COMPLEXITY OF QUANTUM FIELD THEORIES

MAX ZIMET

Abstract. Quantum field theories (QFTs) reconcile special relativity and quantum mechanics. We discuss the computational complexity of these theories. In particular, we present the recently-devised algorithm of Jordan, Lee, and Preskill which gives an efficient simulation of ϕ^4 theory in $d = 1, 2$, and 3 spatial dimensions with a non-relativistic quantum computer, allowing for the computation of scattering probabilities. The algorithm's run time is polynomial in the desired precision, the number of incoming particles, and their energy. The fastest known classical algorithm is exponentially slower when we desire high precision or when the ϕ^4 coupling constant is large.

1. Introduction

Aside from theories that contain infinitely many types of particles, such as string theory, quantum field theory is the unique way to reconcile special relativity and quantum mechanics[1]. In the common case where one may ignore the effects of gravity, such as in particle accelerators, particle physicists use the standard model, which is a quantum field theory, to understand and predict the results of scattering experiments. In this paper, we investigate the ability of non-relativistic quantum computers to efficiently compute QFT scattering probabilities.

In particular, we present the recently-developed algorithm of Jordan, Lee, and Preskill which gives an efficient simulation of massive ϕ^4 theory with a non-relativistic quantum computer, allowing for the computation of scattering probabilities[2]. This theory is far simpler than the standard model (the only field it contains is a simplified version of the Higgs field – there are no electrons, photons, etc.)[3]. The algorithm presented here for simulating ϕ^4 theory does not work for the standard model, as the latter contains features such as chiral fermions and gauge interactions which are not accounted for by the ϕ^4 algorithm. Nevertheless, it is an indication that the quantum circuit model, which uses non-relativistic quantum mechanics, may be powerful enough to efficiently simulate all of nature (ignoring gravity)[2]. Put another way, since nature can efficiently simulate non-relativistic quantum computers, this indicates that non-relativistic and special-relativistic quantum mechanics may have the same computation power[4].

1.1. **Classical Intuition.** If we take away quantum mechanics, then it does seem that non-relativistic and special-relativistic computers have the same power (in the sense that each may solve the same set of problems in polynomial time). We argue this following Aaronson[4, 5]. The state space of both special relativity and Newtonian mechanics is a "classical" state space. That is, the state of the computer at a given time, in each case, is a list of the positions and momenta of all particles involved in the computation; this list evolves deterministically. So, the

Date: December 13, 2012.

only possible changes in computation power will be due to relativistic effects such as time dilation and the absolute speed limit c. The former could conceivably give more computation power in the relativistic case, since a person could get on a fast-moving rocket after leaving a computer on Earth to work on a hard computation. When he or she returns to the Earth, he or she would find all of his or her friends and family dead, but the computation would be complete. The problem with this argument is that to attain rapid enough speeds for time dilation to be significant enough, enough energy would need to be produced in such a way that a black hole would result, sucking the rocket in[5]. The effect of the speed limit c would be concerning, except Turing machines have been shown to be polynomial-time equivalent to random access machines, so that massive parallel computations being limited by c are not a concern[4]. Thus, it seems likely that non-relativistic and relativistic computers are equally powerful if we get rid of quantum mechanics.

1.2. **Motivation.** There are essentially three different methods with which one may attempt to study a quantum field theory. The one typically taught in QFT classes is that of perturbation theory, in which one considers a theory in which particles are essentially free; more precisely, they interact with each other only via weak interactions. This method has had tremendous success, especially in allowing computations with the QFT describing electromagnetism: quantum electrodynamics (QED)[3]. However, the coupling constants describing the strengths of the interactions of different QFTs are not always weak. For example, in quantum chromodynamics (QCD), the QFT describing the strong force, the coupling constant is large at certain energies. We then need non-perturbative methods[1, 6, 3].

