Wavefunction evolution under the unitary renormalisation group method

I.   GENERAL IDEA BEHIND A REVERSE URG ANALYSIS

A reverse URG analysis involves studying the evolution of wavefunctions from the IR to the UV, by implementing inverse renormalisation group transformations. This is made possible by the fact that the IR fixed point Hamiltonians are often tractable and allow the computation of wavefunctions. Such an analysis gives insight into how states change as high-energy quantum fluctuations are resolved during the Hamiltonian renormalisation from the UV to the IR. It also allows studying the evolution of entanglement measures and correlation functions.

We will start with a very simple IR ground state wavefunction, and then go back towards the UV ground state by applying the inverse unitary operator $U^\dagger$:

$$U : \underbrace{|1, 2, ..., N\rangle}_{\text{UV ground state}} \rightarrow |1, 2, ..., N-1\rangle |N\rangle \rightarrow ... \rightarrow \underbrace{|1, 2, ..., N^*\rangle |N^*+1\rangle ... |N\rangle}_{\text{IR ground state}}$$

$$U^\dagger : \underbrace{|1, 2, ..., N^*\rangle |N^*+1\rangle ... |N\rangle}_{\text{IR ground state}} \rightarrow |1, 2, ..., N^*+1\rangle |N^*+2\rangle ... |N\rangle \rightarrow ... \rightarrow \underbrace{|1, , 2, ..., N\rangle}_{\text{UV ground state}}$$

The first process is the forward RG which we used to obtain the scaling equations. The second process is the reverse RG which we will undertake now. The method has already been applied to study the renormalisation of entanglement in various models of strong correlation. In general, we start with an IR wavefunction that consists a certain number of momentum states n_1 still entangled with the impurity, and the remaining momentum states n_2 disentangled from the impurity. The former are said to be part of the emergent Kondo cloud, while the latter are said to be part of the integrals of motion (IOMs) and appear in direct product with the cloud+impurity system in the ground state wavefunction. Each momentum state is tagged with two conduction bath levels, one above the Fermi surface and one below. These will be represented as $k, \pm$. If we represent the emergent cloud momentum states as $\{k_i\}$ and the IOM states as $\{q_i\}$, the ground state wavefunction can be written as

$$|\Psi\rangle_0 = (\otimes_{i=1}^{n_2} |q_i, -, \uparrow\rangle |q_i, -, \downarrow\rangle) |\Phi\rangle_{\text{cloud}} (\otimes_{i=1}^{n_2} |q_i, +, \uparrow\rangle |q_i, +, \downarrow\rangle) \tag{1}$$

Throughout, we have assumed that the IOMs below the Fermi surface are occupied while those above are unoccupied. This means:

$$|\Psi\rangle_0 = (\otimes_{i=1}^{n_2} |\hat{n}_{q_i, -, \uparrow} = 1\rangle |\hat{n}_{q_i, -, \downarrow} = 1\rangle) |\Phi\rangle_{\text{cloud}} (\otimes_{i=1}^{n_2} |\hat{n}_{q_i, +, \uparrow} = 0\rangle |\hat{n}_{q_i, +, \downarrow} = 0\rangle) \tag{2}$$

$|\Phi\rangle_{\text{cloud}}$ will be obtained by diagonalising a small system of n_1 conduction bath k states. This completes the construction of the IR ground state $|\Psi\rangle_0$. The algorithm of reverse RG then involves applying unitary transformations on this IR wavefunction; the unitary transformations will be the inverse of those that were applied during the forward RG algorithm. Since the forwards RG transformations decoupled k states and transformed them into the IOMS, the inverse transformations will take the momentum states in the IOMs and re-entangle them with the impurity+cloud system, one at a time. This is shown in fig. 1.

The next step is to write down the unitaries that will take us from the IR ground state to the UV ground state. In the forward RG, we used the following unitary for decoupling the shell ϵ_q and spin β is

$$U_{q\beta} = \frac{1}{\sqrt{2}} \left[1 + \eta_{0\beta} + \eta_{1\beta}^\dagger \right] . \tag{3}$$

Here, the subscripts 0 and 1 indicate it decouples an electron above and below the Fermi surface respectively. The inverse transformation for re-entangling $q\beta$ is

$$U_{q\beta}^\dagger = \frac{1}{\sqrt{2}} \left[1 + \eta_{0\beta}^\dagger + \eta_{1\beta} \right] . \tag{4}$$

