

Dynamical implementation of canonical second quantization on a quantum computer

Juan José Gálvez-Viruet^{*} and Felipe J. Llanes-Estrada[✉]

*Departamento de Física Teórica and IPARCOS, Facultad CC. Físicas,
Universidad Complutense de Madrid, Plaza de las Ciencias 1, Madrid 28040, Spain*

 (Received 25 June 2024; accepted 16 October 2024; published 19 December 2024)

We develop theoretical methods for the implementation of creation and destruction operators in separate registers of a quantum computer, allowing for a transparent and dynamical creation and destruction of particle modes in second quantization in problems with variable particle number. We establish theorems for the commutation (anticommutation) relations on a finite memory bank and provide the needed symmetrizing and antisymmetrizing operators. Finally, we provide formulas in terms of these operators for unitary evolution under conventional two- and four-body Hamiltonian terms, as well as terms varying the particle number. In this formalism, the number of qubits needed to codify n particles with N_p modes each is of order $\mathcal{O}(n \log_2 N_p)$. Such scaling is more efficient than the Jordan-Wigner transformation which requires $\mathcal{O}(N_p)$ qubits, whenever there are a modest number of particles with a large number of states available to each (and less advantageous for a large number of particles with few states available to each).

DOI: [10.1103/PhysRevD.110.116018](https://doi.org/10.1103/PhysRevD.110.116018)

I. INTRODUCTION

Quantum computing is widely held as a promising tool to advance many-body physics in all areas where quantum phenomena are of importance, particularly nuclear and strongly coupled particle physics [1]. Quantum mechanical applications at fixed particle number abound [2], importantly for quantum chemistry and typical condensed matter Hamiltonians and increasingly nuclear physics [3,4]. In particular, quantum simulation of fermion systems is one of the most advanced applications of quantum computers. First algorithms were proposed 20 years ago [5–7], and most of them assume the second quantization formalism, in which the identification between the fermion occupation-number basis and a quantum memory of qubits is straightforward. Remarkably, [5] also considered the use of “first quantization” and already discussed algorithms to antisymmetrize a quantum memory. Nowadays, there are very efficient algorithms for fixed particle-number problems, in particular, to solve the electronic structure problem; see, for example, [2,8–13] or [14] for a recent review. Almost all of them use the occupation-number basis, but, as recent studies have shown, first quantization, in which the

quantum memory is antisymmetrized explicitly (as it is in this work), can scale better in terms of qubit counts and gate complexities [15–19].

Many of the techniques developed in quantum chemistry can be used for the simulation of quantum field theories but have to be generalized to accommodate a varying number of particles, to encode bosons as well as fermions and to consider the exponentiation of general Hermitian operators. Usually, these problems are solved by leaving aside particles and encoding fields instead, with lattice gauge theories a salient example [20–28]. In this work, we have adopted the few-body point of view in which the basic object is the particle [29]. In systems where a mass gap is present or spontaneously arises, this can be advantageous by reducing the number of modes that need to be taken into account to a given precision.

We have developed a natural implementation of the formalism of second quantization in a way that is scalable to large model spaces and computer memories without relying on the Jordan-Wigner transformation which, at every step of a calculation, puts the translation of the physical system into the quantum computer out of the reach of many possible users, due to its bookkeeping and improvable scalability.

Our philosophy here is to take one step of abstraction further so that the memory of the quantum computer is organized in a way similar to quantum field theory in canonical second quantization in the particle basis. This is intended to somewhat free the physicist from low-level thinking on the computer qubits but allow for higher-level objects, the particle registers, to become the basic stones for

^{*}On leave at Argonne National Laboratory, Lemont, IL 60439, USA.

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

building physical implementations. This register-particle codification was first introduced in [30] to simulate a boson theory, and it is similar to “compact encoding” [31]. The main differences with respect to previous studies is that our implementation of creation and annihilation operators exactly fulfills both commutation (or anticommutation) relations provided the memory (not completely occupied) is filled following a certain order. Furthermore, our operators proceed changing only a small amount of registers on each time-evolution step, avoiding the costs of having to advance the entire memory. We expect this particle register with clear physical interpretation to be useful for formulations of field theory in light-front gauge [32,33], time-axial gauge [24], or eventually Coulomb gauge [34] theories.

Readers familiar with object-oriented programming languages will quickly recognize the parallel among the particle creation and annihilation operators and the creator and destructor methods of a class, here the “particle,” whose attributes are its physical quantum numbers. This is a natural analogy. The particular feature of second quantization is that the creator and annihilation operators need to follow specific rules, in the form of (anti)commutation relations that need to be mapped to the computer’s methods.

This is certainly not the most efficient way to code field-theory problems for a quantum computer, with many intelligent ideas [22] proposed in the past decades, but with fast hardware improvements it becomes attractive to think of user-friendly ways of coding physical problems that will reduce the overhead time for developers.

Section II of this article is dedicated to implementing one-body operators. Then, Sec. III discusses the simultaneous occupation of two registers and how Bose symmetry among those two particles can be described: The addition of an extra register to accommodate two particles must comply with the requirements of quantum statistics. In it, we also start addressing the canonical commutation relations among creation and destruction operators.

In Sec. IV A, we generalize the concepts to a computer with an arbitrary number of registers and holding an arbitrary (but small) number of particles. The commutator of the earlier creation and annihilation operators will there be seen to be well implemented over the multiparticle Fock space, except for the boundary term that arises when the number of particles is large enough to saturate the number of qubits available.

We follow with Sec. IV B, dedicated to the implementation of fermion anticommuting operators. The creation operators must now avoid filling two separate registers with the same quantum numbers, which would be forbidden by Pauli’s exclusion principle, and properly implementing the anticommutation relations among the creation and destruction operators is a way to guarantee it.

The idea of the effort is to make a way to dynamically handle variable numbers of particles in running time

available as well as allow the computer to maintain superpositions of states with different particle numbers during execution. Section V then explains the implementation of evolution operators via exponentiation of the operators in the so-constructed Fock algebra and their products in the case that the particle number is constant. A discussion of the computational cost and scalability of the construction is also specified. This section is followed by an equivalent discussion for the more general case of particle-number-changing operators in Sec. VI. Further discussion about needed memory size and scalability is given in Sec. VII B, and a final discussion is laid out in Sec. VII D. Several demonstrations too extensive for the main text are presented, in the spirit of exhaustiveness, in the appendixes. This includes a glossary of symbols in Appendix C that may prove useful to the reader given the notation-intensive presentation.

II. IMPLEMENTATION OF SINGLE-PARTICLE OPERATORS

The abstraction that we pursue is to assign a “register” to each particle active in a process. Such registers are sets of qubits encoding the quantum numbers q of the represented particle, with one additional qubit that we dub the “presence or absence” qubit denoted as P/A , which is a handle allowing one to decide whether one particle is represented by that register or whether it should be considered empty and available. If an additional particle is needed, an empty register has to be activated and its presence qubit set to 1.

For a particle with momentum, spin, and internal quantum numbers such as a color, the vacuum state (corresponding to the absence of the particle) is conveniently taken as

$$|\Omega\rangle \equiv |0\rangle_{P/A} \otimes |0\rangle_{\text{spin}} \otimes |00\rangle_{\text{color}} \otimes |0\dots 0\rangle_{\text{momentum}}; \quad (1)$$

the number of possible values of a quantum number dictates how many qubits are necessary for its representation: Codification of spin requires one qubit, codification of color [35] two qubits, and codification of momenta, which is a continuous variable, requires discretization on a variable number of qubits to reach the wished precision. From here on, we consider N_p different values of momenta codified in $\lceil \log_2 N_p \rceil$ qubits [or, in three dimensions, remark which we will usually omit in the rest of the article, $(\lceil \log_2 N_p \rceil)^3$ qubits]. In the vacuum state, each register is in a reference state, here taken to be the null ket, that does not represent any valid quantum number.

Our first goal is to represent the action of the basic creation operator in Fock space, a staple of many-body theory, $a_{s,c,p}^\dagger$.

To implement it, we write it in terms of *set* operators \mathfrak{S} acting on each of the product Hilbert spaces describing each degree of freedom and control operators \mathfrak{C}_{ij} :

$$a_{s,c,p}^\dagger \equiv \mathfrak{C}_{10} \otimes \mathfrak{s}_s^\dagger \otimes \mathfrak{s}_c^\dagger \otimes \mathfrak{s}_p^\dagger, \quad (2)$$

so that

$$\begin{aligned} a_{s,c,p}^\dagger |\Omega\rangle &\equiv \mathfrak{C}_{10} |0\rangle_{P/A} \otimes \mathfrak{s}_s^\dagger |0\rangle \otimes \mathfrak{s}_c^\dagger |00\rangle \otimes \mathfrak{s}_p^\dagger |0\dots 0\rangle \quad (3) \\ &= |1\rangle_{P/A} \otimes |s\rangle \otimes |c\rangle \otimes |p\rangle \equiv |1scp\rangle. \quad (4) \end{aligned}$$

The first qubit, that indicated the presence or absence of a particle in the register, has now been set to $|1\rangle_{P/A}$. The control operator \mathfrak{C}_{10} also prevents the attempt of creating a quantum if the register is already encoding another one (“occupied register”). Control operators acting over the presence and absence qubits fulfill

$$\mathfrak{C}_{ij} |k\rangle = \delta_{jk} |i\rangle, \quad \mathfrak{C}_{ik} \mathfrak{C}_{im} = \delta_{ki} \mathfrak{C}_{im}. \quad (5)$$

The Hermitian conjugate of the abstract a^\dagger operators in many-body theory are the annihilation operators; they can be implemented as in

$$a_{s,c,q} = (a_{s,c,q}^\dagger)^\dagger = \mathfrak{C}_{01} \otimes \mathfrak{s}_s \otimes \mathfrak{s}_c \otimes \mathfrak{s}_q, \quad (6)$$

where, in intuitive notation, \mathfrak{s} are *scrap* operators which can clear the quantum numbers: \mathfrak{s}_s returns states of spin $|s\rangle$ (spin up or down) to the reference spin state $|s_0\rangle$ (spin down, for example), with analogous behavior for \mathfrak{s}_c and \mathfrak{s}_q . (Because the quantum computer will encode the unitary operators arising from exponentiation of combinations of a^\dagger and a , the fact that, e.g., $\mathfrak{C}_{10} |1\rangle = 0$ yields an unrepresentable null vector will pose no problem, as it exponentiates to the identity.)

In summary, the operations that this implementation of the Fock algebra can be used for are as follows.

- (i) To occupy an unassigned register, encoding a particle in it,

$$\begin{aligned} a_{s,c,p}^\dagger |\Omega\rangle &= \mathfrak{C}_{10} |0\rangle_{P/A} \otimes \mathfrak{s}_s^\dagger |0\rangle \otimes \mathfrak{s}_c^\dagger |00\rangle \otimes \mathfrak{s}_p^\dagger |0\dots 0\rangle \\ &= |1scp\rangle. \quad (7) \end{aligned}$$

- (ii) To vacate an occupied register if its quantum numbers match those of the destruction operator, resetting them to vacuum default,

$$\begin{aligned} a_{s',c',q'} |1scp\rangle &= |0\rangle \otimes \mathfrak{s}_{s'} |s\rangle \otimes \mathfrak{s}_{c'} |c\rangle \otimes \mathfrak{s}_{q'} |q\rangle \\ &= \delta_{s's} \delta_{c'c} \delta_{q'q} |\Omega\rangle. \quad (8) \end{aligned}$$

- (iii) To replace the quantum numbers of an existing particle occupying a register, employing combinations that conserve particle number (or, as a specific case, to pass on as with the identity) such as

$$\begin{aligned} &(a_{s_2,c_2,q_2}^\dagger a_{s'_2,c'_2,q'_2}) \cdot (a_{s_1,c_1,q_1}^\dagger a_{s'_1,c'_1,q'_1}) |1s_0c_0q_0\rangle \\ &= \delta_{s_0,s'_1} \delta_{c_0,c'_1} \delta_{q_0,q'_1} a_{s_2,c_2,q_2}^\dagger a_{s'_2,c'_2,q'_2} |1s_1c_1q_1\rangle \\ &\quad \times \delta_{s_0,s'_1} \delta_{c_0,c'_1} \delta_{q_0,q'_1} \delta_{s_1,s'_2} \delta_{c_1,c'_2} \delta_{q_1,q'_2} |1s_2c_2q_2\rangle \quad (9) \end{aligned}$$

or, acting on an empty register,

$$\begin{aligned} &(a_{s'_2,c'_2,q'_2} a_{s_2,c_2,q_2}^\dagger) \cdot (a_{s'_1,c'_1,q'_1} a_{s_1,c_1,q_1}^\dagger) |\Omega\rangle \\ &= \delta_{s_1,s'_1} \delta_{c_1,c'_1} \delta_{q_1,q'_1} a_{s'_2,c'_2,q'_2} a_{s_2,c_2,q_2}^\dagger |\Omega\rangle \\ &= \delta_{s_2,s'_2} \delta_{c_2,c'_2} \delta_{q_2,q'_2} \delta_{s_1,s'_1} \delta_{c_1,c'_1} \delta_{q_1,q'_1} |\Omega\rangle. \quad (10) \end{aligned}$$

III. IMPLEMENTATION OF TWO-PARTICLE BOSON OPERATORS

A. Basic necessary properties

In this section, we address the setting and handling of two bosons in the memory. The new complication, with respect to dealing with only one particle, is the need to represent the abstract symmetry under boson exchange independently of the underlying hardware.

Let us start by the vacuum states, omitting quantum numbers other than momentum to lighten the expressions:

$$\begin{aligned} |\Omega\rangle &= |\Omega\rangle_2 \otimes |\Omega\rangle_1 \\ &\equiv |0\rangle_{P/A} |0\dots 0\rangle_{\text{momentum}} \otimes |0\rangle_{P/A} |0\dots 0\rangle_{\text{momentum}}. \quad (11) \end{aligned}$$

We have highlighted the two product states by the presence and absence and a “momentum” subscripts, that we now drop, with the ordering of the two states specifying which is which.

The action of any creation operators $a_i^{(2)\dagger}$ over this pair of registers is defined to emulate their corresponding action on Fock space. [The superindex (2) indicates that it acts over a two-register memory, which will be useful to keep track of to distinguish the various operators discussed.]

If applied over the vacuum, it creates a first particle, which we assign to the rightmost register labeled (1):

$$a_{q_1}^{(2)\dagger} |\Omega\rangle = |q_1\rangle \equiv |\Omega\rangle_2 \otimes (|1\rangle|q_1\rangle)_1 = |\Omega\rangle_2 \otimes |1q_1\rangle_1. \quad (12)$$

If that register is already occupied by one boson, the creation operator needs to assign the second (note Bose’s symmetry):

$$\begin{aligned} a_{q_1}^{(2)\dagger} |\Omega\rangle_2 \otimes |1p_1\rangle_1 &\equiv \frac{1}{\sqrt{2}} (|1q_1\rangle_2 \otimes |1p_1\rangle_1 \\ &\quad + |1p_1\rangle_2 \otimes |1q_1\rangle_1). \quad (13) \end{aligned}$$

Finally, if both registers already contain a boson, no further particle can be encoded in this computer with only two registers.

The creation operator can then do nothing, and we naturally choose to set its action to null:

$$a_{q_1}^{(2)\dagger} |1p_2\rangle_2 \otimes |1p_1\rangle_1 = 0. \quad (14)$$

This is a condition at the boundary of the computer memory: Its size must truncate the Fock space and, as usual, introduce a discretization error in the second quantized problem being treated. Adding further memory will alleviate it, so we will return to this below in Sec. IV A.

In the case in which only one object sits in the two-register memory, we need to resolve the ambiguity of choosing one of them to carry the active quantum numbers.

We select to stack particles from right to left, entailing the exclusion of states such as $|1p_1\rangle_2 \otimes |\Omega\rangle_1$, i.e., with the first register empty and the second occupied. Note that no symmetry is here necessary, since the registers are just sections of memory and do not represent physical particles: Here there is just one and nothing to symmetrize.

The action of the corresponding annihilation operators is also defined to mimic their behavior in Fock space:

$$a_{q_1}^{(2)} |\Omega\rangle = 0;$$

$$a_{q_1}^{(2)} |\Omega\rangle_2 \otimes |1p_1\rangle_1 = \delta_{q_1 p_1} |\Omega\rangle_2 \otimes |\Omega\rangle_1 = \delta_{q_1 p_1} |\Omega\rangle, \quad (15)$$

and

$$a_{q_1}^{(2)} (|1p_2\rangle_2 \otimes |1p_1\rangle_1)_S = \delta_{q_1 p_2} |\Omega\rangle_2 \otimes |1p_1\rangle_1 + \delta_{q_1 p_1} |\Omega\rangle_2 \otimes |1p_2\rangle_1, \quad (16)$$

where $(|1p_2\rangle_2 \otimes |1p_1\rangle_1)_S$ is a two-particle symmetric state with momenta p_1 and p_2 .

With these requirements we can now try to write down the higher-level creation and annihilation operators in terms of the lower-level set and scrap operators from Sec. II.

B. Implementation

We find convenient to split the Fock operator acting on the two-register memory into two auxiliary suboperators, one acting on each of the two registers when that register is the last occupied one (which regrettably requires an additional subindex):

$$\begin{aligned} a_{q_1,1}^{(2)\dagger} &= (\mathfrak{C}_{00} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_{q_1}^\dagger)_1, \\ a_{q_1,2}^{(2)\dagger} &= (\mathfrak{C}_{10} \otimes \mathfrak{s}_{q_1}^\dagger)_2 \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1. \end{aligned} \quad (17)$$

(Parentheses isolate operators acting on the first or second register; the control terms \mathfrak{C}_{ij} act on presence qubits, and the identity \mathbf{i} and set \mathfrak{s} operators act on momentum qubits.) The construction of these auxiliary operators is straightforward: The first one sets the quantum numbers of the first, empty, register to the particle being added and flips its

presence qubit from absence to presence, while leaving the second register untouched.

The second operator leaves the first register, assumed occupied, untouched and adds the particle to the second. Before summing it with the first to form the total creation operator, we need to note that its action produces a two-particle state that does need to be symmetric under exchange. Hence, we multiply it by a symmetrizer:

$$a_{q_1}^{(2)\dagger} = a_{q_1,1}^{(2)\dagger} + \frac{1}{\sqrt{2}} (\mathbb{I} \otimes \mathbb{I} + \mathcal{P}_{21}) a_{q_1,2}^{(2)\dagger}. \quad (18)$$

[Note that symmetrization cannot be trivially implemented in runtime after the initial state preparation, because it projects out the antisymmetric parts of the Hilbert space, which is not a unitary operation germane to a quantum computer. We can circumvent the problem as shown in Eq. (72) below.]

The \mathcal{P}_{21} permutation operation swaps the quantum numbers encoded in each of the two registers:

$$\mathcal{P}_{21} |1p_1\rangle |1p_2\rangle \equiv |1p_2\rangle |1p_1\rangle, \quad (19)$$

whereas \mathbb{I} is the “register-level identity” which leaves the quantum numbers in the corresponding register unchanged.

If we substitute the auxiliary operators of Eq. (17) into Eq. (18), we can give a direct expression for the creation operator over the two-register memory:

$$\begin{aligned} a_{q_1}^{(2)\dagger} &= (\mathfrak{C}_{00} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_{q_1}^\dagger)_1 \\ &+ \frac{1}{\sqrt{2}} (\mathbb{I} \otimes \mathbb{I} + \mathcal{P}_{21}) (\mathfrak{C}_{10} \otimes \mathfrak{s}_{q_1}^\dagger)_2 \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1. \end{aligned} \quad (20)$$

This equation fulfills the quantum field-theory requirements of Eqs. (12) and (13) in the two-body sector.

The annihilation operators are then found by Hermitian conjugation

$$\begin{aligned} a_{q_1}^{(2)} &= (\mathfrak{C}_{00} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_{q_1})_1 \\ &+ (\mathfrak{C}_{01} \otimes \mathfrak{s}_{q_1})_2 \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1 \frac{1}{\sqrt{2}} (\mathbb{I} \otimes \mathbb{I} + \mathcal{P}_{21}). \end{aligned} \quad (21)$$

The first piece acts when only the first register is occupied, annihilating that one particle; but, if both are filled, its action is null, and it is the second piece that is active and erases the particle in the second register. We now turn to showing that this construction satisfies the canonical commutation relations in the Fock space restricted to two memory registers.

C. Canonical commutation relations among bosons

Boson operators must fulfill, at the operator level,

$$[a_{q_1} a_{q_2}^\dagger] = a_{q_1} a_{q_2}^\dagger - a_{q_2}^\dagger a_{q_1} = \delta_{q_1 q_2}. \quad (22)$$

In this subsection, we show that, in a two-register memory, this is satisfied for occupation numbers zero and one but that the two-particle case takes a boundary term (which can be avoided only by adding further registers).

The first of the two terms of the commutator $[a_{q_1}, a_{q_2}^\dagger]$ is, from Eqs. (20) and (21),

$$\begin{aligned} a_{q_1}^{(2)} a_{q_2}^{(2)\dagger} &= [(\mathfrak{C}_{00} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_{q_1})_1] \\ &\quad \times [(\mathfrak{C}_{00} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_{q_2}^\dagger)_1] \\ &\quad + [(\mathfrak{C}_{01} \otimes \mathfrak{s}_{q_1})_2 \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1] (\mathbb{I} \otimes \mathbb{I} + \mathcal{P}_{21}) \\ &\quad \times [(\mathfrak{C}_{10} \otimes \mathfrak{s}_{q_2}^\dagger)_2 \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1] \end{aligned} \quad (23)$$

[the crossed terms cancel because of Eq. (5)]. Associating together the operators over each subspace allows this to be written as

$$\begin{aligned} a_{q_1}^{(2)} a_{q_2}^{(2)\dagger} &= (\mathfrak{C}_{00} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{00} \otimes \mathfrak{s}_{q_1} \mathfrak{s}_{q_2}^\dagger)_1 \\ &\quad + (\mathfrak{C}_{00} \otimes \mathfrak{s}_{q_1} \mathfrak{s}_{q_2}^\dagger) \otimes (\mathfrak{C}_{11} \otimes \mathbf{i}) \\ &\quad + \mathcal{P}_{21} (\mathfrak{C}_{10} \otimes \mathfrak{s}_{q_2}^\dagger) \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_{q_1}). \end{aligned} \quad (24)$$

The reversed-order product is

$$\begin{aligned} a_{q_2}^{(2)\dagger} a_{q_1}^{(2)} &= [(\mathfrak{C}_{00} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_{q_2}^\dagger)_1] \\ &\quad \times [(\mathfrak{C}_{00} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_{q_1})_1] \\ &\quad + (\mathbb{I} \otimes \mathbb{I} + \mathcal{P}_{21}) [(\mathfrak{C}_{10} \otimes \mathfrak{s}_{q_2}^\dagger)_2 \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1] \\ &\quad \times [(\mathfrak{C}_{01} \otimes \mathfrak{s}_{q_1})_2 \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1], \end{aligned} \quad (25)$$

or, upon expanding the product and employing again the simplification of the factor \mathcal{P}_{12} because of the preprogrammed symmetrization of two-particle states,

$$\begin{aligned} a_{q_2}^{(2)\dagger} a_{q_1}^{(2)} &= (\mathfrak{C}_{00} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_{q_2}^\dagger \mathfrak{s}_{q_1})_1 \\ &\quad + (\mathfrak{C}_{11} \otimes \mathfrak{s}_{q_2}^\dagger \mathfrak{s}_{q_1})_2 \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1 \\ &\quad + (\mathfrak{C}_{11} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_{q_2}^\dagger \mathfrak{s}_{q_1})_1. \end{aligned} \quad (26)$$

Note the action of the diagonal operator with $q_1 = q_2$ as a number operator: If only the rightmost register is occupied, it returns the same state multiplied by 1 due to the leftmost term. If both are occupied with bosons of equal q_1 , it will return twice the state due to the middle and right pieces. With these expressions in Eq. (26), we can now act the commutator over each of the three sectors of the Fock space abstracted from two-particle registers.

1. Zero-particle state

The zero-particle state is the vacuum, and the action thereon of the commutator reads

$$\begin{aligned} [a_{q_1}^{(2)}, a_{q_2}^{(2)\dagger}] |\Omega\rangle &= a_{q_1}^{(2)} a_{q_2}^{(2)\dagger} |\Omega\rangle_2 |\Omega\rangle_1 - a_{q_2}^{(2)\dagger} a_{q_1}^{(2)} |\Omega\rangle_2 |\Omega\rangle_1 \\ &= |\Omega\rangle_2 \otimes (|0\rangle \mathfrak{s}_{q_1} \mathfrak{s}_{q_2}^\dagger |0\dots 0\rangle)_1 \\ &= \delta_{q_1 q_2} |\Omega\rangle, \end{aligned} \quad (27)$$

where in the last line we used Eqs. (4) and (6).

2. One-particle states

In this case we also have a faithful representation of what is expected in second quantization:

$$\begin{aligned} [a_{q_1}^{(2)}, a_{q_2}^{(2)\dagger}] |p\rangle &= a_{q_1}^{(2)} a_{q_2}^{(2)\dagger} |\Omega\rangle_2 |1p\rangle_1 - a_{q_2}^{(2)\dagger} a_{q_1}^{(2)} |\Omega\rangle_2 |1p\rangle_1 \\ &= (|0\rangle \mathfrak{s}_{q_1} \mathfrak{s}_{q_2}^\dagger |0\dots 0\rangle)_2 \otimes |1p\rangle_1 \\ &\quad + (|0\rangle \mathfrak{s}_{q_1} |p\rangle)_2 \otimes (|1\rangle \mathfrak{s}_{q_2}^\dagger |0\dots 0\rangle)_1 \\ &\quad - |\Omega\rangle_2 (|1\rangle \mathfrak{s}_{q_2}^\dagger \mathfrak{s}_{q_1} |p\rangle)_1 \\ &= \delta_{q_1 q_2} |p\rangle + \delta_{q_1 p} |q_2\rangle - \delta_{q_1 p} |q_2\rangle \\ &= \delta_{q_1 q_2} |p\rangle; \end{aligned} \quad (28)$$

so far, these operators act as the standard commutation relations do.

3. Two-particle states

Finally, we show the action of the commutator on the doubly occupied and symmetrized (as denoted by the subindex S) two-particle register:

$$\begin{aligned} [a_{q_1}^{(2)}, a_{q_2}^{(2)\dagger}] (|1q\rangle |1p\rangle)_S &= (a_{q_1}^{(2)} a_{q_2}^{(2)\dagger} - a_{q_2}^{(2)\dagger} a_{q_1}^{(2)}) \frac{1}{\sqrt{2}} (|1q\rangle_2 |1p\rangle_1 + |1p\rangle_2 |1q\rangle_1) \\ &= -a_{q_2}^{(2)\dagger} a_{q_1}^{(2)} \frac{1}{\sqrt{2}} (|1q\rangle_2 |1p\rangle_1 + |1p\rangle_2 |1q\rangle_1) \\ &= -\frac{1}{\sqrt{2}} ((|1\rangle \mathfrak{s}_{q_2}^\dagger \mathfrak{s}_{q_1} |q\rangle)_2 |1p\rangle_1 + (|1\rangle \mathfrak{s}_{q_2}^\dagger \mathfrak{s}_{q_1} |p\rangle)_2 |1q\rangle_1 \\ &\quad + |1q\rangle_2 (|1\rangle \mathfrak{s}_{q_2}^\dagger \mathfrak{s}_{q_1} |p\rangle)_1 + |1p\rangle_2 (|1\rangle \mathfrak{s}_{q_2}^\dagger \mathfrak{s}_{q_1} |q\rangle)_1) \\ &= -\frac{1}{\sqrt{2}} (\delta_{q_1 q} |q_2 p\rangle + \delta_{q_1 p} |q_2 q\rangle + \delta_{q_1 p} |q q_2\rangle + \delta_{q_1 q} |p q_2\rangle) \\ &= -\delta_{q_1 q} |q_2 p\rangle_S - \delta_{q_1 p} |q_2 q\rangle_S. \end{aligned} \quad (29)$$

As expected due to the boundary term in Eq. (12), in which the creation operator acting over two occupied registers is by necessity defined to be 0, a boundary mismatch appears in this calculation, that in textbook field theory should yield $\delta_{q_1 q_2}(|1q\rangle|1p\rangle)_S$ instead of the outcome of Eq. (29).

If a third, unoccupied register is added to the memory, the commutator acting over two-particle states can then be faithfully reproduced. This leads us to generalizing to n particles in Sec. IV A.

IV. IMPLEMENTATION OF N -PARTICLE CREATION AND ANNIHILATION OPERATORS

A. Commuting operators

We are now ready to explain the general case for a large memory bank containing an unspecified, variable number of particles. In this section, we present the generalization of the preceding codifications in Secs. II and III to the n -register case.

