Our actual experimental setup is shown in Fig. 8.7, which takes into consideration the compensation of optical delay between different spatial modes. The simplified set of wave plates for the realizations \mathbb{L} and \mathbb{R} of the eight 4×4 unitary transformations $U(k\Delta t)$ is listed in Tab. 8.2, including the setting angles of wave plates. Two BDs and six HWPs (HWP₁-HWP₆) are used to realize \mathbb{S} and compensate for the optical delay. The setting angles of HWP₁-HWP₆ are listed in Tab. 8.3.

Table 8.2: The simplified sets of wave plates with indicated setting angles for realization of eight 4×4 unitary transformation. The subscript k of L , L' , R and R' corresponds to the k th unitary transformation $U(k\Delta t)$. Q, quarter-wave plate; HWP, half-wave plate; WP, wave plate.

L/R	WPs	Angles ($^\circ$)	L'/R'	WPs	Angles ($^\circ$)
L_1	Q-H-H	90,0,-3.3	L'_1	H-H-H	90,0,157.5
R_1	H-H-Q	0,48.4,90	R'_1	H-Q-Q	0,22.5,22.5
L_2	Q-H-H	90,0,-6.7	L'_2	H-H-H	90,0,157.5
R_2	H-H-Q	0,51.7,90	R'_2	H-Q-Q	0,22.5,22.5
L_3	Q-H-H	90,0,-10.5	L'_3	H-H-H	90,0,157.5
R_3	H-H-Q	0,55.5,90	R'_3	H-Q-Q	0,22.5,22.5
L_4	Q-H-Q	90,14.9,0	L'_4	H-H-H	90,0,157.5
R_4	Q-H-H-Q	0,0,30.1,0	R'_4	Q-Q-Q-Q	0,0,22.5,22.5
L_5	Q-H-H	90,0,-20.2	L'_5	H-H-H	90,0,157.5
R_5	H-H-Q	0,65.2,90	R'_5	H-Q-Q	0,22.5,22.5
L_6	Q-H	90,0,26.8	L'_6	H-H	90,22.5
R_6	H-H-Q	0,-18.2,90	R'_6	H-Q-Q	0,22.5,22.5
L_7	Q-H-H	90,0,-35.0	L'_7	Q-Q-H	0,0,112.5
R_7	H-Q	100.0,90	R'_7	H-H	0,22.5
L_8	Q-H-H-Q	90,0,-44.4,0	L'_8	Q-Q-Q-Q	-22.5,-22.5,90,90
R_8	Q-H-H-Q	0,0,90.6,0	R'_8	Q-Q-Q-Q	0,0,22.5,22.5

In order to implement the proposed centrality algorithm experimentally, we discretise the CTQW time-evolution operator $U(t) = e^{-iLt}$ given by Eq. 8.4 for the 4-vertex star graph, using 8 time steps of $\Delta t = 9/40$ to ensure that we sample the probability dis-

Table 8.3: Setting angles of HWP₁ and HWP₂ for realization of the eight 4×4 unitary transformations. The setting angles of HWP₃, HWP₄, and HWP₅ are set to -45° , and that of HWP₆ is set to 45° for all eight $U(k\Delta t)$ values.

k	1	2	3	4	5	6	7	8
HWP ₁ (°)	54.1	63.1	71.8	80.0	87.5	86.3	82.0	80.3
HWP ₂ (°)	144.1	153.1	161.8	170.0	177.5	176.3	172.0	170.3

tribution adequately over one period (note that $T = \pi/\sqrt{3} \approx 8\Delta t$). After application of the unitary time-evolution operator $U(k\Delta t)$ with $k \in \{1, 2 \dots 8\}$, the quantum state is measured by a two-qubit projective measurement. A polarising beam splitter is then used to perform the projective measurement on the photons with the computational basis $\{|\tilde{0}\tilde{0}\rangle, |\tilde{0}\tilde{1}\rangle, |\tilde{1}\tilde{0}\rangle, |\tilde{1}\tilde{1}\rangle\}$, with the photons detected by avalanche photon-diodes in coincidence with the trigger set D_T , with a coincident window of 3ns. The clicks of detectors D_0 , D_1 , D_2 , and D_3 correspond to the probabilities of the final state projected onto the basis $\{|\tilde{0}\tilde{0}\rangle, |\tilde{0}\tilde{1}\rangle, |\tilde{1}\tilde{0}\rangle, |\tilde{1}\tilde{1}\rangle\}$ — we record the clicks for 5s, and more than 18000 coincidence counts are detected in the overall measurement time. Note that, as the experiment is performed with a single photon, there are no coincidence counts between D_1 , D_2 , D_3 , and D_4 .

The measured probability distributions are shown in Fig. 8.8, which are in excellent agreement with the theoretical predictions given by Eq. 8.5. Here we use the 1-norm distance,

$$d = \frac{1}{2} \sum_{x=0,1,2,3} |P^{exp}(x) - P^{th}(x)|, \quad (8.17)$$

to evaluate the quality of experimental demonstration. Each term in the 1-norm distance evaluates the difference between the experimental and theoretical probability in each basis; thus, a 1-norm distance of 0 indicates perfect match between the experimental and theoretical results. For each of the eight discrete unitary time-evolution operators $U(k\Delta t)$, we obtain $d_1 = 0.003$, $d_2 = 0.020$, $d_3 = 0.026$, $d_4 = 0.039$, $d_5 = 0.031$, $d_6 = 0.031$, $d_7 = 0.017$, and $d_8 = 0.009$. The distances are all smaller than 0.04, which indicates successful experimental demonstrations of the 4×4 unitary transformations.

8.5 Conclusion

In this study, we have proposed a CTQW-based quantum centrality algorithm, shown that it correlates well with classical measures, and verified its performance on general random

graphs. The proposed quantum measure was then successfully implemented experimentally for the first time for a 4-vertex star graph. Notably, this algorithm requires an N -dimensional Hilbert space, compared to discrete-time quantum walk-based algorithms, which require N^2 dimensions for the same graph. Furthermore, this algorithm preserves the full quantum behavior of the walker, unlike the QSW, which mutes the quantum behaviour of the walker due to decoherence.

Secondly, this study reports the first successful physical demonstration of a quantum centrality algorithm on a 4-vertex star graph. In our physical implementation of the proposed CTQW centrality algorithm, the unitary operation of the walker on the graph is decomposed into unitary transformations on a two dimensional subspace, and realised by operating on the polarisation and spacial modes of a single photon. This method can be used to decompose, in principle, any dimensional unitary operations into series of two dimensional unitary operations. By making use of the coherent property of photons, the technology in our experiment is a competitive candidate for demonstrating arbitrary unitary operations, allowing it to be utilised for a wide array of quantum algorithms and quantum information processes.

However, there is one setback to our CTQW centrality scheme compared to the Quantum PageRank and QSW; both of these allow for centrality analysis on *directed* networks, via the Szegedy formalism and decoherence respectively, whilst the CTQW remains undefined and non-unitary. In the next chapter, we will use the framework of PT-symmetry to define a generalization of the CTQW, the pseudo-Hermitian η -CTQW, allowing us to extend our centrality scheme to select directed graphs.

8.6 Acknowledgments

This work was supported by NSFC (Grants No. 11474049 and No. 11674056), NSFJS (Grant No. BK20160024), and the Scientific Research Foundation of the Graduate School of Southeast University (Grant No. YBJJ1623). Thank you to X. Zhan, Z. Bian, K. Wang, and J. Li, who performed the experiment, generated figures associated with the experimental results and set-up, and performed data analysis. X. Zhan and P. Xue drafted the experimental section of the manuscript. J. A. Izaac would like to thank the Hackett Foundation and The University of Western Australia for financial support.

CHAPTER 9

CTQW centrality on directed graphs via PT-symmetry

This chapter is based on the paper by J. A. Izaac, J. B. Wang, P. C. Abbott, and X. S. Ma, “Quantum centrality testing on directed graphs via PT-symmetric quantum walks”, arXiv:1607.02673v3 [quant-ph] (2017)., with minimal modification to match the formatting, notation, and terminology of this thesis. The paper has been submitted for publication in Physical Review A.

In the previous chapter, we introduced a CTQW-based quantum centrality scheme that was shown to highly correlate with the classical eigenvector centrality — perhaps acting as an extension of eigenvector centrality to the quantum realm. Unfortunately, the CTQW framework is only defined for *undirected graphs*; attempting to extend it as-is to directed network structures results in a loss of unitarity of the quantum walker, and we are no longer able to interpret the state vector as a probability amplitude. In order to rectify this issue, in this chapter we introduce an extension of the framework that allows us to propagate CTQWs on a subset of directed graphs that exhibit a property known as PT-symmetry. As a result, we are able to generalize the CTQW centrality scheme to include PT-symmetric directed graphs, and we present a statistical analysis highlighting that, on directed graphs, this quantum centrality scheme continues to strongly agree with classical centrality measures such as PageRank.

9.1 Introduction

As noted, a major disadvantage of the quantum walk is the condition of unitarity, required due to the quantum nature of the walkers. It follows that the standard quantum walk is unable to model or analyse directed network structures, without either a) resulting in non-unitary dynamics, or b) modifying the framework. This serves as a particular hindrance in extending established quantum algorithms (e.g. centrality measures, quantum search, graph isomorphism) and quantum dynamical models (such as transport of electrons or excitons) to systems with directions or biased potentials. Compare this to a

classical random walk, where as long as the transition matrix remains stochastic, directed networks pose no problem. Consequently, various work-arounds have been proposed for dealing with this non-unitary behaviour, for example Szegedy quantum walks [12, 13] and quantum stochastic walks (QSW) [13, 78]. Unfortunately, as previously discussed, there are numerous downsides to these approaches — either requiring a significantly expanded Hilbert space, or muting the effect of quantum behaviour (due to environmental dephasing and loss in the open-quantum walk).

An alternative solution to this issue arrives in the form of PT-symmetry. First discovered by Bender and Boettcher [81], PT-symmetry started off as a simple curiosity — the appearance of non-Hermitian Hamiltonians that exhibit a real eigenspectra, thus allowing non-unitary probability conservation via a redefinition of the Hilbert space inner product. This was attributed to parity-time symmetry of the non-Hermitian Hamiltonian, and over time was generalised to allow for non-Hermitian Hamiltonian symmetry under a combination of *any* linear and non-linear operators [81–85]. Simultaneously, PT-symmetric Hamiltonians were finding use in theoretical models of observed and predicted phenomena in condensed matter [173], quantum field theory [174, 175], and being observed and implemented in numerous optical experiments [176–179]; whilst older non-Hermitian studies in condensed matter [180] and nuclear physics [181] have been reformulated in the PT-symmetry framework [182]. Recently, PT-symmetry has been used to model directed one-dimensional discrete-time quantum walks (DTQWs) [183], and considered in the case of continuous-time quantum walks (CTQWs) [86].

In this study, we present a rigorous framework for PT-symmetric CTQWs, extending the formalism of Salimi and Sorouri [86] to ensure the initial quantum state vector is preserved. This is then broadened to the cases of multi-particle systems and interdependent networks, before being utilised to measure vertex centrality in several example graphs, in an extension of chapter 8. We show that our PT-symmetric centrality scheme compares well to the classical PageRank algorithm [40], and in some cases even breaks the vertex rank degeneracy characterized by the PageRank.

Similarly to the previously introduced CTQW centrality measure on undirected graphs, but unlike previous quantum centrality measures [12, 13, 78], for a graph of N vertices this scheme requires a Hilbert space of dimension N (compare this to the Szegedy quantum walk based PageRank scheme, which requires a N^2 dimensional Hilbert space), and without the classical decoherence required for QSWs. Furthermore, we show that this formalism is equivalent to considering an undirected, yet weighted, complete graph with self-loops, providing a structural interpretation that may lead to simple experimental im-

plementation.

This chapter is structured as follows. In Sec. 9.2, we detail the standard CTQW, and briefly discuss the issues with extending this to directed graphs. A brief introduction to the PT-symmetry and pseudo-Hermitian formalism is presented in Sec. 9.3, tailored towards a quantum walker application. In Sec. 9.4, we introduce our framework for PT-symmetric continuous-time quantum walks, and discuss how this can be interpreted as a mapping to a weighted and undirected CTQW on a complete graph; the framework is then extended to multi-particle systems and interdependent networks. A potential application, network centrality, is presented in Sec. 9.5, and a statistical analysis performed to verify its performance over an ensemble of randomly generated Erdős-Rényi and scale-free graphs. Finally, our conclusions are provided in Sec. 9.6.

9.2 CTQWs on directed graphs

Consider an arbitrary undirected graph $G(V, E)$, composed of vertices $v_j \in V$ and edges $(v_i, v_j) \in E$, with corresponding adjacency matrix A . As we saw in Sec. 3.2.2, for a continuous-time quantum walk on graph G , the Hamiltonian can either be given by the graph Laplacian or the adjacency matrix, depending on the chosen convention. Furthermore, the graph Laplacian (Eq. 2.14) is constructed from the Hermitian adjacency matrix and is Hermitian itself — as a result, no matter what we choose for the Hamiltonian, for a CTQW on an undirected graph we are guaranteed an Hermitian Hamiltonian and therefore unitary time-evolution. Hence, the norm of $|\psi(t)\rangle$ is conserved under a continuous-time quantum walk, as required.

Let's now modify G so that it is a *directed* graph — that is, the edgeset $(i, j) \in E$ is now described by an *ordered* pair of vertices. As the adjacency matrix is no longer symmetric ($A_{ij} \neq A_{ji} \ \forall i, j$), the Hamiltonian H is no longer Hermitian, and thus we no longer have unitary time evolution ($UU^\dagger \neq I$). As a consequence, the total norm of the state vector, previously corresponding to quantum walker probability as per the Born rule, is no longer conserved,

$$\langle \psi(t) | \psi(t) \rangle = \langle \psi(0) | U^\dagger U | \psi(0) \rangle \neq \langle \psi(0) | \psi(0) \rangle, \quad (9.1)$$

and may in fact grow or decay exponentially. Various modifications proposed for dealing with this non-unitary behaviour (for example, Szegedy quantum walks [12, 13] and open-quantum walks [13, 78]) require a significantly expanded Hilbert space, resulting in

considerable resource overhead in physical implementation.

9.3 PT-symmetry

Whilst non-Hermitian Hamiltonians with complex eigenvalues result in exponentially growing or decaying time-evolution, a wide variety of non-Hermitian Hamiltonians have been found to possess real eigenvalue spectra. It was first noted by Bender and Boettcher [81] that particular non-Hermitian Hamiltonians with real spectra exhibited PT-symmetry, that is,

$$[H, \mathcal{PT}] = 0 \quad (9.2)$$

where $\mathcal{P} : (\hat{x}, \hat{p}) \rightarrow (-\hat{x}, -\hat{p})$ is the parity transformation operator, and $\mathcal{T} : (\hat{x}, \hat{p}) \rightarrow (\hat{x}, -\hat{p})$ the time reflection operator satisfying $\{\mathcal{T}, i\} = 0$ (anti-linearity) [81, 184]. On the basis of this observation, it was posited that invariance of a Hamiltonian under PT-transformations provides a more general condition for the reality of eigenspectra than simply Hermiticity. Immediately, research into PT-symmetric Hamiltonians found it was not so clear-cut; due to the anti-linearity of the \mathcal{PT} operator, a \mathcal{PT} invariant Hamiltonian may still undergo spontaneous symmetry breaking, leading to complex conjugate pairs of eigenvalues [185, 186]. Furthermore, although the existence of PT-symmetry is a sufficient condition for real spectra, it is not necessary. This same property can be found in Hamiltonians not exhibiting PT-symmetry — thus failing to account for the existence of *all* non-Hermitian Hamiltonians with real eigenspectra.

An alternative framework was put forward by Mostafazadeh [82]. Denoted pseudo-Hermiticity, it was shown that for all diagonalizable non-Hermitian Hamiltonians exhibiting a real eigenspectra, there exists a positive semidefinite linear operator $V = \eta^\dagger \eta$ such that

$$H^\dagger = VHV^{-1}. \quad (9.3)$$

Additionally, it was proven that every PT-symmetric and diagonalizable Hamiltonian is pseudo-Hermitian [82]. Coupled with the fact that $V = I$ corresponds to the case of Hermitian H , it was claimed that the pseudo-Hermiticity framework is the correct generalization of Hermiticity to non-Hermitian Hamiltonians.

Subsequent research has further explored the connections and similarities between PT-symmetry and pseudo-Hermiticity [83], with a flurry of papers released proclaiming the

supremacy of one or the other, in the constant struggle to be seen as the more general of the two. Of particular note, it has been shown that even though the pseudo-Hermitian similarity transform is decidedly linear, pseudo-Hermiticity is a necessary and sufficient condition for H to admit anti-linear symmetry [187–189]; i.e. the condition of pseudo-Hermiticity is equivalent to the condition $[H, \Omega] = 0$, where Ω is an anti-linear invertible or involutory operator. On this basis, one can conclude that any time-reversal invariant Hamiltonian belongs to the class of pseudo-Hermitian Hamiltonians, although the converse is not true, as Ω is not always guaranteed to be \mathcal{T} .

These two competing frameworks were finally reconciled by Bender and Mannheim [190], who introduced the concept of *generalized* PT-symmetry; here, \mathcal{P} represents *any* linear operator (not just parity), and likewise \mathcal{T} represents *any* anti-linear operator; the chosen operators \mathcal{P} and \mathcal{T} need not commute. This generalized PT-symmetry condition is necessary and sufficient for reality of the characteristic equation,

$$|H - \lambda I| = 0, \quad (9.4)$$

which results in real eigenvalues if \mathcal{PT} and H are simultaneously diagonalizable, and complex conjugate pairs if not¹. Thus, a Hamiltonian with a real eigenspectra necessarily displays (generalized) PT-symmetry, regardless of its diagonalizability — providing the generalisation of Hermiticity so sought after in the original parity-time and pseudo-Hermiticity frameworks. From hereon in, use of the term ‘PT-symmetry’ will refer to generalized PT-symmetry.

Under this new, more general, framework, pseudo-Hermiticity exists as a subset of PT-symmetry [84, 190], and has been expanded to include cases where H is non-diagonalizable [189] — in such cases, it is no longer possible to satisfy the pseudo-Hermiticity similarity transform with a linear operator V that is positive semidefinite — leading to spontaneous symmetry breaking and complex conjugate pairs of eigenvalues. Moreover, Bender and Mannheim [190] provides a criteria to determine whether a positive semi-definite V exists for known PT-symmetric Hamiltonians². In the following section, we will briefly outline the pseudo-Hermitian operator framework, and provide a method for determining $V = \eta^\dagger \eta$ in cases where there is no spontaneous PT-symmetry breaking.

¹This is an example of spontaneous PT-symmetry breaking. Even though a Hamiltonian may display \mathcal{PT} invariance (i.e. $[H, \mathcal{PT}] = 0$), the eigenstates of H , denoted $|\phi_n\rangle$, are not necessarily simultaneously eigenstates of \mathcal{PT} , due to the antilinearity of the \mathcal{PT} operator. If this is the case, then the eigenspectrum is composed of complex conjugate pairs of eigenvalues, and $\mathcal{PT}|\phi_n\rangle$ provides the eigenstates of \mathcal{PT} .

²If $[\mathcal{C}, \mathcal{PT}] = 0 \forall \mathcal{C}$ s.t. $\mathcal{C}^2 = 1$ and $[\mathcal{C}, H] = 0$, then there exists a positive semidefinite linear operator $V = \mathcal{C}\mathcal{P}$ s.t. $VHV^{-1} = H^\dagger$

Let H be a non-Hermitian matrix. It is pseudo-Hermitian (and thus PT-symmetric), if it is related by a similarity transform to a Hermitian matrix \tilde{H} ,

$$\tilde{H} = \eta H \eta^{-1}, \quad (9.5)$$

where η is frequently referred to in the literature as the *pseudo-Hermitian operator* or *metric*. Without loss of generality, we assume η is an Hermitian operator ($\eta = \eta^\dagger$). Due to the properties of a similarity transform, the eigenvalues of H will be the same as \tilde{H} and necessarily real³. Taking the conjugate transpose of this result, we get

$$\tilde{H}^\dagger = (\eta^{-1})^\dagger H^\dagger \eta^\dagger = \eta^{-1} H^\dagger \eta. \quad (9.6)$$

Since \tilde{H} is Hermitian, $\eta^{-1} H^\dagger \eta = \eta H \eta^{-1}$, and thus a pseudo-Hermitian matrix must satisfy the following similarity transform with its conjugate transpose:

$$H^\dagger = \eta^2 H \eta^{-2}. \quad (9.7)$$

Rewriting this in the form $\eta^2 H = H^\dagger \eta^2$, note that the right-hand side is simply the Hermitian conjugate of the left-hand side. This suggests that the following redefinition of the inner product,

$$\langle \cdots | \cdots \rangle_\eta := \langle \cdots | \eta^2 | \cdots \rangle, \quad (9.8)$$

should be sufficient to conserve the systems probability. Indeed, by using the Schrödinger equation, we see that this is in fact the case when working with pseudo-Hermitian operators:

$$\begin{aligned} \frac{d}{dt} \langle \psi(t) | \psi(t) \rangle_\eta &= \left\langle \frac{d}{dt} \psi(t) \middle| \eta^2 \middle| \psi(t) \right\rangle + \left\langle \psi(t) \middle| \eta^2 \middle| \frac{d}{dt} \psi(t) \right\rangle \\ &= \left\langle \psi(t) \middle| i H^\dagger \eta^2 \middle| \psi(t) \right\rangle - \left\langle \psi(t) \middle| \eta^2 i H \middle| \psi(t) \right\rangle \\ &= i \left\langle \psi(t) \middle| \left(H^\dagger \eta^2 - \eta^2 H \right) \middle| \psi(t) \right\rangle \\ &= 0. \end{aligned}$$

As Hermitian matrices are always diagonalisable by their unitary eigenbasis ($\tilde{H} = P \Lambda P^\dagger$ where $P^{-1} = P^\dagger$), it follows from the similarity relation Eq. 9.5 that pseudo-Hermitian

³See Appendix A.1

matrices must also be diagonalisable⁴:

$$H = \eta^{-1} \tilde{H} \eta = (\eta^{-1} P) \Lambda (P^\dagger \eta) = (P^\dagger \eta)^{-1} \Lambda (P^\dagger \eta).$$

Diagonalisable matrices must admit a biorthonormal eigenbasis⁵ [191],

$$H |\psi_j\rangle = \lambda_j |\psi_j\rangle, \quad (9.9)$$

$$H^\dagger |\phi_j\rangle = \lambda_j |\phi_j\rangle, \quad j = 1, 2, \dots, n, \quad (9.10)$$

where $\langle \phi_i | \psi_j \rangle = \delta_{ij}$ and $\lambda_j \in \mathbb{R}$ due to pseudo-Hermiticity. The completeness relation is given by

$$I = \sum_j |\psi_j\rangle \langle \phi_j|. \quad (9.11)$$

By applying the pseudo-Hermiticity relation (Eq. 9.7) to the biorthonormal eigenvector equations, we can deduce a method of constructing η . For instance,

$$H^\dagger |\phi_j\rangle = \eta^2 H \eta^{-2} |\phi_j\rangle = \lambda_j |\phi_j\rangle. \quad (9.12)$$

Pre-multiplying both sides by η^{-2} ,

$$H (\eta^{-2} |\phi_j\rangle) = \lambda_j (\eta^{-2} |\phi_j\rangle), \quad (9.13)$$

it can be seen that

$$|\psi_j\rangle = \eta^{-2} |\phi_j\rangle \Leftrightarrow |\phi_j\rangle = \eta^2 |\psi_j\rangle. \quad (9.14)$$

Hence, η acts to transform the pseudo-Hermitian operator to a Hermitian operator, and η^2 acts to transform between the pseudo-Hermitian biorthonormal eigenbasis. It therefore follows that the biorthonormal eigenbasis is the basis for the inner product space defined by Eq. 9.8. Combining this result with the biorthonormal completeness relation, we arrive

⁴In fact, solving this equation allows you to find an expression for η in terms of the left eigenvectors of H , $\langle \phi_j|$, and the eigenvectors of \tilde{H} , $|\tilde{\psi}_j\rangle$: $\eta_{ij} = \sum_k |\tilde{\psi}_k\rangle_i \langle \phi_j|_k$

⁵See Appendix A.3 for a short explanation why.

at a method of constructing the pseudo-Hermitian operator η ⁶:

$$\eta = \sqrt{\eta^2 \sum_j |\psi_j\rangle \langle \phi_j|} = \sqrt{\sum_j |\phi_j\rangle \langle \phi_j|} \quad (9.15)$$

and similarly

$$\eta^{-1} = \sqrt{\sum_j |\psi_j\rangle \langle \phi_j| \eta^{-2}} = \sqrt{\sum_j |\psi_j\rangle \langle \psi_j|}. \quad (9.16)$$

Since we have defined $V = \eta^2$, it follows that

$$V = \sum_j |\phi_j\rangle \langle \phi_j| \Leftrightarrow V^{-1} = \sum_j |\psi_j\rangle \langle \psi_j|. \quad (9.17)$$

Further, as V is positive semidefinite, we can be assured that the square root function applied to the operators in Eq. 9.15 and Eq. 9.16 is well defined, albeit admitting multiple solutions. For consistency, we will choose η to be the *principal square root* — the unique positive semidefinite square root of V .

So, to briefly summarise, H is necessarily pseudo-Hermitian (and consequently PT-symmetric) if it satisfies **any one** of the following equivalent conditions:

1. **H is similar to a Hermitian matrix.** There exists a Hermitian operator η and a Hermitian matrix \tilde{H} such that $\tilde{H} = \eta H \eta^{-1}$.
2. **H is similar to its own Hermitian conjugate.** There exists a positive Hermitian operator η such that $H^\dagger = \eta^2 H \eta^{-2}$.
3. **H has real eigenvalues and is diagonalizable** Note that a matrix is diagonalisable if and only if it has n linearly-independent eigenvectors⁷ [191].

9.4 Pseudo-Hermitian continuous-time quantum walks

First introduced by Salimi and Sorouri [86], pseudo-Hermitian continuous-time quantum walks take advantage of the pseudo-Hermitian structure of various graphs in order to implement directed quantum walks. In their study, the transition probability of the pseudo-

⁶Note that when H is Hermitian, the left and right eigenvectors are identical, and this equation reduces to the trivial result $\eta = \eta^{-1} = I$.

⁷See Appendix A.2 for details regarding this linear algebra result

Hermitian CTQW at vertex j at time t is defined to be

$$\mathbb{P}_j(t) = |\langle j | \eta e^{-iHt} | \psi(0) \rangle|^2. \quad (9.18)$$

where $H = L$, with $L_{ij} = \delta_{ij} \sum_k A_{ik} - A_{ij}$ the directed graph Laplacian consisting of in-degrees on the diagonal.

Note that this does *not* preserve the initial state;

$$\mathbb{P}_j(0) = |\langle j | \eta | \psi(0) \rangle|^2 \neq |\langle j | \psi(0) \rangle|^2. \quad (9.19)$$

Thus, if $|\psi(0)\rangle$ is chosen to be an equal superposition over all vertices, this will not be reflected in the quantum walk at time $t = 0$, making this definition unsuitable for algorithms such as centrality testing and graph isomorphism. Thus, rather than utilise their implementation, we present an alternative formulation.

As our aim is to experimentally produce a pseudo-Hermitian CTQW for network analysis, rather than redefine the Hilbert space inner-product as per Eq. 9.8, the inner product will not be modified⁸. Instead, we have three options available:

- A. **No modification:** assuming analytic continuation, implement the time-evolution operator on directed graphs using the normal framework for the CTQW, $U = e^{-iHt}$, with no redefinition of the inner-product. This results in a non-unitary and non-norm conserving time-evolution — and hence the state vector cannot be interpreted as a probability amplitude — but the pseudo-Hermiticity of the system ensures the norm will just oscillate with no exponential growth and decay.
- B. **Modify the time-evolution operator:** the non-unitary time evolution operator from (A) is instead modified as follows,

$$\tilde{U}(t) = \eta U(t) \eta^{-1} = \eta e^{-iHt} \eta^{-1}, \quad (9.20)$$

where $\tilde{U}(t)$ is unitary due to the pseudo-Hermitian similarity transform of the matrix exponential. This reflects the underlying directional structure of the graph, whilst allowing for probability conservation. Note that whilst the product is unitary, η and $U(t)$ are non-unitary matrices.

- C. **Modify the Hamiltonian:** in this approach, the pseudo-Hermitian Hamiltonian is

⁸As an aside, Appendix A.4 details a proof showing that this redefined inner product preserves the η -CTQW norm.

modified via similarity transform to make it Hermitian,

$$\tilde{U}(t) = e^{-i\tilde{H}t} = e^{-i\eta H\eta^{-1}t}. \quad (9.21)$$

This preserves the directional structure of the graph, whilst allowing us to use the standard CTQW framework. Furthermore, it has the potential to be implemented experimentally via quantum simulation.

Note that options (B) and (C) are equivalent — it is only their resulting experimental implementations which would differ. From hereon, the modified CTQW walk outlined in (B) and (C) will be referred to as the (pseudo-Hermitian) η -CTQW, to distinguish it from the standard non-unitary CTQW in (A).

At this point, a subtlety should be pointed out; the research underpinning the papers constituting this chapter (the pseudo-Hermitian CTQW centrality scheme) and the previous chapter (CTQW-based centrality measure and experimental implementation) were undertaken concurrently in collaborations with experimental groups. As a result, decisions made early in the research stage needed to be abided by, lest the theory deviated from planned experimental set-ups. In the case of chapter 8, the decision was made to choose $H = A$, due to the desire to initiate the CTQW in the equal superposition state (a stationary state of the graph Laplacian). However, when establishing the groundwork for the pseudo-Hermitian η -CTQW, the decision was instead made to choose $H = L$, the graph Laplacian with in-degree on the diagonal, for several reasons:

- When working with a PT-symmetric system, choosing to work with the graph Laplacian feels natural from a physical perspective — the PT-symmetry literature is generally concerned with *continuous-space* Hamiltonians, and the graph Laplacian provides a discrete analogue to the kinetic energy/Laplacian operator.
- This choice for the Hamiltonian would keep our work consistent with that of Salimi and Sorouri [86], even though we have modified their proposed framework.
- For the non-Hermitian directed graph Laplacian defined as in Eq. 2.14, the all ones vector is no longer an eigenvector, and thus no longer an impedance to designing a CTQW-based centrality measure.

For this thesis, modifying either chapter to ensure Hamiltonian consistency was considered, either by utilizing the adjacency matrix in this chapter, or by letting $H = L$ in the previous chapter and altering the initial CTQW state to be proportional to the square root of total vertex degree ($|\psi(0)\rangle \propto \sum_j \sqrt{d_j} |j\rangle$), as per Rossi et al. [150].

Ultimately, however, as this thesis is primarily presented as a thesis by publication, it was decided that it would be best to present each paper as published with minimal modification for consistency of formatting, notation, and terminology. We now return to the subject at hand, and provide an example of the η -CTQW framework applied to a 3-vertex directed graph.

9.4.1 3-vertex directed graph

Consider the 3-vertex graph in Fig. 9.1(c). Its Hamiltonian is found by calculating the graph Laplacian,

$$H = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix}. \quad (9.22)$$

Whilst not Hermitian ($H^\dagger \neq H$), the eigenvalues ($\lambda = 0, 2, 2$) are all real and the eigenvectors are linearly independent; thus H is PT-symmetric and pseudo-Hermitian.

Standard CTQW (non-probability conserving)

Solving the Schrödinger equation for the standard CTQW (Fig. 9.1(a)), we find that although there is no exponential growth or decay of the walker's total probability (due to the pseudo-Hermiticity), nevertheless, the norm of the state vector oscillates and is not conserved.

In order to experimentally implement this CTQW, we wish to decompose the time evolution operator such that the time dependence is restricted to a single diagonal unitary matrix. Diagonalising H yields the following decomposition:

$$U(t) = \frac{1}{2} \begin{bmatrix} 1 & -1 & -1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{-2it} & 0 \\ 0 & 0 & e^{-2it} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 2 \\ -1 & 1 & -1 \end{bmatrix}. \quad (9.23)$$

Note that whilst the diagonal matrix is unitary, the two outer matrices are non-unitary, resulting in a non-unitary time evolution operator $U(t)$.

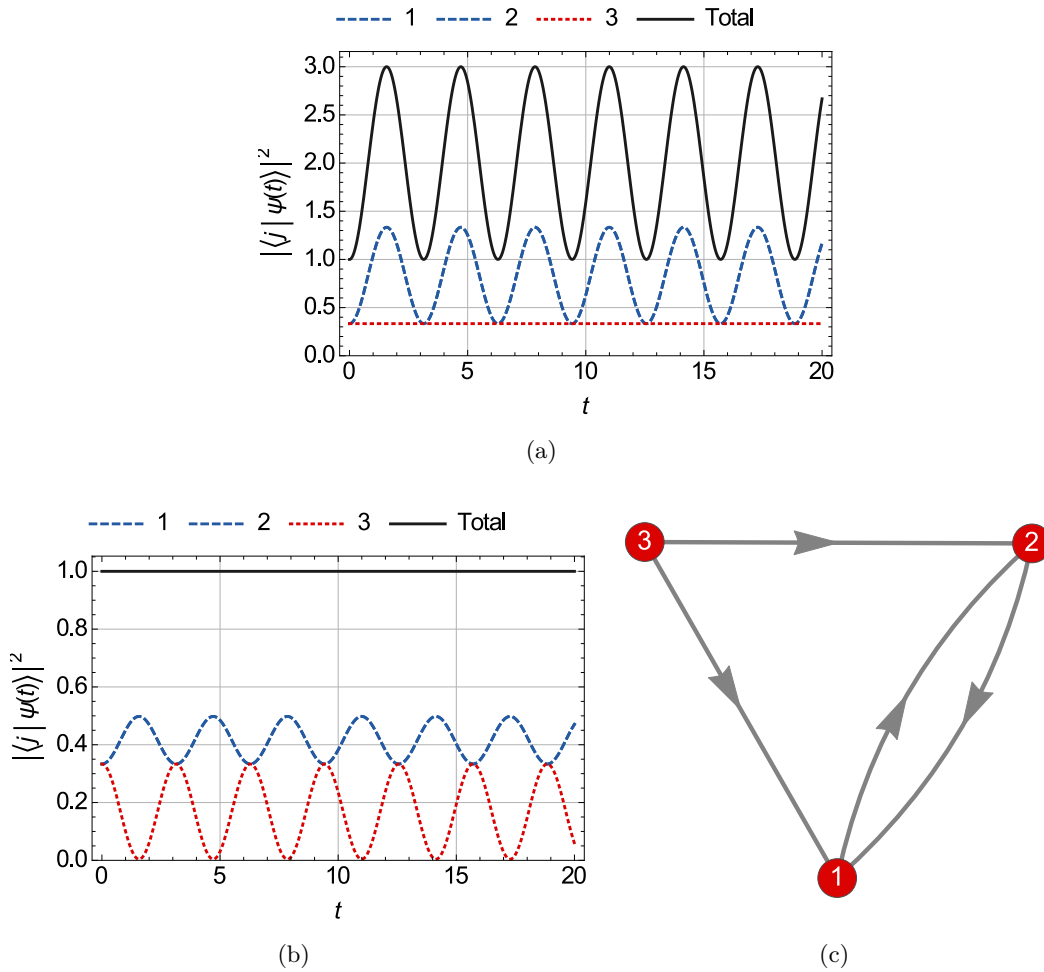


Figure 9.1: Vertex norm over time for the (a) non-unitary CTQW and (b) pseudo-Hermitian η -CTQW on the 3-vertex directed graph in (c). The walker is initially in an equal superposition of vertex states $|\psi(0)\rangle = \frac{1}{\sqrt{3}} \sum_j |j\rangle$. Vertex 1 and 2 have equal norm (blue, dashed), with a higher time-average than vertex 3 (red, dotted). The total norm of the system is given by the black, solid line.

η -CTQW (probability conserving)

By calculating the biorthonormal eigenbasis, we can derive the pseudo-Hermitian operator η of the graph,

$$\eta = \frac{1}{6} \begin{bmatrix} 3 + 2\sqrt{2} & -3 + 2\sqrt{2} & \sqrt{2} \\ -3 + 2\sqrt{2} & 3 + 2\sqrt{2} & \sqrt{2} \\ \sqrt{2} & \sqrt{2} & 5\sqrt{2} \end{bmatrix}. \quad (9.24)$$

Thus, the modified time-evolution operator is given by $\tilde{U}(t) = \eta e^{-iHt} \eta^{-1}$ — see Fig. 9.1(b) for how this affects the dynamics of the quantum walk.

In order to find a useful decomposition, we need to be able to diagonalise $\eta H \eta^{-1}$; i.e.

$$D = S^\dagger \eta H \eta^{-1} S. \quad (9.25)$$

Once S is calculated, by similarity it can also be used to diagonalise $\tilde{U}(t)$. This results in the following decomposition:

$$\tilde{U}(t) = S \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{-2it} & 0 \\ 0 & 0 & e^{-2it} \end{bmatrix} S^\dagger \quad (9.26)$$

where

$$S = \frac{1}{3} \begin{bmatrix} 2 & -\frac{3}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 2 & \frac{3}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 1 & 0 & 2\sqrt{2} \end{bmatrix}. \quad (9.27)$$

Note that, unlike the non-probability conserving case, here all three of the decomposed matrices are unitary.

9.4.2 Alternative interpretation

To get an understanding of the transformation from the non-probability conserving CTQW to the pseudo-Hermitian η -CTQW, let's have a look at the pseudo-Hermitian Hamiltonian

for the 3-vertex graph discussed above,

$$\tilde{H} = \eta H \eta^{-1} = \frac{1}{9} \begin{bmatrix} 10 & -8 & -4 \\ -8 & 10 & -4 \\ -4 & -4 & 16 \end{bmatrix}. \quad (9.28)$$

Describing a discrete graph structure, the η -Hamiltonian \tilde{H} is symmetric, yet contains fractional quantities — we may interpret \tilde{H} as the Laplacian of a weighted, undirected complete graph with self-loops. (Note that this is just one valid interpretation, and is not unique). In doing so, it is convenient to use a more rigorous definition of the Laplacian more suited to weighted graphs.

The oriented incidence matrix M of an undirected weighted graph $G(V, E, W)$, with vertex set $V = \{v_1, v_2, \dots, v_N\}$, edge set $E = \{e_1, e_2, \dots, e_m\}$, and edge weights $W = \{w_1, w_2, \dots, w_m\}$, is the $n \times m$ matrix associated with a particular orientation of the edges of G , and was defined earlier in Eq. 2.12. Recall that, given the oriented incidence matrix of a weighted graph, we can compute the $N \times N$ *weighted Laplacian* using the relationship

$$L = MWM^T, \quad (9.29)$$

where W is a diagonal matrix containing the weights $\{w(v_i, v_j)\}$ associated with the edges $\{(v_i, v_j)\}$. For a complete graph over N vertices with self-loops and arbitrary edge weighting, it turns out we can compute the weighted Laplacian directly,

$$L_{ij} = \begin{cases} \sum_{k=1}^N w_{ik} + 3w_{ii}, & i = j \\ -w_{ij}, & i \neq j \end{cases} \quad (9.30)$$

where $w(v_i, v_j) = w(v_j, v_i)$ and the size of the set W is $|E| = \frac{1}{2}N(N+1)$. Thus, if we have an $N \times N$ pseudo-Hamiltonian \tilde{H} , then solving $\tilde{H} = L_{ij}$ provides the edge weighting

$$w(v_i, v_j) = \begin{cases} \frac{1}{4} \sum_{k=1}^N \tilde{H}_{ik}, & i = j \\ -\tilde{H}_{ij}, & i \neq j. \end{cases} \quad (9.31)$$

That is, we can interpret the η -CTQW of a *directed* N -vertex graph in terms of a standard CTQW on a *undirected* complete graph with self-loops, and edge weights given above. The undirected edge weightings allows us to approximate the directed dynamics we require.

For example, let's return to the 3-vertex graph examined previously. Using the pseudo-Hamiltonian of the graph (Eq. 9.28), we can find the edge weights W via Eq. 9.31. See Fig. 9.2 for the result.

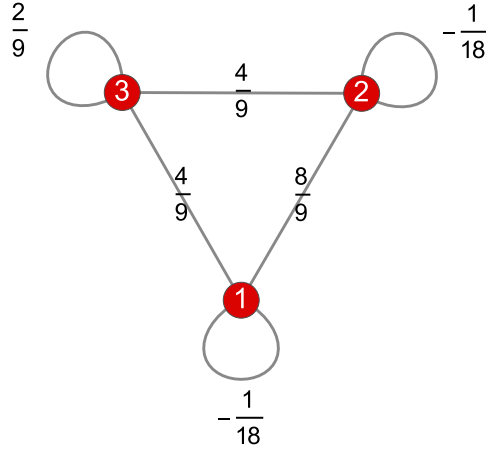


Figure 9.2: Performing an η -CTQW on the 3-vertex directed graph in Fig. 9.1(c) is equivalent to performing a standard CTQW on the undirected edge-weighted graph shown above.

9.4.3 Multi-particle η -CTQW

If we wish to extend the standard CTQW to simulate P distinguishable particles on graph G , the Hamiltonian of the system is expanded to act on a N^P Hilbert space, as follows:

$$H^{(P)} = H_1 \oplus H_2 \oplus \cdots \oplus H_P + \Gamma \quad (9.32)$$

where H_j is the free-particle Hamiltonian of the j th particle on graph G , Γ represents a potential interaction between the particles, and \oplus is the tensor or Kronecker sum defined by

$$A_{n \times n} \oplus B_{m \times m} = A_{n \times n} \otimes I_{m \times m} + I_{n \times n} \otimes B_{m \times m}. \quad (9.33)$$

Note that the free particle Hamiltonians are identical ($H_1 = H_2 = \cdots = H_P$), allowing this to be rewritten as

$$H^{(P)} = H^{\oplus P} + \Gamma. \quad (9.34)$$

In the case of no-interaction ($\Gamma = 0$), then it is trivial to show that if H is pseudo-Hermitian, then so is $H^{(P)}$, with pseudo-Hermitian operator $\eta^{(P)}$ given by

$$\eta^{(P)} = \underbrace{\eta \otimes \eta \otimes \cdots \otimes \eta}_P = \eta^{\otimes P}, \quad (9.35)$$

and η the pseudo-Hermitian operator of the single particle Hamiltonian H .

When $\Gamma \neq 0$, then additional care must be taken — unlike Hermitian matrices, the pseudo-Hermitian matrices are *not* closed under addition, so $H^{(P)}$ is no longer guaranteed to be pseudo-Hermitian (this can easily be shown by counter-example).

9.4.4 Interdependent networks

Aside from the Cartesian product of graphs discussed above, another method of combining graph structures are interdependent networks. In the real-world, very few networks operate independently, instead interacting and depending on a myriad of other networks [192, 193] — examples include modelling cascading failures between power grids, communication networks, and physiological/biochemical systems. As such, being able to extend the pseudo-Hermitian CTQW to model interdependent networks greatly expands the scope of the framework. In this section, we will consider an interconnected network of two pseudo-Hermitian graphs and determine the properties that must be satisfied for the resulting network to have guaranteed pseudo-Hermiticity.

Let A_1 and A_2 refer to the adjacency matrices of two graphs, G_1 and G_2 , respectively. The resulting interdependent network Hamiltonian is constructed as follows:

$$A = \begin{bmatrix} A_1 & B_0 \\ B_0^T & A_2 \end{bmatrix} \quad (9.36)$$

where B_0 is the adjacency matrix representing the edges connecting the two sets of vertices $V(A_1)$ and $V(A_2)$. The Hamiltonian of this graph, defined by Eq. 2.14, can then be written as

$$H = \begin{bmatrix} \mathcal{H}_1 & -B_0 \\ -B_0^T & \mathcal{H}_2 \end{bmatrix} \quad (9.37)$$

where

$$(\mathcal{H}_u)_{ij} = (H_u)_{ij} + \sum_k (B_0)_{ik} \delta_{ij}, \quad u \in \{1, 2\}, \quad (9.38)$$

and H_1 and H_2 are the Hamiltonians of A_1 and A_2 respectively.

Theorem 9.1. *The Hamiltonian of an interdependent network of two graphs A_1 and A_2 , with pseudo-Hermitian Hamiltonians H_1 and H_2 respectively, will itself exhibit pseudo-Hermiticity if the inter-network connections B_0 is pseudo-Hermitian and degree-regular, and the commutation relations $H_1 B_0 = B_0 H_2$ and $H_2 B_0^T = B_0^T H_1$ are satisfied.*

Proof. B_0 is degree regular with degree c — thus

$$\sum_k (B_0)_{ik} \delta_{ij} = c \delta_{ij} \Rightarrow \mathcal{H}_u = H_u + cI. \quad (9.39)$$

If we decompose the Hamiltonian into the sum

$$H = \begin{bmatrix} \mathcal{H}_1 & 0 \\ 0 & \mathcal{H}_2 \end{bmatrix} + \begin{bmatrix} 0 & -B_0 \\ -B_0^T & 0 \end{bmatrix} = A + B, \quad (9.40)$$

we can now prove that the interdependent network Hamiltonian is always pseudo-Hermitian if the two components A and B commute and are themselves pseudo-Hermitian.

Since H_1 and H_2 are pseudo-Hermitian, they are therefore diagonalisable by matrices Q_u :

$$\Lambda_u = Q_u^{-1} H_u Q_u, \quad (9.41)$$

where Λ_u are real diagonal matrices of eigenvalues. It follows that

$$Q_u^{-1} \mathcal{H}_u Q_u = Q_u^{-1} (H_u + cI) Q_u = \Lambda_u + cI. \quad (9.42)$$

That is, \mathcal{H}_1 and \mathcal{H}_2 are simultaneously diagonalisable and exhibit real eigenspectra — satisfying the criteria for pseudo-Hermiticity. From here, it is trivial to show that A is diagonalised as follows:

$$\Lambda_A = \begin{bmatrix} \Lambda_1 + cI & 0 \\ 0 & \Lambda_2 + cI \end{bmatrix} = Q_A^{-1} A Q_A \quad (9.43)$$

where

$$Q_A = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \quad (9.44)$$

and Λ_A is real.

As B_0 is also pseudo-Hermitian, therefore it is also diagonalisable with real eigenvalues; $\Lambda_{B_0} = Q_B^{-1} B_0 Q_B$. Similarly to above, we can use this result to diagonalise the matrix B :

$$\Lambda_B = \begin{bmatrix} \Lambda_{B_0} & 0 \\ 0 & -\Lambda_{B_0} \end{bmatrix} = Q_B^T B Q_B \quad (9.45)$$

where

$$Q_B = \frac{1}{\sqrt{2}} \begin{bmatrix} Q_{B_0} & Q_{B_0} \\ Q_{B_0} & -Q_{B_0} \end{bmatrix} \quad (9.46)$$

and Λ_B is real.

It follows from the above analysis that if H_1 and H_2 are pseudo-Hermitian, and the interconnections adjacency matrix B_0 is pseudo-Hermitian and degree regular, that the matrices A and B are also pseudo-Hermitian.

To ensure that the sum $A + B$ remains pseudo-Hermitian, we can make use of the well-known property that commuting diagonalisable matrices are simultaneously diagonalisable [194]. This requires the additional constraint, $[A, B] = 0$, resulting in the following two conditions that must be satisfied:

$$H_1 B_0 = B_0 H_2, \quad (9.47a)$$

$$H_2 B_0^T = B_0^T H_1. \quad (9.47b)$$

If these are both satisfied, then A and B must be simultaneously diagonalised by a matrix S :

$$H = A + B = S(P_A \Lambda_A P_A^T + P_B \Lambda_B P_B^T) S^{-1}, \quad (9.48)$$

where P_A and P_B are permutation matrices. From here, we can see that the eigenvalues of H are contained in the set of elements $\{\lambda_1 \pm \lambda_{B_0} + 1\} \cup \{\lambda_2 \pm \lambda_{B_0} + 1\}$ and thus are necessarily real. Therefore, the interdependent network of pseudo-Hermitian graphs — which commute with their interconnections — also exhibits pseudo-Hermiticity. \square

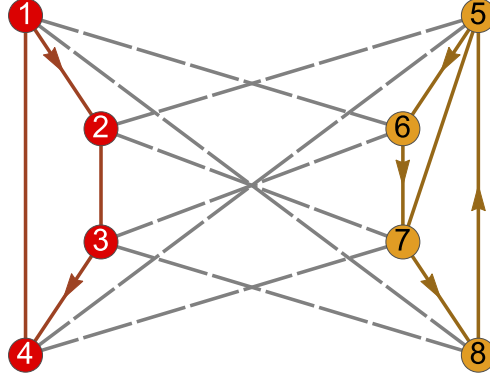


Figure 9.3: A pseudo-Hermitian interconnected network composed of two 4-vertex pseudo-Hermitian graphs (red and yellow respectively) connected via undirected circulant interconnections (gray, dashed).

An example of a pseudo-Hermitian interconnected network in which the two graph Hamiltonians commute with the interconnections is given in Fig. 9.3. Note that the above theorem is not a necessary condition for interdependent network pseudo-Hermiticity — examples can be constructed where $[A, B] \neq 0$, yet $A + B$ remains diagonalisable and pseudo-Hermitian. Nevertheless, this results provides a useful method of constructing interdependent networks with guaranteed pseudo-Hermiticity.

9.5 Centrality testing

In this section, we propose a quantum centrality measure based on the pseudo-Hermitian η -CTQW, which has the advantage of a purely quantum propagation and smaller statespace than both the Quantum PageRank and QSW, and extends the centrality measure of chapter 8 to PT-symmetric directed graphs.

9.5.1 Proposed quantum scheme and examples

Proceeding as per Sec. 8.2, we propose the following measure for assigning a centrality value to vertex j on a directed graph represented by the pseudo-Hermitian Hamiltonian H :

$$v_j = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \left| \langle j | e^{-i\eta H \eta^{-1} t} \left(\frac{1}{\sqrt{N}} \sum_k |k\rangle \right) \right|^2 dt. \quad (9.49)$$

Note that this method simply requires a Hilbert space of dimension N . This scheme will then be applied to the 3-vertex graph discussed previously.

Firstly, consider the non-norm conserving standard CTQW walk in the previous section. As the pseudo-Hermitian Hamiltonian preserves the graph eigenspectra, we expect the resulting dynamics to reflect properties that can be classically extracted from the spectra, such as vertex centrality. Thus, one potential way of extracting this information is simply to calculate the time-average of the probability on each vertex, and normalise it by the total probability time-average.

Solving the Schrödinger equation for the unmodified Hamiltonian evolving the initial state $|\psi(0)\rangle = \frac{1}{\sqrt{3}}(|1\rangle + |2\rangle + |3\rangle)$, we get the exact solution

$$|\psi(t)\rangle = \frac{1}{\sqrt{6}}\sqrt{5 - 3\cos(2t)}(|1\rangle + |2\rangle) + \frac{1}{\sqrt{3}}|3\rangle, \quad (9.50)$$

where

$$\langle\psi(t)|\psi(t)\rangle = 2 - \cos(2t). \quad (9.51)$$

The norm of the state vector has a time average over $t = [0, \pi]$ (one period) of

$$\int_0^\pi \langle\psi(t)|\psi(t)\rangle dt = 2\pi, \quad (9.52)$$

therefore, calculating the time average of this result over $t = [0, \pi]$:

$$\frac{1}{2\pi} \int_0^\pi |\langle j|\psi(t)\rangle|^2 dt = \frac{5}{12}\delta_{j1} + \frac{5}{12}\delta_{j2} + \frac{1}{6}\delta_{j3}. \quad (9.53)$$

That is, by this centrality measure, vertices 1 and 2 are ranked equal first, followed by vertex 3.

Next, consider the probability conserving η -CTQW case, $\tilde{U}(t) = \eta e^{-iHt}\eta^{-1}$. As we have an exact representation of η , we are also able to solve for $|\tilde{\psi}(t)\rangle$ exactly:

$$|\tilde{\psi}(t)\rangle = \frac{1}{\sqrt{243}}\sqrt{101 - 20\cos(2t)}(|1\rangle + |2\rangle) + \frac{1}{\sqrt{243}}\sqrt{41 + 40\cos(2t)}|3\rangle. \quad (9.54)$$

Calculating the time average,

$$\frac{1}{\pi} \int_0^\pi |\langle j|\tilde{\psi}(t)\rangle|^2 dt = \frac{101}{243}\delta_{j1} + \frac{101}{243}\delta_{j2} + \frac{41}{243}\delta_{j3}. \quad (9.55)$$

	Eigenvector	PageRank	CTQW	η -CTQW
1	0.5	0.475	0.416667	0.415638
2	0.5	0.475	0.416667	0.415638
3	0	0.05	0.166667	0.168724

Table 9.1: Centrality ranking of the vertices of 3-vertex graph Fig. 9.1(c), using the classical PageRank method, the non-norm conserving pseudo-Hermitian CTQW, and the probability conserving pseudo-Hermitian CTQW (η -CTQW)

	Eigenvector	PageRank	CTQW	η -CTQW
1	0.292893	0.25	0.386364	0.339192
2	0.207107	0.25	0.113636	0.160808
3	0.292893	0.25	0.386364	0.339192
4	0.207107	0.25	0.113636	0.160808

Table 9.2: Centrality ranking of the 4 vertices of graph Fig. 9.4, using the classical PageRank method, the non-probability conserving pseudo-Hermitian CTQW, and the probability conserving pseudo-Hermitian CTQW (η -CTQW)

The numerical values of the CTQW and η -CTQW centrality rankings have been tabulated in Tab. 9.1, alongside the classical PageRank ($\alpha = 0.85$) and eigenvector centrality rankings. Note that the two CTQW rankings strongly agree with the classical PageRank and eigenvector rank. Furthermore, the numerical values of the two CTQW rankings only differ by a maximum of about 1.23%, indicating that the pseudo-Hermitian similarity transform preserves information regarding vertex centrality in this particular example. A general statistical analysis will be carried out in the subsequent section.

Let's now consider a 4-vertex pseudo-Hermitian directed graph, as shown in Fig. 9.4. Note that this graph is composed of two sets of equivalent vertices; vertices 1 and 3 (in-degree

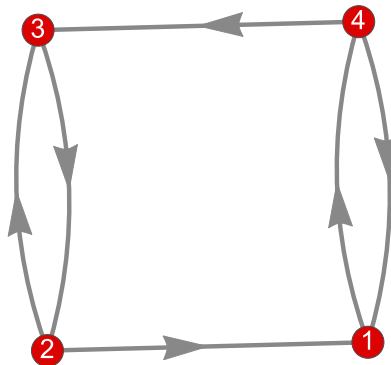


Figure 9.4: 4-vertex directed graph

= 2 and out-degree = 1), and vertices 2 and 4 (in-degree = 1 and out-degree = 2). Like the 3-vertex graph analysed previously, the CTQW probability can also be solved exactly, and thus the same method is applied to determine the vertex centrality.

The results of the CTQW and η -CTQW centrality test can be seen in Tab. 9.2. In this case, both CTQW formulations and the eigenvector centrality ranked vertices 1 and 3 above vertices 2 and 4. Intuitively, this is perhaps expected, as vertices 1 and 3 have a greater in-degree and lower out-degree than vertices 2 and 4, resulting in walker probability accumulating on these two vertices. Interestingly, the classical PageRank measure does not distinguish between these two sets of vertices, as the limiting distribution of a (classical) random walk on this graph results in a unitary distribution. The agreement of the CTQW-based measures to the eigenvector centrality, as opposed to the PageRank, lends further credence to the suggestion that the CTQW measures centrality via a similar process to the eigenvector centrality.

As a final example, we briefly examined a pseudo-Hermitian interdependent network consisting of a 4-vertex directed graph and a 3-vertex directed graph, connected via complete interconnections (i.e. $B_0 = J$). The graph and various centrality rankings of the vertices are shown in Fig. 9.5 — it can be seen that the pseudo-Hermitian η -CTQW centrality ranking strongly agrees with the classical PageRank and eigenvector results, with the only disagreement involving the rankings of vertices 3 and 6, as well as 1 and 4 (both display degeneracy in the classical measures). Of note, the η -CTQW (along with the PageRank and eigenvector centrality) breaks many of the vertex rank degeneracies seen in the standard non-unitary CTQW; perhaps indicating that the pseudo-Hermitian CTQW — itself a mapping of a directed graph to an undirected, yet weighted, complete graph — provides a better overall picture of the vertex ranks. Interestingly, the η -CTQW is the only ranking to break the top-ranked tie seen in the other measures, assigning slightly more importance to vertex 3 compared to vertex 6.

Investigating the correlation of the η -CTQW centrality measure to various classical measures, we once again employ Vigna's τ rank correlation coefficient (Def. 8.4), as described in Sec. 8.3. The results of this analysis can be seen in Tab. 9.3. All compared centralities display very strong correlation ($\tau \geq 0.8$), with the exception of the CTQW and η -CTQW, with a correlation value of $\tau = 0.761$ (still indicative of a strong rank correlation, $0.6 \leq \tau < 0.8$). This is most likely due to the degeneracy seen in the CTQW ranking, which is completely broken in the η -CTQW; these resulting ties slightly lower the τ value, even though they do not cause disagreeing rankings per se. Finally, note that the η -CTQW achieves its highest correlation value with the eigenvector centrality at

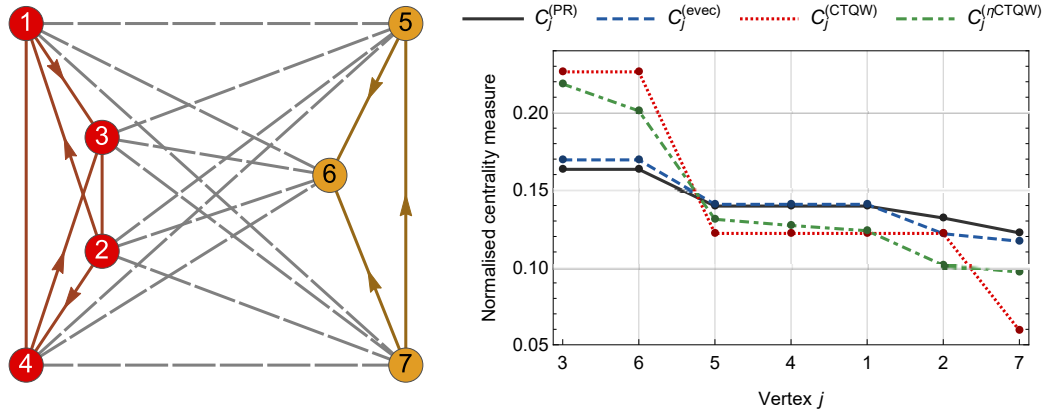


Figure 9.5: **Left:** Interdependent network, consisting of a directed 4-vertex graph (red) connected to a directed 3-vertex graph (yellow) via complete interconnections (gray, dashed). **Right:** Centrality ranking of the vertices, ordered from highest ranking to lowest ranking vertex. Measures used include the classical PageRank (black, solid), the eigenvector centrality (blue, dashed), the standard CTQW (red, dotted) and the pseudo-Hermitian η -CTQW (green, dot-dashed).

	PageRank	Eigenvector	CTQW	η -CTQW
PageRank	1.	0.912	0.937	0.896
Eigenvector	0.912	1.	0.896	0.906
CTQW	0.937	0.896	1.	0.761
η -CTQW	0.896	0.906	0.761	1.

Table 9.3: Vigna's τ rank correlation coefficient compared for various classical (PageRank and eigenvector) and quantum (non-unitary CTQW and pseudo-Hermitian η -CTQW) centrality measures applied to the 7-vertex interdependent network in Fig. 9.5.

$\tau = 0.906$, edging out correlation with the PageRank at $\tau = 0.896$, further suggesting that the η -CTQW assigns vertex centrality values in a method similar to classical eigenvector centrality.

In these three examples, we have seen that the pseudo-Hermitian CTQW preserves the vertex centrality information from the original directed graphs, resulting in a vertex rank identical (barring broken degeneracy) to the classical PageRank. Whilst these relatively small examples allow us to verify the results of the centrality ranking by intuitively and qualitatively examining the graph structures by eye, this analysis is not sufficient to ensure that the centrality ranking proposed here generalises to other PT-symmetric graph structures. To do so, a statistical analysis featuring randomly generated directed graphs is required.

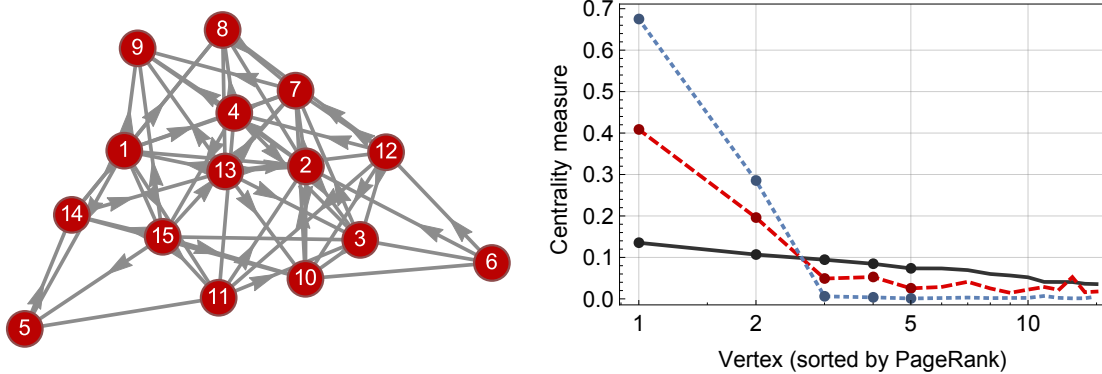


Figure 9.6: **Left:** Randomly generated pseudo-Hermitian Erdős-Rényi graph $G(15, 0.3)$, with bidirectional edges allowed. **Right:** Centrality ranking of the vertices, ordered from most central to least central as per the classical PageRank (black). This is compared to the non-unitary CTQW (blue, dotted) and pseudo-Hermitian η -CTQW (red, dashed).

9.5.2 Random directed networks

To investigate the reliability of the pseudo-Hermitian CTQW on directed graphs, a statistical analysis will be undertaken using randomly generated directed networks, in a similar fashion to Sec. 8.3. Again, we consider two classes of random networks — Erdős-Rényi networks, and scale-free networks.

In order to produce a PT-symmetric directed graph satisfying the Erdős-Rényi degree distribution, we take two approaches. Firstly, we generate numerous directed Erdős-Rényi graphs (with parameters $N = 15$, $p = 0.3$) using the Python software package NetworkX [195], and selecting from these 300 which satisfy pseudo-Hermiticity. An example is presented in Fig. 9.6, alongside a plot of the PageRank, non-unitary CTQW, and η -CTQW centrality measures for the pictured example. In this particular example, all three measures agree on the location of the top two ranked vertices, with slight discrepancies for the remaining vertices. Note that, from here onward, the eigenvector centrality is no longer included as a comparison, as we can no longer guarantee its performance — a majority of the graphs in this and subsequent ensembles contain acyclic and non-strongly connected components that result in an eigenvector centrality value of zero.

Our second approach to generating pseudo-Hermitian directed Erdős-Rényi networks was motivated by computational constraints with using NetworkX, and a desire to generate larger pseudo-Hermitian graphs in a slightly more systematic way. Here, we first create an undirected graph of N vertices with edges given by Bernoulli distribution with probability p . We then upper triangularize the resulting adjacency matrix, by setting everything below

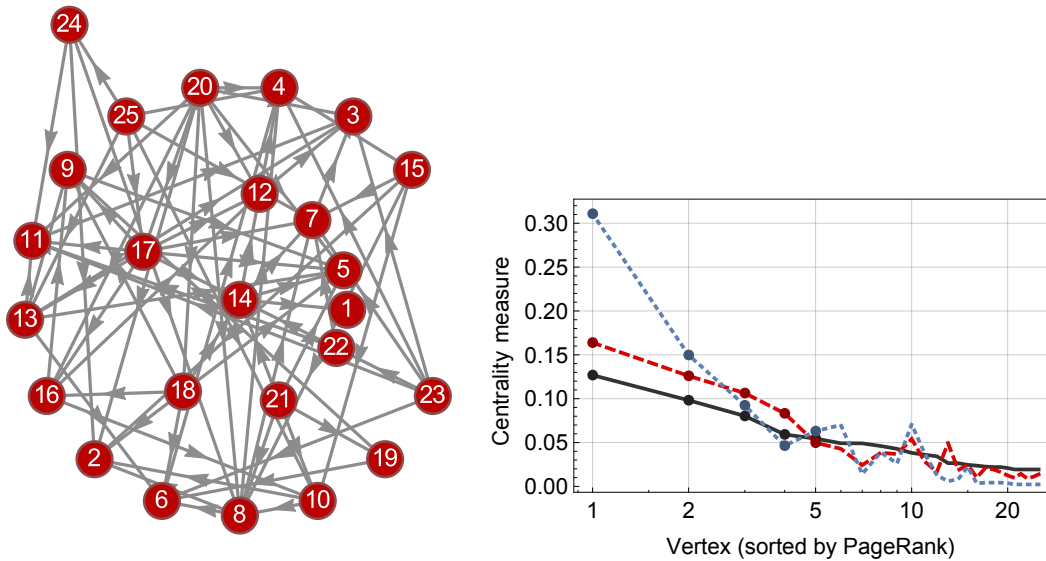


Figure 9.7: **Left:** Randomly generated directed pseudo-Hermitian Erdős-Rényi graph $G(25, 0.3)$, where every edge is directed. **Right:** Centrality ranking of the vertices, ordered from most central to least central as per the classical PageRank (black). This is compared to the non-unitary CTQW (blue, dotted) and pseudo-Hermitian η -CTQW (red, dashed).

the diagonal to zero; in effect, imbuing direction on every edge in a systematic fashion. By restricting the adjacency matrix to be triangular, it is trivial to see that the Hamiltonian will also be triangular — with eigenvalues given by the diagonal elements of H , the set of vertex in-degrees:

$$\lambda = \{\text{deg}^-(v_i) \mid i = 1, \dots, N\} \quad (9.56)$$

where $\text{deg}^-(v_i)$ is a function returning the in-degree of vertex v_i . As such, we ensure a real eigenspectrum, and simply restrict our random graph generator to output graphs with diagonalisable Hamiltonians in order to guarantee pseudo-Hermiticity.

An example of a randomly generated pseudo-Hermitian directed Erdős-Rényi graph using this method is shown in Fig. 9.7, generated with parameters $N = 25$ and $p = 0.3$, alongside the results of the PageRank and η -CTQW centrality measures. It can be seen that the η -CTQW and the PageRank strongly agree on the relative vertex rankings — identically ranking the top four most central vertices, and satisfying the same general trend thereafter. This indicates that the η -CTQW continues to yield an admissible vertex centrality measure for larger, randomly generated graphs than in the previous section.