There are two classes of non-perturbative methods: exact and numerical. We may only solve a very restricted class of theories exactly – such theories typically have infinitely many fields or reside in one spatial dimension[3]. Therefore, for computing quantities in realistic field theories we turn to numerical solutions. The algorithms developed for classical computers have, thus far, been able to numerically calculate static quantities, such as mass ratios. However, they have not been able to compute scattering amplitudes, which are important quantities describing the dynamics of interacting particles[2]. It is therefore natural to hope for the existence of an efficient quantum algorithm for computing scattering probabilities. Given the discussion in Sec. 1.1, this does not seem to be an unreasonable hope. In fact, the question of whether quantum computers could efficiently simulate QFTs motivated Feynman to introduce the concept of a quantum computer three decades ago[7].

2. What is QFT?

QFT is the natural result of combining special relativity with quantum mechanics. Its physical motivation is very simple: it relies on only one equation from special relativity and one inequality from quantum mechanics. From special relativity, we have the relation $E = m$[1] giving the energy of a particle of mass m at rest.

[1] Throughout this paper, we work in units where $\hbar = c = 1$, as is conventional in the QFT literature. These constants may be restored in any equation through the use of dimensional analysis. In our units, every quantity we will be concerned with has units of energy to some power. For instance, mass has units of energy, as can be seen from $E = mc^2$, distance has units of 1/energy, as can be seen from the formula $E = 2\pi\hbar c/\lambda$ giving the energy of a photon in terms of its wavelength, speed is unitless, since c is a speed, and time has units of 1/energy, as is shown by the formula $d = vt$ giving the distance d traveled by an object with speed v for a time t.

Quantum mechanics gives us the Heisenberg uncertainty principle $\Delta E \Delta t \geq 1/2$, which gives a lower bound on an experimenter's uncertainty in the energy of a system, ΔE, as a function of the amount of time, Δt, during which the system does not change appreciably[8]. We thus see that there is a non-zero probability for the universe to violate energy conservation, as long as this violation is only for a short time[3].[2] In particular, since we can turn energy into mass, we see that there is a non-zero probability for particles to be created from the vacuum, as long as they exist only for short amounts of time if their existence violates energy conservation. We call such short-lived particles "virtual particles." These particles can be used to mediate the transfer of energy and momentum from one real particle to another. This is the QFT explanation for forces, such as electromagnetism.

We need a way to mathematically describe the dynamics of an uncertain number of particles. We therefore define a field to be a mathematical construct (as opposed to a physically observable quantity, such as a particle) which we envision as permeating all of spacetime, and which has excitations which are particles of a given type at a given point.[3] For example, in the standard model there is an electron field, a gluon field, etc. We further complicate matters by demanding that each field obey either bosonic or fermionic statistics. This constrains the occupation number of each field at every point in spacetime – that is, the number of particles associated with a given field which may be found at a given spacetime point. QFT is then the quantum mechanics of a collection of fields.

In quantum mechanics, time evolution is performed via application of a unitary operator. Since QFT is simply the quantum mechanics of fields, we expect this to remain the case. Just as in quantum mechanics, we evolve our state by a time Δt using the operator

$$U = e^{-iH\Delta t},^{4} \tag{2.1}$$

where H is the Hamiltonian of our system. (The Hamiltonian is a Hermitian operator whose eigenstates are the states with definite values of energy. The energy of an eigenstate is its eigenvalue). We require locality of our field theory, so the Hamiltonian of our system is of the form

$$H = \int d^3 \, \mathcal{H}(x),$$

where $\mathcal{H}$ is called the "Hamiltonian density" of our system[3].

[2] Technically, in perturbative QFT calculations both energy and momentum are conserved at every interaction vertex in a Feynman diagram. It is perhaps more precise to say that the masses of the virtual particles – those particles which exist only for a short time, and which cannot be observed – can have any value, including imaginary values. This is true if we identify the square of the 4-momentum of a virtual particle with its squared mass, as we do with observable particles.

[3] This point is in space if we are using the Schrödinger picture of quantum mechanics, in which operators are time-independent, and is in spacetime if we are using the Heisenberg picture, in which operators depend on time.

