iNORG: An open-source quantum impurity solver package based on the natural orbitals renormalization group

Jia-Ming Wang[a,b], Yi-Heng Tian[a,b], Yin Chen[a,b], Ru Zheng[c], Rong-Qiang He[a,b,*], Zhong-Yi Lu[a,b,*]

[a]*Department of Physics, Renmin University of China, Beijing 100872, China*
[b]*Key Laboratory of Quantum State Construction and Manipulation (Ministry of Education), Renmin University of China, Beijing 100872, China*
[c]*Department of Physics, School of Physical Science and Technology, Ningbo University, Ningbo 315211, China*

Abstract

In the context of dynamical mean-field theory (DMFT) calculations for strongly correlated electron systems, quantum impurity solvers play a central computational role in treating correlated lattice models and realistic materials. Consequently, developing efficient and robust quantum impurity solvers remains a key challenge. In this paper, we present an open-source quantum impurity solver package based on the natural orbitals renormalization group (NORG) method, dubbed iNORG. This software delivers high accuracy with reduced computational cost by optimizing the bath representation using natural orbitals and incorporating advanced features such as efficient Hilbert space selection and efficient algorithms for computing Green's functions. We first introduce the basic principle of the NORG method and then discuss the implementation details. The software framework, major features, and installation procedure for iNORG are explained as well. Finally, several simple examples are presented to demonstrate the usage of iNORG.

Keywords: Quantum impurity model, natural orbitals renormalization group method, dynamical mean-field theory

PROGRAM SUMMARY

Program Title: iNORG
CPC Library link to program files: (to be added by Technical Editor)
Developer's repository link: https://github.com/jmw-phys/iNORG
Code Ocean capsule: (to be added by Technical Editor)
Licensing provisions (please choose one): AGPL-3.0
Programming language: C++
Supplementary material:
*Journal reference of previous version:**
*Does the new version supersede the previous version?:**
*Reasons for the new version:**
*Summary of revisions:**
Nature of problem (approx. 50-250 words): Quantum many-body calculations in dynamical mean-field theory (DMFT) hinge on solving an Anderson-type impurity model embedded in a self-consistent bath. For realistic multi-orbital systems the bath must be discretised with tens to hundreds of orbitals, so the full Fock-space dimension rises exponentially and quickly overwhelms conventional exact-diagonalisation (ED). Quantum Monte-Carlo incurs a severe sign-problem at low temperature and requires ill-conditioned analytic continuation, while numerical renormalization group loses accuracy for low-temperature features with many correlated orbitals. Consequently there is a pressing need for an impurity solver that can retain ED-level accuracy for large, general-interaction impurity models yet scale far more favourably with bath size, enabling high-resolution zero-temperature DMFT studies of strongly correlated materials.
Solution method (approx. 50-250 words): iNORG addresses this challenge with the natural orbitals renormalization group (NORG). The bath is first transformed to an optimal natural-orbital basis extracted from the single-particle density matrix of the correlated

*Corresponding author.
E-mail address: jmw@ruc.edu.cn (J.-M. Wang), rqhe@ruc.edu.cn (R.-Q. He), zlu@ruc.edu.cn (Z.-Y. Lu)

ground state. A hierarchy of natural-orbital occupancy constraints (NOOC) compresses the space by freezing the degrees of freedom associated with nearly full or empty orbitals, thereby building a highly compressed active space. The reduced Hamiltonian is solved by a Lanczos algorithm, while single-particle Green's functions are obtained from continued-fraction expressions within the Krylov basis. hybridization functions are discretised automatically using a Levenberg-Marquardt fit, and computation-intensive kernels call multi-threaded BLAS and LAPACK. The overall complexity scales polynomially ($\sim O(N_{\text{bath}}^3)$).

Additional comments including restrictions and unusual features (approx. 50-250 words): The current version of `iNORG` is restricted to zero-temperature, single-site impurity models and presently supports only density-density Coulomb interactions; extensions for finite-temperature calculations, cluster DMFT, and two-particle response functions are planned for future releases. Among its notable features is an adaptive natural orbital occupancy constraint (NOOC) scheme that dynamically adjusts the active Hilbert space to optimize computational efficiency. The implementation introduces a 'NORG-Table' data structure that caches sparse matrix information, enabling rapid Hamiltonian reconstruction across self-consistent iterations. Furthermore, the automated bath discretization routine provides advanced user control, allowing for non-uniform frequency grids to be specified for the hybridization function fitting, which is particularly useful for resolving sharp spectral features.

1. Introduction

In recent years, strongly correlated electron systems, such as high-temperature superconductors and quantum magnets, have garnered significant attention due to their rich physical phenomena and potential applications in advanced materials. A central challenge in this field is accurately describing the behavior of electrons in the presence of strong interactions, which generally requires solving complex quantum many-body problems. One effective approach to tackle these systems is dynamical mean-field theory (DMFT), which maps a lattice problem onto an effective impurity model embedded in a self-consistently determined bath [1, 2].

A central component of DMFT is the impurity solver, which computes the self-energy of the impurity model. The Anderson impurity model (AIM) [3], which describes localized electrons with Coulomb interactions in a conduction bath, is fundamental to studying strongly correlated systems. Generally, it underpins phenomena like the Kondo effect [4], dilute magnetic impurities [5], and impurity quantum phase transitions [6]. In DMFT [1, 2], the AIM plays a central role, as lattice models are mapped to self-consistent impurity problems, with multi-orbital systems posing significant computational challenges due to complex inter-orbital hybridizations [7, 8].

Analytical solutions for the AIM are typically infeasible, necessitating numerical methods. Various impurity solvers have been developed, each with its strengths and limitations. Traditional approaches include exact diagonalization (ED) [9] and Lanczos [10], which are accurate for small bath sizes but scale exponentially with Hilbert space size; quantum Monte Carlo (QMC) [11], suited for finite temperatures but limited at low temperatures or real frequencies; the numerical renormalization group (NRG) [12], effective for low energies but less so for high energies or multi-orbital systems; and hierarchical equations of motion (HEOM) [13], which is efficient under specific conditions but costly with increasing bath complexity.

To address these challenges, the natural orbitals renormalization group (NORG) method has been proposed as an efficient and accurate impurity solver [14, 15]. NORG leverages the concept of natural orbitals (NOs) [16, 17] to find a compact ground-state representation for the AIM, exploiting correlation sparsity. This enables efficient approximations with few Slater determinants, reducing computational complexity. In this article, we introduce `iNORG`, an open-source implementation of the NORG method designed to solve quantum impurity models effectively, particularly for multi-orbital systems where other methods struggle.

2. Model and methods

This section first introduces the Anderson impurity model solved by `iNORG` and explains the motivation for optimizing the bath-orbital representation using natural orbitals. Subsequently, we systematically present the definition of natural orbitals and orbital transformations, and detail the self-consistent procedure of the iterative natural orbitals renormalization group (NORG) method and subspace screening strategies. Finally, we introduce the Fock encoding of many-body wave functions and the numerical implementation of obtaining ground states using the Lanczos-Krylov method, laying the theoretical foundation for subsequent implementation and performance optimization.

2.1. Anderson impurity model

Within the dynamical mean-field theory (DMFT), solving the Anderson impurity model (AIM) [3] constitutes the computational core. The AIM describes a localized interacting subsystem (the impurity) coupled to a non-interacting bath, and it is widely studied for strongly correlated electron phenomena, such as the Kondo effect and Mott transitions. Typically, the bath represents conduction electrons. The general form of the Hamiltonian can be expressed as:

$$H_{\text{imp}} = H_{\text{loc}} + H_{\text{bath}} + H_{\text{hyb}}\,, \tag{1}$$

where

$$\begin{aligned} H_{\text{loc}} &= \sum_{\alpha\beta} E_{\alpha\beta} d^\dagger_\alpha d_\beta + \sum_{\alpha\beta\gamma\delta} U_{\alpha\beta\gamma\delta} d^\dagger_\alpha d^\dagger_\beta d_\gamma d_\delta\,, \\ H_{\text{hyb}} &= \sum_{\mathbf{k}\alpha\beta} V^{\alpha\beta}_{\mathbf{k}} d^\dagger_\alpha c_{\beta\mathbf{k}} + \text{h.c.}\,, \\ H_{\text{bath}} &= \sum_{\alpha\mathbf{k}} \epsilon_{\alpha\mathbf{k}} c^\dagger_{\alpha\mathbf{k}} c_{\alpha\mathbf{k}}\,. \end{aligned} \tag{2}$$

Here, $d^\dagger_\alpha$ and $c^\dagger_{\alpha\mathbf{k}}$ are the electron creation operators for the impurity and bath, respectively, with α and β being spin-orbital indices. H_{loc} accounts for both the single-particle energies and Coulomb interactions on the impurity orbitals. H_{hyb} characterizes the hybridization between the impurity and bath orbitals, while H_{bath} corresponds to the non-interacting conduction bath. Solving the AIM requires balancing computational speed and accuracy. Three main solvers span this spectrum: the Hubbard-I approximation, the exact diagonalization (ED), and the quantum Monte Carlo (QMC). Although the model itself is compact, reliable Green's functions must be obtained from computationally intensive numerical methods. The Hubbard-I runs quickly but misses strong-correlation physics [18]. The ED yields exact results for small clusters but its cost grows exponentially with system size [19]. The QMC can handle larger systems, yet it suffers from the sign problem and the need to analytically continue data from imaginary to real frequencies [11]. Modern DMFT studies demand many orbitals, very low temperatures, and high real-frequency resolution. No single solver can yet meet all these requirements simultaneously.

This computational bottleneck has motivated the development of the natural orbitals renormalization group (NORG) approach [14, 15], which circumvents many traditional limitations through a fundamentally different strategy: the iterative optimization of the orbitals themselves for a fixed many-body wave function that complies with natural orbital constraints. By fully utilizing the inherent sparsity in the AIM and focusing computational resources on the most quantum-mechanically active degrees of freedom while effectively treating less relevant contributions, the NORG achieves a remarkable balance between accuracy and efficiency, which opens new possibilities for tackling previously intractable many-body problems. In the last few years, the NORG method has been successfully used as an impurity solver in DMFT for many projects, such as the investigation of low-energy inter-band Kondo bound states in orbital-selective Mott phases [20], the DFT+DMFT analysis of correlated electronic structures in layered nickelates [21], the study of non-Fermi-liquid regimes and antiferromagnetic correlations in bilayer Hubbard models [22], and the elucidation of two-orbital $s_\pm$-wave superconductivity in high-pressure $La_3Ni_2O_7$ [23].

2.2. Natural orbitals renormalization group method

2.2.1. Natural orbitals

Natural orbitals (NOs) was first proposed by Löwdin in Chemistry [16, 24]. We demonstrated that natural obitals provide an optimal single-particle basis for describing the many-body quantum state $|\Psi\rangle$ of the AIM [14]. They are defined as the eigenvectors of the single-particle density matrix (SPDM) with its element defined as follows,

$$D_{ij} = \langle\Psi|\hat{c}^\dagger_i \hat{c}_j|\Psi\rangle, \tag{3}$$

where $\hat{c}^\dagger_i$ and $\hat{c}_j$ are creation and annihilation operators in an arbitrary initial single-particle basis $\{|i\rangle\}$, respectively. Diagonalizing the SPDM yields the NOs, denoted as $\{|k\rangle\}$, with corresponding occupation numbers n_k satisfying $0 \le n_k \le \text{n}_{\text{max}}$, where $\text{n}_{\text{max}} = 1$ for spinless fermions and $\text{n}_{\text{max}} = 2$ for spatial orbitals with spin degeneracy.