To generate random pseudo-Hermitian scale-free graphs, we make use of the directed Barabási-Albert algorithm: at each time-step, a vertex with m directed edges is introduced

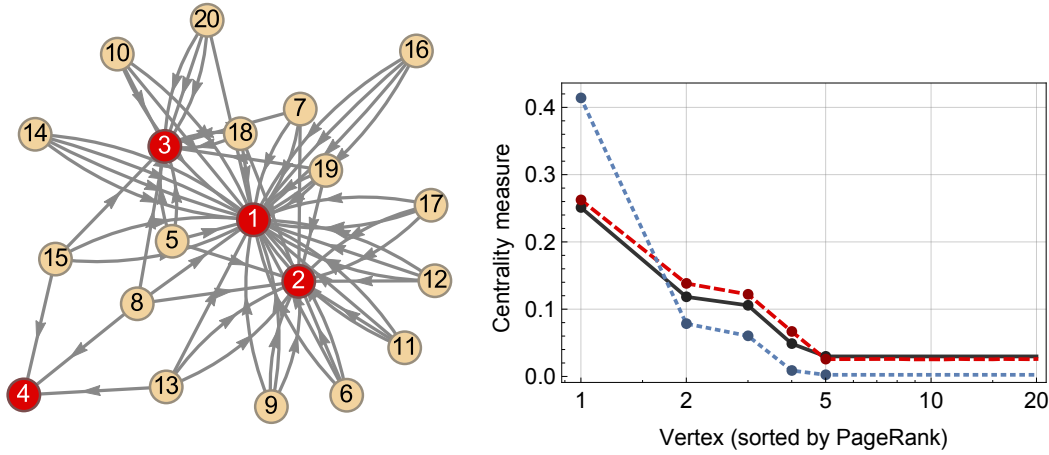


Figure 9.8: **Left:** Randomly generated 20 vertex pseudo-Hermitian graph, with scale-free in-degree distribution where $m = 4$. The more highly connected ‘hubs’ (vertices with higher in-degree) are labelled in red. **Right:** Centrality ranking of the vertices, ordered from most central to least central as per the classical PageRank (black). This is compared to the non-unitary CTQW (blue, dotted) and the pseudo-Hermitian η -CTQW (red, dashed).

to the system, and preferentially attached to existing vertices with higher degrees (with probability of being connected to vertex i given by $p_i = k_i / \sum_j k_j$). This process continues until we have a graph containing the required number of vertices. Note that the directed Barabási-Albert algorithm only produces edges with a unidirectional flow of information [97] — if chosen such that the in-degree distribution is scale-free, then the out-degree distribution will be of the form δ_{mk} , and vice versa.

However, as a result, a fortunate side effect of the directed Barabási-Albert algorithm is that if we choose all m edges introduced with each additional vertex to be *inward*-pointing edges (resulting in $\deg^-(v_i) = m \forall i$), then the graph is necessarily lower triangular, leading to a Hamiltonian with real eigenspectrum as given by Eq. 9.56. Similarly, if we choose all m edges introduced with each vertex to be *outward*-pointing edges (resulting in $\deg^+(v_i) = m \forall i$), the Hamiltonian will be upper-triangular and Eq. 9.56 continues to hold. Thus, like the Erdős-Rényi case described previously, to ensure pseudo-Hermiticity we simply ensure the resulting randomly generated scale-free Hamiltonian is diagonalizable.

Fig. 9.8 shows a pseudo-Hermitian directed graph constructed via the Barabási-Albert algorithm with parameters $N = 25$, $m = 4$, such that the in-degree vertex distribution is scale-free. By examining the classical PageRank and η -CTQW centrality measures, we see that they provide identical rankings for all 20 vertices, correctly picking out and ordering

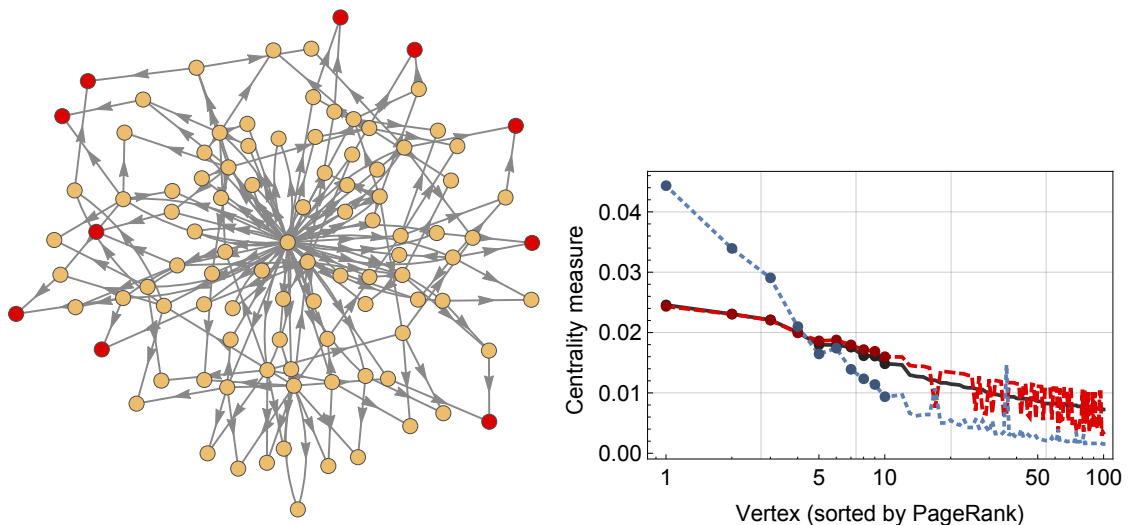


Figure 9.9: **Left:** Randomly generated 100 vertex pseudo-Hermitian graph, with scale-free out-degree distribution where $m = 3$. The top 10 vertices where network flow is likely to accumulate are labelled in red. **Right:** Centrality ranking of the vertices, ordered from most central to least central as per the classical PageRank (black). This is compared to the non-unitary CTQW (blue, dotted) and the pseudo-Hermitian η -CTQW (red, dashed).

the four ‘hubs’ (marked in red) with larger in-degree. Meanwhile, in Fig. 9.9 we have a pseudo-Hermitian directed graph constructed via the Barabási-Albert algorithm with parameters $N = 100$, $m = 3$, such that the *out-degree* is scale-free, and constant in-degree of 3. Again, the PageRank and η -CTQW display a high correlation, with the ranking of the top four most central vertices identical. However, in this case a subtlety must be addressed — the PageRank algorithm is known to correlate with in-degree [148, 196], as is the η -CTQW scheme by construction of the Hamiltonian in Eq. 2.14. Hence, rather than assigning higher measures to vertices that are out-degree ‘hubs’, both algorithms are preferentially selecting top-ranked vertices based on in-degree distribution. These correspond to the vertices at which the probability flow of a random walk is likely to accumulate after significant time.

9.5.3 Statistical analysis

So far, we have considered particular Erdős-Rényi and scale-free randomly generated graphs — to ensure that the pseudo-Hermitian CTQW centrality scheme remains valid in general, it is pertinent to undertake a statistical analysis of an ensemble of random graphs. Ensembles of 300 random Erdős-Rényi ($N = 25$, $p = 0.3$, bidirectional edges permitted), 100 random Erdős-Rényi ($N = 25$, $p = 0.3$, bidirectional edges not permitted),

100 random in-degree scale-free ($N = 20$, $m = 3$), and 100 random out-degree scale-free ($N = 40$, $m = 3$) were generated, and the PageRank, non-unitary CTQW, and η -CTQW vertex ranking determined for each graph. The mean and standard deviation of these centralities are plotted in Fig. 9.10 for each ensemble, with the η -CTQW compared to both the non-unitary CTQW and classical PageRank. Furthermore, Vigna's τ rank correlation coefficient has been averaged across the ensemble, and is displayed on each plot.

Studying the results of Fig. 9.10, we may draw several conclusions. Firstly, the η -CTQW continues to reflect the directed structure of the network, agreeing with the non-unitary CTQW across all ensembles on the top 5 ranked vertices. This agreement is similarly reflected in Vigna's τ correlation coefficient, with $\tau \gtrsim 0.6$ for every ensemble — with lower values perhaps due to small discrepancies for lower ranked vertices.

Still, this statistical analysis has its drawbacks. The shaded areas, representing one standard deviation from the mean centrality values, indicate general ranking agreement across an ensemble *only* when narrow enough and with a steep enough gradient such that each consecutive point, when moved upward/downward by one standard deviation, does not cause a swap in ranking (e.g. Fig. 9.10(c)). Further, the converse is not true — a large standard deviation does not imply a lack of agreement in ranking. In fact, two centrality measures could produce the exact same ranking across an entire ensemble, yet one measure might simply have a greater variance in the values it assigns to the vertices. Finally, we note as in Sec. 8.3.2 that the Jaccard measure of set similarity [162] provides a better understanding of how frequently two centrality measures agree on the k top-most ranked vertices, compared to Vigna's τ , which is better utilized as a correlation measure.

Thus, we undertake our analysis in a similar manner to that undertaken previously when considering undirected graphs. Firstly, for each graph, unordered sets containing the n most central vertices according to each measure were compared — the fraction of matching vertices providing a quantitative value for the agreement between the two measures. Finally, these were averaged over the entire ensemble, providing a general measure of the agreement between the PageRank and the η -CTQW, with uncertainty approximated by calculating the Agresti-Coull 95% confidence interval [163].

The results of the statistical analysis are presented in Fig. 9.11. When considering just the most central vertex, the PageRank and η -CTQW are in excellent agreement in the case of the unidirectional Erdős-Rényi and scale-free ensembles, ranging from 95% to 100% agreement. As the number of vertices compared increases, there is a small decrease in the Jaccard set similarity, with all three ensembles of random graphs exhibiting agreement factors in the range of 90% for the top two and three most central vertices. By the time

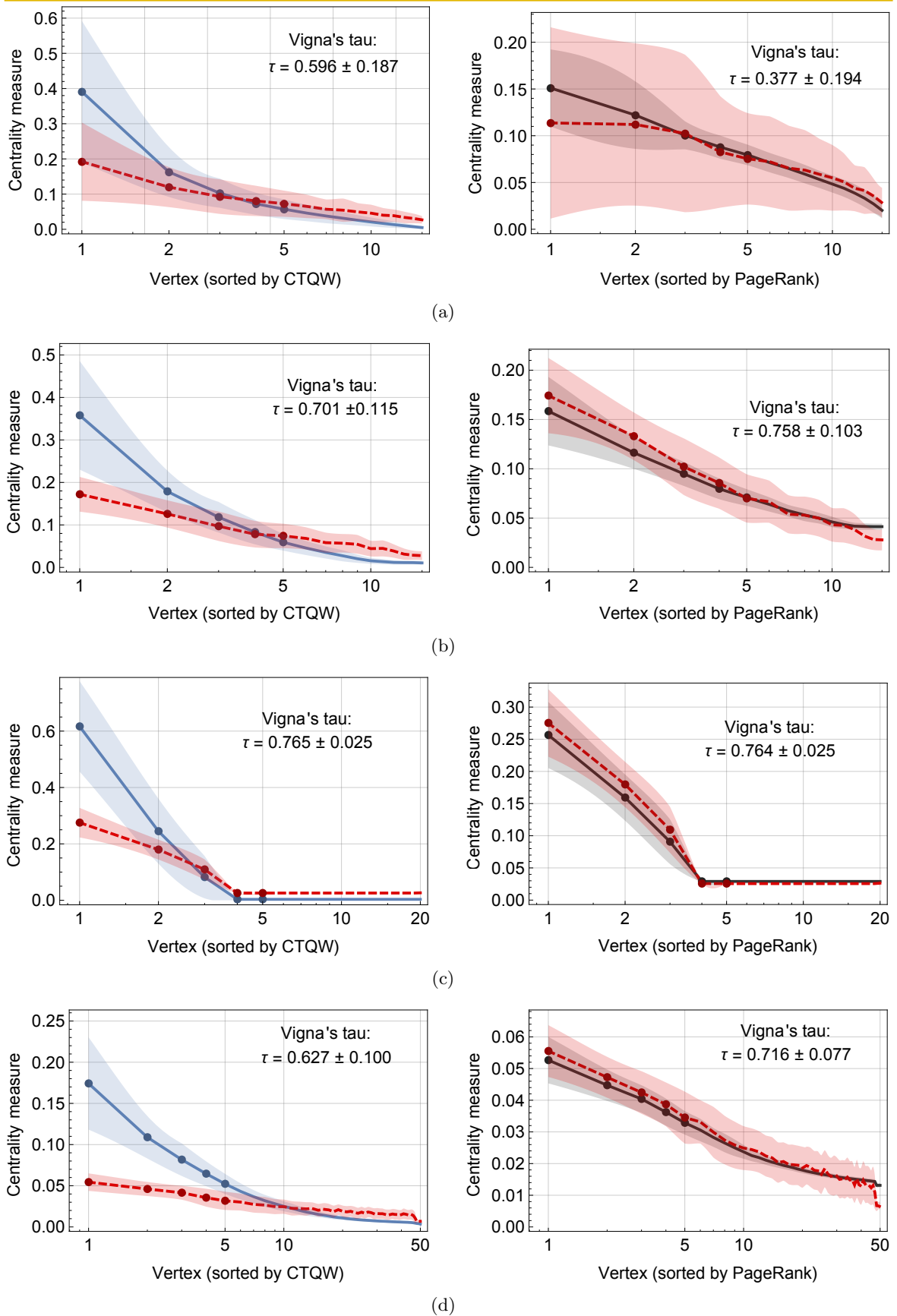


Figure 9.10: Centrality measure values for the η -CTQW (red, dashed), compared against the non-unitary CTQW (blue), and PageRank (black), averaged over an ensemble of (a) 300 Erdős-Rényi graphs with bidirectional edges allowed, (b) 100 Erdős-Rényi graphs with unidirectional edges, (c) 100 in-degree scale-free graphs, and (d) 100 out-degree scale-free graphs. The shaded areas represent one standard deviation from the mean.

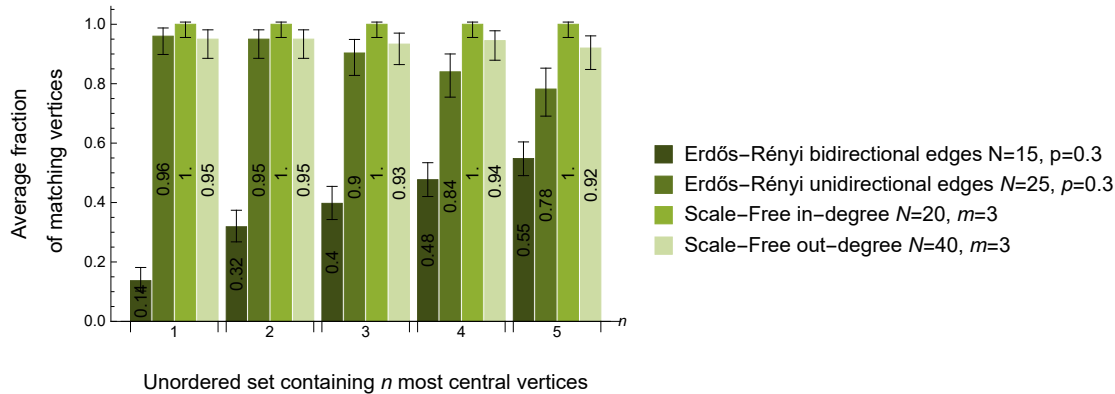


Figure 9.11: Chart showing the Jaccard set similarity between the classical PageRank algorithm and the η -CTQW centrality scheme for an ensemble of 300 directed Erdős-Rényi graphs with bidirectional edges permitted, 100 directed Erdős-Rényi graphs with unidirectional edges, 100 directed in-degree scale-free graphs, and 100 directed out-degree scale-free graphs. Each bar represents the unordered set containing the n most central vertices as determined by the PageRank and η -CTQW scheme, whilst the vertical axis gives the average fraction of matching vertices between these two sets. The error bars indicate the Agresti-Coull 95% confidence interval.

we consider five vertices, scale-free networks retain an excellent agreement of 100% and 92%, whilst the unidirectional Erdős-Rényi ensemble exhibits a reasonably good agreement factor of 78%. These trends can be partially explained by considering the behaviour of the degree distributions:

- Erdős-Rényi networks, with a majority of vertices having degree close to the mean, generally results in the highest ranked vertices having similar centrality measures. As such, beyond the top three, small variations in the PageRank and η -CTQW vertex ordering appear, leading to discrepancies.
- In-degree scale-free networks, with a small number of highly connected vertices, should easily distinguish these vertices (the ‘hubs’) as most central to the network. Beyond the hubs, the power law characteristic results in the majority of remaining vertices having similar degree — leading to small variations in vertex ordering, and thus the discrepancies observed between the PageRank and η -CTQW as more vertices are compared.

The Erdős-Rényi ensemble with bidirectionality of edges permitted is the one outlier; in Fig. 9.11, it can be seen that the Jaccard set similarity between the PageRank and η -CTQW when considering just the top-most ranked vertex is only 13%. This could be due to a multitude of factors:

- the PageRank might provide a significantly different rank to other classical measures

(such as the eigenvector centrality), which the η -CTQW is more inclined to agree with;

- localisation of the η -CTQW may be occurring, due to either classical effects [144, 147] or quantum effects (Anderson localisation).

Whatever the reason, further investigation is required to determine the likely cause of the discrepancy. Note that this is not a negative result per se — depending on the model represented by the graph structure, the η -CTQW could be providing a better result of marking influential and central nodes. However, this analysis is beyond the scope of this study, and is reserved for future research. Nevertheless, the results presented here show that the η -CTQW provides centrality rankings for several classes of randomly generated graphs that are consistent with the classical PageRank algorithm.

9.6 Conclusion

In this study, we have introduced and expanded a framework for continuous-time quantum walks on directed graphs, by utilising PT-symmetry. In the case of interdependent networks of directed graphs, a sufficient condition for ensuring PT-symmetry was detailed, and the directed walk formalism was shown to be equivalent to simulating a continuous-time quantum walker on an undirected, weighted, complete graph with self-loops. This may potentially lead to easily-implementable experimental directed continuous-time quantum walks.

Finally, we have introduced a quantum scheme for centrality testing on directed graphs, by utilising PT-symmetric continuous-time quantum walks — unlike other directed quantum-walk based centrality-measures, our method does not require expanding the Hilbert space to ensure unitary behaviour. A statistical analysis was performed, confirming the CTQW centrality measure proposed here is consistent with classical centrality measures for various classes of randomly generated directed graphs.

Preliminary results on 4-vertex pseudo-Hermitian directed graphs have shown that the CTQW centrality ranking distinguishes non-equivalent sets of vertices that the classical PageRank cannot. This is likely due to the CTQW providing an eigenvector-like centrality measure in the quantum regime; calculating the rank correlation coefficients supports this interpretation. However, further work is required to fully understand the distinguishing power of the pseudo-Hermitian CTQW centrality measure. Following on from this work, we aim to utilise the PT-symmetric CTQW framework to model and simulate behaviour in

physical biochemical systems, such as electron or exciton transport. Future work will also involve exploring methods of implementing the PT-symmetric CTQW centrality scheme on physical systems.

9.7 Acknowledgments

J. A. Izaac would like to thank Thomas Loke, Yogesh Joglekar, Ping Xu, and Yongping Zhang for valuable discussions regarding PT-symmetry and interdependent networks. J. A. Izaac would also like to thank the Hackett foundation and The University of Western Australia for financial support.

CHAPTER 10

Conclusion

Over the course of this thesis, we have explored various facets of the continuous-time quantum walk — from definition to simulation, from application to physical implementation, and even extensions to non-Hermitian systems — gaining new insights to the behaviour and potential applications of continuous-time quantum walks. In this chapter, the results obtained will be summarised, following by possible future work and final remarks.

In considering methods of efficient simulation of the CTQW on a classical computer, we presented a distributed memory software framework *pyCTQW*, which allows the rapid simulation of interacting multi-particle CTQW-based systems. With a backend written using Fortran and a Python user interface, *pyCTQW* takes advantage of the speed provided by Fortran, as well as the extensibility and rich user environment supplied by Python. As a result, *pyCTQW* includes built-in visualisation and data analysis, and allows automatic calculation of quantities such as particle entanglement. We also detailed various numerical algorithms for implementing the matrix exponential numerically, including Krylov subspace techniques, and the Chebyshev series expansion. Moreover, *pyCTQW* has been designed explicitly for distributed memory computation, taking advantage of parallel speedup in its computations — when propagating a quantum walk via the Chebyshev series expansion, *pyCTQW* outperforms all other existing numerical libraries for large graphs, and is approximately one order of magnitude faster than similar implementations utilizing the Krylov algorithm. Finally, we presented an algorithm for mapping a multi-fermion CTQW on a graph to a single particle walk on a reduced-vertex weighted graph; allowing the CTQW to be propagated with a significant speedup and reduction in classical computational resources. *Mathematica* functions are provided to implement this algorithm, and a numerical analysis of the time complexity is performed.

We then proceeded to evaluate potential algorithmic applications of continuous-time quantum walks — focusing on network theory algorithms that may benefit from the ability of the quantum walker to sample graph structure quadratically faster than its classical ana-

logue. We began by considering the graph isomorphism problem, which has motivated the creation of numerous quantum walk-based algorithms in an attempt to efficiently distinguish all classes of non-isomorphic graphs. It is well established that the single particle quantum walk cannot distinguish non-isomorphic pairs of SRGs with the same family parameters, and the same has been proven true of the multi-particle non-interacting CTQW. In this study, we proved using the Bose-Mesner algebra that single-particle perturbed CTQWs are unable to distinguish non-isomorphic SRGs with family parameters $(16,6,2,2)$, unlike the single-particle perturbed DTQW. Thus, we have shown that perturbing the CTQW does not afford it any greater distinguishing power, and conclude that the DTQW is more powerful than CTQWs for distinguishing non-isomorphic graphs.

We then investigated applications of the CTQW to network centrality, and proposed a CTQW-based centrality algorithm. Performing an extensive statistical analysis of the proposed algorithm over various ensembles of random graphs, it was shown that the CTQW centrality demonstrates strong correlation with classical centrality measures such as the eigenvector centrality (correlation coefficient $\tau \approx 0.8 - 0.9$) and the PageRank (correlation coefficient $\tau \approx 0.7 - 0.8$). Furthermore, the CTQW-based measure agrees with the eigenvector centrality on the location of the most central node approximately 95% of the time, leading us to conjecture that the CTQW centrality algorithm provides an analogue to the eigenvector centrality in the quantum realm — this is supported by taking the power series of the time-evolution operator, and comparing it to a well-known matrix power expansion of the eigenvector centrality. This study also reports the first successful physical implementation of a quantum centrality algorithm on a 4-vertex star graph. Implemented via linear optics, the unitary time-evolution operator of the walker is decomposed into multiple unitary operators on a two dimensional subspace, and realised by operating on the polarisation and spacial modes of single photons. This is aided by the fact that the CTQW centrality measure only requires an N -dimensional Hilbert space, compared to previously proposed quantum algorithms, which require an N^2 dimensional space or muting the quantum behaviour via decoherence.

The last major study is concerned with modifying the proposed CTQW centrality measure to allow for centrality analysis on directed networks, as the CTQW remains hitherto undefined and non-unitary on non-Hermitian graph structures. To do so, we have introduced a generalisation of the CTQW, the pseudo-Hermitian η -CTQW, which utilises the framework of PT-symmetry to ensure unitary time-evolution is preserved on PT-symmetric directed graphs. Expanding the results to directed interdependent network, a sufficient condition for ensuring PT-symmetry was detailed, and the pseudo-Hermitian CTQW formalism was shown to be equivalent to simulating the CTQW on a directed graph via an

undirected, edge-weighted graph. Finally, we extend our CTQW centrality measure to PT-symmetric directed graphs, and show that the η -CTQW centrality measure provides an eigenvector-like centrality measure for the quantum regime; this is hypothesized to be due to the spectra-preserving property of the pseudo-Hermitian similarity transform. A statistical analysis over an ensemble of random graphs was then performed, confirming that the η -CTQW centrality measure proposed here remains consistent with the results of classical centrality measures, for various classes of randomly generated directed graphs — in the case of scale-free directed graphs, the η -CTQW measure agrees with the classical PageRank approximately 95% of the time, although showing less agreement when considering Erdős-Rényi graphs with allowed bidirectionality of edges. Furthermore, the η -CTQW centrality measure improves on the eigenvector centrality measure by providing meaningful results on directed graphs which are not accessible to the classical eigenvector centrality.

Quantum walks remain a fundamental tool, building off decades of study into random walk modelling and quantum computation to link numerous interdisciplinary fields of study, including quantum information theory, network analysis, and quantum dynamical modelling. The work presented as part of this thesis attempts to significantly improve our ability to numerically simulate and analyse such systems, whilst providing insight into various possible network theory applications. Nevertheless, many questions remained unanswered. It is currently unknown whether multi-particle quantum walks with or without interactions will solve the graph isomorphism problem — whilst suspected to be unlikely, this deserves further study. It is also interesting to ask, aside from network centrality, what other graph and network properties — such as robustness, percolation, density, and similarity — may be calculated via quantum walks, leading to future quantum algorithms.

The pseudo-Hermitian CTQW framework, meanwhile, may provide methods of modelling and simulating behaviour of physical biochemical systems, such as electron or exciton transport, and should be examined closely. Future work will also consider the ability to determine, based simply on graphical properties, whether particular directed graphs satisfy PT-symmetry, and define classes of directed graphs sharing the property of PT-symmetry. Furthermore, physical implementation of the pseudo-Hermitian CTQW centrality scheme will be explored. Finally, future versions of *pyCTQW* will aim to further increase the efficiency of CTQW numerical simulation, whilst including features such as quantum stochastic walks via superoperators, efficient simulation of fermionic and bosonic multi-particle quantum walks, and the ability to efficiently simulate in parallel large ensembles of random graphs, for example for quantum percolation and centrality analysis.

In a broader sense, quantum walks are expected to play an important role in quantum information theory going forward, as they lie at the intersection between rapid advances in quantum computation and the modelling of both classical and quantum dynamical systems. Examples include network theory, where entangled communication networks and quantum network correlations are revolutionising how we characterise network structure and information flow; biochemistry, where chemical reactions, biological evolution, and coherent energy transport are being modelled by quantum walk-based models more suited to their underlying quantum behaviour; and computer science, where quantum machine learning, pattern matching, and quantum walk-based optimisation are expected to provide breakthrough advances. Considering the powerful algorithmic groundwork laid by classical random walks in the century since their conception, we have yet to scratch the surface of what quantum walks are capable of.

References

- [1] Y. Aharonov, L. Davidovich, and N. Zagury, “Quantum random walks”, *Physical Review A* **48**, 1687–1690 (1993).
- [2] J. Kempe, “Quantum random walks: an introductory overview”, *Contemporary Physics* **44**, 307–327 (2003).
- [3] K. Manouchehri and J. Wang, *Physical implementation of quantum walks* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2014).
- [4] A. M. Childs, E. Farhi, and S. Gutmann, “An example of the difference between quantum and classical random walks”, *Quantum Information Processing* **1**, 35–43 (2002).
- [5] J. K. Gamble, M. Friesen, D. Zhou, R. Joynt, and S. N. Coppersmith, “Two-particle quantum walks applied to the graph isomorphism problem”, *Physical Review A* **81**, 052313 (2010).
- [6] K. Rudinger, J. Gamble, M. Wellons, E. Bach, M. Friesen, R. Joynt, and S. Coppersmith, “Noninteracting multiparticle quantum random walks applied to the graph isomorphism problem for strongly regular graphs”, *Physical Review A* **86**, 022334 (2012).
- [7] J. A. Izaac, J. B. Wang, and Z. J. Li, “Continuous-time quantum walks with defects and disorder”, *Physical Review A* **88**, 042334 (2013).
- [8] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, “Exponential algorithmic speedup by a quantum walk”, in *Proceedings of the thirty-fifth annual ACM symposium on theory of computing, STOC '03* (2003), 59–68.
- [9] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum algorithm for the hamiltonian NAND tree”, arXiv preprint quant-ph/0702144 (2007).
- [10] A. M. Childs and J. Goldstone, “Spatial search by quantum walk”, *Physical Review A* **70**, 022314 (2004).
- [11] S. Berry and J. Wang, “Quantum-walk-based search and centrality”, *Physical Review A* **82**, 042333 (2010).
- [12] G. D. Paparo, M. Müller, F. Comellas, and M. A. Martin-Delgado, “Quantum google in a complex network”, *Scientific Reports* **3**, 02773 (2013).
- [13] T. Loke and J. B. Wang, “Efficient quantum circuits for continuous-time quantum walks on composite graphs”, *Journal of Physics A: Mathematical and Theoretical* **50**, 055303 (2017).
- [14] J. A. Izaac, X. Zhan, Z. Bian, K. Wang, J. Li, J. B. Wang, and P. Xue, “Centrality measure based on continuous-time quantum walks and experimental realization”, *Physical Review A* **95**, 032318 (2017).

-
- [15] B. L. Douglas and J. B. Wang, “A classical approach to the graph isomorphism problem using quantum walks”, *Journal of Physics A: Mathematical and Theoretical* **41**, 075303 (2008).
- [16] S. D. Berry, P. Bourke, and J. B. Wang, “qwViz: visualisation of quantum walks on graphs”, *Computer Physics Communications* **182**, 2295–2302 (2011).
- [17] K. Rudinger, J. K. Gamble, E. Bach, M. Friesen, R. Joynt, and S. N. Coppersmith, “Comparing algorithms for graph isomorphism using discrete- and Continuous-Time quantum random walks”, *Journal of Computational and Theoretical Nanoscience* **10**, 1653–1661 (2013).
- [18] A. Mahasinghe, J. A. Izaac, J. B. Wang, and J. K. Wijerathna, “Phase-modified CTQW unable to distinguish strongly regular graphs efficiently”, *Journal of Physics A: Mathematical and Theoretical* **48**, 265301 (2015).
- [19] M. Sarovar, A. Ishizaki, G. R. Fleming, and K. B. Whaley, “Quantum entanglement in photosynthetic light-harvesting complexes”, *Nature Physics* **6**, 462–467 (2010).
- [20] G. S. Engel, T. R. Calhoun, E. L. Read, T. Ahn, T. Mančal, Y. Cheng, R. E. Blankenship, and G. R. Fleming, “Evidence for wavelike energy transfer through quantum coherence in photosynthetic systems”, *Nature* **446**, 782–786 (2007).
- [21] E. Collini, C. Y. Wong, K. E. Wilk, P. M. G. Curmi, P. Brumer, and G. D. Scholes, “Coherently wired light-harvesting in photosynthetic marine algae at ambient temperature”, *Nature* **463**, 644–647 (2010).
- [22] G. S. Engel, “Quantum coherence in photosynthesis”, 22nd Solvay Conference on Chemistry **3**, 222–231 (2011).
- [23] M. Mohseni, P. Rebentrost, S. Lloyd, and A. Aspuru-Guzik, “Environment-assisted quantum walks in photosynthetic energy transfer”, *The Journal of Chemical Physics* **129**, 174106–174106-9 (2008).
- [24] P. Rebentrost, M. Mohseni, I. Kassal, S. Lloyd, and A. Aspuru-Guzik, “Environment-assisted quantum transport”, *New Journal of Physics* **11**, 033003 (2009).
- [25] A. M. Childs, “Universal computation by quantum walk”, *Physical Review Letters* **102**, 180501 (2009).
- [26] A. M. Childs and D. Gosset, “Levinson’s theorem for graphs II”, arXiv:1203.6557 (2012).
- [27] K. Pearson, “The problem of the random walk”, *Nature* **72**, 294–294 (1905).
- [28] B. D. Hughes, *Random walks and random environments: volume 1: random walks* (Oxford University Press, Oxford, New York, Mar. 1995).

-
- [29] M. Kac, “Random walk and the theory of brownian motion”, *The American Mathematical Monthly* **54**, 369–391 (1947).
- [30] P. M. Kareiva and N. Shigesada, “Analyzing insect movement as a correlated random walk”, *Oecologia* **56**, 234–238 (1983).
- [31] E. E. Holmes, M. A. Lewis, J. E. Banks, and R. R. Veit, “Partial differential equations in ecology: spatial interactions and population dynamics”, *Ecology* **75**, 17–29 (1994).
- [32] F. L. Bookstein, “Random walk and the existence of evolutionary rates”, *Paleobiology* **13**, 446–464 (1987).
- [33] J. E. Neigel and J. C. Avise, “Application of a random walk model to geographic distributions of animal mitochondrial DNA variation.”, *Genetics* **135**, 1209–1220 (1993).
- [34] D. H. Tanaka, M. Yanagida, Y. Zhu, S. Mikami, T. Nagasawa, J.-i. Miyazaki, Y. Yanagawa, K. Obata, and F. Murakami, “Random walk behavior of migrating cortical interneurons in the marginal zone: Time-Lapse analysis in Flat-Mount cortex”, *Journal of Neuroscience* **29**, 1300–1311 (2009).
- [35] L. Bachelier, *Théorie de la spéculation* (Gauthier-Villars, 1900).
- [36] E. F. Fama, “Random walks in stock market prices”, *Financial Analysts Journal* **21**, 55–59 (1965).
- [37] P. H. Cootner, *The random character of stock market prices* (The M.I.T. Press, Cambridge Mass., 1970).
- [38] M. Gori, M. Maggini, and L. Sarti, “Exact and approximate graph matching using random walks”, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**, 1100–1111 (2005).
- [39] J. D. Noh and H. Rieger, “Random walks on complex networks”, *Physical Review Letters* **92**, 118701 (2004).
- [40] S. Brin and L. Page, “Proceedings of the seventh international world wide web ConferenceThe anatomy of a large-scale hypertextual web search engine”, *Computer Networks and ISDN Systems* **30**, 107–117 (1998).
- [41] O. Mülken and A. Blumen, “Continuous-time quantum walks: models for coherent transport on complex networks”, *Physics Reports* **502**, 37–87 (2011).
- [42] A. M. Childs, D. Gosset, and Z. Webb, “Universal computation by multiparticle quantum walk”, *Science* **339**, 791–794 (2013).
- [43] T. Loke and J. B. Wang, “Efficient circuit implementation of quantum walks on non-degree-regular graphs”, *Physical Review A* **86**, 042338 (2012).

-
- [44] T. Loke and J. B. Wang, “Efficient quantum circuits for szegedy quantum walks”, arXiv:1609.00173 [quant-ph] (2016).
- [45] X. Qiang, T. Loke, A. Montanaro, K. Aungskunsiri, X. Zhou, J. L. O’Brien, J. B. Wang, and J. C. F. Matthews, “Efficient quantum walk on a quantum processor”, *Nature Communications* **7**, 11511 (2016).
- [46] Y. Bromberg, Y. Lahini, R. Morandotti, and Y. Silberberg, “Quantum and classical correlations in waveguide lattices”, *Physical Review Letters* **102**, 253904 (2009).
- [47] A. Peruzzo, M. Lobino, J. C. F. Matthews, N. Matsuda, A. Politi, K. Poulios, X. Zhou, Y. Lahini, N. Ismail, K. Wörhoff, Y. Bromberg, Y. Silberberg, M. G. Thompson, and J. L. OBrien, “Quantum walks of correlated photons”, *Science* **329**, 1500–1503 (2010).
- [48] L. Sansoni, F. Sciarrino, G. Vallone, P. Mataloni, A. Crespi, R. Ramponi, and R. Osellame, “Two-Particle Bosonic-Fermionic quantum walk via integrated photonics”, *Physical Review Letters* **108**, 010502 (2012).
- [49] M. Karski, L. Förster, J. Choi, A. Steffen, W. Alt, D. Meschede, and A. Widera, “Quantum walk in position space with single optically trapped atoms”, *Science* **325**, 174–177 (2009).
- [50] H. Schmitz, R. Matjeschk, C. Schneider, J. Glueckert, M. Enderlein, T. Huber, and T. Schaetz, “Quantum walk of a trapped ion in phase space”, *Physical Review Letters* **103**, 090504 (2009).
- [51] M. A. Broome, A. Fedrizzi, B. P. Lanyon, I. Kassal, A. Aspuru-Guzik, and A. G. White, “Discrete Single-Photon quantum walks with tunable decoherence”, *Physical Review Letters* **104**, 153602 (2010).
- [52] E. Farhi and S. Gutmann, “Quantum computation and decision trees”, *Physical Review A* **58**, 915–928 (1998).
- [53] F. Strauch, “Connecting the discrete- and continuous-time quantum walks”, *Physical Review A* **74**, 030301 (2006).
- [54] P. Philipp and R. Portugal, “Exact simulation of coined quantum walks with the continuous-time model”, arXiv:1606.04030 [quant-ph] (2016).
- [55] L. Bergamaschi and M. Vianello, “Efficient computation of the exponential operator for large, sparse, symmetric matrices”, *Numerical linear algebra with applications* **7**, 27–45 (2000).
- [56] C. Moler and C. Van Loan, “Nineteen dubious ways to compute the exponential of a matrix, Twenty-Five years later”, *SIAM Review* **45**, 3–49 (2003).
- [57] N. J. Higham, “The scaling and squaring method for the matrix exponential revisited”, *SIAM Review* **51**, 747–764 (2009).

-
- [58] A. H. Al-Mohy and N. J. Higham, “Computing the action of the matrix exponential, with an application to exponential integrators”, *SIAM Journal on Scientific Computing* **33**, 488–511 (2011).
- [59] C. Leforestier, R. Bisseling, C. Cerjan, M. Feit, R. Friesner, A. Guldborg, A. Hammerich, G. Jolicard, W. Karrlein, H. Meyer, N. Lipkin, O. Roncero, and R. Kosloff, “A comparison of different propagation schemes for the time dependent schrödinger equation”, *Journal of Computational Physics* **94**, 59–80 (1991).
- [60] J. Fang, “One step time propagation method for systems with time-dependent hamiltonians”, *Chemical Physics Letters* **263**, 759–766 (1996).
- [61] J. B. Wang and T. T. Scholz, “Time-dependent approach to scattering by chebyshev-polynomial expansion and the fast-Fourier-transform algorithm”, *Physical Review A* **57**, 3554–3559 (1998).
- [62] J. B. Wang and S. Midgley, “Quantum waveguide theory: a direct solution to the time-dependent schrödinger equation”, *Physical Review B* **60**, 13668–13675 (1999).
- [63] H. Guo, “Chebyshev propagation and applications to scattering problems”, in *Theory of chemical reaction dynamics*, edited by A. Lagana and G. Lendvay, NATO Science Series II: Mathematics, Physics and Chemistry 145 (Springer Netherlands, Jan. 2005), pp. 217–229.
- [64] M. Ndong, H. Tal-Ezer, R. Kosloff, and C. P. Koch, “A chebychev propagator with iterative time ordering for explicitly time-dependent hamiltonians”, *The Journal of Chemical Physics* **132**, 064105 (2010).
- [65] U. Schöning, “Graph isomorphism is in the low hierarchy”, in *STACS 87*, edited by F. J. Brandenburg, G. Vidal-Naquet, and M. Wirsing, Lecture Notes in Computer Science 247 (Springer Berlin Heidelberg, Jan. 1987), pp. 114–124.
- [66] J. R. Dias and G. W. A. Milne, “Chemical applications of graph theory”, *Journal of Chemical Information and Computer Sciences* **32**, 1–1 (1992).
- [67] S. D. Berry and J. B. Wang, “Two-particle quantum walks: entanglement and graph isomorphism testing”, *Physical Review A* **83**, 042317 (2011).
- [68] D. Emms, S. Severini, R. C. Wilson, and E. R. Hancock, “Coined quantum walks lift the cospectrality of graphs and trees”, *Pattern Recognition* **42**, 1988–2002 (2009).
- [69] H. Wang, J. Wu, X. Yang, and X. Yi, “A graph isomorphism algorithm using signatures computed via quantum walk search model”, *Journal of Physics A: Mathematical and Theoretical* **48**, 115302 (2015).
- [70] X. Qiang, X. Yang, J. Wu, and X. Zhu, “An enhanced classical approach to graph isomorphism using continuous-time quantum walk”, *Journal of Physics A: Mathematical and Theoretical* **45**, 045305 (2012).

-
- [71] S. P. Borgatti, “Centrality and network flow”, *Social Networks* **27**, 55–71 (2005).
- [72] D. J. Brass, “Being in the right place: a structural analysis of individual influence in an organization”, *Administrative Science Quarterly* **29**, 518–539 (1984).
- [73] H. Ibarra and S. B. Andrews, “Power, social influence, and sense making: effects of network centrality and proximity on employee perceptions”, *Administrative Science Quarterly* **38**, 277–303 (1993).
- [74] H. Ibarra, “Network centrality, power, and innovation involvement: determinants of technical and administrative roles”, *Academy of Management Journal* **36**, 471–501 (1993).
- [75] D. S. Sade, “Sociometrics of macaca mulatta III: n-path centrality in grooming networks”, *Social Networks, Special Issue on Non-Human Primate Networks* **11**, 273–292 (1989).
- [76] G. Amitai, A. Shemesh, E. Sitbon, M. Shklar, D. Netanel, I. Venger, and S. Pietrokovski, “Network analysis of protein structures identifies functional residues”, *Journal of Molecular Biology* **344**, 1135–1146 (2004).
- [77] G. D. Paparo and M. A. Martin-Delgado, “Google in a quantum network”, *Scientific Reports* **2**, 00444 (2012).
- [78] I. Sinayskiy and F. Petruccione, “Open quantum walks: a short introduction”, *Journal of Physics: Conference Series* **442**, 012003 (2013).
- [79] P. E. Falloon, J. Rodriguez, and J. B. Wang, “QSWalk: a mathematica package for quantum stochastic walks on arbitrary graphs”, arXiv:1606.04974 [quant-ph] (2016).
- [80] M. Szegedy, “Quantum speed-up of markov chain based algorithms”, in 45th annual IEEE symposium on foundations of computer science (Oct. 2004), pp. 32–41.
- [81] C. M. Bender and S. Boettcher, “Real spectra in Non-Hermitian hamiltonians having PT symmetry”, *Physical Review Letters* **80**, 5243–5246 (1998).
- [82] A. Mostafazadeh, “Pseudo-Hermiticity versus PT symmetry: the necessary condition for the reality of the spectrum of a non-Hermitian hamiltonian”, *Journal of Mathematical Physics* **43**, 205–214 (2002).
- [83] A. Mostafazadeh, “Pseudo-Hermiticity versus PT-symmetry III: equivalence of pseudo-Hermiticity and the presence of antilinear symmetries”, *Journal of Mathematical Physics* **43**, 3944–3951 (2002).
- [84] P. D. Mannheim, “PT symmetry as a necessary and sufficient condition for unitary time evolution”, *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* **371**, 20120060 (2013).

-
- [85] J. Li, Y. Miao, and Z. Xue, “A possible method for Non-Hermitian and Non-PT-Symmetric hamiltonian systems”, PLoS ONE **9**, e97107 (2014).
- [86] S. Salimi and A. Sorouri, “Pseudo-Hermitian continuous-time quantum walks”, Journal of Physics A: Mathematical and Theoretical **43**, 275304 (2010).
- [87] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, “Quantum walks on graphs”, in Proceedings of the thirty-third annual ACM symposium on theory of computing, STOC '01 (2001), 50–59.
- [88] G. Kirchhoff, “Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird”, Annalen der Physik **148**, 497–508 (1847).
- [89] S. Chaiken and D. Kleitman, “Matrix tree theorems”, Journal of Combinatorial Theory, Series A **24**, 377–381 (1978).
- [90] C. W. Wu, “On Rayleigh–Ritz ratios of a generalized laplacian matrix of directed graphs”, Linear Algebra and its Applications **402**, 207–227 (2005).
- [91] F. Bauer, “Normalized graph laplacians for directed graphs”, Linear Algebra and its Applications **436**, 4193–4222 (2012).
- [92] D. Kalita, “Determinant of the laplacian matrix of a weighted directed graph”, in *Combinatorial matrix theory and generalized inverses of matrices*, edited by R. B. Bapat, S. J. Kirkland, K. M. Prasad, and S. Puntanen (Springer India, 2013), pp. 57–62.
- [93] R. van der Hofstad, *Random graphs and complex networks* (Cambridge University Press, Cambridge, 2016).
- [94] P. Erdős and A. Rényi, “On random graphs, i”, Publicationes Mathematicae (Debrecen) **6**, 290–297 (1959).
- [95] E. N. Gilbert, “Random graphs”, The Annals of Mathematical Statistics **30**, 1141–1144 (1959).
- [96] P. Erdős and A. Rényi, “On the evolution of random graphs”, in Publication of the mathematical institute of the hungarian academy of sciences (1960), 17–61.
- [97] R. Albert and A. Barabási, “Statistical mechanics of complex networks”, Reviews of Modern Physics **74**, 47–97 (2002).
- [98] C. Song, S. Havlin, and H. A. Makse, “Self-similarity of complex networks”, Nature **433**, 392–395 (2005).
- [99] A. Barabási and R. Albert, “Emergence of scaling in random networks”, Science **286**, 509–512 (1999).

-
- [100] A. Barabási, R. Albert, and H. Jeong, “Scale-free characteristics of random networks: the topology of the world-wide web”, *Physica A: Statistical Mechanics and its Applications* **281**, 69–77 (2000).
- [101] A. M. Childs, “On the relationship between continuous- and Discrete-Time quantum walk”, *Communications in Mathematical Physics* **294**, 581–603 (2010).
- [102] J. Smith, “Algebraic aspects of Multi-Particle quantum walks”, PhD (University of Waterloo, Dec. 2012).
- [103] J. D. Whitfield, C. A. Rodríguez-Rosario, and A. Aspuru-Guzik, “Quantum stochastic walks: a generalization of classical random walks and quantum walks”, *Physical Review A* **81**, 022323 (2010).
- [104] T. G. Wong, L. Tarrataca, and N. Nahimov, “Laplacian versus adjacency matrix in quantum walk search”, *Quantum Information Processing* **15**, 4029–4048 (2016).
- [105] M. Born, “Zur quantenmechanik der stossvorgänge”, *Zeitschrift für Physik* **37**, 863–867 (1926).
- [106] P. R. Dukes, “Indistinguishable quantum walks on graphs relative to a bipartite quantum walker”, arXiv:1610.08421 [quant-ph] (2016).
- [107] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks* (Princeton University Press, July 2010).
- [108] J. A. Izaac and J. B. Wang, “pyCTQW: a continuous-time quantum walk simulator on distributed memory computers”, *Computer Physics Communications* **186**, 81–92 (2015).
- [109] P. Peterson, “F2PY: a tool for connecting fortran and python programs”, *International Journal of Computational Science and Engineering* **4**, 296 (2009).
- [110] C. Moler and C. Van Loan, “Nineteen dubious ways to compute the exponential of a matrix”, *SIAM Review* **20**, 801–836 (1978).
- [111] T. Auckenthaler, M. Bader, T. Huckle, A. Spörl, and K. Waldherr, “Matrix exponentials and parallel prefix computation in a quantum control problem”, *Parallel Computing* **36**, 359–369 (2010).
- [112] R. B. Sidje, “Expokit: a software package for computing matrix exponentials”, *ACM Transactions on Mathematical Software (TOMS)* **24**, 130–156 (1998).
- [113] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, “Efficient management of parallelism in object oriented numerical software libraries”, in *Modern software tools in scientific computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen (1997), 163–202.

-
- [114] S. Balay, J. Brown, Buschelman, Kris, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc users manual*, tech. rep. ANL-95/11 - Revision 3.4 (Argonne National Laboratory, 2013).
- [115] S. Balay, J. Brown, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc web page*, <http://www.mcs.anl.gov/petsc/index.html>, 2013.
- [116] V. Hernandez, J. E. Roman, and V. Vidal, “SLEPc: scalable library for eigenvalue problem computations”, *Lecture Notes in Computer Science* **2565**, 377–391 (2003).
- [117] V. Hernandez, J. E. Roman, and V. Vidal, “SLEPc: a scalable and flexible toolkit for the solution of eigenvalue problems”, *ACM Trans. Math. Software* **31**, 351–362 (2005).
- [118] C. Campos, J. E. Roman, E. Romero, and A. Tomas, *SLEPc users manual*, tech. rep. DSIC-II/24/02 - Revision 3.3 (D. Sistemes Informàtics i Computació, Universitat Politècnica de València, 2012).
- [119] C. Benedetti, F. Buscemi, and P. Bordone, “Quantum correlations in continuous-time quantum walks of two indistinguishable particles”, *Physical Review A* **85**, 042314 (2012).
- [120] A. A. Melnikov and L. E. Fedichkin, “Quantum walks of interacting fermions on a cycle graph”, *Scientific Reports* **6**, 34226 (2016).
- [121] L. Wang, L. Wang, and Y. Zhang, “Quantum walks of two interacting anyons in one-dimensional optical lattices”, *Physical Review A* **90**, 063618 (2014).
- [122] K. Poullos, D. Fry, M. Lobino, J. C. F. Matthews, J. D. A. Meinecke, A. Peruzzo, X. Zhou, M. G. Thompson, J. L. O’Brien, A. Politi, R. Keil, A. Szameit, N. Ismail, K. Wörhoff, and N. Matsuda, “Multi-particle quantum walks with 2D & 3D arrayed waveguides”, in 16th european conference on integrated optics, ECIO 2012 (Apr. 2012).
- [123] L. Novo, S. Chakraborty, M. Mohseni, H. Neven, and Y. Omar, “Systematic dimensionality reduction for quantum walks: optimal spatial search and transport on Non-Regular graphs”, *Scientific Reports* **5**, 13304 (2015).
- [124] F. Pilar, *Elementary quantum chemistry*, 2nd Edition, Dover Books on Chemistry (Dover Publications, Mineola, N.Y., 2013).
- [125] J. M. Zhang and R. X. Dong, “Exact diagonalization: the Bose–Hubbard model as an example”, *European Journal of Physics* **31**, 591 (2010).
- [126] J. W. Raymond and P. Willett, “Maximum common subgraph isomorphism algorithms for the matching of chemical structures”, *Journal of Computer-Aided Molecular Design* **16**, 521–533 (2002).

-
- [127] D. S. Johnson, “The NP-completeness column”, *ACM Trans. Algorithms* **1**, 160–176 (2005).
- [128] S.-y. Shiao, R. Joynt, and S. N. Coppersmith, “Physically-motivated dynamical algorithms for the graph isomorphism problem”, *Quantum Info. Comput.* **5**, 492–506 (2005).
- [129] B. Weisfeiler and A. Lehman, “A reduction of a graph to a canonical form and an algebra arising during this reduction”, *Nauchno-Technicheskaya Informatsia* **2**, 12–16 (1968).
- [130] D. G. Higman, “Coherent configurations”, *Geometriae Dedicata* **4**, 1–32 (1975).
- [131] C. J. Colbourn and J. H. Dinitz, eds., *Handbook of combinatorial designs*, 2nd ed, Discrete mathematics and its applications (Chapman & Hall/Taylor & Francis, Boca Raton, FL, 2007).
- [132] F. Levestin and C. Maldonado, “Generalized quadrangles and subconstituent algebra”, *CUBO A Mathematical Journal* **12**, 53–75 (2010).
- [133] J. Smith, “k-Boson quantum walks do not distinguish arbitrary graphs”, arXiv:1004.0206 [quant-ph] (2010).
- [134] B. Curtin and I. Daqqa, “The subconstituent algebra of strongly regular graphs associated with a latin square”, *Designs, Codes and Cryptography* **52**, 263–274 (2009).
- [135] J. T. Go, “The terwilliger algebra of the hypercube”, *European Journal of Combinatorics* **23**, 399–429 (2002).
- [136] P. Terwilliger, “The subconstituent algebra of an association scheme, (Part i)”, *Journal of Algebraic Combinatorics* **1**, 363–388 (1992).
- [137] E. Bannai and A. Munemasa, “The terwilliger algebras of group association schemes”, *Kyushu Journal of Mathematics* **49**, 93–102 (1995).
- [138] D. Knoke and R. S. Burt, “Centrality”, *Applied Network Analysis*, 195–222 (1983).
- [139] G. Lawyer, “Understanding the influence of all nodes in a network”, *Scientific Reports* **5**, 8665 (2015).
- [140] J. Liu, Z. Ren, and Q. Guo, “Ranking the spreading influence in complex networks”, *Physica A: Statistical Mechanics and its Applications* **392**, 4154–4159 (2013).
- [141] R. B. Rothenberg, J. J. Potterat, D. E. Woodhouse, W. W. Darrow, S. Q. Muth, and A. S. Klovdahl, “Choosing a centrality measure: epidemiologic correlates in the colorado springs study of social networks”, *Social Networks, Social networks and infectious disease: HIV/AIDS* **17**, 273–297 (1995).

-
- [142] T. J. Grosser, V. Lopez-Kidwell, and G. Labianca, “A social network analysis of positive and negative gossip in organizational life”, *Group & Organization Management* **35**, 177–212 (2010).
- [143] S. P. Borgatti and M. G. Everett, “A graph-theoretic perspective on centrality”, *Social Networks* **28**, 466–484 (2006).
- [144] R. Pastor-Satorras and C. Castellano, “Distinct types of eigenvector localization in networks”, *Scientific Reports* **6**, 18847 (2016).
- [145] P. Bonacich, “Power and centrality: a family of measures”, *American Journal of Sociology* **92**, 1170–1182 (1987).
- [146] N. Perra and S. Fortunato, “Spectral centrality measures in complex networks”, *Physical Review E* **78**, 036107 (2008).
- [147] T. Martin, X. Zhang, and M. E. J. Newman, “Localization and centrality in networks”, *Physical Review E* **90**, 052808 (2014).
- [148] N. Litvak, W. R. W. Scheinhardt, and Y. Volkovich, “In-Degree and PageRank: why do they follow similar power laws?”, *Internet Mathematics* **4**, 175–198 (2007).
- [149] D. F. Gleich and K. Kloster, “Seeded PageRank solution paths”, *European Journal of Applied Mathematics* **27**, 812–845 (2016).
- [150] L. Rossi, A. Torsello, and E. R. Hancock, “Node centrality for Continuous-Time quantum walks”, in *Structural, syntactic, and statistical pattern recognition*, Vol. 8621, edited by P. Fränti, G. Brown, M. Loog, F. Escolano, and M. Pelillo (Springer Berlin Heidelberg, Berlin, Heidelberg, 2014), pp. 103–112.
- [151] L. C. Freeman, “Centrality in social networks conceptual clarification”, *Social Networks* **1**, 215–239 (1978).
- [152] M. G. Kendall, “A new measure of rank correlation”, *Biometrika* **30**, 81–93 (1938).
- [153] J. D. Evans, *Straightforward statistics for the behavioral sciences* (Brooks/Cole Pub. Co, Pacific Grove, 1996).
- [154] H. Kim and R. Anderson, “Temporal node centrality in complex networks”, *Physical Review E* **85**, 026107 (2012).
- [155] D. Chen, L. Lü, M. Shang, Y. Zhang, and T. Zhou, “Identifying influential nodes in complex networks”, *Physica A: Statistical Mechanics and its Applications* **391**, 1777–1787 (2012).
- [156] A. Lulli, L. Ricci, E. Carlini, and P. Dazzi, “Distributed current flow betweenness centrality”, in *2015 IEEE 9th international conference on Self-Adaptive and Self-Organizing systems* (Sept. 2015), pp. 71–80.

-
- [157] Y. Du, C. Gao, X. Chen, Y. Hu, R. Sadiq, and Y. Deng, “A new closeness centrality measure via effective distance in complex networks”, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **25**, 033112 (2015).
- [158] W. R. Knight, “A computer method for calculating kendall’s tau with ungrouped data”, *Journal of the American Statistical Association* **61**, 436–439 (1966).
- [159] M. G. Kendall, “The treatment of ties in ranking problems”, *Biometrika* **33**, 239–251 (1945).
- [160] E. Yilmaz, J. A. Aslam, and S. Robertson, “A new rank correlation coefficient for information retrieval”, in *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval, SIGIR ’08* (2008), 587–594.
- [161] S. Vigna, “A weighted correlation index for rankings with ties”, in *Proceedings of the 24th international conference on world wide web, WWW ’15* (2015), 1166–1176.
- [162] A. S. Maiya and T. Y. Berger-Wolf, “Online sampling of high centrality individuals in social networks”, in *Advances in knowledge discovery and data mining* (June 2010), pp. 91–98.
- [163] A. Agresti and B. A. Coull, “Approximate is better than ”Exact” for interval estimation of binomial proportions”, *The American Statistician* **52**, 119–126 (1998).
- [164] D. W. Berry, A. M. Childs, and R. Kothari, “Hamiltonian simulation with nearly optimal dependence on all parameters”, in *2015 IEEE 56th annual symposium on foundations of computer science (FOCS)* (Oct. 2015), pp. 792–809.
- [165] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, “Experimental realization of any discrete unitary operator”, *Physical Review Letters* **73**, 58–61 (1994).
- [166] G. Stewart, “On the perturbation of Pseudo-Inverses, projections and linear least squares problems”, *SIAM Review* **19**, 634–662 (1977).
- [167] G. W. Stewart, “Computing the CS decomposition of a partitioned orthonormal matrix”, *Numerische Mathematik* **40**, 297–306 (1982).
- [168] C. C. Paige and M. Wei, “History and generality of the CS decomposition”, *Linear Algebra and its Applications* **208**, 303–326 (1994).
- [169] B. D. Sutton, “Computing the complete CS decomposition”, *Numerical Algorithms* **50**, 33–65 (2008).
- [170] I. Dhand and S. K. Goyal, “Realization of arbitrary discrete unitary transformations using spatial and internal modes of light”, *Physical Review A* **92**, 043813 (2015).
- [171] S. K. Goyal, F. S. Roux, A. Forbes, and T. Konrad, “Implementation of multidimensional quantum walks using linear optics and classical light”, *Physical Review A* **92**, 040302 (2015).

-
- [172] J. A. Izaac, J. B. Wang, P. C. Abbott, and X. S. Ma, “Quantum centrality testing on directed graphs via PT-symmetric quantum walks”, arXiv:1607.02673v3 [quant-ph] (2017).
- [173] B. Basu-Mallick and A. Kundu, “Exact solution to the calogero model with competing long-range interactions”, *Physical Review B* **62**, 9927–9930 (2000).
- [174] C. M. Bender, “PT symmetry in quantum field theory”, *Czechoslovak Journal of Physics* **54**, 13–28 (2004).
- [175] C. M. Bender, H. F. Jones, and R. J. Rivers, “Dual \mathcal{P} -symmetric quantum field theories”, *Physics Letters B* **625**, 333–340 (2005).
- [176] X. Luo, J. Huang, H. Zhong, X. Qin, Q. Xie, Y. S. Kivshar, and C. Lee, “Pseudo-Parity-Time symmetry in optical systems”, *Physical Review Letters* **110**, 243902 (2013).
- [177] B. Peng, Ş. K. Özdemir, F. Lei, F. Monifi, M. Gianfreda, G. L. Long, S. Fan, F. Nori, C. M. Bender, and L. Yang, “Parity-time-symmetric whispering-gallery microcavities”, *Nature Physics* **10**, 394–398 (2014).
- [178] A. Regensburger, C. Bersch, M. Miri, G. Onishchukov, D. N. Christodoulides, and U. Peschel, “Parity-time synthetic photonic lattices”, *Nature* **488**, 167–171 (2012).
- [179] C. E. Rüter, K. G. Makris, R. El-Ganainy, D. N. Christodoulides, M. Segev, and D. Kip, “Observation of parity–time symmetry in optics”, *Nature Physics* **6**, 192–195 (2010).
- [180] F. J. Dyson, “General theory of Spin-Wave interactions”, *Physical Review* **102**, 1217–1230 (1956).
- [181] F. H. M. Faisal and J. V. Moloney, “Time-dependent theory of non-Hermitian schrodinger equation: application to multiphoton-induced ionisation decay of atoms”, *Journal of Physics B: Atomic and Molecular Physics* **14**, 3603 (1981).
- [182] K. Jones-Smith, “A ‘Dysonization’ scheme for identifying quasi-particles using non-Hermitian quantum mechanics”, *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* **371**, 20120056 (2013).
- [183] K. Mochizuki, D. Kim, and H. Obuse, “Explicit definition of PT symmetry for nonunitary quantum walks with gain and loss”, *Physical Review A* **93**, 062116 (2016).
- [184] F. M. Fernández, R. Guardiola, J. Ros, and M. Znojil, “A family of complex potentials with real spectrum”, *Journal of Physics A: Mathematical and General* **32**, 3105 (1999).
- [185] S. Weigert, “PT-symmetry and its spontaneous breakdown explained by anti-linearity”, *Journal of Optics B: Quantum and Semiclassical Optics* **5**, S416 (2003).

-
- [186] C. F. de Morisson Faria and A. Fring, “Time evolution of non-Hermitian hamiltonian systems”, *Journal of Physics A: Mathematical and General* **39**, 9269 (2006).
- [187] A. Mostafazadeh, “Pseudo-Hermiticity versus PT-symmetry. II. a complete characterization of non-Hermitian hamiltonians with a real spectrum”, *Journal of Mathematical Physics* **43**, 2814–2816 (2002).
- [188] L. Solombrino, “Weak pseudo-Hermiticity and antilinear commutant”, *Journal of Mathematical Physics* **43**, 5439–5445 (2002).
- [189] G. Scolarici and L. Solombrino, “On the pseudo-Hermitian nondiagonalizable hamiltonians”, *Journal of Mathematical Physics* **44**, 4450–4459 (2003).
- [190] C. M. Bender and P. D. Mannheim, “Symmetry and necessary and sufficient conditions for the reality of energy eigenvalues”, *Physics Letters A* **374**, 1616–1620 (2010).
- [191] M. Anthony and M. Harvey, *Linear algebra: concepts and methods* (Cambridge University Press, Cambridge, 2012).
- [192] J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, “Networks formed from interdependent networks”, *Nature Physics* **8**, 40–48 (2012).
- [193] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. D. Porcellinis, and R. Setola, “Modelling interdependent infrastructures using interacting dynamical models”, *International Journal of Critical Infrastructures* **4**, 63 (2008).
- [194] C. Nehaniv, P. Dini, P. Van Mieghem, and P. Van Mieghem, *Construction of interdependent networks and efficiency improvements in their stability calculation*, tech. rep. Technical FP7-288021 (Information and Communication Technologies, 2014).
- [195] A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using NetworkX”, in *Proceedings of the 7th python in science conference*, edited by G. Varoquaux, T. Vaught, and J. Millman (2008), pp. 11–15.
- [196] S. Fortunato, M. Boguñá, A. Flammini, and F. Menczer, “Approximating PageRank from In-Degree”, in *Algorithms and models for the Web-Graph*, Vol. 4936, edited by W. Aiello, A. Broder, J. Janssen, and E. Milios (Springer Berlin Heidelberg, Berlin, Heidelberg, 2008), pp. 59–71.

APPENDICES

APPENDIX A

Linear algebra

In this appendix, we provide proofs for a couple of well-established linear algebra theorems referenced in Sec. 9.3 and Sec. 9.4.

A.1 Similar matrices have the same eigenvalues

Theorem A.1. *If there exists a non-singular matrix S such that*

$$A = SBS^{-1}, \tag{A.1}$$

then A and B are similar matrices and share the same eigenvalues.

Proof. Let B satisfy the eigenvalue equation

$$B\mathbf{x}_j = \lambda_j\mathbf{x}_j. \tag{A.2}$$

Pre-multiplying the eigenvalue equation by S ,

$$\begin{aligned} \Rightarrow SB\mathbf{x}_j &= \lambda_jS\mathbf{x}_j \\ \Rightarrow (SBS^{-1})S\mathbf{x}_j &= \lambda_jS\mathbf{x}_j \\ \Rightarrow A(S\mathbf{x}_j) &= \lambda_j(S\mathbf{x}_j), \end{aligned} \tag{A.3}$$

it can be seen that A has eigenvalues λ_j with corresponding eigenvectors $S\mathbf{x}_j$. \square

A.2 An $n \times n$ matrix is diagonalizable iff it has n linearly independent eigenvectors

Theorem A.2. $A \in \mathbb{C}^{n \times n}$ is diagonalizable, that is, there exists a non-singular column matrix P and diagonal matrix $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ such that

$$A = P\Lambda P^{-1}, \quad (\text{A.4})$$

if and only if A has n linearly independent eigenvectors.

Proof. Assume A has n eigenvectors, $|v_1\rangle, |v_2\rangle, \dots, |v_n\rangle$, satisfying the eigenvalue equation

$$A|v_j\rangle = \lambda_j|v_j\rangle \Leftrightarrow (A|v_j\rangle)_i = \lambda_j|v_j\rangle_i. \quad (\text{A.5})$$

Next, construct the matrix $P = [|v_1\rangle |v_2\rangle \cdots |v_n\rangle]$, consisting of the eigenvectors along the columns, and the diagonal matrix $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$.

Now, expanding the matrix product $P\Lambda$ using summation notation,

$$\begin{aligned} (P\Lambda)_{ij} &= \sum_k P_{ik} \Lambda_{kj} = \sum_k |v_k\rangle_i \lambda_k \delta_{kj} \\ &= \sum_k \sum_m A_{im} |v_k\rangle_m \delta_{kj} \\ &= \sum_m A_{im} |v_j\rangle_m \\ &= (AP)_{ij}. \end{aligned}$$

Now, P is non-singular if and only if the columns of P are linearly independent ($\text{rank}(P) = n$); this corresponds to the n eigenvectors of A being linearly independent. If this condition is satisfied, then

$$P\Lambda = AP \Rightarrow \Lambda = P^{-1}AP \quad (\text{A.6})$$

and thus A is diagonalisable. □

A.3 All diagonalisable matrices have a biorthonormal eigenbasis

Theorem A.3. Suppose $A \in \mathbb{C}^{n \times n}$ is a diagonalizable matrix, that is, there exists a non-singular column matrix $P = [|\psi_1\rangle |\psi_2\rangle \cdots |\psi_n\rangle]$ and diagonal matrix $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ such that

$$A = P\Lambda P^{-1}. \quad (\text{A.7})$$

A then has a complete set of biorthonormal eigenvectors $\{|\psi_j\rangle, |\phi_j\rangle\}$ defined by

$$A|\psi_j\rangle = \lambda_j|\psi_j\rangle, \quad (\text{A.8})$$

$$\langle\phi_j|A = \langle\phi_j|\lambda_j \Leftrightarrow A^\dagger|\phi_j\rangle = \lambda_j^*|\phi_j\rangle, \quad (\text{A.9})$$

where λ_n denote the eigenvalues and $j = 1, 2, \dots, n$. The eigenvectors satisfy the orthonormality relation,

$$\langle\phi_i|\psi_j\rangle = \delta_{ij}, \quad (\text{A.10})$$

and are complete over $\mathbb{C}^{n \times n}$,

$$I = \sum_i |\psi_i\rangle \langle\phi_i|. \quad (\text{A.11})$$

Proof. Post-multiply Eq. A.7 by P , giving

$$AP = P\Lambda. \quad (\text{A.12})$$

Writing this using summation notation,

$$\begin{aligned} \Rightarrow \sum_k A_{ik} P_{kj} &= \sum_k P_{ik} \Lambda_{kj} \\ \Rightarrow \sum_k A_{ik} |\psi_j\rangle_k &= \sum_k |\psi_k\rangle_i \lambda_j \delta_{jk} \\ \Rightarrow \sum_k A_{ik} |\psi_j\rangle_k &= \lambda_j |\psi_j\rangle_i, \end{aligned} \quad (\text{A.13})$$

it can be seen that we recover the right eigenvector equation

$$A|\psi_j\rangle = \lambda_j|\psi_j\rangle, \quad j = 1, 2, \dots, n. \quad (\text{A.14})$$

To recover the left eigenvector equation, premultiply Eq. A.7 by the *row* matrix $P^{-1} = [\langle\phi_1|, \langle\phi_2|, \dots, \langle\phi_n|]$, and proceed in a similar fashion:

$$\begin{aligned}
P^{-1}A &= \Lambda P^{-1} \\
\Rightarrow \sum_k P_{ik}^{-1} A_{kj} &= \sum_k \Lambda_{ik} P_{kj}^{-1} \\
\Rightarrow \sum_k \langle\phi_i|_k A_{kj} &= \sum_k \lambda_i \delta_{ik} \langle\phi_k|_j \\
\Rightarrow \sum_k \langle\phi_i|_k A_{kj} &= \langle\phi_i|_j \lambda_i. \tag{A.15}
\end{aligned}$$

Taking the conjugate transpose of both sides, we recover the adjoint eigenvector equation:

$$A^\dagger |\phi_j\rangle = \lambda_j^* |\phi_j\rangle, \quad j = 1, 2, \dots, n. \tag{A.16}$$

To show that the left and right eigenvectors are orthogonal, consider $P^{-1}P = I$ and recall the construction of P and P^{-1} :

$$\begin{aligned}
\Rightarrow \sum_k P_{ik}^{-1} P_{kj} &= \delta_{ij} \\
\Rightarrow \sum_k \langle\phi_i|_k |\psi_j\rangle_k &= \delta_{ij} \\
\Rightarrow \langle\phi_i|\psi_j\rangle &= \delta_{ij}. \tag{A.17}
\end{aligned}$$

To show that the left and right eigenvectors form a complete basis, consider $PP^{-1} = I$:

$$\begin{aligned}
\Rightarrow \sum_k P_{ik} P_{kj}^{-1} &= \delta_{ij} \\
\Rightarrow \sum_k |\psi_k\rangle_i \langle\phi_k|_j &= \delta_{ij} \\
\Rightarrow \sum_k |\psi_k\rangle \langle\phi_k| &= I. \tag{A.18}
\end{aligned}$$

□

A.4 The redefined pseudo-Hermitian inner product preserves the η -CTQW norm

Theorem A.4. *The redefined inner product*

$$\langle \cdots | \cdots \rangle_\eta := \langle \cdots | \eta^2 | \cdots \rangle \quad (\text{A.19})$$

preserves the norm of a pseudo-Hermitian CTQW.

