DE LA RECHERCHE À L'INDUSTRIE

www.cea.fr

Development of DFT+DMFT and cRPA in ABINIT

Bernard Amadon[1], Valentin Planes[1], Nicolas Cliche[1], Jean-Baptiste Morée[1], Gregory Geneste[1], Yann Pouillon[2] and Jean Michel Beuken[3]

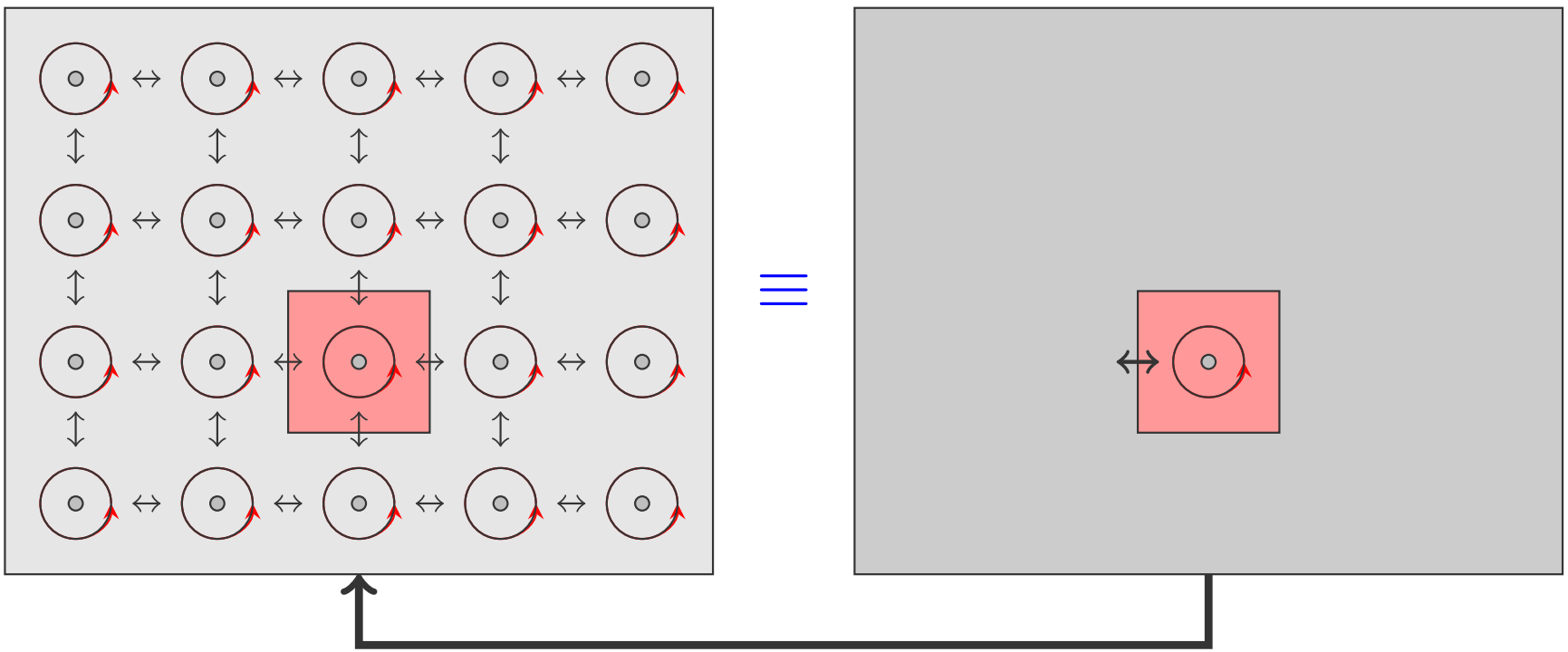
1 *CEA, DAM, DIF, F-91297 Arpajon, France*
2 *University of Cantabria, Santander*
3 *UCL, Louvain La neuve, Belgique*

May 2017

DFT+DMFT: Outline of the presentation.