In the framework of the NORG method [14, 15], the NO basis is iteratively refined to optimize the representation of a many-body wave function, enabling efficient solutions of complex quantum systems. To formulate the NORG

method, we first transform the initial single-particle basis into a general orbital basis $\mathcal{S}$. In this basis, the single-particle orbitals are orthonormal and denoted as $|g\rangle = d_g^\dagger|\text{vac}\rangle$ with $g = 1, 2, \ldots, L$, where $d_g^\dagger$ and d_g are the creation and annihilation operators for the g-th orbital. The transformation to this basis can be written as $|g\rangle = \sum_{i=1}^{L} U_{gi}^\dagger |i\rangle$, where U_{gi} are the elements of a unitary matrix U. Then, the corresponding operator transformations are expressed as:

$$d_g^\dagger = \sum_{i=1}^{L} U_{gi}^\dagger c_i^\dagger, \quad d_g = \sum_{i=1}^{L} U_{ig} c_i, \tag{4}$$

or, in the matrix form:

$$\begin{pmatrix} d_1^\dagger \\ d_2^\dagger \\ \vdots \\ d_L^\dagger \end{pmatrix} = U^\dagger \begin{pmatrix} c_1^\dagger \\ c_2^\dagger \\ \vdots \\ c_L^\dagger \end{pmatrix}, \quad \begin{pmatrix} d_1, & d_2, & \cdots, & d_L \end{pmatrix} = \begin{pmatrix} c_1, & c_2, & \cdots, & c_L \end{pmatrix} U.$$

Since U is unitary, the new operators satisfy the fermionic anticommutation relations: $\{d_g, d_{g'}^\dagger\} = \delta_{gg'}$, $\{d_g, d_{g'}\} = 0$, $\{d_g^\dagger, d_{g'}^\dagger\} = 0$. Correspondingly, the inverse transformation can be obtained by:

$$c_i^\dagger = \sum_{g=1}^{L} U_{ig} d_g^\dagger, \quad c_i = \sum_{g=1}^{L} U_{gi}^\dagger d_g. \tag{5}$$

In the representation $\mathcal{S}$ (which NORG aims to optimize), the wave function $|\Psi\rangle$ can be expanded by considering the occupation of a specific orbital g:

$$|\Psi\rangle = \sum_i h_i^\circ \left|\phi_i^{\circ,g}\right\rangle + \sum_j h_j^\bullet \left|\phi_j^{\bullet,g}\right\rangle, \tag{6}$$

where $\left|\phi_i^{\circ,g}\right\rangle$ are N-electron Slater determinants not containing the g-th orbital, and $\left|\phi_j^{\bullet,g}\right\rangle$ are those containing the g-th orbital. These Slater determinants are orthonormal:

$$\langle \phi_i^m | \phi_j^n \rangle = \delta_{ij}\delta_{mn}, \quad m, n \in \{\circ, \bullet\},$$

and the coefficients satisfy the normalisation condition:

$$\sum_i |h_i^\circ|^2 + \sum_j |h_j^\bullet|^2 = 1.$$

The occupation number of orbital g is

$$n_g^d \equiv \langle \Psi | d_g^\dagger d_g | \Psi \rangle. \tag{7}$$

Using Eq. (6), we obtain

$$n_g^d = \sum_j |h_j^\bullet|^2 = 1 - \sum_i |h_i^\circ|^2, \tag{8}$$

which shows that n_g^d directly measures the occupation of orbital g in $|\Psi\rangle$. The NORG method iteratively optimizes $\mathcal{S}$ (by adjusting U) so that the n_g^d of transformed orbitals approach 0 or $\mathrm{n_{max}}$. This allows a clear distinction between 'active' (important) and 'inactive' (secondary) degrees of freedom, facilitating an effective solution of the many-body problem.

In the NORG framework, orbitals in the optimized basis $\mathcal{S}$ are classified by their occupation numbers n_g^d:

- **Inactive orbitals**: $n_g^d \approx 0$ (nearly empty) or $n_g^d \approx \mathrm{n_{max}}$ (nearly full), with $\mathrm{n_{max}} = 1$ for spinless fermions or 2 for spatial orbitals with spin degeneracy. These orbitals can be "frozen" to reduce the Hilbert space dimension, with an associated error of approximately n_g^d or $(\mathrm{n_{max}} - n_g^d)$.

- **Active orbitals**: Orbitals with n_g^d significantly different from 0 and n_{max}, indicating substantial quantum fluctuations. These require explicit treatment for an accurate system description.

This classification thus partitions the NO basis into active and inactive subsets. The NORG method leverages this by iteratively optimizing the basis $\mathcal{S}$ to distinguish between active and inactive NOs, enabling a compressed representation of the many-body wave function $|\Psi\rangle$. The key advantage is that $|\Psi\rangle$ converges more rapidly in this basis, allowing computational effort to focus on active NOs while approximating inactive ones. Hence, this reduction in Hilbert space complexity is the cornerstone of NORG's efficiency.

2.2.2. Renormalization strategy

NORG is a non-perturbative numerical technique tailored for quantum impurity models that effectively exploits the NO basis [14, 15]. It is especially effective for systems with sparse interactions, as is typical in the AIMs from DMFT, in which interactions are confined to impurity sites.

The core NORG workflow is an iterative self-consistent procedure:

1. **Initialize**: Start from an initial set of single-particle orbitals (e.g., the original bath sites).
2. **Construct subspace**: Apply different constraints to different orbitals to construct an effective Hilbert subspace.
3. **Build effective Hamiltonian**: Construct the Hamiltonian matrix $\hat{H}_{\text{eff}}$ within the chosen active subspace.
4. **Solve ground state**: Solve for the approximate ground state $|\Psi_{\text{sub}}\rangle$ of $\hat{H}_{\text{eff}}$, typically using an iterative method such as Lanczos.
5. **Compute SPDM**: Compute the SPDM D_{sub} using the obtained $|\Psi_{\text{sub}}\rangle$.
6. **Update NOs**: Diagonalize D_{sub} to obtain a new set of NOs and their occupations n_k.
7. **Iterate**: Use the updated NOs to refine the selection of the active subspace and repeat steps 3–7 until convergence criteria (e.g., ground state energy, occupations) are met.

By iteratively refining the NO basis and dynamically adjusting the active space to focus on the most relevant quantum fluctuations (active NOs) while freezing the less important ones (inactive NOs), NORG achieves high accuracy with significantly reduced computational cost compared to the ED of the full problem. The computational complexity scales polynomially with the number of bath sites N_{bath} (typically $O(N_{\text{bath}}^3)$ or $O(N_{\text{bath}}^4)$), making it feasible to handle systems with hundreds of bath orbitals.

In the NORG computational process, the model is transformed from a lattice basis (the real-space representation) to the natural orbital basis (the optimized basis composed of the eigenstates of the single-particle density matrix). The orbital transformation aims to improve computational efficiency through the optimized NO basis. Two transformation strategies can be adopted:

1. **Transforming all orbitals simultaneously:** The entire single-particle density matrix (SPDM) is diagonalized to obtain a unitary transformation U acting on all orbitals. All sites in the original model (including interacting impurity orbitals and non-interacting bath orbitals) are mapped to the NO space. However, interaction terms (such as the local Hubbard U) are then distributed across multiple NOs, increasing the number of non-zero elements in the Hamiltonian matrix and making it denser. The computational complexity grows as $O(N^4)$ with the total number of orbitals N.
2. **Transforming only non-interacting orbitals:** Only the block of the SPDM corresponding to the bath orbitals (non-interacting sites) is diagonalized, while the impurity orbitals (interacting sites) are left unchanged. This strategy avoids distributing the interaction terms, keeping the Hamiltonian matrix sparse. Although this transformation may introduce weak off-diagonal couplings among the new bath NOs, the number of non-zero Hamiltonian elements increases only slightly. The computational complexity grows as $O(N_{\text{bath}}^3)$ with the number of bath orbitals N_{bath}, significantly improving efficiency. Importantly, because the impurity orbitals are unchanged, their physical properties are preserved, and the overall effectiveness of the algorithm is not compromised.

2.3. Representation of many-body wave functions

In the second quantization framework, many-electron wave functions are typically represented using Fock states. For a system with M spatial orbitals, each comprises two spin degrees of freedom (up and down), resulting in a total

of $2M$ spinorbitals. By assigning an occupation number $n_i \in \{0, 1\}$ to each orbital, the complete many-body Hilbert space can be constructed as:

$$\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_{2M},$$

where $\mathcal{H}_N$ represents the subspace with electron occupation number N, and the total dimension is 2^{2M}. Since the Hilbert space dimension grows exponentially with the number of orbitals $2M$, practical calculations usually focus only on the subspace $\mathcal{H}_N$ with a specific occupation number N.

Within this framework, creation and annihilation operators $\hat{c}_i^\dagger$ and $\hat{c}_i$ flip the occupation of orbital i ($i = 0, 1, \ldots, 2M-1$) between 0 and 1. This "0-1" binary nature reflects the Pauli exclusion principle, i.e., each orbital can be occupied by at most one fermion.

In the `iNORG` code, a single-particle basis $\{\alpha\}$ is first fixed, and based on this, the Fock basis $\{|I\rangle\}$ is constructed. The orbital numbering follows the convention: spin-up ($\uparrow$) orbitals first, followed by spin-down ($\downarrow$) orbitals. Specifically, orbitals are numbered from 0 to $2M-1$, where:

$$0, 1, \ldots, M-1 \quad (\uparrow \text{ states}); \qquad M, M+1, \ldots, 2M-1 \quad (\downarrow \text{ states}).$$

Fock states are encoded using a $2M$-bit binary number, where "1" indicates the orbital is occupied, and "0" indicates it is unoccupied. For example, consider a system with $M = 3$, having 6 orbitals numbered 0 ($\uparrow$), 1 ($\uparrow$), 2 ($\uparrow$), 3 ($\downarrow$), 4 ($\downarrow$), 5 ($\downarrow$). The binary encoding convention has the least significant bit (LSB) corresponding to orbital 0, increasing to orbital 5. A specific Fock state binary string is:

$$\overbrace{001}^{0\text{–}2}\ \overbrace{100}^{3\text{–}5},$$

indicating that orbital 2 ($\uparrow$) and orbital 3 ($\downarrow$) are occupied. Its corresponding decimal value is:

$$\begin{aligned} &0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 0 \times 2^5 \\ &= 0 + 0 + 4 + 8 + 0 + 0 = 12. \end{aligned} \tag{9}$$

This binary encoding facilitates computer storage and manipulation, providing an efficient representation for numerical solutions of many-body problems.

2.4. Lanczos algorithm

The Lanczos algorithm has been thoroughly described in multiple review articles [25, 26]. This section briefly outlines the basic idea of the algorithm and provides its pseudocode. The Lanczos algorithm is suitable for solving for the extremal eigenvalues and eigenvectors of large, sparse matrices and can also be used to compute spectral functions (such as single-particle Green's functions), which we will introduce in the following sections. In this section, we focus exclusively on groundstate calculation.

The core idea of the Lanczos method is to avoid direct diagonalization of the full Hamiltonian H. Instead, starting from a random initial state vector $|\varphi_0\rangle$, it iteratively constructs an orthonormal basis for a much smaller subspace, known as the Krylov subspace:

$$\mathcal{K}_m(H, |\varphi_0\rangle) = \text{span}\{|\varphi_0\rangle, H|\varphi_0\rangle, H^2|\varphi_0\rangle, \ldots, H^{m-1}|\varphi_0\rangle\}. \tag{10}$$

In this basis, the projection of the original Hamiltonian H becomes a small, tridiagonal matrix, T_m. The genius of the method is that the extremal eigenvalues of T_m converge rapidly to the extremal eigenvalues of the full H. By diagonalizing this small tridiagonal matrix, one can obtain a highly accurate approximation of the ground state energy and wave function of the original system with a modest number of iterations (typically $m \approx 200$) [27–30].

