

w2dynamics: Local one- and two-particle quantities from dynamical mean field theory[☆]



Markus Wallerberger^{a,b,*}, Andreas Hausoel^c, Patrik Gunacker^a, Alexander Kowalski^c, Nicolaus Parragh^c, Florian Goth^c, Karsten Held^a, Giorgio Sangiovanni^c

^a Institute of Solid State Physics, TU Wien, 1040 Vienna, Austria

^b Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA

^c Institut für Theoretische Physik und Astrophysik, Universität Würzburg, D-97074 Würzburg, Germany

ARTICLE INFO

Article history:

Received 2 May 2018

Received in revised form 1 September 2018

Accepted 9 September 2018

Available online 22 September 2018

Keywords:

(continuous-time) quantum Monte Carlo

Anderson impurity model

Dynamical mean field theory

Green's functions

ABSTRACT

We describe the hybridization-expansion continuous-time quantum Monte Carlo code package “w2dynamics”, developed in Wien and Würzburg. We discuss the main features of this multi-orbital quantum impurity solver for the Anderson impurity model, dynamical mean field theory as well as its coupling to density functional theory. The w2dynamics package allows for calculating one- and two-particle quantities; it includes worm and further novel sampling schemes. Details about its download, installation, functioning and the relevant parameters are provided.

Program summary

Program title: w2dynamics

Program Files doi: <http://dx.doi.org/10.17632/rjs7kmjm2b.1>

Licensing provisions: GNU General Public License v3

Programming language: Python, Fortran 90, and C++11

Required dependencies: cmake ($\geq 2.8.5$), MPI, LAPACK, FFTW3, Python (≥ 2.4)

Optional dependencies: NFFT, pip, numpy (≥ 1.4), scipy (≥ 0.10), h5py, mpi4py, configobj

Nature of problem: Numerically unbiased solutions of one- and two-particle propagators for quantum impurity models at finite temperature. Approximate solutions for general lattice models with strong electronic correlation.

Solution method: Continuous-time quantum Monte Carlo in the hybridization expansion, including worm sampling, for the impurity problem. Dynamical mean field theory solver for the lattice problem.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Strongly correlated electron systems exhibit a range of fascinating phenomena such as the Mott–Hubbard metal-to-insulator transition, spin- and charge-density wave states, giant magnetoresistance, and heavy fermion behavior. However even a numerical solution of correlated lattice models, not to speak of realistic materials calculations, is impossible in practice, except for very few lattice sites, because the Fock space scales exponentially with the number of lattice sites and orbitals. The single impurity Anderson

model (SIAM) [1] is the rare exception of a correlated electron model that is non-trivial while a numerical solution is still feasible. For a single (or a few) orbital(s) this can be done by the numerical renormalization group (NRG) [2], matrix product states (MPS) [3–5], or, if the conduction electron bath is discretized by a few sites, by the exact diagonalization. When more orbitals are taken into account or for calculating general two-particle Green's functions with three independent frequencies, continuous-time quantum Monte Carlo (CT-QMC) approaches [6,7] are the method of choice.

The core of the w2dynamics package is a CT-QMC calculation of n -particle Green's functions for a generalized SIAM using the hybridization expansion (CT-HYB) [8]; closely related codes are ALPS [9,10], TRIQS [11], and EDMFTF [12]. The CT-HYB approach expands the partition function in terms of the bath hybridization and stochastically samples the resulting determinants of the diagrammatic series. We employ the matrix-vector technique for the local time evolution [7]. Furthermore worm sampling is used

[☆] This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding author at: Department of Physics, University of Michigan, Ann Arbor, MI 48109, USA.

E-mail addresses: mwallerb@umich.edu (M. Wallerberger), sangiovanni@physik.uni-wuerzburg.de (G. Sangiovanni).

¹ User mailing list: w2dynamics-users@list.tuwien.ac.at.

to compute most general two-particle Green's functions [13] and improved estimators are used for the self-energy [14] which helps to reduce the CT-HYB's signal-to-noise ratio which is particularly problematic at high frequencies [14].

The CT-HYB algorithm of w2dynamics might serve directly to study quantum dots connected to non-interacting leads or magnetic impurities in weakly interacting solids. Dynamical mean field theory (DMFT) [15], on the other hand, maps Hubbard-like lattice Hamiltonians onto the self-consistent solution of a SIAM. Indeed a major application of w2dynamics is the solution of lattice Hamiltonians within DMFT. These might be model systems or, for materials calculations, Hamiltonians derived from *ab initio* calculations such as density functional theory (DFT) [16,17] or the GW approximation [18].

The most widely used approach of the latter kind is DFT+DMFT [19–22], other approaches include GW+DMFT [23–25] and more general embedding schemes [26,27]. All these methods first attempt to identify the correlated subspace, map that space onto an effective model Hamiltonian by estimating the model parameters, correct for double counted contributions already included in the *ab-initio* calculation, and finally find an (approximate) solution for this *ab initio*-derived model Hamiltonian. The last step is provided by w2dynamics which includes an interface to wannier90 [28] generated Hamiltonians, obtained e.g. from Wien2k [29] through the wien2wannier interface [30]. Depending on the method employed, the w2dynamics results of the subspace might need to be embedded again into the full *ab-initio* space, and one might (optionally) iterate the whole procedure until self-consistency [31–33].

While DMFT is restricted to site-local correlations, post-DMFT methods aim at including non-local correlations on top of DMFT. One route to this end is cluster extensions [34] of DMFT which consider a cluster of sites within a DMFT-like bath. In principle, w2dynamics allows for such calculations by considering the whole cluster as the impurity, but with CT-HYB one is faced with a severe sign problem already at moderate cluster sizes. Other CT-QMC methods are better in this case, but are restricted to single orbitals and/or density–density-like interactions [7].

Another route is diagrammatic extensions of DMFT [35–44]. These require the calculation of the local two-particle vertex which depends on three frequencies and can be calculated by w2dynamics; simplified variants [42] are based on three-leg (two-frequency) vertices while at a higher order in the expansion around DMFT (or for an error estimate) also three-particle vertices are required [45]. One of the strong points of w2dynamics is the calculation of general multi-orbital two-particle Green's functions, which are needed to obtain the local vertices of post-DMFT approaches and for calculating DMFT susceptibilities. This is based on an intensive code and algorithmic development, including worm sampling [13], improved estimators [14], and vertex asymptotics [46]; w2dynamics can also be employed to calculate selected three-particle Green's functions [45]. From the local vertex calculated by w2dynamics, the diagrammatic extensions of DMFT determine in turn non-local vertices and self-energies, using ladder or parquet diagrams [44].

The outline of the paper is as follows: A description of the code is provided in Sections 2 and 3. Here, Section 2 provides an overview of the Python scripts for doing self-consistent DMFT calculations of various types, which wrap around the CT-HYB code. Section 3 in turn outlines the CT-HYB code for solving the SIAM. This includes the Monte Carlo steps of e.g. inserting and removing hybridization lines in Section 3.1, routines for estimating physical observables in Section 3.2, and the calculation of the local and bath weight in Sections 3.4 and 3.6, respectively. Section 4 provides a brief description on how to install and run the code; and Section 5 describes which parameters the user can (has to) set and the format of the hdf5 data output. These latter two user-oriented sections only provide a very first introduction; for further details we refer

the reader to the online tutorial and wiki (cf. Section 4.1). Finally, Section 6 provides a conclusion.

2. Dynamical mean field theory (DMFT) solver

Before turning to the w2dynamics solution of a SIAM, let us discuss the Python scripts that allow for an outer DMFT loop. This DMFT treatment allows us to deal with Hubbard-like lattice model, which in full generality can be written as follows:

$$\hat{\mathbf{H}}_{\text{int}} = \sum_{R,R'} \sum_{I,J} \sum_{\mu\nu} \hat{\mathbf{H}}_{\mu,\nu}^{(IJ)}(R,R') \hat{\mathbf{c}}_{\mu}^{\dagger(I)}(R) \hat{\mathbf{c}}_{\nu}^{(J)}(R') + \frac{1}{2} \sum_R \sum_I \sum_{\kappa\lambda,\mu\nu} U_{\kappa\lambda,\mu\nu}^{(I)} \hat{\mathbf{c}}_{\kappa}^{\dagger(I)}(R) \hat{\mathbf{c}}_{\lambda}^{\dagger(I)}(R) \hat{\mathbf{c}}_{\nu}^{(I)}(R) \hat{\mathbf{c}}_{\mu}^{(I)}(R). \quad (1)$$

Here, $\hat{\mathbf{c}}_{\mu}^{(I)}(R)$ annihilates a fermion, R denotes the coordinates of a unit cell or supercell, I denotes the atom index within a supercell, and μ runs over spin-orbitals. $\hat{\mathbf{H}}$ parameterizes the quadratic or non-interacting term, while U parameterizes the interaction, which is assumed to be local for each atom. With bold typeface we indicate full matrices in the basis of all “correlated” and “ligand” spin-orbitals (included in the low-energy subspace) of all atoms of the unit cell, while non-bold-typeface tensors are restricted to the “correlated” space only as e.g. the Coulomb interaction. The small hat-symbol indicates that the quantity is a vector or a matrix, even if restricted to one of the atoms of the unit cell. The corresponding basis is defined by all spin-orbitals of the low-energy subspace associated to that atom.

The DMFT code is implemented in Python, since the challenges there are mostly conceptual, and Python allows for a more compact and clearer implementation. Most of the numerical problems are standard matrix operations as well as fast Fourier transforms, which can be deferred to the numpy package at essentially no loss of computing speed. For the CT-QMC solver (see Section 3), this is not a viable strategy. Instead, it has been implemented in Fortran 90, and the interface is mediated by the f2py interface generator included with numpy.

The different parts of the DMFT loop were carefully modularized in order to allow for their effective unit testing and re-use in different codes. A high-level UML class diagram is shown in Fig. 1, which is centered around the “driver class” DMFTStep. In the following, we will explain the working of the code by going through the different modules in the code.

Lattice. This module defines an abstract class Lattice for the computation of the local Green's function $\hat{\mathbf{G}}_{\text{loc}}(i\nu)$.

With bold typeface we indicate again all quantities within the full basis of all “correlated” and “ligand” spin-orbitals (included in the low-energy subspace) of all atoms of the unit cell. The local Green's function is calculated from the k -integrated Dyson equation for the lattice model

$$\hat{\mathbf{G}}_{\text{loc}}(i\nu) = \frac{1}{N_k} \sum_k \left[(i\nu + \mu) \hat{\mathbf{1}} - \hat{\mathbf{H}}(k) - \hat{\Sigma}(i\nu) - \hat{\Sigma}_{\text{DC}} \right]^{-1}. \quad (2)$$

This is the definition used by setting DOS=ReadIn, the definition Eq. (2) is directly realized using $\hat{\mathbf{H}}(k)$ (read from HkFile) on the given grid of k -points. With DOS=Bethe the code replaces, for the special case of a Bethe lattice, the k -summation by an integral over a semicircular density of states. DOS=nano gives the possibility of adding frequency-dependent functions describing the (physical) hybridization to non-interacting external leads, see Refs. [47,48].