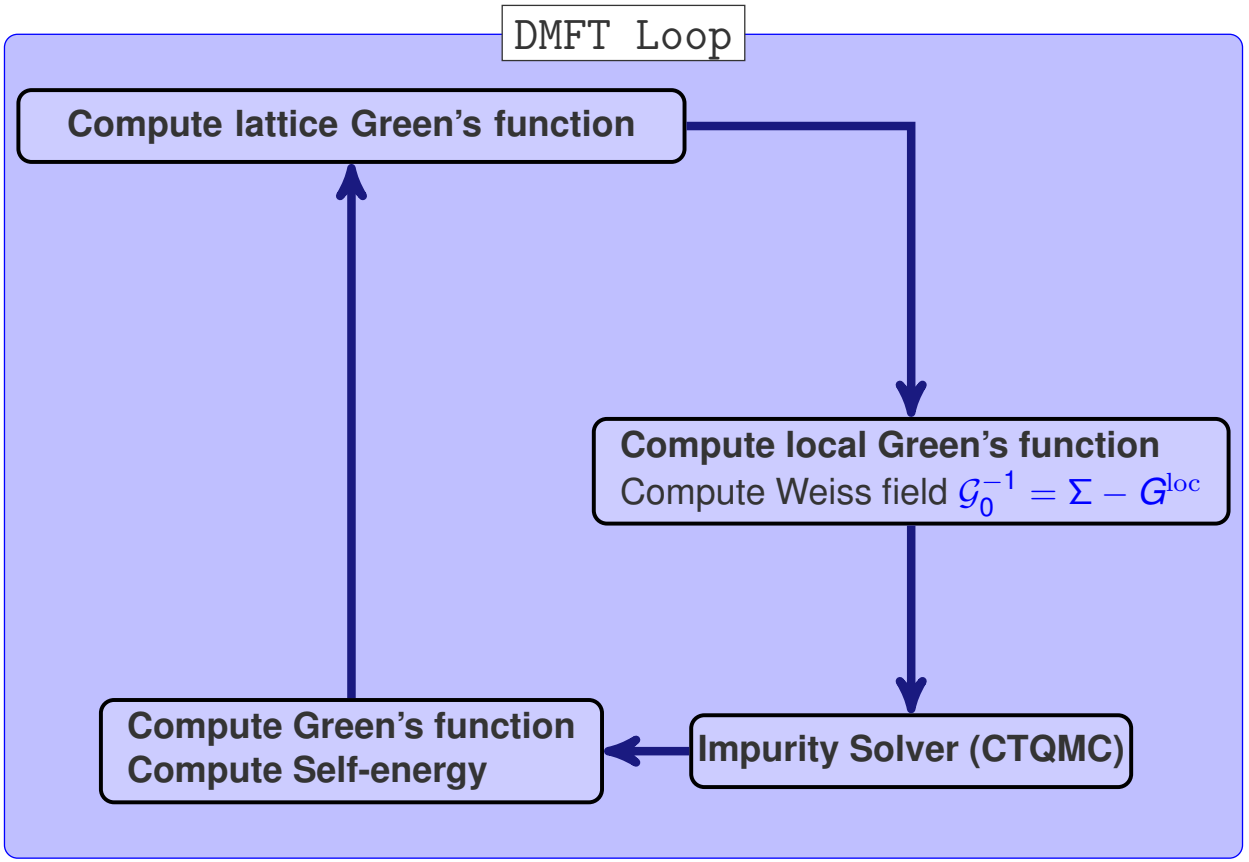
- Reminder: DFT+DMFT.
- Continuous time quantum Monte Carlo for DFT+DMFT in ABINIT
  - Interface with TRIQS
- Calculation of U with cRPA.
  - Definition of correlated orbitals: exemple of p orbitals.
- Tutorials
  - DFT+DMFT
  - cRPA
- Perspectives