The orthonormal basis vectors $\{|\varphi_i\rangle\}$ that tridiagonalize the Hamiltonian are generated via a three-term recurrence relation. Starting with a normalized random vector $|\varphi_0\rangle$, the sequence is generated as follows:

$$\begin{aligned} \alpha_i &= \langle\varphi_i|H|\varphi_i\rangle \\ \beta_{i+1}|\varphi_{i+1}\rangle &= (H - \alpha_i)|\varphi_i\rangle - \beta_i|\varphi_{i-1}\rangle. \end{aligned} \tag{11}$$

where $\beta_0 = 0$ and $|\varphi_{-1}\rangle = 0$. In each step, the new vector $|\varphi_{i+1}\rangle$ is orthogonalized against the previous two vectors and then normalized. The coefficients $\{\alpha_i\}$ and $\{\beta_i\}$ form the diagonal and off-diagonal elements, respectively, of the tridiagonal matrix T_m.

3. Implementations and optimizations

3.1. Overall workflow of `iNORG`

This section describes the implementation details and optimization techniques of `iNORG`, an open-source C++ software package [31] specifically designed for impurity solving within the DMFT framework. It employs the natural orbitals renormalization group (NORG) method [14] to efficiently handle strongly correlated electron behavior in quantum impurity models. Figure 1 illustrates the main computational workflow of `iNORG`, which we will explore in detail in the following sections.

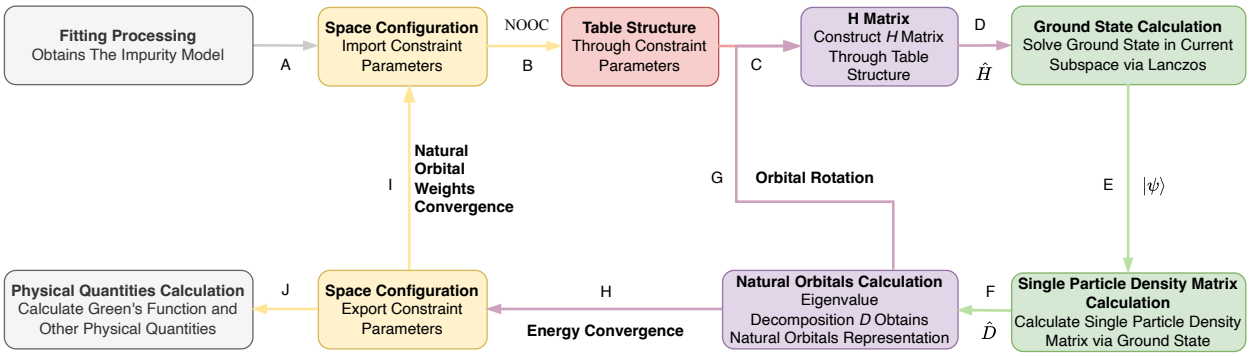

Figure 1: Flowchart of the natural orbitals renormalization group (NORG) method as implemented in the `iNORG` package. The process starts from an Anderson impurity model defined by DMFT (A). It then enters a self-consistent cycle (B-I) to optimize the natural orbitals by repeatedly constructing and solving the Hamiltonian within a selected Hilbert subspace. After convergence, physical observables are calculated (J).

Step A: Obtain the local hybridization function $\Gamma_{\mathrm{loc}}(z)$ via DMFT, which characterizes the interaction between the impurity and the environmental bath. The preprocessing module discretizes the continuous $\Gamma_{\mathrm{loc}}(z)$ into a finite number of bath sites. In the Anderson Impurity Model ($H_{\mathrm{imp}} = H_{\mathrm{loc}} + H_{\mathrm{bath}} + H_{\mathrm{hyb}}$, see Eq. (2)), this determines H_{hyb} and H_{bath}, ensuring the impurity's hybridization function $\Gamma_{\mathrm{imp}}(z) \approx \Gamma_{\mathrm{loc}}(z)$. Interaction parameters (e.g., Coulomb interaction strength) are input to define H_{loc}, thus specifying the impurity model.

Step B: Screen the Hilbert subspace using the natural orbital occupancy constraint (NOOC) algorithm, constructing the NORG-Table structure required for computation.

Step C: Utilize the NORG-Table structure to store the mapping between many-body bases in the Hilbert subspace and operator matrix elements in the single-particle basis, eliminating the need to construct various many-body operator matrices during subsequent iterations.

Step D: Apply the Lanczos algorithm to the Hamiltonian $\hat{H}$ to solve for the ground-state wave function $|\psi\rangle$ and ground-state energy E_0 within the subspace.

Step E: Compute the single-particle density matrix $D_{\alpha\beta} = \langle\psi| c_\alpha^\dagger c_\beta |\psi\rangle$ based on $|\psi\rangle$.

Step F: Perform eigenvalue decomposition on D to obtain natural orbitals $\{\phi_i\}$ and their occupancies $\{n_i\}$.

Step G: Check whether the energy difference between consecutive ground states satisfies the convergence criterion. If not, return to Step C to optimize the subspace; otherwise, proceed to Step H.

Step H: Export the occupancy numbers in the natural orbital representation and record the natural orbital transformation matrix.

Step I: Calculate the norm difference of natural orbital occupancies between consecutive iterations, $d(n_{\mathrm{i,last}}, n_{\mathrm{i,new}}) = \sum_{i=1}^{n} |n_{\mathrm{i,last}} - n_{\mathrm{i,new}}|$. If $d \le \Delta_d$, proceed to Step J; otherwise, return to Step B to adjust the Hilbert subspace.

Step J: After natural orbital convergence, compute the Green's function and other physical quantities.

To realize this workflow, `iNORG` adopts the following technical strategies during development to ensure flexibility, stability, and extensibility:

1. Modular and Independent Design: All modules are written in C++17 using object-oriented programming, allowing independent operation while reducing coupling for easy maintenance and expansion.

2. Development Environment: Targeted for Linux platforms, using Makefiles for compilation management, integrated with MPI parallelism and Intel oneAPI's BLAS and LAPACK libraries.
3. Model Support: Defaults to DMFT calculations for multi-orbital Bethe lattices, with the ability to support other lattice models through modifications.
4. Software Integration: Compatible with DFT+DMFT software (e.g., ZEN, eDMFT), enabling parameter passing via configuration files for seamless data exchange and collaborative computation.

The implementation of these technical strategies, especially the principle of modular design, results in the software architecture illustrated in Fig. 2. This diagram outlines the core modules of `iNORG` and uses color-coding to visually link each module to its corresponding steps in the workflow flowchart (as shown in Fig. 1).

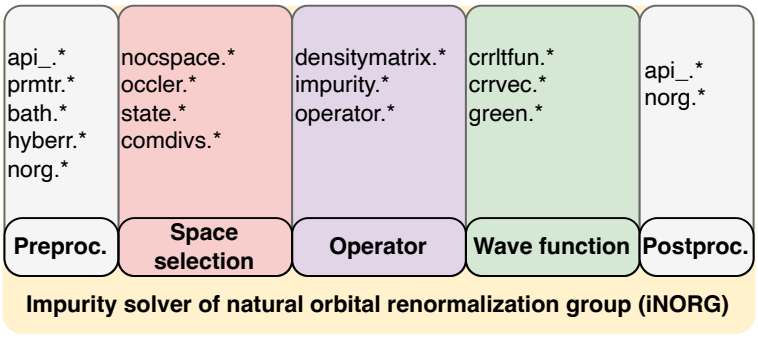

Figure 2: Organizational structure of the `iNORG` program. The first row shows the module abbreviations, and the second row indicates the corresponding module functions.

The functions of these modules are summarized below. Owing to space constraints, the following subsections present only the essential technical details required to reproduce the program, omitting repetitive descriptions of similar implementation steps.

- **Preprocessing and Postprocessing**: Implements hybridization function fitting (Step A) and physical quantity computation (Step J), detailed in Section 3.2.
- **Space selection**: Includes NOOC (Step B), operator relation storage (Step C), and convergence judgment (Step I), detailed in Sections 3.3, 2.3, and 3.4.
- **Operator**: Rapid construction of many-body operators (Step D) and orbital transformations (Step F), detailed in Sections 3.5.
- **Wave function**: Ground-state solution via the Lanczos algorithm (Step D), wave function analysis (Step E), and physical quantity computation (Steps H, J), detailed in Sections 3.6, 3.7, and 3.8.

3.2. Hybridization function fitting algorithm

In this software package, we have implemented an automated hybridization function processing capability, employing the Levenberg–Marquardt fitting algorithm to discretize the impurity model within the DMFT framework. In the context of strongly correlated electron systems under DMFT, the continuous local hybridization function $\Gamma_{\mathrm{loc}}(z)$ must be discretized into a finite number of bath sites to construct an effective model coupled to the impurity orbitals. This process can be viewed as a nonlinear least-squares problem, aiming to minimize the error between the fitted hybridization function $\Gamma_{\mathrm{imp}}(z)$ and the original $\Gamma_{\mathrm{loc}}(z)$, where $\Gamma_{\mathrm{loc}}(z) = z - \mathcal{G}_l^{-1}(z)$ is defined by the local Green's function $\mathcal{G}_l(z)$.

3.2.1. Fitting details

The Levenberg–Marquardt algorithm combines the strengths of the Gauss–Newton method and gradient descent, using a damping factor λ to balance numerical stability and convergence speed. In each iteration, the parameter update equation is:

$$\boldsymbol{\theta}_{\text{new}} = \boldsymbol{\theta}_{\text{old}} - (J^{\mathrm{T}}J + \lambda I)^{-1} J^{\mathrm{T}}\mathbf{r}, \tag{12}$$

where $\mathbf{r}$ is the residual vector, and J is the Jacobian matrix. The damping factor λ adjusts dynamically based on fitting performance: when the residual χ^2 decreases, λ is reduced to accelerate convergence; when χ^2 increases, λ is increased to ensure stability. The algorithm exhibits near-quadratic convergence near the optimal solution, effectively reducing iteration counts and enhancing discretization efficiency and accuracy [32].

The choice of initial parameters is crucial for convergence. Leveraging physical prior knowledge (e.g., spectral symmetry or hybridization strength estimates) to set the initial $\{\epsilon_k, V_k\}$ parameters reduces the risk of converging to local minima. In numerical implementation, we monitor the condition number of the Jacobian matrix to ensure stability and employ stopping criteria (e.g., χ^2 change below a threshold or maximum iteration limit) to guarantee robust and efficient fitting.

Specifically, the discretized hybridization function is expressed as:

$$\Gamma_{\text{imp}}(z) = \sum_{k=1}^{N_b} \frac{|V_k|^2}{z - \epsilon_k}, \tag{13}$$

where N_b is the number of bath sites. On discrete complex frequency points $\{z_n\}_{n=1}^{N_z}$e.g., Matsubara frequencies $z_n = i\omega_n$the target residual function is:

$$\begin{aligned}\chi^2 = &\sum_n w_n \left|\Gamma_{\text{loc}}(z_n) - V(\epsilon_{\text{bath}} - z_n)^{-1}V^{\dagger}\right|^2 \\ &+ w_{N+1}\left|VV^{\dagger} - (K_{0,\text{loc}} - (H_{0,\text{loc}})^2)\right|^2 \\ &+ w_{N+2}\left(\sum_k \epsilon_{\text{bath},k}^4\right)^2,\end{aligned} \tag{14}$$

where the first term measures the fitting error between $\Gamma_{\text{loc}}(z_n)$ and the discretized hybridization function, which is formed from the impurity-bath coupling matrix V and the bath energy levels ϵ_{bath} as $V(\epsilon_{\text{bath}} - z_n I)^{-1}V^{\dagger}$. The second term introduces bath sum rules [33] that control bath parameters by comparing $VV^{\dagger}$ with the local non-interacting part, $K_{0,\text{loc}} - (H_{0,\text{loc}})^2$. The moments $H_{0,\text{loc}}$ and $K_{0,\text{loc}}$ are defined by integrating the non-interacting Hamiltonian H_0 and its square over the Brillouin zone: $H_{0,\,\text{loc}} \equiv \int_{\text{BZ}} \frac{\mathrm{d}k^d}{(2\pi)^d} H_0$ and $K_{0,\,\text{loc}} \equiv \int_{\text{BZ}} \frac{\mathrm{d}k^d}{(2\pi)^d} (H_0)^2$. The third term acts as a normalization constraint, limiting the fourth power of bath energy levels $\epsilon_{\text{bath},k}$ to prevent parameter divergence. While a simple function is used here as an example, normalization constraints can also take more complex forms.