Consider scalar particles with only their momentum as quantum number, with such momenta codified in n_p qubits. The null ket is taken as a reference and does not represent any valid momentum:

$$|0\dots 0\rangle \equiv \underbrace{|0\rangle_{P\dots} |0\rangle_P}_{n_p} \quad (30)$$

Consider also presence and absence qubits initially in the state $|0\rangle_{P/A}$: The n -register vacuum is then written as

$$\begin{aligned} |\Omega\rangle &\equiv |\Omega\rangle_n \otimes \dots \otimes |\Omega\rangle_1 \\ &= \underbrace{(|0\rangle_{P/A}|0\dots 0\rangle)_n \otimes \dots \otimes (|0\rangle_{P/A}|0\dots 0\rangle)_1}_n \end{aligned} \quad (31)$$

Generally, operators acting over n registers consist of projection and control operators $\mathfrak{C}_{ij} = |i\rangle\langle j|$ defined in Eq. (5) over the presence and absence qubits and set and scrap operators \mathfrak{s}^\dagger , \mathfrak{s} over qubits representing quantum numbers as defined in Eqs. (7) and (8). For example, an operator that creates a particle in the first register in an n -register quantum computer is

$$a_{p,1}^{(n)\dagger} \equiv (\mathfrak{C}_{00} \otimes \mathbf{i})_n \dots (\mathfrak{C}_{00} \otimes \mathbf{i})_2 (\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger)_1; \quad (32)$$

indeed,

$$\begin{aligned} a_{p,1}^{(n)\dagger} |\Omega\rangle &= (\mathfrak{C}_{00}|0\rangle_{P/A} \mathbf{i}|0\dots 0\rangle)_n \dots (\mathfrak{C}_{10}|0\rangle_{P/A} \mathfrak{s}_p^\dagger|0\dots 0\rangle)_1 \\ &= (|0\rangle_{P/A}|0\dots 0\rangle)_n \dots (|1\rangle_{P/A}|p\rangle)_1. \end{aligned} \quad (33)$$

We next define the generalization of creation and annihilation operators for an arbitrary number of particles. Note that in Eq. (12) the two-register operator

$$\mathcal{S}_2 = \frac{1}{\sqrt{2}}(\mathbb{I} \otimes \mathbb{I} + \mathcal{P}_{21}) \quad (34)$$

is a two-particle *symmetrizer*. Thus, it is tempting (but inefficient) to generalize the expression in Eq. (14) by adding operators that create particles in each of the registers and higher-order symmetrizers, for example, in the case of three registers, changing $\mathcal{S}_2 \rightarrow \mathbb{I} \otimes \mathcal{S}_2$ and introducing a three-particles symmetrizer \mathcal{S}_3 :

$$\begin{aligned} a_q^\dagger|_3 &\equiv \mathbb{I}^{\otimes 3} \cdot (\mathfrak{C}_{00} \otimes \mathbf{i})_3 \otimes (\mathfrak{C}_{00} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_1 \\ &+ \mathbb{I} \otimes \mathcal{S}_2 \cdot (\mathfrak{C}_{00} \otimes \mathbf{i})_3 \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_2 \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1 \\ &+ \mathcal{S}_3 \cdot (\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_3 \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_2 \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1; \end{aligned} \quad (35)$$

this generalization would be correct if applied to unsymmetrized two-particle states, but in the case of symmetric states it produces extra contributions:

$$\begin{aligned} a_q^\dagger|00\rangle_3 (|1p_1\rangle_2 |1p_2\rangle_1)_S &= \mathcal{S}_3 |1q\rangle_3 (|1p_1\rangle_2 |1p_2\rangle_1)_S \\ &= \frac{1}{\sqrt{6}}(\mathbb{I}^{\otimes 3} + \mathcal{P}_{21} + \mathcal{P}_{31} + \mathcal{P}_{32} \\ &+ \mathcal{P}_{231} + \mathcal{P}_{312}) |1q\rangle (|1p_1\rangle |1p_2\rangle)_S \\ &= 2(|1q\rangle |1p_1\rangle |1p_2\rangle)_S, \end{aligned} \quad (36)$$

since we are symmetrizing again the already symmetrized part. Additionally, even if designing adequate normalizations to overcome this, numerous redundant operations would be wasted resymmetrizing already symmetric parts of the wave function.

Thus, in the case of three or more registers, symmetrization or antisymmetrization of states will be implemented in a stepwise fashion, assuming that earlier creation operations already symmetrized a state with $n-1$ particles and only the newly added one requires symmetrization with the others. Imposing symmetric states, the three-particle symmetrizer should be changed to a ‘‘step symmetrizer’’:

$$\mathcal{S}_3 \rightarrow \mathcal{S}_{3\leftarrow 2} \equiv \frac{1}{\sqrt{3}}(\mathbb{I}^{\otimes 3} + \mathcal{P}_{32} \otimes \mathbb{I}_1 + \mathcal{P}_{31} \otimes \mathbb{I}_2) \quad (37)$$

for the general case, and, omitting the identities for conciseness, we have $\mathcal{S}_{n\leftarrow n-1}$, which promotes an $(n-1)$ -particle symmetric state to an n -particle symmetric one:

$$\mathcal{S}_{n\leftarrow n-1} \equiv \frac{1}{\sqrt{n}}(\mathbb{I}^{\otimes n} + \mathcal{P}_{n(n-1)}) + \dots + \mathcal{P}_{n2} + \mathcal{P}_{n1}. \quad (38)$$

Note that, with this choice, the symmetrizer is not idempotent, but instead $\mathcal{S}_{n\leftarrow n-1}^2 = \sqrt{n}\mathcal{S}_{n\leftarrow n-1}$.

1. Implementation of creation operators

The auxiliary operator that attempts to create a particle on the j th register in an n -register memory segment now reads

$$\begin{aligned} a_{q,j}^{(n)\dagger} &= \mathbb{I}^{\otimes n-j} \otimes \mathcal{S}_{j \leftarrow (j-1)} \cdot (\mathfrak{C}_{00} \otimes \mathbf{i})_n \otimes \cdots \otimes \underbrace{(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_j}_{\text{}} \\ &\quad \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_{j-1} \otimes \cdots \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1 \\ &= \mathcal{S}_{j \leftarrow (j-1)} \cdot \mathbb{P}_0^{(n-j)} \otimes \underbrace{(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_j}_{\text{}} \otimes \mathbb{P}_{j-1}^{(j-1)}, \end{aligned} \quad (39)$$

where the identity multiplying the j -register step symmetrizer is omitted to alleviate notation and $\mathbb{P}_j^{(n)}$ is the projector

onto states with exactly j particles on an n -register quantum computer:

$$\begin{aligned} \mathbb{P}_j^{(n)} &= \mathbb{P}_0^{(n-j)} \otimes \mathbb{P}_j^{(j)} \\ &= [(\mathfrak{C}_{00} \otimes \mathbf{i})_n \otimes \cdots \otimes (\mathfrak{C}_{00} \otimes \mathbf{i})_{j+1}] \\ &\quad \otimes [(\mathfrak{C}_{11} \otimes \mathbf{i})_j \otimes \cdots \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1], \end{aligned} \quad (40)$$

which fulfil $\mathbb{P}_j^{(n)} \mathbb{P}_i^{(n)} = \delta_{ij}$ (because of the unequal number of particle presence and /absence control operators \mathfrak{C}_{00} and \mathfrak{C}_{11} for different values of i and j). The action of this operator over a register that already contains $i - 1$ particles is automatically restricted to filling $j = i$ (last particle on the stack) by its very definition, so that

$$\begin{aligned} a_{q,j}^{(n)\dagger} |\Omega\rangle^{\otimes n-i+1} \otimes (|1p_{i-1}\rangle_{i-1} \cdots |1p_1\rangle_1)_S \\ &= a_{q,j}^{(n)\dagger} \cdot \mathbb{P}_0^{(n-i+1)} \otimes \mathbb{P}_{i-1}^{(i-1)} \cdot |\Omega\rangle^{\otimes n-i+1} \otimes (|1p_{i-1}\rangle_{i-1} \cdots |1p_1\rangle_1)_S \\ &= \mathcal{S}_{j \leftarrow (j-1)} \cdot \mathbb{P}_0^{(n-j)} \otimes \underbrace{(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_j}_{\text{}} \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathbb{P}_0^{(n-i+1)} \otimes \mathbb{P}_{i-1}^{(i-1)} \cdot |\Omega\rangle^{\otimes n-i+1} \otimes (|1p_{i-1}\rangle_{i-1} \cdots |1p_1\rangle_1)_S \\ &= \delta_{i,j} |\Omega\rangle^{\otimes n-i+1} \otimes [\mathcal{S}_{i \leftarrow i-1} |1q\rangle (|1p_{i-1}\rangle_{i-1} \cdots |1p_1\rangle_1)_S] \\ &= \delta_{i,j} |\Omega\rangle_n \cdots (|1q\rangle_i |1p_{i-1}\rangle_{i-1} \cdots |1p_1\rangle_1)_S, \end{aligned} \quad (41)$$

where in the fourth line we used the orthogonality of projectors to produce a $\delta_{i,j}$.

The total creation operator that implements the field-theory requirements is the addition of auxiliary creation operators over each register:

$$\begin{aligned} a_q^{(n)\dagger} &= \sum_{i=1}^n a_{q,i}^{(n)\dagger} \\ &= \sum_{i=1}^n \mathcal{S}_{i \leftarrow (i-1)} \cdot \mathbb{P}_0^{(n-i)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_i \otimes \mathbb{P}_{i-1}^{(i-1)}; \end{aligned} \quad (42)$$

note that $\mathcal{S}_{2 \leftarrow 1} = \mathcal{S}_2$, so for the case $n = 2$ we recover the two-particle case studied earlier in Sec. III. The action of the full $a_q^{(n)\dagger}$ over an i -particle state $|\Omega\rangle_n \cdots (|1p_i\rangle_i \cdots |1p_1\rangle_1)_S$ is then

$$\begin{aligned} a_q^{(n)\dagger} |\Omega\rangle_n \cdots (|1p_i\rangle_i \cdots |1p_1\rangle_1)_S \\ &= \sum_{j=1}^n a_{q,j}^{(n)\dagger} |\Omega\rangle_n \cdots (|1p_i\rangle_i \cdots |1p_1\rangle_1)_S \\ &= \sum_{j=1}^n \delta_{j,i+1} |\Omega\rangle_n \cdots (|q\rangle_{i+1} |1p_i\rangle_i \cdots |1p_1\rangle_1)_S \\ &= |\Omega\rangle_n \cdots (|q\rangle_{i+1} |1p_i\rangle_i \cdots |1p_1\rangle_1)_S. \end{aligned} \quad (43)$$

At last, its action over an n -particle state the result is by definition zero:

$$a_p^{(n)\dagger} (|1p_n\rangle |1p_{n-1}\rangle \cdots |1p_1\rangle)_S = 0, \quad (44)$$

which is no more than a boundary condition due to the quantum computer's finite memory.

2. Implementation of annihilation operators

Annihilation operators are the adjoint of creation operators. For the auxiliary ones acting over the j register on an n -register quantum computer, we (quite directly) find

$$\begin{aligned} a_{p,j}^{(n)} &= (\mathfrak{C}_{00} \otimes \mathbf{i})_n \otimes \cdots \otimes \underbrace{(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j}_{\text{}} \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_{j-1} \\ &\quad \otimes \cdots \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1 \cdot \mathcal{S}_{j \leftarrow (j-1)} \\ &= \mathbb{P}_0^{(n-j)} \otimes \underbrace{(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j}_{\text{}} \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{S}_{j \leftarrow (j-1)}; \end{aligned} \quad (45)$$

its action over an i -particle state, with $j > i$, reads

$$\begin{aligned} a_{p,j}^{(n)} |\Omega\rangle^{\otimes n-i} (|1p_i\rangle_i \cdots |1p_1\rangle_1)_S \\ &= \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j \otimes \mathbb{P}_{j-1}^{(j-1)} |\Omega\rangle^{\otimes n-j} \\ &\quad \times [\mathcal{S}_{j \leftarrow j-1} |\Omega\rangle^{\otimes j-i} (|1p_i\rangle_i \cdots |1p_1\rangle_1)_S] = 0; \end{aligned} \quad (46)$$

the step-symmetrizer in the last line mixes the vacuum registers with the occupied ones without changing their quantum numbers, which means that $j - i$ of the j registers are always empty registers. This being so, the term in the second line, which acts upon the entire memory, will

always yield null because the first j control operators are of the form \mathfrak{C}_{i1} , which always find at least one empty register and, thus, yield null. When the index $j \leq i$, the annihilation reads

$$\begin{aligned}
a_{p,j}^{(n)} |\Omega\rangle^{\otimes n-i} (|1p_i\rangle_i \dots |1p_1\rangle_1)_S &= \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{S}_{j \leftarrow j-1} |\Omega\rangle^{\otimes n-i} (|1p_i\rangle_i \dots |1p_1\rangle_1)_S \\
&= \sqrt{j} \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j \otimes \mathbb{P}_{j-1}^{(j-1)} |\Omega\rangle^{\otimes n-i} (|1p_i\rangle_i \dots |1p_1\rangle_1)_S \\
&= \delta_{ij} \sqrt{i} \frac{\sqrt{(i-1)!}}{\sqrt{i!}} \sum_{l=1}^i |\Omega\rangle^{\otimes n-i} (\mathfrak{C}_{01}|1\rangle \otimes \mathfrak{s}_p |p_l\rangle)_i (|1p_i\rangle_{i-1} \dots |1p_{l+1}\rangle_l |1p_{l-1}\rangle_{l-1} \dots |1p_1\rangle_1)_S \\
&= \delta_{ij} \sum_{l=1}^i \delta_{pp_l} |\Omega\rangle^{\otimes n-i+1} (|1p_i\rangle_{i-1} \dots |1p_{l+1}\rangle_l |1p_{l-1}\rangle_{l-1} \dots |1p_1\rangle_1)_S. \tag{47}
\end{aligned}$$

Summing over the particle number j , we arrive at the annihilation operator

$$a_q^{(n)} = \sum_{j=1}^n a_{q,j}^{(n)} = \sum_{j=1}^n \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{S}_{j \leftarrow (j-1)}. \tag{48}$$

The action of the operator over an i -particle symmetric state is

$$\begin{aligned}
a_q^{(n)} |\Omega\rangle_n \dots |\Omega\rangle_{i+1} (|1p_i\rangle_i \dots |1p_1\rangle_1)_S &= \sum_{j=1}^n a_{q,j}^{(n)} |\Omega\rangle_n \dots |\Omega\rangle_{i+1} (|1p_i\rangle_i \dots |1p_1\rangle_1)_S \\
&= \sum_{l=1}^i \delta_{pp_l} |\Omega\rangle^{\otimes n-i+1} (|1p_i\rangle_{i-1} \dots |1p_{l+1}\rangle_l |1p_{l-1}\rangle_{l-1} \dots |1p_1\rangle_1)_S, \tag{49}
\end{aligned}$$

which is exactly what we expect from an annihilation operator acting over an i -particle symmetric state. Note that this leaves behind the initial memory piece, with i registers, occupied by only $i - 1$ particles (properly symmetrized) and precisely the last i th register vacant.

3. Number operator

We next check the adequacy of the resulting number operator¹:

$$\begin{aligned}
\widehat{N}^{(n)} &:= \sum_p a_p^{(n)\dagger} a_p^{(n)} = \sum_p \sum_{j,j'} a_{p,j}^{(n)\dagger} a_{p,j'}^{(n)} \\
&= \sum_{p,j,j'} \mathcal{S}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathbb{P}_0^{(n-j')} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{j'} \otimes \mathbb{P}_{j'-1}^{(j'-1)} \cdot \mathcal{S}_{j' \leftarrow j'-1} \\
&= \sum_j \mathcal{S}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes \left(\mathfrak{C}_{11} \otimes \sum_p \mathfrak{s}_p^\dagger \mathfrak{s}_p \right)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{S}_{j \leftarrow j-1} \\
&= \sum_j \mathcal{S}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{S}_{j \leftarrow j-1} \\
&= \sum_j \mathcal{S}_{j \leftarrow j-1} \cdot \mathbb{P}_j^{(n)} \cdot \mathcal{S}_{j \leftarrow j-1}, \tag{50}
\end{aligned}$$

¹ $\sum_p \mathfrak{s}_p^\dagger \mathfrak{s}_p = \mathbf{i} - |0\dots 0\rangle\langle 0\dots 0|$, but $|0\dots 0\rangle\langle 0\dots 0|$ can be ignored because occupied registers with $|0\dots 0\rangle$ never occur.

where the sum over j' is simplified using a $\delta_{j,j'}$ emerging from multiplication of projectors in the second line. The operator between symmetrizers is just a projector over j -occupied registers, so a permutation between terms acting over registers j and $k \in \{1, \dots, j-1\}$ gives the same operator. Thus, $\mathbb{P}_j^{(n-j)}$ and $\mathbb{I}^{\otimes n-j} \otimes \mathcal{S}_{j-j-1}$ commute, and

$$\begin{aligned} \widehat{N}^{(n)} &= \sum_j \mathcal{S}_{j \leftarrow j-1} \cdot \mathbb{P}_j^{(n)} \cdot \mathcal{S}_{j \leftarrow j-1} = \sum_j \mathcal{S}_{j \leftarrow j-1}^2 \cdot \mathbb{P}_j^{(n)} \\ &= \sum_j \sqrt{j} \mathcal{S}_{j \leftarrow j-1} \cdot \mathbb{P}_j^{(n)}, \end{aligned} \quad (51)$$

when acting over symmetric states with j' particles. We see that $N^{(n)}$ is diagonal with the expected eigenvalues j' :

$$\begin{aligned} \widehat{N}^{(n)} |\Omega\rangle_n \dots (|1p_{j'}\rangle_{j'} \dots |1p_1\rangle_1)_S \\ &= \sum_{j'} \sqrt{j'} |\Omega\rangle_n \dots \mathcal{S}_{j' \leftarrow j'-1} (|1p_{j'}\rangle_{j'} \dots |1p_1\rangle_1)_S \\ &= j' |\Omega\rangle_n \dots (|1p_{j'}\rangle_{j'} \dots |1p_1\rangle_1)_S. \end{aligned} \quad (52)$$

Having established the validity of the number operator, we can next check that $a^{(n)\dagger}$ and $a^{(n)}$ for n particles form a canonically conjugate pair.

4. Commutation relations

We now adopt Eqs. (42) and (45) as the definition of the creation and annihilation operators for n registers.

These operators satisfy the following commutation relations in the Hilbert subspace of symmetric states:

$$\begin{aligned} [a_{q_1}^{(n)}, a_{q_2}^{(n)\dagger}] &= \delta_{q_1, q_2} (\mathbb{C}_{00} \otimes \mathbf{i})_n \otimes \sum_{j=0}^{n-1} \mathbb{P}_j^{(n-1)} \\ &\quad - S_{n \leftarrow n-1} \cdot (\mathbb{C}_{11} \otimes \mathfrak{s}_{q_1}^\dagger \mathfrak{s}_{q_2})_n \otimes \mathbb{P}_{n-1}^{(n-1)} \cdot S_{n \leftarrow n-1}, \end{aligned} \quad (53)$$

the operator $\sum_{j=0}^{n-1} \mathbb{P}_j^{(n-1)}$ projects over properly filled memories, in which registers are filled from right to left. Thus, $\mathbb{I}^{(n-1)} - \sum_{j=0}^{n-1} \mathbb{P}_j^{(n-1)}$, with $\mathbb{I}^{(n-1)}$ the identity over $(n-1)$ registers, is zero over properly filled memories, and we can write

$$\begin{aligned} \mathcal{A}_{i \leftarrow i-1} (|1p_i\rangle_i \dots |1p_1\rangle_1)_A \\ &= \frac{1}{\sqrt{i}} (\mathbb{I}^{\otimes n} - \mathcal{P}_{i(i-1)} - \dots - \mathcal{P}_{i1}) \sum_{\sigma} (-1)^{\eta_{\sigma}} \sigma (|1p_i\rangle_i \dots |1p_1\rangle_1) \\ &= \frac{1}{\sqrt{i}} \left[\sum_{\sigma} (-1)^{\eta_{\sigma}} \sigma - \sum_{\sigma} (-1)^{\eta_{\sigma}} \mathcal{P}_{i(i-1)} \sigma - \dots - \sum_{\sigma} (-1)^{\eta_{\sigma}} \mathcal{P}_{i1} \sigma \right] |1p_i\rangle_i \dots |1p_1\rangle_1. \end{aligned} \quad (57)$$

$$\begin{aligned} [a_{q_1}^{(n)}, a_{q_2}^{(n)\dagger}]_{\text{properly filled}} \\ &= \delta_{q_1, q_2} (\mathbb{C}_{00} \otimes \mathbf{i})_n \otimes \mathbb{I}^{(n-1)} \\ &\quad - S_{n \leftarrow n-1} \cdot (\mathbb{C}_{11} \otimes \mathfrak{s}_{q_1}^\dagger \mathfrak{s}_{q_2})_n \otimes \mathbb{P}_{n-1}^{(n-1)} \cdot S_{n \leftarrow n-1}. \end{aligned} \quad (54)$$

The first term on the right-hand side of Eq. (53) is, in this notation, what one would expect of canonical commutation relations, the identity in a Fock space of up to $n-1$ particles (the \mathbb{C}_{00} control guarantees that the n th register is empty).

The second deviates from second quantization and emerges due to the quantum computer's finite memory: It vanishes if less than n particles occupy the memory. As a corollary thereof, the canonical commutation relations are exactly satisfied for symmetric states of up to $n-1$ particles (in an n -register quantum computer) when only the first term contributes.

The demonstration of Eq. (53), that requires a few bookkeeping steps, is presented in Appendix B.

B. Anticommuting operators

With the experience of the boson operators in Sec. IV A, it is now easy to implement anticommuting ones and direct (if tedious) to derive their properties. The most notable difference is that the recursive step symmetrizers of Eq. (38) should be changed to step antisymmetrizers, which (unsurprisingly) read

$$\mathcal{A}_{n \leftarrow n-1} = \frac{1}{\sqrt{n}} (\mathbb{I}^{\otimes n} - \mathcal{P}_{n(n-1)} - \mathcal{P}_{n(n-2)} - \dots - \mathcal{P}_{n2} - \mathcal{P}_{n1}) \quad (55)$$

that antisymmetrize any dynamically added particle with all the previously filled ones.

Totally antisymmetric states can be written as

$$(|1p_i\rangle_i \dots |1p_1\rangle_1)_A = \sum_{\sigma} (-1)^{\eta_{\sigma}} \sigma (|1p_i\rangle_i \dots |1p_1\rangle_1), \quad (56)$$

where σ runs over the permutations of i elements and η_{σ} is the parity of the permutation σ : $(-1)^{\eta_{\sigma}} = 1$ if the permutation is even, and (-1) if odd. The action of the step antisymmetrizer over this state is

By the rearrangement lemma, no two $\mathcal{P}_{ij}\sigma$ permutations are equal, so the sums $\sum_{\sigma} \mathcal{P}_{ij}\sigma$ cover all possible permutations. Also, since \mathcal{P}_{ij} is a simple permutation, the parity of $\mathcal{P}_{ij}\sigma$ is the negative of the parity of σ . Using the same index, σ , for all the new permutations, we have that all the terms in Eq. (57) are identical, so

$$\begin{aligned} & \mathcal{A}_{i \leftarrow i-1}(|1p_i\rangle_i \dots |1p_1\rangle_1)_A \\ &= \frac{1}{\sqrt{i}} \left[\sum_{\sigma} (-1)^{n_{\sigma}} \sigma + \sum_{\sigma} (-1)^{n_{\sigma}} \sigma + \dots + \sum_{\sigma} (-1)^{n_{\sigma}} \sigma \right] \\ & \quad \times |1p_i\rangle_i \dots |1p_1\rangle_1 \\ &= \sqrt{i} (|1p_i\rangle_i \dots |1p_1\rangle_1)_A. \end{aligned} \quad (58)$$

1. Implementation of the creation operators

We start again by an auxiliary operator that creates a particle precisely at the j th register in an n -register quantum computer, which now reads

$$\begin{aligned} b_{q,j}^{(n)\dagger} &= \mathbb{I}^{\otimes n-j} \otimes \mathcal{A}_{j \leftarrow j-1} \cdot (\mathfrak{C}_{00} \otimes \mathbf{i})_n \otimes \dots \otimes \underbrace{(\mathfrak{C}_{10} \otimes \mathfrak{s}_p^{\dagger})_j}_{\text{creation}} \\ & \quad \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_{j-1} \otimes \dots \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1 \\ &= \mathcal{A}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes \underbrace{(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^{\dagger})_j}_{\text{creation}} \otimes \mathbb{P}_{j-1}^{(j-1)}. \end{aligned} \quad (59)$$

Its action over a register that contains i particles is similar to the commuting case, but symmetric states should be changed to antisymmetric ones:

$$\begin{aligned} b_{q,j}^{(n)\dagger} |\Omega\rangle^{\otimes n-i+1} (|1p_{i-1}\rangle_{i-1} \dots |1p_1\rangle_1)_A \\ &= \delta_{i,j} |\Omega\rangle^{\otimes n-i+1} \otimes [\mathcal{A}_{i \leftarrow i-1} |1q\rangle_i (|1p_{i-1}\rangle_{i-1} \dots |1p_1\rangle_1)_A] \\ &= \delta_{i,j} |\Omega\rangle^{\otimes n-i} (|1q\rangle_i |1p_{i-1}\rangle_{i-1} \dots |1p_1\rangle_1)_A, \end{aligned} \quad (60)$$

where the Kronecker delta appears due to the same reasons it was generated among commuting operators. The field-theoretic creation operator is implemented by the addition of the creation operators over each register:

$$\begin{aligned} b_{p,j}^{(n)} |\Omega\rangle^{\otimes n-i} (|1p_i\rangle_i \dots |1p_1\rangle_1)_A &= \sqrt{j} \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j \otimes \mathbb{P}_{j-1}^{(j-1)} |\Omega\rangle^{\otimes n-i} (|1p_i\rangle_i \dots |1p_1\rangle_1)_A \\ &= \sqrt{j} \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j \otimes \mathbb{P}_{j-1}^{(j-1)} |\Omega\rangle^{\otimes n-i} \frac{\sqrt{(i-1)!}}{\sqrt{i!}} \left[|1p_i\rangle_i (|1p_{i-1}\rangle_{i-1} \dots |1p_1\rangle_1)_A \right. \\ & \quad \left. - \sum_{l=1}^{i-1} |1p_l\rangle_l (|1p_{i-1}\rangle_{i-1} \dots |1p_l\rangle_l \dots |1p_1\rangle_1)_A \right] \\ &= \delta_{ij} \sum_{l=1}^i (-1)^{i-l} |\Omega\rangle^{\otimes n-i} (\mathfrak{C}_{01} |1\rangle \otimes \mathfrak{s}_p |p_l\rangle)_i (|1p_i\rangle_i \dots |1p_{l+1}\rangle_{l+1} |1p_{l-1}\rangle_{l-1} \dots |1p_1\rangle_1)_A \\ &= \delta_{ij} \sum_{l=1}^i (-1)^{i-l} \delta_{pp_l} |\Omega\rangle^{\otimes n-i+1} (|1p_i\rangle_i \dots |1p_{l+1}\rangle_{l+1} |1p_{l-1}\rangle_{l-1} \dots |1p_1\rangle_1)_A. \end{aligned} \quad (65)$$

$$\begin{aligned} b_q^{(n)\dagger} &= \sum_{j=1}^n b_{q,j}^{(n)\dagger} \\ &= \sum_{j=1}^n \mathcal{A}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes \underbrace{(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^{\dagger})_j}_{\text{creation}} \otimes \mathbb{P}_{j-1}^{(j-1)}; \end{aligned} \quad (61)$$

its action over an i -particle antisymmetric state $|\Omega\rangle_n \dots (|1p_i\rangle_i \dots |1p_1\rangle_1)_A$ is

$$\begin{aligned} b_q^{(n)\dagger} |\Omega\rangle^{\otimes n-i+1} (|1p_i\rangle_i \dots |1p_1\rangle_1)_A \\ &= \sum_{j=1}^n b_{q,j}^{(n)\dagger} |\Omega\rangle^{\otimes n-i+1} (|1p_i\rangle_i \dots |1p_1\rangle_1)_A \\ &= \sum_{j=1}^n \delta_{j,i+1} |\Omega\rangle^{\otimes n-i} (|q\rangle_{i+1} |1p_i\rangle_i \dots |1p_1\rangle_1)_A \\ &= |\Omega\rangle^{\otimes n-i} (|q\rangle_{i+1} |1p_i\rangle_i \dots |1p_1\rangle_1)_A. \end{aligned} \quad (62)$$

Finally, we need to adopt a boundary condition upon reaching a full memory, and we again choose

$$b_q^{(n)\dagger} (|1p_n\rangle_n \dots |1p_1\rangle_1)_A = 0. \quad (63)$$

2. Implementation of the annihilation operators

By Hermitian conjugation of the creation operators in Eq. (59), we directly obtain

$$\begin{aligned} b_{p,j}^{(n)} &= (\mathfrak{C}_{00} \otimes \mathbf{i})_n \otimes \dots \otimes \underbrace{(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j}_{\text{annihilation}} \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_{j-1} \\ & \quad \otimes \dots \otimes (\mathfrak{C}_{11} \otimes \mathbf{i})_1 \cdot \mathcal{A}_{j \leftarrow (j-1)} \\ &= \mathbb{P}_0^{(n-j)} \otimes \underbrace{(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j}_{\text{annihilation}} \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{A}_{j \leftarrow (j-1)}, \end{aligned} \quad (64)$$

whose action over an i -particle state with $j > i$ is null by the same reasons argued in the commuting case; cf. Eq. (46). The case with $j \leq i$ is similar, but some care is due for minus signs coming from antisymmetry:

As in the commuting case, the action of $(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_i$ “extracts” all the different momenta from $(|1p_i\rangle_i \dots |1p_1\rangle_1)_A$. To complete the discussion, we can give the annihilation operator in register-operator notation as

$$b_q^{(n)} = \sum_{j=1}^n b_{q,j}^{(n)} = \sum_{j=1}^n \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{A}_{j \leftarrow j-1}, \quad (66)$$

which is the same expression as for a boson operator upon change of step symmetrizers for step antisymmetrizers. We note again that $a_q^{(n)}$ is defined to act over symmetric states, whereas $b_q^{(n)}$ is the appropriate one on antisymmetric states. The action of the annihilation operator over an i -particle antisymmetric state is

$$\begin{aligned} b_q^{(n)} |\Omega\rangle_n \dots |\Omega\rangle_{i+1} (|1p_i\rangle_i \dots |1p_1\rangle_1)_A &= \sum_{j=1}^n b_{q,j}^{(n)} |\Omega\rangle_n \dots |\Omega\rangle_{i+1} (|1p_i\rangle_i \dots |1p_1\rangle_1)_A \\ &= \sum_j \delta_{ij} \sum_{l=1}^i (-1)^{i-l} \delta_{pp_l} |\Omega\rangle^{\otimes n-i+1} (|1p_i\rangle_{i-1} \dots |1p_{l+1}\rangle_l |1p_{l-1}\rangle_{l-1} \dots |1p_1\rangle_1)_A \\ &= \delta_{pp_l} \sum_{l=1}^i (-1)^{i-l} |\Omega\rangle^{\otimes n-i+1} (|1p_i\rangle_{i-1} \dots |1p_{l+1}\rangle_l |1p_{l-1}\rangle_{l-1} \dots |1p_1\rangle_1)_A. \end{aligned} \quad (67)$$

3. Anticommutation relations

Creation and annihilation operators then fulfill the anticommutation relations

$$\begin{aligned} \{b_{q_1}^{(n)}, b_{q_2}^{(n)\dagger}\} &= \delta_{q_1, q_2} (\mathfrak{C}_{00} \otimes \mathbf{i})_n \otimes \sum_{j=0}^{n-1} \mathbb{P}_j^{(n-1)} \\ &+ \mathcal{A}_{n \leftarrow n-1} \cdot (\mathfrak{C}_{11} \otimes \mathfrak{s}_{q_1}^\dagger \mathfrak{s}_{q_2})_n \otimes \mathbb{P}_{n-1}^{(n-1)} \cdot \mathcal{A}_{n \leftarrow n-1}, \end{aligned} \quad (68)$$

where $\mathbb{P}_j^{(n)}$ is the projector over states with j particles on an n -register section of memory and $\mathbb{I}^{(n-1)}$ is the identity over $(n-1)$ registers. The demonstration is somewhat cumbersome, so as in the case of bosons it is relegated to Appendix B.

