Quantum Phase Transitions in the 1D Bose-Hubbard Model with DMRG

Kirill Khoruzhii
Technische Universität München, Max-Planck-Institut für Quantenoptik
(Dated: July 26, 2024)

This study investigates quantum phase transitions in the 1D Bose-Hubbard. Noted for its Mott insulator (MI) and superfluid (SF) phases, the model is analyzed using Density Matrix Renormalization Group (DMRG) methods. The correlation length ξ and average site occupancy $\langle n_j \rangle$ are examined. Our findings indicate that in the MI phase, particle mobility is suppressed and $\Gamma(r)$ decays exponentially, whereas in the SF phase, particles are delocalized, leading to polynomial decay. We also explore the impact of bond dimension D in DMRG, noting that small D yields mean-field-like solutions.

The investigation of quantum phase transitions in strongly correlated systems is an important and fascinating area of research. The one-dimensional Bose-Hubbard model is a minimal bosonic many-particle model that captures the essential physics of interacting bosons on a lattice [1]

$$H = -t \sum_j (a_j^\dagger a_{j+1} + \text{h.c.}) + \tfrac{1}{2} U \sum_j n_j(n_j - 1) - \mu \sum_j n_j$$

and garnered significant attention due to the existence of both the Mott insulator (MI) and superfluid (SF) phases, as well as the successful application of DMRG [2].

The MI phase is characterized by its lack of particle mobility due to strong interactions, resulting in a fixed integer number of particles per site and an energy gap for particle-hole excitations. This phase exhibits suppressed number fluctuations and localized particles, leading to short correlation lengths and an insulating behavior. In the MI phase, the correlation function $\Gamma(r) = \langle a_j^\dagger a_{j+r} \rangle$ decays *exponentially* with distance (fig. 2.1).

On the other hand, the SF phase features delocalized particles that can move freely across the lattice, leading to long-range phase coherence and the absence of an energy gap. This phase is marked by large number fluctuations and a diverging correlation length, indicating a coherent, macroscopic quantum state. In the SF phase, the correlation function $\Gamma(r)$ decays *polynomially* with distance (fig. 2.3).

I. METHODOLOGY

In this article, we will focus on three key quantities to distinguish between these phases: the correlation length

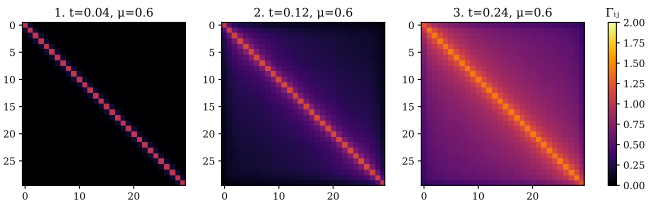

Figure 1. The characteristic form of Γ_{ij} at three points in the phase space: MI (1), critical point (2), and SF (3). The same points are used for illustration purposes throughout the text.

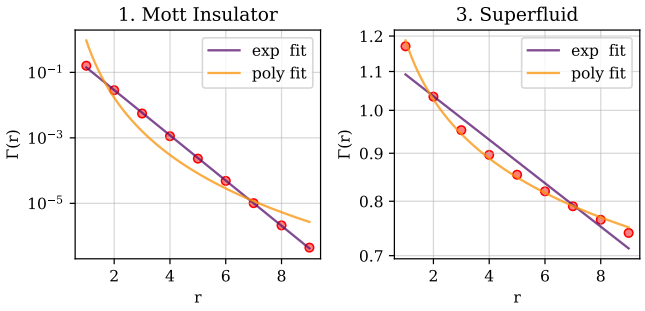

Figure 2. Demonstration of the exponential decay of $\Gamma(r)$ for MI (1) and polynomial decay for SF (3).

ξ, the average site occupancy $\langle n_j \rangle$, and the polynomial degree K with which Γ^2 decays, following the approach in [1]. These properties provide valuable insights into the nature of the MI and SF phases. By examining these parameters, we aim to deepen our understanding of the phase transitions within the one-dimensional Bose-Hubbard model.

We use finite-size DMRG [3, 4], with the primary results presented for a lattice with $L = 30$ sites, a cutoff $n_c = 5$, open boundary conditions, and a bond dimension $D = 300$ (unless otherwise specified). Fixing $U = 1$, we examine the system's behavior in the (t, μ)-plane.