[4] This is overly simplified. We want to choose U so that our time evolution is that governed by the Schrödinger equation. The U of Eq. (2.1) will only agree with the Schrödinger equation if H is time-independent. Otherwise, we need to integrate a time-dependent H over the time through which we want to time evolve our state, rather than simply multiply H by Δt. In fact, there is one further complication if H does not commute with itself at different times, as is typically the case in QFT. In this case, we must "time-order" our exponential. For an explanation of this process, the reader is referred to [3].

We note that this formulation of QFT places time on a separate footing from space, hiding the Lorentz symmetry inherent in special relativity.[5] We can see that this formulation is, in fact, consistent with special relativity by switching from the Hamiltonian formalism of unitary time evolution to the Lagrangian formalism, also known as Feynman's path integral formalism. These two formulations of QFT can be proven to be equivalent. The path integral involves the computation of quanities from

$$Z = \int \mathcal{D}\phi \, e^{i \int d^4x \, \mathcal{L}(\phi)}, \tag{2.2}$$

the QFT equivalent of the partition function of statistical mechanics. Here, $\mathcal{D}\phi$ is a (mathematically ill-defined) integration measure for integrating over functions on spacetime, and we integrate over all possible field configurations ϕ; $\mathcal{L}$ is the "Lagrangian density." We note that Z is manifestly Lorentz invariant, if $\mathcal{L}$ is, since our integral in the exponential is. We note that the requirement that $\mathcal{L}$ be Lorentz invariant severely restricts the set of allowed Lagrangian densities for QFTs; since the Lagrangian density for a theory completely determines the theory, the set of relativistic QFTs is severely restricted.[6]

3. Difficulties in Simulating a QFT with a Quantum Computer

There are a number of challenges to overcome in order to simulate ϕ^4 theory with a quantum computer. First, there is the difficulty of what the simulation should output, since the initial state only determines a probability distribution for the ouput state, not the output state itself. We overcome this difficulty by having the algorithm sample from this probability distribution. We require that the probability of the algorithm outputting a certain result be within $\pm\epsilon$ of the true probability to obtain the result. Next, depending on the total energy of particles in our initial state, we are kinematically allowed (i.e., allowed by conservation of energy and momentum) to have a large number of outgoing particles. We therefore allow our algorithm to take time polynomial in the total energy of particles in our initial state.

Third, the value of the field ϕ (remember, we are considering a theory with only one field) at every spacetime point can take on uncountably many values. It is therefore necessary to cut off the absolute value of the field (for simplicity, we will require our field to be real – that is, $\phi^\dagger = \phi$) at a maximum value $\phi_{\max}$, and to discretize the possible values of ϕ, binning the uncountable set $[-\phi_{\max}, \phi_{\max}]$ into bins of width δ_ϕ. In order to represent the value of the field at each point, we require

$$n_b = O(\log(\phi_{\max}/\delta_\phi)) \tag{3.1}$$

qubits per spacetime point. We determine values of $\phi_{\max}$ and δ_ϕ which are consistent with our error probability ϵ in Sec. 6.

n_b qubits for each point in spacetime still amounts to infinitely many qubits, so we discretize our theory, putting it on a d-dimensional spatial lattice $\Omega = a\mathbb{Z}^d_{\hat{L}}$

[5] Lorentz symmetry is the invariance of the laws of physics as one changes between different inertial reference frames. This, in a sense, allows for time and space to be rotated into each other without changing the form of physical equations. More precisely, Lorentz symmetry is the symmetry $SO(1,3)$. This is the group of matrices Λ for which, for any 4-vector $v = (t,x,y,z)^T$, if we define $(t',x',y',z')^T = \Lambda v$ then $v^2 := t^2 - x^2 - y^2 - z^2 = t'^2 - x'^2 - y'^2 - z'^2$.

[6] It is further restricted if we require that a theory be renormalizable.

with $\mathcal{V} = \hat{L}^d$ lattice sites and a lattice constant of a. We then require $n_b\mathcal{V}$ qubits to represent the state of our field at a given time. We call the short-distance cutoff the ultraviolet, or UV, cutoff, and the long-distance cutoff the infrared, or IR, cutoff. (These names come from wavelengths of light – UV light has a short wavelength, while IR light has a long wavelength). There is important physics involved in why we need a small UV and a large IR cutoff; this physics must be considered in determining discretization errors. We will consider the error arising from the IR cutoff in Sec. 7, and the error arising from the UV cutoff in Sec. 8.