Again, we see that the first term reproduces the canonical second quantization relation, and the second one is a boundary term appearing only when the memory is full.

V. EXPONENTIATION OF PARTICLE-NUMBER PRESERVING OPERATORS

In this section, we are going to exemplify the encoding developed with simple few-body Hamiltonian evolution. The dynamics is implemented in terms of unitary operators, which are Lie exponentials of Hermitian operators written in terms of creation and annihilation operators. The two basic cases that we examine are the free evolution and the momentum-exchange kernel without particle-number variation.

We shall concentrate on fermion operators b and b^\dagger and only on occasion quote analogous results for bosons.

A. Free-evolution operator

The free-evolution operator for fermion particles on an N -register quantum computer is

$$\begin{aligned} \mathcal{U}_{11}^f(\Delta t) &= \exp\left(-i\Delta t \sum_q E_q b_q^{(n)\dagger} b_q^{(n)}\right) \\ &= \exp\left(-i\Delta t \sum_q E_q \sum_{j,j'} b_{q,j}^{(n)\dagger} b_{q,j'}^{(n)}\right), \end{aligned} \quad (69)$$

while the corresponding operator for bosons, $\mathcal{U}_{11}^b(\Delta t)$, is obtained by replacing fermion for boson operators. (The 11 subindex indicates that both ket and bra contain exactly one particle.) To implement the exponential, we should first write it in terms of set and scrap operators, by Eq. (66):

$$\begin{aligned} b_{q,j'}^{(n)\dagger} b_{q,j}^{(n)} &= \mathcal{A}_{j' \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j')} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_{j'} \\ &\otimes \mathbb{P}_{j'-1}^{(j'-1)} \cdot \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_q)_j \\ &\otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{A}_{j \leftarrow j-1} \\ &= \delta_{j',j} \mathcal{A}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_q)_j \\ &\otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{A}_{j \leftarrow j-1}, \end{aligned} \quad (70)$$

where the tensor product of identities multiplying the step antisymmetrizers is omitted. For boson operators,

$$\begin{aligned} a_{q,j'}^{(n)\dagger} a_{q,j}^{(n)} &= \delta_{j',j} \mathcal{S}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_q)_j \\ &\otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{S}_{j \leftarrow j-1}. \end{aligned} \quad (71)$$

The operator conserves the number of particles (it has neither \mathfrak{C}_{01} nor \mathfrak{C}_{10} terms), and the set-scrap pair selects configurations with a momentum q in the j th register and discards the rest. The subsequent antisymmetrizer (symmetrizer) to its left then ensures a superposition over all registers from 1 to j so that each may have that momentum q . The same effect can be achieved by applying the operator $(\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_q)$ to each register, assuming the memory is already antisymmetrized (symmetrized):

$$\begin{aligned} & \mathcal{A}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_q)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{A}_{j \leftarrow j-1} \\ &= \sum_{k=0}^{j-1} \mathbb{P}_k^{(n-j+k)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_q)_{j-k} \otimes \mathbb{P}_{j-1-k}^{(j-1-k)} \cdot \frac{\mathcal{A}_{j \leftarrow j-1}}{\sqrt{j}}, \end{aligned} \quad (72)$$

and, correspondingly,

$$\begin{aligned} & \mathcal{S}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_q)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{S}_{j \leftarrow j-1} \\ &= \sum_{k=0}^{j-1} \mathbb{P}_k^{(n-j+k)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_q)_{j-k} \otimes \mathbb{P}_{j-1-k}^{(j-1-k)} \cdot \frac{\mathcal{S}_{j \leftarrow j-1}}{\sqrt{j}}, \end{aligned} \quad (73)$$

note that both expressions are valid, because $\sum_{k=0}^{i-1} \mathbb{P}_k^{(n-i+k)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_q)_{i-k} \otimes \mathbb{P}_{i-1-k}^{(i-1-k)}$ is a symmetry-preserving operator.

Because number-preserving operators over different registers commute, the Baker-Campbell-Hausdorff formula becomes trivial, $e^{A+B} = e^A e^B$, and the free-evolution operator from Eq. (69) can then be factorized:

$$\begin{aligned} \mathcal{U}_{11}^f(\Delta t) &= \exp \left(-i\Delta t \sum_q E_q \sum_{j'} \delta_{j'j} \mathcal{A}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_q)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{A}_{j \leftarrow j-1} \right) \\ &= \prod_j \exp \left(\sum_{k=0}^{j-1} \mathbb{P}_k^{(n-j+k)} \otimes \left(\mathfrak{C}_{11} \otimes (-i\Delta t) \sum_q E_q \mathfrak{s}_q^\dagger \mathfrak{s}_q \right)_{j-k} \otimes \mathbb{P}_{j-1-k}^{(j-1-k)} \cdot \frac{\mathcal{A}_{j \leftarrow j-1}}{\sqrt{j}} \right). \end{aligned} \quad (74)$$

Using the symmetry-preserving property of the exponent,

$$\mathcal{U}_{11}^f(\Delta t) = \prod_j \left\{ \mathbb{I}^{(n)} + \sum_{m=1}^{\infty} \left[\sum_k \mathbb{P}_k^{(n-j+k)} \otimes \left(\mathfrak{C}_{11} \otimes (-i\Delta t) \sum_q E_q \mathfrak{s}_q^\dagger \mathfrak{s}_q \right)_{j-k} \otimes \mathbb{P}_{j-1-k}^{(j-1-k)} \right]^m \cdot \frac{\mathcal{A}_{j \leftarrow j-1}}{\sqrt{j}} \right\}, \quad (75)$$

where we have separated the $m = 0$ term of the sum. Because of the step antisymmetrizer \mathcal{A} , this operator reduces to the identity when applied to symmetric states. But this is never the case, since by construction the fermion states are antisymmetrized upon creation of each particle. Thus, with all input states being antisymmetric, \mathcal{A} becomes redundant, and the operator may be written down as a double product:

$$\begin{aligned} \mathcal{U}_{11}^f(\Delta t) &= \prod_j \left\{ \mathbb{I}^{(n)} + \sum_{m=1}^{\infty} \left[\sum_k \mathbb{P}_k^{(n-j+k)} \otimes \left(\mathfrak{C}_{11} \otimes (-i\Delta t) \sum_q E_q \mathfrak{s}_q^\dagger \mathfrak{s}_q \right)_{j-k} \otimes \mathbb{P}_{j-1-k}^{(j-1-k)} \right]^m \right\} \\ &= \prod_{jk} \exp \left[\mathbb{P}_k^{(n-j+k)} \otimes \left(\mathfrak{C}_{11} \otimes (-i\Delta t) \sum_q E_q \mathfrak{s}_q^\dagger \mathfrak{s}_q \right)_{j-k} \otimes \mathbb{P}_{j-1-k}^{(j-1-k)} \right], \end{aligned} \quad (76)$$

where in the last step we used commutativity of terms with different index k .

The exponential can be further simplified using the idempotence of projector operators $(\mathbb{P}_l^{(m)})^n = \mathbb{P}_l^{(m)}$ and of diagonal control operators $(\mathfrak{C}_{11})^n = \mathfrak{C}_{11}$:

$$\mathcal{U}_{11}^f(\Delta t) = \prod_{ik} \left\{ \sum_{\substack{l=0 \\ l \neq i}} \mathbb{P}_n^{(l)} + \mathbb{P}_k^{(n-i+k)} \otimes (\mathfrak{C}_{11} \otimes \mathcal{U}(\Delta t))_{i-k} \otimes \mathbb{P}_{i-1-k}^{(i-1-k)} \right\}, \quad (77)$$

where we have defined an auxiliary register-level exponentiation (we can think of it as a composite gate) leading to an operator in terms of set and scrap operators which formally looks like an evolution operator itself:

$$\mathcal{U}_{11}(\Delta t) \equiv \exp \left[-i\Delta t \sum_q E_q \mathfrak{s}_q^\dagger \mathfrak{s}_q \right]. \quad (78)$$

The products over j and k can be transformed in a sum over j using the orthogonality properties of projectors, finally obtaining

$$\mathcal{U}_{11}^f(\Delta t) = \mathbb{P}_0^{(n)} + \sum_{i=1}^n \mathbb{P}_0^{(n-i)} \prod_{k=i}^1 \otimes (\mathfrak{C}_{11} \otimes \mathfrak{U}_{11}(\Delta t))_k; \quad (79)$$

for the case of bosons, the memory is instead symmetric, but the operator has the same formal structure. In cases in which both fermions and bosons need to be taken into account, each type of particle would have a separately assigned memory space. We will return to this issue in Sec. VIB below.

The sums over i in expressions such as Eq. (79) are complete decompositions and guarantee that no state will be projected out to 0 (which is not possible in unitary evolution) in spite of all the control and projector operators present in the expression which could apparently make it vanish.

1. Gate-level implementation and costs

Equation (79) has a clear interpretation in terms of quantum circuits; see Fig. 1. Therein, each square represents an implementation of Eq. (78) controlled by the relevant presence qubit. Each one-body operator acts when the corresponding presence or absence qubit is 1, so that the corresponding register is occupied.

We now consider a direct implementation to give upper bounds in terms of CNOTs and single-qubit gates. From Eq. (79), it is clear that there are $\mathcal{O}(n)$ \mathfrak{U}_{11} gate applications, with n the number of registers, each being an exponential of $\mathcal{O}(N_p)$ diagonal terms, one for each codified momenta. As diagonal transformations commute and, therefore, can be applied one at a time, one possible implementation consists on looping over the different momenta and controlling on each, applying the corresponding phase transformation. Since momenta is codified in $\mathcal{O}(\log_2 N_p)$ qubits, applying a gate controlled on an specific momentum costs $\mathcal{O}(\log_2 N_p)$ CNOTs and

single-qubit gates. Finally, as there are $\mathcal{O}(N_p)$ momenta, the total cost scales as $\mathcal{O}(nN_p \log_2 N_p)$ CNOTs and single-qubit gates. See Chap. 4 in Ref. [36] for a discussion of how to implement general controlled transformations.

B. Momentum-exchange operator

We now consider one of the simplest unitary operators whose action changes the values stored in registers. Four-body terms are a staple in various fields with instantaneous potentials or contact terms exchanging momentum among a conserved number of particles and allow for contacting with nonrelativistic, potential quantum mechanics at fixed particle number. In this subsection, we consider this number-conserving part of quadrilinear operators, largely to illustrate the control of momentum exchanges.

We adopt a discretization of p with a set of momenta equispaced by Δ (not to be confused with Δt ; see glossary) which sets the scale of the computation and is an infrared cutoff akin to the lattice side length in a classical gauge theory computation. The coded momenta can be spelled out as

$$\{p_{\min}, \dots, p_{-1}, p_0 = 0, p_1, \dots, p_{\max}\} \quad (80)$$

with $p_l = p_0 + l\Delta = l\Delta$, $l \in \{\Lambda_{\min}, \dots, \Lambda_{\max}\}$, and $|\{\Lambda_{\min}, \dots, \Lambda_{\max}\}| = N_p$; for simplicity, we keep $-\Lambda_{\min} = \Lambda_{\max} = \Lambda$. Negative values of p are tagged in the storage by assigning a negative sign to the integer subindex.

The four-body ($2 \rightarrow 2$) momentum-exchange term between fermion particles exponentiates to the unitary operator

$$\begin{aligned} \mathcal{U}_{22}^f(\Delta t) &= \exp \left[-i\Delta t \sum_{\xi=-\Lambda}^{\Lambda} \left(\lambda_{\xi} \sum_{q,p} b_{q+\xi\Delta}^{\dagger} b_p^{\dagger} b_{p+\xi\Delta} b_q \right) \right] \\ &\equiv \exp(-i\Delta t h_{22}^f), \end{aligned} \quad (81)$$

where λ_{ξ} controls the probability or intensity of the momentum exchange and the limits of sums over q and p should be chosen so that $s = q + \xi\Delta = q + p_{\xi}$ and $r = p + \xi\Delta = p + p_{\xi}$ are codified (both positive and

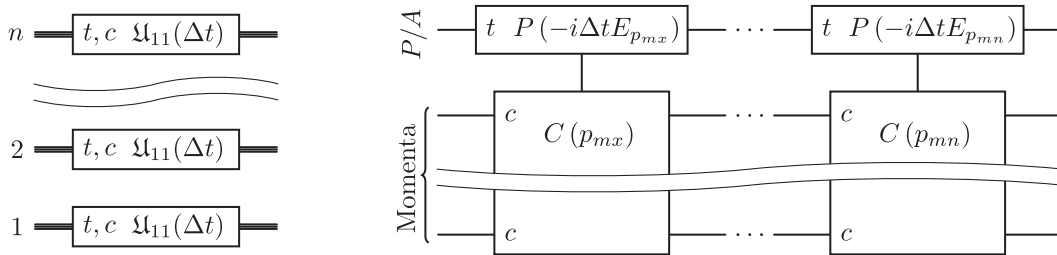


FIG. 1. Left: schematic circuit implementing the free-evolution term of Eq. (69); thick lines represent registers. Right: direct implementation of \mathfrak{U}_{11} in terms of controlled phase gates: $C(p_i)$ indicates that the corresponding phase gate applies only when p_i is stored on memory; p_{mx} is the maximum momentum on the grid, $l = \Lambda_{\max}$ on Eq. (80), while p_{mn} is the minimum, $l = \Lambda_{\min}$. On a gate, t denotes target registers, whereas c/ac denotes control and anticontrol registers; often, only presence qubits are used as controls.

negative values of ξ are included):

$$h_{22}^f = \sum_{\xi=-\Lambda}^{\Lambda} \sum_{\frac{p}{\Delta}=\max(-\Lambda, -\Lambda-\xi)}^{\min(\Lambda, \Lambda-\xi)} \sum_{\frac{q}{\Delta}=\max(-\Lambda, -\Lambda+\xi)}^{\min(\Lambda, \Lambda+\xi)} \lambda_{\xi}(b_{q+p\xi}^{\dagger} b_p^{\dagger} b_{p+p\xi} b_q) = \sum_{\xi qp} (b_s^{\dagger} b_p^{\dagger} b_r b_q \lambda_{\xi}) \Big|_{\substack{s=q+p\xi \\ r=p+p\xi}}. \quad (82)$$

Each term in the sum, using Eqs. (61) and (66) and the projectors $\mathbb{P}_i^{(j)}$ introduced under Eq. (39), is

$$\begin{aligned} b_p^{(n)\dagger} &= \sum_{j=1}^n \mathcal{A}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_p^{\dagger})_j \otimes \mathbb{P}_{j-1}^{(j-1)}, \\ b_p^{(n)} &= \sum_{j=1}^n \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{A}_{j \leftarrow j-1}, \end{aligned} \quad (83)$$

and we have

$$b_s^{\dagger} b_p^{\dagger} b_r b_q = \sum_{j=1}^{n-1} \mathcal{A}_{j+1 \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j+1)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_s^{\dagger} \mathfrak{s}_q)_{j+1} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_p^{\dagger} \mathfrak{s}_r)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{A}_{j+1 \leftarrow j-1}, \quad (84)$$

where $\mathcal{A}_{j+1 \leftarrow j-1} = \mathcal{A}_{j+1 \leftarrow j} \mathcal{A}_{j \leftarrow j-1}$. By Eq. (72) applied twice, one for each of these two step-antisymmetrizer operators to the left, two different contributions are generated, one with $(\mathfrak{C}_{11} \otimes \mathfrak{s}_s^{\dagger} \mathfrak{s}_q)$ to the left of $(\mathfrak{C}_{11} \otimes \mathfrak{s}_p^{\dagger} \mathfrak{s}_r)$ and another with the opposite ordering:

$$\begin{aligned} b_s^{\dagger} b_p^{\dagger} b_r b_q &= \sum_{j=1}^{n-1} \sum_{k=0}^{j-1} \sum_{l=0}^k \mathbb{P}_l^{(n-(j+1)+l)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_s^{\dagger} \mathfrak{s}_q)_{j+1-l} \otimes \mathbb{P}_{k-l}^{(k-l)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_p^{\dagger} \mathfrak{s}_r)_{j-k} \otimes \mathbb{P}_{j-1-k}^{(j-1-k)} \cdot \frac{\mathcal{A}_{j+1 \leftarrow j-1}}{\sqrt{j(j+1)}} \\ &+ (pr) \leftrightarrow (sq). \end{aligned} \quad (85)$$

The bookkeeping becomes less cumbersome if we use other summation indices— $l \rightarrow 1-i$, $k \rightarrow j-k$, and $i \rightarrow i-k$:

$$\begin{aligned} b_s^{\dagger} b_p^{\dagger} b_r b_q &= \sum_{j=1}^{n-1} \sum_{k=1}^j \sum_{i=1+k}^{j+1} \mathbb{P}_{j+1-i}^{(n-i)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_s^{\dagger} \mathfrak{s}_q)_i \otimes \mathbb{P}_{i-k-1}^{(i-k-1)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_p^{\dagger} \mathfrak{s}_r)_k \otimes \mathbb{P}_{k-1}^{(k-1)} \cdot \frac{\mathcal{A}_{j+1 \leftarrow j-1}}{\sqrt{j(j+1)}} \\ &+ (pr) \leftrightarrow (sq). \end{aligned} \quad (86)$$

Before exponentiation, it is convenient to separate projectors acting on presence qubits and operators acting on momentum qubits:

$$\begin{aligned} h_{22}^f &= \sum_{j=1}^{n-1} \sum_{k=1}^j \sum_{l=1+k}^{j+1} \mathbb{P}_{j+1-i}^{(n-l)} \otimes \{(\mathfrak{C}_{11} \otimes)^{\otimes l-k+1}\} \\ &\times \left\{ \sum_{\xi} \lambda_{\xi} \left(\sum_{pq} (\mathfrak{s}_s^{\dagger} \mathfrak{s}_q)_l \otimes \mathbf{i} \otimes \cdots \otimes (\mathfrak{s}_p^{\dagger} \mathfrak{s}_r)_k + \sum_{pq} (\mathfrak{s}_p^{\dagger} \mathfrak{s}_r)_l \otimes \mathbf{i} \otimes \cdots \otimes (\mathfrak{s}_s^{\dagger} \mathfrak{s}_q)_k \right) \right\} \otimes \mathbb{P}_{k-1}^{(k-1)} \cdot \frac{\mathcal{A}_{j+1 \leftarrow j-1}}{\sqrt{j(j+1)}} \\ &= \sum_{j=1}^{n-1} \sum_{k=1}^j \sum_{l=1+k}^{j+1} \mathbb{P}_{j+1-l}^{(n-i)} \otimes \{(\mathfrak{C}_{11} \otimes)^{\otimes l-k+1}\} \{ \mathfrak{h}_{22,l,k} \} \otimes \mathbb{P}_{k-1}^{(k-1)} \cdot \frac{\mathcal{A}_{j+1 \leftarrow j-1}}{\sqrt{j(j+1)}}. \end{aligned} \quad (87)$$

The notation $\{.\}\{.\}$ reminds us that we have separated the part of the operator that acts on momentum and the part that acts on presence or absence to different terms. In the third line, we defined a multiregister-level implementation of the potential that exclusively acts on the momentum part of the registers between the l th and the k th (identities over momentum qubits have being omitted):

$$\mathfrak{h}_{22,l,k} = \sum_{\xi} \lambda_{\xi} \left[\left(\sum_q \mathfrak{s}_s^{\dagger} \mathfrak{s}_q \right)_l \otimes \left(\sum_p \mathfrak{s}_p^{\dagger} \mathfrak{s}_r \right)_k + \left(\sum_p \mathfrak{s}_p^{\dagger} \mathfrak{s}_r \right)_l \otimes \left(\sum_q \mathfrak{s}_s^{\dagger} \mathfrak{s}_q \right)_k \right], \quad (88)$$

which allowed us to separate the control operators in Eq. (87). Returning to the exponential of Eq. (81)

$$\mathcal{U}_{22}^f(\Delta t) = \exp \left[-i\Delta t \sum_{j=1}^{n-1} \sum_{k=1}^j \sum_{l=1+k}^{j+1} \mathbb{P}_{j+1-l}^{(n-i)} \otimes \{(\mathfrak{C}_{11} \otimes)^{\otimes l-k+1}\} \{ \mathfrak{h}_{22,l,k} \} \otimes \mathbb{P}_{k-1}^{(k-1)} \cdot \frac{\mathcal{A}_{j+1 \leftarrow j-1}}{\sqrt{j(j+1)}} \right]. \quad (89)$$

Here comes a delicate step: Exponents with different j commute because of projectors, but terms with different l and k do not, in general, although it can be shown they do commute provided that, for each term with ξ , there exists a term with $-\xi$ and that $\lambda_{\xi} = \lambda_{-\xi}$. For a central (or otherwise parity-even) potential, it is natural to take $\lambda_{\xi} = \lambda_{-\xi}$; and considering this to be the case, the following decomposition is exact:

$$\mathcal{U}_{22}^f(\Delta t) = \prod_{j=1}^{n-1} \prod_{k=1}^j \prod_{l=1+k}^{j+1} \exp \left[-i\Delta t \mathbb{P}_{j+1-l}^{(n-l)} \otimes \{(\mathfrak{C}_{11} \otimes)^{\otimes l-k+1}\} \{ \mathfrak{h}_{22,l,k} \} \otimes \mathbb{P}_{k-1}^{(k-1)} \cdot \frac{\mathcal{A}_{j+1 \leftarrow j-1}}{\sqrt{j(j+1)}} \right]. \quad (90)$$

From here on, the calculation's steps are similar to the ones used in Sec. VA. First, \mathcal{A} and the operator before the \cdot commute because of Eq. (72), so the step antisymmetrizer can be factorized from the exponential:

$$\mathcal{U}_{22}^f(\Delta t) = \prod_{j=1}^{n-1} \prod_{k=1}^j \prod_{l=1+k}^{j+1} \left\{ \mathbb{I}^{(n)} + \sum_{m=1}^{\infty} \left[-i\Delta t \mathbb{P}_{j+1-l}^{(n-l)} \otimes \{(\mathfrak{C}_{11} \otimes)^{\otimes l-k+1}\} \{ \mathfrak{h}_{22,l,k} \} \otimes \mathbb{P}_{k-1}^{(k-1)} \right]^m \cdot \frac{\mathcal{A}_{j+1 \leftarrow j-1}}{\sqrt{j(j+1)}} \right\}; \quad (91)$$

note also that the step antisymmetrizer does nothing on fermion memories and that projectors and control operators are idempotent to arrive to the following:

$$\mathcal{U}_{22}^f(\Delta t) = \prod_{j=1}^{n-1} \prod_{k=1}^j \prod_{l=1+k}^{j+1} \left\{ \mathbb{I}^{(n)} + \mathbb{P}_{j+1-l}^{(n-l)} \otimes \{(\mathfrak{C}_{11} \otimes)^{\otimes l-k+1}\} \left\{ \sum_{m=1}^{\infty} (-i\Delta t \mathfrak{h}_{22,l,k})^m \right\} \otimes \mathbb{P}_{k-1}^{(k-1)} \right\} \quad (92)$$

$$= \prod_{j=1}^{n-1} \prod_{k=1}^j \prod_{l=1+k}^{j+1} \left\{ \sum_{\substack{l=0 \\ l \neq j+1}}^n \mathbb{P}_l^{(n)} + \mathbb{P}_{j+1-l}^{(n-l)} \otimes \{(\mathfrak{C}_{11} \otimes)^{\otimes l-k+1}\} \{ \mathfrak{u}_{22,l,k}(\Delta t) \} \otimes \mathbb{P}_{k-1}^{(k-1)} \right\}, \quad (93)$$

where in the last step we subtracted from the identity the term corresponding to $m = 0$ and defined

$$\mathfrak{u}_{22,l,k}(\Delta t) = \exp(-i\Delta t \mathfrak{h}_{22,l,k}). \quad (94)$$

Finally, products are decomposed as in the free-evolution case to arrive at

$$\mathcal{U}_{22}^f(\Delta t) = \mathbb{P}_0^{(n)} + \mathbb{P}_1^{(n)} + \sum_{j=1}^{n-1} \mathbb{P}_0^{(n-1-j)} \otimes \{(\mathfrak{C}_{11})^{\otimes j+1}\} \left\{ \prod_{k=1}^j \prod_{l=1+k}^{j+1} \mathfrak{u}_{22,(l,k)}(\Delta t) \right\}, \quad (95)$$

with

$$\mathfrak{u}_{22,l,k}(\Delta t) = \exp \left(-i\Delta t \sum_{\xi} \lambda_{\xi} \left\{ \left(\sum_p \mathfrak{s}_{p+\xi\Delta}^{\dagger} \mathfrak{s}_p \right)_l \otimes \left(\sum_q \mathfrak{s}_q^{\dagger} \mathfrak{s}_{q+\xi\Delta} \right)_k + \left(\sum_q \mathfrak{s}_q^{\dagger} \mathfrak{s}_{q+\xi\Delta} \right)_l \otimes \left(\sum_p \mathfrak{s}_{p+\xi\Delta}^{\dagger} \mathfrak{s}_p \right)_k \right\} \right), \quad (96)$$

which is an auxiliary ‘‘momentum exchanger’’ between register in which all identities have been omitted to alleviate notation.