The DMFT Loop



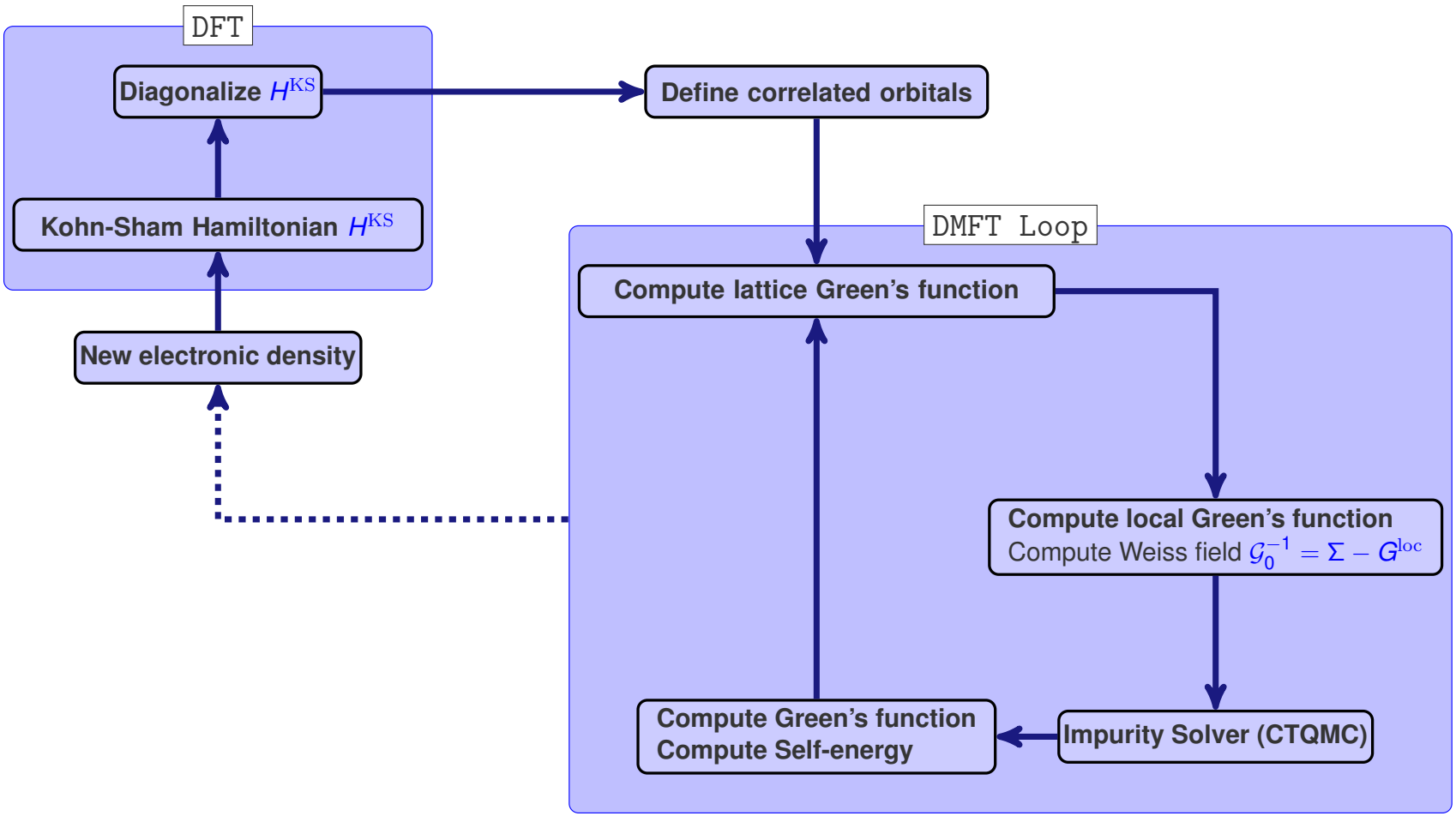
W. Metzner and D. Vollhardt Phys. Rev. Lett. 62 (3) 324 (1989)
A. Georges and G.Kotliar Phys. Rev. B 45 (12) 6479 (1992)
Antoine Georges, Gabriel Kotliar, Werner Krauth, and Marcelo J. Rozenberg Rev. Mod. Phys. 68, 13 (1996)