4. ϕ^4 Theory in $d = 1, 2, 3$ Spatial Dimensions

The ϕ^4 Hamiltonian density is

$$\mathcal{H} = \frac{1}{2}\pi^2 + \frac{1}{2}(\nabla\phi)^2 + \frac{1}{2}m_0^2\phi^2 + \frac{\lambda_0}{4!}\phi^4, \quad \pi = \dot{\phi}. \tag{4.1}$$

Here, we have introduced π, the momentum conjugate to ϕ; the dot in $\dot{\phi}$ indicates a time derivative, while ∇ indicates the spatial gradient. Our spatial discretization turns ∇ into a finite-difference operator. The parameters m_0 and λ_0 are called, respectively, the "bare mass" and the "bare coupling constant." We note that the physical mass of a ϕ particle depends on both the value of m_0 and the value of λ_0, as interactions (due to the final term in $\mathcal{H}$) change the physical mass so that it differs from the bare mass. In fact, in one and two spatial dimensions there is a value of λ_0, which we call λ_c, at which our QFT reaches a critical point. At this critical point, our QFT undergoes a phase transition. More importantly for our purposes, at this point the physical mass becomes zero. Since we rely on the physical mass being non-zero in our algorithm, in a number of places, we require $\lambda_0 < \lambda_c$. However, we will allow λ_0 to be arbitarily close to λ_c. In three spatial dimensions, ϕ^4 theory is believed to be trivial, in the sense that interactions are turned off as our lattice spacing a approaches zero. We consider $d = 3$ spatial dimensions nonetheless, as our spacing a is not exactly zero and the behavior of the theory near $a = 0$ is still interesting. When $d = 3$, there is no critical point λ_0[2].

5. Overview of the Algorithm

We now give an overview of the algorithm. Because we want to use the Suzuki-Trotter quantum simulation algorithm, we employ the Hamiltonian formulation of QFT.

We first discuss the Suzuki-Trotter algorithm, as it is used in steps 2-6 below. This algorithm is described in [9, 10]. It is an algorithm for efficiently simulating quantum mechanical systems using a quantum computer. Using the k-th order Suzuki-Trotter formula, we can simulate time evolution due to a unitary operator e^{-iHt} with a quantum circuit containing $O((t\mathcal{V})^{1+1/2k})$ gates; the near-linear scaling with $\mathcal{V}$ is due to the locality of our theory[2].

We now discuss each of the six main steps steps of the algorithm.

5.1. Prepare the Vacuum State for $\lambda_0 = 0$.

We first set $\lambda_0 = 0$. This gives an exactly solvable quantum field theory; it is uninteresting, however, as the fields do not interact. We can find the ground state of this theory by using creation and annihilation operators[3], as in the case of the 3-dimensional quantum harmonic oscillator[8]. More specifically, we write, referring to the Hamiltonian density with $\lambda_0 = 0$ as $H^{(0)}$ and defining the reciprocal lattice $\Gamma = \frac{2\pi}{\hat{L}a}\mathbb{Z}^d_{\hat{L}}$,

$$H^{(0)} = \sum_{\mathbf{p}\in\Gamma} L^{-1}\omega_{\mathbf{p}} a^\dagger_{\mathbf{p}} a_{\mathbf{p}} + E^{(0)} I,$$

where I is the identity operator,

$$\omega_{\mathbf{p}} = \sqrt{m_0^2 + \frac{4}{a^2}\sum_{j=1}^{d}\sin^2\left(\frac{ap_j}{2}\right)},$$

and $E^{(0)}$ is an irrelevant zero-point (i.e., ground state) energy. We note that $\omega_{\mathbf{p}}$ is the energy of an excitation of momentum $\mathbf{p}$; the limit of this quantity as $a \to 0$ is $\sqrt{m_0^2 + \mathbf{p}^2}$. This is the energy, according to special relativity, of a non-interacting particle of mass m_0 and momentum $\mathbf{p}$, as we expect.