Through ξ and K, we can obtain information about particle mobility and correlations. A correlation length larger ξ signifies longer-range correlations, indicating the

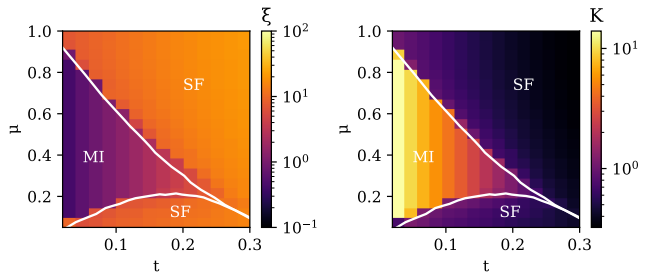

Figure 3. Phases of the 1D Bose-Hubbard model. Phase boundaries are drawn according to the results from [1]. Near the boundaries, the divergence of ξ and the approach of K to the critical value $K_c = 1/2$ can be observed.

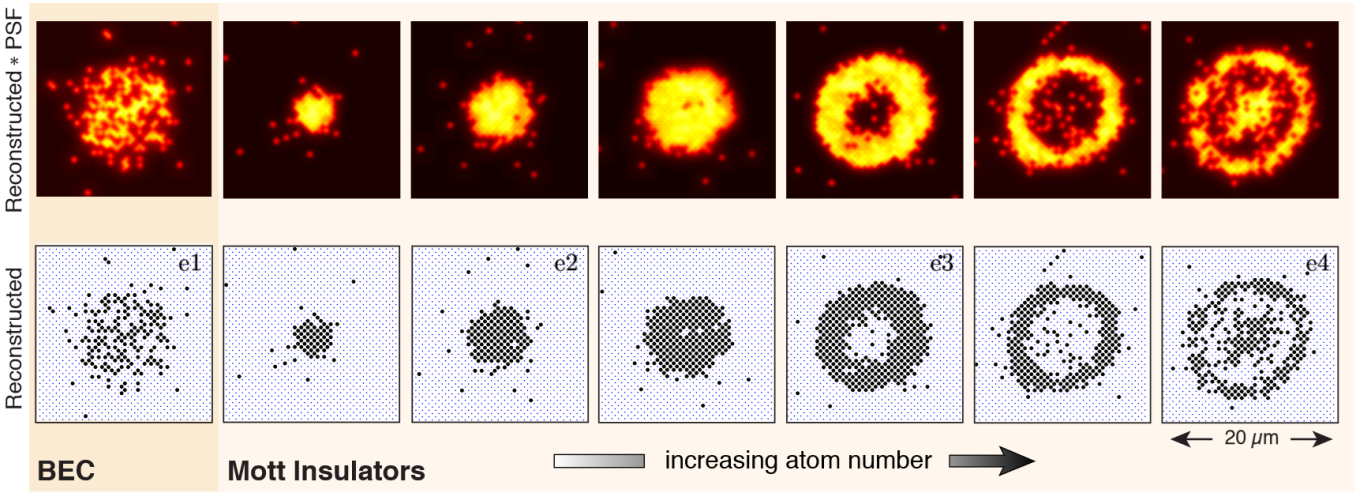

Figure 4. High-resolution fluorescence images of a BEC (SF) and Mott insulators (MI) with in-plane harmonic confinement from [5]. Experimentally obtained images of a BEC (e1) and Mott insulators for increasing particle numbers (e2, e3, e4) in the zero-tunneling limit, showing $\langle n_j \rangle \mod 2$. Clear integer filling of the lattice with $n = 1, 2, 3$ can be observed.

extent of coherence in the system, typical in the superfluid phase, while a smaller ξ indicates short-range correlations, characteristic of the Mott insulator phase.