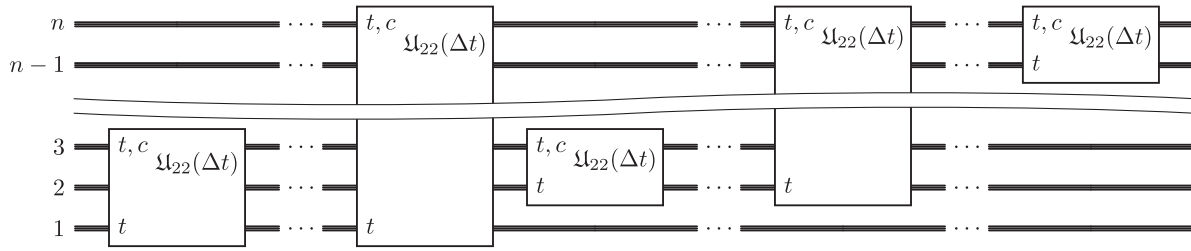


FIG. 2. Schematic circuit implementing the four-body term Eq. (95). See Fig. 1 for an explanation of the lettering.

The corresponding boson expression can be derived changing step antisymmetrizers for step symmetrizers, and the result is equivalent.

1. Gate-level implementation and costs

The circuit implementation of Eq. (95) in terms of the subcircuits corresponding to Eq. (96) is represented in Fig. 2. The total number of subcircuits is now the square of

the number of registers: For each value of j , there are $j - 1$ of these subcircuits ($\mathbf{U}_{22,j,1}$ to $\mathbf{U}_{22,j,(j-1)}$); in total, this gives $\sum_{j=2}^n (j - 1) = \frac{n(n-1)}{2} = \mathcal{O}(n^2)$ (the usual number of exchange terms for pairwise particle interactions).

To bound the costs of each of these subcircuits, we consider a direct but very inefficient procedure. First, the Trotter formula is used to decompose the exponential Eq. (96) in terms of fixed momenta:

$$\begin{aligned} \mathbf{U}_{22,l,k}(\Delta t) &= \prod_{\xi} \prod_{\substack{\min(\Lambda, \Lambda - (+)\xi) \\ \xi \frac{p(q)}{\Delta} = \max(-\Lambda, -\Lambda - (+)\xi)}}^{\min(\Lambda, \Lambda - (+)\xi)} \exp(-i\Delta t \lambda_{\xi} \{ (\mathfrak{s}_{p+p_{\xi}}^{\dagger} \mathfrak{s}_p)_l \otimes (\mathfrak{s}_{q-p_{\xi}}^{\dagger} \mathfrak{s}_q)_k + \text{H.c.} \}) + \mathcal{O}(\Delta t^2) \\ &\equiv \prod_{\xi p q} \mathbf{U}_{22,l,k}(p, q, \xi; \Delta t). \end{aligned} \quad (97)$$

Each $\mathbf{U}_{22,l,k}(p, q, p_{\xi}; \Delta t)$ is a rotation between states $|p\rangle_l |q\rangle_k$ and $|p + p_{\xi}\rangle_l |q - p_{\xi}\rangle_k$, so it can be easily diagonalized by a change of basis (generalization of the Hadamard gate) with matrix $T_{(p,q,\xi)}$ given by

$$\begin{aligned} |+\rangle_{\xi} &= T_{(p,q,p_{\xi})}^{-1} |p\rangle_l |q\rangle_k \\ &= \frac{1}{\sqrt{2}} (|p\rangle_l |q\rangle_k + |p + p_{\xi}\rangle_l |q - p_{\xi}\rangle_k), \end{aligned} \quad (98)$$

$$\begin{aligned} |-\rangle_{\xi} &= T_{(p,q,p_{\xi})}^{-1} |p + p_{\xi}\rangle_l |q - p_{\xi}\rangle_k \\ &= \frac{1}{\sqrt{2}} (|p\rangle_l |q\rangle_k - |p + p_{\xi}\rangle_l |q - p_{\xi}\rangle_k), \end{aligned} \quad (99)$$

which can be implemented by a Gray code and a controlled Hadamard, resulting in an asymptotic cost of $\mathcal{O}(\log_2^2 N_p)$ CNOTs and single-qubit gates. The transformed rotation is diagonal, e.g.,

$$\mathbf{U}_{22(l,k)}(p, q, p_{\xi}; \Delta t) |+\rangle_{\xi} = \exp(-i2\Delta t \lambda_{\xi}) |+\rangle_{\xi}, \quad (100)$$

and similarly for $|-\rangle_{\xi}$ changing the phase sign. Each of these phase transformations controlled with $\mathcal{O}(2 \log_2 N_p)$ qubits is implementable with $\mathcal{O}(\log_2 N_p)$ CNOTs and single-qubit gates. The costs are, thus, dominated by the change of basis, so the full implementation of the four-body term is

upper bounded by $\mathcal{O}(n^2 N_p^3 \log_2^2 N_p)$ CNOTs and single-qubit gates. See Chap. 4 in Ref. [36] for a discussion about Gray codes and the implementation of general unitary transformations.

VI. EXPONENTIATION OF PARTICLE-NUMBER-CHANGING OPERATORS

The formalism that we have here developed would not be particularly advantageous with evolution operators such as those of Secs. VA and VB, that do not really require a dynamical memory but can instead be statically implemented in other ways at fixed particle number; but it comes into its own when confronted with particle-number-changing operators. The difficulty in encoding these comes from the lack of commutativity in the exponentials and the implementation of symmetrizers and antisymmetrizers. We here propose a way of encoding them which is consistent with all the earlier discussion and definitions, without prejudice that more efficient organizations can be thought of in future work. We have chosen to exemplify with one-field operators (“tadpoles”) and three-field operators (“splitting”) that occupy the rest of this section.

A. One-field operators (tadpoles)

One-field operators do not typically appear in field-theoretical Hamiltonian densities formulated on a stable

vacuum, as they are linear in the fields (although anomalous terms do appear in QCD formulated in the time-axial gauge), but they can be useful in dynamical situations such as a system rolling down a classical potential. In any case, our purpose here is to use the simplest possible operator which can expose the changing number of particles, and these are obviously one-field terms. We first address fermion operators which, though not appearing in physical processes [the superselection rule associated with (2π) rotations forbids the creation of an odd number of fermions] provide a useful stepping stone for other algorithms. Then we will address the boson tadpole in Sec. VI A 3.

The resulting unitary operator is then

$$\begin{aligned} \mathcal{U}_{10}^f(\Delta t) &= \exp \left[-i\Delta t \sum_p \lambda_p (b_p^{(n)\dagger} + b_p^{(n)}) \right] \\ &= \exp \left[-i\Delta t \sum_p \lambda_p \sum_j (b_{p,j}^{(n)\dagger} + b_{p,j}^{(n)}) \right]. \end{aligned} \quad (101)$$

The first step is to decompose the operator in products of terms separately acting over each register, which amounts to transforming the exponential of the sum over j into a product of exponentials. We iterate to factor each of the j -indexed pieces in turn. Because of the noncommutativity and the Baker-Campbell-Hausdorff rule for noncommuting exponents, $e^A e^B = e^{A+B+\frac{1}{2}[A,B]+\dots}$, we expect errors of the order of $(\Delta t)^2$. To factor out the first $\sum_p \lambda_p (b_{p,1}^{(n)\dagger} + b_{p,1}^{(n)})$ we check its commutation with the rest:

$$\left[(b_{p,1}^{(n)\dagger} + b_{p,1}^{(n)}), \sum_{j=2}^n (b_{q,j}^{(n)\dagger} + b_{q,j}^{(n)}) \right] = b_{q,2}^{(n)\dagger} b_{p,1}^{(n)\dagger} - b_{q,2}^{(n)} b_{p,1}^{(n)}, \quad (102)$$

as expected (for any operator changing the number of particles), the factorization is not exact, and some error depending on λ_l and Δt is introduced if particles are created or annihilated one at a time. Such uncertainty is controlled

by the Trotter step with small parameter Δt , whose asymptotically small values would allow the use of the (approximate at finite Δt) product formula

$$\begin{aligned} \mathcal{U}_{10}^f &= \exp \left[-i\Delta t \sum_j \sum_p \lambda_p (b_{p,j}^{(n)\dagger} + b_{p,j}^{(n)}) \right] \\ &= \prod_j \exp \left[-i\Delta t \sum_p \lambda_p (b_{p,j}^{(n)\dagger} + b_{p,j}^{(n)}) \right] \\ &\quad + O((\Delta t)^2) \end{aligned} \quad (103)$$

up to $O((\Delta t)^2)$ in the exponents (thus, the calculation is systematically improvable for $\Delta t \rightarrow 0$ independently of λ , even for strong coupling). A more rigorous error bound should also consider the norm of the first nonvanishing commutators; here, there are n of those, each with a norm $O(N_p \bar{\lambda}^2 = \sqrt{\sum_p \sum_q \lambda_p^2 \lambda_q^2})$, so controlling the error suggests operating with $\Delta t \ll \hbar / \sqrt{n N_p \bar{\lambda}^2}$. In the rest of the manuscript, the errors are not written explicitly and, in particular, its dependence on first commutators is not taken into account.

To continue, creation and annihilation operators are written in terms of their set and scrap counterparts:

$$\begin{aligned} b_{p,j}^{(n)\dagger} + b_{p,j}^{(n)} &= \mathcal{A}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \\ &\quad + \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \mathcal{A}_{j \leftarrow j-1}, \end{aligned} \quad (104)$$

but it is not clear how such a term can be exponentiated, as Eq. (72) is not applicable because of the change in particle number. The issue can be circumvented if we manage to rewrite that expression in terms of some new unitary operators (note the circumflex hat), the ordered step antisymmetrizer and its conjugate, $\hat{\mathcal{A}}_{j \leftarrow j-1}^\dagger$ and $\hat{\mathcal{A}}_{j \leftarrow j-1}$, satisfying

$$\begin{aligned} b_{p,j}^{(n)\dagger} &= \mathcal{A}_{j \leftarrow j-1} \cdot \mathbb{P}_0^{(n-j)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger)_j \otimes \mathbb{P}_{j-1}^{(j-1)} \\ &\equiv \hat{\mathcal{A}}_{j \leftarrow j-1}^\dagger \cdot \mathbb{P}_0^{(n-j)} \otimes ((\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger)_j \otimes \mathbb{P}_{j-1}^{(j-1)})_{f,ord} \cdot \hat{\mathcal{A}}_{j \leftarrow j-1}, \end{aligned} \quad (105)$$

where the subscripts f, ord are explained in detail in the next paragraphs. It is obvious that the naïve antisymmetrizers \mathcal{A} are not invertible (and, thus, not possibly unitary), because their application projects out part of the Hilbert space. To construct a unitary operator which is implementable on a quantum computer, we need an invertible rule. This is possible if we force an initial ordering in the registers, so that the antisymmetrizer takes the ordered set producing

an antisymmetrized state, whereas its inverse takes the antisymmetrized state and returns it to the unique, properly ordered state in a bijective manner. Such a procedure was already outlined by Abrams and Lloyd 20 years ago [5]; see [17] for a more up to date discussion. Thus, we need to distinguish between the \mathcal{A} antisymmetrizer, closer to that of field theory, and the $\hat{\mathcal{A}}$ antisymmetrizer, specific for computer registers which have a well-defined order.

To achieve unitarity, the new operator $\hat{A}_{j \leftarrow j-1}^\dagger$ is defined to act as an antisymmetrizer only over the ordered subset of the j -particle states and otherwise acts as the identity over less occupied states:

$$\hat{A}_{j \leftarrow j-1}^\dagger \begin{cases} [|1p\rangle_j(|1p_{j-1}\rangle_{j-1}\dots|1p_1\rangle_1)_{\mathcal{A}}]_{\text{ord}} = (|1p\rangle_j|1p_{j-1}\rangle_{j-1}\dots|1p_1\rangle_1)_{\mathcal{A}} \\ |\Omega\rangle_j(|1p_{j-1}\rangle_{j-1}\dots|1p_1\rangle_1)_{\mathcal{A}} = |\Omega\rangle_j(|1p_{j-1}\rangle_{j-1}\dots|1p_1\rangle_1)_{\mathcal{A}} \end{cases} \quad (106)$$

Here, a state $|1p\rangle_j(|1p_{j-1}\rangle_{j-1}\dots|1p_1\rangle_1)_{\mathcal{A}}$ is said to be ordered and subscripted so, if the momentum mode on the j th register is the largest, in the sense defined by some appropriate ordering criteria. (Note that, as the antisymmetrization is performed one register at a time, the procedure is different to the one outlined in [5].) We here make the following obvious and natural choices, first on the discretized momenta, subsequently on the ordering of states ensured *ad hoc* by the corresponding creation criterion.

Order criterion on momenta.—A mode $|p_i\rangle$ is larger or equal than $|p_j\rangle$ if and only if $i \geq j$. It can be checked on runtime by an order oracle

$$O(|x\rangle|p_i\rangle|p_j\rangle) = \begin{cases} |x \oplus 1\rangle|p_i\rangle|p_j\rangle & \text{if } i \geq j \\ |x\rangle|p_i\rangle|p_j\rangle & \text{otherwise.} \end{cases} \quad (107)$$

Order criterion on creation operators.—If a new particle is to be created, the final state should have its largest mode, in the sense defined by the order criterion, on the recently occupied register. (This is to facilitate the invertible antisymmetrization.)

From here on, these criteria are enforced and remarked by the subscript *ord*, as in Eq. (105). The other subscript there, *f*, is a reminder that no mode is to be created if it is already on memory and that $(-)$ signs should occur each time two-particle registers are permuted. We now discuss the detailed implementation of each piece, starting from the unitary operator \hat{A} .

1. Implementation of unitary step antisymmetrizers and fermion-SWAP gates

As for the definition in Eq. (106), the ordered step operator $\hat{A}_{j \leftarrow j-1}^\dagger$ antisymmetrizes states provided the j th register is occupied by the most sizable momentum and acts as the identity if the j th register is empty. Both situations can be distinguished by using controls over the presence qubit of register j .

Implementation of \hat{A}^\dagger .—The passage from (to) an ordered state to (from) an antisymmetrized one is realized invoking a subroutine named “locate the largest” (LL) described in Appendix A 1 that returns, in an auxiliary register, a true qubit pointing to the largest mode. To illustrate the antisymmetrization step, consider the probe state $|\phi_0\rangle = |1p_j\rangle_j(|1p_{j-1}\rangle_{j-1}\dots|1p_1\rangle_1)_{\mathcal{A}}$ with the largest mode on the last occupied slot $|p_j\rangle > |p_k\rangle$, $k \in [1, j-1]$. The algorithm proceeds as follows (we show the particle content of the register at each step).

- (1) Add an auxiliary register B with j qubits with the last of them in state $|1\rangle$, so pointing to $|1p_j\rangle$:

$$|\phi_1\rangle = |1, 0, \dots, 0\rangle_B |1p_j\rangle_j(|1p_{j-1}\rangle_{j-1}\dots|1p_1\rangle_1)_{\mathcal{A}} \quad (108)$$

- (2) Create an equally weighted superposition of states in the auxiliary register with the qubit in state 1 in each of the positions in turn:

$$|\phi_2\rangle = \frac{1}{\sqrt{j}} (|1, 0, \dots, 0\rangle_B + |0, 1, \dots, 0\rangle_B + \dots + |0, \dots, 1\rangle_B) |1p_j\rangle_j(|1p_{j-1}\rangle_{j-1}\dots|1p_1\rangle_1)_{\mathcal{A}} \quad (109)$$

This costs $\mathcal{O}(\log_2 j)$ single-qubit gates.

- (3) Exchange the first momentum register with whichever is pointed at by the single 1 in the auxiliary B and then change the sign to account for fermion antisymmetry. Note that an antisymmetric state of $j-1$ registers can be written as

$$|p_j\rangle_j(|p_{j-1}\rangle_{j-1}\dots|p_l\rangle_l\dots|p_1\rangle_1)_{\mathcal{A}} = \sum_{k=1}^{j-1} (-1)^{|l-k|} |p_j\rangle_j(|p_{j-1}\rangle_{j-1}\dots|\overline{p_k}\rangle_l\dots|p_{k+1}\rangle_k|p_{k-1}\rangle_{k-1}\dots|p_1\rangle_1)_{\mathcal{A}}, \quad (110)$$

where momentum in register l is fixed to p_k . The SWAP thus produces an interchange between momenta p_k and p_j :

$$|\phi_3\rangle = \frac{1}{\sqrt{j}} \left\{ |1, 0, \dots, 0\rangle_B |1p_j\rangle_j(|1p_{j-1}\rangle_{j-1}\dots|1p_1\rangle_1)_{\mathcal{A}} - \sum_{l=1}^{j-1} |0, \dots, \frac{1}{l}, \dots, 0\rangle_B \sum_{k=1}^{j-1} (-1)^{|l-k|} |p_k\rangle_j(|p_{j-1}\rangle_{j-1}\dots|\overline{p_j}\rangle_l\dots|p_{k+1}\rangle_k|p_{k-1}\rangle_{k-1}\dots|p_1\rangle_1)_{\mathcal{A}} \right\} \quad (111)$$

This costs $\mathcal{O}(j)$ SWAPs and phase gates, each controlled by an auxiliary qubit and the presence qubit of register j . For a total of $\mathcal{O}(N_p)$ momenta codified in $\mathcal{O}(\log_2 N_p)$ qubits, this can be decomposed in $\mathcal{O}(j \log_2 N_p)$ CNOTs and $\mathcal{O}(j \log_2 N_p)$ single-qubit gates.

- (4) We now want to disentangle register B . For this, we will take (in an orderly manner) all its qubits to $|0\rangle$ and factor it off. By hypothesis, the one qubit in the state $|1\rangle$ was marking the position of the largest mode, and the LL algorithm from Appendix A 1 can be inverted to return it to zero:

$$|\phi_4\rangle = \frac{1}{\sqrt{j}} |0, \dots, 0\rangle_B \left\{ |1p_j\rangle_j (|1p_{j-1}\rangle_{j-1} \dots |1p_1\rangle_1)_A - \sum_{l=1}^{j-1} \sum_{k=1}^{j-1} (-1)^{|l-k|} |p_k\rangle_j (|p_{j-1}\rangle_{j-1} \dots |p_j\rangle_l \dots |p_{k+1}\rangle_k |p_{k-1}\rangle_{k-1} \dots |p_1\rangle_1)_A \right\}; \quad (112)$$

returning the register with momentum p_j to its initial position (summing over l), we have

$$|\phi_4\rangle = |0, \dots, 0\rangle_B (|p_j\rangle_j |p_{j-1}\rangle_{j-1} \dots |p_1\rangle_1)_A, \quad (113)$$

Costs: The costs of the LL algorithm over j registers (j particles in memory) involves $\mathcal{O}(j^2)$ CNOTs, single-qubit gates, and order-oracle calls.

Thus, the total costs are $\mathcal{O}(j^2 + j \log_2 N_p)$ CNOTs and single-qubit gates, $\mathcal{O}(j^2)$ order-oracle calls, and $\mathcal{O}(\log_2 j)$ rotations.

We now face the implementation of the order criterion on creation operators. The “fermion” and “order” subscripts impose two conditions over the creation and annihilation term of Eq. (105): (a) Mode duplication is forbidden, and (b) upon completion, the largest mode must sit on the last occupied register. To fulfill these requirements, we proceed one momentum at a time (which implies the use of the Trotter expansion). We will also need a piece of code which allows the agile exchange of entire particle registers, to which we now devote our attention.

Fermion-SWAP gate, $fS_j(p, q)$, and fermion-exchange gate, $fX_j(p)$.—We need to be able to exchange fermions for two reasons. One is the ability to order the largest mode at the last register so that the ordered step antisymmetrizer can work. And importantly, to satisfy Pauli exclusion, avoid creating a mode for an existing momentum p : When it is to be created, we attempt to bring it to the front to nullify the action of the creation operator in case it is already present in the memory (obviously, by p we mean here the complete fermion orbital including color and other degrees of freedom, but not the presence qubit).

At a lower level of programming, we need to introduce a fermion-SWAP gate depending on two modes $fS_j(p, k)$. This function will be overloaded: Its implementation is different depending on whether the first input is a general mode p , the vacuum mode 0 or the largest mode L . We then

write either $fS_j(L, p)$ or $fS_j(0, p)$ to distinguish these two cases.

Once both implemented, we can define a fermion-exchange operator for momentum p as

$$fX_j(p) = (\mathfrak{C}_{00})_j \otimes fS_j(0, p) + (\mathfrak{C}_{11})_j \otimes fS_j(L, p). \quad (114)$$

The gate is implemented by the following sequence of operations (in which only one empty register is assumed):

- (1) By means of an auxiliary register B with j qubits, “mark” the particle-registers with modes k and p for $fS_j(p, k)$; 0 and p for $fS_j(p, 0)$; and finally the largest mode and p for $fS_j(p, L)$. This is achieved by controlled-X gates or the LL algorithm correspondingly. The costs of this procedure amount to $\mathcal{O}(\log_2 N_p)$ Toffoli gates for an X gate controlled on $\mathcal{O}(\log_2 N_p)$ qubits; this can be decomposed in $\mathcal{O}(\log_2 N_p)$ CNOTs and $\mathcal{O}(\log_2 N_p)$ single-qubit gates. The LL algorithm adds $\mathcal{O}(j^2)$ CNOTs, single-qubit gates, and oracle calls.
- (2) Permute the marked register with the j th, using the auxiliary qubits in state $|1\rangle$. Account a $(-)$ sign for fermion antisymmetry. The associated costs are now as follows: one-qubit control of SWAPs adds only a constant overhead in CNOTs and single-qubit gates, so this requires $\mathcal{O}(j \log_2 N_p)$ CNOTs and $\mathcal{O}(j \log_2 N_p)$ single-qubit gates. The $(-)$ signs can be added by applying $\mathcal{O}(j)$ phase gates controlled on the auxiliary qubit pointing to the interchanged register.
- (3) Apply again the first step, so the auxiliary register returns to its initial state. The cost here is the same as in step 1.

The total costs amounts to $\mathcal{O}(j \log_2 N_p)$ CNOT and single-qubit gates for $fS_j(p, k)$ and $fS_j(p, 0)$, $\mathcal{O}(j^2)$ order-oracle calls and $\mathcal{O}(j^2 + j \log_2 N_p)$ CNOT and single-qubit gates for $fS_j(p, L)$.

2. Implementation of Eq. (105), the qubit-level exponentiation

Introduction of these gates in Eq. (105) gives

$$\begin{aligned}
b_{p,j}^{(n)\dagger} + b_{p,j}^{(n)} &= \hat{\mathcal{A}}_{j\leftarrow j-1}^\dagger \cdot \mathbb{P}_0^{(n-j)} \otimes ((\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger)_j \otimes \mathbb{P}_{j-1}^{(j-1)} + \text{H.c.})_{f,ord} \cdot \hat{\mathcal{A}}_{j\leftarrow j-1} \\
&= \hat{\mathcal{A}}_{j\leftarrow j-1}^\dagger \cdot [(\mathfrak{C}_{00})_j \otimes \text{fS}_j(p, 0) + (\mathfrak{C}_{11})_j \otimes \text{fS}_j(p, L)] \cdot \mathbb{P}_0^{(n-j)} \otimes ((\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger)_j \otimes \mathbb{P}_{j-1}^{(j-1)} + \text{H.c.}) \\
&\quad \cdot [(\mathfrak{C}_{00})_j \otimes \text{fS}_j(0, p) + (\mathfrak{C}_{11})_j \otimes \text{fS}_j(L, p)] \cdot \hat{\mathcal{A}}_{j\leftarrow j-1} \\
&= \hat{\mathcal{A}}_{j\leftarrow j-1}^\dagger \cdot \text{fX}_j^\dagger(p) \cdot \mathbb{P}_0^{(n-j)} \otimes ((\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger)_j \otimes \mathbb{P}_{j-1}^{(j-1)} + \text{H.c.}) \cdot \text{fX}_j(p) \cdot \hat{\mathcal{A}}_{j\leftarrow j-1}, \tag{115}
\end{aligned}$$

where $\text{fS}_j^\dagger(p, k) = \text{fS}_j(k, p)$. To arrive to the second equality, note that both the order and no mode-duplication requirements are satisfied: $\hat{\mathcal{A}}_{j\leftarrow j-1}$ leaves the largest mode on register j when it is occupied, so controlled on the presence of j being $|1\rangle$, applying the $\text{fS}_j(L, p)$ gate leaves momentum p on register j , and it can be annihilated. On the other hand, if register j is empty, $\hat{\mathcal{A}}_{j\leftarrow j-1}$ does nothing and $\text{fS}_j(0, p)$ tries to SWAP the p and vacuum modes; if this is achieved, the j register ends up in the state $|0p\rangle$, which

does not change upon action of $(\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger)_j$ and avoids mode duplication; if otherwise there was no mode p on memory, $\text{fS}_j(0, p)$ does nothing and the creation procedure carries on. Subsequent application of $\text{fX}_j^\dagger(p)$ reorders the memory if register j ends up being occupied or returns $|0p\rangle$ to $|\Omega\rangle$ if it is empty.

With these operators, we may proceed to exponentiation. Since antisymmetrizers do not depend on the created or annihilated momentum and $\hat{\mathcal{A}}_{j\leftarrow j-1}^\dagger \hat{\mathcal{A}}_{j\leftarrow j-1} = \mathbb{I}^{(j)}$,

$$\begin{aligned}
&\exp\left(-i\Delta t \sum_p \lambda_p (b_{p,j}^{(n)\dagger} + b_{p,j}^{(n)})\right) \\
&= \hat{\mathcal{A}}_{j\leftarrow j-1}^\dagger \cdot \left[\exp\left(-i\Delta t \sum_p \lambda_p \text{fX}_j^\dagger(p) \cdot \mathbb{P}_0^{(n-j)} \otimes ((\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger)_j + \text{H.c.}) \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \text{fX}_j(p)\right) \right] \cdot \hat{\mathcal{A}}_{j\leftarrow j-1}, \tag{116}
\end{aligned}$$

where operators $\mathbb{I}^{\otimes n-j}$ multiplying antisymmetrizers have been omitted. To continue with exponentiation, we use the Trotter formula for the sum over p and use $\text{fX}_j^\dagger(p)\text{fX}_j(p) = \mathbb{I}^{(j)}$:

$$\begin{aligned}
\mathcal{U}_{10}^f &= \prod_j \exp\left[-i\Delta t \sum_p \lambda_p (b_{p,j}^{(n)\dagger} + b_{p,j}^{(n)})\right] \\
&= \prod_j \left\{ \hat{\mathcal{A}}_{j\leftarrow j-1}^\dagger \cdot \prod_p [\text{fX}_j^\dagger(p) \cdot \exp(-i\Delta t \lambda_p \mathbb{P}_0^{(n-j)} \otimes ((\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger)_j + \text{H.c.}) \otimes \mathbb{P}_{j-1}^{(j-1)}) \cdot \text{fX}_j(p)] \cdot \hat{\mathcal{A}}_{j\leftarrow j-1} \right\} \\
&\quad + \mathcal{O}(\Delta t^2). \tag{117}
\end{aligned}$$

Finally, using the idempotency of operators \mathfrak{C}_{11} , \mathfrak{C}_{00} , and \mathfrak{i} and similar transformations to those of the preceding sections, we can write down the exponential as

$$\mathcal{U}_{10}^f = \prod_j \left(\sum_{\substack{l=0 \\ l \neq j-1}} \mathbb{P}_l^{(n)} + \hat{\mathcal{A}}_{j\leftarrow j-1}^\dagger \cdot \prod_p [\text{fX}_j^\dagger(p) \cdot \mathbb{P}_0^{(n-j)} \otimes \mathfrak{U}_{10,j}(\Delta t, \lambda_p) \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \text{fX}_j(p)] \cdot \hat{\mathcal{A}}_{j\leftarrow j-1} \right) + \mathcal{O}(\Delta t^2) \tag{118}$$

Note that using the LL algorithm allows writing the $\hat{\mathcal{A}}$ and $\hat{\mathcal{A}}^\dagger$ operators outside the products over modes in Eq. (118), but it is also possible to implement “mode-dependent antisymmetrizers” $\hat{\mathcal{A}}(p)$ and $\hat{\mathcal{A}}^\dagger(p)$ that are generally cheaper, although they should be repeated for each mode.

$$\mathfrak{U}_{10,j}(\Delta t, \lambda_p) = \exp(-i\Delta t \lambda_p (\mathfrak{C}_{10} \otimes \mathfrak{s}_p^\dagger + \mathfrak{C}_{01} \otimes \mathfrak{s}_p)_j). \tag{119}$$

Gate-level implementation and costs.—Each term in the j product on Eq. (118) can be represented as depicted in Fig. 3 (the controlled gates therein use presence qubits).