Proof. Let the quantum walk evolve from initial state $|\psi(0)\rangle$ for $t \geq 0$ seconds;

$$|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle. \quad (\text{A.20})$$

Expanding out the inner product in terms of the pseudo-Hermitian biorthonormal eigenbasis ($I = \sum_n |\psi_n\rangle \langle \phi_n|$):

$$\begin{aligned} \langle \psi(t) | \psi(t) \rangle_\eta &= \langle \psi(t) | \eta^2 | \psi(t) \rangle \\ &= \sum_n \langle \psi(t) | \eta^2 | \psi_n \rangle \langle \phi_n | \psi(t) \rangle \\ &= \sum_n \langle \psi(0) | e^{iH^\dagger t} \eta^2 | \psi_n \rangle \langle \phi_n | e^{-iHt} | \psi(0) \rangle \\ &= \sum_n \langle \psi(0) | e^{iH^\dagger t} | \phi_n \rangle \langle \phi_n | e^{-iHt} | \psi(0) \rangle, \end{aligned} \quad (\text{A.21})$$

where we have made use of the fact that $\eta^2 |\psi_n\rangle = |\phi_n\rangle$ on the last line. Substituting in the eigenvalue equation $\langle \phi_n | H = \langle \phi_n | \lambda_n$ (where $\lambda_n \in \mathbb{R}$),

$$\begin{aligned} \langle \psi(t) | \psi(t) \rangle_\eta &= \sum_n e^{i\lambda_n t} e^{-i\lambda_n t} \langle \psi(0) | \phi_n \rangle \langle \phi_n | \psi(0) \rangle \\ &= \sum_n \langle \psi(0) | \phi_n \rangle \langle \phi_n | \psi(0) \rangle \\ &= \langle \psi(0) | \left(\sum_n |\phi_n\rangle \langle \phi_n| \right) | \psi(0) \rangle \\ &= \langle \psi(0) | \eta^2 | \psi(0) \rangle \\ &= \langle \psi(0) | \psi(0) \rangle_\eta. \end{aligned} \quad (\text{A.22})$$

Thus, $\langle \psi(t) | \psi(t) \rangle_\eta = \langle \psi(0) | \psi(0) \rangle_\eta$ for all $t \geq 0$, and hence probability is conserved. \square

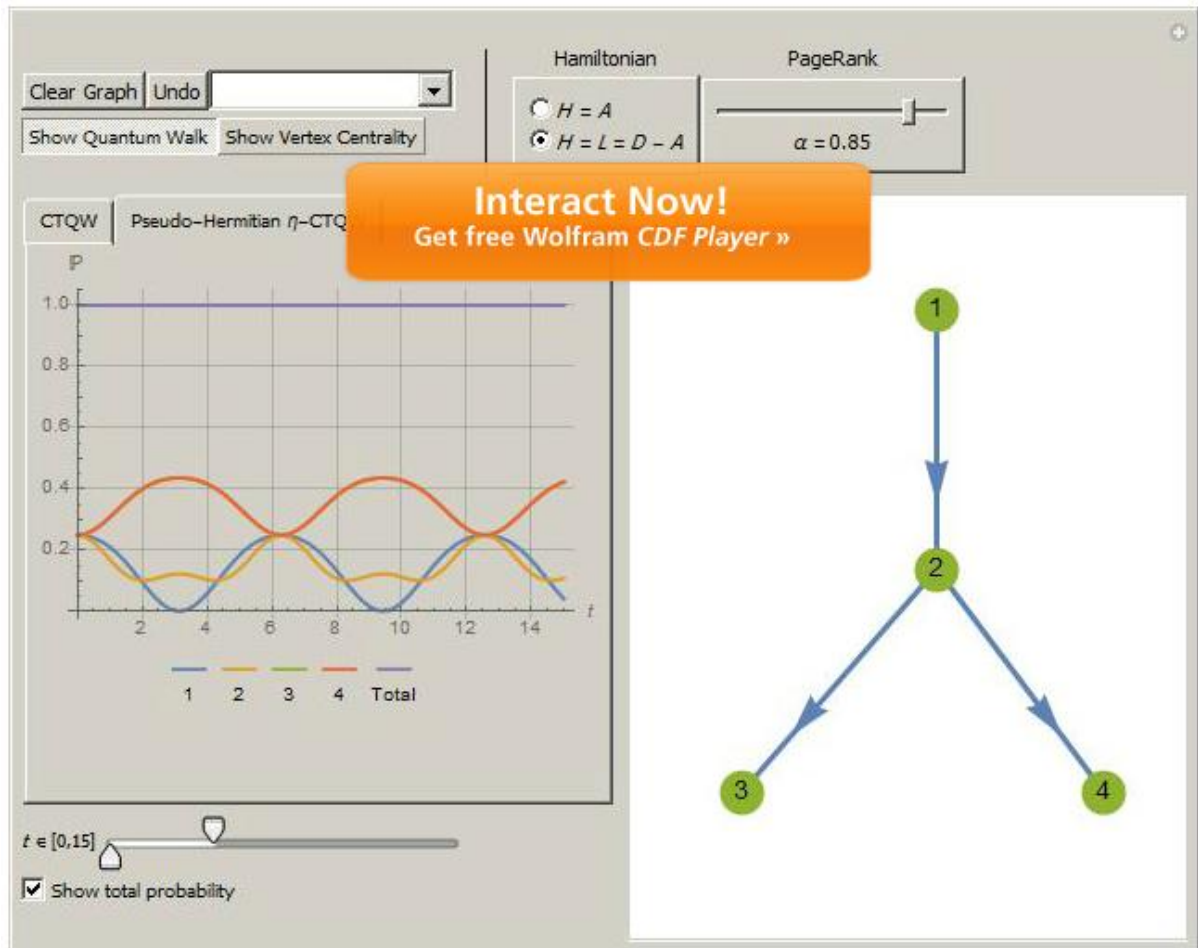
APPENDIX B

Wolfram Demonstration

In this appendix, we provide a copy of the Wolfram Demonstration entitled PT-symmetric Quantum Walks and Centrality Testing on Directed Graphs, completed and submitted as part of the study into CTQW-based network centrality on directed graphs. This demonstration provides an intuitive interactive visualisation of PT-symmetric CTQWs (and the corresponding vertex centrality values) for an arbitrary drawn graph, and may be downloaded via the Wolfram Demonstrations Project website.



PT-symmetric Quantum Walks and Centrality Testing on Directed Graphs



This Demonstration lets you draw a graph using the mouse (or choose a premade graph from the drop-down menu) and then view the continuous-time quantum walk (CTQW) probability distribution. If the graph drawn has direction, then the CTQW probability blows up to infinity. This is avoided, however, for directed graphs with PT-symmetry (pseudo-Hermiticity). You can also choose to view the vertex centrality measurement of the graph, comparing the classical PageRank algorithm, the time-averaged CTQW, and the pseudo-Hermitian CTQW.

To draw the graph, click and drag to create vertices and edges, click to create disjoint vertices, and click an existing vertex to create a self-loop.

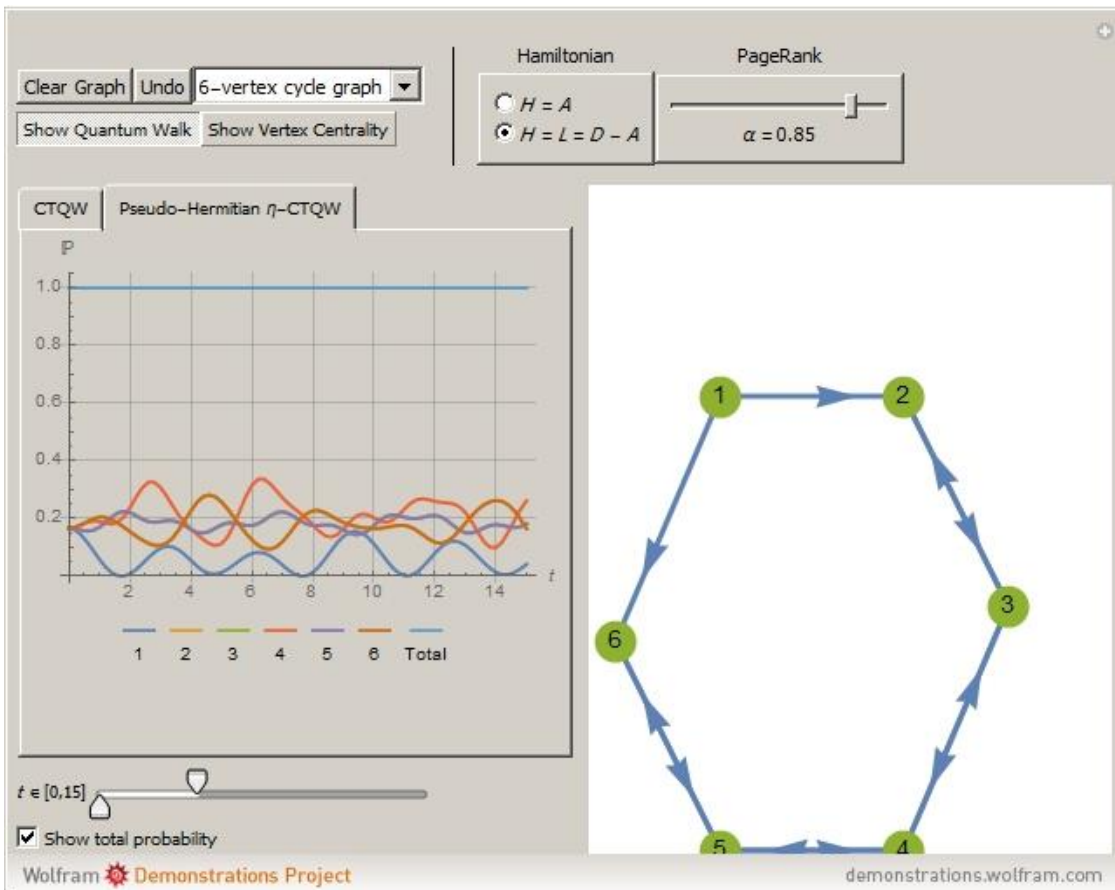
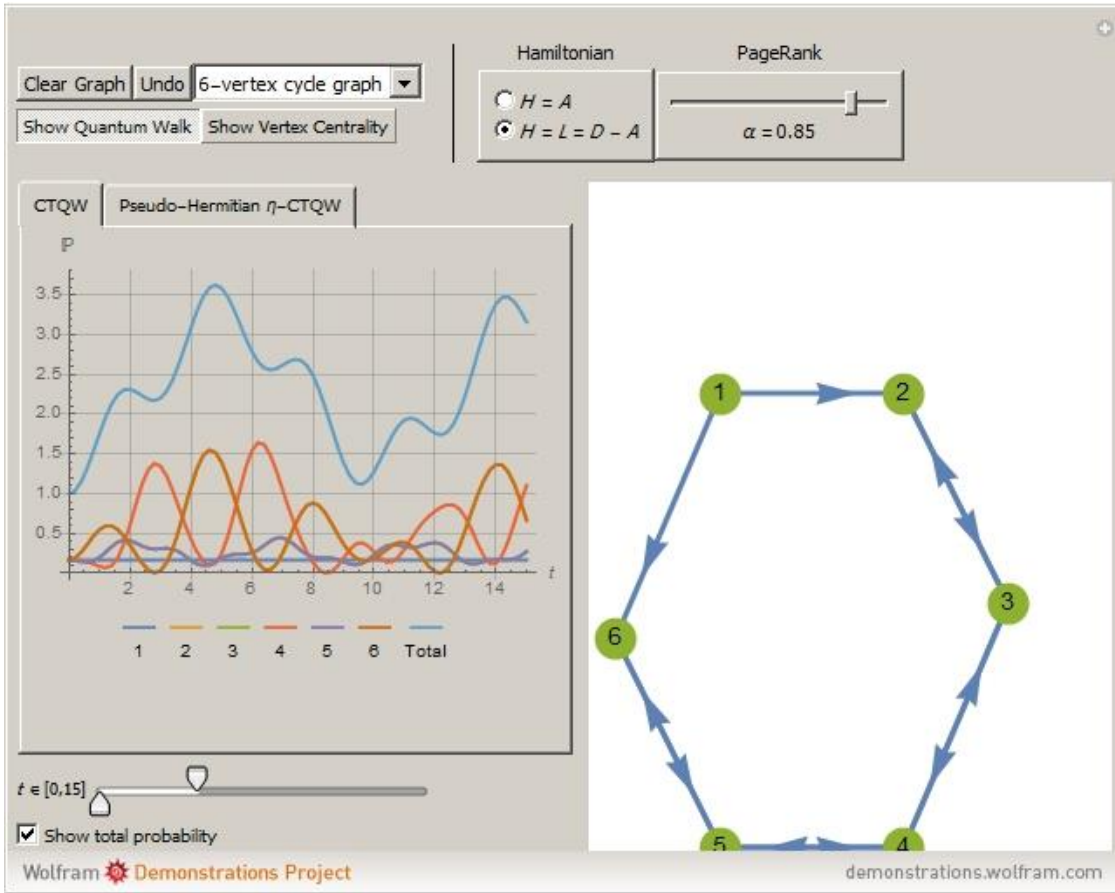
Vertices can be deleted by right-clicking and moved by holding down the CTL/CMD key and dragging.

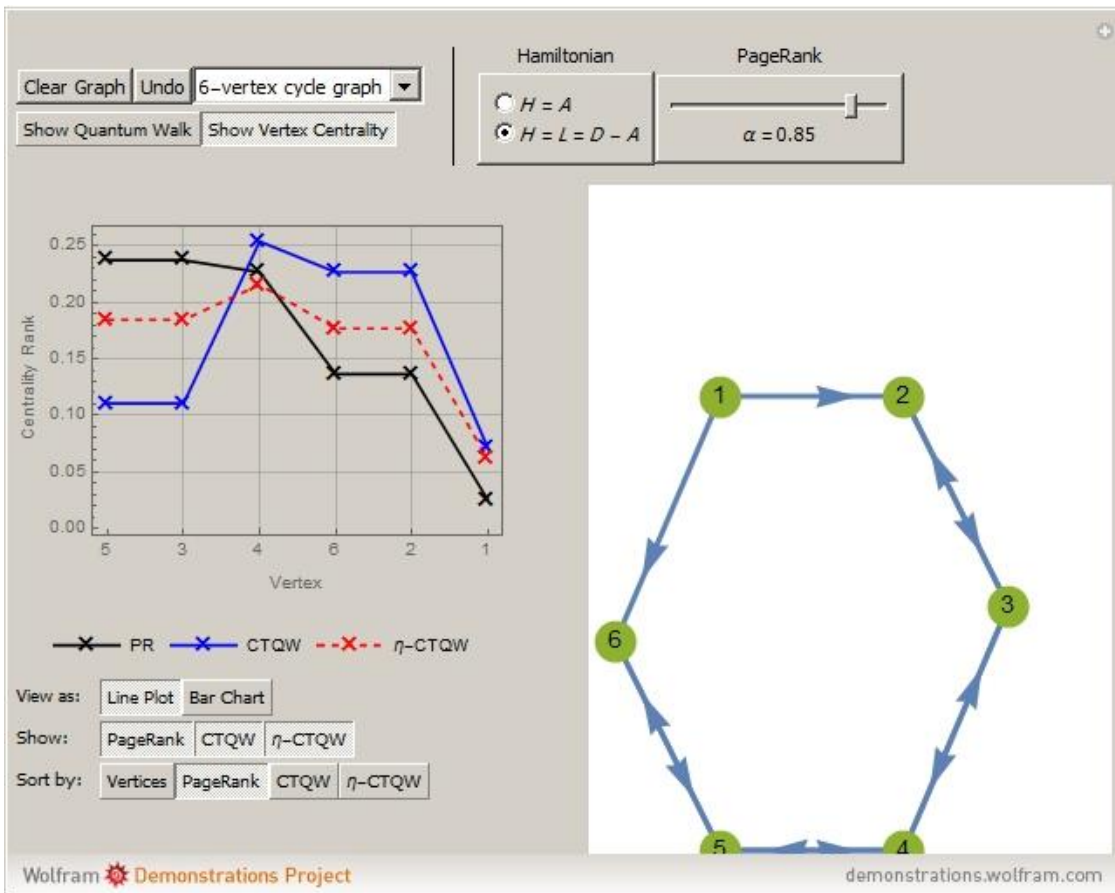
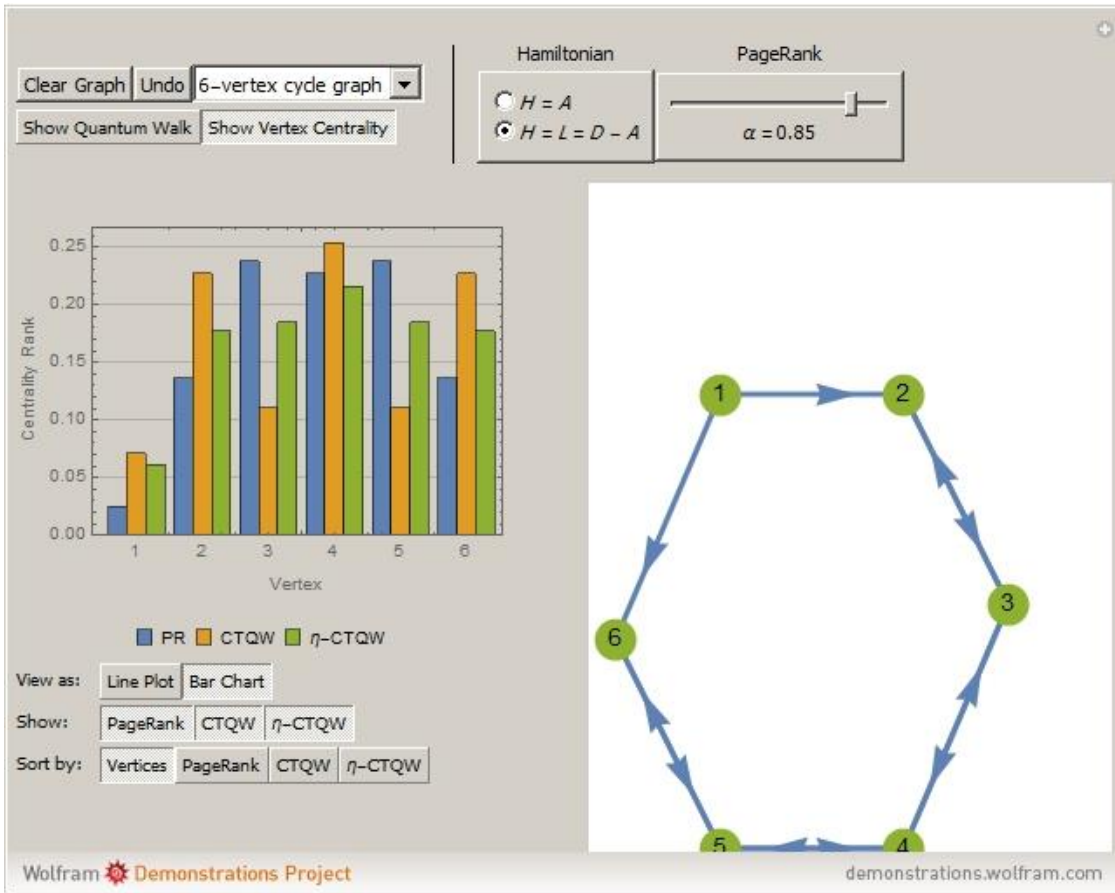
To make a directed edge undirected, draw a new edge along it in the opposite direction.

Contributed by: [Josh Izaac](#)

After work by: [Josh Izaac](#), [Jingbo Wang](#), [Paul C. Abbott](#), and [Xiaosong Ma](#)

SNAPSHOTS





DETAILS

Snapshot 1: The CTQW of the directed six-vertex cycle graph is shown. As the graph is directed, the Hamiltonian is non-Hermitian, and it can be seen that probability is not conserved.

Snapshot 2: This is the pseudo-Hermitian CTQW (η -CTQW) of the same graph; by pseudo-Hermiticity, we can define a similarity transform to allow the walker probability to be conserved.

Snapshot 3: This shows the vertex centrality ranks of the graph using three different measures: (1) the classical PageRank; (2) the non-unitary CTQW; and (3) the unitary η -CTQW.

Snapshot 4: This is similar to snapshot 3, however this time plotted on a line plot, with the x -axis sorted from vertices with highest PageRank (left) to vertices with lowest PageRank (right).

In a classical random walk, when a walker is on a vertex with n possible edges to walk along, the walker flips an n -sided coin to decide which of the n edges to walk down. However, in a quantum walk, the walker utilizes quantum superposition to walk down *all* possible edges. This leads to markedly different properties to the classical walk, and the quantum walk is able to propagate through a graph quadratically faster than the classical walk [1]. In fact, the quantum walk has been shown to be a system of universal quantum computation—any quantum circuit can be reformulated as a quantum walk on a graph [2].

The continuous-time quantum walk on a graph is defined as follows. For a graph $G(V, E)$, composed of vertices $j \in V$ and edges $(i, j) \in E$ and with adjacency matrix A , the Hamiltonian can be given by either $H = A$ or $H_{ij} = (\sum_k A_{ik}) \delta_{ij} - A_{ij}$, depending on the convention preferred (this can be set in the

Demonstration using the Hamiltonian radio buttons). Solving the Schrödinger equation, $i \frac{d}{dt} |\psi\rangle = H |\psi\rangle$ gives the formal solution $|\psi(t)\rangle = e^{-iHt} |\psi_0\rangle$, where $|\psi_0\rangle$ is the initial state, and the state is given by the complex wavefunction $|\psi(t)\rangle = \sum_j \alpha_j(t) |j\rangle$. In this Demonstration, the initial state has been chosen to

be an equal superposition of all vertices; $|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_j |j\rangle$.

If the graph G is directed, then the adjacency matrix A and the Hamiltonian H become nonsymmetric and non-Hermitian. As a consequence, the time-evolution operator $U = e^{-iHt}$ is no longer unitary ($UU^\dagger \neq I$), causing the probability of the walker over time to blow up to infinity. As such, the standard CTQW is unsuited to walks on directed graphs. This can be seen from the Demonstration by viewing plots of the CTQW for a directed graph, for example, the graph $1 \rightarrow 2 \rightarrow 3$.

One solution explored here comes in the form of PT-symmetry [3]. Graphs that exhibit PT-symmetry, while still non-Hermitian and non-unitary, will have real eigenvalues. This results in a total probability that *oscillates* with time—this can be seen by choosing an example of a PT-symmetric graph from the drop-down box in the Demonstration. In order to ensure the probability remains constant with time, we can utilize the "pseudo-Hermiticity" of these graphs by finding an operator η such that $\tilde{H} = \eta H \eta^{-1}$, where \tilde{H} is Hermitian yet retains information about the structure of the graph. This is known as a pseudo-Hermitian CTQW [4], or η -CTQW for short, and can be viewed by clicking the "Pseudo-Hermitian η -CTQW" tab.

Finally, we compare the vertex centrality ranking of the graph vertices using the CTQW, η -CTQW, and the classical PageRank algorithm. For the CTQW and η -CTQW, this is done simply by using the time-average probability for each vertex. The PageRank algorithm was created by Google for ranking their

search results; it finds the fixed points of the so-called Google matrix, defined by $G_{ij} = \alpha \tilde{A}_{ij} + \frac{1}{N} (1 - \alpha)$, where \tilde{A} is the column normalized adjacency matrix, and α is generally chosen to be 0.85. In this Demonstration, you can view the vertex rankings in a line plot or a bar chart, and reorder the plots

based on each ranking algorithm. In the toolbar, there is also a slider allowing you to change the value of α used in the PageRank algorithm.

References

- [1] J. Kempe, "Quantum Random Walks: An Introductory Overview", *Contemporary Physics*, **44**(4), 2003 pp. 307–327. doi:10.1080/00107151031000110776.
- [2] A. M. Childs, "Universal Computation by Quantum Walk," *Physical Review Letters*, **102**(18), 2009. doi:10.1103/PhysRevLett.102.180501.
- [3] C. M. Bender and S. Boettcher, "Real Spectra in Non-Hermitian Hamiltonians Having PT Symmetry," *Physical Review Letters*, **80**(24), 1998 pp. 5243–5246. doi:10.1103/PhysRevLett.80.5243.
- [4] J. Izaac, J. B. Wang, P. C. Abbott, and X. S. Ma, "Quantum Centrality Testing on Directed Graphs via PT-Symmetric Quantum Walks", 2016. arxiv.org/abs/1607.02673.
- [5] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems*, **30**(1–7), 1998 pp. 107–117. doi:10.1016/S0169-7552(98)00110-X.

RELATED LINKS

PageRankCentrality (Wolfram *Mathematica* Documentation Center)
Continuous-Time Quantum Walk (Wolfram Demonstrations Project)
Quantum Random Walk (Wolfram Demonstrations Project)

PERMANENT CITATION

Josh Izaac

"PT-symmetric Quantum Walks and Centrality Testing on Directed Graphs"
<http://demonstrations.wolfram.com/PTSymmetricQuantumWalksAndCentralityTestingOnDirectedGraphs/>
Wolfram Demonstrations Project
Published: July 18, 2016

APPENDIX C

Published articles

In this appendix, all papers published during the course of my PhD candidature have been reproduced, in the order listed in my publication list. Note that the thesis does not comprise of all of the studies produced here; nevertheless, they are provided for completeness. For an outline detailing which papers are relevant to this thesis, please refer to the Statement of Contribution.

Position-defect-induced reflection, trapping, transmission, and resonance in quantum walksZ. J. Li,^{1,2,*} J. A. Izaac,² and J. B. Wang^{2,†}¹*Institute of Theoretical Physics, Shanxi University, Taiyuan, 030006, China*²*School of Physics, The University of Western Australia, Crawley, WA 6009, Australia*

(Received 1 November 2012; published 17 January 2013)

We investigate the scattering properties of quantum walks by considering single and double position defects on a one-dimensional line. This corresponds to introducing, at designated positions, delta potential defects for continuous-time quantum walks and phase-defect Hadamard coins for discrete time quantum walks. The delta potential defects can be readily considered as potential barriers in discrete position space, which affect the time evolution of the system in a similar way as the quantum wave-packet dynamics in a continuous position space governed by Schrödinger's equation. Although there is no direct analogy of potential barriers in the theoretical formulation of discrete time quantum walks, in this paper we show that the phase defects in the coin space can be utilized to provide similar scattering effects. This study provides means of controlling the scattering properties of quantum walks by introducing designated position-dependent defects.

DOI: [10.1103/PhysRevA.87.012314](https://doi.org/10.1103/PhysRevA.87.012314)

PACS number(s): 03.67.Lx, 05.40.Fb, 05.45.Mt

I. INTRODUCTION

Quantum walks are the quantum analog to the classical random walk, extended to take into account superposition, interference, and quantum correlations. It provides a comprehensive framework to study quantum dynamics in discrete and structured space. Compared to the classical random walk, quantum walks exhibit markedly different behavior; for instance, a quantum walk can propagate quadratically faster than its classical counterpart, is a time-reversible process rather than a memoryless Markov process, and results in a probability distribution drastically different from the classically expected Gaussian [1]. As with classical random walks, there are two related but fundamentally different formulations of quantum walks—the discrete-time quantum walk (DTQW) [2] and the continuous-time quantum walk (CTQW) [3]. Due to their unintuitive dynamical behavior, quantum walks have been extensively explored over the past decade, which may provide methods of modeling complex biological systems [4,5] or hold the key to radical new quantum algorithms [1,6–11].

Disorder is unavoidable in most quantum systems. The evolution of quantum walks in a discrete environment with static and dynamic disorders have been studied both theoretically and experimentally, demonstrating a variety of interesting dynamics such as ballistic and diffusive spreading [12–16], as well as Anderson and bound-state localization [15–23]. In CTQW, disorder can be represented by position- or time-dependent potential defects in the diagonal elements of the transition matrix. The position dependence can be chosen as a random distribution within a certain range [19] or as a function distribution such as a Cauchy distribution [15], while time dependence can also be either random or regular [15]. In DTQW, disorder is introduced through the coin operator; the position-dependent coin provides static disorder, while the time-dependent coin gives rise to dynamic disorder [13,23]. In this paper, we study the scattering properties of quantum walks, including reflection, trapping, and resonance transmission for

both CTQW and DTQW with a wide range of defect settings. In particular, we investigate the roles played by the potential defects in CTQW and the phase defects in DTQW with respect to the control of quantum walk behaviors. Furthermore, we extend the work of previous studies (which were primarily concerned with quantum walkers initially localized at a defect) to consider an initially free quantum walk interacting with multiple defects.

In what follows we present, in Sec. II, an introductory overview of CTQW and DTQW, and describe the single and double point-defect model. In Sec. III, we give the evolution properties of QW's for three initial cases: (1) the particle lies at the node containing the defect (the “distinguished node”), (2) the particle lies to the side of the defect, and (3) the particle lies between two defects. In Sec. III D, we investigate resonance transmission when two defects are present with specific separation. Finally, Sec. IV contains our conclusions.

II. QUANTUM WALK POINT-DEFECT MODEL**A. Continuous-time quantum walk**

The continuous-time quantum walk was first posited by Farhi and Gutmann [3], motivated by a desire to establish a general framework to study coherent transport in discrete systems. The continuous-time quantum walk can be regarded as a quantization of the corresponding classical continuous-time random walk, with the system now evolving as per the Schrödinger equation rather than the Markovian master equation. As a result, classical probabilities are replaced by quantum probability amplitudes.

To illustrate, consider a continuous-time random walk on the discrete graph $G(V, E)$, composed of unordered vertices $j \in V$ and edges $e_i = (j, k) \in E$ connecting two vertices j and k . The transition rate matrix H is defined as

$$H_{jk} = \begin{cases} -\gamma_{jk} & \text{for } j \neq k \text{ if node } j \text{ is connected to node } k, \\ 0 & \text{for } j \neq k \text{ if node } j \text{ is not connected to node } k, \\ S_j & \text{for } j = k, \end{cases} \quad (1)$$

*zjli@sxu.edu.cn

†jingbo.wang@uwa.edu.au

where γ_{jk} is the probability per unit time for making a transition from node j to node k and for H to be conservative:

$$S_j = \sum_{k=1, k \neq j}^N \gamma_{jk}. \quad (2)$$

Classically, the state of the random walker is fully described by the probability distribution vector $\mathbf{P}(t)$, with its time evolution governed by the master equation

$$\frac{d\mathbf{P}(t)}{dt} = H\mathbf{P}(t),$$

which has the formal solution $\mathbf{P}(t) = \exp(-Ht)\mathbf{P}(0)$.

Extending the above description to the quantum realm involves replacing the real valued probability distribution vector $\mathbf{P}(t)$ with a complex valued wave function $|\psi(t)\rangle$, and adding the complex notation i to the evolution exponent; i.e.,

$$|\psi(t)\rangle = \exp(-iHt)|\psi(0)\rangle. \quad (3)$$

The quantum transition matrix H , often referred to as the system Hamiltonian, is required to be Hermitian and thus the above time evolution is unitary—guaranteeing that the norm of $|\psi(t)\rangle$ is conserved under CTQWs. The complex-valued state vector $|\psi(t)\rangle = \sum_j a_j(t)|j\rangle$, where $a_j(t) = \langle j|\psi(t)\rangle \in \mathbb{C}$, represents the probability amplitude of the walker being found at node $|j\rangle$ at time t , with $|a_j(t)|^2 = |\langle j|\psi(t)\rangle|^2$ the resulting probability.

For CTQWs on an infinite line, if each node is assumed to be connected only to its neighboring nodes by a constant transition rate $\gamma = 1$, the action of the corresponding Hamiltonian H_0 on the state vector $|\psi(t)\rangle$ leads to the inner product relationship

$$\langle j|H_0|\psi\rangle = 2\langle j|\psi\rangle - \langle j+1|\psi\rangle - \langle j-1|\psi\rangle. \quad (4)$$

Note that this equation is identical to the first-order finite difference approximation of the continuous-space operator $-\nabla^2$, which generates the time evolution of a free particle. In an analogous fashion we can regard the discrete position-space Hamiltonian H_0 given in Eq. (4) as the generator of a *free* CTQW. The significant difference in propagation behavior between these two systems arises due to the use of a discrete position space for the CTQW. It was shown by Manouchehri and Wang [24] that the discreteness of position space in Eq. (4) is a *necessary* condition for a CTQW to display its characteristic propagation behavior, as opposed to continuous-space wave-packet dispersion.

Symmetries that are present in continuous-space quantum systems, for instance invariance under spatial translation for free particles, can also be formulated for discrete space systems such as the continuous-time quantum walk on an infinite line. For instance, consider the discrete translational operator \hat{T}_n , which acts on the set of orthonormal vertex states $|j\rangle$ such that $\hat{T}_n|j\rangle = |j+n\rangle$. This operator is unitary, and as such can be written in the form $\hat{T}_n = e^{i\hat{k}n}$, where \hat{k} is an Hermitian operator and the generator of the translation. In cases where the state of a quantum walker is invariant under spatial translation, the Hermiticity of \hat{k} indicates that its eigenstates $|k\rangle = \sum_j e^{ikj}|j\rangle$ form a complete orthonormal basis, satisfying the eigenvalue equation $H_0|k\rangle = 2(1 - \cos k)|k\rangle$ for $-\pi \leq k < \pi$. These eigenstates are useful when investigating resonance scattering

in defect containing CTQW systems, as they provide a method of producing biased walks (since the initial state $|k\rangle$ has a well-defined “momentum” of k) similar to the continuous-space propagation of a plane wave. This allows its interaction multiple barriers to be considered in an analogous fashion.

Continuous-time quantum walks in the presence of absorbing barriers have previously been studied by Mülken *et al.* [25] and Agliari *et al.* [26]. Alternatively, walks in the presence of reflecting barriers (or defects) have been considered by Keating *et al.* [19] (using a Cauchy distributed defect) and Childs *et al.* [27] (a single defect), in the context of decoherence and algorithmic speedup, respectively. In this paper, we will be primarily concerned with implementing reflecting barriers, and characterizing the resulting behavior of quantum walking systems.

Keeping with the case of the CTQW on an infinite line, let us assume that there are defects present at particular nodes $|m\rangle$. This breaks the translational symmetry, and as such the walker can no longer be considered free. To account for these defects, the Hamiltonian matrix is modified in the following way:

$$H = H_0 + \Gamma, \quad \Gamma = \sum_m \Gamma_m |m\rangle\langle m|, \quad (5)$$

where we have introduced a real diagonal matrix Γ , with positive elements corresponding to defects or reflecting barriers placed at vertex $|m\rangle$ with strength Γ_m . The probability of the walker being found at node $|j\rangle$ at time t can thus be given by $|\langle j|e^{-iHt}|\psi(0)\rangle|^2$. In subsequent sections, we will consider reflecting barriers placed at specific nodes in order to investigate the scattering properties of quantum walks.

B. Discrete-time quantum walk

The discrete-time quantum walk also has a very similar mathematical formalism to that of its classical counterpart, and is implemented by a concatenation of coin operations and successive position shifts. It takes place in the Hilbert space $\mathcal{H}_P \otimes \mathcal{H}_C$ where, in the case of an infinite line, the position Hilbert space \mathcal{H}_P is spanned by the position basis states $|i \in \mathbb{Z}\rangle$ and the “coin” Hilbert space \mathcal{H}_C is spanned by the coin basis states $|c = 0, 1\rangle$. A single step time evolution operator is given by

$$U = \left(\sum_c (|c\rangle\langle c| \otimes S_c) \right) \left(C \otimes \sum_i |i\rangle\langle i| \right), \quad (6)$$

where C is the coin operator, and S_c is a conditional translation operator defined as $S_c = \sum_i |i + (-1)^{c+1}\rangle\langle i|$. As with the continuous-time case, the system is described via the state vector $|\psi(t)\rangle = \sum_i \sum_c f_{ic}(t)|i\rangle \otimes |c\rangle$, with its evolution from initial state $|\psi(0)\rangle$ for discrete time t calculated via repeated use of the unitary operator given by Eq. (6), i.e.,

$$|\psi(t)\rangle = U^t |\psi(0)\rangle.$$

The major point of difference between the classical and quantum implementation of the discrete time walk is the use of a quantum, rather than classical, coin operator—with the result that the walker now has the possibility of being in a superposition of possible coin states $|c\rangle$ at every step. It should be noted that the resulting coherence is a source of most of the counterintuitive behavior of the DTQW; if the coin operator is applied randomly, or if we were to measure the coin state after

each time step, the superposition is destroyed and we recover the classical random walk.

When working with DTQWs, the coin degrees of freedom offer a wide range of control over the evolution of the walk. Of particular interest are position-dependent coin operators, as they act to break down the translational symmetry of the unitary operation in Eq. (6)—thus modification of the DTQW coin operation on a small number of nodes may be used to create an analogous system to the node-defect Hamiltonian in CTQW. It has been shown that the full range of possible DTQW evolutions obtained by different coin operators can also be obtained by fixing the coin operator, and choosing a range of different initial coin states [28]. Without loss of generality, one often restricts the coin operator to one with real coefficients; in the case of an unbiased quantum walk on a line, this leaves the Hadamard coin

$$C_H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

as the only possible choice. Taking this into account, the unitary operation in Eq. (6) is modified as follows:

$$U = \left(\sum_c (|c\rangle\langle c| \otimes S_c) \right) \left(\sum_j C_j (\otimes |j\rangle\langle j|) \right), \quad (7)$$

where the phase-defect Hadamard coin $C_j = e^{i\phi_j} C_H$ is applied at designated “defect nodes,” and $C_j = C_H$ otherwise.

III. LOCALIZATION, TRAPPING, AND REFLECTION

Localization and trapping by a single phase defect in discrete-time quantum walks have been studied in the literature, especially with the quantum walker starting at the defect position [13,23,29]. In this section we analyze the influence of single and double defects on the dynamical evolution of both discrete- and continuous-time quantum walks, with the quantum walker initially located at the origin but the defects at various positions. In particular, we investigate the physical interpretation of these defects in both CTQW and DTQW.

A. Single defect—localization

Let us first consider the case where a defect is present at the origin $|j = 0\rangle$ of an infinite line. Figure 1 shows that both CTQW and DTQW have similar probability distributions, with sharp peaks present at the origin. For comparison, the dashed line depicts the probability distribution of the free quantum walk without the defect. Further investigation shows that the amplitude of the peak, or alternatively the slope of standard deviation with time, is dependent on the strength of the defect potential for the CTQW [Fig. 2(a)]; the stronger the defect potential, the larger the probability amplitude at the origin. The peak in the probability distribution can be understood as bound-state localization, occurring due to the presence of the defect which generically generates a bound state in its surrounding [22,23]. The peak in the probability distribution is the fingerprint of this bound state [30]. Besides this large peak at the origin, two smaller peaks are also observed at the same locations as the ballistic peaks of the free quantum walk—inferring that the linear relationship between the states’

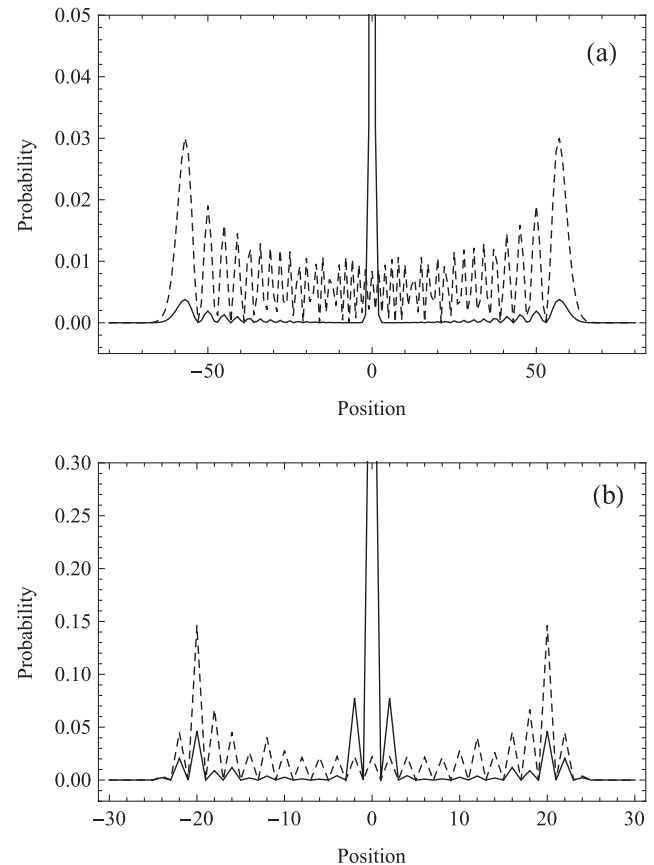


FIG. 1. Probability distribution at time $t = 30$ for (a) CTQW with a defect potential $\Gamma_0 = 5$; (b) DTQW with a phase defect $\phi_0 = \pi$ and initial state $(|1\rangle + i|0\rangle)/\sqrt{2}$. The dashed line corresponds to a quantum walk without a defect.

standard deviation with time, $\sigma \sim t$, remains unchanged in spite of a drastically reduced amplitude for the two smaller peaks.

The similarity of the DTQW and CTQW probability distributions in Fig. 1 suggests the possibility of manipulating the DTQW coin degrees of freedom, in an attempt to reproduce the CTQW physical effects attributed to the point defects. As discussed in Sec. II B, a wide range of possible evolutions of the DTQW can be produced by either varying the initial state (and using a fixed coin operator), or by varying the coin operator. Figure 2(b) displays the probability at the origin of the DTQW as a function of ϕ in the phase-defect Hadamard coin, for the initial coin states $(|1\rangle + i|0\rangle)/\sqrt{2}$, $|1\rangle$, and $|0\rangle$. It can be seen that the amplitude of the sharp peak depends strongly on the initial coin state of the system as well as the phase defect ϕ . For instance, using the initial coin state $(|1\rangle + i|0\rangle)/\sqrt{2}$, no localization is observed for any phase in the domain $\phi \in (0, \pi/4)$, since its overlap with the stationary bound state is close to zero. The probability at the origin is symmetric about $\phi_0 = 0$ for initial coin state $|0\rangle$ and $|1\rangle$, but not for initial state $(|1\rangle + i|0\rangle)/\sqrt{2}$. Also note that, for a given initial state, the phase defect ϕ appears to have the same function as the defect potential Γ in CTQW; both act to change the amplitude of the localization peak centered at the origin.

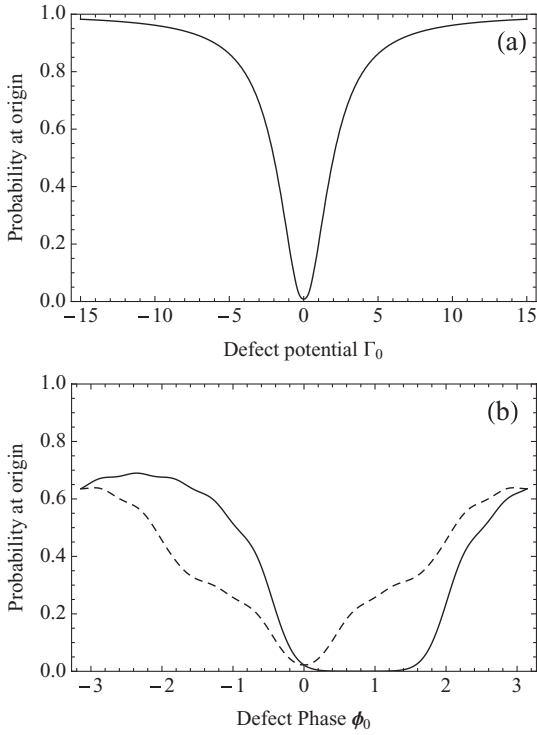


FIG. 2. Probability at the origin at time $t = 30$ as a function of (a) the defect potential Γ_0 for CTQW and (b) phase defect ϕ_0 for DTQW. In (b), the solid line corresponds to the initial coin state $(|1\rangle + i|0\rangle)/\sqrt{2}$, while the dashed line corresponds to the coin initial state $|1\rangle$ or $|0\rangle$.

B. Single defect—reflection

Next, we discuss the case where the single point defect is located away from where the quantum walk starts. As an example, consider the quantum walk initially located at the origin with one defect $\Gamma_{15} = 5$ on an infinite line. The resulting probability distribution over the discrete position space at time $t = 30$ is shown in Fig. 3(a). Some important features to note: (1) it evolves symmetrically in both the left and right direction, which is identical to a “free” quantum walk prior to its interaction with the barrier; (2) upon interacting with the barrier, it is largely reflected with a small probability of transmission; (3) the transmitted component continues to evolve ballistically as per the free quantum walk; (4) the larger the defect potential, the weaker the transmitted amplitude, as shown in Fig. 4(a); and (5) the reflected component interferes with the original propagating component, resulting in a complex interference pattern and asymmetric distribution compared to the free quantum walker.

Figure 3(b) shows very similar propagation behavior in the case of DTQW, now using a phase-defect Hadamard coin $\phi_{15} = \pi$ and starting with initial state $(|1\rangle + i|0\rangle)/\sqrt{2}$. In this case, the transmission amplitude through the reflecting barrier can be manipulated by altering the phase value ϕ_{15} as shown in Fig. 4(b). It can be seen that the transmission amplitude is symmetric about $\phi_{15} = 0$ for initial coin state $|0\rangle$ or $|1\rangle$, but not for initial state $(|1\rangle + i|0\rangle)/\sqrt{2}$.

The above observations are very similar to the wave-packet dynamics in continuous position space when potential barriers

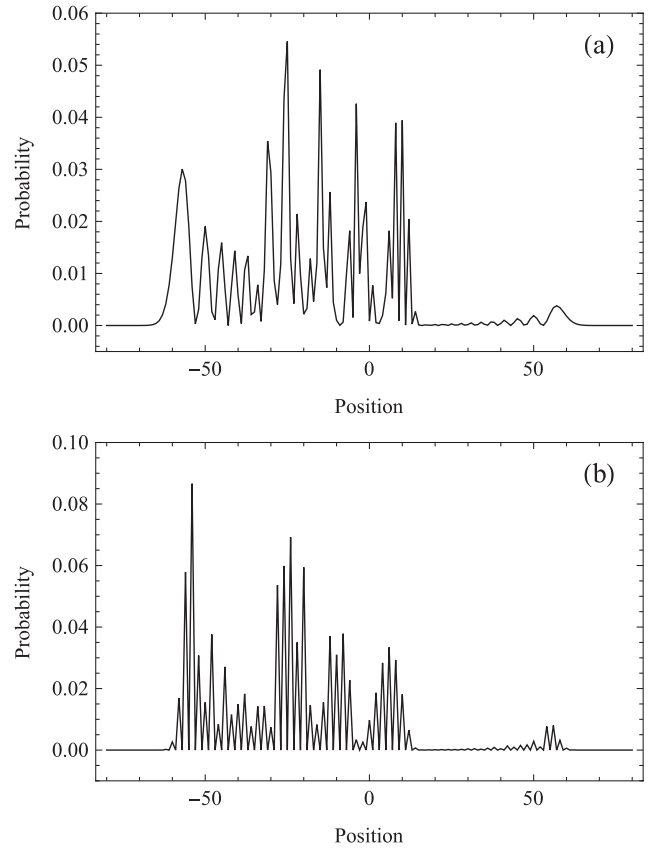


FIG. 3. Probability distribution for (a) CTQW at time $t = 30$ with a defect potential $\Gamma_{15} = 5$, and (b) DTQW after $t = 80$ steps with a phase defect $\phi_{15} = \pi$ and initial coin state $(|1\rangle + i|0\rangle)/\sqrt{2}$.

are applied. In CTQW, the delta potential defects can be readily considered as potential barriers in discrete space, which bring about similar behaviors. Although there is no direct analogy of potential barriers in the theoretical formulation of DTQW, here we show that the additional coin degree of freedom can be utilized to provide similar scattering effects.

C. Double defects—trapping

Another scenario we can consider is to start the quantum walker at the origin between two reflecting barriers at nodes $j = \pm 15$ on an infinite line. The resulting probability distributions for both CTQW and DTQW are shown in Fig. 5. Both Figs. 5(a) and 5(b) demonstrate similar behavior with the probability distribution mostly confined between the two barriers. Smaller group peaks are observed outside the barriers, which are symmetric about the origin, and their amplitudes depend on the strength of the defect potential Γ_j for CTQW and the phase defect ϕ_j for DTQW. It is interesting to note that the gap between the consecutively transmitted group peaks is simply the distance between the two reflecting barriers, which suggests that the walker is reflected each time it interacts with a defect barrier and consequently bounces back and forth in between. As shown in Fig. 6, the total probability of the quantum walker trapped between the two barriers decreases stepwise as the time increases, with almost constant step time corresponding to the trapping time before significant

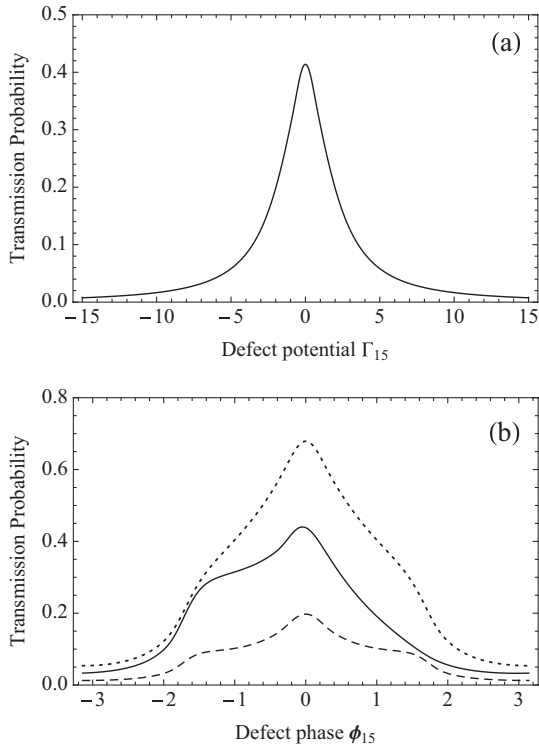


FIG. 4. Transmission probability as a function of (a) the defect potential Γ_{15} for CTQW after time $t = 30$ and (b) the phase defect ϕ for DTQW after $t = 80$ steps. In (b) the solid line corresponds to the coin initial state $(|1\rangle + i|0\rangle)/\sqrt{2}$, whilst the dotted and dashed lines correspond to the coin initial states $|1\rangle$ and $|0\rangle$, respectively.

transmission occurs. The steady emission of group peaks, as shown in Fig. 5, may lead to potential applications such as generating pulses with their magnitudes and time separation controlled by altering the strength and position of the defect barriers.

D. Double defects—resonance

In continuous position space, it has been well established analytically that a quantum particle with energy E incident on double rectangular, cosh, and delta barriers shows complete transmission at particular values of E in the classically forbidden region when $E < V_0$ [31–34]. In this section, we investigate similar resonance behaviors of CTQW in discrete space, in particular on an infinite line with two delta barriers of amplitude α and β placed at vertices $|0\rangle$ and $|L\rangle$. In this case, the Hamiltonian matrix is

$$H = \sum_j (2|j\rangle\langle j| - |j-1\rangle\langle j| - |j+1\rangle\langle j|) + \alpha|0\rangle\langle 0| + \beta|L\rangle\langle L|. \quad (8)$$

Let a momentum eigenstate $|k\rangle$ be incident on the barriers from the left-hand side. The resulting time-independent scattered state $|\psi_s(k)\rangle$ can then be written as [27]

$$|\psi_s(k)\rangle = \begin{cases} |k\rangle + r_1(k)|-k\rangle, & j < 0, \\ t_1(k)|k\rangle + r_2(k)|-k\rangle, & 0 \leq j < L, \\ t_2(k)|k\rangle, & L \leq j, \end{cases} \quad (9)$$

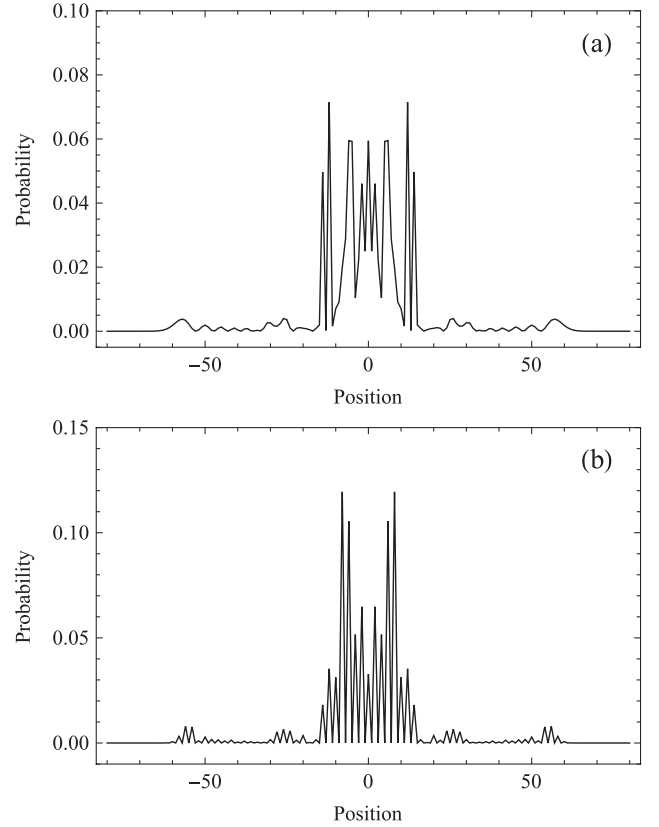


FIG. 5. Probability distribution for a walker initially localized at origin between two barriers for (a) CTQW at time $t = 30$ with the defect potentials $\Gamma_{\pm 15} = 5$, and (b) DTQW after $t = 80$ steps with phase defects $\phi_{\pm 15} = \pi$ and initial state $(|1\rangle + i|0\rangle)/\sqrt{2}$.

where $t_1(k)$ and $t_2(k)$ give the proportion of the momentum eigenstate transmitted through barriers 1 and 2, respectively, and $r_1(k)$, $r_2(k)$ the proportion reflected at each interaction. Recall that, in Sec. II A, it was shown that the orthogonal basis $|k\rangle$ diagonalizes the free Hamiltonian matrix H_0 . In the case of double delta barriers, the set of states $|\psi_s(k)\rangle$, $k \in [-\pi, \pi)$ given in Eq. (9) diagonalizes the Hamiltonian $H = H_0 + \Gamma$. Outside the barrier region (i.e., $j \neq -1, 0, L-1, L$), it can be easily shown that

$$\frac{\langle j|H|\psi_s(k)\rangle}{\langle j|\psi_s(k)\rangle} = 2(1 - \cos k).$$

Inside the barrier region (i.e., $j = -1, 0, L-1, L$), the relationship $E(k) = \langle j|H|\psi_s(k)\rangle/\langle j|\psi_s(k)\rangle = 2(1 - \cos k)$ must also hold, which leads to the following system of equations:

$$t_1(k) + r_2(k) - r_1(k) = 1, \quad (10a)$$

$$\frac{e^{2ik}[r_1(k) + t_1(k)] + r_2(k) + 1}{e^{ik}[r_2(k) + t_1(k)]} - \alpha = 2 \cos k, \quad (10b)$$

$$\frac{(-1 + 2e^{ik})e^{2ikL}t_1(k) - e^{2ik(L+1)}t_2(k) - e^{3ik}(-2 + e^{ik})r_2(k)}{e^{i(2kL+k)}t_1(k) + e^{3ik}r_2(k)} = 2(1 - \cos k), \quad (10c)$$

$$\frac{r_2(k)e^{-2ik(L-1)} + e^{2ik}t_2(k) + t_1(k)}{e^{ik}t_2(k)} - \beta = 2 \cos k. \quad (10d)$$

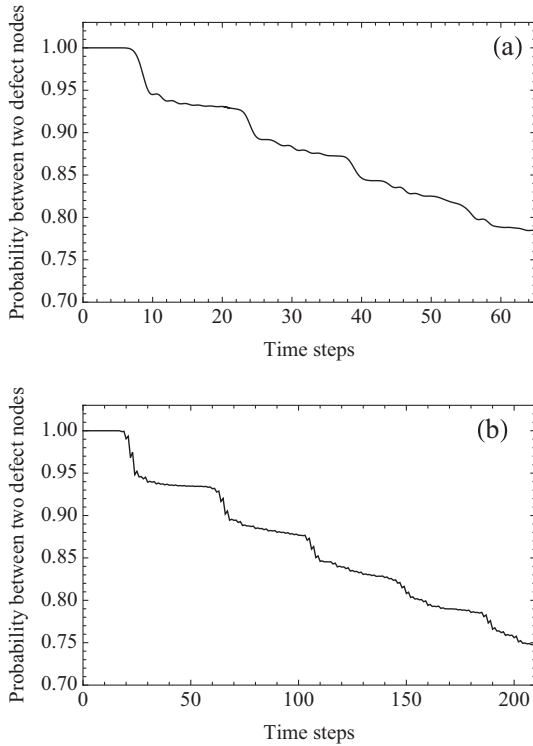


FIG. 6. Trapping probability as a function of time t for (a) CTQW with defect potential $\Gamma_{\pm 15} = 5$, and (b) DTQW with phase defects $\phi_{\pm 15} = \pi$ and initial state $(|1\rangle + i|0\rangle)/\sqrt{2}$.

Equation (10) can be solved to give analytic solutions to $t_1(k)$, $t_2(k)$, $r_1(k)$, and $r_2(k)$ —thus producing a set of states $|\psi_s(k)\rangle$ which are eigenstates of H with eigenvalues $E(k) = 2[1 - \cos(k)]$. The transmission coefficient $T(k) = |t_2(k)|^2$ of a momentum eigenstate on an infinite line, incident on two barriers of amplitude α and β , respectively, and separated by distance $L \in \mathbb{N}_0$, is therefore found to be

$$T(k) = \left[1 + \frac{\csc^2 k}{4}(\alpha^2 + \beta^2 + 2\alpha\beta \cos 2kL) + \frac{\csc^3 k}{4}\alpha\beta(\alpha + \beta) \sin 2kL + \frac{\csc^4 k}{4}\alpha^2\beta^2 \sin^2 kL \right]^{-1}. \quad (11)$$

$$L = \begin{cases} n\pi/k + \frac{2}{k} \tan^{-1} \left[\frac{1}{2} \csc(k)(\alpha + \sqrt{\alpha^2 + 4 \sin^2 k}) \right], & \beta = \alpha, \\ n\pi/k, & \beta = -\alpha, \end{cases} \quad (13)$$

where $L, n \in \mathbb{N}_0$. Resonant peaks are illustrated in Fig. 8 as a function of barrier separation L , in the two cases $\alpha = \beta$ and $\alpha = -\beta$.

Special mention should be made of the case $k = \pi/2$, corresponding to a CTQW momentum eigenstate with the largest possible group velocity $[\partial E(k)/\partial k = 2 \sin k]$. In this instance, the transmission oscillates between two

We also note that a Green's function approach involving transmission and reflection coefficients has been applied to study scattering quantum walks on general graphs, which were shown to be unitarily equivalent to the discrete-time quantum walks [35–38] with the possibility of future work extending this framework to the continuous-time quantum walks. Nevertheless, the above analytical derivation is straightforward and provides the transmission and reflection coefficients in simple forms for quantum walks on the line.

The transmission coefficient is plotted in Fig. 7 over $0 \leq k \leq \pi$ and for various values of L , α , and β . Oscillating behavior is clearly visible, with the frequency of oscillation increasing rapidly as L increases. It is also observed that when $\beta = 0$ and $L = 0$ (i.e., there is only a *single* barrier of amplitude α) the resonance behavior vanishes—multiple barriers are a necessary condition for CTQW resonance on the infinite line. When $\alpha = 0$ as well, then we are simply observing transmission of a momentum eigenstate in the case of no defects ($H = H_0$) and $T(k) = 1 \forall k$.

By solving the Schrödinger equation with Hamiltonian $H = -\nabla^2 + \alpha\delta(x) + \beta\delta(x + L)$ in continuous space, it can be shown [39,40] that the transmission coefficient is given by

$$T(k) = \left[1 + \frac{\alpha^2\beta^2}{4k^4} \sin kL^2 + \frac{1}{4k^3} 2\alpha\beta(\alpha + \beta) \sin 2kL + \frac{1}{4k^2}(\alpha^2 + \beta^2 + 2\alpha\beta \cos 2kL) \right]^{-1}. \quad (12)$$

The above equation is identical to Eq. (11) when taking the limit $\csc k \rightarrow 1/k$ (the first-order approximation of $\sin k$). The significance of this is twofold. First, with this transformation, the eigenvalue equation $\hat{H}|k\rangle = 2(1 - \cos k)|k\rangle$ becomes $\hat{H}|k\rangle = k^2|k\rangle$, and we recover the energy of a plane wave in continuous space. Second, this relationship highlights the similarity of resonant behavior between quantum dynamical systems and continuous-time quantum walks. The marked differences between the two systems are a result of the finite domain of k , or equivalently the discrete nature of the CTQW position space.

Returning to discrete space, of particular interest is the case $\beta = |\alpha|$, where oscillating resonant behavior is now possible. Solving for $T(k) = 1$ with $\beta = |\alpha|$ in Eq. (11), we find that the distances between defects at which resonance occurs are given by the set

values for successive values of L . Note that when $\alpha = \beta$, there is never perfect transmission; the values oscillate between $T(\pi/2) = 1/(\alpha^2 + 1)$ and $T(\pi/2) = 1/(1 + \alpha^4/4)$ for even and odd L , respectively. However, when $\alpha = -\beta$, perfect transmission occurs for even L , with odd values of L resulting in a reduced transmission of $T(\pi/2) = 1/[1 + \alpha^2(\alpha^2 + 4)/4]$.

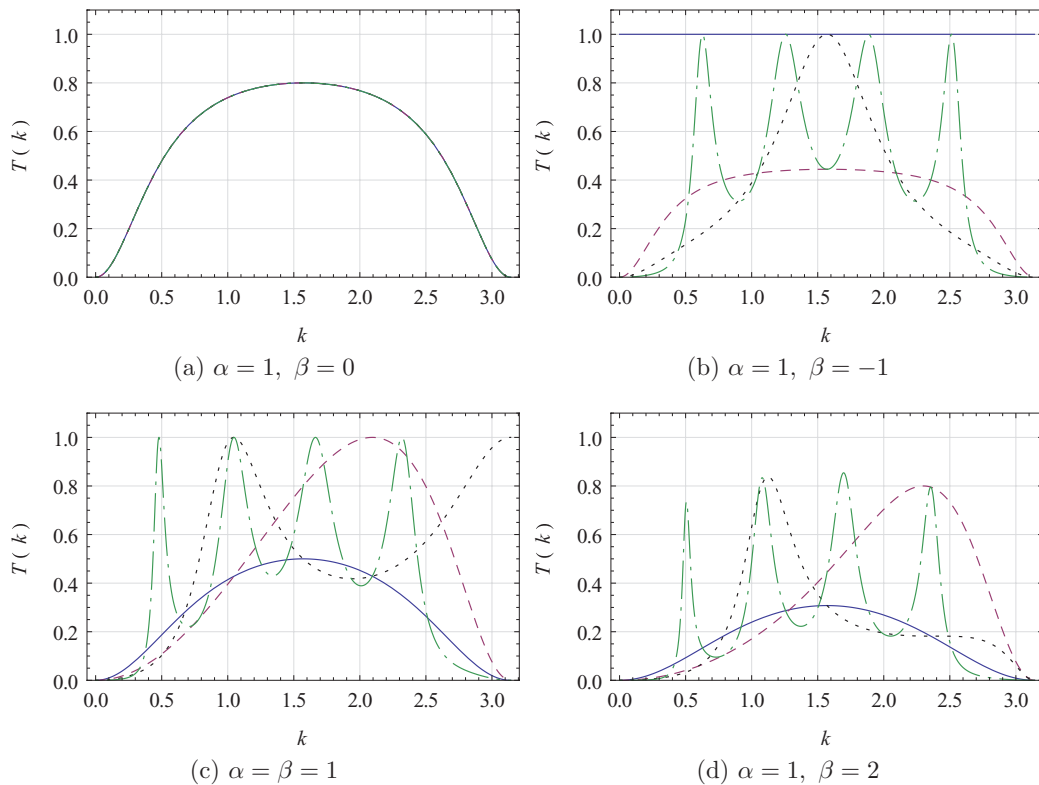


FIG. 7. (Color online) Transmission coefficient vs momentum of a CTQW momentum eigenstate incident on two defects of amplitude α and β , separated by distances $L = 0$ (solid, blue), $L = 1$ (dashed, red), $L = 2$ (dotted, black), and $L = 5$ (dot-dashed, green). The results are shown for the cases (a) $\alpha = 1, \beta = 0$ (i.e., a single defect), (b) $\alpha = 1, \beta = -1$, (c) $\alpha = \beta = 1$, and (d) $\alpha \neq |\beta|, \alpha, \beta \neq 0$.