The correlation length can be determined in two ways: first, by definition,

$$\xi(r) = \frac{\sum_r r^2 \Gamma(r)}{\sum_r \Gamma(r)},$$

although this approach is more sensitive to the finite size of the system. Alternatively, assuming an exponential decay $\Gamma_{\mathrm{MI}}(r) \propto e^{-r/\xi}$, ξ can be found through fitting the calculated correlator, which works well in the MI region. Similarly, by fitting the decay in the SF and critical regions, where we expect a polynomial decay, we can determine the exponent K via $\Gamma^2_{\mathrm{SF}}(r) \propto r^{-K}$. The results of fitting with these two functions are presented in Fig. 2, where it is evident that the decay is exponential for MI and polynomial for SF

Additionally, the average number of particles per site $\langle n_j \rangle$ is a crucial indicator of the MI phase. In the MI phase, $\langle n_j \rangle$ is typically an integer (fig. 4), reflecting the fixed number of particles due to strong repulsive interactions, when particle mobility is suppressed, and the system is incompressible. In contrast, in the superfluid phase, $\langle n_j \rangle$ can vary continuously, indicating higher particle mobility and the absence of a gap for particle-hole excitations. To quantify the deviation from integer filling, we introduce $\delta n_j = n_j - 1$, which helps in analyzing the fluctuations and transitions between the phases.

II. PHASE DIAGRAM

For $t \in [0, 0.3]$ in steps of 0.02 and $\mu \in [0, 1]$ in steps of 0.05 at $U = 1$, the correlators Γ_{ij} were calculated. From these, the values of ξ and K were obtained through fitting, and the results are presented in Fig. 3. The correlation length exhibits a sharp divergence near the critical point $\ln \xi \sim 1/\sqrt{t_c - t}$ subject to finite size limitations. The critical exponent K_{c} of the phase transition is also known from Luttinger liquid theory [1, 6]. At the BKT transition with $\langle n_j \rangle = 1$, K is expected to be $K_{\mathrm{c}} = 1/2$. However, even without this knowledge, it is quite possible to distinguish the two regions based on K. In this context, Luttinger liquid theory simply helps to define this boundary more precisely. The divergence of ξ and the critical value $K = K_{\mathrm{c}}$ are in complete agreement with the phase boundaries defined in [1].

Note that the diagonal part of the calculated correlators $\Gamma_{jj} = n_j$, and by plotting $\langle n_j \rangle$ in the (t, μ)-plane, we can clearly observe the unit filling in the MI phase (Fig. 5). The characteristic points 1, 2, and 3 used earlier for illustration are also marked here. Using linear interpolation with smoothing, we have also delineated the

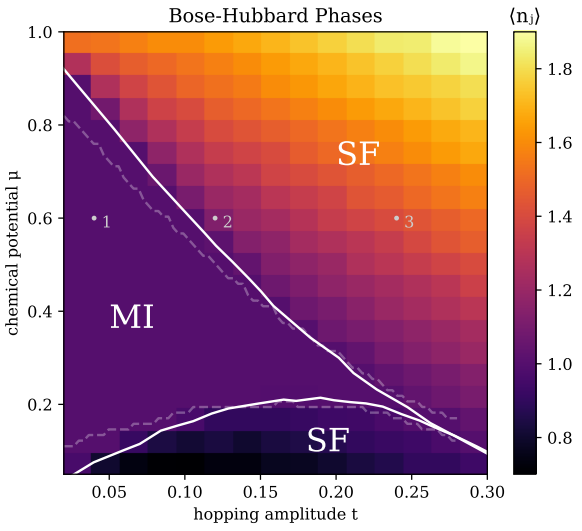

Figure 5. The average number of particles per site. It is evident that the filling is strictly one in the MI region.