The result is analogous to that of the 3-dimensional quantum harmonic oscillator: our ground state is a multivariate Gaussian wavefunction in the variables $\{\phi(\mathbf{x})|\mathbf{x} \in \Omega\}$. We prepare this ground state using the method of Kitaev and Webb[11]. The efficiency of the Kitaev-Webb method is limited by the computation of the LDL^T decomposition of a matrix; this, classically, can be done in $O(\mathcal{V}^{2.373})$ time[12, 13, 14].

5.2. **Excite Separated Wavepackets.** We initially want to create particles which are well-separated in space, but which have a well-defined momentum. We cannot perfectly localize our particles without losing all information about their momenta, and vice versa, due to the Heisenberg uncertainty principle $\Delta p \Delta x \geq 1/2$. So, we excite a wavepacket whose position and momenta can be known fairly accurately. We use

$$a^\dagger_{\mathbf{x}} = \sum_{\mathbf{p}\in\Gamma} L^{-d} e^{-i\mathbf{p}\cdot\mathbf{x}} \sqrt{\frac{1}{2\omega(\mathbf{p})}} a^\dagger_{\mathbf{p}}$$

to create a particle at location $\mathbf{x}$. Then,

$$a^\dagger_\psi = \eta(\psi) \sum_{\mathbf{x}\in\Omega} a^d \psi(\mathbf{x}) a^\dagger_{\mathbf{x}}$$

is the creation operator for a wavepacket whose shape is described by $\psi(x)$. ($\eta(\psi)$ is a normalization factor, so that $[a_\psi, a^\dagger_\psi] = 1$, as we would like).

We would like to simulate the effect of applying $a^\dagger_\psi$ to the vacuum. However, this operator is not unitary. So, we append an additional qubit to our system and define

$$H_\psi = a^\dagger_\psi \otimes |1\rangle\langle 0| + a_\psi \otimes |0\rangle\langle 1|.$$

We then find that

$$e^{-iH_\psi \pi/2}|\text{vacuum}\rangle|0\rangle = -ia^\dagger_\psi|\text{vacuum}\rangle|1\rangle.$$

We therefore want to simulate H_ψ for time $\pi/2$. After simulating this unitary, with the Suzuki-Trotter algorithm, we discard the auxiliary qubit. We repeat this process for each incoming particle.

We can express H_ψ in terms of ϕ and π, and we then find that simulating H_ψ with the adiabatic algorithm is similar to simulating H.

As we are creating these particles, we do not want the wavepackets of the particles that we have already created to change. As discussed in Sec. 7, this is not an issue if our lattice is large enough.

5.3. **Adiabatically Turn on the Interaction.** We adiabatically change λ_0 to the value we want it to take, using the Suzuki-Trotter algorithm. We do not want this simulation to change the number of particles in our scattering experiment. However, there is an energy gap between our desired state and states with different numbers of particles, since our particles are massive. Therefore, at a given value of λ_0, the rate at which we can change λ_0 depends on the mass of the particles. However, recall that the physical mass depends on λ_0. If the theory is weakly-coupled then we can perturbatively compute this dependence of the physical mass on λ_0, and therefore know how quickly to change λ_0. However, we cannot compute this mass if the theory is strongly coupled. So, what we do is compute the physical mass at each value of λ_0, using our quantum computer, and use this to know how quickly to change λ_0.

To compute the physical mass, we proceed with the following process. First, we prepare the interacting ground state for some small value of λ_0, and a state with one particle in it at this same value of λ_0, using the methods of the preceding paragraph. We then compute the energy of each state, using phase estimation (as discussed in Sec. 5.6), and find the difference. This gives us the physical mass of a particle at this small value of λ_0. We then use this value of the mass to know how to construct the ground- and one-particle-states for a slightly larger value of λ_0. Repeating this process, we find the value of the physical mass for a number of different values of λ_0.

We also do not want our turning on the coupling to change the locations of or broaden our particles' wavepackets. To solve this problem, we simulate backward time evolution before and after each change in λ_0, in a sense made more precise in [2].