The code supports calculations at a fixed total filling N_{target} by adjusting the chemical potential to $N(\mu) = N_{\text{target}}$. For a k -discretized Hamiltonian this root finding for μ involves a series of local Green's function computation at different μ . This may

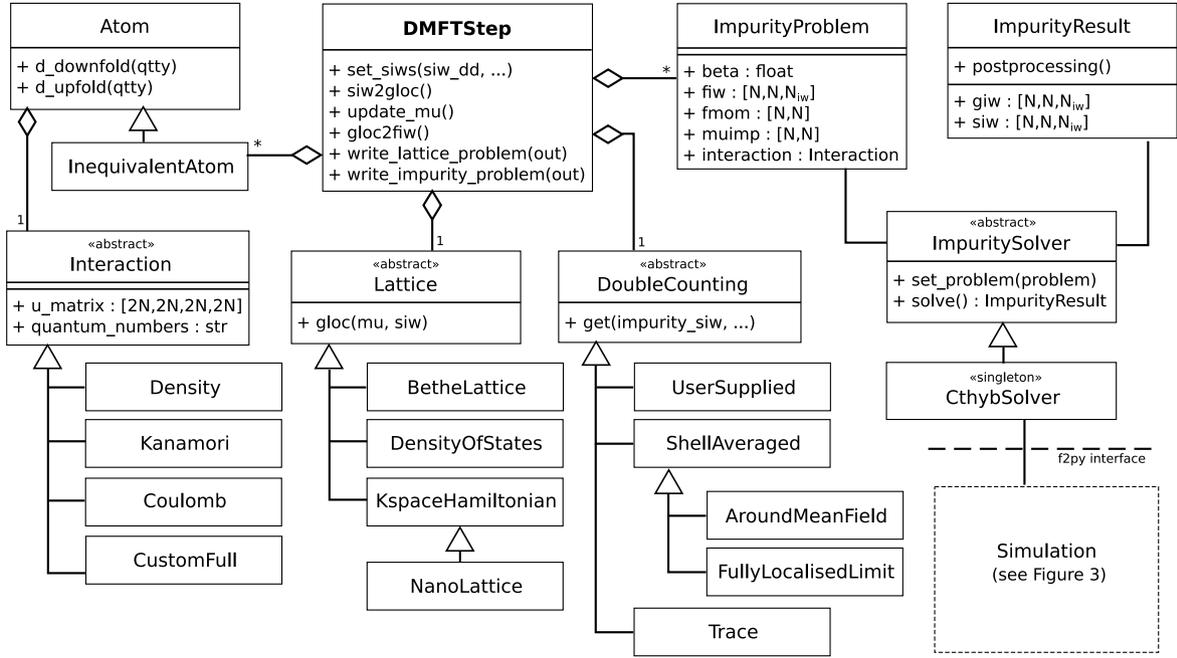


Fig. 1. Simplified unified modeling language (UML) class diagram of the Python structures relevant to the DMFT self-consistency loop. The triangle denotes a subclass relation; the diamond an aggregation, i.e. that one object “has an” other object. All base classes reside in a Python module with the same name in the dmft directory (except for classes beginning with Impurity, which all reside in impurity.py).

easily become the bottleneck of the code for large systems since it scales as $O(N_{\text{fl}}^3 N_k N_{\omega} N_{\text{it}})$, where N_{fl} is the number of spin-orbitals, N_k is the number of k -points, N_{ω} is the number of frequencies, and N_{it} is the number of iterations in the root finder. One can however pre-compute the eigenvalues of $(\hat{H} - \hat{\Sigma})$, store them as $\eta_i(k, i\nu)$, and rewrite the density as follows:

$$N(\mu) = \frac{1}{\beta} \sum_{\nu} e^{-i\nu 0^-} \text{tr} \hat{G}_{\text{loc}}(i\nu) \\ = \frac{1}{\beta N_k} \sum_{k,\nu} \sum_{i=1}^{N_{\text{fl}}} \frac{e^{-i\nu 0^-}}{i\nu + \mu - \eta_i(k, i\nu)}. \quad (3)$$

This is a CPU–memory tradeoff as it reduces the scaling to $O(N_{\text{fl}}^3 N_k N_{\omega}) + O(N_{\text{it}} N_{\text{fl}} N_k N_{\omega})$ while requiring $O(N_{\text{fl}} N_k N_{\omega})$ of memory. If not enough memory is available, the code falls back to the direct strategy.

The lattice code itself is (trivially) parallelized over frequency since instead of $N(\mu)$ the trace over $\hat{G}_{\text{loc}}(i\nu)$ and the corresponding model is computed, which can then be summed at a higher level.

Atom. There are several reasons why realistic calculations of materials often require several “correlated” atoms, $I = 1, \dots, N_{\text{at}}$, in the unit cell: among others magnetic supercells, heterostructures and surfaces. To some extent, the choice of which orbitals belong to the same atom is arbitrary and it lies in the responsibility of the user that the grouping into “atoms” is consistent with the (local) interaction parameters used between the orbitals.

In single-site DMFT, there are only intra-atomic self-energies and no self-energies between different atoms [49–51,47]. $\hat{\Sigma}(i\nu)$ is therefore an atom-block-diagonal matrix in the basis sketched in Fig. 2, whereas $\hat{H}(k)$ contains instead hoppings between atoms as well as intra-atomic local and non-local elements. One can go from the full basis of all N_{fl} “flavors” within the supercell (i.e., atom, spin and orbital flavor; full matrix as sketched in Fig. 2) to the block of a given atom I with N_{fl}^I flavors (spin and orbital; block Atom I in Fig. 2):

$$\hat{G}^{(I)}(i\nu) = D_I \hat{G}_{\text{loc}}(i\nu) D_I^{\text{tr}} \quad (4a)$$

$$\hat{\Sigma}(i\nu) = \sum_{I=1}^{N_{\text{at}}} D_I^{\text{tr}} \hat{\Sigma}^{(I)}(i\nu) D_I, \quad (4b)$$

where D_I is the $N_{\text{fl}}^{(I)} \times N_{\text{fl}}$ matrix that projects (“downfolds”) to the impurity problem of the I th atom with $N_{\text{fl}}^{(I)}$ local degrees of freedom; $\hat{G}^{(I)}$ and $\hat{\Sigma}^{(I)}$ are the corresponding local Green’s function and self-energy matrices, respectively. In our case, D_I just projects out some rows and columns of the lattice problem (Fig. 2). The transposed $N_{\text{fl}} \times N_{\text{fl}}^{(I)}$ matrix D_I^{tr} allows for an extension (“upfolding”) of atom I -based quantities to the full supercell of all atoms through Eq. (4b), and satisfies $D_I D_I^{\text{tr}} = \delta_{IJ} \mathbf{1}^{(I)}$. The up- and downfolding procedure is handled by the Atom class.

What we call here “ p ”- or ligand-orbitals are degrees of freedom that enter the low-energy problem and can hence exchange electrons with the “correlated” subspace of the “ d ”-orbitals. These “ p ”-orbitals are treated as non-interacting, have a vanishing self-energy and do not need to be included in the CT-QMC for the SIAM. The code can however also take care of an explicit Hartree contribution by setting U_{pp} and U_{pd} kind of interactions between the “ p ”-orbitals and between “ p ”- and “ d ”-orbitals, respectively [52–54]. Please note that also the double counting becomes important if both, “ p ”- and “ d ”-orbitals are included; it may also include a contribution from an Hartree-like frequency-independent self-energy associated to these “spectator” degrees of freedom.

The code can take local equivalence of atoms into account to minimize the number of impurity problems that have to be solved (class InequivalentAtom). If the parameter EPSEQ (see Section 5) is set, the code will attempt to detect whether two atoms are equivalent by comparing their G_0^{-1} matrices and the interaction U . The user can either tune EPSEQ or set its own equivalence pattern. Magnetic biases given in the initial configuration (se-shift) as well as different interaction parameters also influence the equivalence pattern.

Both considering the “ p ”-orbitals as non-interacting as well as ignoring correlation between atoms inside a super-cell allows one to work around the exponential scaling of the impurity solver with the number of impurity orbitals. Let us note here, that as a matter of course the user can also include all atoms of the supercell into one w2dynamics “atom”. In this case also correlations and self-energy

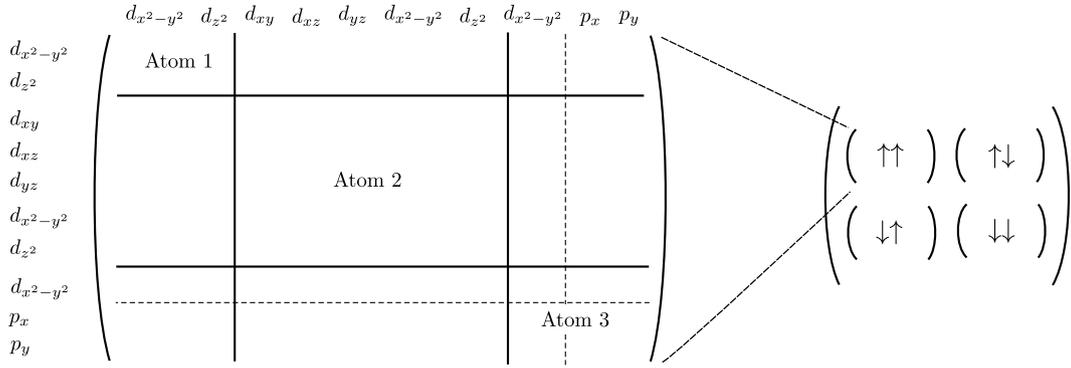


Fig. 2. Example of the atom, spin and orbital structure of the single-particle Hamiltonian $\hat{H}(k)$ (at a fixed k -point), input for DOS=ReadIn. The unit cell of this example contains three atoms, the first one with e_g -orbitals only, the second with the full d -shell and the third with a $x^2 - y^2$ -orbital and two ligands (called here “ p ”). Each element of this Hamiltonian matrix has to have a real and an imaginary part. The only syntax requirement for the DMFT loop to work correctly is that for each atom, the “correlated” orbitals (in this case the d ones) have to come before the “uncorrelated” ones.

Table 1

Available parametrizations of the local interaction. $\hat{c}_{i\sigma}$ annihilates a fermion on the corresponding impurity, $n_{i\sigma} = \hat{c}_{i\sigma}^\dagger \hat{c}_{i\sigma}$ is the corresponding density operator, i, j, k, l are impurity orbitals and σ, σ' denote spins. $N = \sum_i (n_{i\uparrow} + n_{i\downarrow})$ is the total occupation, $S_z = \sum_i (n_{i\uparrow} - n_{i\downarrow})$ the total spin in z direction and PS denotes the pattern of singly-occupied orbitals (see Ref. [55] for a definition). Note that the quantities conserved by the interaction given here may still be broken by the one-particle (non-interacting) part of the local Hamiltonian.