Next consider the boundaries $k \rightarrow 0$ and $k \rightarrow \pi$. In this case, if $\alpha = -\beta$, transmission occurs only when the two defects overlap and therefore “cancel” each other out, resulting in a free CTQW. When $\alpha = \beta$, we have the following: if $\alpha > 0$, then $\lim_{k \rightarrow \pi} T(k) = \delta_{\alpha, 2/L}$ and $\lim_{k \rightarrow 0} T(k) = 0$; if $\alpha < 0$, then $\lim_{k \rightarrow 0} T(k) = \delta_{\alpha, -2/L}$ and $\lim_{k \rightarrow \pi} T(k) = 0$.

For other values of k , various forms of “enveloping” in the transmission occur as shown in Fig. 8 for the case $k = 1.45$.

As the CTQW momentum eigenstates $|k\rangle$ form a complete set for the discrete position space, any arbitrary initial state can be decomposed into momenta components—opening the possibility for artificially placed multiple defects

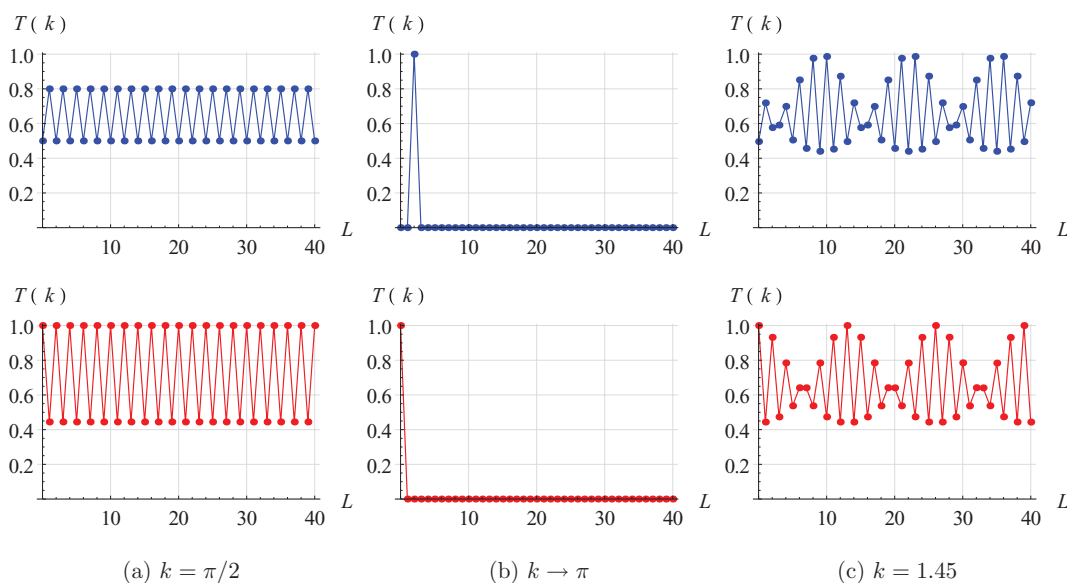


FIG. 8. (Color online) Transmission coefficient $T(k)$ as a function of barrier separation L for momentum eigenstate $|k\rangle$ incident on a double reflecting barrier. Top: $\alpha = \beta = 1$; bottom: $\alpha = -\beta = 1$.

in CTQW systems to selectively control transmission and reflection.

IV. CONCLUSIONS

The quantum walk formalism provides a powerful framework to study the dynamics of quantum particles in a structured discrete position space, while its time evolution can be either discrete (CTQW) or continuous (DTQW). Although the theoretical and physical models of CTQW and DTQW are fundamentally different, their characteristic propagation behavior are often similar. In this paper, detailed comparisons were made between the two models by propagating the quantum walker in both continuous and discrete time on an infinite line in the presence of single or double “defects.”

As expected, the delta defect potentials in CTQW cause localization, reflection, transmission, and trapping in a similar way as the quantum wave-packet dynamics in a continuous position space governed by Schrödinger’s equation. However, there is no direct analogy of potential barriers in the theoretical formulation of discrete-time quantum walks. In this paper, we demonstrate that adding phase defects in the coin degree of freedom in DTQW plays the same role as the potential barriers in CTQW. We also demonstrate that the effect of altering the phase parameter ϕ in the discrete-time quantum walk is equivalent to changing the barrier strength (Γ) in the continuous-time case. Other definitions of coin phase defects can be equally effective, opening up a wide range

of possibilities for the design of efficient quantum-walk-based algorithms.

Moreover, a detailed derivation of the transmission coefficient for the case of a CTQW with a double-point defect is provided, and contrasted to the continuous-space case. It was shown that resonance behavior well established in quantum dynamics extends to the CTQW, with marked differences contributed to the discrete nature of the position space; in particular, under certain conditions only one value of defect separation permits perfect transmission, with perfect reflection occurring for all other integer values of separation.

The effects of disorder and defects on quantum walks are an important field of study, highly relevant in the case of experimental realizations—particularly those with inherent imperfections, or quantum processes which may be unavoidably affected by the environment. As a result of this research, we hope to provide methods to control the spreading of quantum walks, through the use of artificial defects which act to break translational invariance; this could also prove useful in fields such as quantum information processing.

ACKNOWLEDGMENTS

Z.J.L. acknowledges support from the National Nature Science Foundation of China (Grant No. 10974124) and Shanxi Scholarship Council. J.A.I. thanks the John and Patricia Farrant foundation for financial support. We also thank the anonymous referee for providing very constructive comments and suggestions.

-
- [1] J. Kempe, *Contemp. Phys.* **44**, 307 (2003).
 - [2] Y. Aharonov, L. Davidovich, and N. Zagury, *Phys. Rev. A* **48**, 1687 (1993).
 - [3] E. Farhi and S. Gutmann, *Phys. Rev. A* **58**, 915 (1998).
 - [4] R. B. Sessions, M. Oram, M. D. Szczelkun, and S. E. Halford, *J. Mol. Biol.* **270**, 413 (1997).
 - [5] M. Mohseni, P. Rebentrost, S. Lloyd, and A. Aspuru-Guzik, *J. Chem. Phys.* **129**, 174106 (2008).
 - [6] N. Shenvi, J. Kempe, and K. Birgitta Whaley, *Phys. Rev. A* **67**, 052307 (2003).
 - [7] B. L. Douglas and J. B. Wang, *J. Phys. A* **41**, 075303 (2008).
 - [8] A. M. Childs, *Phys. Rev. Lett.* **102**, 180501 (2009).
 - [9] D. Reitzner, M. Hillery, E. Feldman, and V. Buzek, *Phys. Rev. A* **79**, 012323 (2009).
 - [10] S. D. Berry and J. B. Wang, *Phys. Rev. A* **83**, 042317 (2011).
 - [11] S. D. Berry and J. B. Wang, *Phys. Rev. A* **82**, 042333 (2010).
 - [12] N. V. Prokofev and P. C. E. Stamp, *Phys. Rev. A* **74**, 020102(R) (2006).
 - [13] N. Konno, *Quantum Inf. Process.* **9**, 405 (2010).
 - [14] P. Törmä, I. Jex, and W. P. Schleich, *Phys. Rev. A* **65**, 052110 (2002).
 - [15] Y. Yin, D. E. Katsanos, and S. N. Evangelou, *Phys. Rev. A* **77**, 022302 (2008).
 - [16] A. Schreiber, K. N. Cassemiro, V. Potocek, A. Gabris, I. Jex, and C. Silberhorn, *Phys. Rev. Lett.* **106**, 180403 (2011).
 - [17] P. W. Anderson, *Phys. Rev.* **109**, 1492 (1958).
 - [18] P. Ribeiro, P. Milman, and R. Mosseri, *Phys. Rev. Lett.* **93**, 190503 (2004).
 - [19] J. P. Keating, N. Linden, J. C. F. Matthews, and A. Winter, *Phys. Rev. A* **76**, 012315 (2007).
 - [20] A. Joye and M. Merkli, *J. Stat. Phys.* **140**, 1025 (2010).
 - [21] A. Ahlbrecht, V. B. Scholz, and A. H. Werner, *J. Math. Phys.* **52**, 102201 (2011).
 - [22] A. Ahlbrecht, A. Alberti, D. Meschede, V. B. Scholz, A. H. Werner, and R. F. Werner, *New J. Phys.* **14**, 073050 (2012).
 - [23] A. Wójcik, T. Łuczak, P. Kurzyński, A. Grudka, T. Gdala, and M. Bednarska-Bzdega, *Phys. Rev. A* **85**, 012329 (2012).
 - [24] K. Manouchehri and J. B. Wang, *J. Phys. A* **40**, 13773 (2007).
 - [25] O. Mülken, A. Blumen, T. Amthor, C. Giese, M. Reetz-Lamour, and M. Weidemüller, *Phys. Rev. Lett.* **99**, 090601 (2007).
 - [26] E. Agliari, A. Blumen, and O. Mülken, *Phys. Rev. A* **82**, 012305 (2010).
 - [27] A. M. Childs, E. Farhi, and S. Gutmann, *Quantum Inf. Process.* **1**, 3543 (2002).
 - [28] B. Tregenna, W. Flanagan, R. Maile, and V. Kendon, *New J. Phys.* **5**, 83 (2003).
 - [29] M. J. Cantero, F. A. Grunbaum, L. Moral, and L. Velazquez, *Rev. Math. Phys.* **24**, 1250002 (2012).
 - [30] Reviewer’s comment.
 - [31] P. E. Falloon and J. B. Wang, *Comput. Phys. Commun.* **134**, 167 (2001).

- [32] H. Yamamoto, Y. Kanie, M. Arakawa, and K. Taniguchi, *Appl. Phys. A: Solids Surf.* **50**, 577 (1990).
- [33] K. Manouchehri and J. B. Wang, *J. Comput. Theor. Nanos.* **3**, 249 (2006).
- [34] A. Dutt and S. Kar, *Am. J. Phys.* **78**, 1352 (2010).
- [35] F. M. Andrade and M. G. E. da Luz, *Phys. Rev. A* **84**, 042343 (2011).
- [36] A. G. M. Schmidt, B. K. Cheng, and M. G. E. da Luz, *Phys. Rev. A* **66**, 062712 (2002).
- [37] E. Feldman and M. Hillery, *Phys. Lett. A* **324**, 277 (2004).
- [38] E. Feldman and M. Hillery, *J. Phys. A: Math. Theor.* **40**, 11343 (2007).
- [39] I. Yanetka, *Physica B: Condens. Matter* **270**, 371 (1999).
- [40] I. Yanetka, *Acta Phys. Pol. A* **116**, 1059 (2009).

Continuous-time quantum walks with defects and disorder

J. A. Izaac,^{*} J. B. Wang,[†] and Z. J. Li*School of Physics, The University of Western Australia, Crawley WA 6009, Australia*

(Received 6 August 2013; published 29 October 2013)

With the advent of physical implementations of quantum walks, a general theoretical and efficient numerical framework is required for the study of their interactions with defects and disorder. In this paper, we derive analytic expressions for the eigenstates of a one-dimensional continuous-time quantum walk interacting with a single defect, before investigating the effects of multiple diagonal defects and disorder, with emphasis on its transmission and reflection properties. Complex resonance behavior is demonstrated, showing alternating bands of zero and perfect transmission for various defect parameters. Furthermore, we provide an efficient numerical method to characterize quantum walks in the presence of diagonal disorder, paving the way for selective control of quantum walks via the optimization of position-dependent defects. The numerical method can be readily extended to higher dimensions and multiple interacting walkers.

DOI: [10.1103/PhysRevA.88.042334](https://doi.org/10.1103/PhysRevA.88.042334)

PACS number(s): 03.67.Lx, 05.40.Fb, 05.45.Mt

I. INTRODUCTION

Since the seminal paper by Aharonov *et al.* [1] establishing a quantum analogy of the classical walk, quantum walks have constituted an important tool in quantum information theory; for example, by motivating the creation of quantum algorithms that are faster and more efficient than their classical analogues [2–8] and by providing methods of universal quantum computation [9–11]—a highly sought-after goal of modern physics. This is a consequence of the markedly different behavior exhibited by quantum walks: by taking into account superposition, interference, and quantum correlations, the quantum walkers propagate quadratically faster than their classical counterpart and result in a probability distribution drastically different from the classically expected behavior [2]. As with classical random walks, there are two related but fundamentally different formulations of the quantum walk: the discrete-time quantum walk (DTQW) and the continuous-time quantum walk (CTQW). While these are related through well-defined limits in the classical case, their relation in the quantum realm is highly nontrivial, as shown by Strauch [12]. In this paper, we will focus on the continuous-time quantum walk, with emphasis on its scattering behavior in the presence of disorder and defects.

Quantum walks have proven incredibly versatile in terms of theoretical applications, with uses ranging from implementing quantum algorithms to modeling complex quantum systems. In order to benefit from these newfound ideas, physical implementations are essential; some recent approaches include the use of waveguides and photonics [13–16] and ion lattices [17–19]. With physical implementations of quantum walks comes the issue of disorder and decoherence affecting the sought after quantum behavior. In a precursor to modern quantum walking systems, the limiting case of a single diagonal defect in a one-dimensional molecular crystal was explored quantitatively by Koster and Slater [20] using tight-binding methods and difference equations, and later extended to take into account nearest-neighbor interactions (resulting

in very effective exciton traps) [21]. Some earlier works by Dean [22] and Thouless [23] also looked at multiple defects in relation to the density and distribution of eigenstates. Such a tight-binding lattice model has found applications in a wide variety of fields. For example, Avgin and Huber [24] applied the one-dimensional, single impurity model to study defects in polyfluorenes.

More recently, the theoretical effects of random disorder in quantum walks have been considered by Yin *et al.* [25], Schreiber *et al.* [26], and Mülken and Blumen [27] (the latter also considering the effects of nonunitary “traps”). Disorder and decoherence, however, may provide additional tools in constructing quantum walks for particular applications—for instance, Keating *et al.* [28] considered the application of disorder-induced Anderson localization in quantum communication. In this work, the point-defect model of diagonal disorder, which is similar to that of Koster and Slater [20], will be used to derive expressions for the CTQW eigenstates for transmission through a single defect. This will then be extended to provide transmission amplitudes through multiple defects and, in particular, highlight resonant and bandlike structures. Furthermore, a general numerical method will be developed that efficiently calculates transmission information for an arbitrary distribution of diagonal defects, allowing a detailed study of defect-induced selective transmission of continuous-time quantum walkers.

This paper is structured as follows. In Sec. II, we introduce the mathematical formalism behind continuous-time quantum walks and, in particular, the diagonal point-defect model. Analytic expressions for the single-defect eigenstates are presented in Sec. III. Expressions for CTQW transmission through multiple, equally spaced defects are then derived in Sec. IV and used to verify the results of a general numerical method for arbitrary defect distributions, which is detailed in Sec. V. Finally, our conclusions are provided in Sec. VI.

II. CTQW DIAGONAL DEFECT MODEL

Continuous-time quantum walks were first introduced by Farhi and Gutmann [29] in 1998 as an extension of the classical theory of Markov processes. While research into continuous-time quantum walks has not been as extensive

^{*}josh.izaac@uwa.edu.au[†]wang@physics.uwa.edu.au

as the discrete-time case, some applications that have arisen include efficient spatial search algorithms that achieve a speedup of \sqrt{N} over classical counterparts [4], exploring topological structure [30], and modeling coherent transport on complex networks [27] (such as mass and energy transport in complex molecular structures). Furthermore, recent experimental evidence for energy transfer through quantum coherence in photosynthetic and other biochemical systems [31–34] suggests that continuous-time quantum walks can be extended to model biological systems—potentially providing new insights into the natural world.

The continuous-time quantum walk can be regarded as a quantization of the corresponding classical continuous-time random walk, with the system now evolving as per the Schrödinger equation rather than the Markovian master equation. As a result, classical probabilities are replaced by quantum probability amplitudes. To illustrate, consider a continuous-time random walk on the discrete graph $G(V, E)$, composed of unordered vertices $j \in V$ and edges $e_i = (j, k) \in E$ connecting two vertices j and k . The transition rate matrix H is defined as

$$H_{jk} = \begin{cases} -\gamma_{jk} & \text{for } j \neq k \text{ if node } j \text{ is connected to node } k \\ 0 & \text{for } j \neq k \text{ if node } j \text{ is not connected to node } k \\ S_j & \text{for } j = k, \end{cases} \quad (1)$$

where γ_{jk} is the probability per unit time for making a transition from node j to node k and for H to be conservative,

$$S_j = \sum_{k=1, k \neq j}^N \gamma_{jk}. \quad (2)$$

Classically, the state of the random walker is fully described by the probability distribution vector $\mathbf{P}(t)$, with its time evolution governed by the master equation

$$\frac{d\mathbf{P}(t)}{dt} = H\mathbf{P}(t),$$

which has the formal solution $\mathbf{P}(t) = \exp(-Ht)\mathbf{P}(0)$.

Extending the above description to the quantum realm involves replacing the real-valued probability distribution vector $\mathbf{P}(t)$ with a complex-valued wave function $|\psi(t)\rangle$ and adding the complex notation i to the evolution exponent, i.e.,

$$|\psi(t)\rangle = \exp(-iHt)|\psi(0)\rangle. \quad (3)$$

The quantum transition matrix H , often referred to as the system Hamiltonian, is required to be Hermitian and thus the above time evolution is unitary—guaranteeing that the norm of $|\psi(t)\rangle$ is conserved under CTQWs. The complex-valued state vector $|\psi(t)\rangle = \sum_j a_j(t)|j\rangle$, where $a_j(t) = \langle j|\psi(t)\rangle \in \mathbb{C}$, represents the probability amplitude of the walker being found at node $|j\rangle$ at time t , with $|a_j(t)|^2 = |\langle j|\psi(t)\rangle|^2$ the resulting probability.

For CTQWs on an infinite line, if each node is assumed to be connected only to its neighboring nodes by a constant transition rate $\gamma = 1$, then the action of the corresponding Hamiltonian H_0 on the state vector $|\psi(t)\rangle$ leads to the inner product relationship

$$\langle j|H_0|\psi\rangle = 2\langle j|\psi\rangle - \langle j+1|\psi\rangle - \langle j-1|\psi\rangle. \quad (4)$$

Symmetries that are present in continuous-space quantum systems, for instance invariance under spatial translation for free particles, can also be formulated for discrete-space systems. This symmetry allows us to define the momentum eigenstate $|k\rangle$: a complete orthonormal basis of the Hamiltonian, satisfying the eigenvalue equation $H_0|k\rangle = 2(1 - \cos k)|k\rangle$ for $-\pi \leq k < \pi$. Analogous in function to the momentum eigenstates encountered in continuous-space quantum mechanics, these are an important tool in studying scattering properties in discrete space and, as such, have been described in detail by Childs *et al.* [3], Farhi *et al.* [10], Mülken and Blumen [27], Childs and Gosset [11], Mülken *et al.* [35], and Childs *et al.* [36] (albeit with slight variations in definition). For example, consider a continuous-time quantum walk on an infinite line, scattering off a defect placed at node $|d\rangle$. To account for these defects, the Hamiltonian matrix is modified in the following way:

$$H = H_0 + \Gamma, \quad \Gamma = \sum_m \Gamma_m |m\rangle\langle m|, \quad (5)$$

where we have introduced a real diagonal matrix Γ , with $m \in \mathbb{Z}, m \in \{d\}$ representing the set of vertices associated with a defect of strength Γ_m . The probability of the walker being found at node $|j\rangle$ at time t can thus be given by $|\langle j|e^{-iHt}|\psi(0)\rangle|^2$.

Now, let the quantum walker be initialized in momentum eigenstate $|k\rangle$ incident from the left; this results in a time-independent scattered state of the form

$$|\psi_s\rangle = \hat{U}|k\rangle = \begin{cases} |k\rangle + r(k)|-k\rangle, & j \leq d \\ t(k)|k\rangle, & j > d \end{cases} \quad (6)$$

(the *Bethe ansatz*), where $t(k)|k\rangle$ and $r(k)|-k\rangle$ are the transmitted and reflected components, respectively. It was shown by Childs *et al.* [3] in the context of algorithmic speedup that, given $|\psi_s(k)\rangle$ remains an eigenstate of $H = H_0 + \Gamma$, a pair of linearly independent equations are produced which uniquely determine $t(k)$ and $r(k)$. This was further extended by Farhi *et al.* [10] in order to calculate the transmission probability due to finite trees and semi-infinite lines attached at singular nodes. Finally, it was demonstrated by Li *et al.* [37] that, in the presence of double diagonal defects, a CTQW system exhibits resonance behavior determined inherently by the nature of the discrete space. In successive sections, we will determine analytic expressions for the complete set of eigenstates for a single defect and relate this to group velocity. Moreover, we will show through analytical derivations that this previously established resonant behavior exists in the case of multiple sets of defects, and describe an efficient numerical method for exploring systems with arbitrary distributions of diagonal defects.

III. SINGLE-DEFECT EIGENSTATES

Using the diagonal defect model defined above for a CTQW containing a single-point defect on an infinite line, the Hamiltonian can therefore be written as

$$H = \sum_j (2|j\rangle\langle j| - |j-1\rangle\langle j-1| - |j+1\rangle\langle j+1|) + \alpha|d\rangle\langle d|. \quad (7)$$

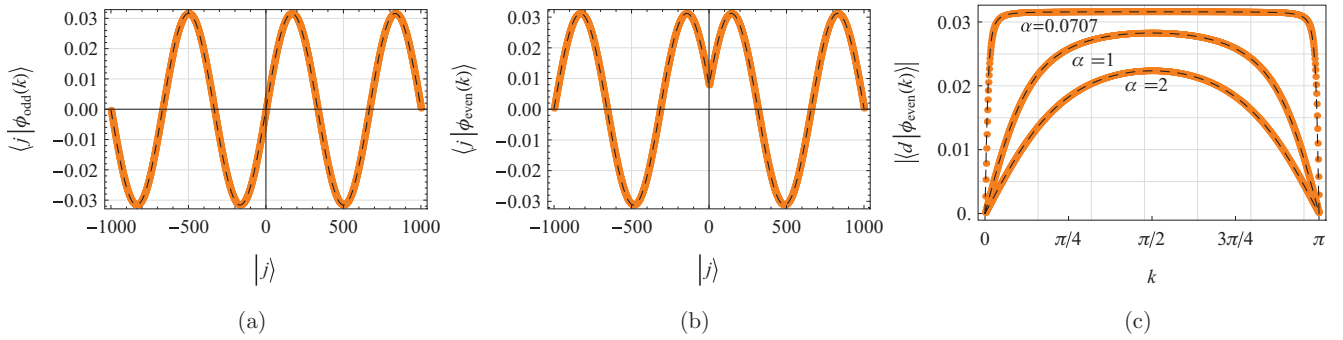


FIG. 1. (Color online) The numeric (orange solid line) and analytic (black dashed line) single-point-defect eigenstates are plotted for the set of (a) odd continuous eigenstates and (b) even continuous eigenstates. (c) The value of the even states at the defect location $|0\rangle$ for defect strength $\alpha = 1/\sqrt{200}, 1, 2$.

For simplicity, and without loss of generality, choose $d = 0$. Now let $|\phi\rangle$ denote the eigenstates of the Hamiltonian, with associated eigenvalues $\lambda \in \mathbb{R}$; i.e., $H|\phi\rangle = \lambda|\phi\rangle$. Expanding $\langle j|H|\phi\rangle$ explicitly using Eq. (7) and the position state decomposition of the eigenstates, $|\phi\rangle = \sum_{j'} |j'\rangle \langle j'|\phi\rangle = \sum_{j'} c_{j'} |j'\rangle$, we arrive at the following recurrence relation:

$$\begin{aligned} \langle j|H|\phi\rangle &= 2c_j - c_{j-1} - c_{j+1} + \alpha c_0 \delta_{j0} \\ &= \lambda c_j \Rightarrow (2 - \lambda)c_j - c_{j-1} - c_{j+1} = \alpha c_0 \delta_{j0}. \end{aligned} \quad (8)$$

We now have an inhomogeneous, linear recurrence equation with constant coefficients—this equation fully determines the eigenstates and eigenvalues of the system. By substituting in the ansatz $c_j = r^j$ [20], we readily see that the general homogeneous solution is

$$\begin{aligned} c_j &= Ar_-^j + Br_+^j \\ &= 2^{-j} A(2 - \lambda - \sqrt{\lambda - 4\sqrt{\lambda}}) \\ &\quad + 2^{-j} B(2 - \lambda + \sqrt{\lambda - 4\sqrt{\lambda}})^j, \end{aligned} \quad (9)$$

where A and B are as yet undetermined functions of α and λ . Using this result as a basis to calculate inhomogeneous solutions to Eq. (8), we must consider regimes where both oscillatory and bound solutions exist.

A. Continuous eigenstates

To determine the continuous eigenstates, consider oscillating solutions of Eq. (9). It is clear that a necessary requirement for oscillating solutions is $|r_{\pm}| \leq 1$; this is satisfied only for $0 \leq \lambda \leq 4$. For convenience, the parametrization $\lambda = 2(1 - \cos k)$, $0 \leq k \leq \pi$, can be used, resulting in a homogeneous solution of the form

$$c_j = Ae^{-ikj} + Be^{ikj}. \quad (10)$$

The method of undetermined coefficients can now be used to calculate the inhomogeneous solutions to the system in this λ regime. We find that two solutions exist, with odd and even symmetry around the defect, respectively:

$$\begin{aligned} \langle j|\phi_{\text{odd}}(k)\rangle &= \frac{1}{\sqrt{\pi}} \sin kj, \quad 0 \leq k \leq \pi, \\ \langle j|\phi_{\text{even}}(k)\rangle &= \frac{1}{\sqrt{\pi}} t_{\alpha}(k) \left(\frac{1}{2} \alpha \csc k \sin k|j| + \cos kj \right), \\ &0 \leq k \leq \pi, \end{aligned} \quad (11)$$

where $t_{\alpha}(k) = 1/[1 + \frac{1}{2}i\alpha \csc k]$ is simply the transmission coefficient of the CTQW incident on a single defect. Also recall that, by construction, both eigenstates satisfy the eigenvalue equation

$$H|\phi(k)\rangle = \lambda(k)|k\rangle, \quad \lambda(k) = 2(1 - \cos k)|k\rangle \quad (12)$$

(as is expected, the eigenvalues are of the same form as Koster and Slater [20] and Merrifield [21] in the case where the lattice parameter $\alpha \rightarrow 1$). Comparing these results to those obtained by numerical analysis (Fig. 1) verifies that these eigenstates do, in fact, represent the complete set of continuous eigenstates. Furthermore, it can be seen that setting $\alpha = 0$ does, indeed, recover the free-space eigenstates $|k\rangle$, $k \in [-\pi, \pi]$.

B. Bound states

For bound-state $|\phi_B\rangle$ solutions to Eq. (8) to be physical, it is required that $\langle j|\phi_B\rangle \rightarrow 0$ as $j \rightarrow \pm\infty$. Restricting our attention to $\lambda < 0$, $\lambda > 4$, the method of undetermined coefficients is applied in the case of odd symmetry and even symmetry ($j \rightarrow |j|$). As in the continuous case, both odd and even bound states exist, with respective conditions $A + B = 0$ and $A = B(\sqrt{\lambda - 4\sqrt{\lambda}} - \alpha)/(\sqrt{\lambda - 4\sqrt{\lambda}} + \alpha)$.

By taking into account the boundary condition, the odd bound state turns out to be nonphysical [$r_-(j) - r_+(j)$ diverges as $j \rightarrow \pm\infty$] and must be discarded. The even bound state, of the form

$$\begin{aligned} \langle j|\phi_B\rangle &= Ar_-(|j|) + Br_+(|j|) \\ &= Br_-(|j|) \frac{\sqrt{\lambda - 4\sqrt{\lambda}} - \alpha}{\sqrt{\lambda - 4\sqrt{\lambda}} + \alpha} + Br_+(|j|), \end{aligned} \quad (13)$$

contains only one divergent term [$r_-(|j|)$] and thus represents a physical bound state when the coefficient of $r_-(|j|)$ is zero for all values of α ; thus, $\lambda = 2 + \alpha\sqrt{1 + 4/\alpha^2}$. Substituting this back into Eq. (13) and normalizing, the bound state of the system is therefore given by

$$\langle j|\phi_B\rangle = \frac{2^{-|j|} \sqrt{|\alpha|}}{(\alpha^2 + 4)^{1/4}} \left(\alpha - \alpha\sqrt{1 + \frac{4}{\alpha^2}} \right)^{|j|}, \quad (14)$$

satisfying the eigenvalue equation

$$H|\phi_B\rangle = \lambda_{\alpha}|\phi_B\rangle, \quad \lambda_{\alpha} = 2 + \alpha\sqrt{1 + \frac{4}{\alpha^2}}. \quad (15)$$

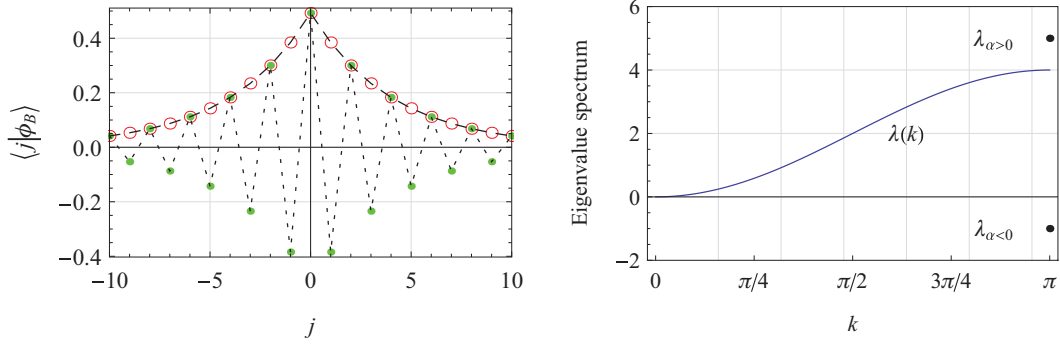


FIG. 2. (Color online) Left: The numeric and analytic single-point defect bound states are plotted and compared for the cases (i) $\alpha < 0$ with the analytic result in long-dashed black lines and the numerical data points in red open circles, and (ii) $\alpha > 0$ with the analytic result in short-dashed black lines and the numerical data points in green dots. Right: The complete eigenvalue spectrum of the Hamiltonian with a single-point defect.

Some properties of the single-defect bound state that can be ascertained from the analytic expression include (1) in the limit $|\alpha| \rightarrow \infty$, $\langle j - d | \phi_B \rangle \rightarrow \delta_{jd}$ and $\lambda_\alpha \rightarrow \infty$ (i.e., the contribution of the bound state to the time evolution of a CTQW outweighs that of the continuous eigenstates for large α) and (2) in the limit $\alpha \rightarrow 0$, $\langle j - d | \phi_B \rangle \rightarrow 0$ and $\lim_{\alpha \rightarrow 0^-} \lambda_\alpha = 0$, $\lim_{\alpha \rightarrow 0^+} \lambda_\alpha = 4$ (i.e., the contribution of the bound state to the time evolution of a CTQW is approximately the trivial solution for small α and can be neglected). It should also be noted that for $\alpha > 0$, the bound-state amplitude shows damped oscillating behavior; this is not the case for $\alpha < 0$ (Fig. 2). However, in both cases, the probability distributions $|\langle v_j | \phi_B \rangle|^2$ are equal.

C. Eigenstate completeness and time evolution

Using the orthogonality relations of the sine and cosine functions, it can be shown that the eigenstates calculated above remain orthonormal with respect to each other; that is, for all values of $0 \leq k \leq \pi$ and $\alpha \in \mathbb{R}$, $\langle \phi_{\text{odd}}(k) | \phi_B \rangle = 0$, $\langle \phi_{\text{even}}(k) | \phi_B \rangle = 0$, and $\langle \phi_{\text{odd}}(k) | \phi_{\text{even}}(k) \rangle = 0$. Coupling these results with the completeness of Hermitian eigenstates, the identity operator for the single defect containing discrete space can be written as

$$\hat{I} = \int_0^\pi dk (|\phi_{\text{odd}}(k)\rangle\langle\phi_{\text{odd}}(k)| + |\phi_{\text{even}}(k)\rangle\langle\phi_{\text{even}}(k)|) + |\phi_B\rangle\langle\phi_B|, \quad (16)$$

and thus the time-evolution operator $\hat{U}(t) \equiv e^{-iHt} = e^{-iHt} \hat{I}$ has the form

$$\hat{U}(t) = \int_0^\pi dk e^{-2it(1-\cos k)} [|\phi_{\text{odd}}(k)\rangle\langle\phi_{\text{odd}}(k)| + |\phi_{\text{even}}(k)\rangle\langle\phi_{\text{even}}(k)|] + e^{-i\lambda_\alpha t} |\phi_B\rangle\langle\phi_B|. \quad (17)$$

This integral form of the time-evolution operator now enables us to construct an expression for the time evolution of an arbitrary state $|\psi\rangle$ from time t to t' ,

$$\langle j' | e^{-iHt'} |\psi(t)\rangle = \sum_j \int_0^\pi dk G_{\alpha,d}(j', t'; j, k) \langle j | \psi(t)\rangle, \quad (18)$$

where

$$G_{\alpha,d}(j', t'; j, k) = \frac{1}{\pi} e^{-2it'(1-\cos k)} \left[\sin k(j' - d) \sin k(j - d) + \frac{1}{4} |t_\alpha(k)|^2 \epsilon(j' - d) \epsilon(j - d) \right] + e^{-i\lambda_\alpha t} \delta(k - 1) \phi_B(j - d) \phi_B(j' - d), \quad (19)$$

with $t_\alpha(k) = 1/[1 + (i\alpha/2) \csc k]$, $\epsilon(j) = \alpha \csc k \sin k|j| + 2 \cos kj$, $\lambda_\alpha = 2 + \alpha\sqrt{1 + 4/\alpha^2}$, $\phi_B(j) = \frac{2^{-|j|}\sqrt{|\alpha|}}{(\alpha^2 + 4)^{1/4}} (\alpha - \alpha\sqrt{1 + 4/\alpha^2})^{|j|}$, $|d\rangle$ the location of the defect, and α the defect amplitude.

Using this integral approach, $\langle j' | e^{-iHt'} |\psi\rangle$ is plotted in the case of $\alpha = 1/\sqrt{2}$, $d = 5$, $|\psi\rangle = |v_1\rangle$, and $t' = 20$ in Fig. 3. It can be seen that the integral and the matrix exponential method are in excellent agreement.

The Green's function offers other advantages compared to the previously considered one. For instance, it is now well defined in the case of the initial state being at the vertex containing the defect, and the form of the expression is the same over all space (i.e., there are no piecewise components and no need to separate the position space into a “transmitted region,” “reflected region,” etc.). Numerous advantages also exist compared to the matrix exponential method: the time evolution can now be explored on a vertex by vertex basis (reducing computation time as we no longer need to consider the entire discrete subspace), numerical integration is less computationally expensive and easier to implement, and asymptotic approximations of the integral can be used to characterize behavior for large or small t and α , among others.

IV. MULTIPLE DEFECTS

Using the same mathematical scaffolding discussed in Sec. II and extending the work of Li *et al.* [37] to multiple defects, the transmission coefficient for N point defects, located at vertices $d_0, \dots, d_i, \dots, d_{N-1}$, can be calculated by

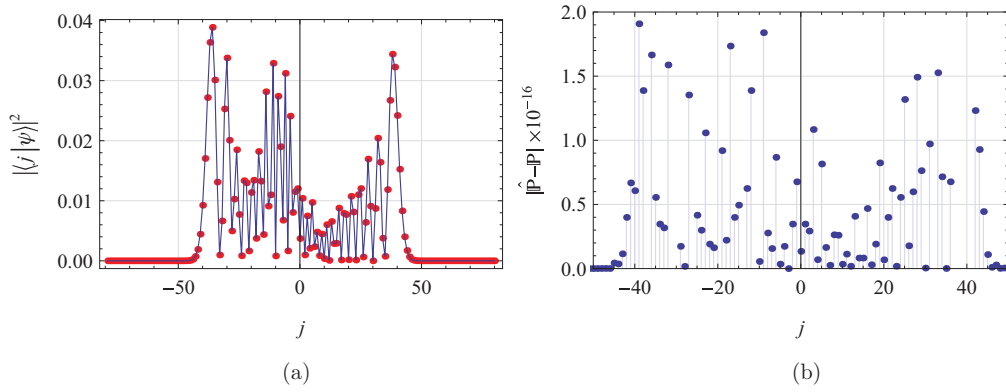


FIG. 3. (Color online) (a) The time evolution of the initial state $|1\rangle$ for time $t' = 20$, with a defect placed at $d = 5$ with strength $\alpha = 1/\sqrt{2}$, using the matrix exponential definition (blue line) and the integral approach (18) (red data points). (b) A plot of the absolute error between both methods.

starting with the scattered state ansatz,

$$|\psi_s(k)\rangle = \begin{cases} |k\rangle + r_1(k)|-k\rangle, & j < d_0 \\ \vdots \\ t_i(k)|k\rangle + r_{i+1}|-k\rangle, & d_{i-1} < j < d_i \\ \vdots \\ t_N(k)|k\rangle, & d_{N-1} \leq j. \end{cases} \quad (20)$$

The Hamiltonian of the system is now given by

$$H = \sum_j (2|j\rangle\langle j| - |j-1\rangle\langle j-1| - |j+1\rangle\langle j+1|) + \sum_{i=0}^{N-1} \alpha_i |d_i\rangle\langle d_i|, \quad (21)$$

and so the eigenvalue condition,

$$\frac{\langle j|H|\psi_s(k)\rangle}{\langle j|\psi_s(k)\rangle} = 2(1 - \cos k)$$

(which must hold for all $j \in \mathbb{Z}$ and $-\pi \leq k < \pi$), can then be applied to all possible regions described by Eq. (20). As before, this produces a set of $2N$ linearly independent linear equations relating $t_1(k), \dots, t_N(k), r_1(k), \dots, r_N(k)$, which can then be solved to calculate all the transmission and reflection amplitudes. Note that there is the added caveat that as N increases, the complexity of the analytic form of $t_i(k)$ and $r_i(k)$ appears to drastically increase—suggesting that for N significantly large, analytic approximations or numerical methods may be preferable.

Using this method, the transmission coefficient was calculated in the case of the 2, 3, 5, and 8-point defects, respectively. For convenience, allowing $d_{i+1} - d_i = L$ and $\alpha_i = \alpha \forall i$ (i.e., all barriers have equal separation and amplitude), the transmission amplitude is given by

$$t_N(k) = \frac{1}{1 + \omega_N} \Rightarrow T_N(k) = |t_N(k)|^2 = \frac{1}{1 + \Omega_N}, \quad (22)$$

where

$$\omega_2 = i\alpha \csc k + \frac{1}{4}\alpha^2(e^{2ikL} - 1)\csc^2 k, \quad (23a)$$

$$\omega_3 = -\frac{1}{8}i\alpha e^{2ikL} \csc^3 k [2 \cos 2kL(\alpha^2 - 2i\alpha \sin k - 3) + 4\alpha \sin k(i - 2 \sin 2kL) - 3i \sin 2k(L+1) + 6i \sin 2kL - 3i \sin 2k(L-1) + 3 \cos 2k(L-1) + 3 \cos 2k(L+1) - 2\alpha^2], \quad (23b)$$

and thus

$$\Omega_2 = \frac{1}{4}\alpha^2 \csc^4 k (\alpha \sin kL + 2 \sin k \cos kL)^2, \quad (24a)$$

$$\Omega_3 = \frac{1}{16}\alpha^2 \csc^6 k [(\alpha^2 - 2) \cos 2kL - 4\alpha \sin k \sin 2kL + \cos 2k(L-1) + \cos 2k(L+1) + \cos 2k - \alpha^2 - 1]^2. \quad (24b)$$

The analytic form of $T(k)$ for the 5- and 8-point defect is too long and complex to be reproduced here. The transmission coefficients for $N = 2, 3, 5$, and 8 are plotted and compared in Fig. 4 for barrier strength $\alpha = 1$ and separations of $L = 0, 1, 2$, and 5. It can be seen that as $N \rightarrow \infty$, the local minimums of $T(k)$ approach zero, while the resonant peaks widen. Furthermore, oscillation amplitudes appear to decrease, and the transition between regions of perfect and zero transmission becomes much sharper. Qualitatively, this appears indicative of the electronic band structure observed in continuous-space models of crystal lattices. The results are also visualized in Fig. 5, showing the effect of defect amplitude α on the transmission coefficient for the case $N = 8$. It is easily seen that, by altering α , a method is provided to control the placement and size of the unity transmission band.

Ultimately, however, this method of calculating the transmission for N defects becomes unwieldy for large N , as the increasing number of linearly independent equations required to calculate $r_i(k), t_i(k)$ scales by $O(2N)$ and results in drastically larger computation times. Further, we are usually only interested in calculating $r_1(k)$ and $t_N(k)$; this motivates the creation of an algorithm to more efficiently characterize CTQW transmission through multiple arbitrary defects.

V. A GENERAL NUMERICAL APPROACH FOR MULTIPLE DEFECTS

In the case of continuous-space quantum mechanics, the Fourier method approach utilized by Yiu and Wang [38],

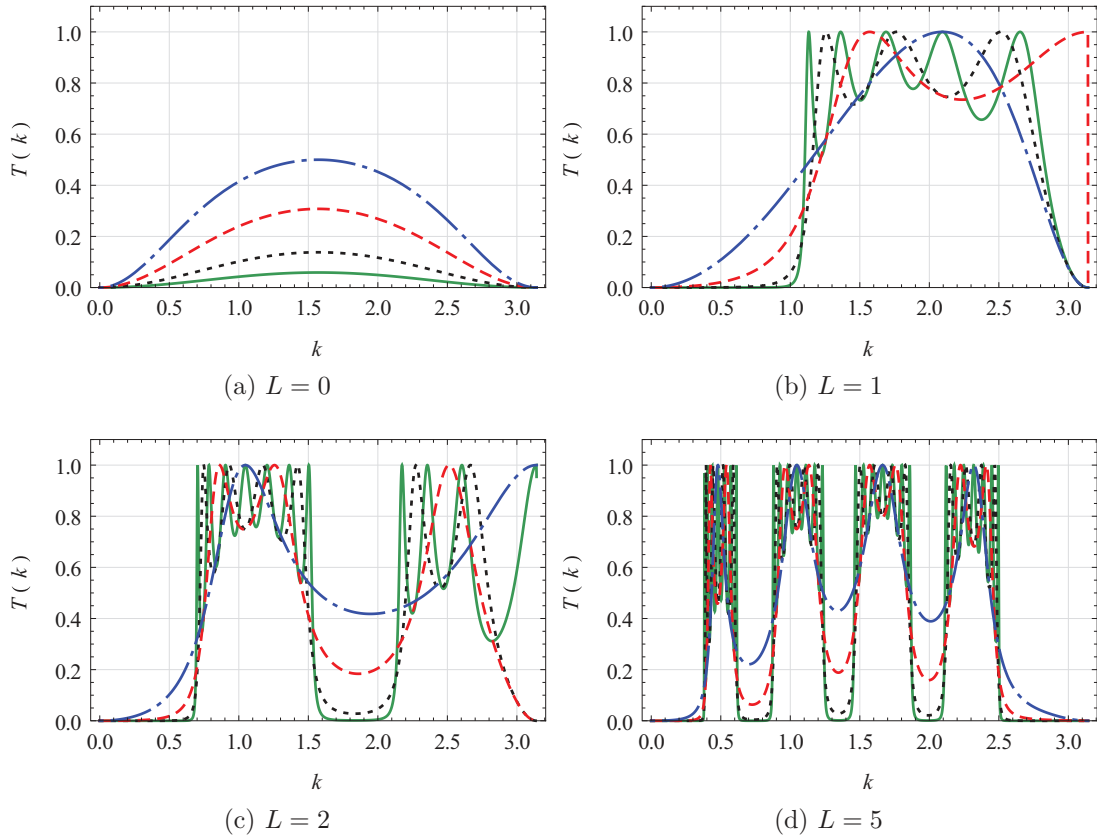


FIG. 4. (Color online) Transmission coefficient vs momentum for a CTQW momentum eigenstate incident on an N -point defect, where $N = 2$ (blue dot-dashed line), $N = 3$ (red dashed line), $N = 5$ (black dotted line), and $N = 8$ (green solid line). The defect-induced barriers are constructed with unity amplitude ($\alpha = 1$) and equal separation L .

Falloon and Wang [39], and Manouchehri and Wang [40] provides an efficient method of numerically calculating the transmission coefficient $T(k)$ over a wide range of k . In this section, this method will be adapted for the CTQW on the infinite line and verified against the multibarrier analytic solution derived in the previous section.

Consider an arbitrary initial state $|\psi_0\rangle$ incident on a set of defects or barriers, with the transmitted component denoted $|\psi_t\rangle$ and produced as per the transmission amplitude $t(k)$. Using the completeness of the momentum eigenstates, these can be written as

$$|\psi_0\rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} dk |k\rangle \langle k | \psi_0\rangle, \quad |\psi_t\rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} dk |k\rangle \langle k | \psi_t\rangle. \quad (25a)$$

Evolving the state $|\psi_0\rangle$ through time via the unitary operator \hat{U}_t , which acts to separate out the transmitted component,

$$\begin{aligned} |\psi_t\rangle &= \hat{U}_t |\psi_0\rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} dk \hat{U}_t |k\rangle \langle k | \psi_0\rangle \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} dk t(k) |k\rangle \langle k | \psi_0\rangle, \end{aligned} \quad (26)$$

and comparing this with Eq. (25a), it can be seen that we require

$$\langle k | \psi_t\rangle = t(k) \langle k | \psi_0\rangle \Rightarrow T(k) = |t(k)|^2 = \frac{|\langle k | \psi_t\rangle|^2}{|\langle k | \psi_0\rangle|^2}. \quad (27)$$

Using position space completeness ($\hat{I} = \sum_j |v_j\rangle \langle v_j|$) coupled with the inner product $\langle k | v_j\rangle = e^{-ikj}$, this expression can be evaluated explicitly in terms of probability amplitudes of the walker at each vertex:

$$T(k) = \frac{|\langle k | \psi_t\rangle|^2}{|\langle k | \psi_0\rangle|^2} = \frac{|\sum_j e^{-ikj} \langle j | \psi_t\rangle|^2}{|\sum_j e^{-ikj} \langle j | \psi_0\rangle|^2}. \quad (28)$$

The numerical calculation of the inner product $\langle j | \psi_t\rangle = \langle j | e^{-iHt} | \psi_0\rangle$ is performed by expanding the matrix exponential via the Chebyshev expansion scheme, as detailed in Wang and Scholz [41]. This method is particularly beneficial for two major reasons: first, since the coefficients of the expansion are Bessel functions, they vanish after a finite number of terms, allowing for an exceptionally high level of accuracy with a comparatively small number of terms. Second, this is an example of a global propagator, negating the need for iterative calculations at smaller time steps which can introduce accumulation error.

Briefly outlining the expansion, we have

$$\begin{aligned} U_t &\equiv e^{-iHt/\hbar} \\ &= e^{-i(\lambda_{\max} + \lambda_{\min})t/2} \left[J_0(\eta) \phi_0(-i\tilde{H}) + 2 \sum_{n=1}^{\infty} J_n(\eta) \phi_n(-i\tilde{H}) \right], \end{aligned} \quad (29)$$

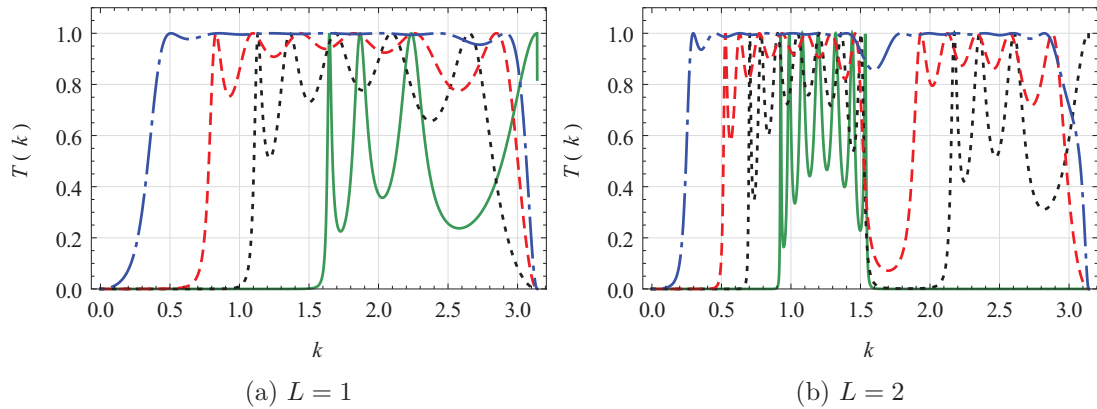


FIG. 5. (Color online) Transmission coefficient vs momentum for a CTQW momentum eigenstate incident on an evenly spaced 8-point defect, with barrier amplitudes $\alpha = 0.1$ (blue dot-dashed line), $\alpha = 0.5$ (red dashed line), $\alpha = 1$ (black dotted line), and $\alpha = 2$ (green solid line).

where λ_{\max} and λ_{\min} are maximum and minimum eigenvalues of the Hamiltonian matrix H , $\eta = (\lambda_{\max} - \lambda_{\min})t/2$, $J_n(\eta)$ are the Bessel functions of the first kind, and ϕ_n are the Chebyshev polynomials. To ensure convergence, the Hamiltonian needs to be normalized as

$$\tilde{\mathcal{H}} = \frac{1}{\lambda_{\max} - \lambda_{\min}} [2H - \lambda_{\max} - \lambda_{\min}]. \quad (30)$$

This method can now be used to investigate the behavior of a CTQW on an infinite line, incident on an arbitrary distribution of diagonal defects. However, for comparison purposes, we will restrict our attention to N equally spaced diagonal defects. The Hamiltonian under investigation is therefore given by

$$H = \sum_j (2|j\rangle - |j-1\rangle - |j+1\rangle) \langle j| + \sum_{n=0}^{N-1} \alpha_n |d+nL\rangle \langle d+nL|, \quad (31)$$

where $\alpha_n \in \mathbb{R}$ represents the amplitude of the n th defect, $d \in \mathbb{Z}$ is the position of the first defect, and $L \in \mathbb{N}$ is the integer spacing between defects. For convenience, the initial state is chosen such that the walker is localized at a vertex to the left of the double reflecting barriers, i.e., $|\psi(0)\rangle = |v_{j'}\rangle$, $j' < 0$.

Noting that

$$\begin{aligned} \left| \sum_j e^{-ikj} \langle j|j'\rangle \right|^2 &= \left| \sum_j e^{-ikj} \delta_{jj'} \right|^2 = |e^{-ikj'}|^2 \\ &= 1 \forall k \in [-\pi, \pi), \end{aligned} \quad (32)$$

this allows the expression for the transmission coefficient given by Eq. (28) to reduce to

$$T(k) = \left| \sum_j e^{-ikj} \langle j|\psi_t\rangle \right|^2 \quad \forall k \in [-\pi, \pi). \quad (33)$$

Using Eq. (29), $T(k) = |\sum_j e^{-ikj} \langle j|e^{-iHt}|j'\rangle|^2$ is calculated for $t = 300$, chosen sufficiently large such that the probability of the walker remaining located between the defects is small, i.e., $\sum_{j=0}^{(N-1)L} | \langle d+j|\psi(t)\rangle|^2 \approx 0$. The results are plotted in Fig. 6 for a variety of different values of L and compared to the analytic solution derived in Sec. IV. It is observed that the numeric results closely match the analytic solutions, with an average absolute difference of $\sigma_T = 2.64 \times 10^{-3}$. As an aside, note the presence of Gibbs phenomenon at the boundaries of the domain due to the use of Fourier methods. By restricting the analysis to $0.5 < k < 2.5$, the absolute difference between the numeric and analytic

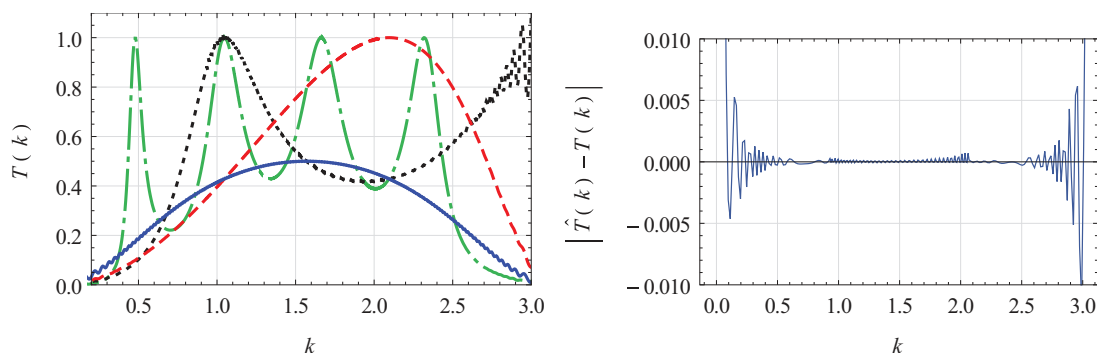


FIG. 6. (Color online) (a) Transmission coefficient vs momentum of a CTQW momentum eigenstate incident on two defects of unity amplitude, separated by distances $L = 0$ (solid blue line), $L = 1$ (dashed red line), $L = 2$ (dotted black line), and $L = 5$ (dot-dashed green line), and calculated via numerical Fourier methods. (b) Absolute difference between numeric and analytic results for the case $L = 1$.

results reduces to $\sigma_T = 3.58 \times 10^{-7}$. This general numerical approach can now be applied to situations where analytical analysis becomes impossible, fine tuning the parameters to increase accuracy.

It should be noted that a major source of error associated with this method is due to a nonzero probability distribution located between defects at measurement time. While this can be avoided by increasing the time step t , it requires that a larger number of vertices be included in the propagation grid, as well as an increase in the number of Chebyshev summation terms required for significant accuracy, potentially increasing the computational time. The additional computational time may prove insignificant with the accelerating availability of computational power. If not, further refinements (such as absorbing boundary conditions and multiple passes) may be needed.

VI. CONCLUSIONS

In the past two decades, quantum walks have played a pivotal role in the field of quantum information theory and current research is suggesting potential applications across a whole range of different fields, making them an invaluable tool in the study of structured, discrete-space systems. Significant advances are also constantly being made in experimental realizations of quantum walks, providing a means to fully utilize the computational power of quantum walkers, while simultaneously requiring an accurate theoretical and efficient numerical framework to provide detailed information on the effects of disorder and scattering on a quantum walk.

In this paper, a one-dimensional continuous-time quantum walk in the presence of multiple defects was explored analytically, with the results highlighting resonance behavior previously observed in the case of double diagonal defects. It was also demonstrated that by increasing the number of defects and by altering the defect amplitudes, “bands” of perfect transmission (surrounded by regions of zero transmission) can be selectively placed in momentum space, allowing for a high level of control of quantum walking characteristics. This provides a link between the tight-binding lattice models widely used in condensed matter physics and the development of quantum information applications based on continuous-time quantum walks. Finally, we extended the Fourier-Chebyshev method utilized in the literature for continuous position space to the discrete space of CTQWs, which provides a general numerical approach to situations where analytical analysis becomes impossible, such as multiple barriers, higher dimensions, and multiple interacting walkers.

Quantum walks remain an important field of study due to their crucial role in both quantum information processing and the modeling of complex quantum systems. As a result of this research, we hope to provide methods to selectively control and efficiently characterize the time evolution of quantum walks by taking advantage of diagonal disorder that can sometimes be unavoidable in physical systems.

ACKNOWLEDGMENT

J.A.I. would like to thank the Hackett Foundation and the University of Western Australia for financial support.

-
- [1] Y. Aharonov, L. Davidovich, and N. Zagury, *Phys. Rev. A* **48**, 1687 (1993).
 - [2] J. Kempe, *Contemp. Phys.* **44**, 307 (2003).
 - [3] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing STOC '03* (ACM, New York, 2003), pp. 59–68.
 - [4] A. M. Childs and J. Goldstone, *Phys. Rev. A* **70**, 022314 (2004).
 - [5] S. D. Berry and J. B. Wang, *Phys. Rev. A* **83**, 042317 (2011).
 - [6] S. D. Berry and J. B. Wang, *Phys. Rev. A* **82**, 042333 (2010).
 - [7] J. K. Gamble, M. Friesen, D. Zhou, R. Joynt, and S. N. Coppersmith, *Phys. Rev. A* **81**, 052313 (2010).
 - [8] B. L. Douglas and J. B. Wang, *J. Phys. A: Math. Theor.* **41**, 075303 (2008).
 - [9] A. M. Childs, *Phys. Rev. Lett.* **102**, 180501 (2009).
 - [10] E. Farhi, J. Goldstone, and S. Gutmann, [arXiv:quant-ph/0702144](https://arxiv.org/abs/quant-ph/0702144).
 - [11] A. M. Childs and D. Gosset, *J. Math. Phys.* **53**, 102207 (2012).
 - [12] F. W. Strauch, *Phys. Rev. A* **74**, 030301(R) (2006).
 - [13] Y. Bromberg, Y. Lahini, R. Morandotti, and Y. Silberberg, *Phys. Rev. Lett.* **102**, 253904 (2009).
 - [14] A. Peruzzo, M. Lobino, J. C. F. Matthews, N. Matsuda, A. Politi, K. Poulios, X. Zhou, Y. Lahini, N. Ismail, K. Wörhoff, Y. Bromberg, Y. Silberberg, M. G. Thompson, and J. L. O’Brien, *Science* **329**, 1500 (2010).
 - [15] L. Sansoni, F. Sciarrino, G. Vallone, P. Mataloni, A. Crespi, R. Ramponi, and R. Osellame, *Phys. Rev. Lett.* **108**, 010502 (2012).
 - [16] J. D. A. Meinecke, K. Poulios, A. Politi, J. C. F. Matthews, A. Peruzzo, N. Ismail, K. Wörhoff, J. L. O’Brien, and M. G. Thompson, *Phys. Rev. A* **88**, 012308 (2013).
 - [17] M. A. Broome, A. Fedrizzi, B. P. Lanyon, I. Kassal, A. Aspuru-Guzik, and A. G. White, *Phys. Rev. Lett.* **104**, 153602 (2010).
 - [18] M. Karski, L. Förster, J. Choi, A. Steffen, W. Alt, D. Meschede, and A. Widera, *Science* **325**, 174 (2009).
 - [19] H. Schmitz, R. Matjeschk, C. Schneider, J. Glueckert, M. Enderlein, T. Huber, and T. Schaetz, *Phys. Rev. Lett.* **103**, 090504 (2009).
 - [20] G. F. Koster and J. C. Slater, *Phys. Rev.* **95**, 1167 (1954).
 - [21] R. E. Merrifield, *J. Chem. Phys.* **38**, 920 (1963).
 - [22] P. Dean, *Proc. Phys. Soc.* **73**, 413 (1959).
 - [23] D. J. Thouless, *J. Phys. C* **5**, 77 (1972).
 - [24] I. Avgin and D. Huber, *J. Lumin.* **129**, 1916 (2009).
 - [25] Y. Yin, D. E. Katsanos, and S. N. Evangelou, *Phys. Rev. A* **77**, 022302 (2008).
 - [26] A. Schreiber, K. N. Cassemiro, V. Potoček, A. Gábris, I. Jex, and C. Silberhorn, *Phys. Rev. Lett.* **106**, 180403 (2011).
 - [27] O. Mülken and A. Blumen, *Phys. Rep.* **502**, 37 (2011).
 - [28] J. P. Keating, N. Linden, J. C. F. Matthews, and A. Winter, *Phys. Rev. A* **76**, 012315 (2007).
 - [29] E. Farhi and S. Gutmann, *Phys. Rev. A* **58**, 915 (1998).

- [30] E. Agliari, A. Blumen, and O. Mülken, *Phys. Rev. A* **82**, 012305 (2010).
- [31] G. S. Engel, T. R. Calhoun, E. L. Read, T. Ahn, T. Mančal, Y. Cheng, R. E. Blankenship, and G. R. Fleming, *Nature (London)* **446**, 782 (2007).
- [32] E. Collini, C. Y. Wong, K. E. Wilk, P. M. G. Curmi, P. Brumer, and G. D. Scholes, *Nature (London)* **463**, 644 (2010).
- [33] D. Mi, G. Liu, J. Wang, and Z. Li, *J. Theor. Biol.* **241**, 152 (2006).
- [34] D. Mi, W. Q. Meng, and Y. Q. Sun, *Phys. Rev. E* **83**, 041901 (2011).
- [35] O. Mülken, A. Blumen, T. Amthor, C. Giese, M. Reetz-Lamour, and M. Weidemüller, *Phys. Rev. Lett.* **99**, 090601 (2007).
- [36] A. M. Childs, D. Gosset, and Z. Webb, *Science* **339**, 791 (2013).
- [37] Z. J. Li, J. A. Izaac, and J. B. Wang, *Phys. Rev. A* **87**, 012314 (2013).
- [38] C. Yiu and J. Wang, *J. Appl. Phys.* **80**, 4208 (1996).
- [39] P. E. Falloon and J. B. Wang, *Comput. Phys. Commun.* **134**, 167 (2001).
- [40] K. Manouchehri and J. B. Wang, *J. Comput. Theor. Nanosci.* **3**, 249 (2006).
- [41] J. B. Wang and T. T. Scholz, *Phys. Rev. A* **57**, 3554 (1998).

pyCTQW: A continuous-time quantum walk simulator on distributed memory computers

Josh A. Izaac*, Jingbo B. Wang**

School of Physics, The University of Western Australia, Crawley WA 6009, Australia

Abstract

In the general field of quantum information and computation, quantum walks are playing an increasingly important role in constructing physical models and quantum algorithms. We have recently developed a distributed memory software package *pyCTQW*, with an object-oriented Python interface, that allows efficient simulation of large multi-particle CTQW (continuous-time quantum walk)-based systems. In this paper, we present an introduction to the Python and Fortran interfaces of *pyCTQW*, discuss various numerical methods of calculating the matrix exponential, and demonstrate the performance behaviour of *pyCTQW* on a distributed memory cluster. In particular, the Chebyshev and Krylov-subspace methods for calculating the quantum walk propagation are provided, as well as methods for visualisation and data analysis.

Keywords: Continuous-time quantum walk, Multiple walkers, Padé approximant, Krylov subspace method, Chebyshev matrix expansion

PROGRAM SUMMARY

Program Title: pyCTQW

Journal Reference:

Catalogue identifier:

Licensing provisions: none

Programming language: Fortran and Python

Computer: Workstation or cluster implementing MPI

Operating system: Any operating system with Fortran, python, and MPI installed.

RAM: Depends on graph size and number of walkers

Number of processors used: One or more processors may be used

Keywords: Continuous-time quantum walk, Multiple walkers, Padé approximant, Krylov subspace method, Chebyshev matrix expansion

Classification: 4.15, 14

External routines/libraries: PETSc [1–3], SLEPc [4–6], MPI, NumPy and SciPy [7–9], Matplotlib [10], NetworkX [11]

Nature of problem: Simulates, visualises and analyzes continuous-time quantum walks on arbitrary undirected graphs.

Solution method: Distributed memory implementations of the matrix exponential, via a choice of Krylov-subspace and Chebyshev expansion techniques, are used to simulate the continuous-time quantum walkers. Visualisation ability is provided via the supplied Python module and Matplotlib.

Restrictions: The size of the quantum walking system is limited by the amount of available memory. The current package implements up to 3 simultaneous walkers with interactions, but it can be readily extended.

Unusual features: In addition to utilising a parallelized Krylov subspace method and Chebyshev approximation scheme to maximise efficiency, *pyCTQW* also provides functions for visualisation of the quantum walk dynamics and calculation of multi-particle entanglement, and allows for arbitrary diagonal defects to be placed on graph nodes to explore transmission and resonance structures.

Running time: Runtime varies depending on the size of the graph, number of processors used, and number of simultaneous walkers.

References

- [1] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), *Modern Software Tools in Scientific Computing*, Birkhäuser Press, 1997, p. 163–202.
- [2] S. Balay, J. Brown, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, PETSc Web page, 2013. <http://www.mcs.anl.gov/petsc>.
- [3] S. Balay, J. Brown, Buschelman, Kris, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, PETSc Users Manual, Technical Report ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.
- [4] V. Hernandez, J. E. Roman, V. Vidal, SLEPc: scalable library for eigenvalue problem computations, *Lecture Notes in Computer Science* 2565 (2003) 377–391.
- [5] V. Hernandez, J. E. Roman, V. Vidal, SLEPc: a scalable and flexible toolkit for the solution of eigenvalue problems, *ACM Trans. Math. Software* 31 (2005) 351–362.
- [6] C. Campos, J. E. Roman, E. Romero, A. Tomas, SLEPc Users Manual, Technical Report DSIC-II/24/02 - Revision 3.3, D. Sistemes Informàtics i Computació, Universitat Politècnica de València, 2012.
- [7] E. Jones, T. Oliphant, P. Peterson, et al., *SciPy: Open source scientific tools for Python*, 2001.

*Corresponding author.

E-mail address: josh.izaac@uwa.edu.au

**E-mail address: jingbo.wang@uwa.edu.au

- [8] P. Peterson, F2PY: a tool for connecting Fortran and python programs, *International Journal of Computational Science and Engineering* 4 (2009) 296.
- [9] T. E. Oliphant, Python for scientific computing, *Computing in Science & Engineering* 9 (2007) 10–20.
- [10] J. D. Hunter, Matplotlib: A 2D graphics environment, *Computing in Science & Engineering* 9 (2007) 90–95.
- [11] A. A. Hagberg, D. A. Schult, P. J. Swart, Exploring network structure, dynamics, and function using NetworkX, in: ps-G. Varoquaux, T. Vaught, J. Millman (Eds.), *Proceedings of the 7th Python in Science Conference*, Pasadena, CA USA, pp. 11 – 15.

1. Introduction

First posited by Aharonov et al. [1] in 1993, a quantum walk is the quantum analogue of the classical random walk, which takes into account quantum properties such as superposition and quantum correlations and leads to markedly different properties. For example, quantum walks propagate quadratically faster than their classical counterparts, with propagation ballistic rather than diffusive [2]. They also exhibit wave-like nature around defects and impurities, such as interference, resonance trapping and transmission [3–5]. As a result of their drastically different propagation properties, there has been an abundance of quantum walk related research in the last decade, with quantum walk formulations motivating the creation of quantum algorithms that are faster and more efficient than their classical analogues [6, 7] – including so far network search and centrality [8, 9] and graph isomorphism analysis [10–13]. Furthermore, quantum walks on graphs have been proven to provide methods of universal computation [14, 15], allowing quantum walk-based systems to play a crucial role in the race to develop a quantum computer. As such, today quantum walks constitute an important tool in quantum information theory.