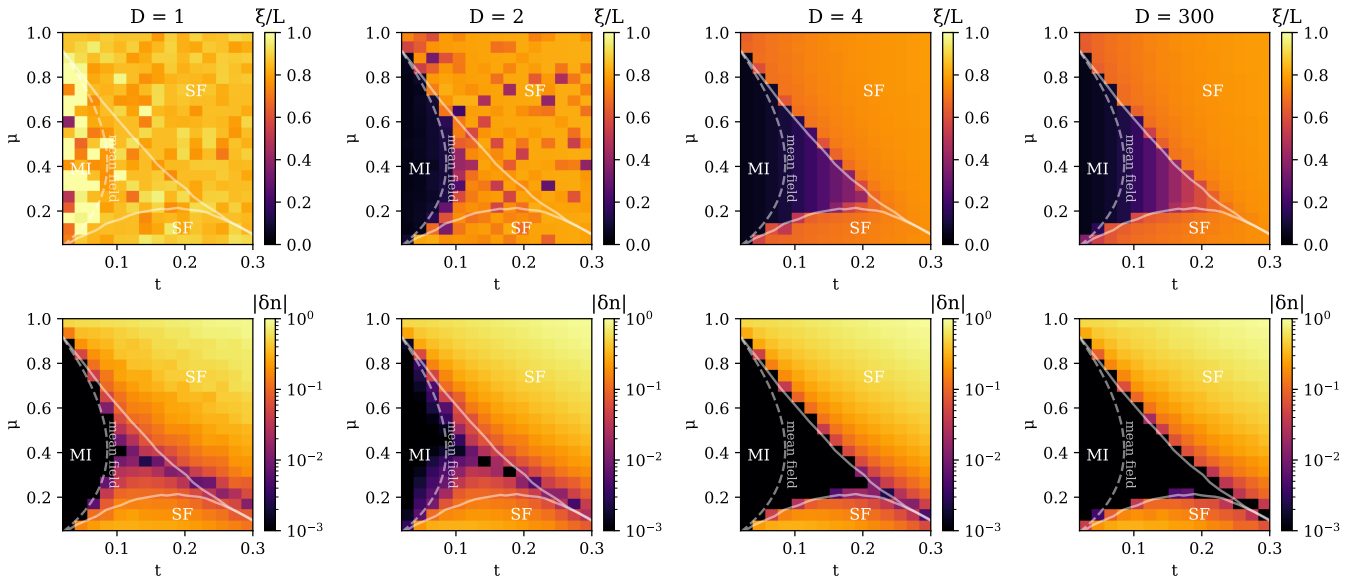

Figure 6. Phase diagrams obtained via DMRG for various bond dimensions D. As D decreases, the solution approaches the mean field solution, both in terms of system filling and correlation length.

boundaries of the $\langle n_j \rangle = 1$ region (white dashed line), which closely align with the phase boundaries. In the SF region, increasing μ leads to a continuous and monotonic increase in the average number of particles.

III. XX-MODEL AND MEAN-FIELF FROM DMRG

It is important to consider the approximations we make in DMRG calculations. For instance, what happens if the bond dimension D is not sufficiently large? In the limit of $D = 1$, we obtain a mean field solution [7], with the phase boundary defined by:

$$1 + 2t\left(\frac{1}{-\mu} + \frac{2}{\mu - U}\right) = 0.$$

However, with $D = 4$, we start to see the characteristic elongated shape of the MI region (Fig. 6).

In the limit of $n_c = 1$, we obtain hard-core bosons, which, through the Jordan-Wigner transformation, correspond to the XX model with paramagnetic and ferromagnetic phases. These phases are also characterized by exponential and polynomial decays, respectively. This gives a characteristic phase boundary $\mu = 2t$ for $\mu, t \ll U$, where μ acts as an external magnetic field [8].

[1] T. D. Kuehner and H. Monien, Phases of the one-dimensional Bose-Hubbard model, Physical Review B **58**, R14741 (1998).

[2] S. R. White, Density matrix formulation for quantum renormalization groups, Phys. Rev. Lett. **69**, 2863 (1992).

[3] G. Catarina and B. Murta, Density-matrix renormalization group: a pedagogical introduction, The European Physical Journal B **96**, 111 (2023).

[4] M. Fishman, S. White, and E. Stoudenmire, The itensor software library for tensor network calculations, SciPost Physics Codebases 10.21468/scipostphyscodeb.4 (2022).

[5] J. F. Sherson, C. Weitenberg, M. Endres, M. Cheneau, I. Bloch, and S. Kuhr, Single-atom-resolved fluorescence imaging of an atomic Mott insulator, Nature **467**, 68 (2010).

[6] T. Giamarchi and A. J. Millis, Conductivity of a luttinger liquid, Phys. Rev. B **46**, 9325 (1992).

[7] M. P. A. Fisher, P. B. Weichman, G. Grinstein, and D. S. Fisher, Boson localization and the superfluid-insulator transition, Phys. Rev. B **40**, 546 (1989).

[8] G. B. Mbeng, A. Russomanno, and G. E. Santoro, The quantum ising chain for beginners, SciPost Physics Lecture Notes 10.21468/scipostphyslectnotes.82 (2024).