Hamiltonian=	Parameters	$H_{\text{int}} =$	Conserves
Density	Udd, Jdd, Vdd	$H_{\text{dd}} = \sum_i U n_{i\uparrow} n_{i\downarrow} + \sum_{i<j}^{\sigma,\sigma'} (V - J \delta_{\sigma\sigma'}) n_{i\sigma} n_{j\sigma'}$	$N, S_z, n_{i\sigma}$
Kanamori	Udd, Jdd, Vdd	$H_{\text{dd}} - J \sum_{i<j} (\hat{c}_{i\uparrow}^\dagger \hat{c}_{i\downarrow}^\dagger \hat{c}_{j\uparrow} \hat{c}_{j\downarrow} + \hat{c}_{i\uparrow}^\dagger \hat{c}_{i\downarrow}^\dagger \hat{c}_{j\downarrow} \hat{c}_{j\uparrow})$	N, S_z, PS
Coulomb	F0, F2, F4	$\sum_n F_n \dots$	N, S_z, L_z
ReadUmatrix	Umatrix	$\frac{1}{2} \sum_{ijkl} U_{ijkl} \sum_{\sigma\sigma'} \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma'}^\dagger \hat{c}_{i\sigma'} \hat{c}_{k\sigma}$	N, S_z

elements between the physical atoms are included, at the expense of the increasing computational burden.

Local interaction. The code supports different parameterization of the local interaction Hamiltonian, summarized in Table 1.

These can be selected by choosing the parameter with the name Hamiltonian (capable of being misunderstood), in the [Atoms] section. The simplest one is the “density–density” scheme (Density), where only two-body interactions that can be written in terms of orbital occupations appear. The couplings of this kind of interaction are set via the parameters Udd, representing the local Hubbard repulsion, the Hund coupling Jdd and Vdd, which need not fulfill specific relation with the other two parameters. Note that the inclusion of Jdd in the density–density scheme breaks SU(2) symmetry, since spin flip and pair hopping terms cannot be cast in this form. These parameters are used uniformly for all orbitals of the given atom. An orbital-dependent density–density type interaction can be set by using a special ReadNormalUmatrix scheme.

Hamiltonian=Kanamori adds the pair-hopping and spinflip terms to the density–density interaction, thus restoring the SU(2) symmetry [56]. This interaction scheme conserves the so-called “PS”-quantum number [57], which allows for a tighter block structure of the local Hamiltonian and thus efficient sampling.

With Hamiltonian=Coulomb the interaction matrix for a complete d -shell is generated from the Slater parameters F_0, F_2 and F_4 in spherical harmonics, and then transformed to the crystal field basis.

The other possibility is to use ReadNormalUmatrix in which a tensor of four orbital indices is expected as input (umatrix=“filename”). This tensor represents a U_{ijkl} full Coulomb tensor, e.g. coming either from spherical Coulomb tensors written in the basis of the cubic crystal field eigenfunctions or directly from constraint random phase approximation (cRPA). ReadUmatrix reads a tensor with four orbital and four spin indices. Reading in the U_{ijkl} tensor, the user has still the possibility of neglecting elements

with a specific structure, for instance removing all entries that are not of the Kanamori type. This may be useful, for instance, in Kanamori calculations of the full d -shell, for which one however wants to keep the difference between intra- t_{2g} and intra- e_g Hund’s couplings. Let us note that if this is used in combination with “ d - p ” calculations, the user is advised that the double-counting values have to be set by hand rather than relying on the “automatic” calculation by w2dynamics.

For general interactions, the code automatically finds the structure of the local part of the impurity Hamiltonian with the minimal block size by adding All to the QuantumNumber list (cf. Section 3.5) [58].

In the case of density–density type of local interactions, the code will support the addition of a retarded interaction term $W^{(l)}$ by setting the Screening parameter. This modifies the effective impurity action as follows [59]:

$$S^{(l)}[c, c^*] \rightarrow S^{(l)}[c, c^*] + \frac{1}{2} \sum_{\mu\nu} \int d^2\tau c_\mu^*(\tau) c_\nu(\tau) \times W_{\mu\nu}^{(l)}(\tau - \tau') c_\nu^*(\tau') c_\mu(\tau'). \quad (5)$$

Impurityproblem. At each step of the DMFT loop, N_{at} impurity problems are generated from the local Green’s function of Eq. (2). The code calculates, for each atom l , the following Weiss field matrix (cf. Refs. [60,61,22]):

$$(\hat{\mathcal{G}}^{(l)}(i\nu))^{-1} = (D_l \hat{\mathbf{G}}(i\nu) D_l^{-1})^{-1} + \hat{\Sigma}^{(l)}(i\nu), \quad (6)$$

where $\hat{\mathbf{G}}$ again denotes the full local Green’s function of dimension $N_{\text{fl}} \times N_{\text{fl}}$, while $\hat{\Sigma}^{(l)}$ and $\hat{\mathcal{G}}^{(l)}$ denote the self-energy and Weiss field, respectively, of the l ’th atom and are of dimension $N_{\text{fl}}^{(l)} \times N_{\text{fl}}^{(l)}$. These Weiss fields, or the hybridization functions Δ related through Eq. (9), define N_{at} auxiliary single-site quantum many-body impurity problems. Each inequivalent impurity problem is then solved by ImpuritySolver.solve(), yielding the two-point Green’s function

of the l th atom $\hat{G}_{\text{imp}}^{(l)}(i\nu)$. The corresponding self-energy is obtained via the Dyson equation

$$\hat{\Sigma}^{(l)}(i\nu) = (\hat{G}^{(l)}(i\nu))^{-1} - (\hat{G}_{\text{imp}}^{(l)}(i\nu))^{-1} \quad (7)$$

Once the self-energy matrices for all atoms are calculated via Eq. (7), the full matrix $\hat{\Sigma}(i\nu)$ is constructed block-wise and inserted back into Eq. (2). This way ImpurityResult generates a new local Green's function for the whole system. The self-consistency loop then goes on by calculating Eq. (6) for each site again. The user is responsible to decide when convergence is reached. To help the stabilization of the DMFT self-consistency loop we mix the previous and the current self-energies, according to the parameter mixing (a value of 0 means that the solution has no contribution from the previous iteration, whereas 1 means that the self-energy is entirely given by that at the previous iteration).

3. CT-HYB impurity solver

The hybridization expansion continuous-time quantum Monte Carlo (CT-HYB) solver provides an unbiased solution to Anderson impurity models arising e.g. from the self-consistent solution of the DMFT equations. The impurity Hamiltonian, in its full generality, can be written as follows:

$$\begin{aligned} H &= H_{\text{loc}}[\hat{c}, \hat{c}^\dagger] + H_{\text{hyb}}[\hat{c}, \hat{c}^\dagger, \hat{f}, \hat{f}^\dagger] + H_{\text{bath}}[\hat{f}, \hat{f}^\dagger] \\ &= \sum_{\kappa\lambda} E_{\kappa\lambda} \hat{c}_\kappa^\dagger \hat{c}_\lambda + \frac{1}{2} \sum_{\kappa\lambda\mu\nu} U_{\kappa\lambda\mu\nu} \hat{c}_\kappa^\dagger \hat{c}_\lambda^\dagger \hat{c}_\nu \hat{c}_\mu \\ &\quad + \sum_{\kappa p} (V_{p\kappa} \hat{f}_p^\dagger \hat{c}_\kappa + \text{h.c.}) + \sum_p \tilde{E}_p \hat{f}_p^\dagger \hat{f}_p, \end{aligned} \quad (8)$$

where \hat{c}_λ annihilates a fermion of spin-orbital λ on the impurity and \hat{f}_p annihilates a fermion on the bath, where the quantum number p can be continuous. For the l th impurity problem, the interaction tensor is $U = U^{(l)}$, and the quadratic part is given by

$$(\hat{G}^{(l)})_{\lambda\mu}^{-1}(i\nu) = (i\nu + \mu)\delta_{\lambda\mu} - E_{\lambda\mu} - \hat{\Delta}_{\lambda\mu}^{(l)}(i\nu) \quad (9)$$

with the hybridization function

$$\hat{\Delta}_{\lambda\mu}^{(l)}(i\nu) = \sum_p \frac{V_{\lambda p}^* V_{p\mu}}{i\nu - \tilde{E}_p}. \quad (10)$$

CT-HYB expands the partition function Z of that SIAM, defined by H and the inverse temperature β , in terms of the hybridization H_{hyb} (for a derivation, see Ref. [7]):

$$\begin{aligned} Z &= \sum_{k=0}^{\infty} \int_0^\beta \prod_{i=1}^k d\tau_i d\tau'_i \sum_{\lambda_i \lambda'_i} \text{tr} \left[T_\tau e^{-\beta H_{\text{loc}}} \prod_{i=1}^k \hat{c}_{\lambda_i}^\dagger(\tau_i) \hat{c}_{\lambda'_i}(\tau'_i) \right] \\ &\quad \times \det [\Delta_{\lambda_i \lambda'_j}(\tau_i - \tau'_j)]_{ij}. \end{aligned} \quad (11)$$

Here T_τ is the time-ordering operator, H_{loc} the local Hamiltonian of the impurity, cf. Eq. (8) and Δ is the hybridization function, cf. Eq. (10). CT-HYB thus corresponds to the sampling over all diagrams with vertices at $\lambda_i, \lambda'_i, \tau_i, \tau'_i$, and the weight of each diagram is given by the product of a local weight, w_{loc} , and a bath weight, w_{bath} . Here, the local weight or local trace is given by

$$w_{\text{loc}} = \text{Tr} \left[T_\tau e^{-\beta H_{\text{loc}}} \prod_{i=1}^k d_{\lambda_i}^\dagger(\tau_i) d_{\lambda'_i}(\tau'_i) \right], \quad (12)$$

which is represented by LocalTrace in the UML Fig. 3 and will be discussed in detail in Section 3.4. The bath part of the weight $w_{\text{bath}} = \det \Delta$ (BathTrace in Fig. 3) is the determinant of noninteracting hybridization functions, whose calculation will be detailed

in Section 3.6. The Monte Carlo importance sampling of different flavors λ_i , imaginary time positions τ_i and expansion orders k in the sum of Eq. (11) is realized through Steps as outlined in Section 3.1. The Monte Carlo measurement of observables, on the other hand, is performed by Estimator as discussed in Section 3.2.

The CT-QMC solver is written in Fortran 90 in order to account for numerical efficiency when sampling the Monte Carlo configuration space. Since Monte Carlo algorithms are trivially parallelizable, we offset the communication and averaging work to the Python infrastructure. Consequently, the Fortran modules are entirely unaware of any parallelization schemes.

The impurity solver is structured into several layers arranged in order of increasing complexity. On the lowest level this includes matrix operations and the random number generator. The intermediate level includes routines for defining many-body operators and states. These are used in the definition of the local problem and the bath problem. On the highest level, the simulation block governs the Metropolis Hastings Monte Carlo procedure. Monte Carlo steps and estimators are defined for the partition function space and the worm space.

3.1. Steps

The expansion of the thermal expectation value of the partition function (or equivalently, any Green's function like object) results in an infinite series of Feynman diagrams which is sampled through the Metropolis Hastings Monte Carlo algorithm. In the case of CT-HYB this is a series in terms of hybridization inclusions, see Table 2. Thus, the Monte Carlo sampling procedure takes place in the space of all possible Feynman diagrams. In order to fulfill ergodicity and (detailed) balance, it is necessary to define Monte Carlo steps, which increase and decrease the expansion order of the infinite series. We usually relate to these steps as 'add' and 'remove'.

Further Monte Carlo steps allow us to change the observable we sample (worm sampling) or decrease auto-correlation lengths (global moves in partition function space and replacement moves in worm space).