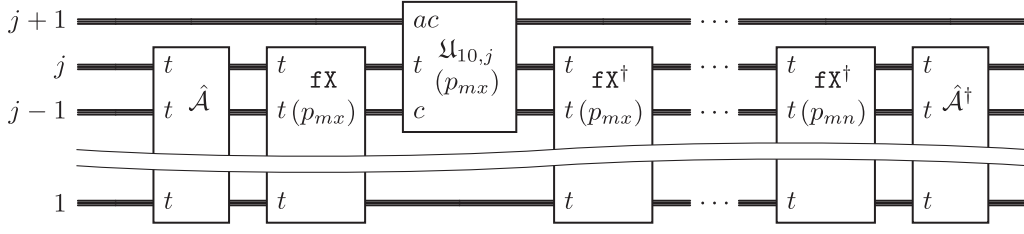


FIG. 3. Circuit implementation of Eq. (118), each thick line representing a particle register. Lettering as in Fig. 1 and the glossary. With time advancing from left to right (the equation backward, from right to left), we start with the antisymmetric state which is unpacked by \hat{A} to one with the largest p in the last occupied register. Then, if a fermion in the memory has p_i , it is brought to the front by a fermion exchange, to be able to block the creation of a second state with equal quantum numbers which would violate Pauli's exclusion. Should that not be the case, $\mathfrak{U}_{10,j}$ introduces the new fermion as indicated by the Hamiltonian. The fermion exchange is then inverted to bring the largest p to the front, these operations are then repeated for each mode, and the memory is antisymmetrized at the end. This row of operators is repeated as needed to complete the wanted number of Trotter steps.

The operators inside the momentum product in Eq. (118) are repeated $\mathcal{O}(N_p)$ times, so we have (for the costs of each operation, we refer to the preceding paragraphs or to Table II)

- (i) $\mathcal{O}(N_p)fX_j(p)$, with a total cost of $\mathcal{O}(jN_p \log_2 N_p)$ CNOTs and single-qubit gates and $\mathcal{O}(j^2)$ oracle calls (these include SWAPs with largest momenta).
- (ii) $\mathcal{O}(1)\hat{A}_{j \leftarrow j-1}^\dagger$, with a total cost of $\mathcal{O}(j^2 + j \log_2 N_p)$ CNOTs, single-qubit gates, and $\mathcal{O}(j^2)$ order-oracle calls.
- (iii) $\mathcal{O}(N_p)\mathfrak{U}_{10,j}(\Delta t, \lambda_p)$, each of these implemented as \mathfrak{U}_{22} back on Sec. VB, with a Gray code and a controlled single-qubit rotation that results in a total cost of $\mathcal{O}(N_p \log_2^2 N_p)$ CNOTs and single-qubit gates. Addition from $j=1$ to $j=n$ gives a total cost of $\mathcal{O}(n^2 N_p \log_2 N_p + N_p \log_2^2 N_p)$ CNOTs and single-qubit gates + $\mathcal{O}(n^3 N_p)$ order-oracle calls.

3. Boson tadpole

Many aspects of implementing boson operators can be read off from the fermion ones by changing the antisymmetrizing operators for symmetrizing ones, which directly leads to operators \hat{S} and \hat{S}^\dagger which symmetrize “semi-ordered” states, as the largest mode can now be repeated (no Pauli exclusion applies). Thus, the preceding algorithms have to be slightly extended. As usual, details are considered in Appendix A 2. Here, we give examples that illustrate the main steps for the case of three registers.

Implementation of \hat{S}^\dagger .—As an example of the implementation of the symmetrizer, consider the symmetrization of the semiordered state $|\phi_1\rangle = |P\rangle_3(|P\rangle_2|q\rangle_1)_S$ with $|P\rangle > |q\rangle$ and two equal momenta. We may proceed as follows.

- (1) Add two auxiliary registers of $n=3$ qubits, C and B :

$$|\phi_1\rangle = |100\rangle_C |000\rangle_B \frac{1}{\sqrt{2}} (|PPq\rangle + |PqP\rangle). \quad (120)$$

- (2) Apply the LL algorithm to B (now there can be more than one qubit marking the largest P):

$$|\phi_2\rangle = |100\rangle_C \frac{1}{\sqrt{2}} (|110\rangle_B |PPq\rangle + |101\rangle_B |PqP\rangle). \quad (121)$$

- (3) Symmetrize register C :

$$|\phi_3\rangle = \frac{1}{\sqrt{3}} (|100\rangle_C + |010\rangle_C + |001\rangle_C) \times \frac{1}{\sqrt{2}} (|110\rangle_B |PPq\rangle + |101\rangle_B |PqP\rangle). \quad (122)$$

- (4) Using register C , apply Eq. (38) with $j=3$ on both the B and the momentum registers; i.e., permute the third of the mode registers and the third of B 's qubits with that mode register and that qubit of B 's which are pointed at by register C :

$$|\phi_4\rangle = \frac{1}{\sqrt{3}} \left[\frac{1}{\sqrt{2}} (|100\rangle_C + |010\rangle_C) |110\rangle_B |PPq\rangle + \frac{1}{\sqrt{2}} (|100\rangle_C + |001\rangle_C) |101\rangle_B |PqP\rangle + \frac{1}{\sqrt{2}} (|010\rangle_C + |001\rangle_C) |011\rangle_B |qPP\rangle \right]. \quad (123)$$

- (5) Register B can now be used to return C to its initial state: For each state of B , we have equal superpositions of states of C with the qubit in state 1 in the positions pointed by B :

- (a) If $|B\rangle = |110\rangle$, interchange $\frac{1}{\sqrt{2}} (|100\rangle_C + |010\rangle_C) \leftrightarrow |000\rangle_C$.
- (b) If $|B\rangle = |101\rangle$, interchange $\frac{1}{\sqrt{2}} (|100\rangle_C + |001\rangle_C) \leftrightarrow |000\rangle_C$.
- (c) If $|B\rangle = |011\rangle$, interchange $\frac{1}{\sqrt{2}} (|010\rangle_C + |001\rangle_C) \leftrightarrow |000\rangle_C$.
- (d) If $|B\rangle = |111\rangle$, interchange $\frac{1}{\sqrt{3}} (|100\rangle_C + |010\rangle_C + |001\rangle_C) \leftrightarrow |000\rangle_C$.

so the state now showcases an unentangled C :

$$|\phi_5\rangle = \frac{1}{\sqrt{3}}|000\rangle_C(|110\rangle_B|PPq\rangle + |101\rangle_B|PqP\rangle + |011\rangle_B|qPP\rangle). \quad (124)$$

It is of note that the cost of this step rises exponentially with the number of registers n , since the possible number of states of B grows as 2^n . We introduce another algorithm named “symmetric-digit-decomposition” discussed in Appendix A 2 to circumvent the issue.

(6) Finally, apply LL in reverse on register B :

$$|\phi_6\rangle = \frac{1}{\sqrt{3}}|000\rangle_C|000\rangle_B(|PPq\rangle + |PqP\rangle + |qPP\rangle). \quad (125)$$

The main difference is, thus, the addition of a new register B to store the positions of the (several) largest modes, so that the superpositions on register C that result on step 4 can be reversed by their invariance under the interchange of qubits pointed to by register B . Notice that this produces the desired states for the other cases with three registers: If there are no repeated momenta, register B will be initialized to $|100\rangle_B$ on step 2 and it will be reversed to $|000\rangle_B$ on step 6, so no changes are produced in step 5. If the three modes were equal, B would be initialized to $|111\rangle_B$ and in step 5 the

symmetrization of step 3 would be applied in reverse, so register C returns to $|000\rangle_C$. Finally, there is also another difference with respect to antisymmetrizers, applying Eq. (38) over an state with m repeated modes produces an extra factor \sqrt{m} which cannot be taken into account in this unitarized versions, so it has to be added to the rotation that creates/destroys particles controlling on the number of repeated modes. Gate costs are discussed in Appendix A 2 and compiled in Table II.

Boson SWAP $bS_j(p, q)$ and *controlled boson exchange gates*.—Similarly to the fermion case, we introduce a boson SWAP gate depending on two modes $bS_j(p, q)$. This allows one to move the largest momentum to and away from the last occupied register, a maneuver needed for the ordered step symmetrizer operations to work.

As an example of its implementation, we use the probe state $|\phi_0\rangle = |p\rangle_4(|p\rangle_3|q\rangle_2|q\rangle_1)_S \equiv |p\rangle|pqq\rangle_S$:

(1) Add two registers B and C of j qubits; mark the positions of q in B and of p in C . Add a register D with $j - 1$ qubits (in this case, $j = 4$)

$$|\phi_1\rangle = |000\rangle_D \frac{1}{\sqrt{3}}(|1100\rangle_C|0011\rangle_B|ppqq\rangle + |1010\rangle_C|0101\rangle_B|ppqq\rangle + |1001\rangle_C|0110\rangle_B|ppqq\rangle). \quad (126)$$

(2) The register with the j th qubit on state 0 marks the interchanges that should occur and it is used to initiate register D using the SDD algorithm:

$$|\phi_2\rangle = \frac{1}{\sqrt{3}} \left[\frac{1}{\sqrt{2}}(|010\rangle_D + |001\rangle_D)|1100\rangle_C|0011\rangle_B|ppqq\rangle + \frac{1}{\sqrt{2}}(|100\rangle_D + |001\rangle_D)|1010\rangle_C|0101\rangle_B|ppqq\rangle + \frac{1}{\sqrt{2}}(|100\rangle_D + |010\rangle_D)|1001\rangle_C|0110\rangle_B|ppqq\rangle \right]. \quad (127)$$

(3) Controlled on each qubit of D , apply a SWAP gate between the corresponding qubits of B and C . Permute also the j th qubit of B and C along with the fourth momentum register with the one pointed by each state in the superpositions of D :

$$|\phi_3\rangle = \frac{1}{\sqrt{3}} \left[\frac{1}{\sqrt{2}}|010\rangle_D|0110\rangle_C|1001\rangle_B|qppq\rangle + \frac{1}{\sqrt{2}}|001\rangle_D|0101\rangle_C|1010\rangle_B|qppq\rangle + \frac{1}{\sqrt{2}}|100\rangle_D|0110\rangle_C|1001\rangle_B|qppq\rangle + \frac{1}{\sqrt{2}}|001\rangle_D|0011\rangle_C|1100\rangle_B|qqpp\rangle + \frac{1}{\sqrt{2}}|100\rangle_D|0101\rangle_C|1010\rangle_B|qppq\rangle + \frac{1}{\sqrt{2}}|010\rangle_D|0011\rangle_C|1100\rangle_B|qqpp\rangle \right], \quad (128)$$

which can be conveniently rearranged:

$$|\phi_3\rangle = \frac{1}{\sqrt{3}} \left[\frac{1}{\sqrt{2}}(|010\rangle_D + |100\rangle_D)|0110\rangle_C|1001\rangle_B|qppq\rangle + \frac{1}{\sqrt{2}}(|001\rangle_D + |100\rangle_D)|0101\rangle_C|1010\rangle_B|qppq\rangle + \frac{1}{\sqrt{2}}(|001\rangle_D + |010\rangle_D)|0011\rangle_C|1100\rangle_B|qqpp\rangle \right]. \quad (129)$$

- (4) The states of C now factorize the invariant superpositions of D under SWAP transformation between qubits marked them, so register D can be uncomputed by applying step 5 above. Registers B and C are uncomputed as they were initialized, so we finally have

$$|\phi_4\rangle = |000\rangle_D |000\rangle_C |000\rangle_B (|q\rangle |ppq\rangle_S). \quad (130)$$

Gate costs are discussed in Appendix A 2 and compiled in Table II.

With these new operators, Eq. (118) for bosons reads

$$\begin{aligned} \mathcal{U}_{10}^f = & \prod_j \left\{ \sum_{\substack{l=0 \\ l \neq j, j-1}} \mathbb{P}_l^{(n)} + \hat{S}_{j \leftarrow j-1}^\dagger \cdot \prod_p [\text{bX}_j^\dagger(p) \cdot \mathbb{P}_0^{(n-j)} \right. \\ & \left. \otimes \mathfrak{U}_{10,j}(\Delta t, \lambda_p) \otimes \mathbb{P}_{j-1}^{(j-1)} \cdot \text{bX}_j(p)] \cdot \hat{S}_{j \leftarrow j-1} \right\} \\ & + \mathcal{O}(\Delta t^2) \end{aligned} \quad (131)$$

with the controlled boson-exchange gate that, in this case, serves only one function (bringing the largest momentum to or away from the j th register for appropriate ordering) but does not need to operate on an empty register, so

$$\text{bX}_j(p) = (\mathfrak{C}_{00})_j \otimes \mathbf{i} + (\mathfrak{C}_{11})_j \otimes \text{bS}_j(L, p). \quad (132)$$

If register j is occupied, the boson-SWAP tries a permutation between the largest mode (in register j by default) and mode p , so it can be annihilated. Mode duplication is not forbidden for bosons, so if register j is empty, no SWAP but the identity is applied.

Gate-level implementation and costs.—The implementation is completely analogous to the fermion case; we just change the fermion operators by their boson counterparts and elevate the corresponding computing costs to Table II. They are

- (i) $\mathcal{O}(N_p)$ $\text{bX}_j(p)$, with a total cost of $\mathcal{O}(j^3 N_p \log_2 j + j N_p \log_2 N_p)$ and $\mathcal{O}(j^2 N_p)$ oracle calls,
- (ii) $\mathcal{O}(1)$ $\hat{S}_{j \leftarrow j-1}^\dagger$, with a cost of $\mathcal{O}(j^3 \log_2 j + j \log_2 N_p)$ CNOTs and single-qubit gates and $\mathcal{O}(j^2)$ oracle calls, and

- (iii) $\mathcal{O}(N_p)$ $\mathfrak{U}_{10,j}(\Delta t, \lambda_p)$ controlled on up most an extra $\mathcal{O}(\log_2 j)$ qubits to correct rotations with mode repetition. This has a total cost of $\mathcal{O}(N_p \log_2^2 N_p)$ CNOTs and single-qubit gates (we ignore the $\mathcal{O}(\log_2 j)$ extra controls since N_p is usually much larger than j).

Addition from $j = 1$ to $j = n$ gives $\mathcal{O}(n^2 N_p \log_2 N_p + N_p \log_2^2 N_p + n^4 N_p \log_2 n)$ CNOTs and single-qubit gates and $\mathcal{O}(n^3 N_p)$ order-oracle calls.

B. Splitting operators

We have now all the elements to work out a splitting operator or vertex interaction. Let us focus, as a master template, on the emission (or absorption) of a boson by a fermion, a process key in QCD gluon emission, nuclear pion emission, and more:

$$\mathcal{U}_{21}(\Delta t, \lambda) = \exp \left[-i \Delta t \sum_{p,k} \lambda_{k-p} (a_{k-p}^\dagger b_p^\dagger b_k + \text{H.c.}) \right]. \quad (133)$$

(The subindices 21 make the particle-number-changing nature of this interaction clear.) The first step is to express the momenta in the above sums in terms of appropriate indices. Setting the annihilated momenta to p_m and the created ones to p_r and p_ξ , the vertex interaction reads

$$\mathcal{U}_{21}(\Delta t, \lambda) = \exp \left[-i \Delta t \sum_{r,m} \lambda_\xi (a_{p_\xi}^\dagger b_{p_r}^\dagger b_{p_m} + \text{H.c.}) \right]_{\xi=m-r}, \quad (134)$$

where the conditions over the indices should be chosen, as we assume henceforth, so that p_r , p_ξ , and p_m are all momenta which can be codified in the grid, just as in Sec. VB. We may as well then drop the p lettering and keep only the mode indices $r, m, \xi \dots$

Boson and fermion states are separately encoded because of their different symmetrization. We, thus, consider n_b registers of bosons and n_f registers of fermions, so that $n = n_b + n_f$, $a_\xi^\dagger = \sum_{j_b} a_{\xi, j_b}^{(n_b)\dagger} \otimes \mathbb{I}^{\otimes n_f}$, and $b_\xi^\dagger = \mathbb{I}^{\otimes n_b} \otimes \sum_{j_f} b_{\xi, j_f}^{(n_f)\dagger}$; thus,

$$\begin{aligned} a_{\xi, j_b}^{(n_b)\dagger} \otimes b_{r, j_f}^{(n_f)\dagger} b_{m, j_f}^{(n_f)} &= a_{\xi, j_b}^{(n_b)\dagger} \otimes \delta_{j_r j_f'} \mathcal{A}_{j_f \leftarrow j_f-1} \cdot \mathbb{P}_0^{(n_f-j_f)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{S}_r^\dagger \mathfrak{S}_m)_{j_f} \otimes \mathbb{P}_{j_f-1}^{(j_f-1)} \cdot \mathcal{A}_{j_f \leftarrow j_f-1} \\ &= \delta_{j_r, j_f'} [\mathcal{S}_{j_b \leftarrow j_b-1} \cdot \mathbb{P}_0^{(n_b-j_b)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{S}_\xi^\dagger)_{j_b} \otimes \mathbb{P}_{j_b-1}^{(j_b-1)}] \\ &\quad \otimes [\mathcal{A}_{j_f \leftarrow j_f-1} \cdot \mathbb{P}_0^{(n_f-j_f)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{S}_r^\dagger \mathfrak{S}_m)_{j_f} \otimes \mathbb{P}_{j_f-1}^{(j_f-1)} \cdot \mathcal{A}_{j_f \leftarrow j_f-1}]. \end{aligned} \quad (135)$$

In the following, we describe two ways of exponentiation of the preceding equation: The first one is analogous to that already used for other terms in H , and particle-exchange symmetry is only partially conserved by the time evolution. Then, in the second paragraph below, exchange-symmetry conservation is imposed by construction.

1. Exponentiation without exact conservation of fermion antisymmetry

Several of the techniques developed earlier are used for exponentiation. First, Eq. (72) can be used to commute the antisymmetrizers, since the number of fermions is conserved:

$$a_{\xi, j_b}^{(n_b)\dagger} \otimes b_{r, j_f}^{(n_f)\dagger} b_{m, j_f}^{(n_f)} = \mathcal{S}_{j_b \leftarrow j_b-1} \cdot \mathbb{P}_0^{(n_b-j_b)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_{\xi}^{\dagger})_{j_b} \otimes \mathbb{P}_{j_b-1}^{(j_b-1)} \\ \otimes \left[\sum_{k=0}^{j_f-1} \mathbb{P}_k^{(n_f+k-j_f)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_r^{\dagger} \mathfrak{s}_m)_{j_f-k} \otimes \mathbb{P}_{j_f-1-k}^{(j_f-1-k)} \right] \cdot \frac{\mathcal{A}_{j_f \leftarrow j_f-1}}{\sqrt{j_f}}. \quad (136)$$

Note that $\mathcal{A}_{j_f-1 \leftarrow j_f-2}$ and $\mathcal{S}_{j_b \leftarrow j_b-1}$ commute, as they act on different parts of the memory. To continue, we turn the sums over j_b and j_f in the exponent into products using the commutation between terms with different j_b or j_f . The antisymmetrizer can be exponentiated out and simplified as in previous sections:

$$\mathcal{U}_{21}(\Delta t, \lambda) = \prod_{j_f, j_b=1} \left(\mathbb{I}^{(n)} + \sum_{l=1}^{\infty} \left[-i\Delta t \sum_{m,r} \lambda_{\xi} \mathcal{S}_{j_b \leftarrow j_b-1} \cdot \mathbb{P}_0^{(n_b-j_b)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_{\xi}^{\dagger})_{j_b} \otimes \mathbb{P}_{j_b-1}^{(j_b-1)} \right. \right. \\ \left. \left. \otimes \left(\sum_{k=0}^{j_f-1} \mathbb{P}_k^{(n_f+k-j_f)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_r^{\dagger} \mathfrak{s}_m)_{j_f-k} \otimes \mathbb{P}_{j_f-k-1}^{(j_f-k-1)} \right) + \text{H.c.} \right]^l \right). \quad (137)$$

We now promote $\mathcal{S}_{j_b \leftarrow j_b-1}$ to a unitary operator by imposing a memory order, add the corresponding inverse operator, and impose an ordering on the boson part of the expression:

$$\mathcal{U}_{21}(\Delta t, \lambda) = \prod_{j_f, j_b=1} \left(\mathbb{I}^{(n)} + \sum_{l=1}^{\infty} \left[-i\Delta t \sum_{m,r} \lambda_{\xi} \hat{\mathcal{S}}_{j_b \leftarrow j_b-1}^{\dagger} \cdot (\mathbb{P}_0^{(n_b-j_b)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_{\xi}^{\dagger})_{j_b} \otimes \mathbb{P}_{j_b-1}^{(j_b-1)})_{b, \text{ord}} \cdot \hat{\mathcal{S}}_{j_b \leftarrow j_b-1} \right. \right. \\ \left. \left. \otimes \left(\sum_{k=0}^{j_f-1} \mathbb{P}_k^{(n_f+k-j_f)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_r^{\dagger} \mathfrak{s}_m)_{j_f-k} \otimes \mathbb{P}_{j_f-k-1}^{(j_f-k-1)} \right) + \text{H.c.} \right]^l \right). \quad (138)$$

Since symmetrizers do not transform the fermion part of the memory, we can pass $\hat{\mathcal{S}}_{j_b \leftarrow j_b-1}$ to the right:

$$\mathcal{U}_{21}(\Delta t, \lambda) = \prod_{j_b, j_f} \left(\mathbb{I}^{(n)} + \hat{\mathcal{S}}_{j_b \leftarrow j_b-1}^{\dagger} \cdot \mathbb{P}_0^{(n_b-j_b)} \otimes \sum_{l=1}^{\infty} \left[-i\Delta t \sum_{m,r} \lambda_{\xi} \sum_{k=0}^{j_f-1} ((\mathfrak{C}_{10} \otimes \mathfrak{s}_{\xi}^{\dagger})_{j_b} \otimes \mathbb{P}_{j_b-1}^{(j_b-1)})_{b, \text{ord}} \right. \right. \\ \left. \left. \otimes \mathbb{P}_k^{(n_f+k-j_f)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_r^{\dagger} \mathfrak{s}_m)_{j_f-k} \otimes \mathbb{P}_{j_f-k-1}^{(j_f-k-1)} + \text{H.c.} \right]^l \cdot \hat{\mathcal{S}}_{j_b \leftarrow j_b-1} \right). \quad (139)$$

To continue, idempotent terms are factorized from the sum over l :

$$\mathcal{U}_{21}(\Delta t, \lambda) = \prod_{j_b, j_f=1} \left(\mathbb{I}^{(n)} + \hat{\mathcal{S}}_{j_b \leftarrow j_b-1}^{\dagger} \cdot \mathbb{P}_0^{(n_b-j_b)} \otimes \{ \mathfrak{C}_{11}^{\otimes(j_b-1)} \otimes \mathfrak{C}_{00}^{\otimes(n_f-j_f)} \otimes \mathfrak{C}_{11}^{\otimes j_f} \} \right. \\ \left. \times \left\{ \sum_{l=1}^{\infty} \left[-i\Delta t \sum_{m,r} \lambda_{\xi} \sum_{k=0}^{j_f-1} ((\mathfrak{C}_{10} \otimes \mathfrak{s}_{\xi}^{\dagger})_{j_b} \otimes \cdots \otimes (\mathbf{i})_1)_{b, \text{ord}} \otimes (\mathfrak{s}_r^{\dagger} \mathfrak{s}_m)_{j_f-k} + \text{H.c.} \right]^l \cdot \hat{\mathcal{S}}_{j_b \leftarrow j_b-1} \right\} \right), \quad (140)$$

where we have omitted the identity operators over the fermion momentum qubits. To resum the exponential, we subtract from the identity $\mathbb{I}^{(n)}$ the terms corresponding to $l = 0$, and since antisymmetrizers and symmetrizers commute with projectors, we arrive at

$$\begin{aligned}
\mathcal{U}_{21}(\Delta t, \lambda) &= \prod_{j_b, j_f=1} (\mathbb{I}^{(n)} - \mathbb{P}_{j_b}^{(n_b)} \otimes \mathbb{P}_{j_f}^{(n_f)} - \mathbb{P}_{j_b-1}^{(n_b)} \otimes \mathbb{P}_{j_f}^{(n_f)}) \\
&\quad + \hat{\mathcal{S}}_{j_b \leftarrow j_b-1}^\dagger \cdot \mathbb{P}_0^{(n_b-j_b)} \otimes \{ \mathfrak{C}_{11}^{\otimes(j_b-1)} \otimes \mathfrak{C}_{00}^{\otimes(n_f-j_f)} \otimes \mathfrak{C}_{11}^{\otimes j_f} \} \{ \mathbf{u}_{21, (j_b, j_f)} \} \cdot \hat{\mathcal{S}}_{j_b \leftarrow j_b-1} \\
&= \prod_{j_b, j_f=1} (\mathbb{I}^{(n)} - \mathbb{P}_0^{(n_b-j_b)} \otimes \mathbb{I} \otimes \mathbb{P}_{j_b-1}^{(j_b-1)} \otimes \mathbb{P}_{j_f}^{(n_f)}) \\
&\quad + \hat{\mathcal{S}}_{j_b \leftarrow j_b-1}^\dagger \cdot \mathbb{P}_0^{(n_b-j_b)} \otimes \{ \mathfrak{C}_{11}^{\otimes(j_b-1)} \otimes \mathfrak{C}_{00}^{\otimes(n_f-j_f)} \otimes \mathfrak{C}_{11}^{\otimes j_f} \} \{ \mathbf{u}_{21, (j_b, j_f)} \} \cdot \hat{\mathcal{S}}_{j_b \leftarrow j_b-1} \quad (141)
\end{aligned}$$

with

$$\mathbf{u}_{21, j}(\Delta t, \lambda) = \exp \left[-i\Delta t \sum_{m, r} \lambda_\xi \sum_{k=0}^{j_f-1} ((\mathfrak{C}_{10} \otimes \mathfrak{s}_\xi^\dagger)_{j_b} \otimes \cdots \otimes (\mathbf{i})_{1, b, ord} \otimes (\mathfrak{s}_r^\dagger \mathfrak{s}_m)_{j_f-k} + \text{H.c.} \right]. \quad (142)$$

The operator of Eq. (140) respects the fermion antisymmetry and boson symmetry of input states exactly, but, as it acts over the entire boson memory, we expect its implementation to be (unnecessarily) expensive. Back in Sec. VB, the exponentiation of the sum over registers (index k) was decomposed exactly in terms of the products of exponentials, since each commutator with λ_ξ canceled a commutator with $\pm\lambda_{-\xi}$. This is not the case here because of the creation and annihilation terms; thus, terms with different k do not commute.

We can then use Trotter's formula, discretizing the time evolution by some Δt to obtain an approximate decomposition which implies that exchange symmetry for fermions is approximated only up to $O(\Delta t^2)$. Note that the exchange symmetry for bosons is exact, as ensured by the application of the step symmetrizer.

Furthermore, the operator is decomposed in terms of fixed emitted boson-momentum ξ pieces to implement the boson creation and annihilation terms as was done for the tadpole in Sec. VIA:

$$\begin{aligned}
\mathcal{U}_{21}(\Delta t, \lambda) &= \prod_{j_b, j_f=1} \left(\mathbb{I}^{(n)} - \mathbb{P}_0^{(n_b-j_b)} \otimes \mathbb{I} \otimes \mathbb{P}_{j_b-1}^{(j_b-1)} \otimes \mathbb{P}_{j_f}^{(n_f)} + \hat{\mathcal{S}}_{j_b \leftarrow j_b-1}^\dagger \cdot \prod_{\xi} \left[\text{bX}_{j_b}^\dagger(\xi) \cdot \mathbb{P}_0^{(n_b-j_b)} \right. \right. \\
&\quad \left. \left. \otimes \left\{ \mathfrak{C}_{11}^{\otimes(j_b-1)} \otimes \mathfrak{C}_{00}^{\otimes(n_f-j_f)} \otimes \mathfrak{C}_{11}^{\otimes j_f} \right\} \left\{ \prod_{k=0}^{j_f-1} \mathbf{u}_{21, (j_b, j_f-k)} \right\} \cdot \text{bX}_{j_b}(\xi) \right] \cdot \hat{\mathcal{S}}_{j_b \leftarrow j_b-1} \right) + \mathcal{O}(\Delta t^2), \quad (143)
\end{aligned}$$

with

$$\mathbf{u}_{21, (j_b, j_f-k)}(p_\xi; \Delta t, \lambda) = \exp \left[-i\Delta t \sum_{m, r} \lambda_\xi (\mathfrak{C}_{10} \otimes \mathfrak{s}_\xi^\dagger)_{j_b} \otimes (\mathfrak{s}_r^\dagger \mathfrak{s}_m)_{j_f-k} + \text{H.c.} \right]. \quad (144)$$

Such decomposition of the exponential in a product over the fermion registers indexed by k avoids duplication of momenta *up to second-order terms* in the Trotter expansion. That is, the Pauli exclusion principle is recovered for $\Delta t \rightarrow 0$.