The wavefunction after reversing one step of the RG will thus be

$$|\Psi\rangle_1 = U_{q\uparrow}^\dagger U_{q\downarrow}^\dagger |\Psi\rangle_0 \tag{5}$$
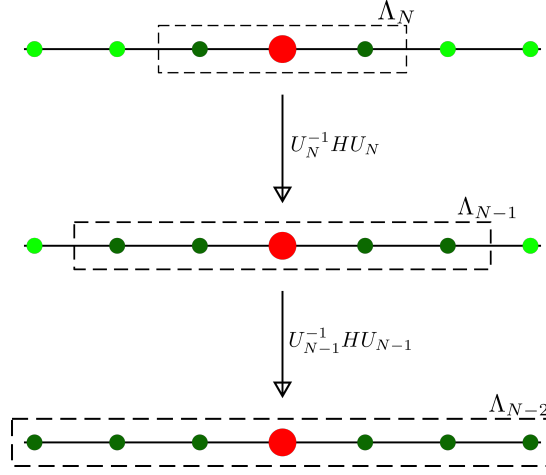
FIG. 1. We start with a Hamiltonian with an impurity site (red) coupled with two conduction electrons (dark green), with four other decoupled electrons (bright green). The dotted rectangle represents the emergent window $(-\Lambda_j, \Lambda_j)$ at each step; the electrons inside that rectangle are still entangled with the impurity, while the ones inside have been decoupled. The next step of reverse RG involves applying the inverse transformation on the Hamiltonian, which will couple two more electrons from the IOMS (hence four dark green circles in the second step), leading to an enlargement of the emergent window. The unitary varies for each step, hence the notation U_j.

Here, q is the first momentum state immediately outside the emergent cloud. Re-entangling further momentum states using their respective unitaries lead to the sequence of wavefunctions $|\Psi\rangle_2, |\Psi\rangle_3$ and so on. A straightforward way of calculating entanglement and correlations across the RG flow is to create a second-quantised prototypical form of the IR state $|\Psi_0\rangle$, and then apply the inverse unitary operators $\{U_i\}$ to this wavefunction in order to generate the family of states $|\Psi_i\rangle = U_i |\Psi_{i-1}\rangle, i > 0$. Quantities like entanglement entropy and correlations can then be calculated by using this family of kets. Such a method, however, soon runs into problems of time complexity and physical memory usage. This is primarily because the operators and the kets are many-body in nature and grow exponentially in size as the number of electronic states are increased.

II. EVOLUTION EQUATION FOR TENSOR NETWORK COEFFICIENTS

We will now show a more efficient method of computing quantities along the reverse RG operations. We will demonstrate this for the much simpler single-channel Kondo problem (which can be obtained from our model by setting all interactions apart from the Kondo coupling J to zero). For this, we will first have to perform some analytical groundwork. The URG transition operators for re-entangling the states at momentum q take the form

$$\eta_{q\beta}^\dagger = \alpha(q) \sum_k \left(S_{\bar{d}}^z \beta c_{q\beta}^\dagger c_{k\beta} + c_{d\bar{\beta}}^\dagger c_{d\beta} c_{q\beta}^\dagger c_{k\bar{\beta}} \right) ,$$

$$\eta_{q\beta} = \alpha(q) \sum_k \left(S_{\bar{d}}^z \beta c_{k\beta}^\dagger c_{q\beta} + c_{d\beta}^\dagger c_{d\bar{\beta}} c_{k\bar{\beta}}^\dagger c_{q\beta} \right) ,$$

(6)

for the Kondo problem, where we have defined $\alpha(q) = \frac{1}{d_2} \frac{J}{2}$. Let (m_l) n_l be the number of momentum states that are (not) entangled with the impurity at the lth step. The total dimension of the system is then $N_l = 1 + 2m_l + 2n_l$, where 1 takes care of the impurity spin and the factors of 2 account for the spin degeneracy. We can write the groundstate at this step in the form

$$|\Psi\rangle_l = \sum_{j=1}^{2^{N_l}} \mathcal{C}_l(s_j) |\sigma_d^z = s_j(d)\rangle \otimes_{i=1}^{m_l} |n_{k_i} = s_j(k_i)\rangle \otimes_{i=1}^{n_l} |n_{q_i} = s_j(q_i)\rangle \tag{7}$$

where $\{s_j; j = 0, \ldots, 2^{N_l} - 1\}$ is the Cartesian product of the set of occupancies $\{0, 1\}$ of the N_l members, and the sum is over all the possible classical number conserved configurations. $s_j(k_i)$ represents the occupancy (either 0 or 1) of the member k_i within the configuration s_j. As an example, the configuration $s_j = 1001$ means that the impurity

spin has spin up ($s_j(0) = 1$), the first and second momentum states are unoccupied ($s_j(1) = s_j(2) = 0$) and the third momentum state is occupied ($s_j(3) = 1$). Note that within the summation of eq. 7, each momentum state also has a spin index which has been suppressed.