Partition function sampling. When expanding the partition function in the hybridization, the set of possible diagrams depends on the type of local interaction considered. The diagrammatic series built from general local interactions includes Feynman diagrams that are not present in the diagrammatic series built from simplified local density-density interactions only. This is because the imaginary time propagation between creation and annihilation operators in Table 2 may then include e.g. a spin flip in two different orbitals. Yet, in both cases the 'add' and 'remove' steps alter the local weight and the bath weight of the configuration, as discussed in Sections 3.4 and 3.6, respectively.

As we employ the sliding-window approach [62], the distance between operators in pair insertions and removals is limited by default. When the maximum τ -difference is not set with the parameter TaudiffMax, an automatic calibration procedure is used to estimate a good value. The percentage for attempted insertion and removal steps is implicitly determined by setting the percentages for attempting all remaining steps. Fast update formulas are used in the bath determinant [7,63].

While 'add' and 'remove' steps alter the diagrammatic series by changing a small subset of operators only, one can further formulate global moves, which act on all operators of the current configuration. These are computationally more expensive, as they require a complete reevaluation of the configuration weight. In the paramagnetic case, the otherwise explicit symmetrization over spins may be imposed onto the system by proposing to flip the spins of all the operators at the same time. This is done for 10% of the global moves. If certain symmetries of the systems are known

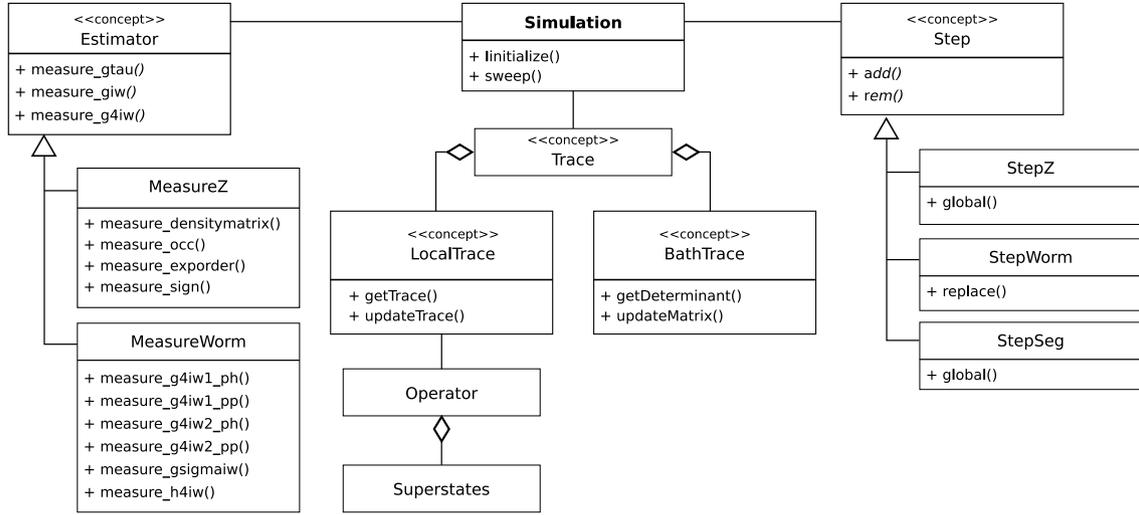


Fig. 3. Simplified UML diagram of the Fortran structures relevant to the CT-HYB solver. `<<concept>>` marks logical units that are technically not represented by Fortran structures.

Table 2

Available updates of a configuration in CT-HYB. The center column is a diagrammatic representation of the update, where the horizontal line represents the imaginary time axis $\tau \in [0, \beta]$, and filled and empty diamonds represent local creation and annihilation operators, respectively. Vertical lines attached to operators indicate the presence of hybridization lines, i.e., operators coming from the hybridization expansion, while the absence of such lines signifies purely local, i.e. worm, operators. R_{prop} denotes a prefactor for the relative probability of the two configurations, which needs to be multiplied to the ratio of the local and bath weights discussed in Sections 3.4 and 3.6, respectively. Here N is the number of impurity spin-orbitals, k the expansion order, and η the balancing parameter between worm and partition function space.

move type	R_{prop}		move type
pair insertion	$\frac{N\beta^2}{(k+1)^2}$		pair removal
quadruplet insertion	$\frac{N^4\beta^4}{(k+2)^4}$		quadruplet removal
pair flavor change	$\frac{k^2 N^2}{k^2 N^2}$		pair flavor change
operator shift	$\frac{k^2\beta}{k^2\beta}$		operator shift
flavour permutation	$\frac{N!}{N!}$		flavour permutation
worm insertion	$\frac{\eta N\beta^2}{1}$		worm removal
worm replacement	$\frac{2k}{2k}$		worm replacement

beforehand, operator permutations may be specified in the parameters file with `SymMove01` to `SymMove99` and an additional parameter `NSymMove` is specifying the number of symmetry moves. E.g. for a 5 orbital system `NSymMove = 1` and `SymMove01 = 7 2 3 4 5 6 1 8 9 10` will propose an exchange of orbital 1 spin-up with orbital 2 spin-down (the first 5 numbers indicate orbitals of spin-up, the last 5 of spin-down). If `NSymMove` $\neq 0$, this is done for 50% of the global moves. Lastly, random permutations of the operator flavors in the trace are performed for the remaining global moves.

Global moves are a way to restore (approximate) orbital- or spin-degeneracies in systems where polarized configurations are separated by a large phase space barrier [64]. They are trivially

accepted if they respect an exact symmetry of the system, and become exponentially suppressed as this symmetry is broken. The total percentage of attempted global moves is specified with the parameter `PercentageGlobalMoves`, which by default is set to 0.5% of total moves.

Worm sampling. Instead of sampling the diagrammatic series of the partition function and extracting an observable by altering individual configurations, one may choose to sample an observable directly. Switching from partition function space (the diagrammatic series given by the denominator of the thermal expectation value, i.e. importance sampling) to worm space (the diagrammatic

series given by the numerator of the thermal expectation value) requires a new set of moves. Primarily, one is interested in extracting Green's function like objects in this fashion, which is why configuration weights of worm spaces only differ from configuration weights of the partition function space by a set of additional local operators (in terms of diagrams these are the external legs). The moves switching between these two spaces are also of the type 'add' or 'remove', but here they only alter the local weight, leaving the bath weight unchanged. The corresponding percentage of attempted inserts and removes is specified with the parameter `PercentageWormInsert`. Within worm space, the sampling with operators connected to the bath is required in order to fulfill ergodicity. Again, the percentage of attempted inserts and removes of such operator pairs is implicitly determined by all remaining steps. While the global moves have not been implemented in worm space, additional replacement steps exist. These attempt to change a specific worm configuration by reattaching hybridization lines from a given operator to the worm operator of the same flavor. The percentage of attempted replacements is specified by the parameter `PercentageWormReplace`.

3.2. Estimators

Once the Monte Carlo sampling of the Feynman diagrammatic series expansion is established, the remaining task is to extract physical observables. This procedure differs when sampling the partition function series and when sampling the observable directly using worm sampling, cf. Ref. [13].

Partition function sampling. The major quantities of interest are the one-particle (imaginary time, Matsubara and Legendre basis) and the two-particle Green's function (Matsubara basis). From the respective Green's functions, the one-particle and two-particle density matrices can be obtained as the all-equal-time components. The histogram denotes the probability distribution of the expansion order k , i.e., the number of pairs in Table 1 (which in the hybridization expansion depends on the kinetic energy). In order to measure an n -particle impurity Green's function in *Partition Function Sampling* mode, all hybridization lines connected to n operators are removed, in order to create n purely local operators that are disconnected from the bath (also coined "hybridization sampling"). In Table 2 this corresponds to taking away the vertical line of n filled creation and n open annihilation operators. Bosonic observables like the local DMFT charge susceptibility $\langle n_{i,\sigma}(\tau)n_{j,\sigma'}(0) \rangle$ or spin susceptibility $\chi_{\text{loc}}(\tau)$ are measured by insertion of density operators. A detailed list of all relevant parameters and the name of the output quantities is given in Table 3.

Worm sampling. The worm algorithm in continuous-time Quantum Monte Carlo algorithms, such as CT-HYB, was introduced in order to allow for more flexibility in terms of the choice of estimators. While partition function sampling restricts the observables to the diagrams present in the infinite series expansion of the partition function, this is not true for worm sampling. Instead, diagrams may be present in the expansion of the observable itself, which are otherwise absent. This is especially true for the two-particle Green's function, when considering a diagonal hybridization function with non-density–density local interactions. Further Green's functions with a subset of the operators occurring at equal times can be easily defined for general local interactions. These are important when extracting the high frequency asymptotics of the fully time-dependent Green's function. A detailed list of all relevant parameters and the name of the output quantities is given in Table 3.

3.3. Statistical uncertainties

Since at its core, CT-HYB is a Monte Carlo procedure, any estimator comes with a statistical uncertainty that drops as $1/\sqrt{N}$ with

the number of Monte Carlo measurements N (parameter `Nmeas`). The prefactor and thus the choice of N for reaching the desired accuracy depends on configuration space, i.e., the type of system under study, as well as the type of estimator used. We state a couple of empirical guidelines to aid users:

1. in partition function space (upper section of Table 3), time-independent estimators such as `rho1` and `rho2` typically have larger variances than Green's function `gtau`, because they are computed from a single point in imaginary time for each diagram.
2. in worm space (lower section of Table 3), estimators with more external indices have larger variances, since the components need to be sampled individually. E. g., `g4iw-worm` is noisier than `g4iw1-worm`, which in turn is noisier than `giw-worm`.

Like any Monte Carlo procedure along a Markov chain, CT-HYB is subject to autocorrelation. This requires each run to be thermalized, i.e., a number of initial Monte Carlo steps N_{th} must be discarded (parameter `Nwarmups`). In order to make sure that the Markov chain "forgets" its starting point, we need $N_{\text{th}} > \tau_{\text{exp}}$, where τ_{exp} is the exponential autocorrelation time [65]. However, τ_{exp} is usually unknown, which is why one commonly sets N_{th} to a fixed fraction, typically 10% to 50% [65,66], of the total number of Monte Carlo steps.

In order to assess some thermalization effects, the observable `gtau-mid-step` is provided, which stores the imaginary-time Green's function averaged over the "center region" in imaginary time, $\tau \in [0.4\beta, 0.6\beta]$, but resolved for each individual Monte Carlo measurement (parameter `Gtau_mid_step=1`). We empirically found that when this quantity reached a plateau in terms of Monte Carlo steps, the calculation was sufficiently thermalized.

Autocorrelation also means that the number of truly uncorrelated samples for each observable is reduced from N to $N/(1 + 2\tau_{\text{int}})$, where τ_{int} is the integrated autocorrelation time [65]. The immediate effect of this is that errorbars, when computed naively over the Markov chain, will be too small. To avoid this, we compute errorbars from the N_{cores} different Monte Carlo runs (given by the N_{cores} CPUs the computation is run on), which are independent provided `Nwarmups` was chosen large enough. Let A_i be the averaged result on the i th core, the error bars on A are estimated as follows:

$$\Delta A^2 = \frac{1}{N_{\text{cores}}(N_{\text{cores}} - 1)} \sum_{i=1}^{N_{\text{cores}}} (A_i - \bar{A})^2. \quad (13)$$

As many measurements in CT-HYB are expensive, one still wants to measure only every $N_{\text{sweep}} > \tau_{\text{int}}$ steps to avoid spending time on measuring cross-correlated results (parameter `NCorr`). A reasonable choice for N_{sweep} is the average "renewal time" of the trace, $\langle k \rangle / R_{\text{rem}}$, where k is the mean expansion order and R_{rem} is the acceptance rate of removing an operator pair [7,63]. Despite these recipes, the actual calculation of the autocorrelation time from the CT-QMC runs would be very useful; doing so for the many different samplings implemented in `w2dynamics` is cumbersome but an important task for the future.