Ultimately, the impurity Hamiltonian is constructed as:

$$H_{\text{imp}} = H_{\text{loc}} + H_{\text{bath}} + H_{\text{hyb}}, \tag{15}$$

where:

$$H_{\text{bath}} = \sum_{k,\sigma} \epsilon_{k,\sigma}^{*} b_{k\sigma}^{\dagger} b_{k\sigma}, \tag{16}$$

$$H_{\text{hyb}} = \sum_{k,\sigma} \left(V_{k,\sigma}^{*} c_{\sigma}^{\dagger} b_{k\sigma} + \text{h.c.}\right), \tag{17}$$

and H_{loc} is the local part of the original lattice model. This model accurately reproduces local dynamical properties within DMFT iterations.

3.2.2. *Fitting performance*

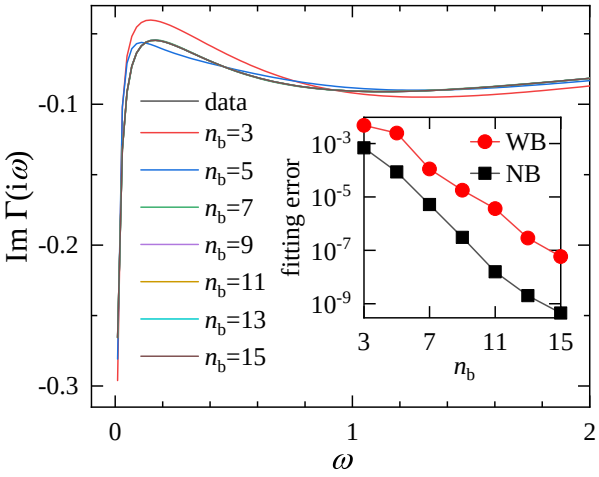

Figure 3: Impurity Model Fitting Performance Example: Displays the imaginary part of the wide-band (WB) hybridization function $\Gamma_{\text{imp}}(i\omega)$, with parameters $U = 3$, $\Delta = 0.3$, $t_2 = 0.5t_1$. The inset shows fitting errors for wide-band (WB) and narrow-band (NB) cases under varying n_b (number of bath sites per impurity orbital).

In quantum impurity models, H_{bath} and H_{hyb} are determined by fitting $\Gamma_{\text{imp}}(z)$ to $\Gamma_{\text{loc}}(z) = z - \mathcal{G}_l^{-1}(z)$. For a diagonal hybridization function matrix (ignoring inter-orbital mixing), we have:

$$\Gamma_{\text{imp},l}(z) = \sum_k \frac{|V_{lk}|^2}{z - \epsilon_{lk}}. \tag{18}$$

Fig. 3 presents a fitting example, demonstrating that fitting quality improves significantly with increasing n_b. The inset shows that fitting error decreases exponentially with n_b; when $n_b \geq 7$, errors for both wide-band (WB) and narrow-band (NB) cases drop below 10^{-4}, validating the method's effectiveness.

3.3. *Natural orbitals occupancy constraint*

Once a specific impurity model to be solved is obtained, the problem we face is how to efficiently and accurately solve this quantum impurity model. In practice, we find that applying the natural orbitals occupancy constraint (NOOC) renders the NORG method for solving impurity models highly efficient and robust [14, 15]. Therefore, the computational workflow described below is based on the NOOC-optimized NORG method, abbreviated as NOOC-NORG.

The basic unit of NOOC can be written as $m^{l\pm}$, where m represents the number of spin-orbitals (i.e., these m orbitals are spin-distinguished single-particle states, so the total electron number in this group is bounded above by m.); m^* denotes an active orbital group, where no occupancy constraints are imposed, allowing the electron number to vary freely from 0 to m; $l+$ (or $l-$) indicates that in the hole constraint group (or electron constraint group), up to l holes (or electrons) are allowed relative to the fully occupied (or empty) state. Specifically:

- For the hole constraint group ($l+$), the fully occupied state means all spin-orbitals are occupied by electrons, and allowing up to l holes implies that the number of electrons is at least $m - l$;
- For the electron constraint group ($l-$), the empty state means all spin-orbitals are unoccupied, and allowing up to l electrons implies that the number of electrons does not exceed l.

Once NOOC is determined, we can construct the corresponding Hilbert subspace for the impurity model based on these constraints, and subsequent calculations (such as ground state solving or Green's function computation) will also be performed within this effective Hilbert subspace. The NOOC-NORG algorithm is therefore a variational

method, making the choice of an appropriate constraint rule the key to solving this impurity model. Building on the directproduct constraint, we devised the joint and PHES constraints to exploit sparse impurity-bath interactions and further accelerate NORG by partitioning the Hilbert space more finely.

3.3.1. Direct product constraint rule

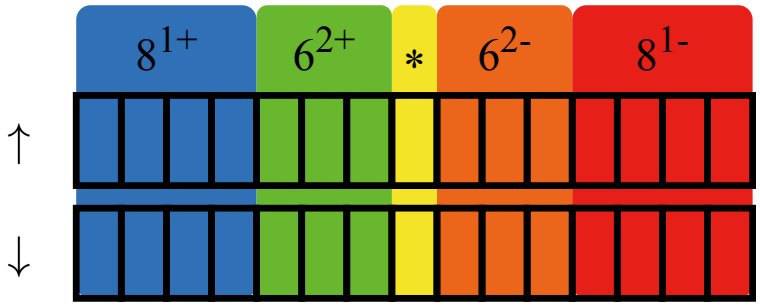

Figure 4: Illustration of the Direct product constraint rule

We consider the direct product constraint rule using an example of a system where a single impurity orbital couples with 15 bath orbitals. The single impurity corresponds to one spatial orbital (2 spin-orbitals), and the bath corresponds to 15 spatial orbitals (30 spin-orbitals). Thus, the system comprises 32 spin-orbitals in total. For clarity in the following discussion, the term "orbital" will refer to a spin-orbital.

Under the direct product constraint rule, the 30 bath orbitals are divided into five groups. The NOOC is applied to each group as follows:

- First group (8 orbitals): Relative to full occupancy (all orbitals occupied), up to 1 hole is allowed, denoted as 8^{1+}.
- Second group (6 orbitals): Relative to full occupancy, up to 2 holes are allowed, denoted as 6^{2+}.
- Third group (2 orbitals): No occupancy restrictions are applied, denoted as 2^{*}. This group's orbital count matches the impurity orbitals. This active group is designed to hybridize effectively with the static impurity orbitals, as it remains unconstrained during the bath transformation.
- Fourth group (6 orbitals): Relative to full emptiness (all orbitals unoccupied), up to 2 electrons are allowed, denoted as 6^{2-}.
- Fifth group (8 orbitals): Relative to full emptiness, up to 1 electron is allowed, denoted as 8^{1-}.

The core of the direct product constraint rule lies in directly combining each NOOC basic unit via a direct product to construct the total Hilbert subspace of the system. Specifically, the system's Hilbert space is formed by the direct product of the subspaces of the five bath orbital groups and the impurity orbital subspace. The impurity orbital constraint is typically denoted as 2^{*}; since the impurity orbitals do not participate in unitary transformation, they are often omitted in notation. Thus, the constraint applied to the bath portion is fully denoted as $8^{1+}6^{2+}2^{*}6^{2-}8^{1-}$.

3.3.2. Joint constraint rule

To uniformly handle electron and hole distributions in the impurity model, we designed the joint constraint rule. While the direct product rule is intuitive, it treats electron-like and hole-like excitations in completely separate groups, which can be rigid and computationally suboptimal. The joint constraint rule overcomes this by blurring the distinction between them and introducing the concept of "change quantity." Specifically, the change quantity n is defined as the number of electrons appearing in regions preset to be fully empty (n_e) plus the number of holes appearing in regions preset to be fully occupied (n_h), i.e., $n = n_e + n_h$.

Under the joint constraint rule, the 30 bath orbitals are divided into three groups:

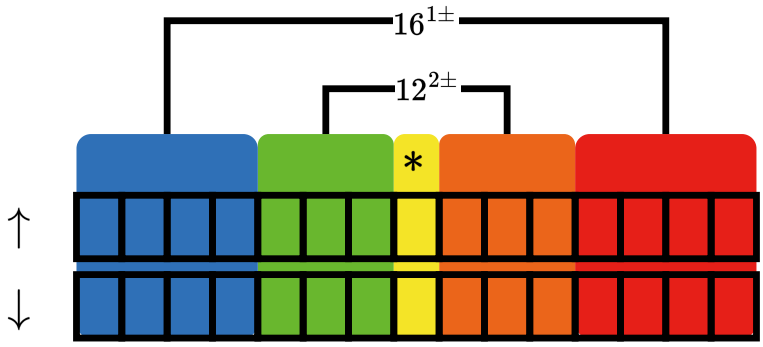

Figure 5: Illustration of the Joint Constraint Rule

- First group: Contains 2 orbitals, strongly coupled to the impurity orbitals, with no occupancy restrictions, denoted as 2^*. Since these orbitals are strongly coupled to the impurity, their occupancy distribution significantly affects the system's properties. Therefore, no restrictions are imposed, preserving their full degrees of freedom.
- Second group: Contains 12 orbitals, with the first 6 preset to be fully occupied and the latter 6 preset to be fully empty, allowing up to 2 change quantities relative to the preset state (i.e., the sum of holes in the first 6 orbitals and electrons in the latter 6 orbitals does not exceed 2), denoted as $12^{2\pm}$.
- Third group: Contains 16 orbitals, with the first 8 preset to be fully occupied and the latter 8 preset to be fully empty, allowing up to 1 change quantity relative to the preset state (i.e., the sum of holes in the first 8 orbitals and electrons in the latter 8 orbitals does not exceed 1), denoted as $16^{1\pm}$.

Here, the notation $M^{n\pm}$ indicates a group of M orbitals where up to n change quantities are allowed, which can be either electrons in preset fully empty regions or holes in preset fully occupied regions, depending on the orbital's preset state. As shown in Figure 5, the division of the three orbital groups and their preset states is illustrated. The innovation of the joint constraint rule lies in uniformly describing holes and electrons through change quantities, offering greater flexibility in adapting to the sparse interaction characteristics in impurity models compared to the traditional direct product constraint rule. Thus, the constraint applied to the bath portion is fully denoted as $2^*12^{2\pm}16^{1\pm}$.