Similarly to classical random walks, there exist two similar but fundamentally different formulations of the quantum walk: the discrete-time quantum walk (DTQW), and the continuous-time quantum walk (CTQW). Unlike the classical regime, their relationship is highly non-trivial [16]; this work focuses on multi-particle continuous-time quantum walks. In recent years, outside the theoreticians’ office, physical implementations of quantum walkers have been demonstrated experimentally, with approaches including the use of waveguides and photonics [17, 18, 20] and ion lattices [21–23]), paving the way for the construction of devices capable of running these quantum algorithms. Additionally, the demand for increasingly accurate models of complex biochemical processes that border the quantum regime (and thus exhibit some form of quantum coherence) have led to the investigation of continuous-time quantum walkers as a viable modelling candidate [24]. So far, early research has explored their potential in analysing energy transport in biological systems [25–29], highlighting the growing versatility of quantum walks whilst providing new insights into the natural world.

However, with the growing number of quantum walk applications – especially those in the biochemical sciences (which can involve numerous interacting walkers on increasingly large and complex graphs) – the ability to efficiently simulate and

analyse quantum walk behaviour becomes essential. In this work, we introduce *pyCTQW*, a software package for efficiently simulating CTQWs, whilst taking advantage of the huge potential provided by modern High Performance Computing (HPC) platforms. Developed using Fortran, a Python module is also available, interfacing directly with the Fortran library via F2PY [30]. This allows *pyCTQW* to take advantage of the many scientific and data visualisation Python libraries available in the much more user-friendly Python environment, whilst avoiding the additional overhead that would be caused by implementing the core algorithms in an interpreted language. As such, *pyCTQW* is a valuable tool for the study, data visualisation and analysis of CTQW based systems.

This paper is structured as follows; in section 2, we introduce the fundamental theory underpinning continuous-time quantum walks, before discussing numerical algorithms used to solve such systems in section 3. The *pyCTQW* software package is then detailed in section 4, with particular attention given to the structure, usage, and installation, followed by specific examples presented in section 5, and computational performance discussed in section 6. Finally, our conclusions are provided in section 7.

2. Continuous-time quantum walks

Continuous-time quantum walks were first introduced by Farhi and Gutmann [31] in 1998, as a quantization of classical Markov processes, with the system now evolving as per the Schrödinger equation rather than the Markovian Master equation. As such, classical probabilities are replaced by quantum probability amplitudes. To illustrate, consider a continuous-time random walk on the arbitrary undirected graph $G(V, E)$, composed of unordered vertices $j \in V$ in addition to edges $e_i = (j, j') \in E$ connecting vertices $j, j' \in V$. The transition rate matrix H is defined as the Laplacian matrix of the graph; that is,

$$H_{jj'} = \begin{cases} -\gamma_{jj'} & j \neq j'; (j, j') \in G \\ 0, & j \neq j'; (j, j') \notin G \\ S_j, & j = j' \end{cases} \quad (1)$$

where $\gamma_{jj'}$ is the probability per unit time of a transition from node j to node j' , and the elements S_j satisfy the condition

$$S_j = \sum_{\substack{j'=1 \\ j' \neq j}}^N \gamma_{jj'} \quad (2)$$

to ensure unitary behaviour. Classically, the state of the random walker is described by a probability distribution $\mathbf{P}(t)$, and its time evolution is governed by the Master equation

$$\frac{d\mathbf{P}(t)}{dt} = H\mathbf{P}(t). \quad (3)$$

which has the formal solution $\mathbf{P}(t) = e^{-Ht}\mathbf{P}(0)$.

Extending the above formulation to the quantum realm involves replacing the real valued probability vector $\mathbf{P}(t)$ with a

complex valued wavefunction $|\psi(t)\rangle$, and multiplying the exponent by the complex factor i ; i.e.

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle. \quad (4)$$

It can be readily seen that this is now of the same form as the Schrödinger equation. Analogously to other continuous-space quantum systems, the transition rate matrix is commonly referred to as the system Hamiltonian, and now must satisfy the additional requirement of Hermiticity to ensure unitary time-evolution, which guarantees that the norm of $|\psi(t)\rangle$ is conserved for all t . The complex-valued state vector $|\psi(t)\rangle = \sum_j a_j(t)|j\rangle$, where $a_j(t) = \langle j|\psi(t)\rangle \in \mathbb{C}$, represents the probability amplitude of the walker being found at node $|j\rangle$ at time t , with $|a_j(t)|^2 = |\langle j|\psi(t)\rangle|^2$ the resulting probability.

In the simulation of continuous-time quantum walk based systems, often the most computationally intensive process is the propagation algorithm itself; that is, the calculation of the evolved state at time t from an initial state $|\psi(0)\rangle$

$$|\psi(t)\rangle = \langle j|e^{-iHt}|\psi(0)\rangle, \quad (5)$$

due to the need to efficiently and accurately calculate a matrix exponential-vector product. In the case of single walkers or small graphs, fast and efficient matrix exponential calculation is readily available using modern numerical packages and toolkits such as *Mathematica*, *Matlab*, and *SciPy*.

Alternatively, consider a CTQW composed of n simultaneous interacting walkers on an arbitrary graph G of N vertices. In order to construct the system Hamiltonian, we take the *Kronecker sum* over all n single walker subspaces, and add on the interaction term:

$$\begin{aligned} H &= \bigoplus_{i=1}^n H^{(i)} + \Gamma_{int} \\ &= \left(H^{(1)} \otimes I_N^{(2)} \otimes \dots \otimes I_N^{(n)} \right) \\ &\quad + \dots + \left(I_N^{(1)} \otimes \dots \otimes I_N^{(n-1)} \otimes H^{(n)} \right) + \Gamma_{int}, \end{aligned} \quad (6)$$

where $H^{(i)}$ is the single walker $N \times N$ Hamiltonian of the i th particle on graph G , $I_N^{(i)}$ represents the $N \times N$ identity matrix acting on particle i , and Γ_{int} is the interaction term acting on an arbitrary number of particles. In the development of *pyCTQW*, we were mainly concerned with on-site interactions between all walkers,

$$\Gamma_{int}(\alpha) = \frac{1}{2}\alpha \sum_{\substack{i,i'=1 \\ i \neq i'}}^n \sum_{j=1}^N (|j\rangle\langle j|)^{(i)} \otimes (|j\rangle\langle j|)^{(i')}, \quad (7)$$

where $\alpha \in \mathbb{R}$ is the interaction constant. In the case of no interactions (i.e. $\alpha = 0$), the above definition of the multi-particle Hamiltonian (Eq. 6) results in a separable time evolution operator,

$$\begin{aligned} \hat{U}(t) &= e^{-iHt} = \bigotimes_{i=1}^n \exp(-iH^{(i)}t) \\ &= \hat{U}(t)^{(1)} \otimes \hat{U}(t)^{(2)} \otimes \dots \otimes \hat{U}(t)^{(n)}, \end{aligned} \quad (8)$$

as expected; however this is not possible when interactions are present. In this work, the Hamiltonian matrix is taken as the same for all particles, and is simply given by the graph Laplacian,

$$H^{(i)} \equiv H_{1p} \equiv \Delta G \quad \forall i \in \{1, \dots, n\}, \quad (9)$$

where the superscript simply defines the *subspace* the Hamiltonian acts on.

The important thing to note from the above multi-particle Hamiltonian is that for n simultaneous walkers on a graph containing N nodes, the system Hamiltonian will have dimensions $N^n \times N^n$ – significantly larger than the Hamiltonian for a single walker. Thus for systems requiring multiple walkers on large graphs, calculating the propagation using the tools mentioned above can potentially take an excessive amount of time, warranting an investigation into alternate algorithms and HPC solutions.

3. Matrix exponential methods

Although published almost 35 years ago, ‘*Nineteen Dubious Ways to Compute the Exponential of a Matrix*’ by Moler and Van Loan [32] remains a heavily influential review of the various methods of approximating the matrix exponential, even necessitating an updated revision in 2003 [33]. Of the various methods described by Molar and Van Loan, today the two most pervasive include the squaring and scaling method (primarily for dense matrices) and Krylov subspace methods (for large sparse matrices) – both commonly used in conjunction with the Padé approximation, a high order series approximation.

An alternative algorithm that has recently found traction in the field of computational physics and quantum chemistry is the so-called Chebyshev series approximation, which takes its name from the Chebyshev polynomials that occur in the series expansion [34–39]. A huge part of what makes the Chebyshev expansion so attractive is the use of Bessel J zero functions as series coefficients, leading to exceptionally fast convergence without sacrificing a high level of accuracy. In fact, recent research has shown the Chebyshev approximation is an efficient alternative to both the scaling and squaring method [40] and Krylov subspace methods [41], which holds significant promise in parallel computation.

The matrix exponential approximations mentioned here will be expanded on below, with the background theory, advantages and implementations summarised (for a more detailed summary, refer to Molar and Van Loan [33]).

3.1. Padé approximation

The Padé approximant uses a rational function of power series in order to approximate the matrix exponential; this is commonly defined as

$$e^A \approx R_{pq}(A) = [D_{pq}(A)]^{-1} N_{pq}(A), \quad (10)$$

where

$$N_{pq}(A) = \sum_{j=0}^p \frac{(p+q-j)! p!}{(p+q)!(p-j)!} \left(\frac{A^j}{j!} \right) \quad (11)$$

and

$$D_{pq}(A) = \sum_{j=0}^q \frac{(p+q-j)!q!}{(p+q)!(q-j)!} \left(\frac{(-A)^j}{j!} \right). \quad (12)$$

Note that for $q = 0$, $p > 0$, this reduces to the Taylor series expansion of e^A ; the Padé approximant simply matches the Taylor series to order $\mathcal{O}(A^{p+q})$. Rather than implementing the above definition directly, however, in practice we typically choose the diagonal Padé approximants $p = q$, as these result in similar accuracy for significantly reduced computational cost. Eq. (10) then reduces to

$$e^A \approx R_{pp}(A) = [N_{pp}(-A)]^{-1} N_{pp}(A), \quad (13)$$

with error given by

$$e^A - R_{pp}(A) = \frac{(-1)^p (p!)^2}{(2p)!(2p+1)!} A^{2p+1} + \mathcal{O}(A^{2p+2}). \quad (14)$$

The main advantage of the Padé approximation over other series expansions is that for matrices where $\|A\|_2 \lesssim 1$, matrix exponentials can be computed to the same level of accuracy with up to half the computation. However, there are important downsides – firstly, a reasonable level of accuracy can only be achieved for $\|A\|_2 \lesssim 1$; issues also arise for large p , increasing the risk that the denominator is ill-conditioned for inversion (although this is highly dependent on the eigenvalue spread of A). Furthermore, the evaluation of the power series $N_{pp}(A)$ requires dense matrix arithmetic, foregoing any speed-ups/memory reductions that may have arisen by manipulating sparse matrices.

3.2. Scaling and squaring method

In order to apply the Padé approximation to matrices of all norms, an iterative process known as the scaling and squaring method is used, which exploits the property $e^A = (e^{A/s})^s$ to reduce the matrix norm of A before finding the Padé approximant. Thus, the adjusted Padé approximant can be written as

$$e^A \approx e^{A+E} = R_{pp}(A/2^s)^{2^s}, \quad (15)$$

where $s \in \mathbb{N}$, and E is a matrix signifying the deviation from the exact value when calculating the matrix exponential of e^A using the Padé approximation. The following inequality is satisfied:

$$\frac{\|E\|}{\|A\|} \leq \frac{2^{3-2p}(p!)^2}{(2p)!(2p+1)!}, \quad (16)$$

where the parameters p and s are chosen such that $\|E\|/\|A\| \leq \epsilon$ (here ϵ is a specified parameter determining the error tolerance), and $p + s$ is *minimised* (as the floating point operations per second for an $n \times n$ matrix is of the order $(p + s + 1/3)n^3$). Very generally, $p = 6$ or 7 is optimal for single precision, whereas $p = 17$ is optimal for double precision. For example, *MATLAB*'s `expm` function uses $p = 6$ and $\max\{s : \|A/2^s\|_\infty \leq 0.5\}$.

This method remains one of the most effective methods of numerically calculating the matrix exponential of dense matrices [42, 43], and as such is widely used in modern numerical packages (such as *MATLAB*'s `expm`, *Mathematica*'s `MatrixExp`, *SciPy*'s `linalg.exp`, and *Expokit*'s Fortran and Matlab libraries).

3.3. Krylov subspace methods

Over the last 35 years, an alternative method has arisen which enables us to harness the computational advantages afforded by large sparse matrices. This is the eponymous Krylov subspace method, which provides an effective scheme for efficiently calculating the matrix-vector product $e^{At}\mathbf{v}$ – avoiding the need to calculate and store a large intermediate matrix. Here A is a $n \times n$ matrix, t is a scaling parameter, and $\mathbf{v} \in \mathbb{R}^n$ is an n -element vector. Whilst not providing the matrix exponential directly, this matrix-vector product arises often in the study of quantum systems, and as such is frequently the desired computational end-product – this is generally the case in CTQWs, where transition amplitudes $\langle j|\hat{U}(t)|\psi(0)\rangle = \langle j|e^{-iHt}|\psi(0)\rangle$ are desired.

This scheme takes its name from the process of approximating an n -dimensional problem (in this case $e^{At}\mathbf{v}$) onto a smaller m -dimensional *Krylov subspace*,

$$\mathcal{K}_m(At, \mathbf{v}) = \text{span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \dots, A^{m-1}\mathbf{v}\}, \quad \forall t \in \mathbb{C}, \quad (17)$$

whereby dense matrix methods become adequately efficient to solve the reduced problem. For example, in the case of $e^{At}\mathbf{v}$, we first use the Arnoldi or Lanczos methods (successful Krylov-based iterative methods in their own right, used for calculating the eigenvalues of large sparse general/symmetric matrices respectively) to calculate the orthonormal basis set $V_m = \{\mathbf{v}_1, \dots, \mathbf{v}_m\} \in \mathbb{R}^{n \times m}$ from $\mathcal{K}_m(A, \mathbf{v})$. A byproduct of this calculation is the upper Hessenberg matrix $H_m \in \mathbb{R}^{m \times m}$, which is the projection of $A_n \in \mathbb{R}^{n \times n}$ onto V_m (i.e. $V_m^T A_n V_m = H_m$). Note that H_m approximates the original sparse matrix A_n ,

$$AV_m = V_m H_m + h_{m+1,m} \mathbf{v}_{m+1} \hat{\mathbf{e}}_m^T \approx V_m H_m, \quad (18)$$

allowing the large sparse matrix exponentiation-vector product problem to be approximated via

$$e^{At}\mathbf{v} \approx \beta V_m e^{tH_m} \hat{\mathbf{e}}_1, \quad (19)$$

since $\mathbf{v} = \beta \mathbf{v}_1$ for some constant β . This can then be tackled via dense matrix methods.

Krylov subspace methods are often combined with the squaring and scaling method when calculating the matrix exponential of a large sparse matrix, due to the latter's efficiency in computing the resulting dense matrix exponential. Moreover, the ability to achieve a high level of accuracy with a relatively small value of m , typically around 40, combined with the fact that successive time-steps can be computed by utilising previously constructed Krylov subspace with little-to-no extra cost, have resulted in the Krylov method becoming a leading approach for large sparse matrices. Today, Krylov matrix exponential methods are available as part of *Mathematica* (using the `MatrixExp[A, v]` function), *Expokit* [44] (a collection of Fortran and Matlab libraries designed for working with Markovian processes), and *SLEPc/slepc4py* (a HPC implementation with C, Fortran and Python bindings), amongst others.

3.4. Chebyshev approximation

Not as common as the Padé and Krylov methods, but rapidly gaining traction in fields such as high performance computing

and quantum mechanics, is the Chebyshev series expansion to the matrix exponential;

$$e^{tA} = e^{(\lambda_{max} + \lambda_{min})t/2} \left[J_0(\alpha)\phi_0(\tilde{A}) + 2 \sum_{n=1}^{\infty} i^n J_n(\alpha)\phi_n(\tilde{A}) \right], \quad (20)$$

where $\lambda_{max}, \lambda_{min} \in \mathbb{C}$ are the eigenvalues of an $n \times n$ matrix A with largest and smallest real parts respectively, $\alpha = i(\lambda_{min} - \lambda_{max})t/2$, and $\phi_n(\tilde{A})$ are the Chebyshev polynomials, satisfying the recurrence relations

$$\phi_0(\tilde{A}) = I, \quad (21a)$$

$$\phi_1(\tilde{A}) = \tilde{A}, \quad (21b)$$

$$\phi_n(\tilde{A}) = 2\tilde{A}\phi_{n-1}(\tilde{A}) - \phi_{n-2}(\tilde{A}). \quad (21c)$$

Note that for maximal convergence, we require normalisation such that $\lambda \in [-1, 1]$; thus

$$\tilde{A} = \frac{2A - (\lambda_{max} + \lambda_{min})I}{\lambda_{max} - \lambda_{min}}. \quad (22)$$

As with the Krylov method, we can calculate the matrix-vector product $e^{At}\mathbf{v}$ directly using the Chebyshev approximation, by post-multiplying the Chebyshev series expansion with \mathbf{v} . This allows us to take full advantage of sparse matrix libraries when calculating the *sparse matrix* \times *dense vector* terms in the summation, whilst avoiding the extra memory usage required to store intermediate matrix terms. Another attractive property of the Chebyshev expansion is the use of Bessel function zeros as series coefficients, as $J_n(\alpha) \approx 0$ for $n > |\alpha|$, allowing for fast convergence and significantly high accuracy after only $|\alpha| \propto t$ terms (see Fig. 1). Whilst various definitions exist for the relative error of the Chebyshev expansion [41], for simplicity it is sufficient to instruct the series to be truncated when the condition

$$|2J_n(\alpha)| \leq \epsilon, \quad (23)$$

is satisfied for a specified tolerance $\epsilon > 0$, as subsequent terms can be considered negligible. In *pyCTQW*, we nominally choose $\epsilon = 10^{-18}$.

As with the Padé approximation, the Chebyshev approximation is only convergent for matrices where $\|A\| < 1$, thus necessitating the scaling procedure $A \rightarrow \tilde{A}$ detailed above. In practice only λ_{max} need be calculated, since the Laplacian for a undirected, defect-free graph is positive-semidefinite, ensuring $\lambda_{min} = 0$, whilst the calculation of λ_{max} is often computationally much less demanding.

However, when working with large finite graphs in the presence of disorder, or infinite graphs, the density of the eigenvalue spectrum around $\lambda = 0$ can lead to time-consuming eigenvalue calculations, resulting in a significant bottleneck when simulating the quantum walk propagation. This issue can be avoided in principle by implementing the scaling and squaring method, as described in subsection 3.2, as it does not require the calculation of eigenvalues:

$$e^{tA} = \left[J_0(\alpha)\phi_0\left(\frac{tA}{2^s}\right) + 2 \sum_{n=1}^{\infty} i^n J_n(\alpha)\phi_n\left(\frac{tA}{2^s}\right) \right]^s. \quad (24)$$

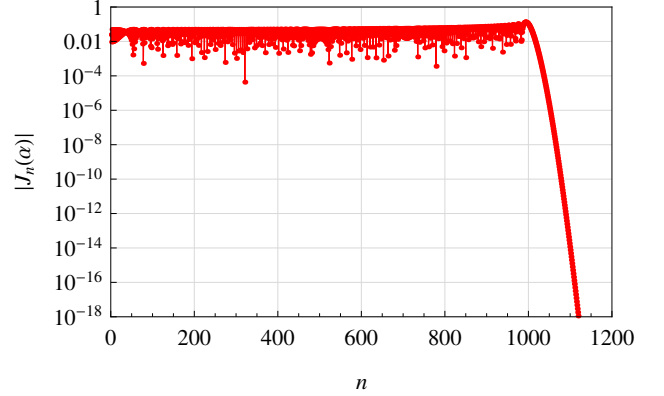


Fig. 1: Absolute value of the Chebyshev coefficients $|J_n(\alpha)|$ for $|\alpha| = 1000$, when calculating e^{At} for a 3×3 matrix A . Note that $|\alpha|$ is proportional to the numerical range of matrix A and time-step t .

Unfortunately, this approach requires the use of dense matrix operators and functions, and also loses the ability to compute the matrix-vector product $e^{At}\mathbf{v}$ directly from a matrix-vector product series, removing many of the properties that make the Chebyshev series expansion so appealing.

As such, we have elected to implement the eigenvalue scaling method in *pyCTQW* as one of the two possible propagator methods (alongside SLEPC's built-in Krylov methods). In order to alleviate the computational expense of calculating λ_{min} or λ_{max} for infinite graphs or graphs containing defects, the user may choose to enter a value for $\hat{\lambda}_{min}$ or $\hat{\lambda}_{max}$ directly – bypassing the eigenvalue calculation – or fallback to the Krylov method. If entering eigenvalue estimates, these should be such that $0 \leq \hat{\lambda}_{min} \leq \lambda_{min} < \lambda_{max} \leq \hat{\lambda}_{max}$, ensuring that the numerical range of \tilde{A} remains within $[-1, 1]$. Note that estimation of the extreme eigenvalues outside the actual numerical range produces the same level of accuracy at the expense of a larger value of $|\alpha|$, which would then require some extra matrix-vector terms in the series expansion to ensure accuracy. Nonetheless, the Chebyshev expansion has been shown to exhibit very low sensitivity to eigenvalue estimates [41], allowing this ‘semi-empirical’ implementation to continue to be an effective alternative to the Krylov method in specific circumstances.

Although the Chebyshev matrix exponentiation method is starting to show great promise, especially in fields of quantum chemistry and physics, the more established Krylov techniques continue to dominate in computational applications – today, *Expokit*[44] is the only well known package providing the Chebyshev methods (albeit, only for dense matrices). By implementing distributed memory Chebyshev methods *in addition* to Krylov methods, *pyCTQW* will be able to provide fast and efficient CTQW propagation methods to work with multiple quantum walkers.

4. Software package

The *pyCTQW* package provides both a Python-based object-orientated framework (*pyCTQW.MPI*) as well as a library (*libctqwMPI*) containing various Fortran subroutines, which

enable the user to simulate and explore various user-defined CTQW systems in a HPC environment. In the latter case, this is performed via the manipulation of graph-constrained quantum walk objects, whereas in the former case a more sequential paradigm is required where the user calls various subroutines on matrix or vector objects.

In this section, installation and dependencies will be addressed briefly, after which usage of both the Python and Fortran components of *pyCTQW* will be discussed in detail.

4.1. Dependencies and installation

As the ability to readily take advantage of modern HPC systems is one of the major design goals of this package, the decision was made to utilize the MPI-based PETSc [45–47] and SLEPc [48, 49] scalable linear algebra libraries. This allowed rapid parallelisation whilst also taking advantage of the optimised data-structures and matrix operations provided by PETSc and SLEPc.

Furthermore, in addition to requiring PETSc and SLEPc for parallel computation, the *pyCTQW* Python module also makes use of a wide variety of scientific and data visualisation libraries, including NumPy, SciPy, Matplotlib and NetworkX. This provides a rich user environment, and builds on the maturity and efficiency of the aforementioned libraries. It should be noted that whilst the Python module requires the *petsc4py* Python extension in order for the user to directly manipulate PETSc data structures and access MPI attributes (such as *rank*), the quantum walking calculations are called directly from the Fortran library, allowing the module to exploit the speed gain afforded by Fortran.

Once these dependencies are met, the Python module can be installed from the source code by running the following terminal command:

```
python setup.py install
```

whilst the Fortran library can be compiled via the included makefile:

```
make fortran
```

For additional installation help and compilations options, for example building *pyCTQW* as a shared library, the user should refer to the online *pyCTQW* documentation [50].

4.2. Fortran library *libctqwMPI*

Of the two interfaces available as part of the software package, the Python module is designed for an easier end-user experience, and thus is much more full featured. However, the underlying Fortran library may still be used, and is described briefly below.

To call functions and subroutines from the included *libctqwMPI* Fortran library, the main Fortran program should have the following basic structure:

```
program main
! load libctqwMPI module
use ctqwMPI

! PETSc headers
```

```
#include <finclude/petsc.h>

PetscErrorCode :: ierr
PetscMPIInt    :: rank

! initialize SLEPc and PETSc
call PetscInitialize(PETSC_NULL_CHARACTER,ierr)
call MPI_Comm_rank(PETSC_COMM_WORLD,rank,ierr)

! program code here

! finalise PETSc
call PetscFinalize(ierr)
end program main
```

Note that this structure is influenced by typical PETSc and SLEPc Fortran programs, due to the dependence on these aforementioned libraries. As such, we can take advantage of PETSc system routines, e.g. allowing command line arguments to be easily added, as well as code profiling capabilities (i.e. using PETSc *LogStages* to create a computational summary of parallel code execution). For more details, refer to the PETSc and SLEPc documentations [2, 6].

Once the *libctqwMPI* module has been included and PETSc and SLEPc have been properly initialised, CTQW subroutines can then be called directly. For example, to calculate the propagation of Hamiltonian *H* from state *psi0* to *psi* after time *t*,

```
call qw_cheby(psi0,psi,t,H,Emin,Emax,rank,n)
```

where *psi0*, *psi* and *H* are properly declared/allocated PETSc vectors/matrices, *t*, *Emin* and *Emax* are PETSc reals, and *rank*, *n* are PETSc integers. For a summary of available parallel CTQW subroutines, see Table 1.

Once the Fortran program is complete, it can be compiled and linked against *libctqwMPI* using the supplied makefile template, and then executed by running

```
mpirun -np X <program> [options]
```

where *X* is the number of MPI nodes to use, and *options* refer to either user defined or inbuilt PETSc/SLEPc command line arguments used to modify program function. For further information and additional details regarding the Fortran interface, the user is advised to refer to the online documentation [50].

4.3. Python module *pyCTQW.MPI*

Once *pyCTQW.MPI* is installed, calculations involving CTQWs can be called in several ways; either via an executable Python script, or as an interactive session, for example, using the *iPython* or *iPython-Notebook* environments. The syntax in both cases is identical, with slight differences in execution. Here, we will be concerned mainly with executable Python scripts, required by most HPC job schedulers.

4.3.1. Initialisation

In order to set-up the Python environment correctly, we must first initialize the PETSc environment, and import the *pyCTQW.MPI* module:

```
#!/usr/bin/env python2.7
import sys, petsc4py
```

Table 1: Summary of parallel CTQW subroutines available in the Fortran `libctqwMPI` library. For more details and required arguments, refer to the online documentation.

<i>Hamiltonian subroutines</i>	
<code>importadjtoh</code>	Import an adjacency matrix from a file, and create a PETSc Hamiltonian matrix
<code>adjtoh</code>	Convert an adjacency array to a PETSc Hamiltonian matrix
<code>hamiltonian_p<N>_line</code>	Create the Hamiltonian matrix representing $N = 1, 2$ or 3 walkers on an infinite line
<i>Statespace subroutines</i>	
<code>p<N>_init</code>	Initialise the statespace of an $N = 1, 2$ or 3 particle CTQW on specified graph nodes
<code>marginal<N></code>	Calculate the marginal probability of particle number $p \in N$ for an $N = \{1, 2, 3\}$ particle CTQW
<i>Quantum walk propagation</i>	
<code>min_max_eigs</code>	Calculate the minimum and maximum eigenvalues of a PETSc matrix using SLEPc
<code>qw_krylov</code>	Propagate the quantum walk for time t using Krylov subspace methods
<code>qw_cheby</code>	Propagate the quantum walk for time t using the Chebyshev series expansion
<code>entanglement</code>	Calculates the 2 particle Von Neumann entropy $S = -\sum_i \lambda_i \log_2 \lambda_i$, where λ_i are the eigenvalues of the reduced density matrix $\rho_2 = \text{Tr}_1(\psi(t)\rangle\langle\psi(t))$

```
petsc4py.init(sys.argv)
from petsc4py import PETSc
import pyCTQW.MPI
```

4.3.2. User defined options

PETSc can be used to create command line arguments for the script; this is useful in HPC environments where multiple jobs are to be submitted for a parameter exploration. For example, the following code creates two command line options, `-t` and `-N`, with default values of 100 and 20 respectively:

```
OptDB = PETSc.Options()
N = OptDB.getInt('N', 100)
t = OptDB.getReal('t', 20)
```

When running the complete parallelised simulation, the options created are invoked as follows,

```
mpirun -np X <script>.py [options]
```

where X is the number of MPI compute nodes/processes. In the case of the latter example,

```
mpirun -np X <script>.py -N 5 -t='0.3'
```

would alter the two user defined variables as indicated. Furthermore, most PETSc and SLEPc subroutines accept command line options which modify their settings; for instance, when using the SLEPc EPS eigensolver, the eigensolver type to be used can be changed dynamically by passing an `eps_type` option:

```
mpirun -np 2 <program> -eps_type='lapack'
```

For more details on built-in PETSc/SLEPc command line options, refer to the PETSc and SLEPc documentation [2, 6].

4.3.3. Rank and local operations

When running on multiple nodes, sometimes only specific nodes are required to perform a specific calculation or operation, for instance, the I/O operations where all nodes already have the same information. Using PETSc, the rank (i.e. the MPI process number) can be determined for each process, and conditional statements are used to control which node performs the I/O operation:

```
rank = PETSc.Comm.Get_rank(PETSc.COMM_WORLD)
if rank == 0:
    print '1P Line\n'
```

Caution is advised, however – the above should only be utilised for user defined functions and processes, as all of the methods and functions available in `pyCTQW.MPI` are designed to work globally on all processes, and should not be created or called on a subset of all available nodes; doing so may result in unresponsive simulations. Moreover, most objects in `pyCTQW.MPI` contain I/O methods (e.g. `pyCTQW.MPI.Graph.exportState()`), alternatively PETSc I/O methods are also available (e.g. `PETSc.Vec.view()`); these are global over all nodes as mentioned above and should be used over custom I/O methods when possible.

4.3.4. Code profiling

PETSc also allows for easy code profiling by supplying the command line option `-log_summary` when executing your script. This functionality is built-in to `pyCTQW.MPI` with log stages created automatically for available methods, for example, when creating the Hamiltonian or initial statespace, finding the eigenvalues, or CTQW propagation, etc. This provides a wealth of important computational information, such as memory usage, communication, and timing data, allowing the user to fine-tune simulation parameters and solvers for ultimate efficiency.

Custom log stages may also be included using the following template:

```
stage1 = PETSc.Log.Stage('First Stage Title')
stage1.push()
# place stage 1 functions/operations here
stage1.pop()
```

4.4. pyCTQW framework

The `pyCTQW.MPI` Python module provides an object-oriented framework for simulating CTQW systems, which involves the creation of CTQW objects storing various data structures (such as Hamiltonians), which can be manipulated via built-in methods and functions (see Table 2).

The number of available methods and attributes is too large to detail in full here. Instead, a two particle CTQW on a Cayley tree will be discussed in the next section to provide a feel of the framework and its structure. For further details, refer to the `pyCTQW` online documentation [50].

Table 2: Summary of parallel CTQW classes available in `pyCTQW.MPI`. For more details and required arguments, please refer to the online documentation.

<i>CTQW submodule</i>	
<code>ctqw.Hamiltonian</code>	Contains methods for initializing, creating and manipulating Hamiltonian matrices
<code>ctqw.EigSolver</code>	Contains methods for setting up and solving the for the boundary eigenvalues of a distributed PETSc matrix
<i>Quantum walks on arbitrary graphs</i>	
<code>Graph</code>	1 particle CTQW on a user-defined graph
<code>Graph2P</code>	2 particle CTQW on a user-defined graph
<code>Graph3P</code>	3 particle CTQW on a user-defined graph
<i>Quantum walks on infinite lines</i>	
<code>Line</code>	1 particle CTQW on an infinite line
<code>Line2P</code>	2 particle CTQW on an infinite line
<code>Line3P</code>	3 particle CTQW on an infinite line

5. Worked example

In this example, a two particle continuous-time quantum walk will be performed on a 3-Cayley tree using `pyCTQW.MPI`. After initialising PETSc and `petsc4py`, the `pyCTQW` module is imported and a `Graph2P` object created, indicating that we will be working on a graph containing $n = 10$ nodes:

```
walk = pyCTQW.MPI.Graph2P(10)
```

Next, we import a 10×10 adjacency matrix of the 3-Cayley tree from a text file¹ with elements delimited by either tabs or spaces²:

```
d = [3,4]
amp = [2.0,1.5]
walk.createH('graphs/cayley/3-cayley.txt','txt',
            d=d,amp=amp,layout='spring',interaction=0.5)
```

In this example, defects are introduced at vertices 3 and 4 of the graph, with amplitudes 2 and 1.5 respectively, and the particles interact with strength 0.5 when co-located at the same vertex; this results in a system Hamiltonian of the form

$$H = H^{(1)} \oplus H^{(1)} + \Gamma_{int}(0.5) \quad (25)$$

where

$$H^{(1)} = L_{3C} + 2|3\rangle\langle 3| + 1.5|4\rangle\langle 4|$$

$$= \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 3 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 5 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 2.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

¹Importing the adjacency matrix from a 2D PETSc binary array, specified by filetype 'bin', is also supported.

²Alternative delimiters can also be passed using the `delimiter` keyword.

is the 1-particle Hamiltonian and L_{3C} is the Laplacian matrix of the 3-Cayley tree. Note that the graph layout is being stored in 'spring' form, which is the default if not specified with alternative layout options including circle, spectral and random. After executing the above code, a Hamiltonian object `walk.H` is produced, allowing the Hamiltonian matrix and its eigenvalues to be accessed.

Now that the system Hamiltonian has been calculated, the initial statespace can be populated. For example, the initial state is chosen to be $|\psi(0)\rangle = \frac{1}{\sqrt{2}}(|0,1\rangle + i|1,1\rangle)$, which is implemented as follows:

```
init_state = [[0,1,1.0/np.sqrt(2.0)],
              [1,1,1.0j/np.sqrt(2.0)]]
walk.createInitState(init_state)
```

Note that the initial state is passed to the CTQW object in the form of an $n \times 3$ array with format $[[x1,y1,amp1],[x2,y2,amp2],\dots]$, which generalizes an $n \times p$ array for a p -particle quantum walk. In more complex cases, an initial statespace can be imported from an $n \times n$ 2D array in text format, or an n^2 element PETSc binary vector, via `PETSc.Graph2P.importInitState()`.

The next process involves setting the eigensolver properties, which are then passed to the `EigSolver` object contained in our `walk` object:

```
walk.EigSolver.setEigSolver(tol=1.e-2,
                           verbose=False,emin_estimate=0.)
```

It is important to note that, since the minimum eigenvalue of a finite graph is necessarily 0, calculation of the minimum eigenvalue is avoided automatically when `pyCTQW` detects a finite graph. To reduce computational time further, `emax_estimate` may also be provided, however this *must* be larger than the *actual* maximum eigenvalue for Chebyshev convergence to be ensured (with convergence time then increasing as $|\text{emax_estimate} - \text{emin_estimate}|$ increases). For additional keywords that can be passed to the eigensolver, see Table 3.

Everything is now set up, allowing propagation to begin. The following commands will allow a record of the calculated probability at specified nodes and/or the global entanglement^{3,4} to be stored over time:

```
# create a probability handle for nodes 0-4,9:
walk.watch([0,1,2,3,4,9])
# create a handle to watch the entanglement
walk.watch(None,watchtype='entanglement',
           verbose=False,esolver='lapack')
```

Finally, we allow the CTQW to propagate over the 3-Cayley tree for $t = 5$, using time-steps of $\Delta t = 0.01$:

```
for dt in np.arange(0.01,t+0.01,0.01):
    walk.propagate(dt,method='chebyshev')
```

³as entanglement is a global measurement, there is a large amount of node communication which may increase overall computation time.

⁴the entanglement watchtype uses an eigenvalue solver, and so accepts `EigSolver` keywords detailed in Table 3.

Table 3: `pyCTQW.MPI` eigensolver keyword arguments

<code>esolver (str)</code>	The default eigensolver algorithm to use. <ul style="list-style-type: none"> • ‘krylovschur’ (<i>default</i>) – Krylov-Schur • ‘arnoldi’ – Arnoldi Method • ‘lanczos’ – Lanczos Method • ‘power’ – Power/Rayleigh Quotient Iteration • ‘gd’ – Generalized Davidson • ‘jd’ – Jacobi-Davidson, • ‘lapack’ – LAPACK eigensolver subroutines • ‘arpack’ – ARPACK subroutines
<code>workType (str)</code>	The eigensolver worktype (either ‘ncv’ or ‘mpd’). The default is to let SLEPc decide.
<code>workSize (int)</code>	Sets the work size if <code>workType</code> is set
<code>tol (float)</code>	Tolerance of the eigensolver (default is to let SLEPc decide)
<code>maxIt (int)</code>	maximum number of iterations of the eigensolver (default is to let SLEPc decide)
<code>verbose (bool)</code>	If True, writes eigensolver information to the console
<code>emax_estimate (float)</code>	Override the calculation of the graphs maximum eigenvalue (must be larger than or equal to the <i>actual</i> maximum eigenvalue for Chebyshev convergence)
<code>emin_estimate (float)</code>	Override the calculation of the graphs minimum eigenvalue (must be smaller than or equal to the <i>actual</i> minimum eigenvalue for Chebyshev convergence)

In this example, we use the Chebyshev propagator, but the Krylov propagator (which is slightly slower) may be used instead. It should also be noted that, as the watch handles are updated every time propagation occurs, the small time-step allows transient data to be collected at a reasonable resolution. However, if the end result of the CTQW propagation is all that is required, it is equally valid to remove the loop and use a time-step $\Delta t = t$, without any loss of accuracy and with some saving in computation time.

Now that the simulation is complete, raw data can be exported and visualisations can be produced, as shown by the following example:

```
# p1 and p2 marginal probabilities at t=5
walk.plot('plot.png')

# 3D graph showing of marginal probability at t=5
walk.plotGraph(output='plot_3D.png')

# p1 and p2 probability over time for node 1
walk.plotNode('plot_node1.png', 1)
# p2 probability over all watched nodes
walk.plotNodes('plot_nodes_particle2.png', p=2)

# plot the entanglement vs. time
walk.plotEntanglement('entanglement.png')

# export the partial trace as a text file
walk.exportPartialTrace('rhoX.txt', 'txt', p=1)
# export the final state to a PETSc binary file
walk.exportState('output_state.bin', 'bin')
```

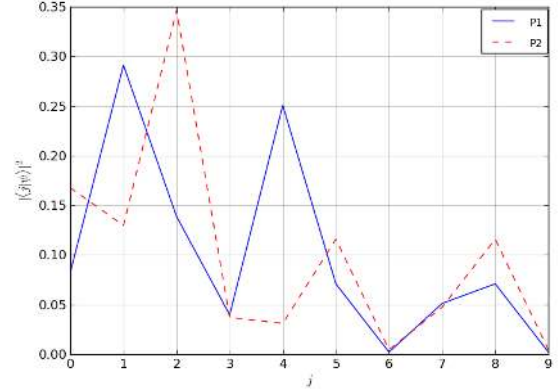
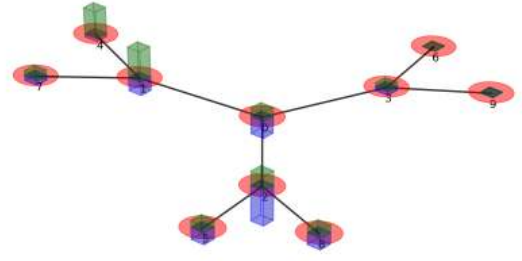


Fig. 2: (Color online) Marginal probability of particles 1 and 2 from the worked example after CTQW propagation over a 3-Cayley tree, as plotted over a 3D representation of the graph (top) and vs. node number (bottom).

A selection of these plots is reproduced in Fig. 2, Fig. 3 and Fig. 4.

Finally, once the CTQW object is no longer needed, memory can be freed by destroying it together with all associated matrices and vectors:

```
walk.destroy()
```

Now that the simulation script is complete, it can be made executable and run on distributed memory systems using `mpirun`.

6. Scaling and performance

In order to effectively determine the performance of the Chebyshev and Krylov exponential methods used in `pyCTQW`, the following study was undertaken comparing the aforementioned algorithms across various implementations. Furthermore, as one of the main design points of `pyCTQW` is to provide a framework for under-the-hood parallelization of CTQW-based simulations, the parallel scaling behaviour of `pyCTQW` was also explored for a variety of input problem sets.

Note that in the following performance study, all computations were completed on a Xeon X5650 based SGI cluster with an InfiniBand low-latency interconnect.

6.1. Library comparison

For this comparison, the problem set chosen is that of a two-particle non-interacting CTQW along a finite line of length N ,

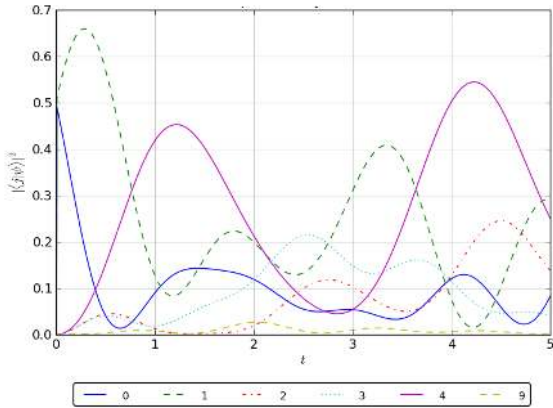


Fig. 3: (Color online) Probability distribution of particle 1 over time, plotted for various graph nodes.

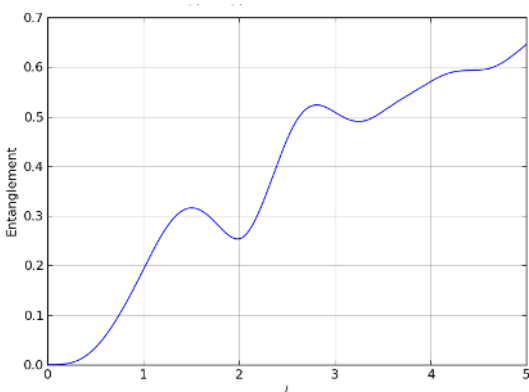


Fig. 4: Von Neumann entropy of the two particle CTQW propagation over a 3-Cayley tree vs. time

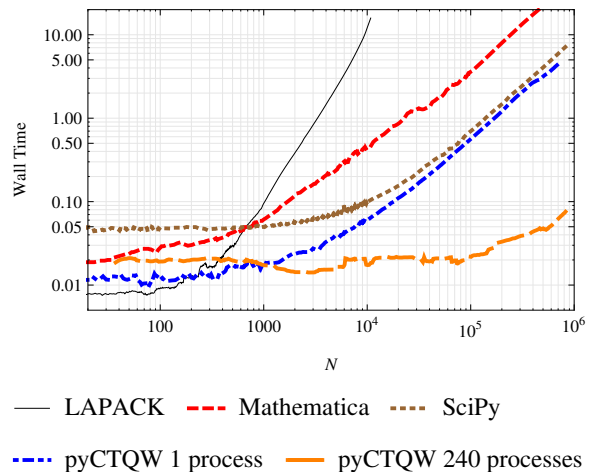


Fig. 5: (Color online) The CPU walltime for a two particle non-interacting CTQW on a finite line of N nodes, propagated for $t = 10$ using a variety of different software libraries/packages as shown by the legend. All simulations are computed using the Chebyshev algorithm, with the exception of *Mathematica* where the in-built `MatrixExp[]` function is used.

with vertices labelled $| -N/2 + 1 \rangle, \dots, |0\rangle, \dots, |N/2\rangle$. This system can be made sufficiently large and complex, and involves the calculation of an $N \times N$ matrix exponential, allowing performance differences to be clearly visible as N scales whilst reducing Hamiltonian construction overheads. To add complexity into the system, defects are placed at vertices $|3\rangle$ and $|4\rangle$ of amplitudes 2 and 1.5, respectively. Propagation is calculated for $t = 10$ from initial entangled state

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}} (|0\rangle \otimes |0\rangle + i|1\rangle \otimes |0\rangle).$$

Fig. 5 shows how the resulting walltime of the system scales as N increases, with the propagation simulated using the Chebyshev algorithm implemented via Intel Fortran/LAPACK, Python/SciPy, and *pyCTQW* (using both a single process and 240 MPI processes on 40 6-core Intel Xeon X5650s). Also plotted for comparison is the same propagation calculated using *Mathematica*'s in-built `MatrixExp[]` function, using the Krylov method and sparse matrices.

It can immediately be seen that *pyCTQW* outperforms all other methods, with the exception of Intel Fortran/LAPACK for $N \lesssim 100$. However, the dense matrix methods available in LAPACK quickly prove unsustainable compared to the other sparse matrix methods for $N \gtrsim 1000$. Considering the remaining implementations, whilst the scaling of SciPy's sparse matrix Chebyshev algorithm scales similarly to a single process of *pyCTQW* for large N (indicating perhaps overhead related to the interpretive nature of Python), the small N behaviour is almost an order of magnitude slower.

Thus, coupled with the two orders of magnitude speedup exhibited by *pyCTQW* when parallelized for $N \sim 10^6$, it can be seen that *pyCTQW* easily outperforms various other implementations for large CTQW systems, whilst still remaining competitive for N small.

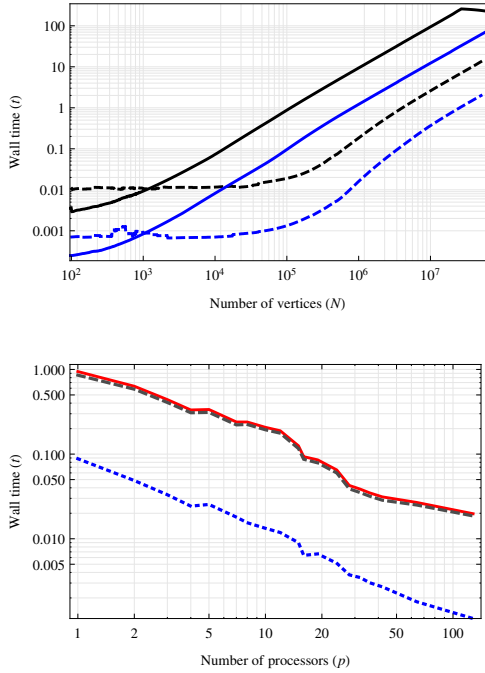


Fig. 6: (Color online) Top: The CPU walltime for a one particle CTQW on a finite line of N nodes, propagated for $t = 5$ using the Krylov (black) and Chebyshev (blue) algorithms for 1 MPI thread (solid line) and 128 MPI threads (dashed line). Bottom: Strong scaling behaviour for problem size $N = 10^5$ shown for the Krylov (black, dashed), Chebyshev (blue, dotted) and Krylov–Schur eigenvalue solver (red, solid).

6.2. Parallel scaling

To analyze the scalability of *pyCTQW*, both one-particle and two-particle CTQWs on a finite line of N vertices were computed using p MPI threads. In both cases, the CTQW was propagated for $t = 5$ with defect $\Gamma = 2|0\rangle\langle 0| + 1.5|1\rangle\langle 1|$. With one particle, propagation commenced from initial state

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle),$$

whereas when two-particles were used, propagation commenced from initial state

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}} (|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$$

with an interaction term of $\Gamma_{int}(1)$.

Fig. 6 highlights the scaling results for the one-particle problem set. Similar to what was seen earlier, the Chebyshev algorithm is approximately one order of magnitude faster than the Krylov algorithm, however both algorithms exhibit analogous scaling behaviour ($\sim O(1/p)$).

The two-particle problem set results are plotted in Fig. 7, this time with parallel speedup (defined as $S_p = T_1/T_p$, where T_p is the walltime for p MPI threads) and parallel efficiency (defined as $E_p = \frac{T_1}{pT_p} = S_p/p$) displayed. Here, it can be seen that both the Krylov and Chebyshev algorithms demonstrate impressive speedup, with little difference between them, although the Chebyshev appears slightly more advantageous. The default

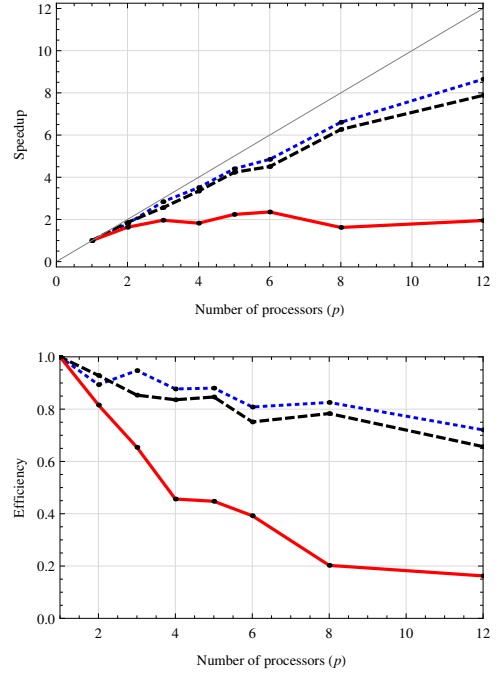


Fig. 7: (Color online) The speedup (top) and efficiency (bottom) for a two particle interacting CTQW on a finite line of $N = 150$ nodes, propagated for $t = 5$ using the Krylov (black, dashed) and Chebyshev (blue, dotted) algorithms, with the Krylov–Schur eigenvalue solver also shown (red, solid). The solid gray diagonal line indicates ideal speedup.

SLEPc eigenvalue solver, in contrast, shows a marked drop in efficiency as the number of processors increases, indicating significant communication overhead. This has the potential to affect the *pyCTQW* simulation time when using the Chebyshev algorithm in the case where the maximum eigenvalues cannot be efficiently calculated.

An accuracy comparison of the Chebyshev and Krylov algorithms has been previously presented by Bergamaschi and Vianello [41], and so has not been reproduced here. Of particular note are their findings that the relative error of the Chebyshev approximation is either comparable or significantly reduced compared to the relative error of the Krylov algorithm, with all errors rising with t . Furthermore, a low sensitivity to eigenvalue estimation on the convergence of the Chebyshev series was also demonstrated, justifying the avoidance of costly eigenvalue solvers that in some cases may result in bottlenecks.

7. Conclusions

Over the last few decades, the fields of quantum information and computation have provided an abundance of theorems and tools that have changed how we model and perceive many great problems in physics. Of these, quantum walks have constituted an important component, resulting in a plethora of quantum algorithms and potential models in a diverse range of fields.

In this paper, we presented a distributed memory software framework, with an object-oriented Python interface *pyCTQW*, that allows the efficient simulation of multi-particle CTQW-based systems. This package also provides easy methods of

visualisation and data analysis. We also detailed the Chebyshev series expansion for the calculation of the matrix exponential, and provided performance data indicating the resulting high degree of speedup achieved over other well established algorithms, such as Krylov-subspace methods.

With future versions of *pyCTQW*, features considered for inclusion include open system quantum walks using superoperators, and extending the allowable number of simultaneous quantum walkers. Furthermore, additional code profiling and testing is likely to result in optimisations and increased performance.

8. Acknowledgements

This work was supported through the use of computing facilities provided by iVEC (the Western Australian Supercomputing Center: Partner Merit Allocation Project 772). The authors would also like to acknowledge valuable discussions with Chris Harris and Chris Bording at iVEC@UWA. JAI would like to thank the Hackett foundation and The University of Western Australia for financial support.

References

- [1] Y. Aharonov, L. Davidovich, N. Zagury, Quantum random walks, *Physical Review A* 48 (1993) 1687–1690.
- [2] J. Kempe, Quantum random walks: An introductory overview, *Contemporary Physics* 44 (2003) 307–327.
- [3] A. M. Childs, E. Farhi, S. Gutmann, An example of the difference between quantum and an example of the difference between quantum and classical random walks, *Quantum Information Processing* 1 (2002) 3543.
- [4] Z. J. Li, J. A. Izaac, J. B. Wang, Position-defect-induced reflection, trapping, transmission, and resonance in quantum walks, *Physical Review A* 87 (2013) 012314.
- [5] J. A. Izaac, J. B. Wang, Z. J. Li, Continuous-time quantum walks with defects and disorder, *PHYSICAL REVIEW A* 88 (2013) 042334.
- [6] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, D. A. Spielman, Exponential algorithmic speedup by a quantum walk, in: *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, STOC '03, ACM, New York, NY, USA, 2003, p. 59–68.
- [7] E. Farhi, J. Goldstone, S. Gutmann, A quantum algorithm for the hamiltonian NAND tree, *arXiv preprint quant-ph/0702144* (2007).
- [8] A. M. Childs, J. Goldstone, Spatial search by quantum walk, *Physical Review A* 70 (2004) 022314.
- [9] S. Berry, J. Wang, Quantum-walk-based search and centrality, *Physical Review A* 82 (2010).
- [10] B. L. Douglas, J. B. Wang, A classical approach to the graph isomorphism problem using quantum walks, *Journal of Physics A: Mathematical and Theoretical* 41 (2008) 075303.
- [11] J. K. Gamble, M. Friesen, D. Zhou, R. Joynt, S. N. Coppersmith, Two-particle quantum walks applied to the graph isomorphism problem, *Physical Review A* 81 (2010).
- [12] S. D. Berry, J. B. Wang, Two-particle quantum walks: Entanglement and graph isomorphism testing, *Physical Review A* 83 (2011) 042317. 00030.
- [13] K. Rudinger, J. Gamble, M. Wellons, E. Bach, M. Friesen, R. Joynt, S. Coppersmith, Noninteracting multiparticle quantum random walks applied to the graph isomorphism problem for strongly regular graphs, *Physical Review A* 86 (2012).
- [14] A. M. Childs, Universal computation by quantum walk, *Physical Review Letters* 102 (2009) 180501.
- [15] A. M. Childs, D. Gosset, Levinson's theorem for graphs II, *arXiv:1203.6557* (2012).
- [16] F. Strauch, Connecting the discrete- and continuous-time quantum walks, *Physical Review A* 74 (2006).
- [17] Y. Bromberg, Y. Lahini, R. Morandotti, Y. Silberberg, Quantum and classical correlations in waveguide lattices, *Physical Review Letters* 102 (2009) 253904.
- [18] A. Peruzzo, M. Lobino, J. C. F. Matthews, N. Matsuda, A. Politi, K. Poulios, X. Zhou, Y. Lahini, N. Ismail, K. Wörhoff, Y. Bromberg, Y. Silberberg, M. G. Thompson, J. L. O'Brien, Quantum walks of correlated photons, *Science* 329 (2010) 1500–1503.
- [19] A. Peruzzo, Multiparticle quantum walks in integrated-waveguide arrays, *SPIE Newsroom* (2010).
- [20] L. Sansoni, F. Sciarrino, G. Vallone, P. Mataloni, A. Crespi, R. Ramponi, R. Osellame, Two-Particle Bosonic-Fermionic quantum walk via integrated photonics, *Physical Review Letters* 108 (2012) 010502.
- [21] M. Karski, L. Förster, J. Choi, A. Steffen, W. Alt, D. Meschede, A. Widera, Quantum walk in position space with single optically trapped atoms, *Science* 325 (2009) 174–177.
- [22] H. Schmitz, R. Matjeschk, C. Schneider, J. Glueckert, M. Enderlein, T. Huber, T. Schaetz, Quantum walk of a trapped ion in phase space, *Physical Review Letters* 103 (2009) 090504.
- [23] M. A. Broome, A. Fedrizzi, B. P. Lanyon, I. Kassal, A. Aspuru-Guzik, A. G. White, Discrete Single-Photon quantum walks with tunable decoherence, *Physical Review Letters* 104 (2010) 153602.
- [24] O. Mülken, A. Blumen, Continuous-time quantum walks: Models for coherent transport on complex networks, *Physics Reports* 502 (2011) 37–87.
- [25] M. Mohseni, P. Rebentrost, S. Lloyd, A. Aspuru-Guzik, Environment-assisted quantum walks in photosynthetic energy transfer, *Journal of Chemical Physics* 129 (2008) 174106.
- [26] M. Sarovar, A. Ishizaki, K. B. Whaley, Quantum entanglement in photosynthetic light-harvesting complexes, *nature Physics* 6 (2010).
- [27] G. S. Engel, T. R. Calhoun, E. L. Read, T. Ahn, T. Mančal, Y. Cheng, R. E. Blankenship, G. R. Fleming, Evidence for wavelike energy transfer through quantum coherence in photosynthetic systems, *Nature* 446 (2007) 782–786.
- [28] E. Collini, C. Y. Wong, K. E. Wilk, P. M. G. Curmi, P. Brumer, G. D. Scholes, Coherently wired light-harvesting in photosynthetic marine algae at ambient temperature, *Nature* 463 (2010) 644–647.
- [29] G. S. Engel, Quantum coherence in photosynthesis, *22nd Solvay Conference on Chemistry* 3 (2011) 222–231.
- [30] P. Peterson, F2PY: a tool for connecting fortran and python programs, *International Journal of Computational Science and Engineering* 4 (2009) 296. 00051.
- [31] E. Farhi, S. Gutmann, Quantum computation and decision trees, *Physical Review A* 58 (1998) 915–928.
- [32] C. Moler, C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, *SIAM Review* 20 (1978) 801–836. Cited by 1123.
- [33] C. Moler, C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, Twenty-Five years later, *SIAM Review* 45 (2003) 3–49. Cited by 0886.
- [34] C. Leforestier, R. Bisseling, C. Cerjan, M. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H. Meyer, N. Lipkin, O. Roncero, R. Kosloff, A comparison of different propagation schemes for the time dependent schrödinger equation, *Journal of Computational Physics* 94 (1991) 59–80.
- [35] J. Fang, One step time propagation method for systems with time-dependent hamiltonians, *Chemical Physics Letters* 263 (1996) 759–766. 00000.
- [36] J. B. Wang, T. T. Scholz, Time-dependent approach to scattering by chebyshev-polynomial expansion and the fast-Fourier-transform algorithm, *Physical Review A* 57 (1998) 3554–3559.
- [37] J. B. Wang, S. Midgley, Quantum waveguide theory: A direct solution to the time-dependent schrödinger equation, *Physical Review B* 60 (1999) 13668–13675.
- [38] H. Guo, Chebyshev propagation and applications to scattering problems, in: A. Lagana, G. Lendvay (Eds.), *Theory of Chemical Reaction Dynamics*, number 145 in NATO Science Series II: Mathematics, Physics and Chemistry, Springer Netherlands, 2005, pp. 217–229. 00002.
- [39] M. Ndong, H. Tal-Ezer, R. Kosloff, C. P. Koch, A chebyshev propagator with iterative time ordering for explicitly time-dependent hamiltonians, *The Journal of Chemical Physics* 132 (2010) 064105. 00009.
- [40] T. Auckenthaler, M. Bader, T. Huckle, A. Spörl, K. Waldherr, Matrix exponentials and parallel prefix computation in a quantum control problem, *Parallel Computing* 36 (2010) 359–369. Cited by 0004.
- [41] L. Bergamaschi, M. Vianello, Efficient computation of the exponential operator for large, sparse, symmetric matrices, *Numerical linear algebra*

- with applications 7 (2000) 27–45. Cited by 0049.
- [42] N. J. Higham, The scaling and squaring method for the matrix exponential revisited, *SIAM Review* 51 (2009) 747–764. Cited by 0040.
 - [43] A. H. Al-Mohy, N. J. Higham, Computing the action of the matrix exponential, with an application to exponential integrators, *SIAM Journal on Scientific Computing* 33 (2011) 488–511. Cited by 0031.
 - [44] R. B. Sidje, Expokit: a software package for computing matrix exponentials, *ACM Transactions on Mathematical Software (TOMS)* 24 (1998) 130–156. Cited by 0386.
 - [45] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), *Modern Software Tools in Scientific Computing*, Birkhäuser Press, 1997, p. 163–202. 00856.
 - [46] S. Balay, J. Brown, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, PETSc web page, <http://www.mcs.anl.gov/petsc/index.html>, 2013.
 - [47] S. Balay, J. Brown, Buschelman, Kris, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, PETSc Users Manual, Technical Report ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013. 01487.
 - [48] V. Hernandez, J. E. Roman, V. Vidal, SLEPc: scalable library for eigenvalue problem computations, *Lecture Notes in Computer Science* 2565 (2003) 377–391. 00020.
 - [49] V. Hernandez, J. E. Roman, V. Vidal, SLEPc: a scalable and flexible toolkit for the solution of eigenvalue problems, *ACM Trans. Math. Software* 31 (2005) 351–362. 00226.
 - [50] J. Izaac, pyCTQW manual, <http://pyctqw.readthedocs.org/>, 2013. 00000.

Phase-modified CTQW unable to distinguish strongly regular graphs efficiently

A Mahasinghe^{1,2}, J A Izaac², J B Wang² and J K Wijerathna¹

¹ Department of Mathematics, University of Colombo, Colombo, Sri Lanka

² School of Physics, The University of Western Australia, Crawley WA 6009, Australia

E-mail: jingbo.wang@uwa.edu.au

Received 9 March 2015, revised 20 April 2015

Accepted for publication 8 May 2015

Published 8 June 2015



CrossMark

Abstract

Various quantum walk-based algorithms have been developed, aiming to distinguish non-isomorphic graphs with polynomial scaling, within both the discrete-time quantum walk (DTQW) and continuous-time quantum walk (CTQW) frameworks. Whilst both the single-particle DTQW and CTQW have failed to distinguish non-isomorphic strongly regular graph families (prompting the move to multi-particle graph isomorphism (GI) algorithms), the single-particle DTQW has been successfully modified by the introduction of a phase factor to distinguish a wide range of graphs in polynomial time. In this paper, we prove that an analogous phase modification to the single particle CTQW does not have the same distinguishing power as its discrete-time counterpart, in particular it cannot distinguish strongly regular graphs with the same family parameters with the same efficiency.

Keywords: quantum walk, graph isomorphism, phase addition

1. Introduction

Graph isomorphism (GI) is an important open problem in mathematics and computer science, with potential applications in a wide variety of fields ranging from network theory to biochemistry—for example, efficient GI algorithms may provide the key to distinguishing molecular representations in chem-informatics [1]. However, the existence of a polynomial-time algorithm for determining GI remains a long-standing unsolved question in computational complexity theory, with the current fastest algorithm scaling as $2^{\mathcal{O}(\sqrt{n \log n})}$ [2]. In fact, the complexity algorithm scaling of the GI problem has yet to be definitively ascertained, though it is conjectured to exist in P and not NP-complete [3]. In the last couple of decades, however, the study of quantum systems has enabled the construction of so-called quantum

algorithms, capable of running exponentially faster than the best known classical algorithms. Most commonly constructed via quantum circuits, an alternate framework for constructing quantum algorithms involves propagating quantum particles on arbitrary graph structures. Referred to as a quantum walk (QW) and proven to provide methods for universal computation [4], they have become an important keystone in the construction of quantum GI algorithms.

The QW is an extension of the Markovian classical random walk to the quantum regime, with the time-evolution of the system instead governed according to the Schrödinger equation. As such, quantum walkers are able to utilize quantum effects (such as superposition and quantum correlation) in order to display drastically differing behaviour to their classical counterparts—for example, QWs propagate in a ballistic manner with quadratic speedup. As with the classical walk, two similar but non-trivially related quantum walk formulations exist; the discrete-time quantum walk (DTQW) [5] and the continuous-time quantum walk (CTQW) [6].

The ability of QWs to distinguish or partition strongly regular graphs (SRGs) is a problem that has been considered numerous times, however single-particle QW algorithms have the tendency to produce identical GI ‘certificates’ for certain non-isomorphic graphs [7–11]. To alleviate this issue, Douglas and Wang [8] developed a perturbed one-particle DTQW-based GI algorithm that was able to successfully distinguish non-isomorphic pairs of SRGs with up to 64 vertices. Multiparticle DTQW algorithms were further explored by Berry and Wang [12]; non-interacting and interacting two-particle DTQWs were used to successfully distinguish specific SRG families. Although lacking the ‘coin flip’ operation found in DTQWs—and thus resulting in a system with reduced degrees of freedom—two-particle interacting CTQW GI algorithms have been proposed that successfully distinguish arbitrary SRGs. Further, it has been proven by Gamble *et al* [7] that a non-interacting CTQW cannot distinguish a pair of SRGs with the same family parameters, contrary to the discrete case.

Whilst the single-particle CTQW was discarded by Gamble *et al* [7] and Shiao *et al* [13] as it was unable to distinguish families of non-isomorphic SRGs with the same family parameters, it is natural to ask: would a *perturbed* one-particle CTQW, modified analogously to the process used by Douglas and Wang [8] for the DTQW, result in a similar improvement in distinguishing power? This has not previously been explored in the literature, due in part to ambiguities in defining the perturbed CTQW system. In this paper, we consider several possible methods of modifying the single-particle CTQW, and prove that none of those considered is as powerful as the phase added DTQW described in Douglas and Wang [8]. Furthermore, we also provide an explanation for the cause of the increased distinguishing power of DTQWs.

2. Graphs and matrix algebras

Let $G(V, E)$ be an undirected graph with vertex set $V = \{v_1, v_2, \dots\}$ and edge set $E = \{(v_i, v_j), \dots\}$ consisting of unordered pairs of connected vertices. A graph G with N vertices is described by its adjacency matrix A , which is an $N \times N$ matrix satisfying

$$A_{uv} = \begin{cases} 1, & \text{if } (u, v) \in E \\ 0, & \text{otherwise} \end{cases}.$$

For completeness, we define a number of distance related measures of the graph:

- The diameter of a graph is the largest number of vertices which must be traversed in order to travel from one vertex to another, and is denoted by d .

- The distance between two vertices u and v is defined as the number of edges in the shortest path connecting them and is denoted by $\text{dist}(u, v)$.
- A graph is said to be distance regular, if, for any pair of vertices u and v , the number of vertices at distance i from vertex u and the number of vertices at distance j from vertex v does not depend on the choice of u and v , just the distance between u and v . Thus, letting $k = \text{dist}(u, v)$, we can then define the intersection number $p_{ij}^{(k)}$.

2.1. The Bose–Mesner Algebra

The Bose–Mesner algebra of a graph G can be constructed as follows. Let M_V denote the algebra of all complex matrices of size $N \times N$, with rows and columns indexed by V , the vertex set of G . Now, define the set of $N \times N$ matrices \mathcal{A}_i such that

$$(\mathcal{A}_i)_{uv} = \begin{cases} 1, & \text{if } \text{dist}(u, v) = i, \\ 0, & \text{otherwise,} \end{cases} \quad i = 0, 1, \dots, d.$$

The Bose–Mesner algebra of G is thus the d -dimensional subalgebra of M_V generated by linear combinations of $\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_d$. In particular, note that:

- $\mathcal{A}_0 = I$ (the $N \times N$ identity matrix)
- $\mathcal{A}_1 = A$ (the adjacency matrix of the graph)
- $\mathcal{A}_0 + \mathcal{A}_1 + \dots + \mathcal{A}_d = J$ (the $N \times N$ all ones matrix)

Furthermore, the relation

$$\mathcal{A}_i \mathcal{A}_j = \sum_{k=0}^d p_{ij}^k \mathcal{A}_k, \tag{1}$$

where p_{ij}^k is the relevant intersection number, holds for the basis matrices of this Bose–Mesner algebra [14].

The dual Bose–Mesner algebra at vertex v_r can be constructed in a similar fashion. Define the diagonal matrices F_0, F_1, \dots, F_d as follows:

$$(F_i)_{uu} = \begin{cases} 1, & \text{if } \text{dist}(v_r, u) = i, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The diagonal matrices F_0, F_1, \dots, F_d form a subalgebra of M_V , referred to as the dual Bose–Mesner algebra of G with respect to the reference vertex v_r [14–16]. It can be observed that this algebra behaves similarly to the Bose–Mesner algebra; for example, the relations $F_0 + F_1 + \dots + F_d = I$ and $F_i F_j = \delta_{ij} F_i$ hold for the dual Bose–Mesner algebra of G with respect to v_r .