3.4. Local trace

The computation of the local weight (12) relates to the solution and, with each Monte Carlo move, efficient update of an exact diagonalization problem with a Fock space size of 2^{N_n} . Apart from possible conditioning issues of the bath determinant, it is thus easily the computational bottleneck for a CT-HYB calculation, and several strategies have been developed to tackle the problem: matrix–matrix and matrix–vector codes implement the time-evolution of many-body states in the thermal expectation value

Table 3
Measurable quantities in partition function and worm sampling.

Output	Parameter (s)	Estimator
sign	–	$s = \langle \text{sgn } w \rangle_{ w }$
hist	–	histogram of expansion order k
gtau	Ntau=N	$G_{i\sigma}(\tau_k) = +\frac{1}{\beta} \text{rect}\left(\frac{\tau_{12}-\tau_k}{\beta/N}\right) \langle T_\tau c_{i\sigma}(\tau_1) c_{i\sigma}^\dagger(\tau_2) \rangle$
gleg	NLegMax	$G_{i\sigma,l} = \frac{\sqrt{2l+1}}{\beta} \int_0^\beta d\tau P_l\left(\frac{\tau_{12}-\tau}{\beta}\right) \langle T_\tau c_{i\sigma}(\tau_1) c_{i\sigma}^\dagger(\tau_2) \rangle$
giw	NGiw,NLegOrder=L	$G_{i\sigma}(iv) = \sum_{l=0}^L t_l(iv) G_{i\sigma,l}$
giw-meas	MeasGiw=1, NGiw	$G_{i\sigma}(iv) = -\frac{1}{\beta} \int_0^\beta d^2\tau e^{iv\tau_{12}} \langle T_\tau c_{i\sigma}(\tau_1) c_{i\sigma}^\dagger(\tau_2) \rangle$
rho1	MeasDensityMatrix=1	$\rho_{i\sigma,j\sigma'}^{(1)} = \langle T_\tau c_{i\sigma}^\dagger c_{j\sigma'} \rangle$
rho2	MeasDensityMatrix=1	$\rho_{i\sigma,j\sigma',k\sigma'',l\sigma'''}^{(2)} = \langle T_\tau c_{i\sigma}^\dagger c_{j\sigma'}^\dagger c_{k\sigma''} c_{l\sigma'''} \rangle$
sztau-sz0	MeasSusz=1	$\chi_{\text{loc}}(\tau) = g^2 \sum_{ij} \left(S_2^i(\tau) S_2^j(0) \right)$ $= g^2 \sum_{ij} \left(\frac{1}{2} (n_{i,\uparrow}(\tau) - n_{i,\downarrow}(\tau)) \frac{1}{2} (n_{j,\uparrow}(0) - n_{j,\downarrow}(0)) \right)$
ntau-n0	MeasSusz=1	$(n_{i,\sigma}(\tau) n_{j,\sigma'}(0))$
g4iw	FourPnt=4, N4iwf,N4iwb	$G_{i\sigma,j\sigma',k\sigma'',l\sigma'''}^{(\text{ph})}(iv, iv', i\omega) = \int_0^\beta d^4\tau e^{iv\tau_{12}+iv'\tau_{34}+i\omega\tau_{14}}$ $\times \langle T_\tau c_{i\sigma}(\tau_1) c_{j\sigma'}^\dagger(\tau_2) c_{k\sigma''}(\tau_3) c_{l\sigma'''}^\dagger(\tau_4) \rangle$
gtau-worm	WormMeasGtau=1	$G_{i\sigma,j\sigma'}(\tau_k) = +\frac{1}{\beta} \text{rect}\left(\frac{\tau_{12}-\tau_k}{\beta/N}\right) \langle T_\tau c_{i\sigma}(\tau_1) c_{j\sigma'}^\dagger(\tau_2) \rangle_W$
giw-worm	WormMeasGiw=1	$G_{i\sigma,j\sigma'}(iv) = -\frac{1}{\beta} \int_0^\beta d^2\tau e^{iv\tau_{12}} \langle T_\tau c_{i\sigma}(\tau_1) c_{j\sigma'}^\dagger(\tau_2) \rangle_W$
gsigmaiw-worm	WormMeasGSigmaiw=1	$[\Sigma G]_{i\sigma,j\sigma'}(iv) = -\frac{1}{\beta} \sum_{klm} \sum_{\sigma''} U_{[ik]lm} \int_0^\beta d^2\tau e^{iv\tau_{12}}$ $\times \langle T_\tau c_{i\sigma}^\dagger(\tau_1) c_{m\sigma}(\tau_1) c_{l\sigma''}(\tau_1) c_{j\sigma'}^\dagger(\tau_2) \rangle$
g4iw1-worm	WormMeasG4iw1=1 N2iwb	$G_{i\sigma,j\sigma',k\sigma'',l\sigma'''}^{(\text{ph})}(i\omega) = \int_0^\beta d^4\tau e^{i\omega\tau_{14}}$ $\times \langle T_\tau c_{i\sigma}(\tau_1) c_{j\sigma'}^\dagger(\tau_1) c_{k\sigma''}(\tau_4) c_{l\sigma'''}^\dagger(\tau_4) \rangle$
g4iw1pp-worm	WormMeasG4iw1PP=1 N2iwb	$G_{i\sigma,j\sigma',k\sigma'',l\sigma'''}^{(\text{pp})}(i\omega) = \int_0^\beta d^4\tau e^{i\omega\tau_{12}}$ $\times \langle T_\tau c_{i\sigma}(\tau_1) c_{j\sigma'}^\dagger(\tau_2) c_{k\sigma''}(\tau_1) c_{l\sigma'''}^\dagger(\tau_2) \rangle$
g4iw2-worm	WormMeasG4iw2=1 N3iwf,N3iwb	$G_{i\sigma,j\sigma',k\sigma'',l\sigma'''}^{(\text{ph})}(iv, i\omega) = \int_0^\beta d^4\tau e^{iv\tau_{12}+i\omega\tau_{23}}$ $\times \langle T_\tau c_{i\sigma}(\tau_1) c_{j\sigma'}^\dagger(\tau_2) c_{k\sigma''}(\tau_3) c_{l\sigma'''}^\dagger(\tau_3) \rangle$
g4iw2pp-worm	WormMeasG4iw2PP=1 N3iwf,N3iwb	$G_{i\sigma,j\sigma',k\sigma'',l\sigma'''}^{(\text{pp})}(iv, i\omega) = \int_0^\beta d^4\tau e^{iv\tau_{12}+i\omega\tau_{23}}$ $\times \langle T_\tau c_{i\sigma}(\tau_1) c_{j\sigma'}^\dagger(\tau_3) c_{k\sigma''}(\tau_2) c_{l\sigma'''}^\dagger(\tau_3) \rangle$
g4iw-worm	WormMeasG4iw=1 FourPnt=8,N4iwf,N4iwb	$G_{i\sigma,j\sigma',k\sigma'',l\sigma'''}^{(\text{ph})}(iv, iv', i\omega) = \int_0^\beta d^4\tau e^{iv\tau_{12}+iv'\tau_{34}+i\omega\tau_{14}}$ $\times \langle T_\tau c_{i\sigma}(\tau_1) c_{j\sigma'}^\dagger(\tau_2) c_{k\sigma''}(\tau_3) c_{l\sigma'''}^\dagger(\tau_4) \rangle$
h4iw-worm	WormMeasH4iw=1 FourPnt=8,N4iwf,N4iwb	$H_{i\sigma,j\sigma',k\sigma'',l\sigma'''}^{(\text{ph})}(iv, iv', i\omega) = \int_0^\beta d^4\tau e^{iv\tau_{12}+iv'\tau_{34}+i\omega\tau_{14}}$ $\times \sum_{pqr} \sum_{\sigma''''} U_{[ip]qr} \langle T_\tau \hat{c}_{p\sigma''''}^\dagger \hat{c}_{q\sigma} \hat{c}_{r\sigma''''} \hat{c}_{j\sigma'}^\dagger \hat{c}_{k\sigma''} \hat{c}_{l\sigma'''}^\dagger \rangle$

[7]. They are defined for any type of local interaction. The simpler segment representation applicable to density–density interactions is thought to be superior due to a favorable scaling in the number of orbitals and enhanced Monte Carlo acceptance rates. For an introduction into CT-HYB in its aforementioned variants see Ref. [7]. The w2dynamics code features the matrix–vector implementation in the eigenbasis of the local Hamiltonian, combined with the newly developed superstate- and state-sampling methods. There the trace (sum) over impurity eigenstates in Eq. (12) is not performed explicitly, but sampled over in a Monte Carlo procedure [67]. Sampling over them in groups defined by the blocks of H_{loc} (superstate-sampling) is used by default. Sampling each eigenstate individually can be switched on with statesampling=1, which is faster, but may affect the average sign.

Further details on how the local trace is calculated can be found in Ref. [67].

3.5. Quantum numbers

Unless broken by the local non-interacting terms, each type of interaction conserves a set of quantum numbers on the impurity (parameter QuantumNumbers, set by default to the values given in Table 1). In order to be useful in CT-HYB, a quantum number Q not only must commute with H_{loc} , but each annihilation operator \hat{c}_i must map any eigenstate $|q\rangle$ of Q to some other eigenstate $|q'\rangle$. This corresponds to a block-diagonal form of H_{loc} . Since the size of the largest block directly enters the exponential scaling, it is important to use as many quantum numbers of the system as possible.

Any spin-independent quartic interaction conserves the number of electrons and the spin in z -direction in the system

(quantum numbers Nt and Szt). The Coulomb interaction in a spherical basis, additionally, conserves angular momentum in the z -direction (quantum number Lzt). In the case of density–density interaction, the Hamiltonian conserves the number of electron in each spin–orbital (quantum number Azt), i.e. we set for density–density QuantumNumbers = Nt Szt Azt. When, on the other hand, non-density–density interactions like Slater–Kanamori or the full Hubbard terms are present in H_{loc} , the time-evolution mixes those states. The Kanamori interaction is $SO(N) \otimes SU(2)$ symmetric, and has a corresponding conserved quantity (the PS quantum number, in the code called Qzt); thus we have QuantumNumbers = Nt Szt Qzt.

For a general interaction, it is advisable to set QuantumNumbers = Nt Szt All. This choice enables an automatic search for quantum numbers, where the states in the occupation number basis are reordered in such a way that H_{loc} becomes block-diagonal with a minimal block size.

It is worth mentioning that QuantumNumbers in principle can also be used to truncate parts of H_{loc} . This can be used to assess the sensitivity of the result to certain classes of terms in the local Hamiltonian or to pre-converge a DMFT run using a computationally cheaper model. Its use is intended for expert use only and it is advisable not to use it as an *a priori* approximation of the interaction.