$\mathcal{U}_{21}(\Delta t, \lambda)$ can be further simplified: Operators outside the braces do not depend on j_f and can be factored out of the corresponding product; terms with different j_f are orthogonal, and controlled operations over properly filled memories [as defined below Eq. (53)] can be simplified. The final operator reads

$$\begin{aligned}
\mathcal{U}_{21}(\Delta t, \lambda) &= \prod_{j_b} \left(\mathbb{I}^{(n)} - \mathbb{P}_0^{(n_b-j_b)} \otimes \mathbb{I} \otimes \mathbb{P}_{j_b-1}^{(j_b-1)} \otimes \mathbb{I}^{(n_f)} \right. \\
&\quad \left. + \hat{\mathcal{S}}_{j_b \leftarrow j_b-1}^\dagger \cdot \prod_{\xi} [\text{bX}_{j_b}^\dagger(\xi) \cdot \hat{\mathbf{U}}_{21, j_b}(p_\xi; \Delta t, \lambda) \cdot \text{bX}_{j_b}(\xi)] \cdot \hat{\mathcal{S}}_{j_b \leftarrow j_b-1} \right) + \mathcal{O}(\Delta t^2), \quad (145)
\end{aligned}$$

where the auxiliary operator $\hat{\mathbf{U}}_{21, j_b}$ is

$$\hat{\mathbf{U}}_{21, j_b}(p_\xi; \Delta t, \lambda) = \prod_{j_f=1}^{n_f} (\mathfrak{C}_{00})_{j_b+1} \otimes \{ (\mathfrak{C}_{11})_{j_b-1} \otimes [(\mathfrak{C}_{11})_{j_f}] \{ \mathbf{u}_{21, (j_b, j_f)}(p_\xi; \Delta t, \lambda) \} + (\mathfrak{C}_{00})_{j_f} \} \{ \mathbb{I} \}, \quad (146)$$

where remaining operators acting on presence qubits are all identities. Note the special combination of square and curly brackets; it denotes that the terms to the left of the square brackets tensor multiply both terms of the sum to the right.

Gate-level implementation and costs.—Figure 4 is the circuit implementation of Eq. (145), while the subcircuit Eq. (146) is found in Fig. 5. The only unknown costs are those of \mathcal{U}_{21,j_b} , which, in turn, are composed of

$\mathcal{O}(n_f^2)\mathcal{U}_{21,(j_b,j_f)}$ gates, whose implementation is similar to that of the exchange term; see Sec. VB.

Each $\mathcal{U}_{21,(j_b,j_f)}$ gate is a single rotation applied to the registers j_b (on the boson memory section) and j_f (on the fermion one). For a fixed p_ξ , we can collect the basis states which are affected by the transformation of the set of registers (only the relevant j_b th and j_f th registers are explicitly written, in pairs with the j_b boson register either empty or occupied)

$$A_{p_\xi} = \{ |\Omega\rangle_{j_b} |1 \max(p_{-\Lambda} + p_\xi, p_{-\Lambda})\rangle_{j_f}, |1 p_\xi\rangle_{j_b} |1 \max(p_{-\Lambda}, p_{-\Lambda} - p_\xi)\rangle_{j_f}, \dots, \\ |\Omega\rangle_{j_b} |1 \max(p_{-\Lambda} + p_\xi, p_{-\Lambda}) + p_k\rangle_{j_f}, |1 p_\xi\rangle_{j_b} |1 \max(p_{-\Lambda}, p_{-\Lambda} - p_\xi) + p_k\rangle_{j_f}, \dots, \\ |\Omega\rangle_{j_b} |1 \min(p_\Lambda + p_\xi, p_\Lambda)\rangle_{j_f}, |1 p_\xi\rangle_{j_b} |1 \min(p_\Lambda - p_\xi, p_\Lambda)\rangle_{j_f} \}, \quad (147)$$

where the momentum cutoffs are respected due to the “max” and “min” selections. $\mathcal{U}_{21,(j_b,j_f)}$ is diagonalized with a change of basis T_{p_ξ,p_k} ($p_\xi > 0$) to a linear combination of the fermion before the splitting and the fermion + boson pair (sharing the momentum) after it:

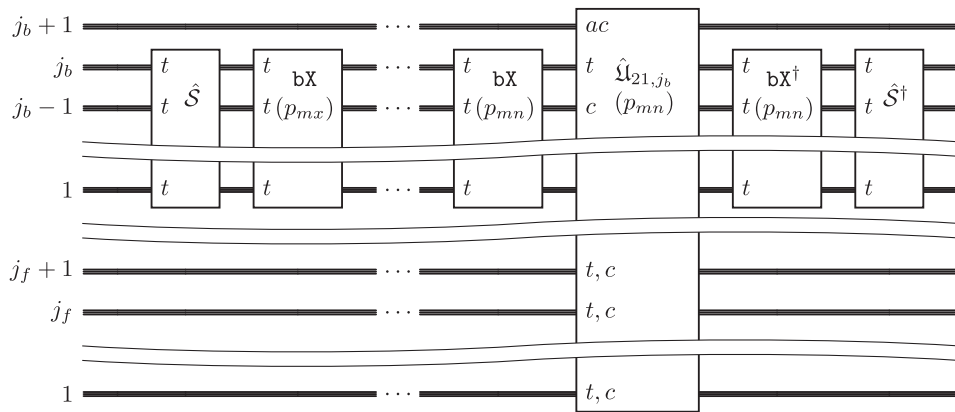


FIG. 4. Circuit implementation of $\mathcal{U}_{21}(\Delta t, \lambda)$ without exact conservation of fermion antisymmetry, given by Eq. (145). See Fig. 1 for an explanation of the t , c , and ac symbols.

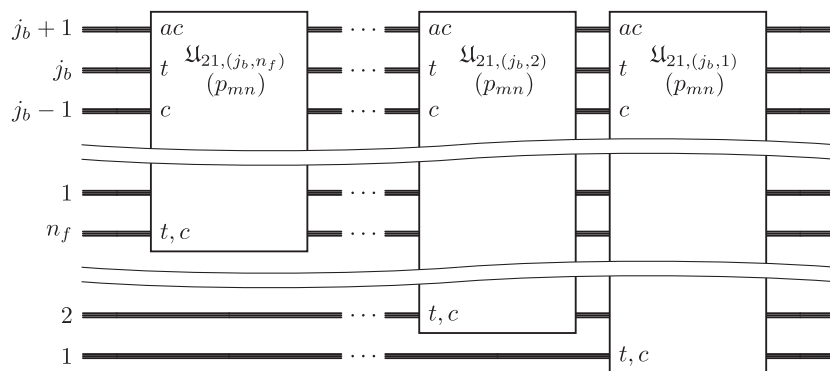


FIG. 5. Circuit implementation of $\hat{\mathcal{U}}_{21,j_b}(p_{mn}; \Delta t, \lambda)$, subroutine of $\mathcal{U}_{21}(\Delta t, \lambda)$, given by Eq. (146). See Fig. 1 for an explanation of the t , c , and ac symbols.

$$\begin{aligned}
T_{p_\xi, p_k}^{-1} |\Omega\rangle_{j_b} |1p_{-\Lambda} + p_\xi + p_k\rangle_{j_f} &= \frac{1}{\sqrt{2}} (|\Omega\rangle_{j_b} |1p_{-\Lambda} + p_\xi + p_k\rangle_{j_f} + |1p_\xi\rangle_{j_b} |1p_{-\Lambda} + p_k\rangle_{j_f}), \\
T_{p_\xi, p_k}^{-1} |1p_\xi\rangle_{j_b} |1p_{-\Lambda} + p_k\rangle_{j_f} &= \frac{1}{\sqrt{2}} (|\Omega\rangle_{j_b} |1p_{-\Lambda} + p_\xi + p_k\rangle_{j_f} - |1p_\xi\rangle_{j_b} |1p_{-\Lambda} + p_k\rangle_{j_f}),
\end{aligned} \tag{148}$$

with $p_k \in [0, p_\Lambda - p_{-\Lambda} - p_\xi]$. For each p_k and p_ξ pair, the T transformation can be implemented using a Gray code, applying a controlled Hadamard, and reversing the Gray code, resulting in a cost of $\mathcal{O}(\log_2^2 N_p)$. Composition of transformations for different p_k gives rise to a gate T_{p_ξ} with a total cost of $\mathcal{O}(N_p \log_2^2 N_p)$ CNOTs and single-qubit gates. $\mathcal{U}_{21, (j_b, j_f)}(p_s)$ is then diagonal in the new basis, explicitly:

$$\begin{aligned}
T_{p_s}^{-1} \mathcal{U}_{21, (j_b, j_f)}(p_s) T_{p_s} &= \exp \left[-i\Delta t \lambda_s (|\Omega\rangle\langle\Omega|)_{j_b} \otimes \left(\sum_{p_r=\max(p_{-\Lambda}, p_{-\Lambda}+p_s)}^{\min(p_\Lambda, p_\Lambda+p_s)} \mathfrak{S}_{p_r}^\dagger \mathfrak{S}_{p_r} \right)_{j_f} \right. \\
&\quad \left. + i\Delta t \lambda_s (|1p_s\rangle\langle 1p_s|)_{j_b} \otimes \left(\sum_{p_r=\max(p_{-\Lambda}, p_{-\Lambda}-p_s)}^{\min(p_\Lambda, p_\Lambda-p_s)} \mathfrak{S}_{p_r}^\dagger \mathfrak{S}_{p_r} \right)_{j_f} \right];
\end{aligned} \tag{149}$$

both terms in the exponential commute and can be separately implemented.

The p_r sums over fermion $\mathfrak{S}_{p_r}^\dagger \mathfrak{S}_{p_r}$ pairs are projectors, which ensure that the transformation is restricted to the subspace of codified momenta for a given value of p_ξ ; if the cutoffs were removed, they would turn to identities by the completeness relation. Thus, we can implement Eq. (149) by first testing whether we are in the subspace of codified momenta, saving the result in an ancillary qubit, applying the corresponding diagonal rotation controlled on that, and finally returning the ancillary qubit to its initial state, with a total cost of $\mathcal{O}(N_p \log_2 N_p)$ CNOTs and single-qubit gates. Remember that the rotation angle should be modified depending on the number of repeated modes in the boson memory, which adds negligible extra costs in the asymptotic limit. Thus, the costs of $\mathcal{U}_{21, (j_b, j_f)}(p_i)$ are dominated by the change of basis, so the operator $\mathcal{U}_{21, (j_b)}(p_i)$ needs $\mathcal{O}(n_f^2 N_p \log_2^2 N_p)$ CNOTs and single-qubit gates.

To conclude, for each j_b on Eq. (145) there are (see Table II for individual costs):

- (i) $\mathcal{O}(N_p) \text{bX}_j(p)$, with a total cost of $\mathcal{O}(j_b^3 N_p \log_2 j_b + j_b N_p \log_2 N_p)$ and $\mathcal{O}(j_b^2 N_p)$ oracle calls,
- (ii) $\mathcal{O}(1) \hat{S}_{j_b \leftarrow j_b - 1}^\dagger$, with a cost of $\mathcal{O}(j_b^3 \log_2 j_b + j_b \log_2 N_p)$ CNOTs and single-qubit gates and $\mathcal{O}(j_b^2)$ oracle calls, and
- (iii) $\mathcal{O}(N_p) \mathcal{U}_{21, (j_b)}(p_i)$ with a total cost of $\mathcal{O}(n_f^2 N_p^2 \log_2^2 N_p)$ CNOTs and single-qubit gates.

Summing over j_b , the final cost to implement \mathcal{U}_{21} is $\mathcal{O}(n_b n_f^2 N_p^2 \log_2^2 N_p)$ CNOTs and single-qubit gates and $\mathcal{O}(n_b^3 N_p)$ order-oracle calls.

2. Exponentiation with exact conservation of fermion antisymmetry

We now discuss an alternative route of exponentiation that preserves antisymmetry by construction although at the expense of an increase in its implementation costs. The key idea is to promote the step antisymmetrizers \hat{A} to unitary operators instead of commuting them with Eq. (72), so that

$$\begin{aligned}
a_{s, j_b}^{(n_b)^\dagger} \otimes b_{r, j_f}^{(n_f)^\dagger} b_{m, j_f - 1}^{(n_f)} &= \hat{S}_{j_b \leftarrow j_b - 1}^\dagger \cdot \hat{A}_{j_f \leftarrow j_f - 1}^\dagger \cdot (\mathbb{P}_0^{(n_b - j_b)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{S}_s^\dagger)_{j_b} \otimes \mathbb{P}_{j_b - 1}^{(j_b - 1)})_{b, \text{ord}} \\
&\quad \otimes (\mathbb{P}_0^{(n_f - j_f)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{S}_r^\dagger \mathfrak{S}_m)_{j_f} \otimes \mathbb{P}_{j_f - 1}^{(j_f - 1)})_{f, \text{ord}} \cdot \hat{A}_{j_f \leftarrow j_f - 1} \cdot \hat{S}_{j_b \leftarrow j_b - 1},
\end{aligned} \tag{150}$$

adding the Hermitian conjugate and the bX gate

$$\begin{aligned}
a_{s, j_b}^{(n_b)^\dagger} \otimes b_{r, j_f}^{(n_f)^\dagger} b_{m, j_f - 1}^{(n_f)} + h.c. &= \hat{S}_{j_b \leftarrow j_b - 1}^\dagger \cdot \hat{A}_{j_f \leftarrow j_f - 1}^\dagger \cdot \mathbb{P}_0^{(n_b - j_b)} \otimes \{ \mathfrak{C}_{11}^{\otimes (j_b - 1)} \otimes \mathfrak{C}_{00}^{\otimes (n_f - j_f)} \} \\
&\quad \times \{ \text{bX}_{j_b}^\dagger(s) \cdot [(\mathfrak{C}_{10} \otimes \mathfrak{S}_s^\dagger)_{j_b} \otimes (\mathbf{i}^{\otimes (n_f - j_f)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{S}_r^\dagger \mathfrak{S}_m)_{j_f} \otimes \mathbb{P}_{j_f - 1}^{(j_f - 1)})_{f, \text{ord}} + h.c.] \cdot \text{bX}_{j_b}(s) \} \\
&\quad \cdot \hat{A}_{j_f \leftarrow j_f - 1} \cdot \hat{S}_{j_b \leftarrow j_b - 1},
\end{aligned} \tag{151}$$

To avoid duplication of any fermion momenta and ensure the appropriate ordering of fermion registers upon rotation between states $|\Omega\rangle_{j_b}|1m\rangle_{j_f}$ and $|s\rangle_{j_b}|1r\rangle_{j_f}$ (simultaneously with creation of the new boson, it behooves one to check that the fermion with changed momentum is not yet in the memory), we use the following algorithm.

- (1) Apply $\hat{\mathcal{A}}_{j_f \leftarrow j_f - 1}$ to unpack from the antisymmetric state the largest momentum of the fermion memory and move it to the last-used register j_f ; then apply $\text{fS}_{j_f}(L, m)$ if register j_b is empty and $\text{fS}_{j_f}(L, r)$ if it is occupied.
- (2) Apply $\text{fS}(r, m)$. In this way, $|\Omega\rangle_{j_b}|1m\rangle_{j_f}$ originated from step 1 changes to $|\Omega\rangle_{j_b}|1r\rangle_{j_f}$ if r is also on memory; similarly, $|1s\rangle_{j_b}|1r\rangle_{j_f}$ changes to $|1s\rangle_{j_b}|1m\rangle_{j_f}$ if m is also on memory.
- (3) Apply to registers j_b and j_f the two-register “splitting” rotation, that attempts to rotate between

$|\Omega\rangle_{j_b}|1m\rangle_{j_f}$ and $|1s\rangle_{j_b}|1r\rangle_{j_f}$; the rotation is, thus, performed provided r is not already on memory when register j_b is empty; but when it is occupied, if m is not in memory. This avoids the situations in which momenta duplication would occur.

- (4) Undo the first two fSs.

As previously for the fermion and boson SWAPs (see Secs. VIA 1 and VIA 3), the algorithm can be written in terms of fermion-SWAP-gate controlled multiplications; for the first two steps,

$$\text{fSp}_{(j_b, j_f)}(m, r) = \text{fS}_{j_f}(m, r) \cdot [(\mathfrak{C}_{00})_{j_b} \otimes \text{fS}_{j_f}(L, m) + (\mathfrak{C}_{11})_{j_b} \otimes \text{fS}_{j_f}(L, r)], \quad (152)$$

returning to the splitting operator of Eq. (150), and including its (boson absorption) Hermitian conjugate operator,

$$\begin{aligned} a_{s, j_b}^{(n_b)^\dagger} \otimes b_{r, j_f}^{(n_f)^\dagger} b_{m, j_f - 1}^{(n_f)} + h.c. &= \hat{\mathcal{S}}_{j_b \leftarrow j_b - 1}^\dagger \cdot \hat{\mathcal{A}}_{j_f \leftarrow j_f - 1}^\dagger \cdot \text{bX}_{j_b}^\dagger(s) \cdot \text{fSp}_{(j_b, j_f)}^\dagger(m, r) \cdot \\ &\mathbb{P}_0^{(n_b - j_b)} \otimes \{ \mathfrak{C}_{11}^{\otimes (j_b - 1)} \otimes \mathfrak{C}_{00}^{\otimes (n_f - j_f)} \} \{ [(\mathfrak{C}_{10} \otimes \mathfrak{s}_s^\dagger)_{j_b} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_r^\dagger \mathfrak{s}_m)_{j_f} + h.c.] \} \otimes \mathbb{P}_{j_f - 1}^{(j_f - 1)} \\ &\cdot \text{fSp}_{(j_b, j_f)}(m, r) \cdot \text{bX}_{j_b}(s) \cdot \hat{\mathcal{A}}_{j_f \leftarrow j_f - 1} \cdot \hat{\mathcal{S}}_{j_b \leftarrow j_b - 1}, \end{aligned} \quad (153)$$

where the \mathfrak{i} operators on fermion and boson registers have been omitted. The exponentiation of this term is now straightforward:

$$\begin{aligned} \mathcal{U}_{21}(\Delta t, \lambda) &= \prod_{j_b, j_f=1} (\mathbb{I}^{(n)} - \mathbb{P}_0^{(n_b - j_b)} \otimes \mathbb{I} \otimes \mathbb{P}_{j_b - 1}^{(j_b - 1)} \otimes \mathbb{P}_{j_f}^{(n_f)}) \\ &+ \hat{\mathcal{S}}_{j_b \leftarrow j_b - 1}^\dagger \cdot \hat{\mathcal{A}}_{j_f \leftarrow j_f - 1}^\dagger \cdot \prod_s [\text{bX}_{j_b}^\dagger(s) \cdot \hat{\mathcal{U}}_{21, (j_b, j_f)}(s; \lambda, \Delta t) \cdot \text{bX}_{j_b}(s) \cdot \hat{\mathcal{A}}_{j_f \leftarrow j_f - 1} \cdot \hat{\mathcal{S}}_{j_b \leftarrow j_b - 1}] + \mathcal{O}(\Delta t^2), \end{aligned} \quad (154)$$

where the auxiliary operator $\hat{\mathcal{U}}_{21, (j_b, j_f)}(s)$ is

$$\begin{aligned} \hat{\mathcal{U}}_{21, (j_b, j_f)}(s; \lambda, \Delta t) &= \prod_r (\text{fSp}_{(j_b, j_f)}^\dagger(m, r) \cdot \\ &\cdot \mathbb{P}_0^{(n_b - j_b)} \otimes \{ \mathfrak{C}_{11}^{\otimes j_b - 1} \otimes \mathfrak{C}_{00}^{\otimes j_f + 1} \} \{ \mathcal{U}_{21, (j_b, j_f)}(r, s; \lambda, \Delta t) \} \otimes \mathbb{P}_{j_f - 1}^{(j_f - 1)} \cdot \text{fSp}_{(j_b, j_f)}(m, r)). \end{aligned} \quad (155)$$

Gate-level implementation and costs.—Equations (154) and (155) are implemented on circuits in Figs. 6 and 7, respectively. The gate count scales as follows, for each j_b and j_f on Eq. (154):

- (1) $\mathcal{O}(1)$ $\hat{\mathcal{S}}, \hat{\mathcal{S}}^\dagger, \hat{\mathcal{A}}, \hat{\mathcal{A}}^\dagger$, with a total cost of $\mathcal{O}(j_b \log_2 N_p + j_f \log_2 N_p)$ CNOT, single-qubit gates, and $\mathcal{O}(j_b^2 + j_f^2)$ order-oracle calls,
- (2) $\mathcal{O}(N_p)$ bX gates, with a total cost of $\mathcal{O}(j_b N_p \log_2 N_p)$ CNOT and single-qubit gates + $\mathcal{O}(j_b^2 N_p)$ order-oracle calls, and
- (3) $\mathcal{O}(N_p) \mathcal{U}_{21, (j_b, j_f)}$, each composed of

- (a) $\mathcal{O}(N_p)$ fSp, in total a cost of $\mathcal{O}(j_f N_p \log_2 N_p)$ CNOT and single-qubit gates + $\mathcal{O}(j_f^2 N_p)$ order-oracle calls; and
- (b) $\mathcal{O}(N_p) \mathcal{U}_{21, (j_b, j_f)}(p_r, p_s)$, implemented as in Sec. VIB 1, but with a change of basis dependent on both p_s and p_k : T_{p_s, p_k} . The cost of each of term is $\mathcal{O}(N_p \log_2 N_p)$, giving a total count of $\mathcal{O}(N_p^2 \log_2 N_p)$.

Thus, the subcircuit costs are $\mathcal{O}(N_p^3 \log_2 N_p)$ CNOT and single-qubit gates + $\mathcal{O}(j_f^2 N_p^2)$ order-oracle calls.

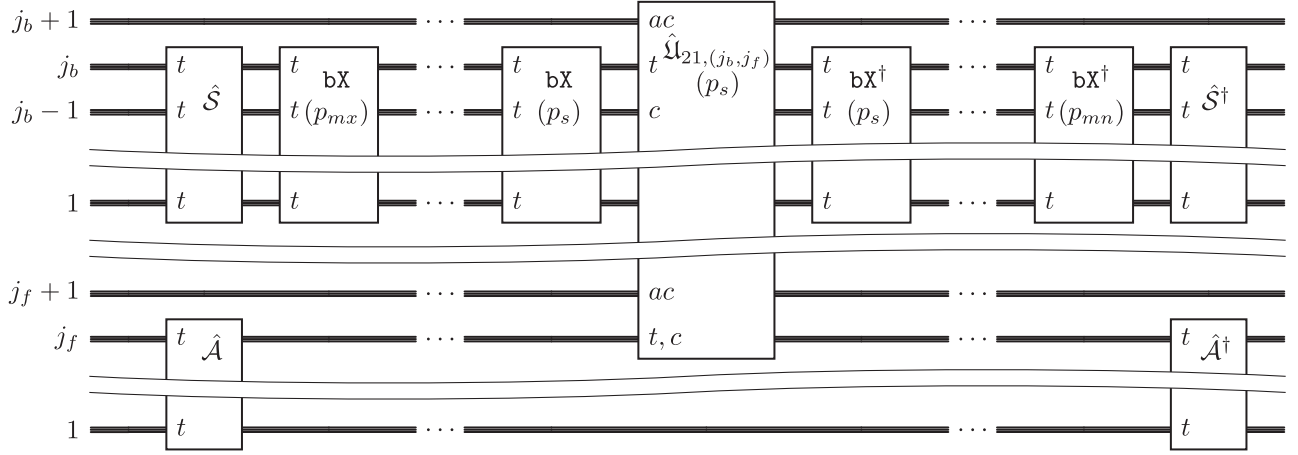


FIG. 6. Circuit implementation of $\mathcal{U}_{21}(\Delta t, \lambda)$ for fixed j_b and j_f with exact conservation of fermion antisymmetry, given by Eq. (154). See Fig. 1 for an explanation of the t , c , and ac symbols.

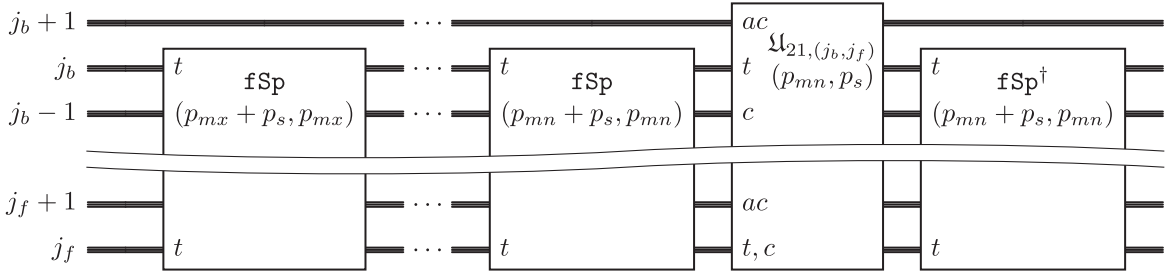


FIG. 7. Circuit implementation of $\mathfrak{U}_{21,(j_b,j_f)}(p_s; \lambda, \Delta t)$ given by Fig. 1, subroutine of $\mathcal{U}_{21}(\Delta t, \lambda)$. See Fig. 1 for an explanation of the t , c , and ac symbols.

Summing over the n_f values of j_f and n_b values of j_b , the dominant costs are $\mathcal{O}(n_f n_b N_p^3 \log_2 N_p)$ CNOT and single-qubit gates + $\mathcal{O}(n_f^2 n_b N_p^2)$ order-oracle calls.

This concludes our exemplification of typical terms in a field-theoretical Hamiltonian written in the particle basis. Further operators can be encoded as needed following the same principles.

VII. COMPILATION OF RESULTS AND DISCUSSION

We are aware that this report makes for a taxing read, given the detailed implementation and discussion, so it is worth providing a clear and concise final overview of what has been achieved.

In this section, we recapitulate the most important operators discussed and their cost in memory, computer time surrogated to the number of single-qubit and CNOT gates (see Table I), and we offer a few final comments.

A. Memory scalability

In the first place, we observe that the memory requirements are dominated by the momentum discretization; to

codify a number of momenta $N_p = \Lambda_{\max} - \Lambda_{\min} + 1 = 2\Lambda + 1$, we need $\lceil \log_2(N_p) \rceil$ qubits; with a maximum number of n particles (registers), we have

$$\text{no. qubits} = \mathcal{O}(n \times \lceil \log_2(N_p) \rceil), \quad (156)$$

and the other quantum numbers of interest (color, spin, flavor, etc.) have a fixed and small memory requirement, so they add subleading $\mathcal{O}(n)$ qubits, as do the qubits allotted for presence and absence controls. We then proceed to the more complicated counting of the number of gates.

B. Gate costs

Tables I and II compile some key operators developed through the manuscript, noting the section where they are defined and their implementation cost in terms of CNOT, single-qubit gates, and order-oracle calls.

A couple of observations are in order. First, let us note that all the costs listed are at most polynomial in the number of active particles j , the size of the momentum discretization implemented N_p , and the maximum number of registers n . While many algorithms can surely be improved, no exponential-cost hurdles appear. This is plausible in

TABLE I. CNOT and single-qubit gate costs of implementing one step of the unitary evolution (with associated Hamiltonian terms \mathcal{H}_{ij}), i being the number of outgoing particles and j that of incoming ones (or vice versa). The variable $n = n_f + n_b$ includes n_f , the number of fermion registers, and n_b , the number of boson registers. N_p stands for the number of different values of the momentum.

Operator	Costs		
	CNOT and single-qubit	Order ORACLE	Section
\mathcal{U}_{11}	$\mathcal{O}(nN_p)$	None	VA
\mathcal{U}_{22}	$\mathcal{O}(n^2N_p^3 \log_2^2 N_p)$	None	VB
\mathcal{U}_{10}	$\mathcal{O}(N_p \log_2^2 N_p)$	$\mathcal{O}(n^3N_p)$	VIA 2
\mathcal{U}_{21}	$\mathcal{O}(n_b n_f^2 N_p^2 \log_2^2 N_p)$	$\mathcal{O}(n_b^3 N_p)$	VIB 1
\mathcal{U}'_{21}	$\mathcal{O}(n_b n_f N_p^3 \log_2 N_p)$	$\mathcal{O}(n_b n_f^3 N_p^2)$	VIB 2

theories where the interactions depend on a finite number of registers (for example, in chromodynamics, the largest number is in the four-gluon vertex; in chiral perturbation theory, the three-body interactions appear at NNLO; there is a large class of few-body problems for which this is the case). The second thing to note is that, though we need propagators which act on all qubits of the memory, if the number of particles is not saturated ($j < n$), the additional, empty registers cost little, as the various gates over different combinations of registers can run in parallel.