We will now apply the URG transition operators on such a general state. As a matter of bookkeeping, we will retain the configurations of states only up to the momentum index that will be decoupled at the current step, because the IOM states beyond it will not be modified at the current iteration.

$$\eta_{q\beta} |\Psi\rangle_l = \sum_{j=1}^{2^{N_l}} \mathcal{C}_l(s_j)\alpha(q) \sum_k \left(S_d^z \beta c_{k\beta}^\dagger c_{q\beta} + c_{d\beta}^\dagger c_{d\overline{\beta}} c_{k\overline{\beta}}^\dagger c_{q\beta} \right) |\sigma_d^z = s_j(d)\rangle \otimes_{i=1}^{m_l} \otimes_\beta |n_{k_i\beta} = s_j(k_i\beta)\rangle \otimes_\beta |n_{q\beta} = s_j(q\beta)\rangle$$
(8)

where $\beta = \pm 1$ runs over the spin indices. Note that the first part of the transition operator only picks out classical configurations that satisfy $n_{k\beta} = 1 - n_{q\beta} = 0$. Similarly, the second part picks out configurations with $n_{k\overline{\beta}} = 1 - n_{q\beta} = 0, \sigma_d^z = \overline{\beta}$. The expression therefore evaluates to

$$\eta_{q\beta} |\Psi\rangle_l = \sum_{j=1}^{2^{N_l}} \mathcal{C}_l(s_j)\alpha(q) \sum_k \left[\delta_{s_j(k\beta),0}\delta_{s_j(q\beta),1} S_d^z \beta c_{k\beta}^\dagger c_{q\beta} |\sigma_d^z = s_j(d)\rangle \otimes_{i=1}^{m_l} \otimes_\beta |n_{k_i\beta} = s_j(k_i\beta)\rangle \otimes_\beta |n_{q\beta} = s_j(q\beta)\rangle \right.$$
$$\left. + \delta_{s_j(d),(1-\overline{\beta})/2}\delta_{s_j(k\overline{\beta}),0}\delta_{s_j(q\beta),1} c_{d\beta}^\dagger c_{d\overline{\beta}} c_{k\overline{\beta}}^\dagger c_{q\beta} |\sigma_d^z = s_j(d)\rangle \otimes_{i=1}^{m_l} \otimes_\beta |n_{k_i\beta} = s_j(k_i\beta)\rangle \otimes_\beta |n_{q\beta} = s_j(q\beta)\rangle \right]$$
(9)

The first Kronecker delta in the second term, $\delta_{s_j(d),(1+\beta)/2}$, ensures that $s_j(0)$ is unity when $\beta = -1$ and zero when $\beta = 1$. We can now apply the fermionic and spin operators on the states. We first consider the spin-preserving operator $c_{k\beta}^\dagger c_{q\beta}$, and apply it on the ket, paying attention to any sign changes arising from fermion exchanges. Since the pair itself is bosonic, it can be transported without incurring any sign change:

$$c_{k\beta}^\dagger c_{q\beta} \otimes_{i=1}^{m_l} \otimes_\beta |s_j(k_i\beta)\rangle \otimes_\beta |s_j(q\beta)\rangle$$
$$= (\otimes_{k_i<k,\beta} |s_j(k_i\beta)\rangle) c_{k\beta}^\dagger c_{q\beta} |s_j(k\beta)\rangle |s_j(k\overline{\beta})\rangle (\otimes_{k_i>k,\beta} |s_j(k_i\beta)\rangle) |s_j(q\beta)\rangle |s_j(q\overline{\beta})\rangle$$
(10)

In order to apply the operator $c_{q\beta}$, we will need to transport it further to the end. This will involve exchanging it consecutively with the fermionic states $k_i \geq k$, and will produce factors of -1:

$$c_{k\beta}^\dagger c_{q\beta} \otimes_{i=1}^{m_l} \otimes_\beta |s_j(k_i\beta)\rangle \otimes_\beta |s_j(q\beta)\rangle$$
$$= \prod_{q>k_i\geq k,\beta} (-1)^{s_j(k_i\beta)} (\otimes_{k_i<k,\beta} |s_j(k_i\beta)\rangle) c_{k\beta}^\dagger |s_j(k\beta)\rangle |s_j(k\overline{\beta})\rangle (\otimes_{k_i>k,\beta} |s_j(k_i\beta)\rangle) c_{q\beta} |s_j(q\beta)\rangle |s_j(q\overline{\beta})\rangle$$
(11)