2.2. Strongly regular graphs

We can now consider a particular type of graph, known as SRGs. Let G be a k -regular graph on N vertices that is neither a complete graph nor null graph; G is then said to be strongly regular with parameters (N, k, λ, μ) if

- every pair of adjacent vertices (AV) have exactly λ common neighbours,
- every pair of non-AV have exactly μ common neighbours.

As a result of these conditions, it can be seen that the diameter of an SRG is equal to two. Thus, the Bose–Mesner algebra of an SRG has a significantly simpler basis, consisting simply

of \mathcal{A}_0 , \mathcal{A}_1 , and \mathcal{A}_2 . Following from the definition of \mathcal{A}_i above, we have the following relations: $\mathcal{A}_0 = I$, $\mathcal{A}_1 = A$, and $\mathcal{A}_2 = J - \mathcal{A}_0 - \mathcal{A}_1 = J - I - A$. Likewise, the dual Bose–Mesner algebra with respect to reference vertex v_r has three basis elements, the diagonal matrices F_0 , F_1 , and F_2 , with each F_i containing the v_r th row (or equivalently the v_r th column) of \mathcal{A}_i along the diagonal.

3. Quantum walks

QWs are the quantum analogue of classical random walks, with time-evolution governed by the Schrödinger equation rather than the Markovian master equation. Over the last decade, QWs have received a significantly increased research focus, due in part to its applicability as a computational model in quantum computation, as well as a wide range of potential applications—including network theory, computer science, quantum information and biochemistry. In particular, the application of QWs in GI testing has been especially noteworthy.

3.1. Discrete-time QWs

The quantum analogue of the discrete-time classical random walk, the DTQW, takes place in the Hilbert space \mathcal{H} , comprised of the tensor product of the space spanned by orthonormal set of vertex states $\{|v_i\rangle: i = 1, 2, \dots, N\}$ and the coin Hilbert space spanned by orthonormal set of coin states $\{|c_i\rangle: i = 1, 2, \dots, d_i\}$, with d_i the degree of vertex v_i . The DTQW on the graph is then achieved via repeated application of the unitary time-evolution operator $\hat{U} = \hat{S} \cdot (1 \otimes \hat{C})$, where C is the coin operator, and S is the shift operator acting on the Hilbert space \mathcal{H} as $\hat{S}|v_i, c_j\rangle = |v_j, c_i\rangle$.

The DTQW unitary operator can alternatively be expressed on the space of all directed edges of the graph; i.e. the space formed by all ordered tuples of vertices in the graph, and hence this representation is by a matrix of dimensions $N^2 \times N^2$. For example, in the case of a d -degree Grover coin ($\hat{C}_G^{(d)} \in \mathbb{R}^{d \times d}$, $(\hat{C}_G^{(d)})_{ij} = 2/d - \delta_{ij}$) applied to a d -degree regular graph, the DTQW unitary time-step operator is expressible as follows:

$$(U)_{ij,kl} = \left[\hat{S} \cdot (1 \otimes \hat{C}_G^{(d)}) \right]_{ij,kl} = \begin{cases} \frac{2}{d_j} - \delta_{il}, & \text{if } j = k, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

where $(i, j), (k, l) \in E$ and d_j denotes the degree of the j th vertex. This form, which is equivalent to the above definition, has been adopted by Emms *et al* [9] and Smith [17].

Various methods have been previously used in the construction of graph certificates (certificates used to uniquely identify a graph and its isomorphisms) in DTQW-based GI algorithms. It has been observed that U and \hat{U} , the two unitary time evolution operators for a pair of non-isomorphic SRGs with same family parameters, share a number of common properties, with the most explicit example being cospectrality. One such method is shown by Emms *et al* [9], whereupon U^n and \hat{U}^n are calculated and the resulting spectra compared, successfully distinguishing certain pairs of SRGs. Alternatively, using a phase-added DTQW modification, Douglas and Wang [8] compared the resulting probabilities at timestep $t = N^2$, and successfully distinguished certain pairs of SRGs with the same family parameters.

3.2. Continuous-time QWs

A separate but non-trivially related QW formulation is the CTQW, itself based on continuous-time classical random walks. As with the DTQW, the CTQW has also been used for GI testing, most commonly by constructing an unbiased graph certificate based on the CTQW's time evolution operator acting on the graph Hamiltonian. Amongst the literature, there are several competing definitions of the Hamiltonian, the most popular including $H = \pm A$ and $H = \pm L$, where A is the adjacency matrix and L is the Laplacian of the graph (constructed via $L = D - A$, with D the diagonal matrix defined by $D_i = \sum_{j=0}^N A_{ij}$). Note that in the case of regular graphs of degree λ , the Laplacian can be written $L = D - A = \lambda I - A$, and thus the time evolution operator becomes $U(t) = e^{-iLt} = e^{-i(\lambda I - A)t} = e^{-i\lambda t} e^{iAt}$ —proportional to e^{iAt} up to a global phase factor. Therefore, as we are working with SRGs, selecting $H = A$ for simplicity and without loss of generality, the time evolution operator of the CTQW is determined by solving the Schrödinger equation, and is expressed as $U(t) = e^{-iHt}$.

Due to the unknown vertex-labelling permutation relation between two isomorphic graphs, the graph certificate is generally constructed by performing a frequency measurement on the resulting probability distribution of the evolved QW—i.e. producing an ordered list. This approach has been adapted by Shiao *et al* [13], Rudinger *et al* [11], Rudinger *et al* [10], Gamble *et al* [7], Douglas and Wang [8], and Berry and Wang [12], all aiming at solving the GI problem in polynomial time by exploiting massive quantum parallelism. These algorithms construct the GI certificates by essentially flattening the matrix of transition amplitudes and discarding detailed structural information, which are therefore efficient (i.e. in polynomial time) but not proven to be universal. In other words, they may ultimately fail to distinguish certain families of non-isomorphic graphs. As such, they only provide a necessary condition of isomorphism, but not a necessary and sufficient condition that the sets of elements of two isomorphic graphs are equivalent.

Of course, modifying the algorithm to account for the detailed structure of $\langle i|U(t)|j \rangle$ would most likely be advantageous, and may be able to distinguish non-isomorphic graphs that produce CTQW time-evolutions with the same list of elements. However, to our knowledge, there does not exist any proposed CTQW GI algorithm that makes use of the structural information in an efficient manner, due to the unknown vertex-labelling permutation relation between two graphs. One could always test all possible permutations of rows and columns between the two $\langle i|U(t)|j \rangle$ matrices, but this would require factorial/exponential time and thus offers no advantage over classical algorithms.

4. QWs with phase

In order to determine whether a QW will be successful in distinguishing two non-isomorphic graphs, it is sufficient to prove that the resulting GI certificates (for example the produced sets of transition amplitudes as described above) do not contain the same elements. In the case of the CTQW and SRGs, it can be demonstrated from the Bose–Mesner algebra of SRGs (section 2.2) that the exponentiated adjacency matrices for two graphs within the same SRG(N, k, λ, μ) family produce the same sorted list of entries—and hence the time evolutions are identical and the graphs are indistinguishable. In order to address this problem, we will attempt to modify the QW evolution in a manner analogous to that performed by Douglas and Wang [8] in the discrete case.

Douglas and Wang [8] considered the possibilities of modifying the DTQW without extending the system to two particles in order to increase its distinguishing power.

Consequently, they successfully modified the single particle DTQW by adding a phase, distinguishing several non-isomorphic SRGs with the same family parameters. In their algorithm, a vertex is selected as the reference vertex and a local phase is applied to all vertices adjacent to it; i.e. $S|v_i, c_j\rangle = e^{i\theta}|v_j, c_i\rangle$. The graph certificate is then taken as the probabilities at the pairs of vertices $\{u, v\}$ of the graph, which makes it uniform up to a vertex permutation.

It is natural to ask whether the continuous counterpart of this phase addition could be successfully used for GI testing. We consider several possible algorithms for CTQW phase additions, restricting ourselves to those which are uniform for permutations of the vertices. To construct the graph certificate, we then utilize transition amplitudes between all pairs of vertices. It should be noted that this is equivalent to the DTQW graph certificate construction as used by Douglas and Wang [8]—thus assuming this follow through in continuous time, we conclude that whenever the method used by Douglas and Wang [8] can distinguish two vertex transitive graphs (for example the first few distinct pairs of SRGs within the same family), so should the method of CTQW phase additions.

We will begin by proving, using the Bose–Mesner algebra, that two non-isomorphic SRGs with the same family parameters cannot be distinguished, before arriving at the explicit continuous counterpart of the phase added DTQW. Finally we consider general phase additions, and compare and contrast the two methods.

Lemma 1. *The ordered transition amplitude lists of a single-particle CTQW are identical for two non-isomorphic SRGs with the same family parameters.*

Proof. Recall that the Hamiltonian is expressible as $H = A$. Since the basis elements of the Bose–Mesner algebra obey the relations

$$\mathcal{A}_i \mathcal{A}_j = \sum_{k=0}^2 p_{ij}^{(k)} \mathcal{A}_k, \quad (4)$$

and $\mathcal{A}_0 = I$, $\mathcal{A}_1 = A$ and $\mathcal{A}_2 = J - I - A$, the time evolution operator can be written as

$$U(t) = e^{-iAt} = \sum_{n=0}^{\infty} \frac{1}{n!} (-it)^n \mathcal{A}^n = \alpha \mathcal{A}_0 + \beta \mathcal{A}_1 + \gamma \mathcal{A}_2, \quad (5)$$

where α , β and γ are time dependent and depend *only* upon the family parameters (N, k, λ, μ) of the SRG. The graph certificate is thus given by $\{\langle i|U(t)|j\rangle: i, j \in V\}$, i.e. the list of elements of the matrix $U(t)$. We can now calculate the diagonal and off-diagonal elements of U :

- Diagonal elements (i.e. $i=j$): as $(\mathcal{A}_1)_{jj} = (\mathcal{A}_2)_{jj} = 0$ and $(\mathcal{A}_0)_{jj} = 1 \forall j$, we therefore have N diagonal elements equal to α .
- Off-diagonal elements ($i \neq j$): we have $(\mathcal{A}_0)_{ij} = 0 \forall i, j$, whilst \mathcal{A}_1 and \mathcal{A}_2 contains a mixture of zeros and ones. Since the SRG is k -regular, the adjacency matrix will contain k non-zero elements for each N vertex. Thus, there will be kN off-diagonal elements equal to β , and $N^2 - N - kN$ off-diagonal elements equal to γ .

Thus, $\{\langle i|U(t)|j\rangle: i, j \in V\}$ contains N elements of α , kN elements of β , and $N^2 - N - kN$ elements of γ . As α , β and γ depend only on (N, k, λ, μ) , two non-isomorphic SRGs with identical family parameters will produce the same time evolution and therefore GI

certificate. Hence, two SRGs with same family parameters are not distinguished the CTQW. \square

As expected, the Bose–Mesner approach provides the same conclusion as previously shown by Gamble *et al* [7].

4.1. SV-phase addition

The first possible option for phase addition includes adding a local phase factor at a selected vertex. We shall call this SV (selected vertex) phase addition. The Hamiltonian in this case is simply the adjacency matrix with phase added along the diagonal at a *single* vertex (the reference vertex); this can be expressed as $H = A + \theta F_0$, and hence the time evolution operator is $U_{SV}(t, \theta) = e^{-i(A+\theta F_0)t}$. Note that it follows that if the two graphs to be tested are vertex transitive, comparing the transition amplitudes at the vertices is sufficient to measure the strength of the SV-phase added CTQW.

Considering pairs of SRGs with same family parameters, the Shrikhande graph and the (4, 4)-lattice graph provide the non-isomorphic pair of $SRG(N, k, \lambda, \mu)$ with minimum N . For these graphs, the family parameters are given by $N = 16, k = 6, \lambda = \mu = 2$, and both these graphs are vertex transitive. Therefore, the transition amplitudes will provide an unbiased graph certificate. The effect of SV-phase added CTQW on this pair is described in lemma 2.

Lemma 2. *The ordered transition amplitude lists of the SV-phase added CTQW are identical for two non-isomorphic SRGs with parameters (16, 6, 2, 2).*

Proof. Recall the Hamiltonian for SV-phase added CTQW is given by $A + \theta F_0$. Consider the terms in the expansion of the SV time evolution operator:

$$U_{SV} = \sum_{n=0}^{\infty} \frac{1}{n!} (-it)^n (A + \theta F_0)^n = \left[I - it(\mathcal{A}_1 + \theta F_0) + \frac{(-it)^2}{2!} (\mathcal{A}_1 + \theta F_0)^2 + \dots \right]. \quad (6)$$

Let us use mathematical induction on n to prove the following result

$$(A + \theta F_0)^n = \sum_{j=0}^2 \alpha_j^{(n)} \mathcal{A}_j + \sum_{j=0}^2 \beta_j^{(n)} F_j + \sum_{j,k=0}^2 \gamma_{j,k}^{(n)} F_j \mathcal{A}_k + \chi^{(n)} \mathcal{A}_1 F_0 + \epsilon^{(n)} \mathcal{A}_2 F_0 + \zeta^{(n)} \mathcal{A}_1 F_0 \mathcal{A}_1 + \eta^{(n)} \mathcal{A}_2 F_0 \mathcal{A}_1, \quad (7)$$

where $\alpha_j^{(n)}, \beta_j^{(n)}, \gamma_{j,k}^{(n)}, \chi^{(n)}, \epsilon^{(n)}$ and $\zeta^{(n)}$ are coefficients depending only upon the family parameters of the SRG. By requiring $\alpha_1^{(1)} = 1, \beta_0^{(1)} = \theta,$ and $\gamma_{j,k}^{(1)} = \chi^{(1)} = \epsilon^{(1)} = \zeta^{(1)} = \eta^{(1)} = 0,$ it can be seen that for the case $n = 1,$ (7) reduces to

$$A + \theta F_0 = \mathcal{A}_1 + \theta F_0. \quad (8)$$

Thus (7) holds true for $n = 1.$

Now, assuming it is true for $n,$ consider the case $n + 1.$ The terms of the expansion of $(A + \theta F_0)^{n+1}$ are derived by right-multiplying each term in (7) by \mathcal{A}_1 and F_0 respectively. Since $\{\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2\}$ form the basis of the Bose–Mesner algebra of SRGs, we can make use of the multiplication closure relation $\mathcal{A}_i \mathcal{A}_j = \sum_{k=0}^d p_{ij}^k \mathcal{A}_k.$ Thus, multiplying the first term by $\mathcal{A}_1,$ we get

$$\left(\sum_{j=0}^2 \alpha_j^{(n)} \mathcal{A}_j \right) \mathcal{A}_1 = \sum_{j=0}^2 \alpha_j^{(n+1)} \mathcal{A}_j, \tag{9}$$

where $\alpha_j^{(n+1)}$ denotes a new coefficient signifying its continued contribution to the first term in the expansion (7). Similarly, right-multiplying the second and third terms of the expansion by \mathcal{A}_1 , we find that they contribute to the $F_j \mathcal{A}_k$ double summation term:

$$\left(\sum_{j=0}^2 \beta_j^{(n)} F_j \right) \mathcal{A}_1 = \sum_{j=0}^2 \gamma_{a_{j,1}}^{(n+1)} F_j \mathcal{A}_1 \tag{10a}$$

$$\left(\sum_{j,k=0}^2 \gamma_j^{(n)} F_j \mathcal{A}_k \right) \mathcal{A}_1 = \sum_{j,k=0}^2 \gamma_{b_{j,k}}^{(n+1)} F_j \mathcal{A}_k, \tag{10b}$$

where $\gamma_{a_{j,k}}^{(n+1)} + \gamma_{b_{j,k}}^{(n+1)} + \gamma_{c_{j,k}}^{(n+1)} = \gamma_{j,k}^{(n+1)}$ ($\gamma_{c_{j,k}}^{(n+1)}$, the final contribution to $\gamma_{j,k}^{(n+1)}$, will be defined later). Similarly, after right-multiplication by \mathcal{A}_1 the fourth and fifth terms can now be written in terms of the sixth and seventh terms of (7), respectively:

$$\left(\chi^{(n)} \mathcal{A}_1 F_0 \right) \mathcal{A}_1 = \zeta^{(n+1)} \mathcal{A}_1 F_0 \mathcal{A}_1 \tag{11a}$$

$$\left(\epsilon^{(n)} \mathcal{A}_2 F_0 \right) \mathcal{A}_1 = \eta^{(n+1)} \mathcal{A}_2 F_0 \mathcal{A}_1. \tag{11b}$$

For the case SRG (16, 6, 2, 2), it can be shown that

$$\mathcal{A}_1 F_0 \mathcal{A}_1^2 = 2(F_1 \mathcal{A}_1 + F_1 \mathcal{A}_2 + 2\mathcal{A}_1 F_0 + F_1) \tag{12a}$$

$$\mathcal{A}_2 F_0 \mathcal{A}_1^2 = 2(F_2 \mathcal{A}_1 + F_2 \mathcal{A}_2 + 2\mathcal{A}_2 F_0 + F_2). \tag{12b}$$

Thus, multiplying the sixth term of (7) by \mathcal{A}_1 in the case of SRG (16, 6, 2, 2), and expanding in terms of the basis elements

$$\left(\zeta^n \mathcal{A}_1 F_0 \mathcal{A}_1 \right) \mathcal{A}_1 = \beta_{a_1}^{(n+1)} F_1 + \gamma_{c_{1,1}}^{(n+1)} F_1 \mathcal{A}_1 + \gamma_{c_{1,2}}^{(n+1)} F_1 \mathcal{A}_2 + 2\chi_a^{(n+1)} \mathcal{A}_1 F_0 \tag{13a}$$

$$\left(\zeta^n \mathcal{A}_2 F_0 \mathcal{A}_1 \right) \mathcal{A}_1 = \beta_{a_2}^{(n+1)} F_2 + \gamma_{c_{2,1}}^{(n+1)} F_2 \mathcal{A}_1 + \gamma_{c_{2,2}}^{(n+1)} F_2 \mathcal{A}_2 + 2\epsilon_a^{(n+1)} \mathcal{A}_2 F_0. \tag{13b}$$

We now return to the general case of an arbitrary SRG, and repeat the previous procedure—this time right-multiplying the terms in (7) by \mathcal{F}_1 :

$$\left(\sum_{j=0}^2 \alpha_j^{(n)} \mathcal{A}_j \right) \mathcal{F}_0 = \beta_{a_0}^{(n+1)} \mathcal{F}_0 + \chi_b^{(n+1)} \mathcal{A}_1 \mathcal{F}_0 + \epsilon_b^{(n+1)} \mathcal{A}_2 \mathcal{F}_0. \tag{14}$$

Now we make use of the property $F_j F_k = \delta_{jk} F_j$ of the dual Bose–Mesner algebra at the reference vertex v_r , and establish the following:

$$\left(\sum_{j=0}^2 \beta_j^{(n)} F_j \right) \mathcal{F}_0 = \beta_{b_0}^{(n+1)} \mathcal{F}_0. \tag{15}$$

Considering the three-fold products of the form $F_j \mathcal{A}_k F_l$, we can now make use of the fact that the (u, v) th entry of this three-fold product is expressible as

$$(F_j \mathcal{A}_k F_l)_{uv} = \begin{cases} 1, & \text{if } u \in X_j(x), v \in X_l(x) \text{ and } u \in X_k(v), \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

(where $X_j(x)$ denotes the set of vertices in the graph located at a distance j from the vertex x [18, 19]) to deduce the following:

$$F_0 \mathcal{A}_1 F_0 = F_2 \mathcal{A}_1 F_0 = F_0 \mathcal{A}_2 F_0 = F_1 \mathcal{A}_2 F_0 = O. \quad (17)$$

Further, since

$$(\mathcal{A}_j F_k)_{uv} = \begin{cases} 1, & \text{if } \text{dist}(u, v) = j, \text{ and } \text{dist}(v, x) = k, \\ 0, & \text{otherwise,} \end{cases} \quad (18)$$

we deduce that

$$F_1 \mathcal{A}_1 F_0 = \mathcal{A}_1 F_0 \quad (19a)$$

$$F_2 \mathcal{A}_2 F_0 = \mathcal{A}_2 F_0. \quad (19b)$$

Recall that $\mathcal{A}_0 = I$; it follows that

$$F_j \mathcal{A}_0 F_0 = F_j F_0 = \begin{cases} F_0, & j = 0, \\ O, & j = 1, 2. \end{cases} \quad (20)$$

Using this fact with (17) and (19), we establish the following:

$$\left(\sum_{j,k=0}^2 \gamma_{j,k}^{(n)} F_j \mathcal{A}_k \right) F_0 = \beta_{c_0}^{(n+1)} F_0 + \chi_c^{(n+1)} \mathcal{A}_1 F_0 + \epsilon_c^{(n+1)} \mathcal{A}_2 F_0. \quad (21)$$

Since $F_j F_k = \delta_{jk} F_j$, the terms $\mathcal{A}_1 F_0$ and $\mathcal{A}_2 F_0$ undergo only a change of coefficients when right-multiplied by F_0 . Finally, in the specific case of SRG (16, 6, 2, 2), it can be shown that

$$\mathcal{A}_1 F_0 \mathcal{A}_1 F_0 = \mathcal{A}_2 F_0 \mathcal{A}_1 F_0 = O. \quad (22)$$

Now, using equations (9–11), (13–15), (21, 22), we establish that

$$(A + \theta F_0)^{n+1} = \sum_{j=0}^2 \alpha_j^{(n+1)} \mathcal{A}_j + \sum_{j=0}^2 \beta_j^{(n+1)} F_j + \sum_{j,k=0}^2 \gamma_{j,k}^{(n+1)} F_j \mathcal{A}_k + \chi^{(n+1)} \mathcal{A}_1 F_0 + \epsilon^{(n+1)} \mathcal{A}_2 F_0 + \zeta^{(n+1)} \mathcal{A}_1 F_0 \mathcal{A}_1 + \eta^{(n+1)} \mathcal{A}_2 F_0 \mathcal{A}_1. \quad (23)$$

Therefore, by the principle of mathematical induction, we establish (7). Next, we consider the term $\mathcal{A}_2 F_0 \mathcal{A}_1$; this three-fold product is expressible as a combination of singletons, two-fold products and the three-fold product $\mathcal{A}_1 F_0 \mathcal{A}_1$:

$$\mathcal{A}_2 F_0 \mathcal{A}_1 = F_1 + \mathcal{A}_1 F_1 + \mathcal{A}_2 F_2 - F_0 \mathcal{A}_1 - \mathcal{A}_1 F_0 \mathcal{A}_1. \quad (24)$$

Observe that the singletons can be expressed as two-fold products, as $\mathcal{A}_0 = F_0 + F_1 + F_2 = I$. Thus, it follows that $(A + \theta F_0)^n$ is expressible as a linear combination of two-fold products and the single three-fold product $\mathcal{A}_1 F_0 \mathcal{A}_1$; and therefore we express $(-it)^n (A + \theta F_0)^n$ as follows;

$$(-it)^n (A + \theta F_0)^n = \sum_{j,k=0}^2 \left(\alpha_{j,k}^{(n)} \mathcal{A}_j F_k + \beta_{j,k}^{(n)} F_j \mathcal{A}_k + \gamma^{(n)} \mathcal{A}_1 F_0 \mathcal{A}_1 \right). \quad (25)$$

Hence, summing over all n , the SV time evolution operator is expressible as

$$U_{SV}(t, \theta) = - \sum_{j,k=0}^2 \left(\alpha_{j,k} \mathcal{A}_j F_k + \beta_{j,k} F_j \mathcal{A}_k + \gamma \mathcal{A}_1 F_0 \mathcal{A}_1 \right), \quad (26)$$

where $\alpha_{j,k}$, $\beta_{j,k}$ and γ depend on the family parameters of the SRG, and implicitly on time t and phase θ .

Now we calculate the contribution of each term in this summation. Since each matrix is comprised solely of ones and zeros, an entry of the matrix U is a summation of $\alpha_{j,k}$, $\beta_{j,k}$, and γ . The contribution of each element can be determined both combinatorially and computationally. Table 1 provides a list of all possible elements in the U_{SV} expansion of graphs in the SRG (16, 6, 2, 2) family. As expected, since $\alpha_{j,k}$, $\beta_{j,k}$ and γ depend *only* on (N, k, λ, μ) , each element in the table occurs an equal number of times in the summation for both U_{SV} and \hat{U}_{SV} . Thus it follows that the two SRGs with parameters (16, 6, 2, 2) are not distinguishable by the SV-phase added CTQW. \square

4.2. AV-phase addition

In this section, a phase θ will be added to all vertices adjacent to a selected *reference vertex*; let us refer to this method as the AV phase added CTQW. Note that this is the continuous analogue of the perturbed DTQW GI algorithm by Douglas and Wang [8], which successfully distinguished all members of SRGs tested.

Mathematically, the AV-phase added CTQW can be achieved by replacing the ones in the v_r th row and column of the Hamiltonian by the phase θ . Consequently, the Hamiltonian for the AV-phase added CTQW is expressible as

$$(H_{AV})_{uv} = \begin{cases} \theta A_{uv} & \text{if } u = v_r \text{ or } v = v_r, \\ A_{uv} & \text{otherwise.} \end{cases} \quad (27)$$

An alternative form of expressing the Hamiltonian is through the use of the elements of the Bose–Mesner algebra and its dual at the reference vertex as discussed in the previous section. This allows us to express the AV-phase added Hamiltonian via the expansion

$$H_{AV} = \mathcal{A}_1 + \theta(F_0 \mathcal{A}_1 + \mathcal{A}_1 F_0). \quad (28)$$

Since the two graphs are vertex transitive, we again take the transition amplitudes over all vertices as the graph certificate. We then prove that the AV-phase added CTQW is not as powerful as its discrete time equivalent, the phase-added DTQW used by Douglas and Wang [8]. Lemma 3 gives the desired result.

Lemma 3. *The ordered transition amplitude lists of the AV-phase added CTQW are identical for two non-isomorphic SRGs with parameters (16, 6, 2, 2).*

Proof. Recall that the AV unitary operator is expressible as

$$U_{AV}(t, \theta) = e^{-i[\mathcal{A}_1 + \theta(F_0 \mathcal{A}_1 + \mathcal{A}_1 F_0)]t}, \quad (29)$$

Table 1. Number of occurrences of all possible coefficients in the expansion of the selected vertex (SV) CTQW unitary operator (26) of two non-isomorphic graphs G and \hat{G} in the SRG(16, 6, 2, 2) family.

Element	Occurrences in U_{SV}	Occurrences in \hat{U}_{SV}
$\alpha_{00} + \beta_{00}$	1	1
$\alpha_{11} + \beta_{01}$	6	6
$\alpha_{22} + \beta_{02}$	9	9
$\alpha_{10} + \beta_{11}$	6	6
$\alpha_{12} + \beta_{11}$	18	18
$\alpha_{22} + \beta_{12}$	36	36
$\alpha_{02} + \beta_{20}$	9	9
$\alpha_{11} + \beta_{21}$	18	18
$\alpha_{12} + \beta_{21}$	36	36
$\alpha_{20} + \beta_{22}$	9	9
$\alpha_{21} + \beta_{22}$	36	36
$\alpha_{22} + \beta_{22}$	36	36
$\alpha_{01} + \beta_{10} + \gamma$	6	6
$\alpha_{11} + \beta_{11} + \gamma$	12	12
$\alpha_{21} + \beta_{12} + \gamma$	18	18

therefore

$$U_{AV}(t, \theta) = I - it \left[\mathcal{A}_1 + \theta(F_0 \mathcal{A}_1 + \mathcal{A}_1 F_0) \right] + \frac{(-it)^2}{2!} \left[\mathcal{A}_1 + \theta(F_0 \mathcal{A}_1 + \mathcal{A}_1 F_0) \right]^2 + \dots \tag{30}$$

Similar to the SV-phase added CTQW, all the higher powers cancel out, reducing to a linear combination of singletons, two-fold products and a single three-fold product again;

$$U_{AV}(t, \theta) = \sum_{j,k=0}^2 \left(\hat{\alpha}_{j,k} \mathcal{A}_i F_j + \hat{\beta}_{j,k} F_i \mathcal{A}_j + \hat{\gamma} \mathcal{A}_1 F_0 \mathcal{A}_1 \right), \tag{31}$$

where, as before, $\hat{\alpha}_{j,k}$, $\hat{\beta}_{j,k}$ and $\hat{\gamma}$ depend on the family parameters of the SRG, and implicitly on time t and phase θ . Therefore, following the same argument as detailed in the proof of lemma 2, the set of transition amplitudes and thus the graph certificates for two non-isomorphic graphs within SRG (16, 6, 2, 2) are identical. \square

5. Discussions and conclusion

It is well known that a standard single-particle DTQW or CTQW cannot distinguish a pair of SRGs with the same family parameters, and furthermore it has been proven by Gamble *et al* [7] that a two particle CTQW with no interaction cannot distinguish such a pair. In the case of the single particle DTQW, variations (such as the addition of phases at a reference vertex) lead to an increased distinguishing power, providing single-particle DTQWs with the ability to distinguish the aforementioned SRGs [8]. However, no analogous modifications had been

previously explored in the case of the single particle CTQW; whilst considered unlikely, it was hitherto unknown for sure whether the continuous counterpart of a single-particle DTQW could be modified to produce a similar increase in distinguishing ability.

In this study, it was proven that various phase-added CTQWs are unable to distinguish non-isomorphic SRGs with family parameters (16, 6, 2, 2), the smallest such set of family parameters with a distinct disadvantage compared to the equivalent DTQW algorithm. Note that although the two-fold and singleton Bose–Mesner products derived in lemma 2 are specific to SRG (16, 6, 2, 2), equivalent relations continue to hold in the case of general SRG parameters—allowing the exponentiated Hamiltonian to remain written as linear combinations of k -fold products. Consequently, the time evolution will persist in producing identical graph certificates for two non-isomorphic SRGs with same parameters; this inference has been supported by numerical computation.

Instead of the phase additions discussed here, one may consider a general phase addition to the single particle CTQW. The corresponding adjacency matrix is expressible as $A \circ P_1 \circ P_2 \circ \dots \circ P_k$, where A is the adjacency matrix, P_i is the i th phase matrix (produced by adding $\theta_i - 1$ to specified elements of J), and \circ is the Hadamard product of matrices (defined as $(A \circ B)_{ij} = A_{ij}B_{ij}$). As long as this belongs to the subalgebra of M_V generated by the basis elements of the Bose–Mesner algebra and the dual Bose–Mesner algebra at the reference vertex v_r , the general phase-modified CTQW will be of similar form to the phase modifications described in this paper. This particular algebra $\langle \mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_d, F_0, F_1, \dots, F_d \rangle$ was first studied by Terwilliger [18], and is therefore referred to as the Terwilliger algebra of the graph with respect to v_r . It should be noted that in most studies the dual Bose–Mesner algebra is analyzed in conjunction with the Bose–Mesner algebra, thus leading towards the Terwilliger algebra [14–16]. Considering the general phase addition, whenever $A \circ P_1 \circ P_2 \circ \dots \circ P_k$ does not belong to the Terwilliger algebra, the exponentiated Hamiltonians of SRGs with same parameters are expected to be distinguishable. Therefore, for a pair of SRGs to be distinguishable, the CTQW has to be phase-modified in a way that $A \circ P_1 \circ P_2 \circ \dots \circ P_k$ does not belong to the Terwilliger algebra.

In order to explain the reduced distinguishing power of the single-particle phase-added CTQW compared to the discrete case, it is worth mentioning that the two QWs evolve in different spaces; even confined to a single-particle system, the Hamiltonian of the CTQW is simply the adjacency matrix of the graph, whilst the discrete-time Hamiltonian is a significantly more complicated expression given by

$$H = \left(\frac{2}{k}J + I \right) \circ (A \otimes I \cdot \Delta_I \cdot A \otimes I), \tag{32}$$

where Δ_I has the same dimensions as $A \otimes I$, and the $(ij)(kl)$ th element is given by

$$(\Delta_I)_{ij,kl} = \begin{cases} 1 & \text{if } i = j = k = l, \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

and \circ is the Hadamard product [17]. Thus, it can be seen that the DTQW evolves in a higher dimensional space compared to the CTQW, allowing it to possess extra distinguishing power. Hence, it follows that a phase addition to the DTQW Hamiltonian is not reducible, as is the case for the CTQW—the complexity contributed to the structure of the DTQW Hamiltonian by the Δ_I term seemingly greatly affects the time evolution of the discrete time walk.

Considering these results, we conclude that the DTQW is significantly more powerful than CTQWs for distinguishing non-isomorphic graphs. Although the DTQW has its own limitations (as shown in [9] and [17]), it has intrinsic extra power compared to the CTQW, due to its time evolution occurring in a larger dimensional Hilbert space.

Acknowledgments

The authors would like to thank the University of Colombo and the University of Western Australia for financial support, in particular through the University of Colombo doctoral grants.

References

- [1] Dias J R and Milne G W A 1992 *J. Chem. Inf. Comput. Sci.* **32** 00004
- [2] Johnson D S 2005 *ACM Trans. Algorithms* **1** 00023
- [3] Schöning U 1987 *STACS 87 (Lecture Notes in Computer Science No. 27)* ed F J Brandenburg, G Vidal-Naquet and M Wirsing (Berlin: Springer) pp 114–124
- [4] Childs A M 2009 *Phys. Rev. Lett.* **102** 180501
- [5] Aharonov Y, Davidovich L and Zagury N 1993 *Phys. Rev. A* **48** 1687–90
- [6] Farhi E and Gutmann S 1998 *Phys. Rev. A* **58** 915
- [7] Gamble J K, Friesen M, Zhou D, Joynt R and Coppersmith S N 2010 *Phys. Rev. A* **81** 052313
- [8] Douglas B L and Wang J B 2008 *J. Phys. A: Math. Theor.* **41** 075303
- [9] Emms D, Severini S, Wilson R C and Hancock E R 2009 *Pattern Recognit.* **42** 1988
- [10] Rudinger K, Gamble J, Wellons M, Bach E, Friesen M, Joynt R and Coppersmith S 2012 *Phys. Rev. A* **86** 022334
- [11] Rudinger K, Gamble J K, Bach E, Friesen M, Joynt R and Coppersmith S N 2013 *J. Comput. Theor. Nanoscience* **10** 1653
- [12] Berry S D and Wang J B 2011 *Phys. Rev. A* **83** 042317
- [13] Shiau S-Y, Joynt R and Coppersmith S N 2005 *Quant. Int. Comp.* **5** 492–506
- [14] Levestin F and Maldonado C 2010 *CUBO A Math. J.* **12** 53
- [15] Curtin B and Daqqa I 2009 *Des. Codes Cryptogr.* **52** 263
- [16] Go J T 2002 *Eur. J. Comb.* **23** 399
- [17] Smith J 2012 Algebraic aspects of multi-particle quantum walks *PhD Thesis* University of Waterloo, Canada
- [18] Terwilliger P 1992 *J. Algebr. Comb.* **1** 363
- [19] Bannai E and Munemasa A 1995 *Kyushu J. Math.* **49** 93

Quantum Fourier transform in computational basis

S. S. Zhou^{1,2} · T. Loke¹ · J. A. Izaac¹ ·
J. B. Wang¹ 

Received: 4 September 2016 / Accepted: 3 January 2017 / Published online: 10 February 2017
© Springer Science+Business Media New York 2017

Abstract The quantum Fourier transform, with exponential speed-up compared to the classical fast Fourier transform, has played an important role in quantum computation as a vital part of many quantum algorithms (most prominently, Shor’s factoring algorithm). However, situations arise where it is not sufficient to encode the Fourier coefficients within the quantum amplitudes, for example in the implementation of control operations that depend on Fourier coefficients. In this paper, we detail a new quantum scheme to encode Fourier coefficients in the computational basis, with fidelity $1 - \delta$ and digit accuracy ϵ for each Fourier coefficient. Its time complexity depends polynomially on $\log(N)$, where N is the problem size, and linearly on $1/\delta$ and $1/\epsilon$. We also discuss an application of potential practical importance, namely the simulation of circulant Hamiltonians.

Keywords Quantum algorithm · Quantum Fourier transform · Computational basis state · Controlled quantum gates

1 Introduction

Since the milestone introduction of Shor’s quantum factoring algorithm [1] allows prime number factorization with complexity $\mathcal{O}(\text{polylog}N)$ —an exponential speed-up compared to the fastest known classical algorithms—there has been an increasing number of quantum algorithm discoveries harnessing the unique properties of quantum mechanics in order to achieve significant increases in computational efficiency. The use

✉ J. B. Wang
jingbo.wang@uwa.edu.au

¹ School of Physics, The University of Western Australia, Crawley, WA 6009, Australia

² Department of Physics, Yale University, New Haven, CT 06520, USA

of the quantum Fourier transform (QFT) [2] in Shor's factoring algorithm is integral to the resulting speed-up.

The fast Fourier transform (FFT), an efficient classical implementation of the discrete Fourier transform (DFT), is a hugely important algorithm, with classical uses including signal processing and frequency analysis [3]. Due to its ubiquity and efficiency (with scaling $\mathcal{O}(N \log N)$), it has been regarded to be one of the most important non-trivial classical algorithms [4].

The QFT [with complexity $\mathcal{O}((\log N)^2)$] algorithm is the natural extension of the DFT to the quantum regime, with exponential speed-up realized compared to the FFT ($\mathcal{O}(N \log N)$), due to superposition and quantum parallelism. The QFT is essentially identical to the FFT in that it performs a DFT on a list of complex numbers, but the result of the QFT is stored as amplitudes of a quantum state vector. In order to extract the individual Fourier components, measurements need to be performed on the quantum state vector. As such, the QFT is not directly useful for determining the Fourier-transformed coefficients of the original list of numbers. However, the QFT is widely used as a subroutine in larger algorithms, including but not limited to Shor's algorithm [1], quantum amplitude estimation [5] and quantum counting [6, 7].

Typically, there are two methods of encoding the result of a quantum algorithm: encoding within the computational basis of the quantum state [5] and encoding within the amplitudes of the quantum state [2]. The QFT fits the latter category and has been successfully used as a foundation for a plethora of other quantum algorithms—for example in the fields of quantum chemistry and simulations [8–10], signal and image processing [11, 12], cryptography [13] and computer science [4, 14]. However, situations arise where we need the Fourier coefficients in the computational basis, for example in order to efficiently implement circulant Hamiltonians with quantum circuits [15].

In this paper, we introduce a new quantum scheme for computing the Fourier transform and storing the results in the computational basis, namely quantum Fourier transform in the computational basis (QFTC). We begin in Sect. 2 by defining the notations and chosen conventions, before detailing the QFTC algorithm for computing the DFT in the computational basis in Sect. 3. This section also includes a thorough analytic derivation of the complexity and error analysis. One possible application of this algorithm, the implementation of circulant Hamiltonians, is then discussed in Sect. 5. In addition, we have provided supplementary material in the appendices, detailing the quantum arithmetic necessary for the QFTC algorithm in Appendix 1 and the implementation of circulant matrix operators in Appendix 2.

2 Definitions and notations

The DFT, applied to a unit vector $\mathbf{x} = (x_0 \ x_1 \ \cdots \ x_{N-1}) \in \mathbb{C}^N$, outputs a unit vector $\mathbf{y} = (y_0 \ y_1 \ \cdots \ y_{N-1})$, where

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j k / N} x_j, \quad k = 0, 1, \dots, N-1. \quad (1)$$

In the following sections, we assume that $N = 2^L$, where L is some integer, as in the conventional FFT and QFT algorithms. The QFT performs the discrete Fourier

transform in amplitudes:

$$\sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{k=0}^{N-1} y_k |k\rangle. \quad (2)$$

The QFTC, on the other hand, enables the Fourier-transformed coefficients to be encoded in the computational basis:

$$|k\rangle \xrightarrow{\text{QFTC}} |k\rangle |y_k\rangle \quad (3)$$

where $|y_k\rangle$ corresponds to the fixed-point binary representation of $y_k \in (-1, 1)$ using two's complement format. Without loss of generality, we will assume the y_k coefficients are real in the following sections. If this is not the case, we can always redefine the inputs as the following:

$$x'_j = \frac{x_j + x_{N-j}^*}{2} \quad (\text{where } x_N = x_0 \text{ and } x_j^* \text{ is the complex conjugate of } x_j) \quad (4)$$

for all j . Applying the DFT to x' then produces a purely real result, $y'_k = \text{Re}(y_k)$. The imaginary components $\text{Im}(y_k)$ can be derived analogously, by applying the DFT to

$$x'_j = \frac{x_j - x_{N-j}^*}{2}. \quad (5)$$

In the proposed QFTC algorithm, the input vector \mathbf{x} is provided by an oracle O_x such that

$$O_x |0\rangle = \sum_{j=0}^{N-1} x_j |j\rangle, \quad (6)$$

which can be efficiently implemented if \mathbf{x} is efficiently computable [16, 18] or by using the qRAM that takes complexity $\log N$ under certain conditions [17, 19–22]. The number of calls to O_x and O_x^\dagger will be included in the overall complexity of the QFTC algorithm. It is worth noting that this algorithm would not work if we do not know how the input vector \mathbf{x} is generated.

3 Quantum Fourier transform in the computational basis

The steps involved in the QFTC algorithm are detailed below (with Fig. 1 depicting the circuit for *Step 1–Step 5* and Fig. 3 for *Step 6–Step 10*). We use 14 registers in our algorithm labelled A, B₁, B₁', B₂, B₂', C, C', D, D', E, E', F, F' and G, among which Reg A stores the subscript k in the Fourier coefficients, Reg G stores the value of y_k , and others are all ancillas. There are $p_0 + 1$ qubits in Reg G (meaning accuracy $\epsilon = 2^{-p_0}$).

Step 0 Initialize all qubits, including ancillas, to $|0\rangle$.

Step 1 Prepare Reg A of L qubits into a superposition of its computational basis states. Here we take $|k\rangle$ as an example:

$$|0^L\rangle \rightarrow |k\rangle, \quad (7)$$

where k is represented in binary as $k_1k_2 \cdots k_L$ with L qubits. Note that subsequent steps can be trivially extended for arbitrary linear combinations, for example of the form $\sum_k u_k |k\rangle$.

Step 2 Prepare an ancillary qubit in Reg B_1 as:

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \quad (8)$$

Step 3 Apply O_x to Reg B_2 of L qubits controlled by Reg B_1 :

$$|0^L\rangle \frac{1}{\sqrt{2}}(|1\rangle + |0\rangle) \xrightarrow{O_x \otimes |1\rangle\langle 1| + \mathbb{I} \otimes |0\rangle\langle 0|} \frac{1}{\sqrt{2}} \left(\sum_{j=0}^{N-1} x_j |j\rangle |1\rangle + |0^L\rangle |0\rangle \right), \quad (9)$$

where j is represented in binary as $j_1j_2 \cdots j_L$ with L digits.

Step 4 Apply $H^{\otimes L}$ to Reg B_2 of L qubits controlled by Reg B_1 :

$$\begin{aligned} & \frac{1}{\sqrt{2}} \left(\sum_{j=0}^{N-1} x_j |j\rangle |1\rangle + |0^L\rangle |0\rangle \right) \xrightarrow{H^{\otimes L} \otimes |0\rangle\langle 0| + \mathbb{I} \otimes |1\rangle\langle 1|} \\ & \sum_{j=0}^{N-1} \frac{1}{\sqrt{2}} \left(x_j |j\rangle |1\rangle + \frac{1}{\sqrt{N}} |j\rangle |0\rangle \right). \end{aligned} \quad (10)$$

Step 5 Apply a controlled phase operator on Reg A, B_1 , B_2 (with details given in Fig. 1b):

$$\begin{aligned} & |k\rangle \sum_{j=0}^{N-1} \frac{1}{\sqrt{2}} \left(x_j |j\rangle |1\rangle + \frac{1}{\sqrt{N}} |j\rangle |0\rangle \right) \\ & \xrightarrow{(\sum_{j,k'} e^{2\pi i j k' / N} |k'\rangle\langle k'| \otimes |j\rangle\langle j|) \otimes |1\rangle\langle 1| + \mathbb{I} \otimes |0\rangle\langle 0|} |k\rangle |\phi_k\rangle, \end{aligned} \quad (11)$$

in which we define $|\phi_k\rangle := \frac{1}{\sqrt{2}}(x_j e^{2\pi i j k / N} |j\rangle |1\rangle + \frac{1}{\sqrt{N}} |j\rangle |0\rangle)$ for simplicity. The function of the controlled phase operator is to add a phase factor $e^{2\pi i j k / N}$ to the quantum state $|k\rangle |j\rangle |1\rangle$ for arbitrary k and j and leave it unchanged when the ancillary qubit is $|0\rangle$.

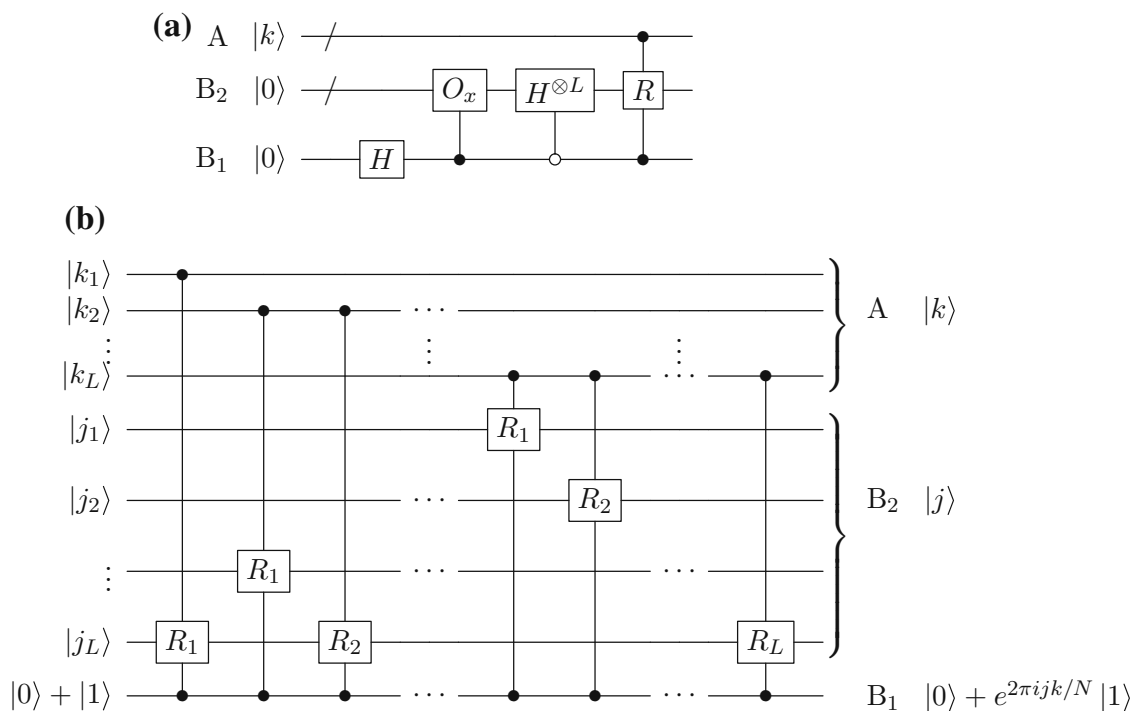


Fig. 1 a Quantum circuit for Step 1-Step 5; b Detailed quantum gates to implement the controlled phase operator in Step 5. Here $R_\ell = |0\rangle\langle 0| + e^{2\pi i/2^\ell} |1\rangle\langle 1|$

Using the Hadamard gate and the pauli-Z gate, we can prepare Reg C, C' in the quantum states $|\phi^\pm\rangle$:

$$\begin{aligned}
 &|0^{L+1}\rangle \xrightarrow{(+): H^{\otimes L} \otimes H; (-): H^{\otimes L} \otimes ZH} |\phi^\pm\rangle \\
 &= \frac{1}{\sqrt{2}} \left(\sum_{j=0}^{N-1} \frac{\pm 1}{\sqrt{N}} |j\rangle |1\rangle + \sum_{j=0}^{N-1} \frac{1}{\sqrt{N}} |j\rangle |0\rangle \right). \tag{12}
 \end{aligned}$$

We have

$$|\langle \phi^\pm | \phi_k \rangle|^2 = \frac{1}{4} (y_k^2 + 1) \pm \frac{y_k}{2}, \tag{13}$$

and

$$|\langle \phi^+ | \phi_k \rangle|^2 - |\langle \phi^- | \phi_k \rangle|^2 = y_k, \tag{14}$$

which leads to the following steps (as detailed in Fig. 3).

Step 6 Prepare $|\phi^+\rangle$ in Reg C and perform the swap test (Fig. 2) with $|\phi_k\rangle$ in Reg B (= B1 + B2). We get

$$|\psi_k^+\rangle = \frac{1}{2} |0\rangle (|\phi_k\rangle |\phi^+\rangle + |\phi^+\rangle |\phi_k\rangle) + \frac{1}{2} |1\rangle (|\phi_k\rangle |\phi^+\rangle - |\phi^+\rangle |\phi_k\rangle). \tag{15}$$

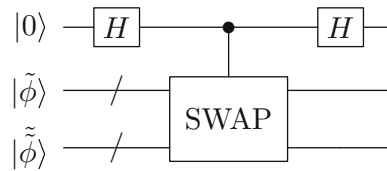


Fig. 2 Swap test. Here $\text{[SWAP]}|\tilde{\phi}\rangle|\tilde{\tilde{\phi}}\rangle = |\tilde{\tilde{\phi}}\rangle|\tilde{\phi}\rangle$. The probability to finally obtain $|0\rangle$ and $|1\rangle$ in the first register is $(1/2)(1 + |\langle\tilde{\phi}|\tilde{\tilde{\phi}}\rangle|^2)$ and $(1/2)(1 - |\langle\tilde{\phi}|\tilde{\tilde{\phi}}\rangle|^2)$, respectively. This procedure is often utilized to estimate the inner product of two quantum states $|\tilde{\phi}\rangle$ and $|\tilde{\tilde{\phi}}\rangle$ [23]

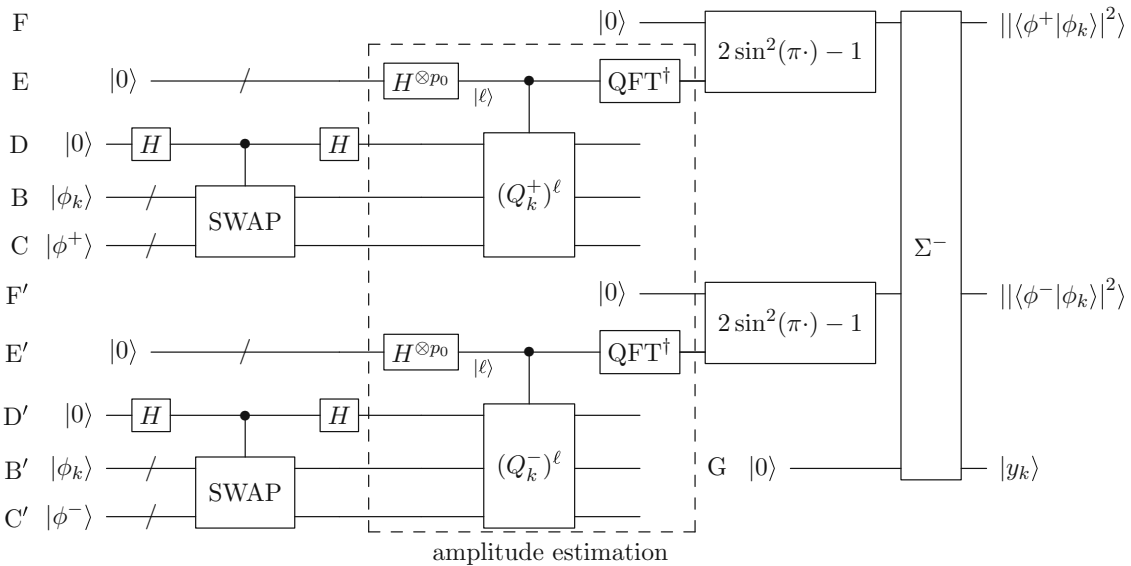


Fig. 3 Quantum circuit for Step 6–Step 10. The Σ^- gate transforms $|\alpha\rangle|\beta\rangle|0\rangle$ into $|\alpha\rangle|\beta\rangle|\alpha - \beta\rangle$ (see Appendix 1)

Step 7 Run amplitude estimation for all k on state $|\psi_k^+\rangle$ and store the phases in Reg E:

$$|\psi_k^+\rangle \rightarrow \left|\frac{\theta_k}{\pi}\right\rangle|\psi_k^\uparrow\rangle + \left|1 - \frac{\theta_k}{\pi}\right\rangle|\psi_k^\downarrow\rangle, \tag{16}$$

where $|\psi_k^+\rangle$ can be decomposed into the sum of $|\psi_k^\uparrow\rangle$ and $|\psi_k^\downarrow\rangle$ which are a pair of un-normalized orthogonal bases (corresponding to two distinct phases in the amplitude estimation procedure detailed below).

Step 8 Compute $|\langle\phi^+|\phi_k\rangle|^2 = (y_k^2 + 1)/4 + y_k/2$ using the quantum multiply–adder and sine gate (see Appendix 1 for details), for all values of k :

$$\left|\frac{\theta_k}{\pi}\right\rangle|\psi_k^\uparrow\rangle + \left|1 - \frac{\theta_k}{\pi}\right\rangle|\psi_k^\downarrow\rangle \rightarrow |\langle\phi^+|\phi_k\rangle|^2 \left(\left|\frac{\theta_k}{\pi}\right\rangle|\psi_k^\uparrow\rangle + \left|1 - \frac{\theta_k}{\pi}\right\rangle|\psi_k^\downarrow\rangle\right), \tag{17}$$

where the value of $|\langle\phi^+|\phi_k\rangle|^2 = 2 \sin^2 \theta_k - 1$ is stored in Reg F.

In the above description of *Step 6–Step 10*,

$$|\psi_k^+\rangle = \sin \theta_k |\psi_k^0\rangle + \cos \theta_k |\psi_k^1\rangle \tag{18}$$

where $|\psi_k^0\rangle$ corresponds to the part of $|\psi_k^+\rangle$ whose first qubit is $|0\rangle$, $|\psi_k^1\rangle$ corresponds to the part of $|\psi_k^+\rangle$ whose first qubit is $|1\rangle$. We can choose $\theta_k \in [0, \pi/2]$ without loss of generality. It can be easily calculated from Eq. 15 that $\sin^2 \theta_k = (1 + |\langle \phi^+ | \phi_k \rangle|^2)/2$. We define $Q_k^+ := -\mathcal{A}_k^+ S_0 (\mathcal{A}_k^+)^{\dagger} S_{\chi}$, where \mathcal{A}_k^+ is the unitary operator performing $|0\rangle_{\text{DBC}} \xrightarrow{\mathcal{A}_k^+} |\psi_k^+\rangle$, $S_0 = \mathbb{I} - 2 |0\rangle_{\text{DBC}} \langle 0|_{\text{DBC}}$ and $S_{\chi} = \mathbb{I} - 2 |0\rangle_{\text{D}} \langle 0|_{\text{D}}$ (subscripts denote labels of registers). According to the amplitude estimation algorithm [7],

$$(Q_k^+)^{\ell} |\psi_k^+\rangle = \sin(2\ell + 1)\theta_k |\psi_k^0\rangle + \cos(2\ell + 1)\theta_k |\psi_k^1\rangle. \tag{19}$$

For any $\ell \in \mathbb{N}$, Q_k^+ acts as a rotation in two-dimensional space $\text{Span}\{|\psi_k^0\rangle, |\psi_k^1\rangle\}$, and it has eigenvalues $e^{\pm i2\theta_k}$ with eigenstates $|\psi_k^{\uparrow, \downarrow}\rangle$ (un-normalized). Therefore, we can generate the state

$$|\psi_k^+\rangle = |\psi_k^{\uparrow}\rangle + |\psi_k^{\downarrow}\rangle \xrightarrow{\text{phase estimation}} \left| \frac{\theta_k}{\pi} \right\rangle |\psi_k^{\uparrow}\rangle + \left| 1 - \frac{\theta_k}{\pi} \right\rangle |\psi_k^{\downarrow}\rangle, \tag{20}$$

by running amplitude estimation of \mathcal{A}_k^+ on $|\psi_k^+\rangle$ and obtain $|\langle \phi^+ | \phi_k \rangle|^2 = |2 \sin^2 \theta_k - 1\rangle$ using the quantum multiply–adder and sine gate (see Appendix 1). The quantum circuit of amplitude estimation procedure is shown in Fig. 3.

Step 9 Repeat *Step 2–Step 8* in Reg B', C', E', F', with $|\phi^+\rangle$ and \mathcal{A}_k^+ replaced by $|\phi^-\rangle$ and \mathcal{A}_k^- , we obtain

$$|\langle \phi^+ | \phi_k \rangle|^2 |0\rangle \rightarrow |\langle \phi^+ | \phi_k \rangle|^2 |\langle \phi^- | \phi_k \rangle|^2 \tag{21}$$

in Reg F, F', where the quantum states in Reg A, B, B', C, C', E, E' are not written out explicitly for simplicity, because they remain unchanged in the following steps.

Step 10 Calculate $|\langle \phi^+ | \phi_k \rangle|^2$ minus $|\langle \phi^- | \phi_k \rangle|^2$ and encode the result in Reg G, using the quantum adder described in Appendix 1:

$$|\langle \phi^+ | \phi_k \rangle|^2 |\langle \phi^- | \phi_k \rangle|^2 |0\rangle \rightarrow |\langle \phi^+ | \phi_k \rangle|^2 |\langle \phi^- | \phi_k \rangle|^2 |y_k\rangle. \tag{22}$$

Step 11 Uncompute the ancillas using the inverse algorithm of *Step 2–Step 9*:

$$|k\rangle |\Psi_k^{\text{ancilla}}\rangle |y_k\rangle \rightarrow |k\rangle |0\rangle |y_k\rangle. \tag{23}$$

4 Complexity analysis

Theorem 1 (QFTC) *Given an input $\sum_k u_k |k\rangle$, the required quantum state $\sum_k u_k |k\rangle |y_k\rangle$ can be prepared to digit accuracy ϵ^1 with fidelity $1 - \delta^2$ using $\mathcal{O}((\log N)^2/(\delta\epsilon))$ one- or two-qubit gates, and $\mathcal{O}(1/(\delta\epsilon))$ calls of controlled- O_x and its inverse.*

Proof First, we consider the complexity involved in \mathcal{A}_k^+ (described in Step 2–Step 6). It contains Hadamard gates, controlled phase operators and swap gates which can be constructed using $\mathcal{O}((\log N)^2)$ one- or two-qubit gates and only one call of controlled- O_x .

The subsequent amplitude estimation block needs $\mathcal{O}(1/(\delta\epsilon))$ applications of $Q_k^+ = -\mathcal{A}_k^+ S_0 (\mathcal{A}_k^+)^{\dagger} S_x$ to obtain accuracy ϵ with fidelity at least $1 - \delta$ [7, 24]. We then use the quantum multiply–adder and sine gate to obtain the value of $|\langle \phi^+ | \phi_k \rangle|^2 = \frac{1}{4}(1 + y_k^2) + y_k/2$ for different $|k\rangle$'s in the computational basis. Using a similar procedure to obtain $|\langle \phi^- | \phi_k \rangle|^2$, we obtain $y_k = |\langle \phi^+ | \phi_k \rangle|^2 - |\langle \phi^- | \phi_k \rangle|^2$ finally. Since the derivative of $\sin x$ is always smaller than one, we set $\epsilon = \Theta(\epsilon)$ in order to guarantee accuracy ϵ in y_k . As detailed in Appendix 1, the quantum multiply–adders and sine gates have complexity $\mathcal{O}(\text{polylog}(1/\epsilon))$ which is smaller than $\mathcal{O}(1/\epsilon)$ in amplitude estimation. Therefore, the complexity of these gates can be omitted.

The total complexity of the proposed circuit will be $\mathcal{O}((\log N)^2/(\delta\epsilon))$ one- or two-qubit gates, and $\mathcal{O}(1/(\delta\epsilon))$ calls of controlled- O_x and its inverse. \square

Throughout the proposed QFTC algorithm, $|k\rangle$ in Reg A is used to control the application of quantum operators acting on other registers, giving us the advantage of parallel calculating y_k for all k . Though values of y_k 's cannot be obtained by a single measurement of $\sum_k |k\rangle |y_k\rangle$, they can be used in subsequent quantum computation once they are encoded in the computational basis.

The disadvantage of the QFTC algorithm to provide the value of $|y_k\rangle$ (as discussed in Sect. 5) compared to the corresponding classical algorithm lies in its accuracy. In the FFT, $\|\tilde{\mathbf{y}} - \mathbf{y}\| < \Theta(\log N) \times \epsilon$ [25, 26]; in the QFTC, however, $\|\tilde{\mathbf{y}} - \mathbf{y}\| < \Theta(\sqrt{N}) \times \epsilon$. Precision at this level would be sufficient for example in Fourier transform spectroscopy when only a small set of frequencies dominate the behaviour of the vectors [27]. However, when high precision is needed, in order to achieve similar precision $\|\tilde{\mathbf{y}} - \mathbf{y}\| < \epsilon$ like the FFT, we will need \sqrt{N} times the complexity in Theorem 1. Then we only have a quadratic, not exponential, speed-up in this case compared to the classical algorithm.

5 Application

One important family of operators is the circulant matrices which have found important applications in, for example, quantum walks on Moöbius strips [28], investigation on

¹ $|y_k - \tilde{y}_k| < \epsilon$, where \tilde{y}_k is the truncated value of y_k with accuracy $\epsilon = 2^{-p_0}$.

² $\left| \langle \Psi^{\text{final}} | \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle |\tilde{y}_k\rangle \right) \right| \geq 1 - \delta$, where $|\Psi^{\text{final}}\rangle$ is the state obtained through the QFTC algorithm.

quantum supremacy [15], biochemical modelling [29], vibration analysis [30] and parallel diagnostic algorithm for super-computing [31].

Circulant matrices are defined as follows [32]:

$$C = \begin{pmatrix} c_0 & c_1 & \cdots & c_{N-1} \\ c_{N-1} & c_0 & \cdots & c_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & \cdots & c_0 \end{pmatrix}, \tag{24}$$

using an N -dimensional vector $\mathbf{c} = (c_0 \ c_1 \ \cdots \ c_{N-1})$. Such matrices are diagonalizable by the discrete Fourier transform (DFT), i.e.

$$C = F \Lambda F^\dagger, \tag{25}$$

where F is the Fourier matrix with $F_{kj} = e^{2\pi ijk/N} / \sqrt{N}$, and Λ is a diagonal matrix of eigenvalues given by $\Lambda_k = \sqrt{N} (F(c_0 \ c_1 \ \cdots \ c_{N-1})^\dagger)_k \equiv \sqrt{N} F_k$. Note that the condition that C is Hermitian (in order to be a Hamiltonian) is equivalent to our assumption in Sect. 2 that the Fourier coefficients F_k are real. Since the eigenvalues of a circulant matrix are Fourier transform of its parameters, we are able to implement circulant quantum operators (non-unitary in general) using the conventional QFT through the manipulation of amplitudes, as detailed in Appendix 2.

This approach cannot be used directly for simulation of (non-sparse) circulant Hamiltonians, where we need to implement e^{-iCt} instead of C . Simulation of circulant Hamiltonians is equivalent to the implementation of continuous-time quantum walks on a weighted circulant graph [33]. Circulant matrices are adjacency matrices of circulant graphs, and c_j characterizes the probability for the walker to transfer from vertex ℓ to vertex $\ell - j$.

In order to simulate e^{-iCt} , we decompose it into $F e^{-i\Lambda t} F^\dagger$, where $e^{-i\Lambda t}$ can be simulated with the aid of the quantum circuit given in simulating diagonal Hamiltonians [34]. If the Fourier coefficients Λ_k are encoded in the computational basis, as performed by the QFTC algorithm, they can then be used to control the phase factor $e^{-i\Lambda_k t}$ added to different eigenstates of the circulant matrix, for the purpose of implementing the diagonal Hamiltonian $e^{-i\Lambda t}$.