3.3.3. PHES constraint rule

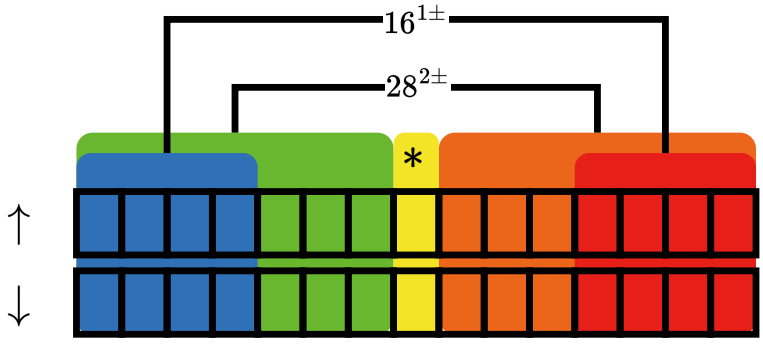

Figure 6: Illustration of the PHES Constraint Rule

In studying the joint constraint rule $2^*12^{2\pm}16^{1\pm}$, we found that it could not effectively distinguish configurations with low ground-state weights. To analyze this limitation, we classified all possible configurations by their number of excitations (or "change quantities"), as shown in Table 1: As shown in the table, configuration #6, $\begin{bmatrix} * & 2 & 1 \end{bmatrix}^1$, has a total of 3 excitations. Although this configuration contributes little to the ground state, the joint constraint rule

Table 1: Classification of configurations by number of excitations under the joint constraint rule

No.	Group 1 Excitations	Group 2 Excitations	Group 3 Excitations	Total Excitations
#1	*	0	0	0
#2	*	1	0	1
#3	*	2	0	2
#4	*	0	1	1
#5	*	1	1	2
#6	*	2	1	3

$(2^{*}12^{2\pm}16^{1\pm})$ fails to exclude it. The inclusion of such low-weight, high-excitation configurations reduces computational efficiency.

To address this issue, we introduce the particle-hole excitation state (PHES) constraint rule, which applies a nested set of conditions. This rule uses the same orbital grouping as the joint constraint (an active group, group 2, and group 3), but it imposes two simultaneous conditions on groups 2 and 3:

1. A local constraint on the outermost group (group 3, with 16 orbitals), limiting its number of excitations to ≤ 1.
2. A global constraint on the combination of group 2 and group 3 (totaling 28 orbitals), limiting their total number of excitations to ≤ 2.

This nested approach allows the PHES constraint to control configurations more precisely. Consider configuration #6, $\left[*\quad 2\quad 1\right]$. Under the previous joint constraint rule, this configuration is allowed because the number of excitations in group 3 (1) satisfies the $16^{1\pm}$ limit, and the number in group 2 (2) satisfies the $12^{2\pm}$ limit. However, the PHES rule rejects it because the total number of excitations in groups 2 and 3 is 3, which violates the global constraint (≤ 2).

Figure 6 illustrates the orbital grouping and nested structure of the PHES constraint, providing an intuitive visualization of its design. By excluding high-excitation configurations like #6, the PHES constraint is more effective than the joint constraint at filtering out low-weight configurations, thereby significantly enhancing computational efficiency. Therefore, the complete constraint applied to the bath orbitals is denoted as $2^{*}(12 + 16^{1\pm})^{2\pm}$.

In summary, `iNORG` implements a hierarchy of NOOC rules, from the foundational direct product constraint to the more sophisticated joint and PHES constraints. This tiered approach provides the user with the flexibility to select the optimal balance between computational accuracy and efficiency for a given impurity problem. The ability to systematically and physically truncate the vast Hilbert space through these rules is the cornerstone of the NORG method's high performance.

3.4. Encoding of many-body wave functions

To store and rapidly retrieve many-body configurations in quantum impurity solvers, it is crucial to develop efficient encoding methods for Fock states, since the dimension of Hilbert space grows exponentially with the number of orbitals. Here we present the optimized state encoding, which directly supports rapid construction and manipulation of Hamiltonian matrices, essential for iterative methods like the Lanczos and NORG algorithms discussed earlier.

3.4.1. Combinatorial number encoding

For any Fock state, it can be viewed as a binary string, where '1' indicates the corresponding orbital is occupied, and '0' indicates empty. For the case of exactly two electrons in six orbitals, a typical scenario encountered in bath orbital encoding within impurity models, the encoding method is demonstrated explicitly. Take $C(6, 2)$ (i.e., $\binom{6}{2} = 15$, meaning exactly 2 out of 6 orbitals are occupied) as an example, with orbitals numbered from 0 to 5. For instance, $|0, 0, 1, 1, 0, 0\rangle$ indicates that both orbitals 2 and 3 (numbering starts from 0) are occupied, corresponding to the binary code "001100", which is 12 in decimal. In this way, any configuration satisfying the constraint 6^{2-} (six orbitals, at most two electrons) can be finally converted into an integer code for easy storage in the program.

[1]Note: In $\left[*\quad a\quad b\right]$, '*' indicates no restriction on the number of excitations in the first group; the second and third groups have a and b excitations, respectively. The total number of excitations is the sum of those in the second and third groups.

However, it is inefficient to directly store these 15 configurations (i.e., 15 integers). As shown in Table 2, the decimal codes (48, 40, 36, 34, 33, 24, 20, 18, 17, 12, 10, 9, 6, 5, 3) are not consecutive, making them unsuitable as direct array indices, which will waste memory. Therefore, to address this problem, we use the combinatorial number system to arrange all states with 2 occupations in lexicographical order, and map them to a unique integer code (ranking). This code equals the number of states preceding it in the lexicographical order. Conversely, given a code, the Fock state can be reconstructed using the unranking method. The table below lists all 15 configurations and their corresponding information:

Table 2: Fock states and their encodings for $C(6,2)$

No.	Occupied orbitals	Fock state	Binary code	Decimal code
0	{0,1}	$\|1,1,0,0,0,0\rangle$	110000	48
1	{0,2}	$\|1,0,1,0,0,0\rangle$	101000	40
2	{0,3}	$\|1,0,0,1,0,0\rangle$	100100	36
3	{0,4}	$\|1,0,0,0,1,0\rangle$	100010	34
4	{0,5}	$\|1,0,0,0,0,1\rangle$	100001	33
5	{1,2}	$\|0,1,1,0,0,0\rangle$	011000	24
6	{1,3}	$\|0,1,0,1,0,0\rangle$	010100	20
7	{1,4}	$\|0,1,0,0,1,0\rangle$	010010	18
8	{1,5}	$\|0,1,0,0,0,1\rangle$	010001	17
9	{2,3}	$\|0,0,1,1,0,0\rangle$	001100	12
10	{2,4}	$\|0,0,1,0,1,0\rangle$	001010	10
11	{2,5}	$\|0,0,1,0,0,1\rangle$	001001	9
12	{3,4}	$\|0,0,0,1,1,0\rangle$	000110	6
13	{3,5}	$\|0,0,0,1,0,1\rangle$	000101	5
14	{4,5}	$\|0,0,0,0,1,1\rangle$	000011	3

In Table 2, configurations are sorted in lexicographical order of occupied orbital numbers (from smallest to largest), with numbers 00 to 14 representing their ranks. For example, the configuration $|0,0,1,1,0,0\rangle$ (occupying orbitals {2,3}) has a rank of 10. The combinatorial ranking/unranking algorithms provide a compact bijection between configurations and integers, significantly enhancing memory and computational efficiency. Using the combinatorial ranking algorithm (Algorithm 1) and unranking algorithm (Algorithm 2), we can achieve bidirectional conversion between configurations and ranks without storing all configurations, thereby saving storage.

Algorithm 1 Combinatorial Ranking Algorithm

Input: comb[1...k] (sorted), n, k
Output: rank
rank $\leftarrow 0$, prev $\leftarrow -1$
for $i \leftarrow 1$ **to** k **do**
  for $j \leftarrow$ prev+1 **to** comb[i]-1 **do**
    rank $\leftarrow$ rank + $C(n-j, k-i)$
  end for
  prev $\leftarrow$ comb[i]
end for
return rank

Algorithm 2 Combinatorial Unranking Algorithm

Input: r, n, k $(0 \le r < C(n,k))$
Output: comb[1...k]
$r' \leftarrow r$, $x \leftarrow 0$
for $i \leftarrow 1$ **to** k **do**
  while $C(n-x, k-i) \le r'$ **do**
    $r' \leftarrow r' - C(n-x, k-i)$
    $x \leftarrow x+1$
  end while
  comb[i] $\leftarrow x$, $x \leftarrow x+1$
end for
return comb

Algorithm 1 systematically computes the lexicographical rank of a given occupation configuration, while Algorithm 2 reconstructs the occupation configuration from its rank. This approach avoids explicit enumeration, making the storage of configurations highly compact.

3.4.2. Mixed radix system

Building on the combinatorial encoding scheme, the mixed radix numeral system, a non-standard positional numeral system where each digit can have a different base, further generalizes state representation to efficiently handle multiple orbital groups. Mixed-radix notation is analogous to writing time as weeks:days:hours:minutes:seconds. In the NOOC algorithm, we utilize the mixed radix system to encode Fock states of multiple orbital groups into a unique integer.

Consider the NOOC constraint $6^{2+}6^{2-}$, where 6^{2+} means there are at most two holes, while 6^{2-} means there exit at most two electrons occupying the six orbitals. The system is divided into two orbital groups, and each group contains 6 orbitals with totaling 12 orbitals. Specifically:

- First group: 6 orbitals with 6^{2+} constraint, i.e., relative to the fully occupied state (6 electrons), up to 2 holes are allowed. Possible configurations include:
  - No holes: $C(6, 0) = 1$ (6 electrons);
  - 1 hole: $C(6, 1) = 6$ (5 electrons, e.g., $|1, 1, 1, 0, 1, 1\rangle$);
  - 2 holes: $C(6, 2) = 15$ (4 electrons).
- Second group: 6 orbitals with 6^{2-} constraint, i.e., relative to the empty state (0 electrons), up to 2 electrons are allowed. Possible configurations include:
  - No electrons: $C(6, 0) = 1$ (0 electrons);
  - 1 electron: $C(6, 1) = 6$ (1 electron);
  - 2 electrons: $C(6, 2) = 15$ (2 electrons, e.g., $|0, 0, 1, 0, 0, 1\rangle$).

Based on their occupation numbers and indices, each orbital group's configurations can be independently encoded. Let A and B be the ranks of the first and second groups, respectively. Then, using the mixed radix system, the entire Fock state's encoding n can be defined as:

$$n = A \times C(6, 2) + B = A \times 15 + B,$$

where $C(6, 2) = 15$ is the number of configurations with 2 electrons in the second group. Thus, n is a unique integer representing the position of the combined Fock state of the first and second groups in lexicographical order. For example, if $A = 3$, and $B = 14$, then $n = 3 \times 15 + 14 = 59$.

This encoding method maps complex Fock states under NOOC constraints to integers, facilitating storage and retrieval, especially for computational tasks that require traversing all possible configurations.

3.4.3. Many-body wave function search

By exploiting physical symmetries, such as U(1) symmetry corresponding to particle-number conservation, the encoding techniques described above enable efficient construction and traversal of the many-body Hilbert space.

For a quantum impurity system, it typically possesses particle number conservation symmetry, i.e., the particle number operator $\hat{N}$ commutes with the Hamiltonian $\hat{H}$: $[\hat{N}, \hat{H}] = 0$. Therefore, the Hilbert space can be decomposed into distinct subspaces $\mathcal{H}_N$ with different particle numbers. In practical calculations, we search for many-body wave functions within a specific subspace $\mathcal{H}_N$.

Take a system with 8 spatial orbitals (16 orbitals) as an example, considering the NOOC constraint $6^{2+}4^{*}6^{2-}$, with orbitals allocated as follows:

- First group: 6 orbitals with 6^{2+} constraint (up to 2 holes);
- Second group: 4 orbitals with 4^{*} constraint (no occupancy constraints), composed of the 2 impurity orbitals and 2 bath orbitals for this example;
- Third group: 6 orbitals with 6^{2-} constraint (up to 2 electrons).

Then the total number of orbitals is $6 + 4 + 6 = 16$.

In the half-filling (particle number $N = 8$) subspace $\mathcal{H}_8$, we use NOOC constraints and particle number conservation to determine the distribution of many-body wave functions:

- First group 6^{2+}: Electron number N_1 can be 4, 5, 6 (corresponding to 2, 1, 0 holes), with configuration numbers $C(6, 4)$, $C(6, 5)$, $C(6, 6)$, i.e., 15, 6, 1.
- Second group 4^{*}: Electron number $N_2 = N - N_1 - N_3 = 8 - N_1 - N_3$, ranging from 0 to 4, with configuration number $C(4, N_2)$.
- Third group 6^{2-}: Electron number N_3 can be 0, 1, 2 (corresponding to 0, 1, 2 electrons), with configuration numbers $C(6, 0)$, $C(6, 1)$, $C(6, 2)$, i.e., 1, 6, 15.