Real hopping terms conserving spin in z -direction are necessary to treat short range non-local correlations within cluster-DMFT, and real amplitudes breaking S_z for magnetic fields with an x -component. Complex hopping amplitudes are used to describe spin–orbit coupling or magnetic fields with an y -component. The

real ones are switched on by `offdiag=1`, the complex ones will soon be available and can be switched on by `complex=1`.

3.6. Bath trace

Using an ab-initio Hamiltonian $H(\mathbf{k})$, one usually gets off-diagonal hybridization functions $\Delta(\tau)$ as well as hopping terms $\mu_{\alpha,\beta}^{\text{imp}} c_{\alpha}^{\dagger} c_{\beta}$, which can be allowed by setting the parameter `offdiag=1`. By rotating the single particle basis these offdiagonal terms can be minimized but never removed for general offdiagonal hybridizations.

Density–density interactions and offdiagonal hybridization functions can be combined straightforwardly. For Kanamori and full-Hubbard interaction combined with offdiagonal hybridization functions complications arise, since interactions such as the spin-flip configuration ($c_{2\downarrow}^{\dagger} c_{1\uparrow}^{\dagger} c_{2\uparrow} c_{1\downarrow}$) or the pair-hopping configuration ($c_{2\downarrow}^{\dagger} c_{2\uparrow}^{\dagger} c_{1\uparrow} c_{1\downarrow}$) are contained in the configuration space. Their contribution is zero when the hybridization function is diagonal in the flavors. They cannot be generated by insertions and removals of pairs of operators, but by insertions or removals of 4 operators [68]. Alternatively the spin-flip and pair-hopping configurations could be generated via so called flavorchange-moves, which change the flavor-indices of two operators, e.g. $c_{1\downarrow}^{\dagger} c_{1\uparrow}^{\dagger} c_{1\uparrow} c_{1\downarrow} \rightarrow c_{2\downarrow}^{\dagger} c_{1\uparrow}^{\dagger} c_{2\uparrow} c_{1\downarrow}$. Flavorchange-moves have acceptance rates orders of magnitudes larger than the 4-operator moves, nevertheless we observe more noise with flavorchange-moves compared to 4-operator-moves.

By default 4-operator-moves are used, and switched on with the parameter `Percentage4OperatorMove`.

4. Installation and community site

4.1. Where to find the project and the code

We have decided to put the source code of this tool developed hitherto in Wien and Würzburg (hence the *w*2 in *w2dynamics*) under the GNU GPL license Version 3, an established open source license. This should be the first, but very important step, towards building a larger community [69] around this code. The goal of this community should be to work together to obtain a DMFT code that is flexible to embed into other tools via its python interface and flexible with regard to the simulatable unit cells. Following recent trends in the community we have set up a github repository at <https://github.com/w2dynamics/w2dynamics> that serves as a central meeting point on where to find the code. The code is available to the public and people can interact with the developers via the issue tracking facilities and the wiki pages so that members of our community can collect information they consider useful for the project. There is a public mailing list w2dynamics-users@list.tuwien.ac.at that is also frequented by the developers; and a second developers mailing list w2dynamics-devel@list.tuwien.ac.at. We think that it is beneficial to users to become a part of our community so that they can benefit from the experience of experts in solid state physics and Monte Carlo techniques. To bring together people from various backgrounds we plan to organize workshops and community meetings that everybody is able to attend. Using an open platform like github we have the technical means to trace the contributions of everybody, to make sure that every contributor is listed on our project website, or, in the case of greater contributions, to authorship on a relevant contribution. Organizing workshops, hosting the servers and investing the time takes some effort but we are committed to invest that to grow an open community.

4.2. Installation

Since the *w2dynamics* code base tries to leverage the power of Fortran and the flexibility of python, the installation is a little bit more involved. In particular, some dependencies have to be taken care of. We tried our best to make the installation as simple as possible on as many platforms as possible. As build system we use `cmake` and we install a lot of the required dependencies if they cannot be found on the user's system. At the time of writing we strictly require `cmake > 2.8.5`, a FORTRAN90 compatible Fortran compiler, a C++ compiler that at least exposes the C++11 RNGs since we only provide a trivial fallback, the BLAS and LAPACK libraries, a python interpreter version 2.4 or bigger, the FFTW library, and an implementation of the MPI API. Without these libraries the installation step will fail. We will automatically install any of the following packages if it is missing: Those packages are the NFFT library, `numpy > 1.2`, `libhdf5 > 1.6` with Fortran bindings, `mpi4py`, `h5py`, `scipy > 0.6` with `f2py`, and `python-configobj`. Note that the compilation of NFFT and `libhdf5` necessitates a C compiler and the associated libraries. Python dependencies that are marked as optional will be downloaded and locally installed using the pip installer.

Let us assume (i) `cmake` is installed, (ii) the source files are downloaded, and (iii) the present working directory is the *w2dynamics* root folder. Then the following `cmake` commands should give working binaries for the CT-HYB code and the Maxent utility:

```
mkdir build
cd build
cmake ..
make
```

We routinely test the compilation of our code on various linux distributions such as Ubuntu, Debian, OpenSuse and CentOS. After compilation, we recommend the reader to do the tutorial. To circumvent the problem of rapidly aging documentation we refer the reader to our wiki page at <https://github.com/w2dynamics/w2dynamics/wiki/Tutorials>. Currently the tutorials show some of the features of *w2dynamics*. We show how to calculate the one-particle Green's function, the self-energy and orbital occupations of a one-orbital Bethe lattice Hubbard model infinite dimensions, how to use Wannier Hamiltonians as input, show a benchmark of performing a DMFT run in different basis sets and then comparing the results, and show how to calculate two-particle Greens functions with worm sampling.

The repository also includes additional test and benchmarks, which serves as a testbed for code development. The benchmarks have a structured layout and usually follow the same pattern. The user has to

- change to the directory of the benchmark which also contains a short description of the problem,
- inspect `Parameters.in`, which contains the parameters in key/value format (see Table 4),
- execute `DMFT.py` in this working folder.
- This produces an HDF5 archive with all simulation data (see Table 5)
- which can be accessed by the `hgrep` commands or the ipython notebook.

5. Input/output

Parameters. The *w2dynamics* code is mainly controlled by specifying a set of input parameters. Due to the versatility of the code a large amount of such input parameters exist, while most have reasonable default values pre-defined in 'auxiliaries/confispec'. By

Table 4
User-defined parameters.

Parameter	Option (type)	Description
[General]		
DOS	ReadIn	wannier90 Hamiltonian (path parameter 'HkFile')
	ReadInSO	wannier90 spin-orbit Hamiltonian (path parameter 'HkFile')
	Bethe	Bethe lattice/ semi-circular density of states
	EDcheck	Discrete bath parameters (input files 'epsk' and 'vk')
DMFTsteps	nano	Finite size (parameter 'readleads' and input file 'leadsfile')
	readDelta	Hybridization function (input files 'deltatau' and 'deltaiw')
beta	(integer)	Number of DMFT iterations
NAt	(float)	Inverse temperature
	(integer)	Atoms per unit cell (number of subgroups in [Atoms])
[Atoms]		
[[1]] ... [[NAt]]		
Hamiltonian	Density	Density–density interaction (defined by Udd, Jdd, Vdd)
	Kanamori	Slater–Kanamori interaction (defined by Udd, Jdd, Vdd)
	Coulomb	Coulomb interaction for the d-shell (defined by F0, F2, F4)
	ReadNormalUmatrix	Interaction matrix (path specified with 'umatrix')
Udd	ReadUmatrix	Spin-dependent interaction matrix (path parameter 'umatrix')
	(float)	Intra-orbital interaction
Nd	(integer)	Number of correlated bands
[QMC]		
Nmeas	(integer)	Monte Carlo measurement steps
Nwarmups	(integer)	Warmup/thermalization steps
NCorr	(integer)	Steps between two successive measurement/auto-correlation

Table 5
hdf5-file structure.

Group	Parent	Child	Dataset (attributes)	Description
.config	/	–	(general.dos), ...	Simulation parameters
.environment	/	–	(SHELL), (PATH), ...	Shell environment
.axes	/	–	iw, tau, ...	Frequency/time axes
.quantities	/	giw, gtau, ...	(axes), (desc), ...	Metadata for quantities
start	/	h-mean, lda-mu, ...	value	dmft initial data
dmft-	/	mu, ineq-, ...	value, error	dmft iteration
stat-	/	mu, ineq-, ...	value, error	Statistic iteration
finish	/	mu, ineq-, ...	value, error	hdf5-link to last iteration
ineq-	dmft-, stat-	giw, gtau, ...	value, error	Inequivalent atom

supplying a configuration file, these default values can be overwritten with user-defined values. A sample config file is distributed together with the code.

The configuration file is structured into the three sections [General], [Atoms] and [QMC], while the [Atoms] section consists of numbered subsections ([[1]], [[2]], ...) for each individual atom defined in the unit cell. A detailed list of mandatory user-defined parameters is given in Table 4.

HDF5 output. Any quantity measured by the impurity solver or calculated during the self-consistency cycle is stored in hierarchical data format (HDF5). Nowadays, this format is the de facto standard and is employed by the major DMFT software packages. File operations for reading and writing data from and to HDF5 are realized through the Python library *h5py*.

The output file consists of several hidden groups ('.config', '.environment', '.axes' and '.quantities'), which store the meta-data of each run. The '.config' group stores all simulation parameters (user-defined or default values). The '.environment' group stores the entire shell-environment and further scheduler-parameters on clusters. The '.axes' stores Matsubara and imaginary time axes, which are shared between different quantities. The link between these quantities and the corresponding axis is defined in '.quantities'.

Apart from the hidden group, the output file consists of several iteration groups ('start', 'dmft-', 'stat-', 'finish'). While 'start' stores information about the non-interacting problem all other iteration groups store quantities further grouped by the inequivalent atoms. A detailed list is given in Table 5.

hgrep utility. For quick access to the data stored in the HDF5 file, the *hgrep* command line utility is included with the code. *hgrep* can extract a subset of the data stored in one or more HDF5 files, and tabulate or plot the resulting data. As an example, the following command

```
hgrep -p test.hdf5 siw 1:3 1 1,4 1 0:20
      plot file  $\Sigma^{(iv)}$  iterations atom(l) orbital j spin $\sigma$  iv
      × field = value - im
      only imaginary part
```

plots the imaginary part of the impurity self-energy stored in the first *test.hdf5* for the first three iterations (FORTRAN-style ranges are supported), impurity problem $l = 1$, orbitals 1 and 4, spin up, in the frequency range $0 \leq iv \leq 20$. A man page is available, which lists more detailed syntax and provides further examples.

6. Conclusions

We have described here the main features of the hybridization-expansion continuous-time quantum Monte Carlo code package "w2dynamics". Users can calculate local two- and four-point fermionic Green's functions of multi-orbital impurity models. These can be either isolated impurities attached to leads or auxiliary impurities in DMFT or DFT+DMFT calculations of lattice Hamiltonians. The level of approximation is the DMFT; w2dynamics does not introduce further approximations to the numerically exact CT-QMC algorithm (as e.g. inner or outer truncations of the local trace).