The DMFT Loop

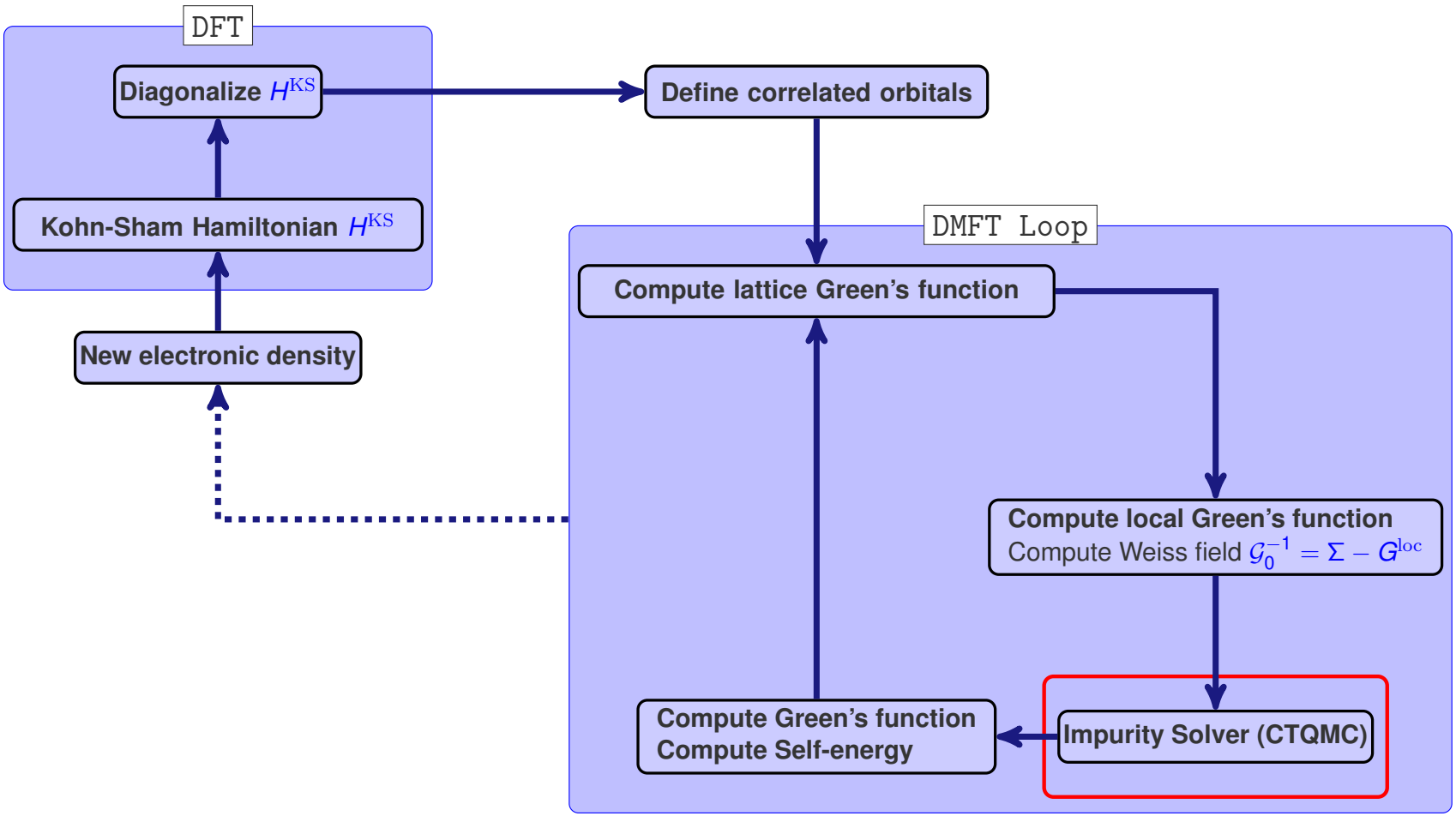


In a realistic system, one needs to define correlated orbitals, and hoppings integrals between all orbitals, and describe interactions among non-correlated orbitals.