By traversing all possible combinations of N_1 and N_3, we obtain N_2, and then check if it is between 0 and 4. For example, if $N_1 = 5$ (1 hole, configuration number $C(6, 5) = 6$) and $N_3 = 1$ (1 electron, configuration number $C(6, 1) = 6$), then $N_2 = 8-5-1 = 2$, with configuration number $C(4, 2) = 6$. For this specific distribution of electrons, the total number of configurations is given by the product of those from each group, yielding $6 \times 6 \times 6 = 216$.

In summary, by combining combinatorial ranking and mixed radix encoding schemes, we significantly reduce the computational resources required for encoding and traversing the many-body Hilbert space, thereby enhancing the overall efficiency of the solver.

3.5. Rapid generation of many-body operators

For computationally feasible treatments of quantum impurity problems, the rapid and efficient generation of many-body operators is crucial, especially in systems characterized by large and sparse Hamiltonian matrices. In this section, we present the key techniques employed in `iNORG` to address these computational challenges.

3.5.1. Hamiltonian construction

In practical calculations, the dimension of the Hilbert space can be very large (e.g., up to 160 million dimensions), making it impractical to directly construct and store the complete Hamiltonian matrix $H_{mn} = \langle m|\hat{H}|n\rangle$ (where $\{|n\rangle\}$ is the Fock state basis). To address the storage limitations and matrix sparsity, we adopt the following efficient method:

We first apply $\hat{H}$ to a reference Fock state $|n\rangle$ and enumerate all states $|m\rangle$ for which $\langle m|\hat{H}|n\rangle \neq 0$. For a Hamiltonian of the form $\hat{H} = \sum_{ij} t_{ij}\hat{c}_i^\dagger \hat{c}_j + \sum_{ijkl} U_{ijkl}\hat{c}_i^\dagger \hat{c}_j^\dagger \hat{c}_k \hat{c}_l$, the action of these operators on a Fock state reveals all accessible transitions. This approach efficiently pinpoints non-zero elements and their corresponding row and column indices.

We store only the non-zero elements in compressed sparse row (CSR) format [34], whose three-array structure (row pointers, column indices, values)[2]enables fast sparse matrix-vector products required by iterative algorithms such as Lanczos.

3.5.2. NORG-Table data structure

To avoid regenerating the CSR matrix from scratch at each iteration, we introduce the NORG-Table. For every non-zero element $\langle m|\hat{H}|n\rangle$, the table stores one row (m, n, t_{tag}), where t_{tag} is a compact integer that identifies the single-particle operator linking $|n\rangle$ and $|m\rangle$.

We label operators as

$$t_{\text{tag}} = \begin{cases} (i-1)N + (j-1), & \hat{c}_i^\dagger \hat{c}_j \\ N^2 + \{[(i-1)N + (j-1)]N + (k-1)\}N + (l-1), & \hat{c}_i^\dagger \hat{c}_j^\dagger \hat{c}_k \hat{c}_l \end{cases} \tag{19}$$

[2]For clarity, consider an example:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 3 & 0 & 4 \end{bmatrix}$$

Its CSR format consists of three arrays: row pointers (row_ptr): [0, 1, 2, 4], column indices (col_ind): [0, 1, 0, 2], values (values): [1, 2, 3, 4].

ensuring a unique range $0 \le t_{\text{tag}} < N^2 + N^4$. Because the NORG-Table is rebuilt once per sweep and then reused, each matrix element can be looked up in constant time.

During NORG iterations, the Table provides direct access to non-zero matrix elements, avoiding repeated traversals of the Hilbert space. The recorded information can be directly used to construct sparse matrices in CSR format, the standard format for storing the sparse matrix of $\hat{H}$, which enables rapid reconstruction of matrix elements in DMFT iterations, thereby significantly enhancing computational efficiency.

3.6. Ground-state solution based on the Lanczos algorithm

At every NORG sweep, we must determine the ground-state vector of the current effective Hilbert subspace. To achieve this aim efficiently, we adopt the Lanczos method. The resulting pseudocode is given in Algorithm 3.

Algorithm 3 Lanczos ground-state solver

Require: Hamiltonian H, initial normalized state φ_0, maximum Krylov dimension m

Set $\mathbf{w}_0 \leftarrow \varphi_0$
Compute $\mathbf{w}_1 \leftarrow H\,\mathbf{w}_0$
for $i = 1$ **to** m **do**
  Compute $\alpha_{i-1} \leftarrow \langle \mathbf{w}_0,\ \mathbf{w}_1 \rangle$
  if {Lanczos convergence criterion [see (20)]} **then**
    break
  end if
  Update $\mathbf{w}_1 \leftarrow \mathbf{w}_1 - \alpha_{i-1}\,\mathbf{w}_0$
  Compute $\beta_i \leftarrow \|\mathbf{w}_1\|$
  Normalize $\mathbf{w}_1 \leftarrow \dfrac{\mathbf{w}_1}{\beta_i}$
  Swap $\mathbf{w}_0$ and $\mathbf{w}_1$
  Update $\mathbf{w}_1 \leftarrow H\,\mathbf{w}_0 - \beta_i\,\mathbf{w}_1$
end for
return The sequences $\{\alpha_0, \alpha_1, \ldots, \alpha_{m-1}\}$ and $\{\beta_1, \beta_2, \ldots, \beta_{m-1}\}$

In the above iterative process, to further determine whether the Lanczos iteration has converged, the concept of "residual" is often introduced. Simply put, if we diagonalize the tridiagonal matrix obtained after i iterations (with diagonal elements $\{\alpha_0, \alpha_1, \ldots, \alpha_{i-1}\}$, and off-diagonal elements $\{\beta_1, \beta_2, \ldots, \beta_{i-1}\}$), we can obtain a series of approximate eigenvalues and eigenvectors (commonly referred to as Ritz values and Ritz vectors). Among them, the Ritz vector corresponding to the lowest energy (ground-state) approximation can be denoted as $\boldsymbol{y} = (y_1, y_2, \ldots, y_i)^T$, and the residual norm of this state under the original Hamiltonian [35] reads

$$\|H\,\mathbf{v} - \theta\,\mathbf{v}\|_2 = \beta_i \left| y_i \right|, \tag{20}$$

where θ is the Ritz value corresponding to the approximate state, $\mathbf{v}$ is the representation of the approximate state in the original space, and β_i is the last off-diagonal element obtained in this iteration. Generally, if $\beta_i\,|\,y_i|$ is less than a certain tolerance (e.g., ϵ), namely

$$\beta_i \left| y_i \right| \le \epsilon, \tag{21}$$

it can be considered that the Krylov subspace is sufficient to approximate the ground state.

In the criterion (21), we rely on the product of the last off-diagonal element β_i and the magnitude of the last component of the corresponding Ritz vector $|y_i|$ as the convergence criterion. When the value is sufficiently small, it indicates that the current Krylov subspace contains sufficient information to approximate the ground state, and the convergence is considered to be achieved. This criterion provides reliable convergence control without materially increasing cost, and optional partial re-orthogonalisation can be activated when numerical loss of orthogonality becomes noticeable.

It is emphasized that numerical errors and stability are crucial issues in this algorithm; if not controlled, numerical errors can accumulate. Typically, methods such as Kahan summation (to reduce floating-point error accumulation) and additional orthogonalisation (to prevent loss of orthogonality) are used to control errors [27–30]. A small spectral

shift can place zero energy near the ground-state level, accelerating convergence. If the system has many degenerate eigenstates, block Lanczos algorithms (solving multiple eigenstates simultaneously) or increasing the Krylov dimension may be more appropriate.

3.7. Continued-fraction evaluation of diagonal Green's functions

For quantum many-body problems, the Green's function encodes the spectral weight, elementary excitations, and other dynamical observables. Once the ground state is available, dynamical properties can be obtained from the single-particle Green's function, which we will evaluate efficiently with a Lanczos-based continued-fraction scheme. In this subsection, we will introduce how to use the continued-fraction inversion algorithm to solve the zero-temperature single-particle (retarded) Green's function [36]. We first review the basic formula of the Green's function, then detail how to construct the tridiagonal matrix during the Lanczos iteration process, and finally use the continued-fraction expansion inversion method to solve the Green's function numerically.

The formula for the zero-temperature retarded Green's function is expressed as follows,

$$\begin{aligned} G^{\mathrm{R}}_{ij\sigma}(\omega + i\eta) &= G^{>}_{ij\sigma}(\omega + i\eta) + G^{<}_{ij\sigma}(\omega + i\eta) \\ &= \langle\psi_0|c_{i\sigma}\frac{1}{\omega + i\eta + E_0 - H}c^{\dagger}_{j\sigma}|\psi_0\rangle + \langle\psi_0|c^{\dagger}_{j\sigma}\frac{1}{\omega + i\eta - E_0 + H}c_{i\sigma}|\psi_0\rangle \end{aligned} \tag{22}$$

where $\eta > 0$ is an infinitesimal broadening factor. In practice, direct inversion or diagonalization is infeasible, due to the typically enormous dimensionality of the Hamiltonian H. Therefore, Lanczos iteration constructs a tridiagonal representation T of H in the Krylov space, thereby transforming the inversion problem into a finite-dimensional subspace. This is exactly the Lanczos continued-fraction method for computing the Green's function. Taking

$$G^{<}(z) = \langle\psi_0|c^{\dagger}\frac{1}{z + H - E_0}c|\psi_0\rangle \tag{23}$$

as an example, we detail its computational implementation below.

We first define the initial state as $|f_0\rangle = c|\psi_0\rangle$, and introduce two sets of Krylov complete bases $\{|n_i\rangle\}$ and $\{|n_j\rangle\}$, then the Green's function $G^{<}(z)$ can be written as

$$\begin{aligned} G^{<}(z) &= \sum_{i,j}\langle f_0|n_i\rangle\ \langle n_i|\ \frac{1}{z + H - E_0}\ |n_j\rangle\ \langle n_j|f_0\rangle \\ &= \sum_{i,j}\delta_{i,0}\ \langle n_i|\ \frac{1}{z + H - E_0}\ |n_j\rangle\ \delta_{j,0} \\ &= \left[\frac{1}{z + T - E_0}\right]_{00}, \end{aligned} \tag{24}$$

where $\delta_{i,0}$ is 1 when $i = 0$ and 0 otherwise, indicating that $|f_0\rangle$ coincides with the first basis vector in the Krylov basis. Here, the subscript "00" denotes the first element of the matrix $\left[(z + T - E_0)^{-1}\right]$. During the Lanczos iteration process, we have the following formulas:

$$\begin{aligned} |n_0\rangle = \frac{c\,|\psi_0\rangle}{\|c\,|\psi_0\rangle\,\|}, \quad \langle n_0| &= \frac{\langle\psi_0|\,c^{\dagger}}{\|c\,|\psi_0\rangle\,\|}, \\ \beta_0 &= 0, \\ \alpha_i &= \langle n_i|\,H\,|n_i\rangle\,, \\ \beta_{i+1}\,|n_{i+1}\rangle &= (H - \alpha_i)\,|n_i\rangle - \beta_i\,|n_{i-1}\rangle\,. \end{aligned}$$

Specifically, the diagonal elements of T are $\{\alpha_i\}$, and the upper (lower) off-diagonal elements are $\{\beta_{i+1}\}$. Using Algorithm 3, we can project the operator H into the corresponding Krylov space to obtain the tridiagonal matrix

T with

$$T = \begin{pmatrix} \alpha_0 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_1 & \beta_2 & \cdots & 0 \\ 0 & \beta_2 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \beta_{N-1} \\ 0 & 0 & 0 & \beta_{N-1} & \alpha_{N-1} \end{pmatrix}. \tag{25}$$

Because T is tridiagonal, this matrix element is expressible as an infinite continued fraction. Using the continued-fraction method [37], we can compute $\left[(z + T - E_0)^{-1}\right]_{00}$, i.e., the value of the first element in the upper-left corner of the inverse matrix, expressed in continued-fraction form as:

$$\begin{aligned} G^<(z) &= \left[(z + T - E_0)^{-1}\right]_{00} \\ &= \cfrac{1}{z - E_0 + \alpha_0 \; - \; \cfrac{\beta_1^2}{z - E_0 + \alpha_1 \; - \; \cfrac{\beta_2^2}{z - E_0 + \alpha_2 \; - \; \ldots}}}, \end{aligned} \tag{26}$$

Similarly, we have:

$$\begin{aligned} G^>(z) &= \left[(z - T + E_0)^{-1}\right]_{00} \\ &= \cfrac{1}{z + E_0 - \alpha_0 \; - \; \cfrac{\beta_1^2}{z + E_0 - \alpha_1 \; - \; \cfrac{\beta_2^2}{z + E_0 - \alpha_2 \; - \; \ldots}}}, \end{aligned} \tag{27}$$

By appropriately truncating at a finite order (e.g., continue until $|\beta_N| < \epsilon$ or up to a prescribed Krylov dimension N), we can obtain a numerical approximation to $G^<(z)$.