The w2dynamics package is similar in spirit to TRIQS [11,68,70] and ALPS [9], all of which have their particular strengths. In the case of w2dynamics these are:

1. The calculation of arbitrary one- and two-particle Green's functions and local physical susceptibilities. This serves as an input for subsequent calculations of physical susceptibility and for diagrammatic extensions of DMFT [44] that use the local vertex as a building block. The package directly interfaces with post-DMFT packages such as AbinitioDGA [71] and LadderDGA [72].
2. Improved sampling techniques such as worm and superstate sampling. Worm sampling [13] overcomes the limitations of the standard hybridization measurement in partition function sampling, and allows to directly estimate all components of the n -particle Green's function including equal-time dynamic susceptibilities, improved estimators for the self-energy [13], and asymptotics of the four-point vertex [46]. Moreover, it extends the applicability of the CT-HYB algorithm to strongly insulating multi-orbital systems, where partition function sampling suffers from severe ergodicity problems [13]. Superstate sampling in turn allows to speed up the computation of the local weight by sampling over the block structure of the local Hamiltonian [67].
3. A large versatility of four-fermion interactions, ranging from the simplest density–density, to the rotationally-invariant Kanamori to the most general full Coulomb four-index tensor. Retardation effects (such as coming from electron-phonon interaction or cRPA screening) can be taken into account [59].
4. A flexible way of interfacing to DFT calculations also for materials with large unit cells made of atoms that can be inequivalent regarding symmetry, number of correlated orbitals, number of ligands, values of the Coulomb interaction, spin orientation, and so on. The input of wannier90 [28] generated Hamiltonians, after a transformation to k -space, is possible.
5. Treating magnetic phases and off-diagonal hybridization functions which may stem e.g. from (real-valued) inter-orbital hoppings. Future releases will offer full support for complex inter-orbital and inter-spin elements in order to run calculations with spin–orbit coupling.

Acknowledgments

We are indebted to M. Ferrero for his help in the early stages of this project and to B. Hartl, O. Janson, J. Kaufmann, P. Chalupa, M. Schüler, D. Springer and A. Valli for contributions to the development of the w2dynamics package. We thank these as well as E. Gull, P. Werner, A. Toschi, J. Tomczak, J. Kuneš, D. Di Sante, M. Aichhorn, A. Hariki for helpful discussions. Financial support is acknowledged from the Austrian Science Fund (FWF) SFB ViCoM F41 (MW), the Deutsche Forschungsgemeinschaft (DFG) research unit through SFB1170 (AH,AK,FG,GS) and research unit FOR 1346 (AH,KH,GS), the Vienna Scientific Cluster (VSC) Research Center funded by the Austrian Federal Ministry of Science and Research (PG), and the European Research Council under the European Union's Seventh Framework Program (FP/2007–2013) through ERC grant no. 306447 (KH). AH, AK, FG and GS gratefully acknowledge the Gauss Centre for Supercomputing e.V. for funding this project

by providing computing time on the GCS Supercomputer SuperMUC at Leibniz Supercomputing Centre (LRZ).

References

- [1] A. Hewson, *The Kondo Problem to Heavy Fermions*, Cambridge University Press, 1993.
- [2] R. Bulla, T. Costi, T. Pruschke, *Rev. Modern Phys.* 80 (2008) 395, <http://dx.doi.org/10.1103/RevModPhys.80.395>. URL <http://link.aps.org/doi/10.1103/RevModPhys.80.395>.
- [3] U. Schollwöck, *Rev. Modern Phys.* 77 (2005) 259.
- [4] M. Ganahl, P. Thunström, F. Verstraete, K. Held, H.G. Evertz, *Phys. Rev. B* 90 (2014) 045144, <http://dx.doi.org/10.1103/PhysRevB.90.045144>. URL <https://link.aps.org/doi/10.1103/PhysRevB.90.045144>.
- [5] F.A. Wolf, A. Go, I.P. McCulloch, A.J. Millis, U. Schollwöck, *Phys. Rev. X* 5 (2015) 041032, <http://dx.doi.org/10.1103/PhysRevX.5.041032>. URL <https://link.aps.org/doi/10.1103/PhysRevX.5.041032>.
- [6] A.N. Rubtsov, V.V. Savkin, A.I. Lichtenstein, *Phys. Rev. B* 72 (2005) 035122, <http://dx.doi.org/10.1103/PhysRevB.72.035122>. URL <http://link.aps.org/doi/10.1103/PhysRevB.72.035122>.
- [7] E. Gull, A.J. Millis, A.I. Lichtenstein, A.N. Rubtsov, M. Troyer, P. Werner, *Rev. Modern Phys.* 83 (2011) 349–404, <http://dx.doi.org/10.1103/RevModPhys.83.349>. URL <http://link.aps.org/doi/10.1103/RevModPhys.83.349>.
- [8] P. Werner, A. Comanac, L. de' Medici, M. Troyer, A.J. Millis, *Phys. Rev. Lett.* 97 (2006) 076405, <http://dx.doi.org/10.1103/PhysRevLett.97.076405>. URL <http://link.aps.org/doi/10.1103/PhysRevLett.97.076405>.
- [9] B. Bauer, L.D. Carr, H.G. Evertz, A. Feiguin, J. Freire, S. Fuchs, L. Gamper, J. Gukelberger, E. Gull, S. Guertler, A. Hehn, R. Igarashi, S.V. Isakov, D. Koop, P.N. Ma, P. Mates, H. Matsuo, O. Parcollet, G. Pawłowski, J.D. Picon, L. Pollet, E. Santos, V.W. Scarola, U. Schollwöck, C. Silva, B. Surer, S. Todo, S. Trebst, M. Troyer, M.L. Wall, P. Werner, S. Wessel, *J. Stat. Mech. Theory Exp.* 2011 (05) (2011) P05001 URL <http://stacks.iop.org/1742-5468/2011/i=05/a=P05001>.
- [10] E. Gaenko, *Comput. Phys. Comm.* 213 (2017) 235–251, <http://dx.doi.org/10.1016/j.cpc.2016.12.009>. URL <http://www.sciencedirect.com/science/article/pii/S0010465516303885>.
- [11] O. Parcollet, M. Ferrero, T. Ayril, H. Hafermann, I. Krivenko, L. Messio, P. Seth, *Comput. Phys. Comm.* 196 (2015) 398–415, <http://dx.doi.org/10.1016/j.cpc.2015.04.023>. URL <http://www.sciencedirect.com/science/article/pii/S0010465515001666>.
- [12] K. Haule, *Phys. Rev. B* 75 (2007) 155113, <http://dx.doi.org/10.1103/PhysRevB.75.155113>. URL <https://link.aps.org/doi/10.1103/PhysRevB.75.155113>.
- [13] P. Gunacker, M. Wallerberger, E. Gull, A. Hausoel, G. Sangiovanni, K. Held, *Phys. Rev. B* 92 (2015) 155102, <http://dx.doi.org/10.1103/PhysRevB.92.155102>. URL <http://link.aps.org/doi/10.1103/PhysRevB.92.155102>.
- [14] P. Gunacker, M. Wallerberger, T. Ribic, A. Hausoel, G. Sangiovanni, K. Held, *Phys. Rev. B* 94 (2016) 125153, <http://dx.doi.org/10.1103/PhysRevB.94.125153>.
- [15] A. Georges, G. Kotliar, W. Krauth, M.J. Rozenberg, *Rev. Modern Phys.* 68 (1996) 13–125, <http://dx.doi.org/10.1103/RevModPhys.68.13>. URL <http://link.aps.org/doi/10.1103/RevModPhys.68.13>.
- [16] R.M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press Cambridge, 2004.
- [17] O. Gunnarsson, M. Calandra, J.E. Han, *Rev. Modern Phys.* 75 (2003) 1085.
- [18] F. Aryasetiawan, O. Gunnarsson, *Rep. Progr. Phys.* 61 (3) (1998) 237. URL <http://stacks.iop.org/0034-4885/61/i=3/a=002>.
- [19] V.I. Anisimov, A.I. Poteryaev, M.A. Korotin, A.O. Anokhin, G. Kotliar, *J. Phys.: Condens. Matter* 9 (35) (1997) 7359, <http://dx.doi.org/10.1088/0953-8984/9/35/010>.
- [20] A.I. Lichtenstein, M.I. Katsnelson, *Phys. Rev. B* 57 (1998) 6884–6895, <http://dx.doi.org/10.1103/PhysRevB.57.6884>.
- [21] G. Kotliar, S.Y. Savrasov, K. Haule, V.S. Oudovenko, O. Parcollet, C.A. Marianetti, *Rev. Modern Phys.* 78 (2006) 865–951, <http://dx.doi.org/10.1103/RevModPhys.78.865>.
- [22] K. Held, *Adv. Phys.* 56 (2007) 829–926, <http://dx.doi.org/10.1080/00018730701619647>.
- [23] P. Sun, G. Kotliar, *Phys. Rev. B* 66 (2002) 085120, <http://dx.doi.org/10.1103/PhysRevB.66.085120>. URL <http://link.aps.org/doi/10.1103/PhysRevB.66.085120>.
- [24] S. Biermann, F. Aryasetiawan, A. Georges, *Phys. Rev. Lett.* 90 (2003) 086402, <http://dx.doi.org/10.1103/PhysRevLett.90.086402>. URL <http://link.aps.org/doi/10.1103/PhysRevLett.90.086402>.
- [25] J.M. Tomczak, *J. Phys. Conf. Ser.* 592 (1) (2015) 012055.
- [26] G. Knizia, G.K.-L. Chan, *Phys. Rev. Lett.* 109 (2012) 186404, <http://dx.doi.org/10.1103/PhysRevLett.109.186404>.
- [27] T.N. Lan, A.A. Kananenka, D. Zgid, *J. Chem. Phys.* 143 (24) (2015) 241102, <http://dx.doi.org/10.1063/1.4938562>.
- [28] A.A. Mostofi, J.R. Yates, Y.-S. Lee, I. Souza, D. Vanderbilt, N. Marzari, *Comput. Phys. Comm.* 178 (2008) 685.
- [29] K. Schwarz, P. Blaha, G.K.H. Madsen, *Comput. Phys. Comm.* 147 (2002) 71–76, [http://dx.doi.org/10.1016/S0010-4655\(02\)00206-0](http://dx.doi.org/10.1016/S0010-4655(02)00206-0). URL <http://www.sciencedirect.com/science/article/B6TJ5-45HWTK3-1/2/587e362e22aaa2e866c005ae5b2571>.