In the following, we will demonstrate how the QFTC algorithm can be used to simulate Hamiltonians with a circulant matrix structure, as shown in Fig. 4:

Step 1 Perform the inverse QFT on $|s\rangle$:

$$|s\rangle = \sum_{k=0}^{N-1} s_k |k\rangle \rightarrow \sum_{k=0}^{N-1} \mathfrak{s}_k |k\rangle. \tag{26}$$

Step 2 Apply the QFTC algorithm (*Step 2–Step 11* in Sect. 3) for \mathbf{c} :

$$\sum_{k=0}^{N-1} \mathfrak{s}_k |k\rangle \rightarrow \sum_{k=0}^{N-1} \mathfrak{s}_k |k\rangle |F_k\rangle. \tag{27}$$

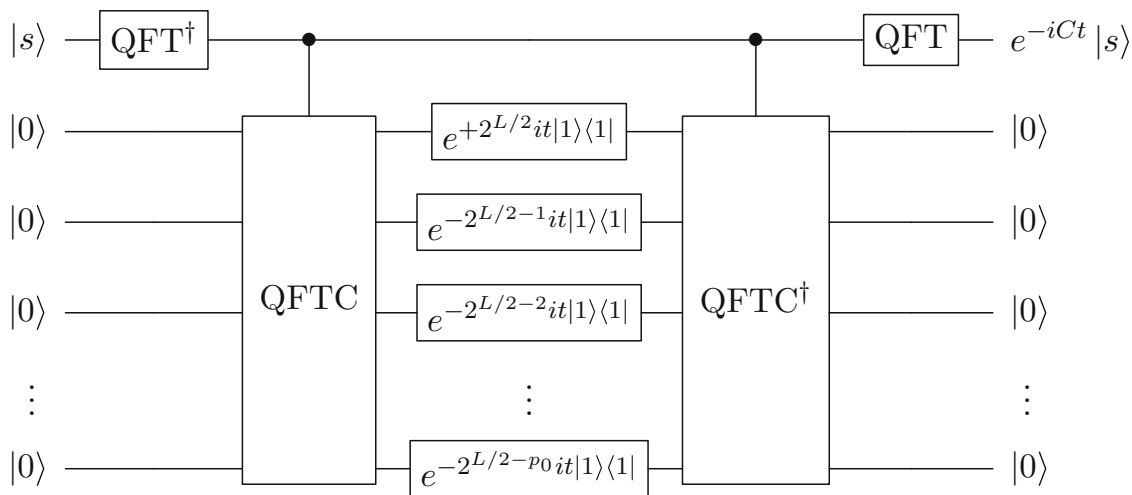


Fig. 4 Simulation of circulant Hamiltonians. $p_0 + 1$ is the number of digits of the resulting Fourier coefficients, and F_k was encoded in the form $f_0.f_1f_2 \cdots f_{p_0}$ as the complementary code for a number between -1 and 1 . Here we define $\text{QFTC} |k\rangle |0\rangle = |k\rangle |F_k\rangle$ (detailed in *Step 2–Step 11* in Sect. 3)

Step 3 Do controlled phase gate $e^{+2^{L/2}it|1\rangle\langle 1|}$ on the first digit (qubit) of $|F_k\rangle$ and $e^{-2^{L/2-p+1}it|1\rangle\langle 1|}$ on the p th digit (qubit) of $|F_k\rangle$ for all $p > 1$:

$$\sum_{k=0}^{N-1} \mathfrak{s}_k |k\rangle |F_k\rangle \rightarrow \sum_{k=0}^{N-1} \mathfrak{s}_k e^{-i\Lambda_k t} |k\rangle |F_k\rangle. \tag{28}$$

Step 4 Undo the QFTC for every $|k\rangle$:

$$\sum_{k=0}^{N-1} \mathfrak{s}_k e^{-i\Lambda_k t} |k\rangle |F_k\rangle \rightarrow \sum_{k=0}^{N-1} \mathfrak{s}_k e^{-i\Lambda_k t} |k\rangle. \tag{29}$$

Step 5 Perform the QFT:

$$\sum_{k=0}^{N-1} \mathfrak{s}_k e^{-i\Lambda_k t} |k\rangle \rightarrow e^{-iCt} |s\rangle. \tag{30}$$

Theorem 2 (Simulation of Circulant Hamiltonians) *The simulation of a circulant Hamiltonian e^{-iCt} can be performed within error δ using $\mathcal{O}(\sqrt{N}t(\log N)^2/\delta^{3/2})$ one- or two-qubit gates, as well as $\mathcal{O}(\sqrt{N}t/\delta^{3/2})$ calls of controlled- O_x and its inverse, where $\mathbf{x} = \mathbf{c}$ is a unit vector in \mathbb{C}^N and C is Hermitian.³*

Proof The error present in the Hamiltonian simulation is fully determined by the precision of the QFTC algorithm. According to the above QFTC complexity analysis, we need $\mathcal{O}((\log N)^2/(\delta\varepsilon))$ one- or two-qubit gates, as well as $\mathcal{O}(1/(\delta\varepsilon))$ calls of

³ $\|e^{-iCt} - \widetilde{e^{-iCt}}\| \leq \delta$, where $\widetilde{e^{-iCt}}$ represents the operator that is actually performed by this algorithm.

controlled- O_x and its inverse, to achieve accuracy ε in F_k . The fidelity achieved for the Hamiltonian simulation, as defined by the squared modulus of inner product, is

$$(1-\delta)^2 \left| \langle e^{-i\tilde{C}t} |s\rangle, e^{-iCt} |s\rangle \rangle \right| = (1-\delta)^2 \left| \sum_{k=0}^{N-1} e^{i(\tilde{\Lambda}_k - \Lambda_k)t} |s_k|^2 \right| > 1 - \mathcal{O}((\sqrt{N}t\varepsilon)^2 + \delta), \quad (31)$$

where the last inequality is derived using

$$\left| e^{i\gamma_1} + |\Gamma| e^{i\gamma_2} \right| = (1 + |\Gamma|^2 + 2|\Gamma| \cos(\gamma_1 - \gamma_2))^{1/2} > (1 + |\Gamma|) \left| \cos \frac{\gamma_1 - \gamma_2}{2} \right|, \quad (32)$$

and $\tilde{\Lambda}_k$ are the estimated (truncated) eigenvalues calculated via the QFTC algorithm. For a fixed δ in the QFTC algorithm, if we choose $\varepsilon = \sqrt{\delta}/(\sqrt{N}t)$, the fidelity will be $1 - \mathcal{O}(\delta)$. We then need $\mathcal{O}((\log N)^2/(\delta\varepsilon)) = \mathcal{O}(\sqrt{N}t(\log N)^2/\delta^{3/2})$ one- or two-qubit gates, as well as $\mathcal{O}(\sqrt{N}t/\delta^{3/2})$ calls of controlled- O_x and its inverse. \square

The complexity in simulation of circulant Hamiltonians would depend linearly on the value of $\sqrt{|c_0|^2 + \dots + |c_{N-1}|^2}$ which was assumed to be 1. This value is always smaller (and normally much smaller) than the spectral norm of the circulant matrix C , which is often used to characterize the complexity in the simulation of dense Hamiltonians [35].

6 Conclusion

In this paper, we proposed a new QFTC algorithm, an efficient quantum scheme to encode the results of the discrete Fourier transform in the computational basis. This algorithm allows us to overcome a main shortcoming of the conventional quantum Fourier transform—the inability to perform operations controlled by the Fourier coefficients. In short, the QFTC utilizes swap tests to obtain a function of the Fourier coefficients in the amplitudes, with individual coefficients then extracted via amplitude estimation and quantum arithmetic.

Secondly, a detailed complexity analysis of the QFTC algorithm was performed, finding it requires $\mathcal{O}((\log N)^2/(\delta\varepsilon))$ calls of one- or two-qubit gates, as well as $\mathcal{O}(1/(\delta\varepsilon))$ calls of controlled- O_x and its inverse, in order to achieve fidelity $1 - \delta$ and precision ε . Note that the overall complexity depends polylogarithmically on N , similarly to the conventional QFT, and we require only controlled phase gates and Hadamard gates. The inverse proportionality with the desired accuracy, ε , occurs due to the application of amplitude estimation within the algorithm.

Finally, we detailed an application of the QFTC algorithm in the simulation of circulant Hamiltonians, which requires $\mathcal{O}(\sqrt{N}t(\log N)^2/\delta^{3/2})$ one- or two-qubit gates, as well as $\mathcal{O}(\sqrt{N}t/\delta^{3/2})$ calls of controlled- O_x and its inverse to achieve fidelity $1 - \delta$. This paves the way for a quantum circuit implementation of continuous-time quantum walks on circulant graphs, with potential applications in a wide array of disciplines. Further applications of the QFTC algorithm are expected.

Acknowledgements The authors would like to thank Ashley Montanaro for constructive comments and Jeremy O'Brien, Jonathan Matthews, Xiaogang Qiang, Lyle Noakes, Chuheng Zhang and Hanwen Zha for helpful discussions.

Appendix 1: Quantum arithmetic

Addition and multiplication are basic elements of arithmetic in classical computer. There have been several proposals on how to build quantum adders and multipliers [36–39], constructed predominately using CNOT gates and Toffoli gates. Draper's addition quantum circuits, however, utilize the quantum Fourier transformation (QFT) [40]. QFT-based multiplication and related quantum arithmetic have also been proposed [41–44]. In this appendix, for completeness, we outline the construction of the quantum arithmetic gates required for the QFTC algorithm in detail.

We show here, using QFT-based circuits and fixed-point number representation, all elementary quantum arithmetic gates used to construct the QFTC circuit (including adders, multipliers and cosine gates) have $\mathcal{O}(\text{poly}(n))$ complexity, where n is the number of qubits (number of digits) representing the number. With accuracy ϵ , this results in $\mathcal{O}(\text{polylog}(1/\epsilon))$ complexity.

QFT multiply-adder

We begin by describing a quantum multiply-adder for real inputs a and b between 0 and 1. Let $|a\rangle = |a_1\rangle |a_2\rangle \cdots |a_m\rangle$ represent the fixed-point number $a = 0.a_1a_2 \cdots a_m$ (same for b). Using this representation, the quantum multiply-adder (QMA), as shown in Fig. 5a, can realize the following transformation,

$$\Pi_{m,n}^{\pm} |a\rangle |b\rangle |c\rangle = |a\rangle |b\rangle |c \pm a \times b\rangle, \quad (33)$$

where m and n denote the number of digits of a and b , respectively.

In quantum multiply-adders, the outputs, unlike the inputs, can be negative and we use the complemental code $c^{(C)} = c_0.c_1c_2 \cdots c_{m+n} \in [0, 2)$ to represent the output $c \in (-1, 1)$ and $c = c^{(C)}$ if c is non-negative and $c = c^{(C)} - 2$ if c is negative. $|c\rangle$ is composed of $|c_0\rangle |c_1\rangle \cdots |c_{m+n}\rangle$. Note that this quantum multiply-adder also applies to any fixed-point-represented numbers by cleverly choosing the appropriate positions of the fractional points.

The quantum multiply-adder can be decomposed into the following form, as shown in Fig. 5b:

$$\Pi_{m,n}^{\pm} = (\mathbb{I} \otimes \mathbb{I} \otimes \text{QFT}^{\dagger}) \times \pi_{m,n}^{\pm} \times (\mathbb{I} \otimes \mathbb{I} \otimes \text{QFT}), \quad (34)$$

where $\pi_{m,n}^{\pm}$ represents an intermediate quantum multiply-adder,

$$\pi_{m,n}^{\pm} |a\rangle |b\rangle |\phi(c)\rangle = |a\rangle |b\rangle |\phi(c \pm a \times b)\rangle \quad (35)$$

with $|\phi(c)\rangle := \text{QFT} |c\rangle$ and $|\phi_k(c)\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi ic \times 2^{m+n-k}} |1\rangle)$, $k = 1, 2, \dots, m+n+1$.

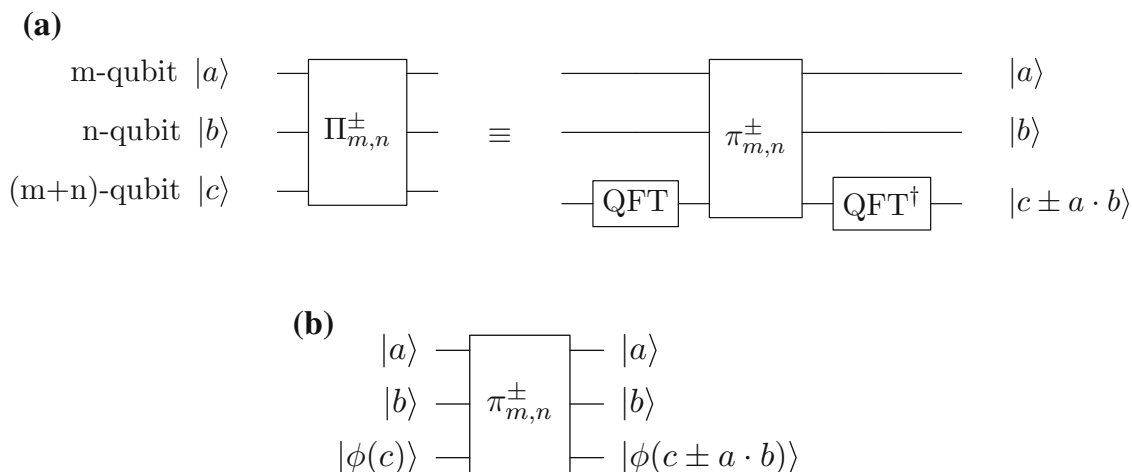


Fig. 5 Quantum circuit of the multiply–adder, **a** quantum multiply–adder, **b** intermediate multiply–adder

Figure 6 shows a detailed quantum circuit construction of $\pi_{m,n}^\pm$, using the QFT adders $2^{-l}\Sigma_{m,n}^\pm$, which act as follows:

$$\boxed{2^{-l}\Sigma_{m,n}^\pm} |b\rangle |\phi(c)\rangle = |b\rangle |\phi(c \pm 2^{-l}b)\rangle. \tag{36}$$

The QFT adders are constructed via controlled phase operations, as shown in Fig. 6c.

After applying the QFT adder $2^{-m}\Sigma_{m,n}^\pm$ (controlled by $|a_m\rangle$) in Fig. 6, we obtain

$$|\phi(c)\rangle \longrightarrow |\phi(c \pm a_m 2^{-m}b)\rangle. \tag{37}$$

Proceeding in a similar fashion, it can be seen that the final output state of the intermediate multiply–adder is

$$\left| \phi(c + a_m 2^{-m}b + \dots + a_1 2^{-1}b) \right\rangle = |\phi(c \pm a \times b)\rangle. \tag{38}$$

To illustrate how the circuit works, take for example the evolution of $\phi_{m+n-l}(c)$ after R_1^\pm, \dots, R_n^\pm :

$$|0\rangle + e^{2\pi ic \times 2^l} |1\rangle \longrightarrow |0\rangle + e^{2\pi ic \times 2^l \pm b} |1\rangle. \tag{39}$$

We then have

$$|\phi_k(c)\rangle \rightarrow |\phi_k(c \pm 2^{-l}b)\rangle.$$

It is clear from Fig. 6c that the QFT adder uses $\mathcal{O}((m+n)n)$ one- or two-qubit gates. Hence, the total complexity of the intermediate QFT multiply–adders is $\mathcal{O}((m+n)mn)$. Thus, with QFT scaling $\mathcal{O}((m+n)^2)$, the total complexity of the quantum multiply–adder $\Pi_{m,n}^\pm$ is $\max\{\mathcal{O}(mn^2), \mathcal{O}(nm^2)\}$.

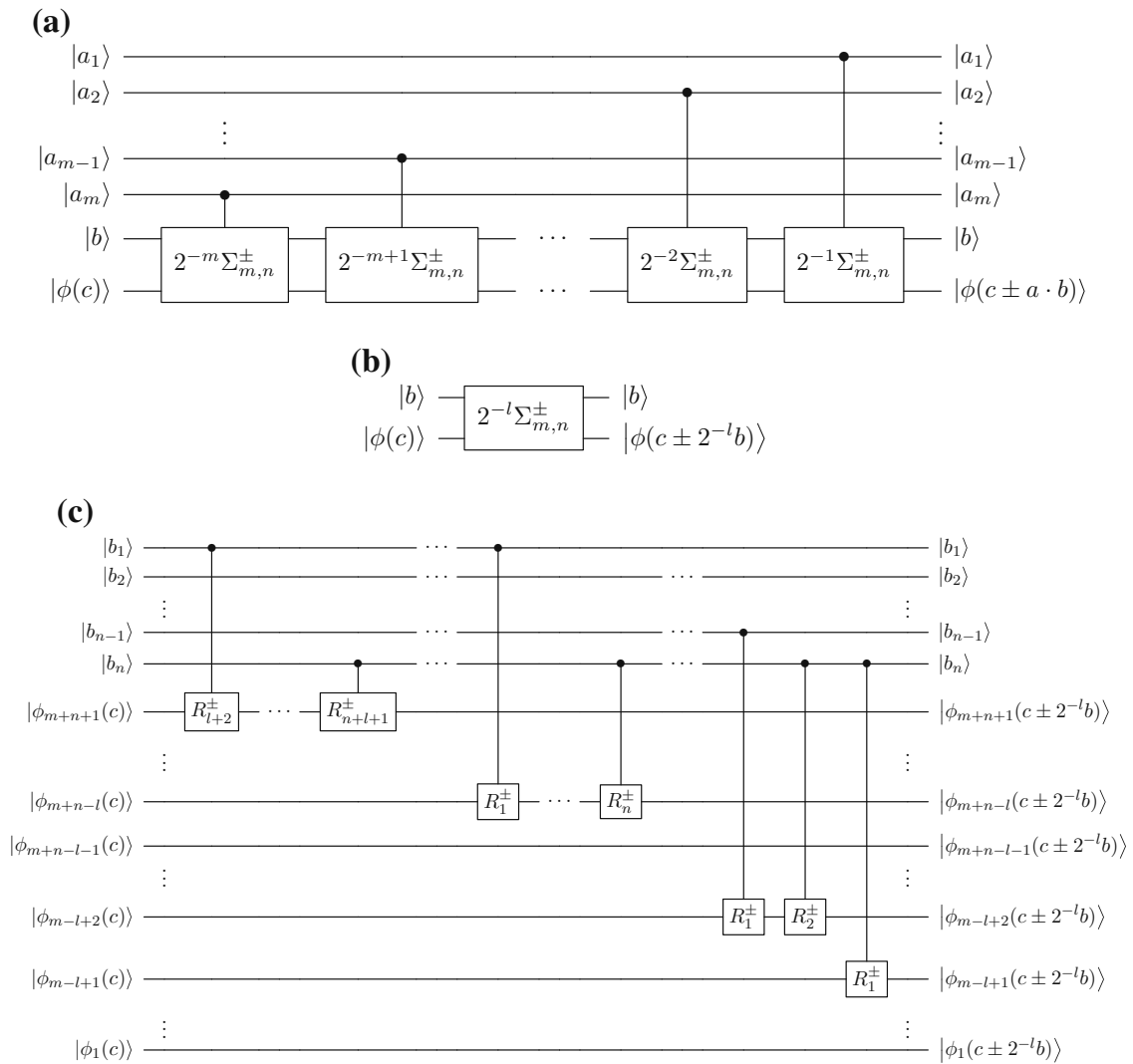


Fig. 6 Quantum circuit of $\pi_{m,n}^\pm$, (a) $\pi_{m,n}^\pm$ gate (b), QFT adder, (c) detailed quantum circuit construction of the QFT adder $2^{-l} \Sigma_{m,n}^\pm$, $R_k^\pm = |0\rangle\langle 0| + e^{\pm 2\pi i/2^k} |1\rangle\langle 1|$

Note that if we choose $l = 0$ in $2^{-l} \Sigma_{m,n}^\pm$ and perform a QFT and an inverse QFT before and after the application of the QFT adder in Eq. 36, we have a quantum adder

$$|b\rangle |c\rangle \rightarrow |b\rangle |c \pm b\rangle. \tag{40}$$

We can also add (or subtract) two numbers without having to destroy their original values encoded in the computational basis, i.e.

$$|b\rangle |c\rangle |0\rangle \rightarrow |b\rangle |c\rangle |b\rangle \rightarrow |b\rangle |c\rangle |b \pm c\rangle \tag{41}$$

by using Eq. 40 twice.

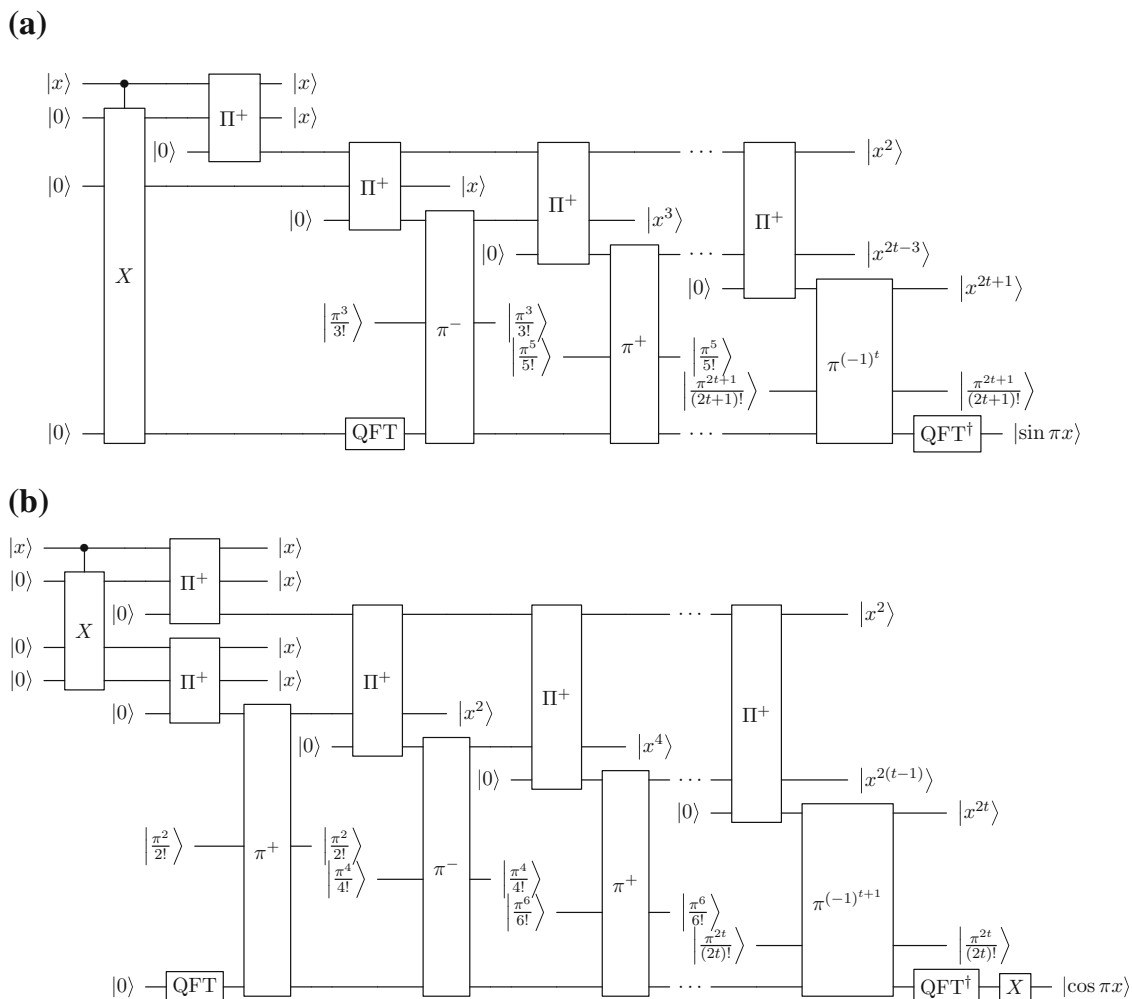


Fig. 7 Quantum circuits of the sine and cosine gates ($|0\rangle$ represents a number of qubits in above circuits where the numbers are omitted). Pauli-X gates are used to transform $|0\rangle$ into $|x\rangle$, and the subscript for all the quantum multiply-adders in above circuits is (p', p') , **a** sine gate, **b** cosine gate

Quantum sine and cosine gate

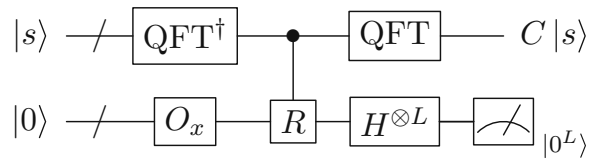
By implementing the Taylor series using the quantum multiply-adder, we are able to build a quantum sine (and cosine) gate. Suppose $x = 0.x_1x_2 \dots x_n$ and $x \in [0, 1)$. We aim to build a sine gate calculating the value of $\sin \pi x$, performing $|x\rangle |0^n\rangle |0^m\rangle \rightarrow |x\rangle |\sin \pi x\rangle |\Psi^{ancilla}\rangle$.

We now consider the error in the truncated Taylor series. First, the error introduced by imprecision in the n -digit representation of x is $\mathcal{O}(2^{-n})$, since the derivative of $\sin \pi x$ is bounded. The Taylor series of $\sin \pi x$ at around $x = 0$ is

$$\sin \pi x = \pi x - \frac{(\pi x)^3}{3!} + \frac{(\pi x)^5}{5!} - \dots + (-1)^t \frac{(\pi x)^{2t+1}}{(2t+1)!} + \frac{(-1)^{t+1} \cos \pi z}{(2t+3)!} (\pi x)^{(2t+3)}. \tag{42}$$

The remainder term for the k th term in the expansion is $\frac{f^{(k+1)}(z)}{(k+1)!} x^{k+1}$, where $z \in (0, x)$, according to Taylor's Theorem [45]. As a result, in Eq. (42), the remainder term (error) is $\frac{(-1)^{t+1} \cos \pi z}{(2t+3)!} (\pi x)^{(2t+3)}$ and is obviously bounded by $\mathcal{O}(2^{-n})$ for $t = n$.

Fig. 8 Implementation of circulant matrices. Here $R |k\rangle |j\rangle = e^{2\pi i k j / N} |k\rangle |j\rangle$



In the sine gate, the $t + 1$ terms $\left\{ \pi x, \frac{(\pi x)^3}{3!}, \dots, (-1)^t \frac{(\pi x)^{2t+1}}{(2t+1)!} \right\}$ are first calculated and then added (or subtracted) together. Suppose each of the $t + 1$ terms has an error within 2^{-p} . Taking $p = n + \lceil \log n \rceil = \mathcal{O}(n)$, the error introduced by adding and subtracting will be $\mathcal{O}(t \times 2^{-p}) = \mathcal{O}(2^{-n})$. Suppose all multiply-adders have p' digits inputs. When errors in y_1, y_2 are within $2^{-(\ell+1)}$ and $y_1, y_2 \leq 1 - 2^{-(\ell+1)}$, $(y_1 + 2^{-(\ell+1)})(y_2 + 2^{-(\ell+1)}) = y_1 y_2 + 2^{-\ell}(y_1 + y_2)/2 + 2^{-2\ell-2} \leq y_1 y_2 + 2^{-\ell}$. It means that by applying the multiply-adders $2t$ times, the error will be 2^{2t} times larger. Thus, we can choose a $p' = \mathcal{O}(p + 2t) = \mathcal{O}(n)$ which guarantees accuracy 2^{-p} in all the powers of x and also all the $t + 1$ terms in the Taylor series.

We conclude that we can choose $t = \mathcal{O}(n)$ and $p' = \mathcal{O}(n)$ so that the total accuracy of the sine gate is bounded by 2^{-n} . Figure 7 shows the quantum circuit for the sine and cosine gate. The complexity of the quantum sine gate can be calculated based on the scaling of quantum multiply-adders which equals to $\mathcal{O}(p'^3)$. The total complexity of the quantum sine gate is $\mathcal{O}(tp'^3) = \mathcal{O}(n^4)$ for accuracy 2^{-n} . To put it in another way, $\mathcal{O}(\text{polylog}(1/\epsilon))$ one- or two-qubit gates are required to achieve accuracy ϵ .

Appendix 2: Implementing circulant operators

Consider an arbitrary state $|s\rangle$. We wish to obtain $C|s\rangle$, where C is an arbitrary circulant matrix. Below, we present a possible algorithm for implementing a circulant matrix quantum operator (see Fig. 8).

Step 1 Perform the inverse QFT on $|s\rangle$:

$$\sum_{k=0}^{N-1} s_k |k\rangle \rightarrow \sum_{k=0}^{N-1} \mathfrak{s}_k |k\rangle. \tag{43}$$

Step 2 Add another register prepared to $\sum_{j=0}^{N-1} c_j |j\rangle$ using O_x ($x = \mathbf{c}$ in Eq. 6):

$$\sum_{k=0}^{N-1} \mathfrak{s}_k |k\rangle \rightarrow \sum_{j,k=0}^{N-1} \mathfrak{s}_k c_j |k\rangle |j\rangle. \tag{44}$$

Step 3 Apply the controlled phase gate so that $|k\rangle |j\rangle \rightarrow e^{2\pi i k j / N} |k\rangle |j\rangle$:

$$\sum_{j,k=0}^{N-1} \mathfrak{s}_k c_j |k\rangle |j\rangle \rightarrow \sum_{j,k=0}^{N-1} \mathfrak{s}_k c_j e^{2\pi i j k / N} |k\rangle |j\rangle. \tag{45}$$

Step 4 Apply Hadamard gates to $|j\rangle$:

$$\sum_{j,k=0}^{N-1} \mathfrak{s}_k c_j e^{2\pi i j k / N} |k\rangle |j\rangle \rightarrow \sum_{j,k=0}^{N-1} \mathfrak{s}_k |k\rangle \left(F_k |0^L\rangle + \sqrt{1 - F_k^2} |0^\perp\rangle \right), \quad (46)$$

where $|0^\perp\rangle$ represents any states perpendicular to $|0^L\rangle$.

Step 5 By post-selecting the ancillary qubit state $|0^L\rangle$, the quantum state in the first register collapses to

$$\frac{1}{\sqrt{\sum_k |F_k \mathfrak{s}_k|^2}} \sum_{k=0}^{N-1} F_k \mathfrak{s}_k |k\rangle. \quad (47)$$

Step 6 Perform the QFT:

$$\text{QFT} \sum_{k=0}^{N-1} \mathfrak{s}_k F_k |k\rangle \propto C |s\rangle. \quad (48)$$

Note that the post-selection probability of obtaining the correct state in *Step 5* is

$$p = \sum_{k=0}^{N-1} |\mathfrak{s}_k F_k|^2, \quad (49)$$

and p equals to $1/N$ when C is unitary. Therefore, using amplitude amplification [7], $\mathcal{O}((\log N)^2 / \sqrt{p})$ one- or two-qubit gates, as well as $\mathcal{O}(1/\sqrt{p})$ calls of O_x , O_s and their inverses, are needed to implement a circulant matrix operation C , where $O_s |0^L\rangle = \sum_{k=0}^{N-1} s_k |k\rangle$.

References

1. Shor, P.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26**, 1484 (1997)
2. Deutsch, D.: Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A Math. Phys. Eng. Sci.* **400**, 97 (1985). doi:[10.1098/rspa.1985.0070](https://doi.org/10.1098/rspa.1985.0070)
3. Bergland, G.: A guided tour of the fast Fourier transform. *IEEE Spectr.* **6**, 41 (1969)
4. Cleve, R., Watrous, J.: Fast parallel circuits for the quantum Fourier transform. In: 41st Annual Symposium on Foundations of Computer Science Proceedings, pp. 526–536 (2000)
5. Kitaev, A.Y.: Quantum measurements and the Abelian stabilizer problem. [arXiv:quant-ph/9511026](https://arxiv.org/abs/quant-ph/9511026) (1995)
6. Brassard, G., Høyer, P., Tapp, A.: Quantum counting. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) *Automata, languages and programming* No. 1443 in *Lecture Notes in Computer Science*, pp. 820–831. Springer, Berlin (1998)
7. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. *Contemp. Math.* **305**, 53–74 (2002)
8. Benenti, G., Strini, G.: Quantum simulation of the single-particle Schrödinger equation. *Am. J. Phys.* **76**, 657 (2008)
9. Kassal, I., Jordan, S.P., Love, P.J., Mohseni, M., Aspuru-Guzik, A.: Polynomial-time quantum algorithm for the simulation of chemical dynamics. *Proc. Natl. Acad. Sci.* **105**(48), 18681 (2008)

10. Szkopek, T., Roychowdhury, V., Yablonoitch, E., Abrams, D.S.: Eigenvalue estimation of differential operators with a quantum algorithm. *Phys. Rev. A* **72**, 062318 (2005)
11. Hales, L., Hallgren, S.: An improved quantum Fourier transform algorithm and applications. In: 41st Annual Symposium on Foundations of Computer Science Proceedings, pp. 515–525 (2000)
12. Schützhold, R.: Pattern recognition on a quantum computer. *Phys. Rev. A* **67**, 062311 (2003)
13. van Dam, W., Hallgren, S., Ip, L.: Quantum algorithms for some hidden shift problems. *SIAM J. Comput.* **36**, 763 (2006)
14. Jordan, S.P.: Fast quantum algorithm for numerical gradient estimation. *Phys. Rev. Lett.* **95**, 050501 (2005)
15. Qiang, X., Loke, T., Montanaro, A., Aungskunsiri, K., Zhou, X., O'Brien, J.L., Wang, J.B., Matthews, J.C.F.: Efficient quantum walk on a quantum processor. [arXiv:1510.08657](https://arxiv.org/abs/1510.08657) [quant-ph] (2015)
16. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009)
17. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum algorithms for supervised and unsupervised machine learning. [arXiv:1307.0411](https://arxiv.org/abs/1307.0411) (2013)
18. Rebentrost, P., Mohseni, M., Lloyd, S.: Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **113**(13), 130503 (2014)
19. Giovannetti, V., Lloyd, S., Maccone, L.: Architectures for a quantum random access memory. *Phys. Rev. A* **78**(5), 052310 (2008)
20. Grover, L., Rudolph, T.: Creating superpositions that correspond to efficiently integrable probability distributions. [arXiv:quant-ph/0208112](https://arxiv.org/abs/quant-ph/0208112) (2002)
21. Kaye, P., Mosca, M.: Quantum networks for generating arbitrary quantum states. [arXiv:quant-ph/0407102](https://arxiv.org/abs/quant-ph/0407102) (2004)
22. Soklakov, A.N., Schack, R.: Efficient state preparation for a register of quantum bits. *Phys. Rev. A* **73**, 012307 (2006)
23. Buhrman, H., Cleve, R., Watrous, J., de Wolf, R.: Quantum fingerprinting. *Phys. Rev. Lett.* **87**, 167902 (2001)
24. Nielsen, M.A., Chuang, I.L.: *Quantum computation and quantum information*. Cambridge University Press, Cambridge (2010)
25. Ramos, G.U.: Roundoff error analysis of the fast Fourier transform. *Math. Comput.* **25**(116), 757–768 (1971)
26. Tasche, M., Zeuner, H.: Worst and average case roundoff error analysis for FFT. *BIT Numer. Math.* **41**(3), 563–581 (2001)
27. Bell, R.: *Introductory Fourier transform spectroscopy*. Elsevier, Amsterdam (2012)
28. Delanty, M., Steel, M.: Discretely-observable continuous time quantum walks on Moöbius strips and other exotic structures in 3D integrated photonics. *Phys. Rev. A* **86**, 043821 (2012)
29. Yoneda, T., Sung, Y.M., Lim, J.M., Kim, D., Osuka, A.: PdII complexes of [44]- and [46] Decaphyrins: the largest Hückel aromatic and antiaromatic, and Möbius aromatic macrocycles. *Angew. Chem. Int. Ed. Engl.* **53**, 13169 (2014)
30. Olson, B.J., Shaw, S.W., Shi, C., Pierre, C., Parker, R.G.: Circulant matrices and their application to vibration analysis. *Appl. Mech. Rev.* **66**, 040803 (2014)
31. Cheng, B., Fan, J., Jia, X., Jia, J.: Parallel construction of independent spanning trees and an application in diagnosis on mobius cubes. *J. Supercomput.* **65**, 1279 (2013)
32. Golub, G.H., Van Loan, C.F.: *Matrix computations*, vol. 3. JHU Press, Baltimore (2012)
33. Mülken, O., Blumen, A.: Continuous-time quantum walks: models for coherent transport on complex networks. *Phys. Rep.* **502**, 37 (2011)
34. Childs, A.M.: *Quantum information processing in continuous time*. Ph.D. thesis, Massachusetts Institute of Technology (2004)
35. Childs, A.M., Kothari, R.: Limitations on the simulation of non-sparse hamiltonians. [arXiv preprint arXiv:0908.4398](https://arxiv.org/abs/0908.4398) (2009)
36. Cuccaro, S.A., Draper, T.G., Kutin, S.A., Moulton, D.P.: A new quantum ripple-carry addition circuit. [arXiv:quant-ph/0410184](https://arxiv.org/abs/quant-ph/0410184) (2004)
37. Draper, T.G., Kutin, S.A., Rains, E.M., Svore, K.M.: A logarithmic-depth quantum carry-lookahead adder. *Quantum Info. Comput.* **6**, 351–369 (2006)
38. Álvarez-Sánchez, J.J., Álvarez-Bravo, J.V., Nieto, L.M.: A quantum architecture for multiplying signed integers. *J. Phys. Conf. Ser.* **128**, 012013 (2008)

39. Vedral, V., Barenco, A., Ekert, A.: Quantum networks for elementary arithmetic operations. *Phys. Rev. A* **54**, 147 (1996)
40. Draper, T.G.: Addition on a quantum computer. [arXiv:quant-ph/0008033](https://arxiv.org/abs/quant-ph/0008033) (2000)
41. Ruiz-Perez, L., Garcia-Escartin, J.C.: Quantum arithmetic with the quantum Fourier transform. [arXiv:1411.5949](https://arxiv.org/abs/1411.5949) [quant-ph] (2014)
42. Maynard, C.M., Pius, E.: A quantum multiply-accumulator. *Quantum Inf. Process.* **13**(5), 1127 (2013). doi:[10.1007/s11128-013-0715-5](https://doi.org/10.1007/s11128-013-0715-5)
43. Maynard, C.M., Pius, E.: Integer arithmetic with hybrid quantum-classical circuits. [arXiv:1304.4069](https://arxiv.org/abs/1304.4069) (2013)
44. Pavlidis, A., Gizopoulos, D.: Fast quantum modular exponentiation architecture for Shor's factoring algorithm. *Quantum Info. Comput.* **14**(7&8), 649–682 (2014)
45. Kline, M.: *Calculus: an intuitive and physical approach*. Courier Corporation, North Chelmsford (1998)

Centrality measure based on continuous-time quantum walks and experimental realizationJosh A. Izaac,¹ Xiang Zhan,² Zhihao Bian,² Kunkun Wang,² Jian Li,² Jingbo B. Wang,^{1,*} and Peng Xue^{2,†}¹*School of Physics, The University of Western Australia, Perth, WA 6009, Australia*²*Department of Physics, Southeast University, Nanjing 211189, China*

(Received 30 September 2016; published 13 March 2017)

Network centrality has important implications well beyond its role in physical and information transport analysis; as such, various quantum-walk-based algorithms have been proposed for measuring network vertex centrality. In this work, we propose a continuous-time quantum walk algorithm for determining vertex centrality, and show that it generalizes to arbitrary graphs via a statistical analysis of randomly generated scale-free and Erdős-Rényi networks. As a proof of concept, the algorithm is detailed on a four-vertex star graph and physically implemented via linear optics, using spatial and polarization degrees of freedoms of single photons. This paper reports a successful physical demonstration of a quantum centrality algorithm.

DOI: [10.1103/PhysRevA.95.032318](https://doi.org/10.1103/PhysRevA.95.032318)**I. INTRODUCTION**

Since the seminal paper by Aharonov *et al.* [1], quantum walks have become a fundamental tool in quantum information theory [2], allowing us to bridge the often more esoteric world of quantum computation and algorithms [3–7] with real-life graph and network theory [8–10] and dynamical quantum modeling applications [11–14]. This is due, in part, to the markedly differing behavior of the quantum walk compared to its classical analog. Harnessing the effects of superposition, quantum coherence, and entanglement, the quantum walk propagates quadratically faster, providing a key source for new quantum algorithms and a platform for universal quantum computation [15–17].

Like the classical case, quantum walks are divided by two distinct approaches: the discrete-time quantum walk (DTQW), which introduces spin states and a quantum coin operation with discrete time-evolution operators; and the continuous-time quantum walk (CTQW), which evolves the walker continuously in time [18]. Due to the enlarged Hilbert space and higher degrees of freedom of the DTQW, the relationship between these two formulations is inherently nontrivial; regardless, an equivalency has been explored using both a limiting approach [19] and percolation theory [20].

One potential application of the quantum walk is in providing an efficient quantum algorithm for vertex centrality ranking in network analysis. Previous studies have proposed algorithms built on the standard discrete-time quantum walk [21], the Szegedy discrete-time quantum walk [22–24], or the continuous-time quantum stochastic walk (QSW) [24–26]. However, while comparing well to classical centrality measures, these have the distinct disadvantage of requiring expanded Hilbert spaces (up to N^2 dimensions for a graph of N vertices) or, in the case of the QSW, muting the quantum behavior due to decoherence.

In this study, we propose an alternative quantum walk centrality algorithm based on the CTQW, allowing us to preserve the full quantum behavior of the walker while limiting the dimension of the Hilbert space to N . Furthermore, we have

experimentally implemented this algorithm in the case of the four-vertex star graph.

This paper is structured as follows. In Sec. II, we describe the continuous-time quantum walk and its relationship to classical random walks. We then briefly discuss graph centrality measures in Sec. III, before introducing our CTQW-based quantum centrality scheme in Sec. IV. A thorough statistical analysis using ensembles of randomly generated graphs is presented in Sec. V, highlighting the suitability of the quantum centrality scheme for general graphs. Then we discuss our experimental implementation via linear optics in Sec. VI, before finally presenting our conclusions in Sec. VII.

II. CLASSICAL AND QUANTUM WALKS**A. Classical random walks**

Consider an arbitrary undirected graph $G(V, E)$, composed of vertices $j \in V$ and edges $(i, j) \in E$, with $|V| = N$. The adjacency matrix of G is a symmetric matrix defined by

$$A_{ij} = \begin{cases} 1, & (i, j) \in E, \\ 0, & (i, j) \notin E. \end{cases} \quad (1)$$

A discrete-time random walk (DTRW) over G is a stochastic Markovian process that evolves as

$$\mathbf{P}^{(n+1)} = T\mathbf{P}^{(n)}, \quad (2)$$

where $\mathbf{P}_i^{(n)}$ represents the probability of finding the walker at vertex i at time step n , and T the transition matrix. As per convention, the transition matrix is normally taken to be

$$T = AD^{-1}, \quad (3)$$

with $D_{ij} = \delta_{ij} \sum_k A_{ki}$ a diagonal matrix containing the vertex degrees of the graph. This ensures that T is *stochastic* ($\sum_k T_{ki} = 1$), preserving the probability of the walker. The steady-state limiting probability distribution of the walker,

$$\lim_{n \rightarrow \infty} T^n \mathbf{P}^{(0)} = \boldsymbol{\pi}, \quad (4)$$

must satisfy the equation $T\boldsymbol{\pi} = \boldsymbol{\pi}$. Thus, the limiting distribution is simply the eigenvector of T with eigenvalue $\lambda = 1$. It is trivial to show that the limiting distribution is therefore

*jingbo.wang@uwa.edu.au

†gnep.eux@gmail.com

proportional to the vertex degree and given by

$$\pi_j = \frac{D_{jj}}{\text{Tr}(D)} = \frac{\sum_i A_{ij}}{\sum_i \sum_j A_{ij}}. \quad (5)$$

Note that in cases where a graph only has even-length closed loops, the walker will only occupy sites an even distance from the initial state at even time-steps—causing the walker never to converge to its stationary distribution π . In such cases, it is useful to redefine the DTRW so that at every time step, the walker has only a ϵ probability of moving per the transition matrix:

$$\begin{aligned} \mathbf{P}^{(n+1)} &= \epsilon T \mathbf{P}^{(n)} + (1 - \epsilon) \mathbf{P}^{(n)} \\ &= \epsilon (T - I) \mathbf{P}^{(n)} + \mathbf{P}^{(n)} \end{aligned} \quad (6)$$

This is known as the *lazy random walk* and is sufficient to break the periodicity and ensure convergence to the limiting probability distribution π .

We can interpret each time step of the lazy random walk as corresponding to a time of ϵ [27]. Thus, by rearranging this equation and taking the limit $\epsilon \rightarrow 0$,

$$\lim_{\epsilon \rightarrow 0} \frac{\mathbf{P}^{(n+1)} - \mathbf{P}^{(n)}}{\epsilon} = -(I - T) \mathbf{P}^{(n)}, \quad (7)$$

we arrive at the master equation, a stochastic Markovian process governing the time evolution of the *continuous-time random walk* (CTRW),

$$\frac{d}{dt} \mathbf{P}(t) = -L \mathbf{P}(t) \quad (8)$$

with solution $\mathbf{P}(t) = e^{-Lt} \mathbf{P}(0)$, where L is the normalized graph Laplacian,

$$L = I - T = (D - A)D^{-1}, \quad (9)$$

such that e^{-Lt} is stochastic, and the walk is probability conserving. For the CTRW, the steady-state limiting probability distribution,

$$\pi = \lim_{t \rightarrow \infty} e^{-Lt} \mathbf{P}(0), \quad (10)$$

must satisfy the equation $e^{-Lt} \pi = \pi$. After expanding the matrix exponential as a Taylor series, it can be seen that this is equivalent to $L\pi = \mathbf{0}$; i.e., π is the null space of L . Note that since $L = I - T$, $L\pi = (I - T)\pi = \mathbf{0} \Rightarrow T\pi = \pi$, and thus the CTRW limiting distribution and the DTRW limiting distribution [Eq. (5)] are identical.

B. Continuous-time quantum walks

The CTRW's quantum analog,¹ the continuous-time quantum walk on graph G , has its time evolution governed instead

¹We do not describe the quantum analog of the DTRW here—the discrete-time quantum walk DTQW—as, unlike the classical case, it has a highly nontrivial relationship with the CTQW due to the existence of an additional ‘coinspace.’ For a good introduction to the coined DTQW, see Kempe [2] or Manouchehri and Wang [6], while Szegedy [28] offers a good introduction to the Szegedy DTQW formalism.

by the Schrödinger equation [18],

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H |\psi(t)\rangle, \quad (11)$$

where H is the system Hamiltonian, encoding the discrete structure of the underlying graph G , and $|\psi(t)\rangle = \sum_j \alpha_j(t) |j\rangle$ the complex-valued state vector. We use atomic units henceforth and, thus, set $\hbar = m = e = 1$. The general solution to the system is

$$|\psi(t)\rangle = U(t) |\psi(0)\rangle = e^{-iHt} |\psi(0)\rangle. \quad (12)$$

Consistent with standard quantum formalism, $\alpha_j(t) = \langle j | \psi(t) \rangle \in \mathbb{C}$ is the probability amplitude, and $|\alpha_j(t)|^2$ the corresponding probability, of the walker found at node j after time t . Unlike the classical CTRW, the CTQW gains properties characteristic of quantum systems—including time reversibility (hence, no limiting state) and superposition, allowing propagation through networks quadratically faster than its classical counterpart [18,29]. However, the CTQW is no longer a stochastic process but, rather, deterministic; the probabilistic nature of the walk comes from measuring the quantum state, rather than the walk's dynamics [30].

It is important to note that there are two competing conventions for the CTQW Hamiltonian that are ubiquitous in the field: the adjacency matrix ($H = A$) and the (combinatorial) Laplacian ($H = D - A$) [31]. Both provide similar dynamics (and are identical for degree regular graphs), with each being preferred for particular applications—the adjacency matrix for simplicity in quantum computation calculations and the Laplacian for its discrete approximation to the kinetic energy operator of quantum mechanics. In this study, we use $H = A$, for reasons that become clear in subsequent sections.

III. GRAPH CENTRALITY

In the study of network structure and graph theory, centrality measures are an integral tool, allowing determination and ranking of vertices deemed to be most important. Due to the large number of physical systems that can be modeled as networks, this has seen wide application across multiple disciplinary fields, including technology (ranking websites for search engines [32], power distribution [33]), business (organizational management [34–36]), biology (grooming networks in macaques [37]), and biochemistry (finding active sites in proteins [38]).

At its most basic, a graph centrality measure C satisfies the following properties:

(i) $C : G(V, E) \rightarrow \mathbb{R}^{|V|}$ is a function or algorithm that accepts a graph and returns a real-valued vector over the set of vertices V .

(ii) Higher values are provided to vertices deemed more ‘important’ or ‘central’ to the graph structure, with lower values provided to vertices with a reduced ‘importance’ or ‘centrality.’

However, what constitutes importance is subjective; it depends on the application or model to be analyzed and how information ‘flows’ throughout the network [39]. For example, information might flow predominantly through paths (a sequence of unique edges and vertices, characteristic of bacterial and viral infections [40]), trails (vertices can be

revisited but each edge is only traversed once—the flow of gossip in social networks [41]), and walks (where there is no restrictions on edge and vertex sequences, for example, banknote exchange in a population). Moreover, this flow can occur through serial duplication (traveling via one edge at each time step: gift exchange) or parallel duplication (traversing multiple edges simultaneously: radio broadcasting).

Thus, it is important to apply a centrality measure that models information flow corresponding to the network under study; failure to do so may result in poor results and even the inability to correctly interpret the results [39]. To deal with this plethora of scenarios, various classical centrality measures have been introduced: degree centrality, eigenvector centrality, betweenness centrality, closeness centrality, and PageRank, among others. Of these, degree, eigenvector, and PageRank centrality are what is known as *radial parallel duplication* measures, which measure network flow via walks emanating from or terminating at particular nodes [42].

A. Degree centrality

The degree centrality measure, calculated via the row sums of the adjacency matrix,

$$C_j^{(\text{deg})} = \frac{\text{deg}(v_j)}{\sum_k \text{deg}(v_k)} = \frac{\sum_i A_{ij}}{\sum_i \sum_j A_{ij}}, \quad (13)$$

is based on walks of length 1 emanating from each vertex and is useful in cases dealing with direct and immediate influence between nodes. Further, it can be seen that the limiting probability distribution of classical random walks is proportional to the node degree, allowing the degree centrality to be simulated via a Markovian process.

B. Eigenvector centrality

Eigenvector centrality, on the other hand, is given by $C_j^{(\text{ev})} = \mathbf{v}_j$, where \mathbf{v} is an eigenvector of the adjacency matrix $A\mathbf{v} = \lambda\mathbf{v}$ corresponding to the maximum eigenvalue to ensure, via the Perron-Frobenius theorem, that the ranking remains strictly positive. It has been shown by Bonacich [43] that the eigenvector centrality is proportional to the row sums of matrix S , $\mathbf{v}_j \propto \sum_i S_{ij}$, where

$$S = A + \frac{1}{\lambda}A^2 + \frac{1}{\lambda^2}A^3 + \dots = \sum_{n=1}^{\infty} \lambda^{1-n} A^n, \quad (14)$$

i.e., the eigenvector centrality counts walks of *all* lengths, weighted inversely by length, from each node. Thus, unlike the degree centrality, the eigenvector centrality considers long-term ‘indirect’ influence: if a vertex is connected to another node with a high number of connections, the first vertex will likewise have a high centrality measure. Consequently, rather than model the eigenvector centrality via the DTRW—which may only sample adjacent vertices at each time step—we can instead use the continuous-time random walk. Due to its matrix exponential time propagator, the CTRW performs walks of *all* lengths at each infinitesimal time step dt .

C. PageRank

One final classical centrality measure which necessitates introduction is the Google PageRank [32]. A variation of the eigenvector centrality, PageRank was developed as a ranking algorithm for sites on the World Wide Web and has accumulated significant prestige as the algorithm behind the Google search engine. In this context, vertices represent websites, with directed edges the links between them. Due to the need to take direction into account, issues arise with eigenvector centrality, namely, nodes with in-degree but no out-degree (‘dangling nodes’) accumulate probability, due to the adjacency matrix’s being nonstochastic. To address this issue, the eigenvector centrality method is instead applied to the Google matrix G ,

$$G = \alpha E + \frac{1}{N}(1 - \alpha)J, \quad 0 \leq \alpha \leq 1, \quad (15)$$

where N is the number of vertices in the graph, E is the *patched adjacency matrix*, column-normalized to ensure that G is stochastic,

$$E_{ij} = \begin{cases} A_{ij} / \sum_k A_{kj}, & \sum_k A_{kj} \neq 0, \\ 1/N, & \sum_k A_{kj} = 0, \end{cases} \quad (16)$$

and J is the all 1’s matrix. The addition of J is to provide a small ‘random surfer effect,’ i.e., a nonzero uniform probability that a walker at a particular vertex can jump to *any* other vertex, even in cases of nonadjacency. In practice, α is generally chosen to be 0.85, providing a good compromise between information flow via hyperlinks and the random surfer effect.

Once the Google matrix is calculated, the PageRank centrality measure is then applied by solving the eigenvector equation

$$G\mathbf{x} = \mathbf{x}, \quad (17)$$

as, per the Perron-Frobenius theorem, the eigenvector corresponding to the largest eigenvalue ($\lambda = 1$ for PageRank, as G is stochastic) will be strictly positive. Note that this equation is identical to that of a DTRW; thus, the PageRank can be modeled as a DTRW with G taken to be the transition matrix. Nevertheless, when $\alpha < 1$, PageRank continues to model its centrality measure on walks of all lengths, due to the random surfer effect. To see this explicitly, it can be easily shown that, in the case $0 \leq \alpha < 1$,

$$\begin{aligned} G\mathbf{x} &= \mathbf{x} \\ \Rightarrow \alpha E\mathbf{x} + \frac{1}{N}(1 - \alpha)J\mathbf{x} &= \mathbf{x} \\ \Rightarrow (I - \alpha E)\mathbf{x} &= \frac{1}{N}(1 - \alpha)J\mathbf{x} \end{aligned} \quad (18)$$

has the exact solution

$$\begin{aligned} \mathbf{x} &= (I - \alpha E)^{-1} \frac{1}{N}(1 - \alpha)J\mathbf{x} \\ &= (1 - \alpha) \left(\sum_{k=0}^{\infty} \alpha^k E^k \right) \sum_{j=1}^N \frac{\mathbf{e}_j}{N} \\ \therefore x_i &= \frac{1}{N}(1 - \alpha) \sum_{j=1}^N \left(\sum_{k=0}^{\infty} \alpha^k E^k \right)_{ij}, \end{aligned} \quad (19)$$

and therefore \mathbf{x} is calculated using walks of k lengths for all $k \in \mathbb{N}$, weighted by α^k [44].

D. Random walk centrality

The random walk centrality (RWC), unlike the centralities previously discussed, is not a radial-volume-based measure (counting the number of walks between each pair of nodes) but rather a radial *length*-based measure, quantifying the length of the walks between nodes [42]. Alternatively, this can be interpreted as a measure of the expected time for information to arrive at a particular node, i.e., the effectiveness or speed of communication [39]. The RWC measure, introduced by Noh and Rieger [45] and based on a DTRW, is given by

$$C_j^{(\text{RWC})} = \frac{\pi_j}{\tau_j}, \quad (20)$$

where π is the random walk limiting distribution, and

$$\tau_j = \sum_{n=0}^{\infty} (T_{jj}^n - \pi_j) \quad (21)$$

is the characteristic relaxation time of vertex j .

E. Quantum centrality measures

The above-described walk-based centrality measures are classical in nature. However, in recent years several quantum centrality measures have been proposed—ranging from quantizations of the aforementioned classical measures to wholly new proposals—that take advantage of the exponential speedups offered by quantum computation. For example, Quantum PageRank (introduced by Paparo *et al.* [22,23] and extended by Loke *et al.* [24]) utilizes the Szegedy quantum walk [28] (a DTQW formulation) to quantize the directed Markov chains encoded by the Google matrix, before taking the long-time average of the walk's probability distribution: in essence, providing a quantum analog of PageRank centrality.

The quantum stochastic walk is another approach, which makes use of the Lindblad master equation to introduce environmental decoherence to a CTQW [24–26]. In practice, this has the effect of creating a continuous-time walk continuum parametrized by ω , with $\omega = 0$ (no dephasing) corresponding to a purely quantum walk (CTQW) on an undirected graph and $\omega = 1$ (complete dephasing) corresponding to a purely classical walk (CTRW) over a digraph. By restricting the domain to $0 < \omega \ll 1$, quantum dynamics and the resulting quantum speedup are retained, however, the walker will eventually converge to the CTRW limiting probability distribution [30]. Similarly to Quantum PageRank, the centrality measure is then given by the long-time average of the probability distribution.

Finally, Berry and Wang [21] proposed a novel method, in which the quantum search algorithm is applied to graph structures via the DTQW; the resulting frequency of successful search probability was shown to correlate with the (lazy) random walk centrality of Noh and Rieger [45]. Thus, the quantum centrality scheme of Berry and Wang [21] differs from the two previous quantum centrality schemes, as it considers the mean speed of the walker in transmitting information over the network: it is a form of *quantum closeness centrality*.

Unfortunately, when it comes to physically implementing these quantum centrality measures, we run into various issues. In all three cases, due to the use of either a coin state (DTQW) or an environment (QSW), the size of the state space must be significantly increased, taking us beyond the experimental ability to simulate quantum graph centrality of even simple graph structures. For example, for a graph of N vertices, the Szegedy DTQW formulation used in the Quantum PageRank scheme requires a state space of size N^2 . As such, the ability to physically realize these quantum centrality measures is currently beyond our reach.

Instead, in the following section we propose a CTQW-based centrality measure—building on the foundation of classical radial centrality measures such as eigenvector centrality while allowing us to take advantage of the quantum speedup afforded versus the CTRW and utilizing a significantly smaller state space than the QSW and DTQW.

IV. CTQW-BASED CENTRALITY MEASURE

Similarly to the Quantum PageRank and the QSW centrality measures, as the time evolution of the CTQW is determined by the Hamiltonian—and thus the underlying network structure—one method for extracting the centrality information is to simply start the walker in an equal superposition of all vertex states, $|\psi(0)\rangle = \frac{1}{\sqrt{N}} \sum_j |j\rangle$ (so as not to bias any one particular vertex) and compare the time-average probability of locating the walker at each vertex.

Now, convention allows for two choices for the Hamiltonian: we may choose either the adjacency matrix A or the graph Laplacian L (given by $L_{ij} = \delta_{ij} \sum_k A_{ik} - A_{ij}$, a discrete approximation to the continuous-space Laplacian). However, the construction of the Laplacian ensures that an equal superposition state is always an eigenvector, resulting in a stationary time evolution:

$$U|\psi(0)\rangle = e^{-iLt} \left(\frac{1}{N} \sum_j |j\rangle \right) = \frac{1}{N} \sum_j |j\rangle \quad \forall t. \quad (22)$$

As such, the Laplacian is ill suited for a CTQW centrality measure, as it will be unable to distinguish vertices more central to the network structure. This is not the case for the adjacency matrix; thus, for the remainder of this work, we set $H = A$.

To briefly summarize, the proposed CTQW centrality scheme works as follows:

- (1) Prepare the quantum walker in an initial equal superposition over all vertex states: $|\psi(0)\rangle = \frac{1}{N} \sum_j |j\rangle$.
- (2) Propagate the walker for time $t \gg 0$: $|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle$, where $H = A$ is the graph adjacency matrix.
- (3) Calculate the long-time average probability distribution of finding the walker at each vertex:

$$C_j^{(\text{CTQW})} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau |\langle j | \psi(t) \rangle|^2 dt. \quad (23)$$

To fully ascertain the reliability of the proposed CTQW centrality measure, we consider both a simple example (allowing us to qualitatively assess the measures performance) and a rigorous statistical analysis comparing the CTQW measure to the PageRank over an ensemble of randomly generated graphs.

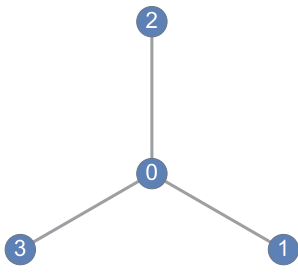


FIG. 1. Four-vertex star graph.

Freeman [46], in his discussion of the canonical formulations of centrality measures, noted that the degree, closeness, and betweenness centralities all attain their highest values for the central node of the star graph; Borgatti and Everett [42], in reviewing Freeman's work, suggested that this may serve as a defining characteristic of a 'proper' centrality measure. Thus, let us consider a four-vertex star graph as an example of the proposed CTQW centrality measure.

For the four-vertex star graph shown in Fig. 1, the adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad (24)$$

with the first vertex (vertex 0) the central node. In this case, the time-evolution operator is given by

$$U(t) = \frac{1}{3} \begin{bmatrix} 3c(t) & s(t) & s(t) & s(t) \\ s(t) & c(t) + 2 & c(t) - 1 & c(t) - 1 \\ s(t) & c(t) - 1 & c(t) + 2 & c(t) - 1 \\ s(t) & c(t) - 1 & c(t) - 1 & c(t) + 1 \end{bmatrix}, \quad (25)$$

where $c(t) = \cos(\sqrt{3}t)$ and $s(t) = -i\sqrt{3} \sin(\sqrt{3}t)$. Using this operator to propagate from an initial equal superposition of vertex states $|\psi(0)\rangle = \frac{1}{4} \sum_j |j\rangle$, the probability of locating the walker on vertex j at time t is

$$\begin{aligned} |\langle j|\psi(t)\rangle|^2 &= |\langle j|U(t)|\psi(0)\rangle|^2 \\ &= \left[\frac{1}{2} - \frac{1}{4}c(2t) \right] \delta_{j0} + \left[\frac{1}{6} + \frac{1}{12}c(2t) \right] \sum_{j'=1}^3 \delta_{jj'} \end{aligned} \quad (26)$$

(Fig. 2). Noting that this probability distribution is periodic with period $T = \pi/\sqrt{3}$, the CTQW centrality measure becomes

$$\begin{aligned} C_j^{(\text{CTQW})} &= \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau |\langle j|\psi(t)\rangle|^2 dt \\ &= \frac{1}{T} \int_0^T |\langle j|\psi(t)\rangle|^2 dt, \end{aligned} \quad (27)$$

yielding values of $1/2$ for $j = 0$ and $1/6$ for $j = 1, 2, 3$. This fits well with what would be expected intuitively: the central vertex (vertex 0) has the highest time-averaged probability, indicating a high centrality measure, while the remaining vertices (1–3) are equivalent and have an equal and lower ranking. The proposed CTQW centrality measure therefore

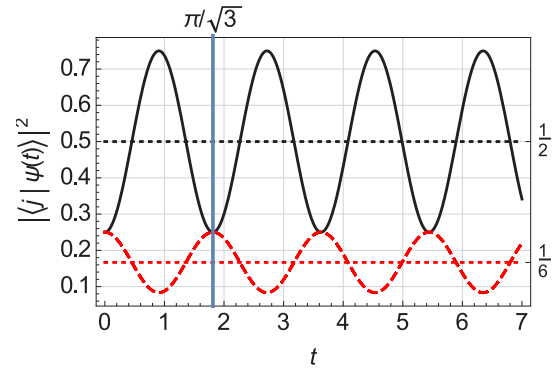


FIG. 2. CTQW probability at vertex 0 (solid black curve) and vertices 1, 2, and 3 (dashed red curve) on a four-vertex star graph. The initial state is an equal superposition of all vertex states. Dotted horizontal lines show the respective long-time averaged probabilities of the respective vertices, with the vertical blue line denoting one period ($T = \pi/\sqrt{3}$).

satisfies one of the defining properties of centrality measures; however, a detailed statistical analysis is required to properly assess its behavior for general graphs.

V. STATISTICAL ANALYSIS

To investigate the reliability of this newly proposed quantum centrality algorithm, it is pertinent to compare its ranking results to classical algorithms on large random graphs. To do so, we consider two classes of random graphs: Erdős-Rényi networks and scale-free networks. A random Erdős-Rényi graph, denoted $G(N, p)$, is comprised of N vertices with edges randomly distributed per the Bernoulli distribution, with probability p [47,48]. For such a network, the vertex degree distribution $P(k)$ (the fraction of vertices with degree k) is binomial in form, resulting in most vertices having a degree close to np , the mean number of connections. Scale-free networks, on the other hand, are characterized by a power-law vertex distribution $P(k) \sim k^{-\gamma}$, due to the presence of a small number of highly connected 'hubs'—with a majority of vertices exhibiting a significantly lower degree [49,50]. As such, they form an important tool in modeling real-life networks with similar characteristics, such as social networks, the World Wide Web, and biochemical molecules [51,52].

A. Correlation to classical measures

First, let us investigate the correlation between the CTQW centrality measure and classical measures. As centrality measures only provide useful information for the top five or so valued vertices (with noise growing successively larger for lower ranked nodes [53]), we consider a randomly generated 20-vertex Erdős-Rényi graph $G(20, 0.3)$ and use this as the basis of our correlation test. The graph generated and its respective vertex centrality values (calculated using the degree centrality, PageRank, eigenvector centrality, CTQW centrality, and RWC centrality) are shown in Fig. 3. Qualitatively, it can be seen that all centrality measures strongly agree on the top-ranked vertices, with slight variations for the lower ranked vertices, as expected.

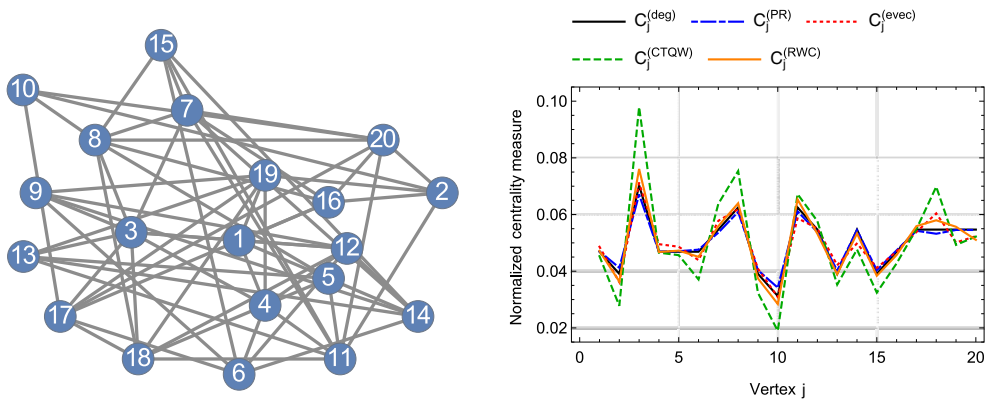


FIG. 3. Left: Randomly generated Erdős-Rényi graph $G(20,0.3)$. Right: Normalized vertex centrality values for vertex j on $G(20,0.3)$. Measures shown are the degree centrality, PageRank, eigenvector centrality, CTQW centrality, and RWC centrality.

To get a more quantitative understanding of the correlation, we employ Kendall's tau rank correlation coefficient [54] ($\tau \in [-1,1]$, where $\tau = 1$ indicates a perfect correlation between ranked lists, $\tau = 0$ indicates no correlation, and $\tau = -1$ indicates a perfect anticorrelation). Kendall's tau correlation coefficients for Fig. 3 are listed in Table I; for additional robustness, this analysis is repeated and averaged over an ensemble of 100 randomly generated $G(20,0.3)$ graphs. It can be seen that there is a significant correlation between the CTQW centrality ranking and the eigenvector centrality ranking ($\tau = 0.592$ averaged across the ensemble). If we recall that the CTQW propagator is the matrix exponential, this is perhaps not so surprising; the CTQW centrality scheme appears to be ranking the graph vertices in a similar fashion to the eigenvector centrality, by considering walks of *all* lengths emanating from each vertex weighted inversely by length.

Note that Kendall's tau coefficient tells us how correlated the *entire* ranked lists are, allowing us to classify the centrality measures based on how they encode information flow through the network. However, beyond the topmost-ranked vertices, centrality measures convey very little useful information regarding remaining vertices; this is more the domain of

influence measures [53,55]. As such, Kendall's tau coefficient is not useful for determining *general agreement* between centrality measures on the location of the *most central* nodes. For example, in Fig. 3 it can be seen that the PageRank (a radial volume measure) and random walk centrality (a radial length measure) are in total agreement on the location of the top three most central vertices while exhibiting a low correlation ($\tau = 0.189$).

B. Agreement on top-ranked vertices

Here, we consider ensembles of larger Erdős-Rényi and scale-free graphs and compare the CTQW centrality to the eigenvector centrality (its closest classical analog) and to the PageRank (the classical centrality measure with arguably the most impact in the last decade). This analysis will allow us to verify the behavior of the CTQW centrality for large graphs of varying degree distributions.

We begin by generating an ensemble of 200 Erdős-Rényi and scale-free graphs (the latter by way of the Barabási-Albert algorithm) and calculating the average PageRank, eigenvector, and CTQW centrality measures over the ensemble. These results are shown in Fig. 4. It can be seen that, on average, the CTQW ranking agrees with the classical algorithms on the location of the five most central vertices, while also following the following the same general trend line (binomial for the Erdős-Rényi, power law for the scale-free). In fact, the CTQW measure for the top five vertices *outperforms* that of the PageRank and eigenvector centrality, by assigning a higher centrality measure, perhaps allowing for greater distinguishability when sampled experimentally. However, it appears that this comes at the cost of larger measure variance compared to PageRank.

We now attempt to quantify the 'agreement' regarding the top-ranked vertices between the quantum and the classical measures. Previously Kendall's tau ranked-list coefficient was used; while this works well for determining correlations between various ranked lists, it is less useful for quantifying how often the ranked lists agree on their first few values. Thus, we detail an alternative approach. For each graph in the ensemble, unordered sets containing the n most central vertices according to each measure were compared—the fraction of matching vertices providing a quantitative value for the

TABLE I. Kendall's tau coefficient comparing the vertex rankings for the labeled centrality measures; (top table) calculated for the Erdős-Rényi graph shown on the left in Fig. 3, and (bottom table) averaged over an ensemble of 100 random Erdős-Rényi graphs $G(20,0.3)$.