Appendix A: MPO Representation of the Hamiltonian

Following [3], we can express the 1D Bose-Hubbard Hamiltonian with several terms:

$$\ldots \overset{4}{\otimes} \mathbb{1} \overset{4}{\otimes} -t\hat{a}^\dagger \overset{2}{\otimes} \hat{a} \overset{1}{\otimes} \mathbb{1} \overset{1}{\otimes} \ldots$$
$$\ldots \overset{1}{\otimes} \mathbb{1} \overset{4}{\otimes} -t\hat{a} \overset{3}{\otimes} \hat{a}^\dagger \overset{1}{\otimes} \mathbb{1} \overset{1}{\otimes} \ldots$$
$$\ldots \overset{1}{\otimes} \mathbb{1} \overset{4}{\otimes} \quad \hat{c} \overset{1}{\otimes} \mathbb{1} \overset{1}{\otimes} \mathbb{1} \overset{1}{\otimes} \ldots$$

where the operator $\hat{c}$ is defined as:

$$\hat{c} = \tfrac{1}{2}U\hat{n}^2 - \left(\mu + \tfrac{U}{2}\right)\hat{n}_j.$$

Thus, we can write our Hamiltonian as an MPO of the form:

$$\hat{H} = \begin{pmatrix} \mathbb{1} & 0 & 0 & 0 \\ \hat{a} & 0 & 0 & 0 \\ \hat{a}^\dagger & 0 & 0 & 0 \\ \hat{c} & -t\hat{a}^\dagger & -t\hat{a} & \mathbb{1} \end{pmatrix}.$$

For $L = 4$, it has been verified that the Hamiltonian obtained from the MPO matches the explicitly written Hamiltonian. Additionally, using Exact Diagonalization (ED), it was confirmed that the ground states obtained from both methods are consistent.

Appendix B: Code listing: Julia

The calculation of the ground state of the system can be implemented using DMRG in Julia and the ITensor package [4], which was convenient for verifying the correctness of the implementation on larger system sizes.

```julia
using NPZ
using ITensors
using ITensorMPS


function calculate_adagacorr(L, mu, U, J, nsweeps, maxdim, cutoff, bc, nc)
    file_name = "data/J$(round(Int,100*J))_mu$(round(Int,100*mu))_L$(L)_bc$(bc)_nc$(nc)_sweeps$(nsweeps)_dim$(maxdim[1]).npy"
    println("processing $(file_name[6:end-4])")

    # Initialize the sites
    sites = siteinds("Boson", L; dim=5, conserve_qns=false)

    # Define the operator sum
    os = OpSum()
    for b in 1:(L - 1)
        os += -mu - U*0.5, "N", b
        os += U*0.5, "N", b, "N", b
        os += -J, "A", b, "Adag", b + 1
        os += -J, "Adag", b, "A", b + 1
    end
    os += -mu - U*0.5, "N", L
    os += U*0.5, "N", L, "N", L
    if bc == 1
        os += -J, "A", 1, "Adag", L
        os += -J, "Adag", 1, "A", L
    end

    # Construct the Hamiltonian as an MPO
    H = MPO(os, sites)

    # Initialize a random MPS
    psi0 = randomMPS(sites)

    # Run DMRG to find the ground state
    energy, psi = dmrg(H, psi0; nsweeps, maxdim, cutoff)

    # Calculate the correlation matrix
    adagacorr = correlation_matrix(psi, "Adag", "A")
    npzwrite(file_name, adagacorr)

    return adagacorr, psi, energy
end
```