The spin-flip operator will produce the same factor. We can now apply the operators and obtain the final expression:

$$\eta_{q\beta} |\Psi\rangle_l =$$
$$\sum_k \sum_{j=1}^{2^{N_l}} \prod_{q>k_i\geq k,\beta} (-1)^{s_j(k_i\beta)} \mathcal{C}_l(s_j)\alpha(q) \left[\delta_{s_j(k\beta),0}\delta_{s_j(q\beta),1} \left(s_j(d) - \frac{1}{2} \right) \beta |\sigma_d^z = s_j(d)\rangle \ldots |n_{k\beta} = 1\rangle \ldots \otimes |n_{q\beta} = 0\rangle \ldots \right.$$
$$\left. + \delta_{s_j(d),(1+\overline{\beta})/2}\delta_{s_j(k\overline{\beta}),0}\delta_{s_j(q\beta),1} |\sigma_d^z = \frac{1+\beta}{2}\rangle \ldots |n_{k\overline{\beta}} = 1\rangle \ldots \otimes |n_{q\beta} = 0\rangle |n_{q\overline{\beta}} = s_j(q\overline{\beta})\rangle \right]$$
(12)

The factor of $\left(s_j(d) - \frac{1}{2} \right)$ arises from the fact that if $s_j(d) = 1, 0$ indicates up and down respectively, then $S_d^z |s_j(d)\rangle = \left(s_j(d) - \frac{1}{2} \right) |s_j(d)\rangle$. We see that applying the transition operator projects the general state on to some other specific states, by starting from a different set of states (as picked out through the Kronecker delta factors). The entire expression in the above equation then serves as a renormalisation of the classical configurations that the state ends up at, with the renormalisation itself arising from the initial states projected on to by the Kronecker delta factors. For example, the first part of the expression tells us that the initial configurations with $s_j(k\beta) = 0$ and $s_j(q\beta) = 1$ renormalise the final state with $s_j(k\beta) = 1$ and $s_j(q\beta) = 0$. We can therefore write down the renormalisation equations of the coefficients $\mathcal{C}_l(s_j)$:

$$\Delta\mathcal{C}_l \left(s_j; s_j(k\beta) = 1, s_j(q\beta) = 0 \right) = \alpha \left(s_j(d) - \frac{1}{2} \right) \beta \mathcal{C}_l \left(s_j; s_j(k\beta) = 0, s_j(q\beta) = 1 \right) \prod_{q>k_i\geq k,\beta} (-1)^{s_j(k_i\beta)}$$
(13)

$$\Delta\mathcal{C}_l \left(s_j; s_j(d) = \frac{1+\beta}{2}, s_j(k\overline{\beta}) = 1, s_j(q\beta) = 0 \right) = \alpha \mathcal{C}_l \left(s_j; s_j(d) = \frac{1-\beta}{2}, s_j(k\overline{\beta}) = 0, s_j(q\beta) = 1 \right) \prod_{q>k_i\geq k,\beta} (-1)^{s_j(k_i\beta)}$$
(14)

These expressions apply individually on all configurations s_j, as long as they satisfy the constraints within the brackets. The indices of s_j that are not constrained can take any value, as long the values are the same on both sides of the equation. All such expressions of renormalisation will have to be added appropriately to obtain the total renormalisation of a particular coefficient.

This only accounts for particle transition operator η. We can similarly apply the hole transition operator $\eta^\dagger$ to our state. The expressions for that case can be obtained from our existing expression in eqs. 13 and 14 simply by performing a particle-hole transformation $c_{q\beta} \to c_{q\beta}^\dagger, c_{k\beta}^\dagger \to -c_{k\beta}$, which leads to the effective transformation $\mathcal{C}_l(s_j) \to -\mathcal{C}_l(1 - s_j)$:

$$\Delta \mathcal{C}_l \left(s_j; s_j(k\beta) = 0, s_j(q\beta) = 1 \right) = -\alpha \left(s_j(d) - \frac{1}{2} \right) \beta \mathcal{C}_l \left(s_j; s_j(k\beta) = 1, s_j(q\beta) = 0 \right) \prod_{q > k_i \geq k, \beta} (-1)^{s_j(k_i\beta)} \qquad (15)$$

$$\Delta \mathcal{C}_l \left(s_j; s_j(d) = \frac{1 - \beta}{2}, s_j(k\bar\beta) = 0, s_j(q\beta) = 1 \right) = -\alpha \mathcal{C}_l \left(s_j; s_j(d) = \frac{1 + \beta}{2}, s_j(k\bar\beta) = 1, s_j(q\beta) = 0 \right) \prod_{q > k_i \geq k, \beta} (-1)^{s_j(k_i\beta)} \qquad (16)$$

Eqs. 13 through eq. 16 completely describe the renormalisation in the coefficients $\mathcal{C}_l$ arising from applying the reverse URG transformations.