One last issue must be dealt with. In one and two spatial dimensions, we must avoid the critical point as we adiabatically turn on the interaction. This may require us to follow a path in parameter space (λ_0, m_0) which takes longer to simulate than the direct path from $\lambda_0 = 0$ to the larger value of λ_0 which we could follow in three spatial dimensions. This introduces dependence on $\frac{1}{\lambda_c - \lambda_0}$ into the run-time of the algorithm.

5.4. **Simulate Scattering.** This is a simple application of the Suzuki-Trotter algorithm, with the Hamiltonian describing our time evolution being the QFT's Hamiltonian, H.

5.5. **Adiabatically Turn off the Interaction.** This is similar to step 3.

5.6. **Measure the Momenta of Outgoing Particles.** We use the method of phase estimation[15] to compute the number of particles with a given momentum, using the fact that $a_{\mathbf{p}}^{\dagger} a_{\mathbf{p}}$ is proportional to the number operator for the $\mathbf{p}$ mode, so that the measured phase of $e^{iL^{-d} a_{\mathbf{p}}^{\dagger} a_{\mathbf{p}}}$ gives the occupation number of the $\mathbf{p}$ mode. This step requires $O(\mathcal{V}^{2+1/2k})$ gates, using a k-th order Suzuki-Trotter formula.

6. Discretization Error Introduced by the Field Cutoff and Field Discretization

Suppose we are in a state $|\psi\rangle$ with $\langle\psi|H|\psi\rangle \leq E$. Then, at a given lattice location $\mathbf{x}$ the probability distribution over $\phi(\mathbf{x})$ has a very low probability for $|\phi(\mathbf{x})|$ to be much larger that $O\left(\sqrt{E}\right)$. It can be shown, by bounding $\langle\psi|\phi(\mathbf{x})|\psi\rangle$ and $\langle\psi|\phi^2(\mathbf{x})|\psi\rangle$ as functions of E and applying Chebyshev's inequality, that a cutoff $\phi_{\max} = O\left(\sqrt{\frac{\mathcal{V}E}{a^d m_0^2 \epsilon}}\right)$ is consistent with our error probability ϵ.

Next, we turn to the determination of δ_ϕ. We can use similar methods as in the previous paragraph, once we notice that the eigenbasis of $\phi(\mathbf{x})$ is the Fourier transform of the eigenbasis of $a^d\pi$, and that therefore discretizing ϕ into bins of width δ_ϕ is equivalent to cutting off the absolute value of $\pi(\mathbf{x})$ at $\pi_{\max} = \frac{1}{a^d \delta_\phi}$. It then follows that we may choose $\pi_{\max} = O\left(\sqrt{\frac{\mathcal{V}E}{\epsilon a^d}}\right)$. From Eq. (3.1), we see that we may choose

$$n_b = O\left(\log\left(\phi_{\max}\pi_{\max}a^d\right)\right) = O\left(\log\left(\frac{\mathcal{V}E}{m_0\epsilon}\right)\right) [2].$$

7. Discretization Error Introduced by the IR Cutoff

As discussed in Sec. 5.2, we want our universe to be infinite so that particles do not interact as we initialize our simulation. Since our particles have non-zero mass, the errors associated with a non-infinite separation δ shrink exponentially with δ: $\epsilon_{IR} \sim e^{-\delta/m}$. So, we may choose the volume of our system to scale as $V \sim n_{\text{in}} \log(1/\epsilon)$. The same considerations hold for manipulations with the output states – the steps in Secs. 5.5 and 5.6. Therefore, we require $V \sim n_{\text{out}}$, where n_{out} is the maximum kinematically-allowed number of particles which can be output.