Thus, by projecting H onto a Lanczos basis, and evaluating the first diagonal element of $(z + T - E_0)^{-1}$ through its continued fraction, it yields the zero-temperature single-particle Green's function with $O(N)$ memory and $O(N^2)$ time.

3.8. Krylov-subspace evaluation of matrix-valued Green's functions

While the continued-fraction Lanczos technique is well suited for a single vector $\langle v|(z-H)^{-1}|v\rangle$, however, it cannot deliver generic matrix elements $\langle u|(z - H)^{-1}|v\rangle$. Therefore, we need to develop algorithms that build a Krylov space capable of treating multiple source and sink states simultaneously. Two practical remedies are widely used: one is the "correction vector" algorithm, and the other is the Green's function algorithm based on the Krylov subspace.

The correction-vector method rewrites the resolvent $(\omega + i\eta + E_0 - H)^{-1}$ as a linear system, which is solved independently for every frequency point ω. Specifically, a correction vector $|\chi(\omega)\rangle$ is introduced, satisfying

$$(\omega + i\eta + E_0 - H)\,|\chi(\omega)\rangle \;=\; A\,|\psi_0\rangle, \tag{28}$$

where A is the corresponding annihilation/creation operator, $|\psi_0\rangle$ is the system's ground state, and η represents the energy-broadening parameter. After numerically solving for $|\chi(\omega)\rangle$, the Green's function can be obtained from the simple inner product $\langle\psi_0|A^\dagger|\chi(\omega)\rangle$. This method allows direct computation in the frequency domain without Fourier transformation, and it is often applied to study dynamical properties of strongly correlated electron systems employing many-body numerical methods, such as the DMRG method[38–40]. However, the cost scales as $\propto N_\mathrm{K} \cdot N_\omega$, where N_K is the Krylov dimension and N_ω the number of sampled frequencies.

Since the correction-vector method requires multiple solutions in practice, which will lead to a large computational load, it is generally not ideal for large-scale strongly correlated models. We now outline the procedure: for the

Krylov-subspace approach, which can uniformly solve the entire matrix Green's function, greatly reducing redundant computations and thus offering better scalability. We define

$$z \equiv E_0 + \omega + i\eta, \qquad \eta > 0,$$

and consider the zero-temperature Green's function

$$\begin{aligned} G^{>}(\omega) &= \langle \phi' | (z - H)^{-1} | \phi \rangle \\ &= \langle \phi' | V^{\dagger} (z - T)^{-1} V | \phi \rangle \\ &= \langle \phi' | V^{\dagger} S^{\dagger} (z - D)^{-1} S V | \phi \rangle \\ &= \langle n | (z - D)^{-1} | m \rangle, \end{aligned} \tag{29}$$

where

$$|m\rangle = S V |\phi\rangle, \qquad \langle n| = \langle \phi' | V^{\dagger} S^{\dagger}.$$

We define $M(z) \equiv S^{\dagger}(z - T)^{-1} S$. The overall computational process can be divided into three numbered steps:

1. Lanczos projection - build V, compute $T = V^{\dagger} H V$.
2. Diagonalisation - find $S^{\dagger} T S = D$.
3. Assembly - for each complex z evaluate $M(z) = S^{\dagger}(z - D)^{-1} S$ and form matrix elements $G_{\alpha\beta}(\omega) = \langle n_\alpha | M(z) | m_\beta \rangle$.

Due to the fact that storing the full vectors V and the source states is prohibitive, we instead retain the projected quantities $|m\rangle$ and $\langle n|$, whose dimension equals the Krylov size (much smaller than the full Hilbert-space dimension). For an $N \times N$ dimensional Green's function matrix, there are N different single-particle excitation states, each of which needs to be projected onto the Krylov space through the Lanczos process. During the Lanczos process, we can obtain

$$\begin{aligned} |m\rangle &= V |\phi\rangle, \\ \langle n| &= \langle \phi' | V^{\dagger}. \end{aligned}$$

It is important to note that in one Lanczos run we use the same V and $V^{\dagger}$, meaning that it can simultaneously generate one $|m\rangle$ and N $\langle n|$ in one Lanczos run. When computing the Green's function matrix later, we need to perform the inner product calculation $\langle n| M(z) |m\rangle$ to obtain the value of the Green's function.

Once a single Lanczos run has produced V, T and S, it requires only $O(N_{\mathrm{K}} N^2 N_\omega)$ arithmetic to evaluate all N^2 matrix elements for N frequency points, whereas the correction-vector method scales as $O(N_{\mathrm{H}} N N_\omega)$. This method not only reduces the numerical complexity of computing large-scale matrices significantly, but also has important applications in the study of spectral properties of strongly correlated systems.

4. Installation and usage

This section details the steps required to obtain, build, and run the `iNORG` software package.

4.1. Obtaining iNORG

`iNORG` is an open-source software package released under the AGPLv3 license. You can obtain the source code by cloning the public repository (https://github.com/jmw-phys/iNORG):

```
$ git clone https://github.com/jmw-phys/iNORG.git
$ cd iNORG
```

4.2. Prerequisites

Before building iNORG, ensure the following dependencies are installed and configured on your system:

- C++ Compiler: A modern compiler supporting the C++17 standard (e.g., GCC 7+, Clang 6+, Intel C++ Compiler 19+).
- MPI Library: An MPI implementation such as OpenMPI, MPICH, or Intel MPI. The corresponding MPI compiler wrappers (mpicxx or similar) should be available in your PATH.
- Intel Math Kernel Library (MKL): Required for optimized BLAS and LAPACK routines. Ensure the MKL environment variables are set correctly (e.g., via mklvars.sh).

4.3. Building iNORG

Follow these steps to compile the iNORG executable:

1. Create a build directory and navigate into it:

```
$ mkdir build
$ cd build
```

2. Configure the build using Make. You might need to specify the path to MKL if it's not found automatically. Replace /path/to/mkl with your MKL installation root.

```
$ make clean  # Optional, but recommended for clean build
$ make
```

Note: Ensure mpicxx points to your desired MPI-enabled C++ compiler.

3. Compile the code using Make:

```
$ make -j N  # Replace N with the number of parallel jobs
```

Upon successful compilation, the executable (e.g., inorg) will typically be located in the build/src or build/bin directory.

4.4. Running iNORG

To run an iNORG calculation:

1. Prepare Input Files: Create the necessary input files in your working directory. This typically includes:
   - A main parameter file (e.g., PARAMS.norg) specifying the model, solver settings, convergence criteria, etc.
   - Files containing the initial hybridization function $\Gamma(i\omega_n)$ if starting a DMFT cycle or using a predefined bath.

   Refer to the documentation or examples for the specific format and required parameters.
2. Set Up Environment (if needed): Ensure the MKL runtime libraries are accessible. This might involve sourcing the mklvars.sh script in your run script or environment:

```
$ source /path/to/mkl/bin/mklvars.sh intel64  # Adjust path and architecture
```

3. Execute: Run the program using mpirun or your system's equivalent MPI launcher. Specify the number of MPI processes (num_procs) and the path to the iNORG executable.

```
$ mpirun -n <num_procs> /path/to/iNORG/build/src/inorg > output.log
```

Redirecting output to a log file (`output.log`) is recommended.

4. Analyze Output: Monitor the log file for convergence information and check the output files generated by `iNORG` (e.g., calculated Green's functions, self-energies, natural orbital occupations).

Important notes:

- The number of MPI processes should be chosen based on your system resources and problem size
- For large systems, it's recommended to first test with a small number of processes
- Check the detailed documentation and example files in the doc/ directory for specific usage scenarios
- The program supports checkpoint/restart functionality for long calculations

For detailed information about input file formats, available parameters, and advanced usage options, please refer to the user manual in the documentation.

5. Examples

This section presents two practical examples demonstrating `iNORG`'s capabilities for solving quantum impurity models. The first example shows a one-shot solution of a multi-orbital impurity problem within DMFT, while the second demonstrates a self-consistent multi-orbital DMFT calculation on a Bethe lattice. Each example includes complete parameter files with detailed explanations and execution instructions.

5.1. Example 1: One-shot solution of multi-orbital impurity problem in DMFT

This example demonstrates solving a three-orbital Anderson impurity model with Coulomb interactions in the context of DMFT. All energy parameters are given in eV units.

Parameter file (`PARAMS.norg`):

```
# Three-orbital Anderson impurity model for DMFT
# All energy parameters in eV units

# Model configuration
mode            1
Ed              [-11.61347089, -11.62251229, -11.63870539]
deg_idx         [1, 2, 3]

# Interaction parameters
U               8.0
J               0.8
CoulombF        'Ising'

# Temperature and fitting
beta            200.0
fit_points      [0, 1, 2, 3, 4, 5, 7, 10, 13, ..., 1088, 2164, 3780]
fit_nbaths      [11, 11, 11, 11, 11, 11]

# NOOC configuration
NOOC            phss_v2
restrain        [0, -2, -3, -4, 0, 4, 3, 2]
pred_gs_deg     2
ful_pcl_sch     1
weight_nooc     [1e-05, 1e-05, 1e-05, 1e-05, 1e-05]
weight_freze    [1e-07, 1e-07, 1e-07, 1e-07, 1e-07]
```

Parameter descriptions:

- `mode`: Solver mode. `1` indicates a one-shot solution of the impurity model.
- `Ed`: On-site energies of the impurity levels (in eV).
- `deg_idx`: Degeneracies corresponding to each impurity level in `Ed`.
- `U`: Intra-orbital Coulomb repulsion parameter $F_0 = 8.0$ eV.
- `J`: Hund's coupling strength $J = 0.8$ eV.
- `CoulombF`: Form of Coulomb interaction. `Ising` uses density-density terms only; `Full` includes rotationally invariant Slater-Condon terms.
- `beta`: Inverse temperature $\beta = 1/(k_B T) = 200$ eV^{-1} (corresponding to $T \approx 0.005$ eV).
- `fit_points`: Discrete Matsubara frequency points for hybridization function fitting, selected using the discrete Lehmann representation [41, 42] with $E_{\text{uv}} = 10$, $\beta = 200$, and relative tolerance 10^{-6}.
- `fit_nbaths`: Number of bath sites for each irreducible representation (six sectors, each with 11 bath sites).
- `NOOC`: Type of Natural-Orbital Occupancy Constraint applied (`phss_v2` scheme, see Section 3.3).
- `restrain`: NOOC distribution parameters among orbital groups. The eight values correspond to different orbital group constraints as detailed in Section 3.3.
- `pred_gs_deg`: Expected ground-state degeneracy (2). If set to 0, degeneracy checking is skipped.
- `ful_pcl_sch`: Enables (1) or disables (0) particle-number subspace screening.
- `weight_nooc`, `weight_freze`: Thresholds for filtering natural-orbital weights in the NOOC procedure.