- [30] J. Kuneš R. Arita, P. Wissgott, A. Toschi, H. Ikeda, K. Held, *Comput. Phys. Comm.* 181 (2010) 1888.
- [31] S.Y. Savrasov, G. Kotliar, *Phys. Rev. B* 69 (2004) 245101, <http://dx.doi.org/10.1103/PhysRevB.69.245101>. URL <https://link.aps.org/doi/10.1103/PhysRevB.69.245101>.
- [32] M. Aichhorn, L. Pourovskii, A. Georges, *Phys. Rev. B* 84 (2011) 054529, <http://dx.doi.org/10.1103/PhysRevB.84.054529>. URL <https://link.aps.org/doi/10.1103/PhysRevB.84.054529>.
- [33] S. Bhandary, E. Assmann, M. Aichhorn, K. Held, *Phys. Rev. B* 94 (2016) 155131, <http://dx.doi.org/10.1103/PhysRevB.94.155131>. URL <https://link.aps.org/doi/10.1103/PhysRevB.94.155131>.
- [34] T. Maier, M. Jarrell, T. Pruschke, M.H. Hettler, *Rev. Modern Phys.* 77 (2005) 1027–1080, <http://dx.doi.org/10.1103/RevModPhys.77.1027>.
- [35] A. Toschi, A.A. Katanin, K. Held, *Phys. Rev. B* 75 (2007) 045118, <http://dx.doi.org/10.1103/PhysRevB.75.045118>. URL <http://link.aps.org/doi/10.1103/PhysRevB.75.045118>.
- [36] A.N. Rubtsov, M.I. Katsnelson, A.I. Lichtenstein, *Phys. Rev. B* 77 (2008) 033101, <http://dx.doi.org/10.1103/PhysRevB.77.033101>. URL <http://link.aps.org/doi/10.1103/PhysRevB.77.033101>.
- [37] G. Rohringer, A. Valli, A. Toschi, *Phys. Rev. B* 86 (2012) 125114, <http://dx.doi.org/10.1103/PhysRevB.86.125114>. URL <https://link.aps.org/doi/10.1103/PhysRevB.86.125114>.
- [38] C. Taranto, S. Andergassen, J. Bauer, K. Held, A. Katanin, W. Metzner, G. Rohringer, A. Toschi, *Phys. Rev. Lett.* 112 (2014) 196402, <http://dx.doi.org/10.1103/PhysRevLett.112.196402>. URL <http://link.aps.org/doi/10.1103/PhysRevLett.112.196402>.
- [39] G. Rohringer, A. Toschi, H. Hafermann, K. Held, V.I. Anisimov, A.A. Katanin, *Phys. Rev. B* 88 (2013) 115112, <http://dx.doi.org/10.1103/PhysRevB.88.115112>. URL <http://link.aps.org/doi/10.1103/PhysRevB.88.115112>.
- [40] A. Valli, T. Schäfer, P. Thunström, G. Rohringer, S. Andergassen, G. Sangiovanni, K. Held, A. Toschi, *Phys. Rev. B* 91 (2015) 115115, <http://dx.doi.org/10.1103/PhysRevB.91.115115>. URL <https://link.aps.org/doi/10.1103/PhysRevB.91.115115>.
- [41] G. Li, *Phys. Rev. B* 91 (2015) 165134, <http://dx.doi.org/10.1103/PhysRevB.91.165134>. URL <http://link.aps.org/doi/10.1103/PhysRevB.91.165134>.
- [42] T. Ayrál, O. Parcollet, *Phys. Rev. B* 93 (2016) 235124, <http://dx.doi.org/10.1103/PhysRevB.93.235124>. URL <http://link.aps.org/doi/10.1103/PhysRevB.93.235124>.
- [43] T. Ayrál, O. Parcollet, *Phys. Rev. B* 94 (2016) 075159, <http://dx.doi.org/10.1103/PhysRevB.94.075159>. URL <http://link.aps.org/doi/10.1103/PhysRevB.94.075159>.
- [44] G. Rohringer, H. Hafermann, A. Toschi, A.A. Katanin, A.E. Antipov, M.I. Katsnelson, A.I. Lichtenstein, A.N. Rubtsov, K. Held, Diagrammatic routes to non-local correlations beyond dynamical mean field theory, *ArXiv e-prints* [arXiv:arXiv:1705.00024](https://arxiv.org/abs/1705.00024).
- [45] T. Ribic, P. Gunacker, S. Isakov, M. Wallerberger, G. Rohringer, A.N. Rubtsov, E. Gull, K. Held, *Phys. Rev. B* 96 (2017) 235127, <http://dx.doi.org/10.1103/PhysRevB.96.235127>. URL <https://link.aps.org/doi/10.1103/PhysRevB.96.235127>.
- [46] J. Kaufmann, P. Gunacker, K. Held, *Phys. Rev. B* 96 (2017) 035114, <http://dx.doi.org/10.1103/PhysRevB.96.035114>. URL <https://link.aps.org/doi/10.1103/PhysRevB.96.035114>.
- [47] A. Valli, G. Sangiovanni, O. Gunnarsson, A. Toschi, K. Held, *Phys. Rev. Lett.* 104 (2010) 246402, <http://dx.doi.org/10.1103/PhysRevLett.104.246402>. URL <https://link.aps.org/doi/10.1103/PhysRevLett.104.246402>.
- [48] M. Schüler, S. Barthel, T. Wehling, M. Karolak, A. Valli, G. Sangiovanni, *Eur. Phys. J. Spec. Top.* 226 (11) (2017) 2615–2640, <http://dx.doi.org/10.1140/epjst/e2017-70049-3>.
- [49] M. Potthoff, W. Nolting, *Phys. Rev. B* 60 (1999) 7834–7849, <http://dx.doi.org/10.1103/PhysRevB.60.7834>. URL <http://link.aps.org/doi/10.1103/PhysRevB.60.7834>.
- [50] S. Florens, *Phys. Rev. Lett.* 99 (2007) 046402, <http://dx.doi.org/10.1103/PhysRevLett.99.046402>. URL <http://link.aps.org/doi/10.1103/PhysRevLett.99.046402>.
- [51] M. Snoek, I. Titvinidze, C. Tóke, K. Byczuk, W. Hofstetter, *New J. Phys.* 10 (9) (2008) 093008. URL <http://stacks.iop.org/1367-2630/10/i=9/a=093008>.
- [52] P. Hansmann, R. Arita, A. Toschi, S. Sakai, G. Sangiovanni, K. Held, *Phys. Rev. Lett.* 104 (2010) 197002, <http://dx.doi.org/10.1103/PhysRevLett.104.197002>. URL <http://link.aps.org/doi/10.1103/PhysRevLett.104.197002>.
- [53] M.J. Han, X. Wang, C.A. Marianetti, A.J. Millis, *Phys. Rev. Lett.* 107 (2011) 206804, <http://dx.doi.org/10.1103/PhysRevLett.107.206804>. URL <http://link.aps.org/doi/10.1103/PhysRevLett.107.206804>.
- [54] P. Hansmann, N. Parragh, A. Toschi, G. Sangiovanni, K. Held, *New J. Phys.* 16 (3) (2014) 033009. URL <http://stacks.iop.org/1367-2630/16/i=3/a=033009>.
- [55] N. Parragh, A. Toschi, K. Held, G. Sangiovanni, *Phys. Rev. B* 86 (2) (2012) 155158, <http://dx.doi.org/10.1103/PhysRevB.86.155158>. URL <http://link.aps.org/doi/10.1103/PhysRevB.86.155158>.
- [56] A. Georges, L. de' Medici, J. Mravlje, *Ann. Rev. Condens. Matter Phys.* 4 (1) (2013) 137–178., <http://dx.doi.org/10.1146/annurev-conmatphys-020911-125045>. arXiv:https://doi.org/10.1146/annurev-conmatphys-020911-125045.
- [57] N. Parragh, A. Toschi, K. Held, G. Sangiovanni, *Phys. Rev. B* 86 (2) (2012) 155158, <http://dx.doi.org/10.1103/PhysRevB.86.155158>. URL <https://link.aps.org/doi/10.1103/PhysRevB.86.155158>.
- [58] N. Parragh, *Strongly Correlated Multi-Orbital Systems - a Continuous-Time Quantum Monte Carlo Analysis* (Ph.D. thesis), University of Würzburg, 2013.
- [59] D. Springer, A. Hausoel, P. Chalupa, A. Kowalski, S. Ciuchi, A. Toschi, G. Sangiovanni, in preparation.
- [60] A. Georges, G. Kotliar, W. Krauth, M.J. Rozenberg, *Rev. Modern Phys.* 68 (1) (1996) 13, <http://dx.doi.org/10.1103/RevModPhys.68.13>.
- [61] G. Kotliar, S.Y. Savrasov, K. Haule, V.S. Oudovenko, O. Parcollet, C.A. Marianetti, *Rev. Modern Phys.* 78 (2006) 865, <http://dx.doi.org/10.1103/RevModPhys.78.865>. URL <http://link.aps.org/doi/10.1103/RevModPhys.78.865>.
- [62] H. Shinaoka, M. Dolfi, M. Troyer, P. Werner, *J. Stat. Mech.* 2014 (6) (2014) P06012 URL <http://stacks.iop.org/1742-5468/2014/i=6/a=P06012>.
- [63] M. Wallerberger (Ph.D. thesis), TU Wien, 2016.
- [64] A. Hariki, A. Hausoel, G. Sangiovanni, J. Kuneš, *Phys. Rev. B* 96 (2017) 155135, <http://dx.doi.org/10.1103/PhysRevB.96.155135>. URL <https://link.aps.org/doi/10.1103/PhysRevB.96.155135>.
- [65] A.D. Sokal, *Monte Carlo methods in statistical mechanics: Foundations and new algorithms*, Lectures at the Cargèse Summer School on Functional integration: Basics and applications, 1996. URL http://www.math.nyu.edu/faculty/goodman/teaching/Monte_Carlo/Sokal.ps.
- [66] B.A. Berg, *Markov Chain Monte Carlo Simulations and their Statistical Analysis*, 2004.
- [67] A. Kowalski, A. Hausoel, M. Wallerberger, P. Gunacker, G. Sangiovanni, State-and superstate-sampling in hybridization-expansion continuous-time quantum Monte Carlo, *ArXiv e-prints*. URL <http://arxiv.org/abs/1807.00361>.
- [68] P. Seth, I. Krivenko, M. Ferrero, O. Parcollet, *Comput. Phys. Comm.* 200 (2016) 274–284, <http://dx.doi.org/10.1016/j.cpc.2015.10.023>. URL <http://www.sciencedirect.com/science/article/pii/S001046551500404X>.
- [69] A. Budd, M. Corpas, M.D. Brazas, J.C. Fuller, J. Goecks, N.J. Mulder, M. Michaut, B.F.F. Ouellette, A. Pawlik, N. Blomberg, *PLoS Comput. Biol.* 11 (2) (2015) 1–11, <http://dx.doi.org/10.1371/journal.pcbi.1003972>. URL <https://doi.org/10.1371/journal.pcbi.1003972>.
- [70] M. Aichhorn, L. Pourovskii, P. Seth, V. Vildosola, M. Zingl, O.E. Peil, X. Deng, J. Mravlje, G.J. Kraberger, C. Martins, M. Ferrero, O. Parcollet, *Comput. Phys. Comm.* 204 (2016) 200–208, <http://dx.doi.org/10.1016/j.cpc.2016.03.014>. URL <http://www.sciencedirect.com/science/article/pii/S0010465516300728>.
- [71] A. Galler, P. Thunström, J. Kaufmann, M. Pickem, J.M. Tomczak, K. Held, The AbinitioD Γ A Project v1.0: Non-local correlations beyond and susceptibilities within dynamical mean-field theory *arXiv:1710.06651*. URL <https://arxiv.org/abs/1710.06651>.
- [72] G. Rohringer, A. Katanin, T. Schäfer, A. Hausoel, K. Held, A. Toschi, *arXiv:github.com/ladderDGA*. URL <https://github.com/ladderDGA/ladderDGA>.