It is interesting to compare the efficiency of this implementation of a second quantized theory with other traditional approaches, to which we now dedicate a paragraph.

C. Comparison with Jordan-Wigner (JW) transformation

A word of comparison with a benchmark method is now in order, and when dealing with second quantization, the Jordan-Wigner transformation comes to the fore. An encoding of a physical problem in a digital quantum

computer (by either technique) consists of a mapping between a given finite Hilbert space (a truncation of the full original one as needed) and the quantum computer's memory:

$$J: H \rightarrow (B^{1/2})^{\otimes n}, \quad (157)$$

with $B^{1/2}$ the Hilbert space of a qubit and n some integer number which scales with the dimension of H . The states in H should be wisely chosen to represent the different configurations of the system to be simulated. For example, for quantum chemistry the STO-nG atomic orbital basis is used [37], whereas for a quark model using the Cornell potential, generalized Laguerre polynomials are enough [38,39], at least for heavy quarks. The common feature of these simulations is that a small number of basis functions or modes are chosen to represent the system, each associated with creation and annihilation operators. Single-particle states are then

$$|\psi_i\rangle = b_i^\dagger |0\rangle, \quad (158)$$

while two-particle states are

$$|\psi_{ij}\rangle = \pm |\psi_{ji}\rangle = b_i^\dagger b_j^\dagger |0\rangle, \quad (159)$$

etc. The fermionic or bosonic nature of the states is expressed as the relative sign between $|\psi_{ij}\rangle$ and $|\psi_{ji}\rangle$. Fermions carry a negative sign, which is a way of considering the anticommutation relations $\{b_i^\dagger, b_j^\dagger\} = \{b_i, b_j\} = 0$ and $\{b_i, b_j^\dagger\} = \delta_{ij}$. Since $|\psi_{ij}\rangle = -|\psi_{ji}\rangle$, states of the form $|\psi_{ii}\rangle$ are excluded, which means that *each mode is either occupied or empty*. A general fermionic state can, therefore, be written as

$$|f_n, \dots, f_1\rangle = b_n^{\dagger f_n} \dots b_2^{\dagger f_2} b_1^{\dagger f_1} |0\rangle, \quad (160)$$

in the occupation-number basis, with $f_i \in 0, 1$, $b_n^{\dagger 0} = 1$, and $b_n^{\dagger 1} = b_n^\dagger$. For the annihilation operators

TABLE II. CNOT and single-qubit gate costs of implementing various operators introduced in Sec. VI; j is the number of active (occupied) registers and N_p the number of different momenta. The fermion and boson SWAPs can be implemented in a cheaper manner when the largest momentum L is not interchanged.

Operator	Costs		
	CNOT and single-qubit	Order ORACLE	Section
$\hat{\mathcal{A}}_{j \leftarrow j-1}$	$\mathcal{O}(j^2 + j \log_2 N_p)$	$\mathcal{O}(j^2)$	Sec. VIA 1
$fS_j(p, q)/fS_j(p, L)$	$\mathcal{O}(j \log_2 N_p)/\mathcal{O}(j^2 + j \log_2 N_p)$	None / $\mathcal{O}(j^2)$	Sec. VIA 1
$fX_j(p)$	$\mathcal{O}(j^2 + j \log_2 N_p)$	$\mathcal{O}(j^2)$	Eq. (114)
$fSp_{(j_b, j_f)}(m, r)$	$\mathcal{O}(j_f^2 + j_f \log_2 N_p)$	$\mathcal{O}(j_f^2)$	Eq. (152)
$\hat{\mathcal{S}}_{j \leftarrow j-1}$	$\mathcal{O}(j^3 \log_2 j + j \log_2 N_p)$	$\mathcal{O}(j^2)$	Sec. VIA 3 and Appendix A 2 b
$bS_j(p, q)/bS_j(p, L)$	$\mathcal{O}(j^3 \log_2 j + j \log_2 N_p)$	None / $\mathcal{O}(j^2)$	Sec. VIA 3 and Appendix A 2
$bX_j(p)$	$\mathcal{O}(j^3 \log_2 j + j \log_2 N_p)$	$\mathcal{O}(j^2)$	Eq. (132)

$$b_i |f_n, \dots, \underbrace{1}_i, \dots, f_1\rangle = (-1)^{\sum_{j=i+1}^n f_j} |f_n, \dots, \underbrace{0}_i, \dots, f_1\rangle, \quad (161)$$

the action of b_i is therefore nonlocal in this basis, since it requires a sign that depends on the occupation state of modes with $j > i$. The Jordan-Wigner mapping identifies the occupation-number basis and that of the direct product of n qubits:

$$\begin{aligned} |f_n, \dots, f_1\rangle &\in H \rightarrow J(|f_n, \dots, f_1\rangle) \\ &= |f_n\rangle \otimes \dots \otimes |f_1\rangle \in (B^{1/2})^{\otimes n}, \end{aligned} \quad (162)$$

so that the creation and annihilation operators are written as

$$b_j^\dagger = Z^{\otimes n-j-1} \otimes \sigma^{-j} \otimes \mathbf{i}^{\otimes j-1}, \quad (163)$$

$$b_j = Z^{\otimes n-j-1} \otimes \sigma^{+j} \otimes \mathbf{i}^{\otimes j-1}, \quad (164)$$

which are also nonlocal. Some improvements are possible in basis such as the one introduced by Bravyi and Kitaev [7], with even-indexed qubits codifying occupation numbers and odd-indexed ones encoding some partial sums $y_i = \sum_{j=i+1}^n f_j$; for more details, see [7,37].

The codification, therefore, uses N qubits to represent N possible single-particle states: The entire Fock space is encoded, from the vacuum to the state obtained from application of all creation operators, i.e., from the 0-particle state to the N -particle state. In contrast, the particle-register codification uses $\mathcal{O}(n \log N)$ qubits, with n the maximum number of particles to be codified and N the number of single-particle states.

It falls off that an implementation based on registers encoding particles is not economical when the number of such particles is high [30]. The necessity of ensuring either Bose-Einstein or Fermi-Dirac statistics imposes tight constraints and renders unphysical most combinations of single-particle states which are encodable, thus increasing the unused fraction of the Hilbert space with the number of particles. For example, using two qubits per particle register to encode momenta allows for up to four different values; composing these registers to generate up to three-particle states will, therefore, involve six qubits in total. Of the 2^6 possible orthogonal states, only the antisymmetric combinations are physical, so we have $\binom{4}{1} = 4$ single-particle states, $\binom{4}{2} = 6$ two-particle states, and $\binom{4}{3} = 4$ three-particle states. A direct Jordan-Wigner transformation or akin may, thus, be preferable on the grounds of memory alone when the number of momenta is very small, so that the logarithmic advantage of the register encoding is not yet manifest: With the same qubits, they are able to represent $\binom{6}{1} = 6$ single-particle states, $\binom{6}{2} = 15$ two-particle states, $\binom{6}{3} = 20$ three-particle states, etc. For larger N but moderate n , the JW encoding is inferior to the register encoding.

D. Final comments

We have proposed a transparent encoding of canonical second quantization of quantum theory for digital quantum computers featuring standard gates.

In traditional field theory, much of the emphasis concentrates on the normal modes a and a^\dagger , and we have striven to provide a straightforward construction thereof in terms of set and scrap \mathfrak{S}^\dagger , \mathfrak{S} quantum-number operators as well as memory-control operators \mathfrak{C}_{ij} that allow us to select read-write memory registers as needed.

Still, in quantum computation, all operations are unitary, so that it is the exponentiation of the creation-destruction algebra which takes the central role and we have provided detailed representations of all needed operations.

Much of the work has been devoted to describing an adequate implementation of the (anti)symmetrization of identical quanta and the bookkeeping necessary to carry out the usual operations (free evolution, particle splitting or annihilation, momentum exchange, etc.). Note that explicit symmetrization is essential for our implementation; as seen in Eqs. (42) and (61), the creation and annihilation operators are written as sums of terms changing only the last register, and for this to work all active modes must “be” somehow on the last register. Of course, this has its drawbacks, and we have struggled in Sec. VI to give an implementation of particle-changing operators while keeping “local” transformations, for the Pauli exclusion principle and the creation criterion in Sec. VI A are conditions over the entire memory.

All standard field-theory requirements such as commutation relations are satisfied up to boundary terms which arise due to the finite computer memory. In future work, we would like to address the necessary (Wilsonian-like) renormalization that arises from varying the memory size. This implies that certain symmetries will be recovered only in the limit of an infinite memory (saliently continuous spacetime symmetries, for which the situation is identical to that in established lattice gauge theory). Internal flavor and color symmetries are exactly implemented in our encoding.

Modern physics circumstances in which having this encoding at hand is promising are many, among them the many-body problem in nuclear physics [40], in hadron physics [41,42], in condensed-matter physics, and others [43].

ACKNOWLEDGMENTS

This work is supported by Spanish MICINN under Grant No. PID2022-137003NB-I00; EU’s 824093 (STRONG2020); Grant FPU21/04180 of the Spanish Ministry of Universities and Universidad Complutense de Madrid under research group 910309 and the IPARCOS institute. The first author also acknowledges the Spanish Ministry of Universities for the mobility grant

“EST24/00358” and thanks the HEP division at Argonne National Laboratory for their hospitality.

APPENDIX A: ALGORITHMS

Generalizations and descriptions of the algorithms mentioned in the main text are collected here for completeness.

1. “Locate the largest” algorithm

We here address the algorithm which locates the largest momentum P encoded in the memory (in the sense of their previously chosen ordering) and brings it to the front for further operation, taking into account that it may be repeated m times if handling bosons.

Given a set of n registers in the state $|p_n, p_2, \dots, p_1\rangle$ with modes on registers k_1, k_2, \dots, k_m all equal to P with the largest index and an auxiliary register with n -qubits initially in the 0-ket, the algorithm performs the transformation

$$\begin{aligned} & |0, \dots, 0\rangle |p_n, \dots, p_1\rangle \\ & \xrightarrow{LL} |0, \dots, \underbrace{1}_{k_m}, \dots, \underbrace{1}_{k_2}, \dots, \underbrace{1}_{k_1}, \dots, 0\rangle \\ & \quad \times |p_n, \dots, \underbrace{P}_{k_m}, \dots, \underbrace{P}_{k_2}, \dots, \underbrace{P}_{k_1}, \dots, p_1\rangle, \end{aligned} \quad (\text{A1})$$

which is obviously reversible. The order relation between states is introduced via an oracle, using the order criteria in Sec. VI A:

$$O(|x\rangle |p_i\rangle |p_j\rangle) = \begin{cases} |x \oplus 1\rangle |p_i\rangle |p_j\rangle & \text{if } i \geq j \\ |x\rangle |p_i\rangle |p_j\rangle & \text{otherwise.} \end{cases} \quad (\text{A2})$$

We now provide a detailed step-by-step description of the algorithm for a three-particle memory with state $|p, q, k\rangle$ and afterward, but only briefly, discuss its generalization to an arbitrary number of registers.

- (1) *Introduction of auxiliary registers.*—Introduce three auxiliary registers: the “mark” register \mathcal{A} , with the same number of qubits as the modes in the initial state; the comparison register B , with one qubit less, all in state 0; and a one qubit-register C :

$$|0\rangle_C |00\rangle_B |000\rangle_{\mathcal{A}} |p, k, q\rangle. \quad (\text{A3})$$

- (2) *Third-position test.*—Test whether the largest mode is on register 3:

- (a) *Comparison of third and second registers.*—Use the oracle with qubit C and the third and second registers of $|p, k, q\rangle$; then, anticontrolling on the state of C , initialize qubit b_0 and, finally, uncompute C . The result is

$$\begin{cases} |0\rangle_C |01\rangle_B |000\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p \geq k \\ |0\rangle_C |00\rangle_B |000\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p < k. \end{cases} \quad (\text{A4})$$

- (b) *Comparison of third and first registers.*—Repeat the process but initialize qubit b_1 instead of b_0 :

$$\begin{cases} |0\rangle_C |11\rangle_B |000\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p \geq k, p \geq q \\ |0\rangle_C |10\rangle_B |000\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p \geq k, p < q \\ |0\rangle_C |01\rangle_B |000\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p < k, p \geq q \\ |0\rangle_C |00\rangle_B |000\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p < k, p < q. \end{cases} \quad (\text{A5})$$

- (c) *End test.*—Use the retrieved information to establish whether the momentum on the third register is one of the largest, which is the case only if the state of B is $|11\rangle_B$. Uncompute B ;

$$\begin{cases} |0\rangle_C |00\rangle_B |100\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p \geq k, p \geq q \\ |0\rangle_C |00\rangle_B |000\rangle_{\mathcal{A}} |p, k, q\rangle & \text{otherwise.} \end{cases} \quad (\text{A6})$$

- (3) *Second-position test.*—Repeat the procedure but now, in the final step, change the second qubit of A instead of the third:

$$\begin{cases} |0\rangle_C |00\rangle_B |100\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p > k, p \geq q \\ |0\rangle_C |00\rangle_B |010\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } k > p, k \geq q \\ |0\rangle_C |00\rangle_B |110\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p = k \geq q \\ |0\rangle_C |00\rangle_B |000\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } q > k, q > p. \end{cases} \quad (\text{A7})$$

- (4) *Third-position test.*—Finally, repeat for the third qubit of A :

$$\begin{cases} |0\rangle_C |00\rangle_B |100\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p > k, p > q \\ |0\rangle_C |00\rangle_B |010\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } k > p, k > q \\ |0\rangle_C |00\rangle_B |001\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } q > k, q > p \\ |0\rangle_C |00\rangle_B |110\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p = k > q \\ |0\rangle_C |00\rangle_B |101\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } p = q > k \\ |0\rangle_C |00\rangle_B |011\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } k = q > p \\ |0\rangle_C |00\rangle_B |111\rangle_{\mathcal{A}} |p, k, q\rangle & \text{if } k = q = p. \end{cases} \quad (\text{A8})$$

The generalization of the algorithm is straightforward—for n initial modes, register A has n qubits, register B $n - 1$ qubits and register C still one, and n position tests are necessary, each one requiring

- (i) $\mathcal{O}(n)$ comparisons $\rightarrow \mathcal{O}(n)$ oracle calls and $\mathcal{O}(n)$ CNOTs;
- (ii) an end test: $\mathcal{O}(n)$ Toffoli gates for a general control $\rightarrow \mathcal{O}(n)$ CNOTs and single-qubit gates; and
- (iii) Total: $\mathcal{O}(n)$ oracle calls + $\mathcal{O}(n)$ CNOTs + $\mathcal{O}(n)$ single-qubit gates.

Thus, the algorithm requires at most $\mathcal{O}(n^2)$ oracle calls, CNOTs, and single-qubit gates. ■

2. Generalization of symmetrizers and boson-SWAPs

This subsection, introducing two useful auxiliary operators, is divided in three parts. We start with a subroutine that is used by both of those operators.

a. Symmetric-digit-decomposition (SDD) algorithm

This is useful in the following two algorithms: for step 5 in the implementation of the generalized symmetrizer presented shortly and for step 4 of the boson-SWAP. Both steps are similar, consisting of the creation of a certain superposition invariant under the interchange of qubits in an n -qubit register C directed by another register B (also of n qubits). Suppose register B is in the state $|m\rangle_B$ and call m_1, m_2, \dots, m_k the binary digits with only one qubit in state $|1\rangle$, so that $m = m_1 + m_2 + \dots + m_k$ is the decomposition of m in powers of 2 (e.g., if $m = 101$, we would have $m_1 = 001$ and $m_2 = 100$); then we seek an algorithm that efficiently performs the transformation

$$|0, \dots, 0\rangle_C |m\rangle_B \rightarrow \frac{1}{\sqrt{k}} (|m_1\rangle_C + |m_2\rangle_C + \dots + |m_k\rangle_C) |m\rangle_B. \quad (\text{A9})$$

If n is a small number, we can simply control on each of the possibilities of m and perform the corresponding transformation, as in Sec. VI A 3, but the possible values of m grow as 2^n with the number of digits n . Instead, we propose the following:

- (1) Add two auxiliary registers D and E with $\log_2 n$ qubits

$$|0\rangle_C |m\rangle_B \rightarrow \underbrace{|0\rangle_E |0\rangle_D}_{\text{auxiliary}} |0\rangle_C |m\rangle_B. \quad (\text{A10})$$

- (2) Initialize D with the number of qubits of B in state 1; i.e., for each qubit of B on state 1, add one to D . The computing costs up to this point become n unit adders over $\log_2 n$ qubits which cost $\mathcal{O}(n \log_2^2 n)$ CNOT and single-qubit gates:

$$|0\rangle_E |0\rangle_D |0\rangle_C |m\rangle_B \rightarrow |0\rangle_E |k\rangle_D |0\rangle_C |m\rangle_B. \quad (\text{A11})$$

- (3) Each qubit b_i of B starting from the right may perform a controlled transformation as follows:

- (a) Controlling on $b_i = 1$, on $E = |l\rangle_E > 0$, and on $D = |k\rangle_D$, apply the following single-qubit rotation on qubit i of register C :

$$|0, \dots, 0\rangle_C \rightarrow \sqrt{\frac{k-l}{k-l+1}} |0, \dots, 0\rangle_C + \frac{1}{\sqrt{k-l+1}} |m_i\rangle. \quad (\text{A12})$$

Costs: Registers E and D have $\mathcal{O}(2 \log_2 n)$ qubits in common, which generates $\mathcal{O}(n^2)$

states. For each of these, we apply a controlled [over $\mathcal{O}(2 \log_2 n)$ qubits] single-qubit rotation, which gives a total cost of $\mathcal{O}(n^2 \log_2 n)$ CNOTs and single-qubit gates.

- (b) Controlled on $b_i = 1$, increase E by one. Costs: $\mathcal{O}(\log_2^2 n)$ CNOTs and single-qubit gates.

In total, step 3 costs $\mathcal{O}(n^3 \log_2 n + n \log_2^2 n)$ CNOT and single-qubit gates and generates the state

$$|0\rangle_E |k\rangle_D |0\rangle_C |m\rangle_B \rightarrow |k\rangle_E |k\rangle_D \frac{1}{\sqrt{k}} (|0\rangle_C + |m_2\rangle_C + \dots + |m_k\rangle_C) |m\rangle_B. \quad (\text{A13})$$

- (4) Using register B , do the transformation $|0, \dots, 0\rangle_C \leftrightarrow |m_1\rangle_C$ using the following procedure:
 - (c) Controlled on $b_1 = 1$ and $c_2 = \dots = c_n = 0$, apply an X gate to c_1 .
 - (d) Controlled on $b_2 = 1$, $b_1 = 0$, and $c_1 = c_3 = \dots = c_n = 0$, apply an X gate to c_2 .
 - (e) Controlled on $b_n = 1$, $b_1 = \dots = b_{n-1} = 0$, and $c_1 = \dots = c_{n-1} = 0$, apply an X gate to c_n .

Step 4. n is the most expensive, requiring a controlled X gate over $2n$ qubits, which can be decomposed in $\mathcal{O}(n^2)$ CNOTs and single-qubit gates. Since there are n steps, this gives a total cost of $\mathcal{O}(n^3)$ CNOTs and single-qubit gates and produces the desired state

$$|k\rangle_E |k\rangle_D \frac{1}{\sqrt{k}} (|0\rangle_C + |m_2\rangle_C + \dots + |m_k\rangle_C) |m\rangle_B \rightarrow |k\rangle_E |k\rangle_D \frac{1}{\sqrt{k}} (|m_1\rangle_C + |m_2\rangle_C + \dots + |m_k\rangle_C) |m\rangle_B. \quad (\text{A14})$$

- (5) Return registers E and D to their initial states. Costs: $\mathcal{O}(\log_2 n)$ CNOT and single-qubit gates to uncompute E from D and $\mathcal{O}(n \log_2^2 n)$ CNOTs and single-qubit gates to uncompute D from B .

The algorithm in total requires $\mathcal{O}(n^3 \log_2 n)$ CNOTs and single-qubit gates. ■

b. Generalization of the symmetrizer $\hat{\mathcal{S}}_{j \leftarrow j-1}^\dagger$

Consider a quantum memory A with j occupied registers, each composed of $n = \log_2 N_p$ qubits, symmetric under the interchange of any two registers from 1 to $j-1$. Consider also that the largest mode sits on register j and that it is repeated k times. The algorithm proceeds as follows:

- (1) Add two auxiliary registers C and B of j qubits. Initialize register C to $|1, 0, \dots, 0\rangle_C$ and register B to $|0, \dots, 0\rangle_B$.
- (2) Apply the LL algorithm to the first j registers of A and store the locations of the largest modes on register B . This entangles B and A . Costs: $\mathcal{O}(j^2)$ oracle calls, CNOTs, and single-qubit gates.

- (3) Symmetrize register C . Costs: $\mathcal{O}(\log_2 j)$ single-qubit gates.
- (4) Permute the j th-mode register of A and the j th qubit of B with that pair of mode register of A and qubit of B marked on register C . This applies the different permutations that define the symmetrizer Eq. (38), so the quantum memory A ends up being symmetric but entangled now also with C . To analyze its costs, note that register C has j qubits, each one used as control for a two-mode register SWAP, which costs $\mathcal{O}(n)$ Toffoli gates, each of which can be decomposed in $\mathcal{O}(n)$ CNOTs and single-qubit gates. This gives a total cost of $\mathcal{O}(jn)$ CNOTs and single-qubit gates. The costs of swapping two qubits on B are negligible compared with the cost of swapping two mode registers.
- (5) We now want to disentangle the auxiliary registers. We first uncompute C . For this, note that step 2 leaves B on a superposition of the form (we omit the corresponding registers of A for conciseness)

$$|1, (0, \dots, 0, \underbrace{1}_{k-1}, \dots, 1)_S\rangle_B, \quad (\text{A15})$$

where the superposition is symmetric under the interchange of the qubits inside the parenthesis. The permutations of step 4 just applies the symmetrizer to this state (and the mode registers), mixing the first 1 in all the remaining positions (and symmetrizing the j th register mode). If there is already a 1 in that position, the permutation leaves the corresponding state on the superposition of B invariant. Consider, for example, $|1, 0, \dots, 0, \underbrace{1}_{k-1}, \dots, 1\rangle_B$; then,

grouping all the states of C that correspond to a permutation which leaves that state invariant, we have

$$\frac{1}{\sqrt{k}} (|1, 0, \dots, 0\rangle_C + |0, \dots, \underbrace{1}_{k-1}, \dots, 0\rangle_C + \dots + |\dots, 0, 1\rangle_C) |1, 0, \dots, 0, \underbrace{1}_{k-1}, \dots, 1\rangle_B. \quad (\text{A16})$$

If, instead, there is a 0 in the position of B to permute, the permutation interchanges that for the 1 of the j th qubit. Since B is symmetric under any interchange of its first $j-1$ qubits, we can arrive to the same final state from a states of B that differ in a permutation; for example, consider

$$\begin{aligned} & |0, \dots, \underbrace{1}_k, \dots, 0\rangle_C |1, \dots, 0, \underbrace{1}_{k-1}, \dots, 1\rangle_B \\ & \rightarrow |0, \dots, \underbrace{1}_k, \dots, 0\rangle_C |0, \dots, 0, \underbrace{1}_k, \dots, 1\rangle_B; \end{aligned} \quad (\text{A17})$$

the same final state can be reached with another state of C from an state of B with some of their final 1's permuted to the k th position, for example,

$$\begin{aligned} & |0, \dots, \underbrace{1}_m, \dots, 0\rangle_C |1, \dots, 0, \underbrace{1}_k, 1, \dots, \underbrace{0}_m, \dots, 1\rangle_B \\ & \rightarrow |0, \dots, \underbrace{1}_m, \dots, 0\rangle_C |0, \dots, 0, \underbrace{1}_k, \dots, 1\rangle_B. \end{aligned} \quad (\text{A18})$$

We can group all this possibilities as before:

$$\begin{aligned} & \frac{1}{\sqrt{k}} (|0, \dots, \underbrace{1}_k, \dots, 0\rangle_C + |0, \dots, \underbrace{1}_{k-1}, \dots, 0\rangle_C \\ & + |0, \dots, \underbrace{1}_{k-2}, \dots, 0\rangle_C + \dots + |\dots, 0, 1\rangle_C) \\ & |0, 0, \dots, 0, \underbrace{1}_k, \dots, 1\rangle_B. \end{aligned} \quad (\text{A19})$$

We see that for all possible interchanges the states of C and B can be grouped together as the registers C and B of the SDD algorithm at the beginning of this section; thus, we can apply it in reverse to arrive to

$$|0\rangle_C |(1, 0, \dots, 0, \underbrace{1}_{k-1}, \dots, 1)_S\rangle_B. \quad (\text{A20})$$

Register B can now be disentangled using the LL algorithm in reverse. Costs: $\mathcal{O}(j^3 \log_2 j)$ CNOTs and single-qubit gates to disentangle C and $\mathcal{O}(j^2)$ order-oracle calls, CNOTs, and single-qubit gates to disentangle B .

The total costs of the algorithm are $\mathcal{O}(j^3 \log_2 j + j \log_2 N_p)$ CNOT and single-qubit gates + $\mathcal{O}(j^2)$ order-oracle calls. ■

c. Generalization of boson-SWAP $bS_j(p, q)$ to allow for repeated momenta

Consider a quantum memory A with j occupied registers, each composed of $n = \log_2 N_p$ qubits, symmetric under the interchange of any two registers 1 to $j-1$. To be specific, mode p sits at register j and it is repeated k times, while mode q is repeated l times (this situation does not arise for fermions):

$$\mathcal{S}_{j-1 \rightarrow 1} [|p\rangle_j \dots |p\rangle_{l+k-1} \dots |p\rangle_{l+1} |q\rangle_l \dots |q\rangle_1]_A. \quad (\text{A21})$$

We proceed as follows:

- (1) Add three auxiliary registers with $j-1$ qubits named B , C , and D . Initialize B with the positions of mode q and C with the positions of mode p ; if p is the largest mode, the LL algorithm is applied instead. Both registers become entangled with A

and among themselves. Without explicitly writing register A , B and C are combined in the state

$$\mathcal{S}_{j-1 \leftarrow 1} [|00\rangle_{j-1} \dots |00\rangle_{l+k} |10\rangle_{l+k-1} \dots |10\rangle_{l+1} |01\rangle_l \dots |01\rangle_1], \quad (\text{A22})$$

where the rightmost qubit corresponds to register B and the left one to register C . So far, the starting costs are $\mathcal{O}(n)$ CNOTs and single-qubit gates for a controlled operation over n qubits. This is repeated j times, so $\mathcal{O}(jn)$ CNOTs and single-qubit gates and $\mathcal{O}(j^2)$ order-oracle calls, CNOT, and single-qubit gates should the LL algorithm be applied.

- (2) Register B and the SDD algorithm from the first paragraph of this section are used to initialize register D :

$$\begin{aligned} & \mathcal{S}_{j-1 \leftarrow 1} \frac{1}{\sqrt{l}} (|0, \dots, \underbrace{1}_l, \dots, 0 \rangle_D + \dots \\ & + |0, \dots, 1, 0\rangle_D + |0, \dots, 1\rangle_D) \\ & \times [|00\rangle_{j-1} \dots |00\rangle_{l+k} |10\rangle_{l+k-1} \dots \\ & \times |10\rangle_{l+1} |01\rangle_l \dots |01\rangle_1]. \end{aligned} \quad (\text{A23})$$

The accrued costs are $\mathcal{O}(j^3 \log_2 j)$ CNOTs and single-qubit gates.