8. Discretization Error Introduced by the UV Cutoff

The theory of the UV cutoff is the subject of the renormalization group in QFT. This is important not only in lattice simulations of QFT, but in perturbative calculations as well, because naive calculations in QFT give horrible infinities, suggesting that new physics becomes significant at small distance scales (i.e., the Planck scale associated with quantum gravity)[3]. The renormalization group requires that we add all terms consistent with our theory's symmetries (for example, $\phi \to -\phi$ and Lorentz symmetry) to the Lagrangian density.[7] These new terms are all suppressed by positive powers of a; the most significant terms are suppressed by powers of a^2. That is, UV discretization errors are $O(a^2)$. So, limiting ϵ determines a and $\mathcal{V} = \frac{V}{a^d}$, where V is the volume of our system, determined by the IR cutoff.

9. Summary

We have described a quantum algorithm which overcomes a number of infinities in quantum field theory and is able to simulate ϕ^4 theory efficiently, using a finite number of qubits. The algorithm's run time is polynomial in the desired precision, the number of incoming particles, and their energy, as is demonstrated by the more complete analysis of the algorithm provided by [2]. The fastest known classical

[7] Actually, in the case of our lattice simulation, our Lagrangian density violates Lorentz symmetry, so we must add Lorentz symmetry-violating terms.

algorithm is exponentially slower when we desire high precision or when the coupling constant is large. The number of quantum gates we must apply in the case of weak coupling scales as

$$G_{\text{weak}} \sim \begin{cases} \left(\frac{1}{\epsilon}\right)^{1.5+o(1)} & : d = 1 \\ \left(\frac{1}{\epsilon}\right)^{2.376+o(1)} & : d = 2 \\ \left(\frac{1}{\epsilon}\right)^{3.564+o(1)} & : d = 3 \end{cases} .$$

In the case of strong coupling, the scaling of the required number of quantum gates in terms of $\lambda_c - \lambda_0$, the total momentum p of the incoming particles, and n_{out} is given in Table 1, which is adapted from [2].

	$\lambda_c - \lambda_0$	p	n_{out}
$d = 1$	$\left(\frac{1}{\lambda_c-\lambda_0}\right)^{8+o(1)}$	$p^{4+o(1)}$	$\tilde{O}(n_{\text{out}}^5)$
$d = 2$	$\left(\frac{1}{\lambda_c-\lambda_0}\right)^{5.04+o(1)}$	$p^{6+o(1)}$	$\tilde{O}(n_{\text{out}}^{7.128})$

TABLE 1. $f(n) = \tilde{O}(g(n))$ means $f(n) = O(g(n) \log^c(n))$ for some c

References

[1] S. Weinberg. *The Quantum Theory of Fields*, volume 1. Cambridge University Press, 1995.

[2] S. Jordan, K. Lee, and J. Preskill. Quantum algorithms for quantum field theories. *Science*, 336:1130–1133, 2012.

[3] M. Peskin and D. Schroeder. *An Introduction to Quantum Field Theory*. Westview Press, 1995.

[4] S. Aaronson. Quantum algorithms for quantum field theories. Shtetl-Optimized: The Blog of Scott Aaronson, Dec. 2011.

[5] S. Aaronson. Relativistic computation. MIT 6.896 Lecture Notes (Scribe – B. Juba), Dec. 2008.

[6] S. Weinberg. *The Quantum Theory of Fields*, volume 2. Cambridge University Press, 1995.

[7] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(467), 1982.

[8] D. Griffiths. *Introduction to Quantum Mechanics*. Pearson Education, Inc., 2nd edition, 2005.

[9] M. Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations. *Physics Letters A*, 146(319), 1990.

[10] D. Berry, G. Ahokas, R. Cleve, and B. C. Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. *Communications in Mathematical Physics*, 270(359), 2007.

[11] A. Kitaev and W. Webb. Wavefunction preparation and resampling using a quantum computer. *arXiv:0801.0342*, 2008.

[12] V. Williams. Breaking the coppersmith-winograd barrier. *http://www.cs.berkeley.edu/ virgi/matrixmult.pdf*, 2011.

[13] J. R. Bunch and J. E. Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Mathematics of Computation*, 28(231), 1974.

[14] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 1990.

[15] A. Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv:quant-ph/9511026*, 1995.

Acknowledgements

The author is grateful to Professor Scott Aaronson for his amazing course in quantum complexity theory.