Execution:

```
$ mpirun -n 4 /path/to/iNORG/build/src/inorg > output_example1.log
```

5.2. Example 2: Multi-orbital DMFT calculation on Bethe lattice

This example demonstrates a self-consistent DMFT calculation for a two-orbital Bethe lattice model with inter-orbital interactions. Energy parameters are in dimensionless units relative to the hopping amplitude.

Parameter file (`PARAMS.norg`):

```
# Two-orbital Bethe lattice DMFT calculation
# Energy parameters in dimensionless units (relative to hopping)

# Model configuration
mode            0
bethe_mu        0.0
bethe_band      2
bethe_t         [0.5, 0.25]

# Bath fitting
fit_nbaths      [7, 7, 7, 7]

# Interaction parameters
bethe_u         3.0
bethe_uprim     2.7

# NOOC configuration
pred_gs_deg     2
NOOC            phss_v2
restrain        [0, -2, -3, -4, 0, 4, 3, 2]
distribute      [1, 0, 1, 2, 1, 2, 1, 0]
```

Parameter descriptions:

Parameters identical to Example 1 (`NOOC`, `restrain`) are not repeated here.

- `mode`: Solver mode. `0` indicates a full self-consistent DMFT loop.
- `bethe_mu`: Chemical potential for the Bethe lattice (set to 0 for half-filling).
- `bethe_band`: Number of orbitals per site (2 for this two-orbital model).
- `bethe_t`: Hopping amplitudes for each orbital. The bandwidth is $4t$ for each orbital.
- `fit_nbaths`: Number of bath sites for each irreducible representation (four sectors with 7 bath sites each).
- `bethe_u`: Intra-orbital Coulomb interaction strength $U = 3.0$.
- `bethe_uprim`: Inter-orbital Coulomb interaction strength $U' = 2.7$.
- `pred_gs_deg`: Expected ground-state degeneracy (2).
- `distribute`: Template defining orbital positions within NOOC groups. The eight values map specific orbitals to the constraint groups defined by `restrain`.

Execution:

```
$ mpirun -n 8 /path/to/iNORG/build/src/inorg > output_example2.log
```

6. Future developments

Future enhancements for `iNORG` are planned across several key areas to expand its capabilities and improve its performance and usability:

1. Advanced features including adaptive NOOC schemes and finite-temperature extensions.
2. Support for two-particle responses and cluster DMFT.
3. Integration with existing DFT+DMFT ecosystems via Python tools.

These developments will solidify `iNORG` as a powerful and versatile tool for cutting-edge research in strongly correlated electron systems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that support the findings of this study will be made available upon reasonable requests to the corresponding author.

Acknowledgement

This work was supported by the National Key R&D Program of China (Grants No. 2024YFA1408601 and No. 2024YFA1408602) and the National Natural Science Foundation of China (Grant No. 12434009). Computational resources were provided by the Physical Laboratory of High Performance Computing in Renmin University of China.

References

[1] Walter Metzner and Dieter Vollhardt. Correlated lattice fermions in $d = \infty$ dimensions. *Phys. Rev. Lett.*, 62:324–327, Jan 1989.

[2] Antoine Georges, Gabriel Kotliar, Werner Krauth, and Marcelo J. Rozenberg. Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions. *Rev. Mod. Phys.*, 68:13–125, Jan 1996.

[3] P. W. Anderson. Localized magnetic states in metals. *Phys. Rev.*, 124:41–53, 1961.

[4] J. Kondo. Resistance minimum in dilute magnetic alloys. *Prog. Theor. Phys.*, 32(1):37–49, 1964.

[5] Alexander Cyril Hewson. *The Kondo problem to heavy fermions*. Cambridge University Press, 1997.

[6] Matthias Vojta. Impurity quantum phase transitions. *Philos. Mag.*, 86(13-14):1807–1846, 2006.

[7] Thomas Maier, Mark Jarrell, Thomas Pruschke, and Matthias H. Hettler. Quantum cluster theories. *Rev. Mod. Phys.*, 77:1027–1080, 2005.

[8] Brigitte Surer, Matthias Troyer, Philipp Werner, Tim O Wehling, Andreas M Läuchli, Aljoscha Wilhelm, and Alexander I Lichtenstein. Multiorbital kondo physics of co in cu hosts. *Phys. Rev. B*, 85(8):085114, 2012.

[9] Reinhard M. Noack and Salvatore R. Manmana. Diagonalization-and numerical renormalization-group-based methods for interacting quantum systems. *AIP Conf. Proc.*, 789(1):93–163, 09 2005.

[10] H.Q. Lin, J.E. Gubernatis, Harvey Gould, and Jan Tobochnik. Exact diagonalization methods for quantum systems. *Comput. Phys.*, 7(4):400–407, 07 1993.

[11] Emanuel Gull, Andrew J Millis, Alexander I Lichtenstein, Alexey N Rubtsov, Matthias Troyer, and Philipp Werner. Continuous-time monte carlo methods for quantum impurity models. *Rev. Mod. Phys.*, 83(2):349–404, 2011.

[12] Kenneth G. Wilson. The renormalization group: Critical phenomena and the kondo problem. *Rev. Mod. Phys.*, 47:773–840, Oct 1975.

[13] ZhenHua Li, NingHua Tong, Xiao Zheng, Dong Hou, JianHua Wei, Jie Hu, and YiJing Yan. Hierarchical liouville-space approach for accurate and universal characterization of quantum impurity systems. *Phys. Rev. Lett.*, 109(26):266403, 2012.

[14] Rong-Qiang He and Zhong-Yi Lu. Quantum renormalization groups based on natural orbitals. *Phys. Rev. B*, 89:085108, Feb 2014.

[15] Jia-Ming Wang, Jing-Xuan Wang, Rong-Qiang He, Li Huang, and Zhong-Yi Lu. Ab initio dynamical mean field theory with natural orbitals renormalization group impurity solver. *npj Computational Materials*, 11:86, 2025.

[16] Per-Olov Löwdin. Quantum theory of many-particle systems. i. physical interpretations by means of density matrices, natural spin-orbitals, and convergence problems in the method of configurational interaction. *Phys. Rev.*, 97(6):1474, 1955.

[17] Charles F Bender and Ernest R Davidson. A natural orbital based energy calculation for helium hydride and lithium hydride. *J. Phys. Chem.*, 70(8):2675–2685, 1966.

[18] John Hubbard. Electron correlations in narrow energy bands. *Proc. R. Soc. London, Ser. A*, 276(1365):238–257, 1963.

[19] Michel Caffarel and Werner Krauth. Exact diagonalization of the anderson model on the bethe lattice. *Phys. Rev. Lett.*, 72(10):1545–1548, 1994.

[20] Jia-Ming Wang, Yin Chen, Yi-Heng Tian, Rong-Qiang He, and Zhong-Yi Lu. Low-energy interband kondo bound states in orbital-selective mott phases. *Phys. Rev. B*, 111:155107, Apr 2025.

[21] Zhenfeng Ouyang, Jia-Ming Wang, Rong-Qiang He, and Zhong-Yi Lu. DFT+DMFT study of correlated electronic structure in the monolayer-trilayer phase of $La_3Ni_2O_7$. *Phys. Rev. B*, 111(12):125111, Mar 2025.

[22] Yin Chen, Yi-Heng Tian, Jia-Ming Wang, Rong-Qiang He, and Zhong-Yi Lu. Non-Fermi liquid and antiferromagnetic correlations with hole doping in the bilayer two-orbital Hubbard model of $La_3Ni_2O_7$ at zero temperature. *Phys. Rev. B*, 110(23):235119, Dec 2024.

[23] Yi-Heng Tian, Yin Chen, Jia-Ming Wang, Rong-Qiang He, and Zhong-Yi Lu. Correlation effects and concomitant two-orbital $s\pm$-wave superconductivity in $La_3Ni_2O_7$ under high pressure. *Phys. Rev. B*, 109(16):165154, Apr 2024.

[24] Per-Olov Löwdin and Harrison Shull. Natural orbitals in the quantum theory of two-electron systems. *Phys. Rev.*, 101:1730–1739, Mar 1956.

[25] Alexander Weiße, Gerhard Wellein, Andreas Alvermann, and Holger Fehske. The kernel polynomial method. *Rev. Mod. Phys.*, 78:275–306, Mar 2006.

[26] J. Jaklič and P. Prelovšek. Lanczos method for the calculation of finite-temperature quantities in correlated systems. *Phys. Rev. B*, 49:5065–5068, Feb 1994.

[27] Kesheng Wu, Andrew Canning, HD Simon, and L-W Wang. Thick-restart lanczos method for electronic structure calculations. *J. Comput. Phys.*, 154(1):156–173, 1999.

[28] S Sundar and BK Bhagavan. Generalized eigenvalue problems: Lanczos algorithm with a recursive partitioning method. *Comput. Math. Appl.*, 39(7-8):211–224, 2000.

[29] Kesheng Wu and Horst Simon. Thick-restart lanczos method for large symmetric eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 22(2):602–616, 2000.

[30] Effrosyni Kokiopoulou, Constantine Bekas, and Efstratios Gallopoulos. Computing smallest singular triplets with implicitly restarted lanczos bidiagonalization. *Appl. Numer. Math.*, 49(1):39–61, 2004.

[31] Jia-Ming Wang. iNORG: An open source quantum impurity solver based on the natural orbitals renormalization group, 2025. GitHub repository.

[32] Xiaofen Wang, Peng Wang, Xiaotong Zhang, Yadong Wan, Wen Liu, and Haodong Shi. Efficient and robust Levenberg–Marquardt algorithm based on damping parameters for parameter inversion in underground metal target detection. *Computers & Geosciences*, 176:105354, 2023.

[33] Erik Koch, Giorgio Sangiovanni, and Olle Gunnarsson. Sum rules and bath parametrization for quantum cluster theories. *Phys. Rev. B*, 78:115102, Sep 2008.

[34] Intel Corporation. OneMKL Developer Reference: Sparse Storage Formats, 2024. Accessed: 2025-02-18.

[35] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU Press, 2013.

[36] E. R. Gagliano and C. A. Balseiro. Dynamical properties of quantum many-body systems at zero temperature. *Phys. Rev. Lett.*, 59:2999–3002, Dec 1987.

[37] R Haydock, V Heine, and MJ Kelly. Electronic structure based on the local atomic environment for tight-binding bands. ii. *J. Phys. C: Solid State Phys.*, 8(16):2591, 1975.

[38] Till D. Kühner and Steven R. White. Dynamical correlation functions using the density matrix renormalization group. *Phys. Rev. B*, 60:335–343, Jul 1999.

[39] Robert Peters. Spectral functions for single- and multi-impurity models using density matrix renormalization group. *Phys. Rev. B*, 84:075139, Aug 2011.

[40] P. E. Dargel, A. Wöllert, A. Honecker, I. P. McCulloch, U. Schollwöck, and T. Pruschke. Lanczos algorithm with matrix product states for dynamical correlation functions. *Phys. Rev. B*, 85:205119, May 2012.

[41] Jason Kaye, Kun Chen, and Olivier Parcollet. Discrete lehmann representation of imaginary time green's functions. *Phys. Rev. B*, 105:235115, Jun 2022.

[42] Jason Kaye, Kun Chen, and Hugo U.R. Strand. libdlr: Efficient imaginary time calculations using the discrete lehmann representation. *Computer Physics Communications*, 280:108458, 2022.