Universiteit
Leiden
The Netherlands

Methods to simulate fermions on quantum computers with hardware limitations

Steudtner, M.

Citation
Steudtner, M. (2019, November 20). *Methods to simulate fermions on quantum computers with hardware limitations*. *Casimir PhD Series*. Retrieved from https://hdl.handle.net/1887/80413

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/80413

Note: To cite this publication please use the final published version (if applicable).

Cover Page

Universiteit Leiden

The following handle holds various files of this Leiden University dissertation:
http://hdl.handle.net/1887/80413

Author: Steudtner, M.
Title: Methods to simulate fermions on quantum computers with hardware limitations
Issue Date: 2019-11-20

Summary

This thesis is a collection of theoretical works aiming at adjusting quantum algorithms to the hardware of quantum computers. The overarching topic of these efforts is to enable digital quantum simulation, the process of approximating the ground state of an arbitrary physical system with elementary operations of a quantum computer. For fermionic systems, a class including molecules and materials, the impact of quantum computing would be undoubtedly high, and algorithms exist for their simulation. However, there is a certain gap between the requirements of those algorithms and what actual quantum devices can provide: it seems that our expectations of a fully-fledged quantum computer still exceed our capabilities to build it. To make quantum simulation feasible, we seek to adapt quantum algorithms to three different types of device limitations within this thesis. In particular, we address two of those issues in the context of fermionic quantum simulation and discuss the third while doing quantum error correction (which is a low-level task of a quantum computer).

Firstly, we consider a shortage of quantum bits. Qubits are a valuable resource that is squandered when quantum information is stored uncompressed in a quantum memory. This is in particular the case with digital quantum simulation when the physical system is modeled using the Jordan-Wigner transform – a longstanding standard of representing fermionic systems as spins (equivalent to qubits). Here, only a small number of configurations that the quantum memory could encode are actually relevant for the simulation. In chapter two, we therefore set out to save qubits by restricting ourselves to only those relevant configurations. The key to our method is that all quantum data stored in the qubit memory is a superposition of bit strings describing the simulated system. In the Jordan-Wigner case, these bit strings are readable as fermionic oc-

cupations (different states describing how the fermions occupy their host system). By using binary codes, only physically relevant occupations are mapped to somewhat shorter bit strings, and so stored on fewer qubits. This procedure however comes at the expense of the simulation runtime, as the fermionic interactions are mapped to operations more complicated than in the Jordan-Wigner case – an effect that increases the stronger the information is abstracted. Nonetheless, our efforts result in a scalable method that helps reducing the qubit requirements a bit.

A different limitation comes from the connectivity of qubits in a quantum device, and the need to keep quantum algorithms short. Of course, we will not be able to perform arbitrary quantum operations on our memory. Qubits have physical locations, and our ability to perform operations that couple two of them is geometrically limited. While scientists and engineers work on realizing devices with square lattice connectivity, fermionic quantum simulation is unfortunately not harnessing that connectivity in an optimal way. This problem is related to the locality of fermionic interactions: when simulating a term that is geometrically local in the fermionic system's two-dimensional embedding, one would like to only read and write on a local group of qubits. In this way, the quantum algorithm can be parallelized. Considering the limits set by decoherence, the resulting decrease in algorithmic runtime might even be the deciding factor for the feasibility of a computation. In chapter three, we again tinker with the way physical states are represented in the memory. The Jordan-Wigner transform, which is not locality-preserving, is here again concatenated with a code layer. This time however, the code is quantum – a construction with code words that can no longer be thought of as bit strings. Rather than relaxing the resource requirements, the number of qubits almost needs to be doubled. This increase in memory is necessary to store nonlocal interactions locally on single qubits, from where those interactions can then be retrieved via local operations. Not only does this allow to simulate two-dimensional systems in a constant number of algorithmic steps, but ensures parallelizability in general.

Lastly, we consider a limitation that is swept under the rug fairly often. For devices with many qubits, constraints on parallel operation will appear due to a limited capability to address the involved qubits at the same time. A realistic layout for a future quantum device is a densely-packed matrix of qubits, so dense in fact that they can only be manipulated by wires contacting the matrix periphery. In chapter four,

we consider such a crossbar design for spin qubits in silicon quantum dots. There, the pulsing of lines that go into the chip allows us to set the electric dot potentials and interdot confinement barriers, by which quantum operations can be facilitated. For all those operations at least two lines must be pulsed affecting only the part of the chip at which they intersect. However, when attempting to run several operations in parallel, there are not just the crosspoints that we intend, but also others that would not occur if the operations were performed in sequence. At those spurious crossings, unanticipated quantum operations are induced. As a way out, we use the quirks of the proposed architecture and for instance move electrons (qubits) between adjacent dots: when the dots at the spurious crossings are empty, spurious operations cannot corrupt the quantum memory. By providing charge distributions with these properties, as well as instructions for the operation of the lines, we find quantum error correction programs compatible with the crossbar architecture.