	Degree	PageRank	Eigenvector	CTQW	RWC
Degree	1.	0.316	0.232	0.232	0.411
PageRank	0.316	1.	0.011	0.011	0.189
Eigenvector	0.232	0.011	1.	1.	0.568
CTQW	0.232	0.011	1.	1.	0.568
RWC	0.411	0.189	0.568	0.568	1.
	Degree	PageRank	Eigenvector	CTQW	RWC
Degree	1.	0.276	0.163	0.15	0.236
PageRank	0.276	1.	0.107	0.076	0.12
Eigenvector	0.163	0.107	1.	0.592	0.277
CTQW	0.15	0.076	0.592	1.	0.228
RWC	0.236	0.12	0.277	0.228	1.

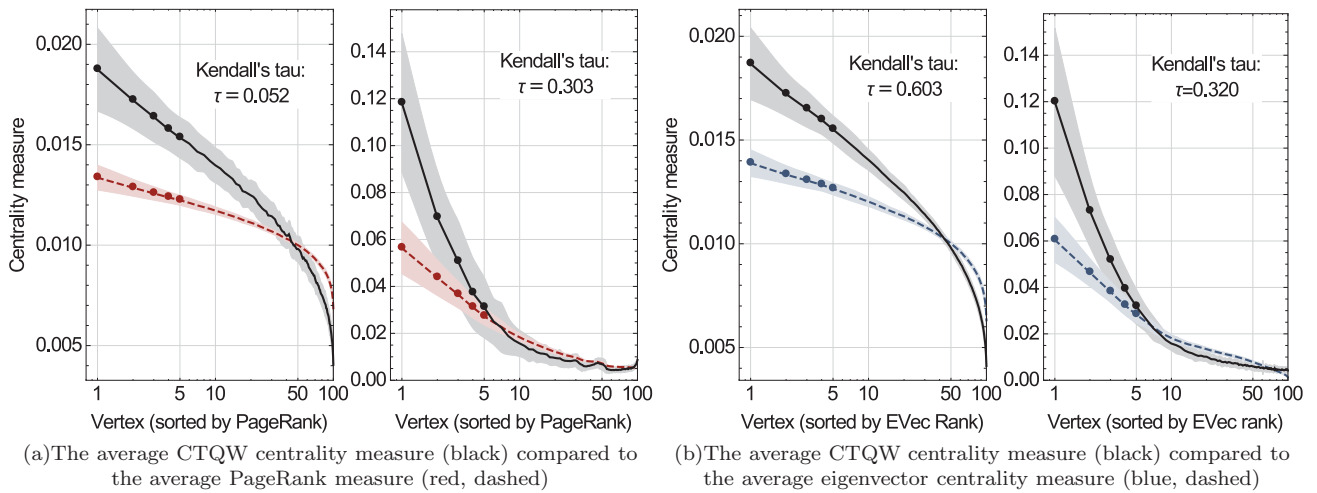


FIG. 4. The average CTQW centrality measure compared to classical centrality measures for vertices in an ensemble of 200 Erdős-Rényi graphs (left) and 200 scale-free graphs (right). The Erdős-Rényi graphs have parameters $N = 100$, $p = 0.3$. The scale-free networks are constructed via the Barabási-Albert algorithm with $N = 100$ and $m = 2$ edges added at every generation. The shaded areas represent one standard deviation from the average centralities, and the top five ranked vertices are shown by the symbols.

agreement between the two measures (termed the *agreement factor*). These agreement factors were then averaged over the entire ensemble, with the uncertainty approximated by calculating the Agresti-Coull 95% confidence interval; the results are presented in Fig. 5.

It can be seen that the eigenvector and CTQW centrality measures are in near-perfect agreement—agreeing on the most central node 99% of the time for Erdős-Rényi networks and 96% of the time for scale-free networks. This is likely due to the strong correlation between the CTQW and the eigenvector

centralities noted previously and indicates that this strong correlation continues to hold for larger graphs of varying degree distributions.

Turning our attention to PageRank, we find a strong agreement with the CTQW measure, albeit not as strong as the eigenvector centrality; regarding the location of the most central vertex, they named the same vertex 88% of the time for scale-free graphs, dropping to 70% for Erdős-Rényi graphs. As the number of vertices compared increases, the agreement factors decrease slightly for the scale-free and increase slightly for the Erdős-Rényi graphs, before both ending at around 80% by the time the top five vertices are compared. This discrepancy might be partially explained by considering the CTQW measure variance in Fig. 4:

(i) For the Erdős-Rényi graphs, a majority of vertices have a degree close to the mean, leading to the top-ranked vertices having similar centrality measures. The average CTQW centrality measure of the second- and third-ranked vertices lies *within* the uncertainty region of the most central vertex; so even as the top five are easily distinguished, changes in their initial ordering might appear.

(ii) For the scale-free graphs, with a small number of connected hubs, the hubs are easily distinguished by both measures. However, beyond the hubs, most vertices have similar degrees due to the power-law distribution, leading to small discrepancies between the measures as more vertices are ranked.

Nonetheless, our results here show that the CTQW measure proposed works excellently as a centrality measure—it assigns higher values to the central node of a star graph and equal lower values to the surrounding nodes, correlates well with the classical eigenvector centrality (allowing us to posit that the CTQW measure extracts centrality in a similar fashion to the eigenvector centrality, namely, via weighted walks of all lengths), and generalizes to arbitrary random scale-free and Erdős-Rényi graphs. Thus, the proposed quantum scheme sufficiently determines the node centrality, and in contrast to the Quantum PageRank algorithm (which requires

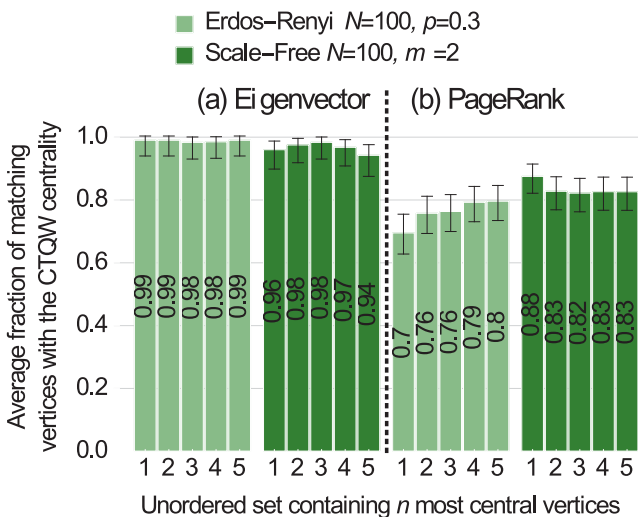


FIG. 5. Charts showing the agreement between the CTQW centrality ranking and (a) the PageRank and (b) the eigenvector centrality ranking for an ensemble of 100 Erdős-Rényi and 100 scale-free graphs. Each bar represents the unordered set containing the n most central vertices as determined by the PageRank and CTQW measures, while the vertical axis gives the average fraction of matching vertices between the two sets. Error bars indicate the Agresti-Coull 95% confidence interval.

computation of the dense Google matrix), preserves the sparse structure of the network in the Hamiltonian: a property that allows for known efficient quantum implementation [56].

In the following section, we build off this result to experimentally implement the CTQW centrality scheme for a star graph using linear optics—a proof-of-concept experiment and the first physical implementation (to our knowledge) of quantum centrality.

VI. EXPERIMENTAL REALIZATION

Linear optics enables the efficient implementation of an arbitrary unitary transformation on various degrees of freedoms of single photons. For example, any 2×2 unitary transformations on the polarizations of single photons can be realized by a set of half-wave plates (HWPs) and quarter-wave plates (QWPs) [57]. Here we aim to devise a linear optics realization of a 4×4 unitary transformation using spatial and polarization degrees of freedom of single photons.

In this experiment, we first prepare a four-dimensional equal superposition quantum state $|\psi(0)\rangle = \frac{1}{2} \sum_{j=0}^3 |j\rangle$ and then perform 4×4 unitary transformations on the state. We obtain the probability distribution through projective measurement on the state. The unitary transformations applied to the initial state $|\psi(0)\rangle$ are $U(k\Delta t)$, where $k \in \{1, 2, \dots, 8\}$.

An arbitrary 4×4 unitary transformation can be decomposed using the cosine-sine decomposition method [58–63]. For each unitary transformation in $U(k\Delta t)$, there exist unitary matrices \mathbb{L} , \mathbb{S} , and \mathbb{R} , such that $U = \mathbb{L}\mathbb{S}\mathbb{R}$, where \mathbb{L} and \mathbb{R} are block-diagonal,

$$\mathbb{L} = \begin{bmatrix} L & 0 \\ 0 & L' \end{bmatrix}, \quad \mathbb{R} = \begin{bmatrix} R & 0 \\ 0 & R' \end{bmatrix}, \quad (28)$$

and \mathbb{S} is an orthogonal cosine-sine matrix,

$$\mathbb{S} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (29)$$

where L , L' , R , and R' are arbitrary 2×2 unitary transformations on two modes. This matrix \mathbb{S} can be further decomposed by a 2×2 unitary transformation $S = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$ in the subspace spanned by modes $\{|0\rangle, |2\rangle\}$ and $\mathbb{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ in the subspace spanned by modes $\{|1\rangle, |3\rangle\}$.

This decomposition method can be used to decompose any higher dimensional unitary operations into series of two-dimensional unitary operations, and thus our technology can be used to realize, in principle, any dimensional unitary operations. However, it is noteworthy that the numbers of beam displacers (BDs) used to prepare a d -dimensional state and to realize a d -dimensional unitary operation are $d/2 - 1$ and $2^{d/2} - 2$, respectively, where d is an even positive integer. In other words, the number of optical elements grows exponentially with the dimension of the unitary operation, and decoherence in cascaded interferometers also increases.

For convenience, we encode the four-dimensional quantum states by two-qubit states as $\{|0\rangle = |\tilde{0}\tilde{0}\rangle, |1\rangle = |\tilde{0}\tilde{1}\rangle, |2\rangle = |\tilde{1}\tilde{0}\rangle, |3\rangle = |\tilde{1}\tilde{1}\rangle\}$. The unitary transformations \mathbb{L} , \mathbb{S} , and \mathbb{R} can

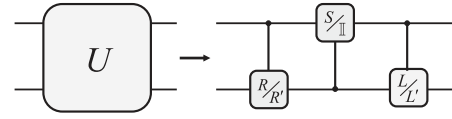


FIG. 6. The quantum circuit for implementing the 4×4 unitary transformation U on a two-qubit system.

be rewritten as

$$\begin{aligned} \mathbb{L} &= |\tilde{0}\rangle\langle\tilde{0}| \otimes L + |\tilde{1}\rangle\langle\tilde{1}| \otimes L', \\ \mathbb{S} &= S \otimes |\tilde{0}\rangle\langle\tilde{0}| + \mathbb{I} \otimes |\tilde{1}\rangle\langle\tilde{1}|, \\ \mathbb{R} &= |\tilde{0}\rangle\langle\tilde{0}| \otimes R + |\tilde{1}\rangle\langle\tilde{1}| \otimes R'. \end{aligned} \quad (30)$$

Then the 4×4 unitary transformations $U(k\Delta t)$ can be implemented by these three controlled two-qubit transformations in Fig. 6.

A schematic of our experimental setup is depicted in Fig. 7. The two qubits are encoded by spatial and polarization modes of single photons. The first qubit $|\tilde{0}\rangle$ ($|\tilde{1}\rangle$) represents the upper (lower) spatial mode of photons, and the second qubit $|\tilde{0}\rangle$ ($|\tilde{1}\rangle$) represents the horizontal (vertical) polarization of photons.

Polarization-degenerated photon pairs are generated by type I spontaneous parametric down-conversion in a 0.5-mm-thick nonlinear β -barium-borate crystal pumped by a 400.8-nm CW diode laser with 90 mW of power. The single photon is generated by triggering of the other photon. Interference filters are used to restrict the photon bandwidth to 3 nm. The photons are in horizontal polarization after the first polarizing beam splitter. The initial state is prepared in two

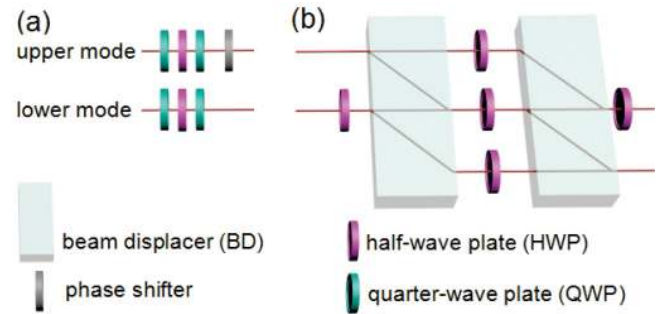


FIG. 7. Conceptual experimental setup, with three controlled unitary transformations, \mathbb{L} , \mathbb{S} , and \mathbb{R} . Red lines represent the optical modes (beams) of single photons. (a) Realization of \mathbb{L} (\mathbb{R}) as a transformation on two spatial modes and two polarization modes of single photons. The spatial mode works as the control qubit and the 2×2 unitary transformation L (R) and L' (R') applied to the polarizations of the photons in different modes can be realized by a set of wave plates. A phase shifter is used to keep the global phase unchanged during the transformation. (b) Realization of \mathbb{S} . The polarization is the control qubit. After the first HWP at 45° and a BD, the horizontally polarized photons in both spatial modes are propagating in the same spatial mode (the middle one) and then the transformation S is applied by using an HWP at $\theta/2$ in the middle mode. Meanwhile, the vertically polarized photons in upper or lower modes are not affected, and after the second BD they are still propagating in upper or low modes. The other HWPs are all set to 45° ; they are used to flip the polarizations and to change the propagating modes of photons after they pass through the BD.

steps. First, after passing through a half-wave plate (HWP_a) at 22.5° , which rotates the polarization of single photons to equal the superposition of horizontal and vertical polarizations, the photons are split into two parallel paths by a birefringent calcite BD, which transmits the vertically polarized photons directly and displaces horizontally polarized photons by 3 mm. Second, two HWPs (HWP_b and HWP_c), at -22.5° and 22.5° , are inserted into the upper and lower modes, respectively, to flip the polarizations. Thus the state of the single photons is prepared with $|\psi(0)\rangle = \frac{1}{\sqrt{2}} \sum_{j=0}^3 |j\rangle$.

For the controlled two-qubit transformations \mathbb{L} and \mathbb{R} , the spatial mode of photons serves as the control qubit and the polarization is the target qubit. In the upper and lower modes, the 2×2 unitary transformations L (R) and L' (R') are applied to the polarization degrees of freedom, which can be realized by a combination QWP and HWP sequence inserted into the corresponding spatial mode.

For the 4×4 unitary transformation \mathbb{S} , the polarization of photons serves as the control qubit. An HWP at 45° inserted into the lower input mode flips the polarization of photons. After the first BD, the horizontally polarized photons in both the upper and the lower input modes are propagating in the same path (the middle one), and then the transformation S is applied by using an HWP at $\theta/2$ in the middle path. Hence $S \otimes |\tilde{0}\rangle\langle\tilde{0}|$ is applied to $|\tilde{0}\tilde{0}\rangle$ and $|\tilde{1}\tilde{0}\rangle$. The vertically polarized photons in the upper or lower input mode are not affected and after the second BD they are still propagating in the upper or lower output modes. That is, $\mathbb{I} \otimes |\tilde{1}\rangle\langle\tilde{1}|$ is applied to $|\tilde{0}\tilde{1}\rangle$ and $|\tilde{1}\tilde{1}\rangle$. Two HWPs at 45° inserted into the other two paths (the propagating paths between two BDs) are used to flip the polarizations of photons in the paths and then the propagating modes change after the photons pass through the following BD. After the second BD, an HWP at 45° is inserted into the upper output mode to compensate for the effect of the first HWP in the lower input mode and flip the polarizations of photons back.

Our actual experimental setup is shown in Fig. 8, which takes into consideration the compensation of optical delay

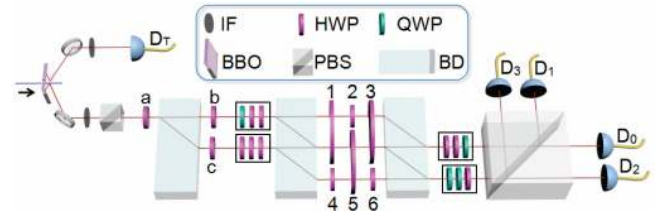


FIG. 8. Practical experimental setup with consideration of both compensation of optical delay between different spatial modes and simplification. BBO, β -barium-borate; BD, beam displacer; HWP, half-wave plate; IF, interference filter; PBS, polarizing beam splitter; QWP, quarter-wave plate.

between different spatial modes. The simplified set of wave plates for the realizations \mathbb{L} and \mathbb{R} of the eight 4×4 unitary transformations $U(k\Delta t)$ is listed in Table II, including the setting angles of wave plates. Two BDs and six HWPs (HWP_1 – HWP_6) are used to realize \mathbb{S} and compensate for the optical delay. The setting angles of HWP_1 – HWP_6 are listed in Table III.

In order to implement the proposed centrality algorithm experimentally, we discretize the CTQW time-evolution operator $U(t) = e^{-iLt}$ given by Eq. (25) for the four-vertex star graph, using eight time steps of $\Delta t = 9/40$ to ensure that we sample the probability distribution adequately over one period (note that $T = \pi/\sqrt{3} \approx 8\Delta t$). After application of the unitary time-evolution operator $U(k\Delta t)$ with $k \in \{1, 2, \dots, 8\}$, the quantum state is measured by a two-qubit projective measurement. A polarizing beam splitter is used to perform the projective measurement on the photons with the computational basis $\{|\tilde{0}\tilde{0}\rangle, |\tilde{0}\tilde{1}\rangle, |\tilde{1}\tilde{0}\rangle, |\tilde{1}\tilde{1}\rangle\}$. The photons are detected by avalanche photon diodes in coincidence with the trigger with a coincident window of 3 ns. The clicks of detectors D_0 , D_1 , D_2 , and D_3 correspond to the probabilities of the final state projected onto the basis $\{|\tilde{0}\tilde{0}\rangle, |\tilde{0}\tilde{1}\rangle, |\tilde{1}\tilde{0}\rangle, |\tilde{1}\tilde{1}\rangle\}$. We record the clicks for 5 s, and more than 18 000 coincidence counts are detected in the overall measurement time. The measured probability distributions are shown in

TABLE II. The simplified sets of wave plates with certain setting angles for realization of eight 4×4 unitary transformations. The subscript k to L , L' , R and R' corresponds to the k th unitary transformation $U(k\Delta t)$. Q, quarter-wave plate; H, half-wave plate; WP, wave plate.

L/R	WPs	Angles (deg)	L'/R'	WPs	Angles (deg)
L_1	Q-H-H	90, 0, -3.3	L'_1	H-H-H	90, 0, 157.5
R_1	H-H-Q	0, 48.4, 90	R'_1	H-Q-Q	0, 22.5, 22.5
L_2	Q-H-H	90, 0, -6.7	L'_2	H-H-H	90, 0, 157.5
R_2	H-H-Q	0, 51.7, 90	R'_2	H-Q-Q	0, 22.5, 22.5
L_3	Q-H-H	90, 0, -10.5	L'_3	H-H-H	90, 0, 157.5
R_3	H-H-Q	0, 55.5, 90	R'_3	H-Q-Q	0, 22.5, 22.5
L_4	Q-H-Q	90, 14.9, 0	L'_4	H-H-H	90, 0, 157.5
R_4	Q-H-H-Q	0, 0, 30.1, 0	R'_4	Q-Q-Q-Q	0, 0, 22.5, 22.5
L_5	Q-H-H	90, 0, -20.2	L'_5	H-H-H	90, 0, 157.5
R_5	H-H-Q	0, 65.2, 90	R'_5	H-Q-Q	0, 22.5, 22.5
L_6	Q-H	90, 0, 26.8	L'_6	H-H	90, 22.5
R_6	H-H-Q	0, -18.2 , 90	R'_6	H-Q-Q	0, 22.5, 22.5
L_7	Q-H-H	90, 0, -35.0	L'_7	Q-Q-H	0, 0, 112.5
R_7	H-Q	100.0, 90	R'_7	H-H	0, 22.5
L_8	Q-H-H-Q	90, 0, -44.4 , 0	L'_8	Q-Q-Q-Q	$-22.5, -22.5, 90, 90$
R_8	Q-H-H-Q	0, 0, 90.6, 0	R'_8	Q-Q-Q-Q	0, 0, 22.5, 22.5

TABLE III. Setting angles of HWP₁ and HWP₂ for realization of the eight 4×4 unitary transformations. The setting angles of HWP₃, HWP₄, and HWP₅ are set to be -45° , and that of HWP₆ is set to be 45° for all eight $U(k\Delta t)$ values.

	k							
	1	2	3	4	5	6	7	8
HWP ₁ (deg)	54.1	63.1	71.8	80.0	87.5	86.3	82.0	80.3
HWP ₂ (deg)	144.1	153.1	161.8	170.0	177.5	176.3	172.0	170.3

Fig. 9, which are in excellent agreement with the theoretical predictions given by Eq. (26). Here we use the norm 1 distance $d = \frac{1}{2} \sum_{x=0,1,2,3} |P^{\text{exp}}(x) - P^{\text{th}}(x)|$ to evaluate the quality of experimental demonstration. For all eight U 's, we obtain $d_1 = 0.003$, $d_2 = 0.020$, $d_3 = 0.026$, $d_4 = 0.039$, $d_5 = 0.031$, $d_6 = 0.031$, $d_7 = 0.017$, and $d_8 = 0.009$. The distances are all smaller than 0.04, which indicates successful experimental demonstrations of the 4×4 unitary transformations.

VII. CONCLUSION

In this study, we have proposed a CTQW-based quantum centrality algorithm, shown that it correlates well with classical measures, and verified its performance on general random graphs. The proposed quantum measure was then successfully implemented experimentally for a four-vertex star graph. Notably, this algorithm requires an N -dimensional Hilbert space, compared to discrete-time quantum-walk-based algorithms, which require N^2 dimensions for the same graph. Furthermore, this algorithm preserves the full quantum behavior of the walker, unlike the QSW, which mutes the quantum behavior of the walker due to decoherence.

In our physical implementation of the proposed CTQW centrality algorithm, the unitary operation of the walker on the graph is decomposed into unitary transformations in a two-dimensional subspace and realized by operating in the

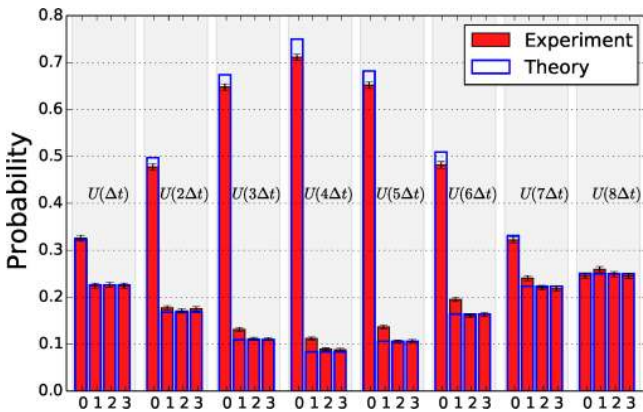


FIG. 9. Photon probability distributions after eight unitary transformations. Red bars represent experimental results. Blue borders represent theoretical predictions. Errors are estimated via propagated Poissonian statistics. A small additional uncertainty may be present in the measurement of nodes 0 and 1 due to the photon's representing the two states with different polarizations but inhibiting the same spatial mode.

polarization and spacial modes of single photons. This method can be used to decompose, in principle, any dimensional unitary operations into series of two-dimensional unitary operations. By making use of the coherent property of photons, the technology in our experiment is a competitive candidate for demonstrating arbitrary unitary operations, allowing it to be utilized for a wide array of quantum algorithms and quantum information processes. This paper reports the successful physical demonstration of a quantum centrality algorithm on a four-vertex star graph.

ACKNOWLEDGMENTS

This work was supported by NSFC (Grants No. 11474049 and No. 11674056), NSFJIS (Grant No. BK20160024), and the Scientific Research Foundation of the Graduate School of Southeast University (Grant No. YBJJ1623). J.A.I. would like to thank the Hackett Foundation and The University of Western Australia for financial support. This work was also supported by the Pawsey centre in Western Australia through the use of its advanced computing resources.

J.A.I. and X.Z. contributed equally to this work.

APPENDIX: EXPLICIT EXPRESSIONS FOR L , R , L' , AND R'

For clarity, we provide here the explicit expressions for L_k , R_k , L'_k , and R'_k , where the subscript k corresponds to the k th unitary transformation $U(k\Delta t)$:

$$L_1 = \begin{bmatrix} -0.9936i & -0.1132i \\ -0.1132 & 0.9936 \end{bmatrix}, \quad L'_1 = \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix},$$

$$R_1 = \begin{bmatrix} 0.9936i & 0.1132 \\ -0.1132i & 0.9936 \end{bmatrix}, \quad R'_1 = \begin{bmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{bmatrix},$$

$$L_2 = \begin{bmatrix} -0.9730i & -0.2307i \\ -0.2307 & 0.9730 \end{bmatrix}, \quad L'_2 = L'_1,$$

$$R_2 = \begin{bmatrix} 0.9730i & 0.2307 \\ -0.2307i & 0.9730 \end{bmatrix}, \quad R'_2 = R'_1,$$

$$L_3 = \begin{bmatrix} -0.9341i & -0.3569i \\ -0.3569 & 0.9341 \end{bmatrix}, \quad L'_3 = L'_1,$$

$$R_3 = \begin{bmatrix} 0.9341i & 0.3569 \\ -0.3569i & 0.9341 \end{bmatrix}, \quad R'_3 = R'_1,$$

$$L_4 = \begin{bmatrix} -0.8686i & 0.4955 \\ -0.4955 & 0.8686i \end{bmatrix}, \quad L'_4 = L'_1,$$

$$R_4 = \begin{bmatrix} 0.8686i & 0.4955 \\ -0.4955 & -0.8686i \end{bmatrix}, \quad R'_4 = R'_1,$$

$$L_5 = \begin{bmatrix} -0.7618i & -0.6478i \\ -0.6478 & 0.7618 \end{bmatrix}, \quad L'_5 = L'_1,$$

$$R_5 = \begin{bmatrix} 0.7618i & 0.6478 \\ -0.6478i & 0.7618 \end{bmatrix}, \quad R'_5 = R'_1,$$

$$L_6 = \begin{bmatrix} 0.5926i & 0.8055i \\ 0.8055 & -0.5926 \end{bmatrix}, \quad L'_6 = L'_1,$$

$$R_6 = \begin{bmatrix} 0.5926i & 0.8055 \\ 0.8055i & -0.5926 \end{bmatrix}, \quad R'_6 = \begin{bmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{bmatrix},$$

$$L_7 = \begin{bmatrix} 0.3415i & -0.9399i \\ 0.9399 & 0.3415 \end{bmatrix}, \quad L'_7 = L'_1,$$

$$R_7 = \begin{bmatrix} 0.3415i & 0.9399 \\ -0.9399i & 0.3415 \end{bmatrix}, \quad R'_7 = R'_6,$$

$$L_8 = \begin{bmatrix} 0.0207i & -0.9998 \\ 0.9998 & -0.0207i \end{bmatrix}, \quad L'_8 = L'_1,$$

$$R_8 = \begin{bmatrix} 0.0207i & 0.9998 \\ 0.9998 & 0.0207i \end{bmatrix}, \quad R'_8 = R'_6.$$

-
- [1] Y. Aharonov, L. Davidovich, and N. Zagury, *Phys. Rev. A* **48**, 1687 (1993).
- [2] J. Kempe, *Contemp. Phys.* **44**, 307 (2003).
- [3] A. M. Childs and J. Goldstone, *Phys. Rev. A* **70**, 022314 (2004).
- [4] N. B. Lovett, S. Cooper, M. Everitt, M. Trevers, and V. Kendon, *Phys. Rev. A* **81**, 042330 (2010).
- [5] S. E. Venegas-Andraca, *Quant. Info. Proc.* **11**, 1015 (2012).
- [6] K. Manouchehri and J. B. Wang, *Physical Implementation of Quantum Walks* (Springer-Verlag, Berlin, 2014).
- [7] Z. J. Li and J. B. Wang, *J. Phys. A: Math. Theor.* **48**, 355301 (2015).
- [8] I. Martínez-Martínez and E. Sánchez-Burillo, *Sci. Rep.* **6**, 23812 (2016).
- [9] S. D. Berry and J. B. Wang, *Phys. Rev. A* **83**, 042317 (2011).
- [10] B. L. Douglas and J. B. Wang, *J. Phys. A: Math. Theor.* **41**, 075303 (2008).
- [11] M. Sarovar, A. Ishizaki, G. R. Fleming, and K. B. Whaley, *Nat. Phys.* **6**, 462 (2010).
- [12] P. Rebentrost, M. Mohseni, I. Kassal, S. Lloyd, and A. Aspuru-Guzik, *New J. Phys.* **11**, 033003 (2009).
- [13] G. S. Engel, T. R. Calhoun, E. L. Read, T. Ahn, T. Mančal, Y. Cheng, R. E. Blankenship, and G. R. Fleming, *Nature* **446**, 782 (2007).
- [14] J. A. Izaac, J. B. Wang, and Z. J. Li, *Phys. Rev. A* **88**, 042334 (2013).
- [15] A. M. Childs, *Phys. Rev. Lett.* **102**, 180501 (2009).
- [16] A. M. Childs, D. Gosset, and Z. Webb, *Science* **339**, 791 (2013).
- [17] X. Qiang, T. Loke, A. Montanaro, K. Aungkunsiri, X. Zhou, J. L. O'Brien, J. B. Wang, and J. C. Matthews, *Nat. Commun.* **7**, 11511 (2016).
- [18] E. Farhi and S. Gutmann, *Phys. Rev. A* **58**, 915 (1998).
- [19] F. W. Strauch, *Phys. Rev. A* **74**, 030301 (2006).
- [20] P. Philipp and R. Portugal, *Quant. Infor. Proc.* **16**, 14 (2017).
- [21] S. D. Berry and J. B. Wang, *Phys. Rev. A* **82**, 042333 (2010).
- [22] G. D. Paparo and M. A. Martin-Delgado, *Sci. Rep.* **2**, 00444 (2012).
- [23] G. D. Paparo, M. Müller, F. Comellas, and M. A. Martin-Delgado, *Sci. Rep.* **3**, 2773 (2013).
- [24] T. Loke, J. W. Tang, J. Rodriguez, M. Small, and J. B. Wang, *Quant. Info. Proc.* **16**, 25 (2017).
- [25] I. Sinayskiy and F. Petruccione, *J. Phys.: Conf. Ser.* **442**, 012003 (2013).
- [26] P. E. Falloon, J. Rodriguez, and J. B. Wang, [arXiv:1606.04974](https://arxiv.org/abs/1606.04974) [quant-ph].
- [27] A. M. Childs, *Commun. Math. Phys.* **294**, 581 (2010).
- [28] M. Szegedy, in *45th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, New York, 2004), pp. 32–41.
- [29] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, in *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing, STOC '03* (Association for Computing Machinery, New York, 2003), pp. 59–68.
- [30] J. D. Whitfield, C. A. Rodríguez-Rosario, and A. Aspuru-Guzik, *Phys. Rev. A* **81**, 022323 (2010).
- [31] T. G. Wong, L. Tarrataca, and N. Nahimov, *Quant. Info. Proc.* **15**, 4029 (2016).
- [32] S. Brin and L. Page, *Comput. Networks ISDN Syst.* **30**, 107 (1998).
- [33] D. Knoke and R. S. Burt, in *Applied Network Analysis* (Sage, Newbury Park, CA, 1983), pp. 195–222.
- [34] D. J. Brass, *Admin. Sci. Q* **29**, 518 (1984).
- [35] H. Ibarra and S. B. Andrews, *Admin. Sci. Q* **38**, 277 (1993).
- [36] H. Ibarra, *Acad. Manage. J.* **36**, 471 (1993).
- [37] D. S. Sade, *Soc. Networks (Special Issue)* **11**, 273 (1989).
- [38] G. Amitai, A. Shemesh, E. Sitbon, M. Shklar, D. Netanel, I. Venger, and S. Pietrokovski, *J. Mol. Biol.* **344**, 1135 (2004).
- [39] S. P. Borgatti, *Soc. Networks* **27**, 55 (2005).
- [40] R. B. Rothenberg, J. J. Potterat, D. E. Woodhouse, W. W. Darrow, S. Q. Muth, and A. S. Klovdahl, *Soc. Networks* **17**, 273 (1995).
- [41] T. J. Grosser, V. Lopez-Kidwell, and G. Labianca, *Group Org. Manage.* **35**, 177 (2010).
- [42] S. P. Borgatti and M. G. Everett, *Soc. Networks* **28**, 466 (2006).
- [43] P. Bonacich, *Am. J. Sociol.* **92**, 1170 (1987).
- [44] D. F. Gleich and K. Kloster, *Eur. J. Appl. Math.* **27**, 812 (2016).
- [45] J. D. Noh and H. Rieger, *Phys. Rev. Lett.* **92**, 118701 (2004).
- [46] L. C. Freeman, *Soc. Networks* **1**, 215 (1978).
- [47] P. Erdős and A. Rényi, *Publ. Math. (Debrecen)* **6**, 290 (1959).
- [48] P. Erdős and A. Rényi, in *Publication of the Mathematical Institute of the Hungarian Academy of Sciences* (Hungarian Academy of Sciences, Budapest, 1960), pp. 17–61.
- [49] A. Barabási and R. Albert, *Science* **286**, 509 (1999).
- [50] A. Barabási, R. Albert, and H. Jeong, *Physica A: Stat. Mech. Appl.* **281**, 69 (2000).
- [51] R. Albert and A. Barabási, *Rev. Mod. Phys.* **74**, 47 (2002).
- [52] C. Song, S. Havlin, and H. A. Makse, *Nature* **433**, 392 (2005).
- [53] G. Lawyer, *Sci. Rep.* **5**, 8665 (2015).
- [54] M. G. Kendall, *Biometrika* **30**, 81 (1938).
- [55] J. Liu, Z. Ren, and Q. Guo, *Physica A: Stat. Mech. Appl.* **392**, 4154 (2013).
- [56] D. W. Berry, A. M. Childs, and R. Kothari, in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, New York, 2015), pp. 792–809.
- [57] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, *Phys. Rev. Lett.* **73**, 58 (1994).
- [58] G. Stewart, *SIAM Rev.* **19**, 634 (1977).
- [59] G. W. Stewart, *Numer. Math.* **40**, 297 (1982).
- [60] C. C. Paige and M. Wei, *Lin. Algebra Appl.* **208**, 303 (1994).
- [61] B. D. Sutton, *Numer. Algorithms* **50**, 33 (2008).
- [62] I. Dhand and S. K. Goyal, *Phys. Rev. A* **92**, 043813 (2015).
- [63] S. K. Goyal, F. S. Roux, A. Forbes, and T. Konrad, *Phys. Rev. A* **92**, 040302 (2015).

Reconfigurable magnetic domain wall pinning using vortex-generated magnetic fields

Aaron C. H. Hurst,¹ Joshua A. Izaac,¹ Fouzia Altaf,¹ Vincent Baltz,² and Peter J. Metaxas^{1,a)}

¹*School of Physics and Astrophysics, M013, University of Western Australia, 35 Stirling Hwy, Crawley, WA 6009, Australia*

²*SPINTEC, Univ. Grenoble Alpes/CNRS/INAC-CEA, F-38000 Grenoble, France*

(Received 8 February 2017; accepted 13 April 2017; published online 2 May 2017)

Although often important for domain wall device applications, reproducible fabrication of pinning sites at the nano-scale remains challenging. Here, we demonstrate that the stray magnetic field generated beneath magnetic vortex cores can be used to generate localized pinning sites for magnetic domain walls in an underlying, perpendicularly magnetized nanostrip. Moreover, we show that the pinning strength can be tuned by switching the vortex core polarity: switching the core polarity so that it is aligned with the magnetization of the expanding domain (rather than against it) can reduce the vortex-mediated wall depinning field by between 40% and 90%, depending on the system geometry. Significant reductions in the depinning field are also demonstrated in narrow strips by shifting the core away from the strips' centers. *Published by AIP Publishing.*

[<http://dx.doi.org/10.1063/1.4982237>]

Control of magnetic domain wall (DW) motion is critical for the realization of DW devices such as logic circuits and (multi-state) memories.^{1–6} Such devices are typically based upon ferromagnetic strips through which DWs propagate. In strips, stable positions for DWs have traditionally been achieved by structural modifications, often via the creation of local constrictions or protrusions⁷ (local modifications to anisotropy have also been employed, e.g., Ref. 8). However, uniform fabrication of structural pinning sites at the nano-scale can be challenging. Another challenge is the need to balance the positional stability afforded by strong pinning sites (critical for data endurance) with a technological requirement for low power domain wall motion/depinning.⁹

A potential solution to the latter challenge is to use pinning sites whose strength can be dynamically modified, enabling, for example, the pinning strength to be (momentarily) reduced when the domain wall position has to be changed. Voltage-induced domain wall gating is one potential route to realize such pinning.^{11,12} However, reconfigurable pinning can also be generated using stray magnetic fields from neighboring ferromagnetic elements.^{13–18} Using ferromagnetic elements in this way also enables *nonvolatile* reconfigurability since the magnetic state of the element can be switched, modifying the stray field profile and thus the strength of the resulting pinning site.

Previous studies of stray-field-mediated pinning have focused primarily on the use of (quasi-)uniformly magnetized, bistable ferromagnetic elements.^{4,13,15,18} In this letter, we use micromagnetic simulation to demonstrate the potential of magnetic vortices^{19–21} for DW pinning (see also Ref. 22). Like DWs,²³ magnetic vortices are inherently reconfigurable and non-uniform magnetization configurations. They can exist within ferromagnetic disks and consist of an in-plane curling magnetization with an out of plane magnetized nano-scale core. Here, we use the highly localized stray magnetic field which exists around the core (and which is parallel to the core magnetization^{20,21,24}) to pin domain walls

in an underlying strip. We show that by switching the core magnetization (which is perpendicular to the disk plane and can be (rapidly) switched^{25–29}), it is possible to toggle between strong and weak DW pinning sites. However, we also show that the pinning strength can also be varied by shifting the core away from the center of the strip, adding an additional degree of tunability without requiring a reversal of the core magnetization. Finally, we note that this approach to pinning enables one to create a pinning site with a reliable characteristic width and strength without the need to accurately and reproducibly fabricate similar nano-scaled structural features. This is because the core size is largely independent of the disk width (for those disks which exhibit a single vortex state: ≈ 100 nm up to the micrometer scale).³⁰

Vortex-mediated DW pinning was studied in a 3 nm thick strip with perpendicular magnetic anisotropy. Two vortex-containing disks with a diameter of 192 nm and a thickness of 12 nm lie above the strip [Fig. 1(a)]. There is a vertical separation, d , between the bottom of the disks and the top of the strip [Fig. 1(b)]. The 768 nm long strip is CoPtCr-like³¹ with saturation magnetization $M_S = 300$ kA/m, out-of-plane uniaxial anisotropy $K = 0.2$ MJ/m³, and exchange stiffness $A_{ex} = 10$ pJ/m. The disks are NiFe-like with $M_S = 860$ kA/m, $A_{ex} = 13$ pJ/m, and negligible intrinsic anisotropy. The damping parameter

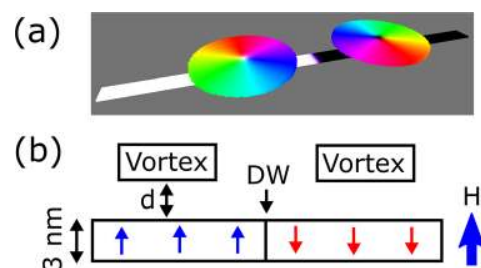


FIG. 1. System schematics. (a) 2 vortices above a DW-containing strip in the zero external field (visualized using Muvview2¹⁰). (b) Side view schematic (not to scale) showing the finite separation, d , between the lower surface of the disks and the upper surface of the strip.

^{a)}Electronic mail: peter.metaxas@uwa.edu.au

is 1 everywhere since we consider relaxed states only. No temperature effects were included nor was inter-element exchange coupling, meaning that the disk-layer interaction is purely magnetostatic. The disks are symmetrically spaced either side of the center of the strip along the strip's long axis. The majority of the presented data has been obtained with MuMax3³² using $3 \times 3 \times 3 \text{ nm}^3$ discretization cells. Good consistency with OOMMF³³ results was also found, which is mentioned below. For both approaches, the system was initialized with a DW-like transition at the middle of the strip and trial vortex configurations in the disks (the vortex chiralities are anti-clockwise unless otherwise specified). The magnetization was then allowed to relax in the zero external field using MuMax3's "minimize" routine (or OOMMF's "relax" routine). Both routines employ a conjugate gradient method to find the ground state magnetization. From this point, the simulation was progressed in a quasi-static manner. An external magnetic field, $H_{ext} = (0, 0, H)$ [Fig. 1(b)], was applied and increased in steps of ΔH starting at the zero field. The minimize/relax routine was run at each field value, starting from the relaxed configuration obtained at the previous field. The system's relaxed configuration was recorded after each step. Note that H was always positive, driving the DW toward the right hand disk [i.e., expanding the left hand domain; Fig. 1(b)].

We will concentrate first on pinning induced by a vortex which has a negative core polarity, $p = -1$. Directly beneath the $p = -1$ core, there is a significant negative out-of-plane field, H_{vortex}^z [Fig. 2(a)]. This field "peak" has a width (full width half maximum) on the order of 20 nm and results in there being a localized region in the strip where there is a negative H_{vortex}^z which is strong and opposes the positive H that is used to drive the DW displacement. This localized field acts as a barrier to DW motion, as can be seen in Fig. 2(b) which shows a snapshot of a DW being pinned just to the left of the core center ($w = 48 \text{ nm}$).

In Fig. 2(c), we follow the normalized z -component of the magnetization within a 36 nm wide strip ($d = 21 \text{ nm}$) as H is increased by steps, ΔH , enabling us to pass from the pinned state to the depinned state. At very low external fields ($\mu_0 H < 1 \text{ mT}$), the DW moves towards the disk where it is pinned near the vortex core [as in Fig. 2(b)]. At the depinning field, H_{dep} , the wall depins, moves past the vortex, and annihilates at the end of the strip. Note that while the wall is pinned at the core, the x -component of the magnetization within the vortex increases [also shown in Fig. 2(c)]. This increase corresponds to a shift of the core in the $+y$ direction [Fig. 2(b)] and is driven by the in-plane component of the stray magnetic field ($+x$ -oriented) which exists above the domain wall^{34–37} [Fig. 2(d)]. This field acts on the vortex's curling magnetization [Fig. 2(e)], inducing a core shift. The direction of this shift depends on the vortex chirality [Figs. 2(f) and 2(g)] since the chirality determines the at-equilibrium core displacement for a given in-plane magnetic field.³⁸

We now look at the dependence of the depinning field on the system geometry for the same $p = -1$ vortex configuration. In Figs. 3(a) and 3(b), we show that increasing the strip width, w , or the disk-strip separation, d , will reduce H_{dep} . The latter d -dependence occurs because as d is

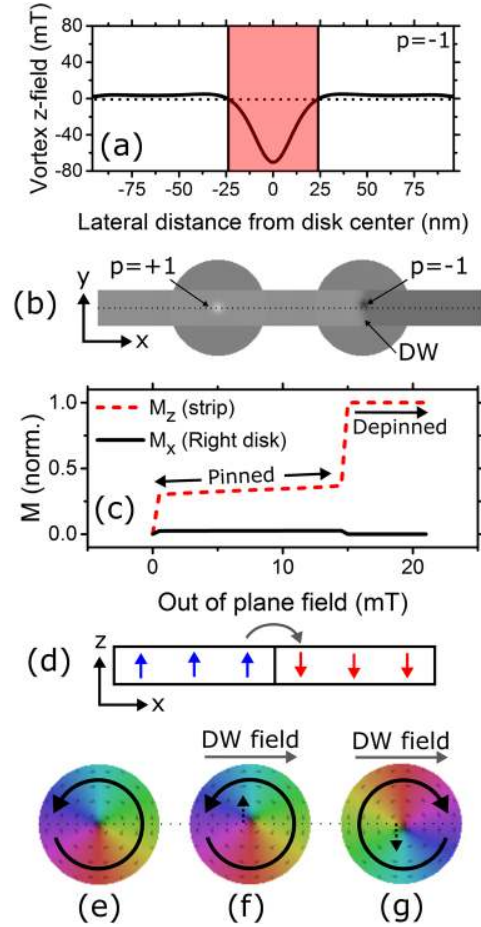


FIG. 2. (a) z -component of the stray field generated by a $p = -1$ vortex, H_{vortex}^z , calculated 10.5 nm below the vortex-containing disk. The (red) shaded region shows where $H_{vortex}^z < 0$ and thus opposes H , generating DW pinning. (b) Bottom-up visualization of a pinned DW ($w = 72 \text{ nm}$) with gray level mapping of the out-of-plane magnetization component. The DW is at the right hand disk, pinned at the down-magnetized ($p = -1$) vortex core (its position is visible as a dark gray dot). (c) M_z in a $w = 36 \text{ nm}$ strip ($d = 21 \text{ nm}$) and M_x in the right hand disk (i.e., in the disk at which the DW becomes pinned) as H is increased in steps. (d) An in-plane stray field (arrow) is generated above the DW. (e) Vortex state with a centered core (i.e., negligible in-plane field). The circular arrow indicates the chirality. When the DW is pinned at the vortex, the DW's in-plane stray field (gray arrow) shifts the core in the (f) $+y$ direction for an anti-clockwise vortex or (g) the $-y$ direction for a clockwise vortex. The black dotted line across (e)–(g) marks the position of the unshifted core, and the small dotted arrows in (f) and (g) show the core shift direction.

increased, the strip will be further from the disk and thus subject to a weaker H_{vortex}^z . Since the pinning is H_{vortex}^z -mediated, this leads to a smaller H_{dep} . The w -dependence can be understood as follows: For very narrow strips, the entire strip width is subject to a strong, negative H_{vortex}^z [as can be inferred from Fig. 2(a)], which generates a high H_{dep} . In contrast, for a wide strip, only the central portion of the strip will be subject to the strong H_{vortex}^z which exists directly below the core. As such, the average H_{vortex}^z acting across the strip is lower, leading to a reduced H_{dep} . Indeed, H_{dep} can be shown to closely match the width-averaged H_{vortex}^z both when varying d [Fig. 3(c); $w = 36 \text{ nm}$] and w [Fig. 3(d); $d = 9 \text{ nm}$]. Note that H_{vortex}^z was calculated in each case for an isolated vortex subject to $H_{dep} - \Delta H$ (i.e., at the field preceding that which generated depinning). We also note that the above results were checked with OOMMF³³ for $d = 9$

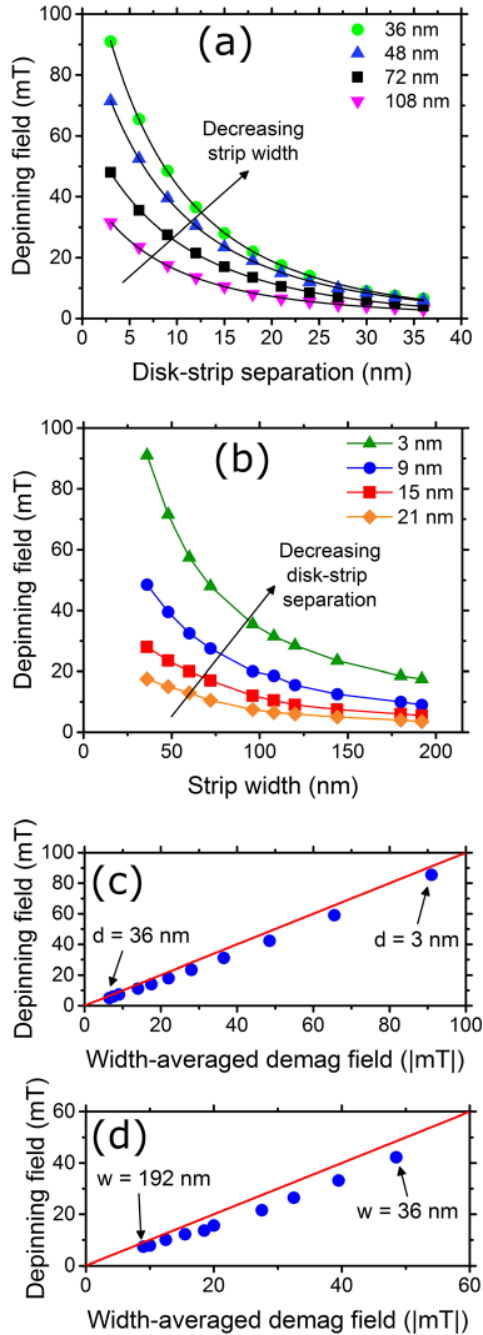


FIG. 3. $p = -1$ depinning field versus (a) disk-strip separation, d , and (b) strip width, w , for various values of w and d . Black lines in (a) are fits of the form $a + \frac{b}{(c+d)}$. Lines in (b) are guides to the eye. Depinning fields for various values of (c) d ($w = 36$ nm) and (d) w ($d = 9$ nm) plotted against the absolute value of the w -averaged z -component of an isolated vortex's stray magnetic field in the region of the DW. The latter was calculated for an isolated disk in the presence of an out of plane field with magnitude $H = H_{\text{dep}} - \Delta H$ (i.e., the field preceding depinning for that particular geometry).

and 15 nm which yielded agreement with the MuMax3-calculated H_{dep} values to within 1 mT. Smaller discretization lengths were also examined using MuMax3 as a further check where reducing the in-plane discretization from 3 nm to 2 nm was found to result in a weak change in the depinning fields (+2.9%).

Switching the polarity of the core to $p = +1$ (achieved by changing the polarity of the pre-relaxation trial vortex state) switches the sign of the vortex stray field, resulting in

H_{vortex}^z now being negative (and thus opposing DW motion) in the regions *surrounding* the core position (rather than the region directly beneath the core) [Fig. 4(a); compare to Fig. 2(a) which shows the stray field for $p = -1$]. As a result, the DW now becomes pinned well before it reaches the core [Fig. 4(b)] rather than being pinned at the core location [Fig. 2(b)]. The part of the H_{vortex}^z profile causing pinning for $p = +1$ [Fig. 4(a)] is also much weaker than the part of the profile directly beneath the vortex core, which generates pinning for the $p = -1$ case. This results in H_{dep} being reduced for $p = +1$. This can be seen in Fig. 4(c) where H_{dep} has been plotted versus d for both core polarities ($w = 36$ nm). Switching the core polarity (which can be achieved rapidly for quick gate modification^{25,26,28,29}) reduces the depinning field by $\approx 80\%$ over a wide range of d [Fig. 4(d)]. As can be seen in Fig. 4(e), however, H_{dep} for $p = +1$ is almost independent of w (tested for $d = 9$ nm). This is consistent with the weak lateral spatial gradient in H_{vortex}^z in the region away from the core [i.e., the part of the vortex-generated field distribution which generates pinning is relatively uniform in space; Fig. 4(a)]. As a result, the %-reduction in H_{dep} when

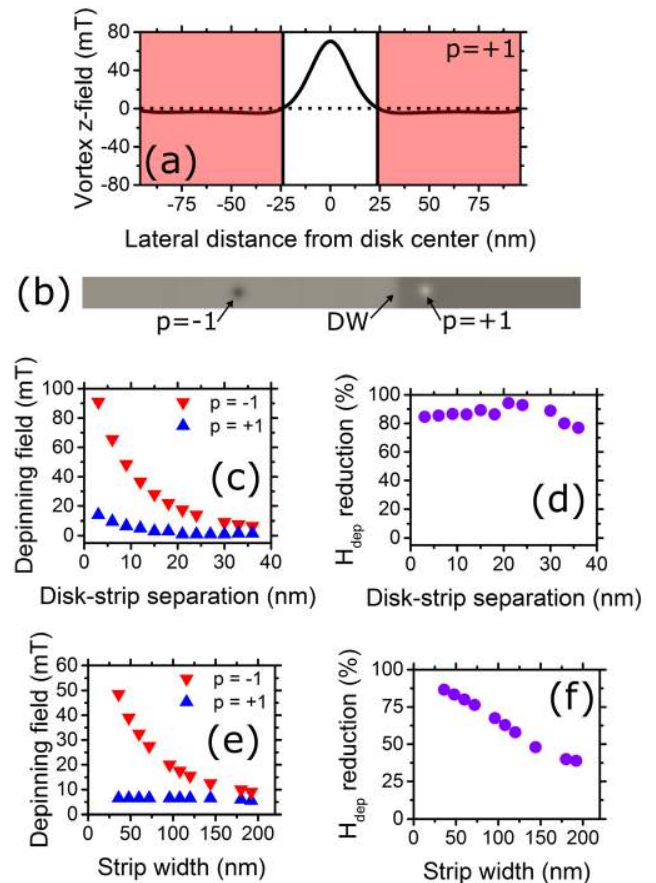


FIG. 4. (a) z -component of the stray field generated by a $p = +1$ vortex, H_{vortex}^z , calculated 10.5 nm below the vortex-containing disc. The (red) shaded region shows where $H_{\text{vortex}}^z < 0$ and thus opposes H , generating DW pinning. (b) Visualization of a pinned DW with gray level mapping of the out-of-plane magnetization component and superposition of the core magnetizations (gray and white dots). The DW is at the right hand disk, pinned well to the left of the right hand vortex core. Lower panels show comparisons of depinning fields for polarities opposing ($p = -1$) and aligned with ($p = +1$) the external field versus (c) d ($w = 36$ nm) and (e) w ($d = 9$ nm). Corresponding percentage reductions in the depinning fields by switching from $p = +1$ to $p = -1$ are shown in (d) and (f).

going from $p = -1$ to $p = +1$ is largest for small w [Fig. 1(f)] since H_{dep} for $p = -1$ becomes weak at high w due to the averaging effect discussed above [Fig. 3(d)].

We now examine the effect that shifting the core has on domain wall depinning ($p = -1$). To induce a core shift, we apply an in-plane magnetic field which is constant in time, H_{IP} . H_{IP} is aligned along the strip's long axis ($+x$), acting to shift the core perpendicular to that axis ($+y$). H_{dep} versus H_{IP} is plotted for four strip widths in Fig. 5 together with a visualization of the out-of-plane component of the disk's stray field ($\mu_0 H_{\text{IP}} = 21$ mT).

In terms of the H_{IP} -dependence of H_{dep} , two clear regimes are seen for the three narrowest strips. For low H_{IP} (below 12 mT in Fig. 5), H_{dep} initially decreases with the in-plane field. In this regime, shifting the core away from the narrow strips' centers reduces the average H_{vortex}^z that acts on the strip, thereby reducing H_{dep} . The initial drop-off in H_{dep} is the steepest for the narrower strips since the core-displacement-driven change in the strip-width-averaged H_{vortex}^z for small core displacements will be the highest. At higher H_{IP} , H_{dep} begins to increase with H_{IP} . In this regime, the core becomes pinned at the vortex core and then again at the disk's right edge. This second pinning event arises due to the $+x$ in-plane field transitioning the disk toward a $+x$ -magnetized state. This induces edge magnetic charges on the $\pm x$ sides of the disk which generate out-of-plane stray fields that act on the strip (inset of Fig. 5). The out-of-plane components of the edge fields are negative on the disk's right hand side and can thus pin the domain wall. Furthermore, the strength of these edge fields grows with H_{IP} as the disk acquires a higher $+x$ magnetization. The edge pinning mechanism dominates core-induced pinning at large H_{IP} , leading to an increasing H_{dep} . Note that compared to the core stray field, these edge fields are relatively uniform in the y -direction (Fig. 5), which results in similar H_{dep} values for each strip width. Finally, we note that H_{dep} in the 192 nm strip exhibits a monotonically increasing trend with the in-plane field, which is similar to that seen for the narrow strips at

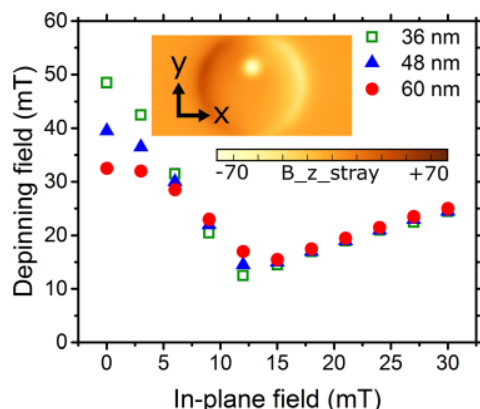


FIG. 5. Depinning field versus in-plane field for four strip widths at $d = 9$ nm. The field is applied in the $+x$ direction, shifting the core in the $+y$ direction. The inset shows the z -component of the stray field below a shifted vortex core ($p = -1$) in an isolated disk, as calculated at the vertical center of the strip region for $d = 9$ nm in a $+x$ -oriented in-plane magnetic field of $+21$ mT (color scale bar for the out-of-plane field is also shown in mT). The position of the $+y$ shifted core can be identified by its localized negative field. Stray fields are also generated below the disk's $\pm x$ edges.

high fields where depinning requires the wall to move past the edge-charge-generated pinning. Indeed, from 6 mT onwards, in the $w = 192$ nm strip, pinning due to edge charges dominates the core-mediated pinning, the latter being intrinsically weak at large w [Fig. 3(b)].

In summary, this work demonstrates the potential to use localized stray magnetic fields generated by vortex states to reliably generate reconfigurable, nano-scale domain wall pinning sites in an underlying perpendicularly magnetized ferromagnetic strip. The domain wall pinning strength can be tuned by switching the vortex core polarity or by shifting the core across the strip (done here via in-plane magnetic fields). This approach enables the development of controllable gates for domain wall motion and logic. These findings also open possibilities for further investigations on harnessing magnetostatic interactions between skyrmions³⁹ and domain walls (or skyrmions and vortices). Extensions to current-induced manipulation of magnetic textures may also be envisaged. There will also be interesting effects to examine full dynamic simulations where one can aim to (e.g.) rapidly toggle the pinning strength (via a switch in the core magnetization or a well-timed core displacement) or dynamically study core displacements (or core-polarity-switching⁴⁰) driven by domain-wall-vortex interactions.

This work was supported by resources provided by the Pawsey Supercomputing Centre with funding from the Australian Government and the Government of Western Australia. The preliminary work was supported by iVEC through the use of advanced computing resources located at iVEC@UWA. P.J.M. acknowledges support from the Australian Research Council's Discovery Early Career Researcher Award scheme (Grant No. DE120100155) and the University of Western Australia (Research Development Award and Early Career Researcher Fellowship Support schemes). A.C.H.H. and J.A.I. were supported by internships from the Pawsey Supercomputing Centre and iVEC, respectively. The authors thank H. Fangohr, M. Albert, R. L. Novak, M. Kostylev, and J. P. Fried for useful discussions.

¹D. A. Allwood, G. Xiong, C. C. Faulkner, D. Atkinson, D. Petit, and R. P. Cowburn, *Science* **309**, 1688 (2005).

²S. Fukami, T. Suzuki, K. Nagahara, N. Ohshima, Y. Ozaki, S. Saito, R. Nebashi, N. Sakimura, H. Honjo, K. Mori, C. Igarashi, S. Miura, N. Ishiwata, and T. Sugibayashi, in 2009 Symposium on VLSI Technology (2009), p. 230.

³J. Jaworowicz, N. Vernier, J. Ferré, A. Maziewski, D. Stanescu, D. Ravelosona, A. S. Jacqueline, C. Chappert, B. Rodmacq, and B. Diény, *Nanotechnology* **20**, 215401 (2009).

⁴S. Breitkreutz, I. Eichwald, J. Kiermaier, G. Hübner, G. Csaba, W. Porod, D. Schmitt-Landsiedel, and M. Becherer, *J. Appl. Phys.* **115**, 17D506 (2014).

⁵S. Lequeux, J. Sampaio, V. Cros, K. Yakushiji, A. Fukushima, R. Matsumoto, H. Kubota, S. Yuasa, and J. Grollier, *Sci. Rep.* **6**, 31510 (2016).

⁶J. A. Currihan-Incorvia, S. Siddiqui, S. Dutta, E. R. Evarts, J. Zhang, D. Bono, C. A. Ross, and M. A. Baldo, *Nat. Commun.* **7**, 10275 (2016).

⁷O. Boulle, G. Malinowski, and M. Kläui, *Mater. Sci. Eng., R* **72**, 159 (2011).

⁸J. H. Franken, H. J. M. Swagten, and B. Koopmans, *Nat. Nanotechnol.* **7**, 499 (2012).

⁹S. S. P. Parkin, M. Hayashi, and L. Thomas, *Science* **320**, 190 (2008).

- ¹⁰G. Rowlands, see <https://github.com/grahamrow/muview2> for “Muview2 micromagnetic viewer”.
- ¹¹J. H. Franken, Y. Yin, A. J. Schellekens, A. van den Brink, H. J. M. Swagten, and B. Koopmans, *Appl. Phys. Lett.* **103**, 102411 (2013).
- ¹²A. Bernard-Mantel, L. Herrera-Diez, L. Ranno, S. Pizzini, J. Vogel, D. Givord, S. Auffret, O. Boulle, I. M. Miron, and G. Gaudin, *Appl. Phys. Lett.* **102**, 122406 (2013).
- ¹³P. J. Metaxas, P. J. Zermatten, J. P. Jamet, J. Ferré, G. Gaudin, B. Rodmacq, A. Schuhl, and R. L. Stamps, *Appl. Phys. Lett.* **94**, 132504 (2009).
- ¹⁴L. O’Brien, D. Petit, E. R. Lewis, R. P. Cowburn, D. E. Read, J. Sampaio, H. T. Zeng, and A. V. Jausovec, *Phys. Rev. Lett.* **106**, 087204 (2011).
- ¹⁵R. Hiramatsu, T. Koyama, H. Hata, T. Ono, D. Chiba, S. Fukami, and N. Ishiwata, *J. Korean Phys. Soc.* **63**, 608 (2013).
- ¹⁶P. J. Metaxas, P. J. Zermatten, R. L. Novak, S. Rohart, J. P. Jamet, R. Weil, J. Ferré, A. Mougin, R. L. Stamps, G. Gaudin, V. Baltz, and B. Rodmacq, *J. Appl. Phys.* **113**, 073906 (2013).
- ¹⁷J. H. Franken, M. A. J. van der Heijden, T. H. Ellis, R. Lavrijsen, C. Daniels, D. McGrouther, H. J. M. Swagten, and B. Koopmans, *Adv. Funct. Mater.* **24**, 3508 (2014).
- ¹⁸R. A. van Mourik, C. T. Rettner, B. Koopmans, and S. S. P. Parkin, *J. Appl. Phys.* **115**, 17D503 (2014).
- ¹⁹R. P. Cowburn, D. K. Koltsov, A. O. Adeyeye, and M. E. Welland, *Phys. Rev. Lett.* **83**, 1042 (1999).
- ²⁰T. Shinjo, T. Okuno, R. Hassdorf, K. Shigeto, and T. Ono, *Science* **289**, 930 (2000).
- ²¹A. Wachowiak, J. Wiebe, M. Bode, O. Pietzsch, M. Morgenstern, and R. Wiesendanger, *Science* **298**, 577 (2002).
- ²²R. L. Novak and L. C. Sampaio, e-print [arXiv:1702.02451](https://arxiv.org/abs/1702.02451).
- ²³L. O’Brien, D. Petit, H. T. Zeng, E. R. Lewis, J. Sampaio, A. V. Jausovec, D. E. Read, and R. P. Cowburn, *Phys. Rev. Lett.* **103**, 077206 (2009).
- ²⁴L. Rondin, J. P. Tetienne, S. Rohart, A. Thiaville, T. Hingant, P. Spinicelli, J. F. Roch, and V. Jacques, *Nat. Commun.* **4**, 2279 (2013).
- ²⁵K. Yamada, S. Kasai, Y. Nakatani, K. Kobayashi, and T. Ono, *Appl. Phys. Lett.* **93**, 152502 (2008).
- ²⁶M. Noske, A. Gangwar, H. Stoll, M. Kammerer, M. Sproll, G. Dieterle, M. Weigand, M. Fähnle, G. Woltersdorf, C. H. Back, and G. Schütz, *Phys. Rev. B* **90**, 104415 (2014).
- ²⁷T. Okuno, K. Shigeto, T. Ono, K. Mibu, and T. Shinjo, *J. Magn. Magn. Mater* **240**, 1 (2002).
- ²⁸B. Van Waeyenberge, A. Puzic, H. Stoll, K. W. Chou, T. Tylliszczak, R. Hertel, M. Fähnle, H. Brückl, K. Rott, G. Reiss, I. Neudecker, D. Weiss, C. H. Back, and G. Schütz, *Nature* **444**, 461 (2006).
- ²⁹K. Yamada, S. Kasai, Y. Nakatani, K. Kobayashi, H. Kohno, A. Thiaville, and T. Ono, *Nat. Mater.* **6**, 270 (2007).
- ³⁰N. Usov and S. Peschany, *J. Magn. Magn. Mater.* **118**, L290 (1993).
- ³¹E. Martinez, L. Torres, and L. Lopez-Diaz, *Phys. Rev. B* **83**, 174444 (2011).
- ³²A. Vansteenkiste, J. Leliaert, M. Dvornik, M. Helsen, F. Garcia-Sanchez, and B. Van Waeyenberge, *AIP Adv.* **4**, 107133 (2014).
- ³³M. J. Donahue and D. G. Porter, “OOMMF user’s guide, version 1.0, inter-agency report NISTIR 6376,” Technical Report NISTIR 6376 (National Institute of Standards and Technology, Gaithersburg, MD, 1999).
- ³⁴S. Wiebel, J. P. Jamet, N. Vernier, A. Mougin, J. Ferré, V. Baltz, B. Rodmacq, and B. Dieny, *Appl. Phys. Lett.* **86**, 142502 (2005).
- ³⁵S. Wiebel, J. P. Jamet, N. Vernier, A. Mougin, J. Ferré, V. Baltz, B. Rodmacq, and B. Dieny, *J. Appl. Phys.* **100**, 043912 (2006).
- ³⁶A. Bellec, S. Rohart, M. Labrune, J. Miltat, and A. Thiaville, *Europhys. Lett.* **91**, 17009 (2010).
- ³⁷P. J. Metaxas, R. L. Stamps, J. P. Jamet, J. Ferré, V. Baltz, and B. Rodmacq, *J. Phys.: Condens. Matter* **24**, 024212 (2012).
- ³⁸M. Schneider, H. Hoffmann, and J. Zweck, *Appl. Phys. Lett.* **79**, 3113 (2001).
- ³⁹J. Sampaio, V. Cros, S. Rohart, A. Thiaville, and A. Fert, *Nature Nanotechnol.* **8**, 839 (2013).
- ⁴⁰P. Wohlhüter, M. T. Bryan, P. Warnicke, S. Gliga, S. E. Stevenson, G. Heldt, L. Saharan, A. K. Suszka, C. Moutafis, R. V. Chopdekar, J. Raabe, T. Thomson, G. Hrkač, and L. J. Heyderman, *Nat. Commun.* **6**, 7836 (2015).