Appendix C: Code listing: Python

The code for DMRG was based on [3], with the convolution functions and the DMRG implementation taken directly from GitHub without modifications.

```python
import numpy as np

nc = 5
d = nc+1

# d-dim local operators
_a    = np.diag((1+np.arange(d-1))**0.5, k=1)   #a
_adag = np.diag((1+np.arange(d-1))**0.5, k=-1)  #a^\dag
_n    = np.matmul(_adag, _a)                    #n
_n2   = np.matmul(_n, _n)                       #n^2

_z    = np.zeros((d,d))                         #0
_I    = np.eye(d)                               #1

def mpo2mat(mpo):
    """
    Convert a Matrix Product Operator (MPO) to a regular matrix.
    Parameters:
    mpo (list of ndarray): List of tensors representing the MPO.
    Returns:
    ndarray: The resulting matrix after conversion.
    """
    N = len(mpo)  # Number of tensors in the MPO
    for l in range(N):
        if l == 0:
            # Initialize the resulting matrix with the first tensor in the MPO
            mat = mpo[l][0, :, :, :]
        else:
            # Perform tensor contraction and reshape
            mat = np.einsum('ijk,jlmn', mat, mpo[l])
            mat = np.transpose(mat, (0, 2, 3, 1, 4))
            mat = np.reshape(mat, (np.shape(mat)[0] * np.shape(mat)[1],
                                   np.shape(mat)[2],
                                   np.shape(mat)[3] * np.shape(mat)[4]))
    return mat[:, 0, :]

def gen_H_mpo(N, t=1.0, mu=1.0, U=1.0):
    """
    Generate the MPO Hamiltonian for an XY chain with N sites and open boundary conditions.

    Parameters:
    N (int): Number of sites in the chain.
    t (float): Hopping parameter.
    mu (float): Chemical potential.
    U (float): Interaction strength.

    Returns:
    list of ndarray: The MPO Hamiltonian.
    """
    # Initialize the MPO Hamiltonian tensor
    Hl = np.zeros((4, d, 4, d))
    Hl[0, :, 0, :] = _I
    Hl[1, :, 0, :] = _a
    Hl[2, :, 0, :] = _adag
    Hl[3, :, 0, :] = -(mu + U / 2) * _n + U / 2 * _n2
    Hl[3, :, 1, :] = -t * _adag
    Hl[3, :, 2, :] = -t * _a
    Hl[3, :, 3, :] = _I

    # Construct the MPO Hamiltonian for each site
    H = [Hl for l in range(N)]
    H[0] = Hl[-1:np.shape(Hl)[0], :, :, :]
    H[N-1] = Hl[:, :, 0:1, :]

    return H

def gen_H_mat(N, t=1.0, mu=1.0, U=1.0):
```

```
    """
    Generate the Hamiltonian matrix for an XY chain with N sites.

    Parameters:
    N (int): Number of sites in the chain.
    t (float): Hopping parameter.
    mu (float): Chemical potential.
    U (float): Interaction strength.

    Returns:
    ndarray: The Hamiltonian matrix.
    """
    # Initialize the Hamiltonian matrix
    H = np.zeros((d ** N, d ** N))

    # Construct the Hamiltonian for hopping terms
    for l in range(N - 1):
        Ileft = np.eye(d ** l)  # Identity matrix for left part
        Hmid = np.kron(_z, _z)
        Hmid += -t * np.kron(_adag, _a)
        Hmid += -t * np.kron(_a, _adag)
        Iright = np.eye(d ** (N - l - 2))  # Identity matrix for right part
        H += np.kron(np.kron(Ileft, Hmid), Iright)

    # Construct the Hamiltonian for on-site terms
    for l in range(N):
        Ileft = np.eye(d ** l)  # Identity matrix for left part
        Hmid = _z.copy()
        Hmid += -(mu + U / 2) * _n + U / 2 * _n2
        Iright = np.eye(d ** (N - l - 1))  # Identity matrix for right part
        H += np.kron(np.kron(Ileft, Hmid), Iright)

    return H


# Define MPO for local operators
# Matrix Product Operator (MPO) for the annihilation operator
_a_MPO = np.zeros((1, d, 1, d))
_a_MPO[0, :, 0, :] = _a
# MPO for the creation operator
_adag_MPO = np.zeros((1, d, 1, d))
_adag_MPO[0, :, 0, :] = _adag
# MPO for the identity operator
_I_MPO = np.zeros((1, d, 1, d))
_I_MPO[0, :, 0, :] = _I

def get_corr(M, i, j):
    """
    Calculate the correlation function for a given MPS and indices i and j.

    Parameters:
    M (list of ndarray): List of tensors representing the Matrix Product State (MPS).
    i (int): Index of the site for the creation operator.
    j (int): Index of the site for the annihilation operator.

    Returns:
    complex: The value of the correlation function.
    """
    # Initialize the auxiliary tensor for contraction
    Taux = np.ones((1, 1, 1))
    for l in range(N):
        if l == i:
            # Contract with the creation operator MPO at site i
            Taux = ZipperLeft(Taux, M[l].conj().T, _adag_MPO, M[l])
        elif l == j:
            # Contract with the annihilation operator MPO at site j
            Taux = ZipperLeft(Taux, M[l].conj().T, _a_MPO, M[l])
        else:
            # Contract with the identity operator MPO at other sites
            Taux = ZipperLeft(Taux, M[l].conj().T, _I_MPO, M[l])
    return Taux[0, 0, 0]
```