The reverse RG analysis would now involve starting with some initial set of coefficients, depending on the IR ground state, and then recursively calculating the coefficients at the next by applying the renormalisation equation on the previous step. Having obtained the sequence of coefficients, various quantities can be calculated using them. The density matrix has the general form

$$\rho = \sum_{j,j'} \mathcal{C}_l(s_{j'})^* \mathcal{C}_l(s_j) |s_j(d)\rangle \langle s_{j'}(d)| \ldots |s_j(q)\rangle \langle s_{j'}(q)| \ . \qquad (17)$$

For discussions regarding bipartite measures of entanglement, it makes sense to divide the system into two subsystems A and B, such that the state $|\Psi\rangle_l$ can be written as

$$|\Psi\rangle_l = \sum_{i_A} \sum_{j_B} C(i_A, j_B) |i_A\rangle |j_B\rangle \ , \qquad (18)$$

where i_A and j_B sum over the eigenstates of the (potentially many-particle) subsystems A and B respectively. From this, the reduced density matrix $\rho(A)$ for the subsystem A can be expressed as the partial trace of the full density matrix over the states $\{j_B\}$ of system B:

$$\rho(A) = \sum_{j_B} \langle j_B|\rho|j_B\rangle = \sum_{\substack{i_A, i'_A, \\ j_B, j'_B, j''_B}} C^*(i'_A, j''_B) C(i_A, j'_B) \langle j''_B|j_B\rangle \langle j_B|j'_B\rangle |i_A\rangle \langle i'_A| = \sum_{i_A, i'_A, j_B} C^*(i'_A, j_B) C(i_A, j_B) |i_A\rangle \langle i'_A| \ . \qquad (19)$$

For numerical purposes, this expression can be neatly expressed as the product of a rectangular matrix C with its hermitian conjugate $C^\dagger$:

$$\rho(A) = CC^\dagger \ . \qquad (20)$$

The matrix C has $\dim_A$ number of rows and n_B number of columns, where $\dim_A$ is the dimension of the Hilbert space of A and n_B is the number of eigenstates of B that appear in the complete eigenstate $|\Psi\rangle_l$ of eq. 18. The elements of C are given by $\langle i_A|C|j_B\rangle = C_{i_A, j_B}$.

However, the least expensive approach to calculating the reduced density matrix is through the Schmidt coefficients.

Using these expressions, entanglement measures like entanglement entropy and mutual information can be obtained easily. One might also wish to compute static correlation functions along the RG flow. As an example, the spin-flip correlation between the impurity spin and a single $k-$ state can be written as

$$\langle \Psi_l | S_d^+ S_k^- + \text{h.c.} | \Psi_l \rangle = \sum_j \delta_{s_j(d)=0} \delta_{s_j(k\uparrow)=1} \delta_{s_j(k\downarrow)=0} \mathcal{C}_l(\bar s_j)^* \mathcal{C}_l(s_j) + \text{h.c.} \ , \qquad (21)$$

where $\bar s_j$ is a configuration that is identical to s_j except for the fact that $s_j(d), s_j(k\uparrow)$ and $s_j(k\downarrow)$ have been flipped. This can be extended to other correlation functions.

We end by suggesting some general guidelines that can help reduce system load while computing the renormalisation in the coefficients:

- Start with as few IOMs as possible, and add more IOMS as you go along. This reduces the time and memory required for the initial steps of the RG.

- Typically, most of the coefficients $\mathcal{C}_l(s)$ of the wavefunction are zero, so it is more efficient to store only the non-zero coefficients. This reduces the size of the coefficients array from exponentially large (2^{N_l}) to only linearly large ($\sim N_l$).

- Try to store as few large arrays at a time as possible. If you need two large arrays A_1 and A_2 but do not need them in the same expression, use the same variable name for them so that the memory of the A_1 gets freed when you reuse its variable to define A_2.

- Try not to retain all the results in memory. Instead, save them in secondary memory by writing to files, and load these files as and when required.

- Separate out memory-intensive tasks into functions which you can call when required; this makes it easier for the system to reallocate that memory when the function has ended.

- Instead of looping over large arrays, use techniques like broadcasting and vectorized operations to perform manipulations.