- (3) Register D now marks the positions of the modes to be interchanged; if $d_i = 1$ (so $b_i = 1$), mode q sits on register i . Thus, registers i and j should be interchanged. The qubits b_i and c_i are also interchanged, since now mode p , instead of q , occupies register i :

$$\begin{aligned} & \mathcal{S}_{j-1 \leftarrow 1} \frac{1}{\sqrt{l}} \{ |0, \dots, 1\rangle_D [|00\rangle_{j-1} \dots |00\rangle_{l+k} |10\rangle_{l+k-1} \dots |10\rangle_{l+1} |01\rangle_l \dots |01\rangle_2 |10\rangle_1] \\ & + |0, \dots, 1, 0\rangle_D [|00\rangle_{j-1} \dots |00\rangle_{l+k} |10\rangle_{l+k-1} \dots |10\rangle_{l+1} |01\rangle_l \dots |10\rangle_2 |01\rangle_1] + \dots \\ & + |0, \dots, \underbrace{1}_l, \dots, 0\rangle_D [|00\rangle_{j-1} \dots |00\rangle_{l+k} |10\rangle_{l+k-1} \dots |10\rangle_{l+1} |10\rangle_l \dots |01\rangle_2 |01\rangle_1] \}; \end{aligned} \quad (\text{A24})$$

By grouping all possible arrangements that lead to the same final states of registers B and C , the equation can be rewritten. For instance, consider the first line in the equation above and focus on the state where registers 1 and $l+m$ ($0 < m < k$) are interchanged. Step 3 then produces the following:

$$\begin{aligned} & |0, \dots, \underbrace{1}_{l+m}, \dots, 0\rangle_D [|00\rangle_{j-1} \dots |00\rangle_{l+k} |10\rangle_{l+k-1} \dots |01\rangle_{l+m} \dots |10\rangle_{l+1} |01\rangle_l \dots |01\rangle_2 |10\rangle_1] \\ & \rightarrow |0, \dots, \underbrace{1}_{l+m}, \dots, 0\rangle_D [|00\rangle_{j-1} \dots |00\rangle_{l+k} |10\rangle_{l+k-1} \dots |10\rangle_{l+m} \dots |10\rangle_{l+1} |01\rangle_l \dots |01\rangle_2 |10\rangle_1], \end{aligned} \quad (\text{A25})$$

the same as the state of the first line above. Grouping all the interchanges that produce the same states, we have

$$\begin{aligned} & \mathcal{S}_{j-1 \leftarrow 1} \frac{1}{\sqrt{k}} (|0, \dots, \underbrace{1}_{l+k-1}, \dots, 0 \rangle_D + \dots + |0, \dots, \underbrace{1}_{l+1}, \dots, 0 \rangle_D + |0, \dots, 1\rangle_D) \\ & \times [|00\rangle_{j-1} \dots |00\rangle_{l+k} |10\rangle_{l+k-1} \dots |10\rangle_{l+m} \dots |10\rangle_{l+1} |01\rangle_l \dots |01\rangle_2 |10\rangle_1], \end{aligned} \quad (\text{A26})$$

which precludes the next step. Costs: Register D has $j-1$ qubits; each one is used as control for a two-mode register SWAP, which costs $\mathcal{O}(n)$ Toffoli gates, each of which can be decomposed in $\mathcal{O}(n)$ CNOTs and single-qubit gates. This gives a total cost of $\mathcal{O}(jn)$ CNOTs and single-qubit gates. The costs of swapping two qubits of B and C are negligible compared with the cost of swapping two mode registers.

- (4) Disentangle register D with the SDD algorithm using register C . Costs: $\mathcal{O}(j^3 \log_2 j)$ CNOTs and single-qubit gates.

- (5) Disentangle registers C and B applying again step 1. Costs: $\mathcal{O}(jn)$ CNOTs and single-qubit gates, $\mathcal{O}(j^2)$ order-oracle calls, CNOTs, and single-qubit gates if the LL algorithm is applied.

The total cost of the algorithm is $\mathcal{O}(j^3 \log_2 j + j \log_2 N_p)$ CNOT and single-qubit gates + $\mathcal{O}(j^2)$ order-oracle calls, CNOT and single-qubit gates, to advance the system forward in time by one step.

Finally, the hypothesis of having mode p on register j can be relaxed by adding one extra qubit to registers B and C and controlling the transformations of D in steps 2. and 4.

on that register having its j th qubit on state 1. Which only adds a control over a single qubit to the SDD algorithm, and therefore a constant number of gates that do not alter the asymptotic costs. ■

APPENDIX B: PROOF OF ANTICOMMUTATION RELATIONS

This appendix discusses the demonstration of equation Eq. (68); the demonstration of Eq. (54) is similar and only briefly mentioned.

Proof. The demonstration is based on induction over n (the maximum number of particles). Take the creation and annihilation operators over n registers, $b_q^{(n)}$ and $b_q^{(n)\dagger}$,

respectively, and assume that the anticommutation relation holds:

$$\begin{aligned} \{b_q^{(n)}, b_p^{(n)\dagger}\} \\ = \delta_{qp} (\mathfrak{C}_{00} \otimes \mathbf{i})_n \otimes \mathbb{I}^{(n-1)} \\ + \mathcal{A}_{n \leftarrow n-1} \cdot (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_p)_n \otimes \mathbb{P}_{n-1}^{(n-1)} \cdot \mathcal{A}_{n \leftarrow n-1}. \end{aligned} \quad (\text{B1})$$

This is now used to prove the $(n+1)$ th case (for bosons, there is a negative sign in the second term and symmetrizers instead of antisymmetrizers). Operators over $n+1$ particles can be written in terms of those for n particles; first, by Eq. (61),

$$\begin{aligned} b_q^{(n+1)\dagger} &= \sum_{i=1}^{n+1} \mathbb{I}^{\otimes(n+1-i)} \otimes \mathcal{A}_{i \leftarrow i-1} \cdot \mathbb{P}_0^{(n+1-i)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_i \otimes \mathbb{P}_{i-1}^{(i-1)} \\ &= + \sum_{i=1}^n \mathbb{I} \otimes (\mathbb{I}^{\otimes(n-i)} \otimes \mathcal{A}_{i \leftarrow i-1}) \cdot \mathbb{P}_0^{(1)} \otimes (\mathbb{P}_0^{(n-i)} \otimes (\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_i \otimes \mathbb{P}_{i-1}^{(i-1)}) + \mathcal{A}_{n+1 \leftarrow n} \cdot (\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_{n+1} \otimes \mathbb{P}_n^{(n)} \\ &\equiv [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_q^{(n)\dagger}] + \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_{n+1} \otimes] [\mathbb{P}_n^{(n)}], \end{aligned} \quad (\text{B2})$$

and similarly

$$b_p^{(n+1)} = [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_p^{(n)}] + [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n}; \quad (\text{B3})$$

the corresponding expressions for bosons are completely analogous. We now calculate $\{b_p^{(n+1)}, b_q^{(n+1)\dagger}\}$:

$$\begin{aligned} b_q^{(n+1)\dagger} b_p^{(n+1)} &= \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} \\ &+ \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_p^{(n)}] \\ &+ [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_q^{(n)\dagger}] [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} \\ &+ [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_q^{(n)\dagger}] [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_p^{(n)}] \\ &= \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} + [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_q^{(n)\dagger} b_p^{(n)}], \end{aligned} \quad (\text{B4})$$

where the second and third terms vanish because, on the one hand, $\mathbb{P}_n^{(n)} b_q^{(n)}$ represents the action of $\mathbb{P}_n^{(n)}$ over a state that does not have n particles because of the annihilation operator, which gives 0. On the other hand, $b_p^{(n)\dagger} \mathbb{P}_n^{(n)}$ represents the action of $b_p^{(n)\dagger}$ over a state with exactly n particles, which is again 0. The other product reads

$$\begin{aligned} b_p^{(n+1)} b_q^{(n+1)\dagger} &= [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n}^2 \cdot [(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger) \otimes] [\mathbb{P}_n^{(n)}] \\ &+ [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_p^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger) \otimes] [\mathbb{P}_n^{(n)}] \\ &+ [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_q^{(n)\dagger}] \\ &+ [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_p^{(n)}] [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_q^{(n)\dagger}] \\ &= \sqrt{n+1} [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \\ &+ [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] [b_p^{(n)} b_q^{(n)\dagger}]. \end{aligned} \quad (\text{B5})$$

In this case, the products of the second and third lines also give zero: on one hand, in the second line we see that the left parenthesis will be 0 if the $n + 1$ register is occupied, while the operators in the parenthesis to the right give an occupied $n + 1$ register, and their multiplication gives 0. On the other hand, the left parentheses in the third line will be 0 if any of the $n + 1$ registers are empty, but looking at the parenthesis to the right we see that this is always the case, so multiplication of both parentheses gives again 0; finally, $\mathcal{A}_{n+1 \leftarrow n}^2 = \sqrt{n+1} \mathcal{A}_{n+1 \leftarrow n}$. The anticommutator, therefore, reads

$$\{b_p^{(n+1)}, b_q^{(n+1)\dagger}\} = [(\mathbb{I} \otimes) \cdot (\mathbb{P}_0^{(1)} \otimes)] \{b_p^{(n)} b_q^{(n)\dagger}\} + \sqrt{n+1} [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] + \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} \quad (\text{B6})$$

Using the induction hypothesis on the first line,

$$\{b_p^{(n+1)}, b_q^{(n+1)\dagger}\} = \delta_{pq} \mathbb{I}^{\otimes n+1} \cdot \mathbb{P}_0^{(2)} \otimes \mathbb{I}^{\otimes (n-1)} + \mathbb{I} \otimes \mathcal{A}_{n \leftarrow n-1} \cdot \mathbb{P}_0^{(1)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_p)_n \otimes \mathbb{P}_{n-1}^{(n-1)} \cdot \mathbb{I} \otimes \mathcal{A}_{n \leftarrow n-1} + \sqrt{n+1} [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] + \mathcal{A}_{n+1 \leftarrow n} \cdot (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_p)_{n+1} \otimes \mathbb{P}_n^{(n)} \cdot \mathcal{A}_{n+1 \leftarrow n}, \quad (\text{B7})$$

where the last line is the boundary term in Eq. (68). For the commutator, the result is similar but with negative signs in the second and fourth lines. The first line can then be rewritten as

$$\delta_{pq} \mathbb{I}^{\otimes n+1} \cdot \mathbb{P}_0^{(2)} \otimes \mathbb{I}^{\otimes (n-1)} = \delta_{q_1 q_2} \mathbb{P}_0^{(2)} \otimes \sum_{j=0}^{n-1} \mathbb{P}_k^{(n-1)} = \delta_{q_1 q_2} (\mathfrak{C}_{00} \otimes \mathbf{i})_{n+1} \otimes \sum_{k=0}^{n-1} \mathbb{P}_k^{(n)}. \quad (\text{B8})$$

To complete the induction step, we need the $k = n$ term of the sum, i.e., $\delta_{q_1 q_2} (\mathfrak{C}_{00} \otimes \mathbf{i})_{n+1} \otimes \mathbb{P}_n^{(n)}$. Applying it to a general n -particle state (the only ones for which the result is not null) gives

$$\delta_{q_1 q_2} (\mathfrak{C}_{00} \otimes \mathbf{i})_{n+1} \otimes \mathbb{P}_n^{(n)} |\Omega\rangle_{n+1} (|1p_n\rangle_n \dots |1p_1\rangle_1)_A = \delta_{q_1 q_2} |\Omega\rangle_{n+1} (|1p_n\rangle_n \dots |1p_1\rangle_1)_A. \quad (\text{B9})$$

We now prove that the action of the second and third lines in Eq. (B7) is the same upon arbitrary states (we concentrate in the case with n filled registers; for the rest, the result is null). Starting from the second line (identity operators multiplying antisymmetrizers are omitted)

$$\begin{aligned} & \mathcal{A}_{n \leftarrow n-1} \cdot \mathbb{P}_0^{(1)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_p)_n \otimes \mathbb{P}_{n-1}^{(n-1)} \cdot \mathcal{A}_{n \leftarrow n-1} |\Omega\rangle_{n+1} (|1p_n\rangle_n \dots |1p_1\rangle_1)_A \\ &= \sqrt{n} \mathcal{A}_{n \leftarrow n-1} \cdot (\mathfrak{C}_{00} \otimes \mathbf{i})_{n+1} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_p)_n \otimes \mathbb{P}_{n-1}^{(n-1)} |\Omega\rangle_{n+1} (|1p_n\rangle_n \dots |1p_1\rangle_1)_A \\ &= \mathcal{A}_{n \leftarrow n-1} \sum_{k=0}^{n-1} (-1)^k \delta_{pp_{n-k}} |\Omega\rangle_{n+1} |1q\rangle_n (|p_n\rangle_{n-1} \dots |p_{n-k+1}\rangle_{n-k} |p_{n-k-1}\rangle_{n-k-1} \dots |1p_1\rangle_1)_A \\ &= \sum_{k=0}^{n-1} (-1)^k \delta_{q_2 p_{n-k}} |\Omega\rangle_{n+1} (|1q_1\rangle_n |p_n\rangle_{n-1} \dots |p_{n-k+1}\rangle_{n-k} |p_{n-k-1}\rangle_{n-k-1} \dots |1p_1\rangle_1)_A, \end{aligned} \quad (\text{B10})$$

we continue with the other term of the right-hand side

$$\begin{aligned} & \sqrt{n+1} [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] |\Omega\rangle_{n+1} (|1p_n\rangle_n \dots |1p_1\rangle_1)_A \\ &= \sqrt{n+1} [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} |1q\rangle_{n+1} (|1p_n\rangle_n \dots |1p_1\rangle_1)_A \\ &= \sqrt{n+1} [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes] [\mathbb{P}_n^{(n)}] (|1q\rangle_{n+1} |1p_n\rangle_n \dots |1p_1\rangle_1)_A \\ &= \delta_{pq} |\Omega\rangle_{n+1} (|1p_n\rangle_n \dots |1p_1\rangle_1)_A + \sum_{k=1}^n (-1)^k \delta_{pp_{n+1-k}} |\Omega\rangle_{n+1} (|1q\rangle_n |p_n\rangle_{n-1} \dots |p_{n-k+2}\rangle_{n+1-k} |p_{n-k}\rangle_{n-k} \dots |1p_1\rangle_1)_A, \end{aligned} \quad (\text{B11})$$

changing $k \rightarrow k + 1$ in the last line:

$$\begin{aligned}
& \sqrt{n+1}(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes \mathbb{P}_n^{(n)} \cdot \mathcal{A}_{n+1 \leftarrow n} \cdot (\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_{n+1} \otimes \mathbb{P}_n^{(n)} |\Omega\rangle_{n+1} (|1p_n\rangle_n \dots |1p_1\rangle_1)_A \\
& = \delta_{pq} |\Omega\rangle_{n+1} (|p_n\rangle_n \dots |1p_1\rangle_1)_A - \sum_{k=0}^{n-1} (-1)^k \delta_{pp_{n-k}} |\Omega\rangle_{n+1} (|1q\rangle_n |p_n\rangle_{n-1} \dots |p_{n-k+1}\rangle_{n-k} |p_{n-k-1}\rangle_{n-k-1} \dots |1p_1\rangle_1)_A, \quad (\text{B12})
\end{aligned}$$

and adding both terms:

$$\begin{aligned}
& (\mathcal{A}_{n \leftarrow n-1} \cdot \mathbb{P}_0^{(1)} \otimes (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_p)_n \otimes \mathbb{P}_{n-1}^{(n-1)} \cdot \mathcal{A}_{n \leftarrow n-1} \\
& \quad + \sqrt{n+1} [(\mathfrak{C}_{01} \otimes \mathfrak{s}_p)_{n+1} \otimes [\mathbb{P}_n^{(n)}] \cdot \mathcal{A}_{n+1 \leftarrow n} \cdot [(\mathfrak{C}_{10} \otimes \mathfrak{s}_q^\dagger)_{n+1} \otimes [\mathbb{P}_n^{(n)}] |\Omega\rangle_{n+1}] (|1p_n\rangle_n \dots |1p_1\rangle_1)_A \\
& = \delta_{pq} |\Omega\rangle_{n+1} (|p_n\rangle_n \dots |1p_1\rangle_1)_A. \quad (\text{B13})
\end{aligned}$$

We, thus, have

$$\begin{aligned}
\{b_p^{(n+1)}, b_q^{(n+1)\dagger}\} & = \delta_{qp} (\mathfrak{C}_{00} \otimes \mathbf{i})_{n+1} \otimes \sum_{k=0}^{n-1} \mathbb{P}_k^{(n)} + \delta_{q_1 q_2} (\mathfrak{C}_{00} \otimes \mathbf{i})_{n+1} \otimes \mathbb{P}_n^{(n)} + \mathcal{A}_{n+1 \leftarrow n} \cdot (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_p)_{n+1} \otimes \mathbb{P}_n^{(n)} \cdot \mathcal{A}_{n+1 \leftarrow n} \\
& = \delta_{qp} (\mathfrak{C}_{00} \otimes \mathbf{i})_{n+1} \otimes \sum_{k=0}^n \mathbb{P}_k^{(n)} + \mathcal{A}_{n+1 \leftarrow n} \cdot (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_p)_{n+1} \otimes \mathbb{P}_n^{(n)} \cdot \mathcal{A}_{n+1 \leftarrow n} \\
& = \delta_{qp} (\mathfrak{C}_{00} \otimes \mathbf{i})_{n+1} \otimes \mathbb{I}^{(n)} + \mathcal{A}_{n+1 \leftarrow n} \cdot (\mathfrak{C}_{11} \otimes \mathfrak{s}_q^\dagger \mathfrak{s}_p)_{n+1} \otimes \mathbb{P}_n^{(n)} \cdot \mathcal{A}_{n+1 \leftarrow n}, \quad (\text{B14})
\end{aligned}$$

which concludes the induction step. \blacksquare

APPENDIX C: GLOSSARY OF SYMBOLS

Given the extensive number of mathematical symbols employed through the article, we believe the reader might find useful to have this reference at hand when looking for a particular one. They are given in Latin alphabetic order, independently of the font in which they are encoded.

1. General

Symbol	Meaning	Defining equation
\mathfrak{C}_{ij}	Control operator $ i\rangle\langle j $ with $i, j \in \{0, 1\}$ standing for absence or presence	Eq. (5)
Δ	Momentum-grid spacing	Under Eq. (80)
Δt	Time interval	
E_q	Energy of a particle with momentum q	
\mathbf{i}	Identity over one-particle qubits encoding momentum	Eq. (17)
\mathbb{I}	Identity over entire one-particle registers	Eq. (18)
$\mathbb{I}^{(n)}$	Identity over n ordered registers	
$\mathfrak{H}_{22,l,k}$	Auxiliary potential exchanging momentum over registers l and k	Eq. (88)
η_σ	Parity of permutation σ ; 0 if σ is even and 1 if σ is odd	
j, k, l, m	Indices over the register number	
λ_ξ	Coupling intensity of terms exchanging momentum p_ξ	
$\Lambda_{\max}, \Lambda_{\min}$	Extreme values of the integer indexing the momentum variable	Eq. (80)
\hat{N}	Number operator	Eq. (50)
N_p	Number of encoded momenta	Eq. (80)
$ \Omega\rangle$	Empty registers, in the state $ 0\rangle_{P/A} \otimes 0\dots 0\rangle_{\text{momentum}}$	Eq. (1)
$ 0\rangle_{P/A}$	Presence or absence qubit, 0 indicating absence	Eq. (1)
$ 0\dots 0\rangle_{\text{momentum}}$	State to which empty momentum registers are assigned	Eq. (1)
$ p_i\rangle, q_i\rangle$	States representing p_i and q_i momenta, respectively	Eq. (4)
$ p_{\min}\rangle, p_{mn}\rangle$	Lowest codified momenta	Eq. (80)
$ p_{\max}\rangle, p_{mx}\rangle$	Largest codified momenta	Eq. (80)

(Continued)

(Continued)

Symbol	Meaning	Defining equation
$\overline{ p_i\rangle}_l$	Denotes that momentum on register l is fixed to p_i	Eq. (110)
\mathcal{P}_{ij}	Permutation operator of registers i and j	Eq. (19)
$\mathbb{P}_j^{(n)}$	Projector over exactly j -occupied registers	Eq. (40)
$\mathfrak{S}_p, \mathfrak{S}_p^\dagger$	Scrap and set operators reassigning momentum qubits from $ p\rangle$ to $ 0\dots 0\rangle$ and vice versa	Eq. (4)
$\mathcal{U}_{ij}^{f/b}(t)$	Evolution operator for fermion and boson quanta, it creates and annihilates i/j particles and vice versa	Eqs. (79), (81), (101), and (133)
$\mathfrak{U}_{ij,l}(t)$	Part of the operator $\mathcal{U}_{ij}^{f/b}(t)$ that performs memory changes l designates registers; it can be a number or a tuple	Eqs. (78), (96), (119), (142), and (146)
ξ	Momentum discretization	
\otimes	Tensor product splitting an operator within a register into components over its subspaces	
\cdot (dot)	Successive application of operators over the entire memory	

2. Boson symbols

Symbol	Meaning	Definition
$a_{p,j}^{(n)}, a_{p,j}^{(n)\dagger}$	Destructor and creator operators for a boson of momentum p at the highest occupied register, j	Eqs. (39) and (45)
$a_p^{(n)}, a_p^{(n)\dagger}$	Boson destructor and creator operators with momentum p , at any register (the sum over j of $a_{p,j}^{(n)}, a_{p,j}^{(n)\dagger}$: They represent the field-theoretical a, a^\dagger)	Eqs. (42) and (47)
$\text{bS}_j(p, q)$	Interchanges modes p and q in a symmetric memory with j occupied registers, the first mode (p) should sit on register j	Sec. VI A 3 and Appendix A 2
$\text{bX}_j(p)$	Auxiliary exchange operator for creation and annihilation of bosons	Eq. (132)
$\mathcal{S}_{n\leftarrow m}$	Nonunitary step symmetrizer; it outputs an n -particle symmetric state from <i>any</i> m -particle symmetric one	Eq. (38)
$\hat{\mathcal{S}}_{n\leftarrow m}^\dagger$	Unitary step symmetrizer; it outputs an n -particle symmetric state from a <i>semiordered</i> m -particle symmetric one	Secs. VI A 3 and Appendix A 2 b

3. Fermion symbols

Symbol	Meaning	Definition
$\mathcal{A}_{n\leftarrow m}$	Nonunitary step antisymmetrizer; it outputs an n -particle antisymmetric state from <i>any</i> m -particle antisymmetric one	Eq. (55)
$\hat{\mathcal{A}}_{n\leftarrow m}^\dagger$	Unitary step antisymmetrizer; outputs an n -particle antisymmetric state from an <i>ordered</i> m -particle antisymmetric one	Sec. VI A 1
$b_{p,j}^{(n)}, b_{p,j}^{(n)\dagger}$	Destructor and creator operators for a fermion of momentum p at the highest occupied register, j	Eqs. (59) and (64)
$b_p^{(n)}, b_p^{(n)\dagger}$	Destructor and creator operators for a fermion of momentum p at any register (the sum over j of $b_{p,j}^{(n)}, b_{p,j}^{(n)\dagger}$)	Eqs. (61) and (66)
$\text{fS}_j(p, q)$	Interchanges modes p and q in an antisymmetric memory with j occupied registers; the first mode (p) should sit on register j	Sec. VI A 1
$\text{fS}_{(j_b, j_f)}(p, q)$	Controlled fermion SWAP depending on whether split boson is in register j	Eq. (152)
$\text{fX}_j(p)$	Auxiliary exchange operator for creation and annihilation of fermions	Eq. (114)

- [1] D. Beck, J. Carlson, Z. Davoudi *et al.*, [arXiv:2303.00113](#).
- [2] B. P. Lanyon, J. D. Whitfield, G. G. Gillett *et al.*, *Nat. Chem.* **2**, 106 (2010).
- [3] P. Wang, W. Du, W. Zuo, and J. P. Vary, *Phys. Rev. C* **109**, 064623 (2024).
- [4] M. Drissi, J. W. T. Keeble, J. R. Sarmiento, and A. Rios, *Phil. Trans. R. Soc. A* **382**, 20240057 (2024).
- [5] D. S. Abrams and S. Lloyd, *Phys. Rev. Lett.* **79**, 2586 (1997).
- [6] G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, *Phys. Rev. A* **64**, 022319 (2001).
- [7] S. B. Bravyi and A. Y. Kitaev, *Ann. Phys. (Amsterdam)* **298**, 210 (2002).
- [8] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, *Mol. Phys.* **109**, 735 (2011).
- [9] B. Toloui and P. J. Love, [arXiv:1312.2579](#).
- [10] O. Di Matteo, A. McCoy, P. Gysbers, A. McCoy, Peter Gysbers, T. Miyagi, R. M. Woloshyn, and P. Navrátil, *Phys. Rev. A* **103**, 042405 (2021).
- [11] W. Kirby, B. Fuller, C. Hadfield, and A. Mezzacapo, *PRX Quantum* **3**, 020351 (2022).
- [12] Y. Shee, P.-K. Tsai, C.-L. Hong, H.-C. Cheng, and H.-S. Goan, *Phys. Rev. Res.* **4**, 023154 (2022).
- [13] B. Huang, N. Sheng, M. Govoni, and G. Galli, *J. Chem. Theory Comput.* **19**, 1487 (2023).
- [14] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, *Rev. Mod. Phys.* **92**, 015003 (2020).
- [15] R. Babbush, D. W. Berry, I. D. Kivlichan, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik, *New J. Phys.* **18**, 033032 (2016).
- [16] S. Bravyi, J. M. Gambetta, A. Mezzacapo, and K. Temme, [arXiv:1701.08213](#).
- [17] D. W. Berry, M. Kieferová, A. Scherer, Y. R. Sanders, G. H. Low, N. Wiebe, C. Gidney, and R. Babbush, *npj Quantum Inf.* **4**, 22 (2018).
- [18] R. Babbush, D. W. Berry, J. R. McClean, and H. Neven, *npj Quantum Inf.* **5**, 92 (2019).
- [19] Y. Su, D. W. Berry, N. Wiebe, N. Rubin, and R. Babbush, *PRX Quantum* **2**, 040332 (2021).
- [20] J. Kogut and L. Susskind, *Phys. Rev. D* **11**, 395 (1975).
- [21] S. P. Jordan, K. S. M. Lee, and J. Preskill, *Science* **336**, 1130 (2012).
- [22] E. Zohar, J. I. Cirac, and B. Reznik, *Phys. Rev. A* **88**, 023617 (2013).
- [23] A. Macridin, P. Spentzouris, J. Amundson, and R. Harnik, *Phys. Rev. A* **98**, 042312 (2018).
- [24] R. C. Farrell, I. A. Chernyshev, S. J. M. Powell, N. A. Zemlevskiy, M. Illa, and M. J. Savage, *Phys. Rev. D* **107**, 054512 (2023).
- [25] T. F. Stetina, A. Ciavarella, X. Li, and N. Wiebe, *Quantum* **6**, 622 (2022).
- [26] A. Macridin, A. C. Y. Li, S. Mrenna, and P. Spentzouris, *Phys. Rev. A* **105**, 052405 (2022).
- [27] R. C. Farrell, I. A. Chernyshev, S. J. Powell, N. A. Zemlevskiy, M. Illa, and M. J. Savage, *Phys. Rev. D* **107**, 054513 (2023).
- [28] R. C. Farrell, M. Illa, A. N. Ciavarella, and M. J. Savage, *PRX Quantum* **5**, 020315 (2024).
- [29] S. Weinberg, *The Quantum Theory of Fields. I: Foundations* (Cambridge University Press, Cambridge, England, 2014).
- [30] J. Barata, N. Mueller, A. Tarasov, and R. Venugopalan, *Phys. Rev. A* **103**, 042410 (2021).
- [31] W. M. Kirby, S. Hadi, M. Kreshchuk, and P. J. Love, *Phys. Rev. A* **104**, 042607 (2021).
- [32] W. Qian, R. Basili, S. Pal, G. Luecke, and J. P. Vary, *Phys. Rev. Res.* **4**, 043193 (2022).
- [33] X. Yao, [arXiv:2205.07902](#).
- [34] M. Gómez Rocha, F. J. Llanes-Estrada, D. Schütte, and S. Villalba-Chávez, *Eur. Phys. J. A* **44**, 411 (2010).
- [35] H. Chawdhry and M. Pellen, *SciPost Phys.* **15**, 205 (2023).
- [36] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 1st ed. (Cambridge University Press, Cambridge, England, 2012), 10.1017/CBO9780511976667.
- [37] J. T. Seeley, M. J. Richard, and P. J. Love, *J. Chem. Phys.* **137**, 224109 (2012).
- [38] D. Gallimore and J. Liao, *Phys. Rev. D* **107**, 074012 (2023).
- [39] N. M. de Arenaza, J. J. Gálvez-Viruet, and F. J. Llanes-Estrada, *Eur. Phys. J. A* **60**, 216 (2024).
- [40] P. Ring and P. Schuck, *The Nuclear Many Body Problem*, 1st ed., Texts and Monographs in Physics (Springer, Berlin Heidelberg, 2004).
- [41] S. Cotanch and F. Llanes-Estrada, *Nucl. Phys. A* **689**, 481 (2001).
- [42] F. J. Llanes-Estrada and S. R. Cotanch, *Phys. Rev. Lett.* **84**, 1102 (2000).
- [43] A. L. Fetter and J. D. Walecka, *Quantum Theory of Many-Particle Systems*, Corr. repr ed. (Dover Publication, Mineola, NY, 2003).