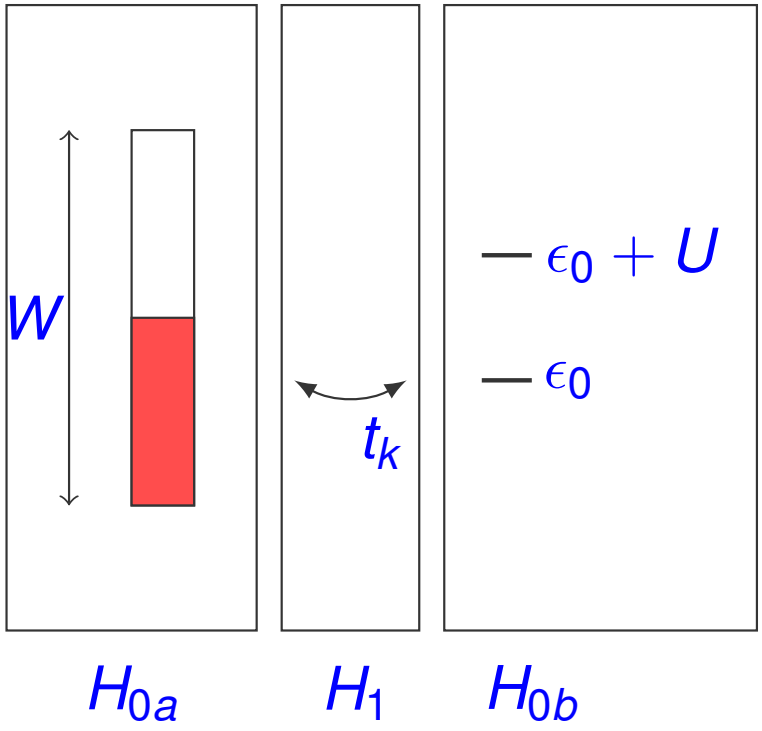
DFT+DMFT scheme



DFT+DMFT scheme



Anderson model



$$H_{\text{Anderson}} = \underbrace{\sum \omega_k a^+_{k,\sigma} a_{k\sigma}}_{H_{0a}} + \underbrace{\sum_{k,\sigma} t_{kf} a^+_{k,\sigma} f_\sigma}_{H_1} + \underbrace{\sum_\sigma \epsilon_f f^+_\sigma f_\sigma + U n_{f\uparrow} n_{f\downarrow}}_{H_{0b}}$$

Which CTQMC code ?

The complete interaction writes (for a system with three orbitals):

$$
\begin{aligned}
\hat{H}^{\mathrm{int}} &= \hat{H}^{\mathrm{nn}} + \hat{H}^{\mathrm{sf}} \\
\hat{H}^{\mathrm{nn}} &= \frac{U}{2} \sum_{(\alpha\sigma)\neq(\alpha'\sigma')} n_{\alpha\sigma} n_{\alpha'\sigma'} - J \sum_{\alpha\neq\alpha',\sigma} n_{\alpha\sigma} n_{\alpha'\bar{\sigma}} \\
&\quad - \frac{3J}{2} \sum_{\alpha\neq\alpha',\sigma} n_{\alpha\sigma} n_{\alpha'\sigma} \\
\hat{H}^{\mathrm{sf}} &= -\frac{J}{2} \sum_{\alpha\neq\alpha',\sigma} (c^{\dagger}_{\alpha\sigma} c^{\dagger}_{\alpha'\bar{\sigma}} c_{\alpha'\sigma} c_{\alpha\bar{\sigma}} + c^{\dagger}_{\alpha'\sigma} c^{\dagger}_{\alpha'\bar{\sigma}} c_{\alpha\sigma} c_{\alpha\bar{\sigma}}).
\end{aligned}
$$

- The exact formulation enables to have the correct degeneracy (3) of an electronic system with a spin S=1.
- In the CTQMC code contained in `ABINIT`, only H^{nn} is taken into account. It is the density density approximation
- Describing the complete (rotationnally invariant) interaction is possible but at the price of a cost prohibiting extensive calculations on *f* electrons systems.

Which CTQMC code ?

- CTQMC in ABINIT (Developed by J. Bieder (J. Bieder and B. Amadon PRB (2014)))
  - Well integrated with ABINIT
  - Fortran 90
  - Fast
  - Uses an approximation
- TRIQS: Toolbox for Research on Interacting Quantum Systems.
  - Licence GPL.
  - C++ & high level object oriented.
  - Module CTHYB: a generic quantum impurity solver based on the TRIQS library.
  - Developed at Ecole Polytechnique in Paris.
  - Exact but computationally expensive.

How to couple ABINIT and TRIQS

- Use iso_c_binding module which enables to interface Fortran and C++ routines.

in 68_dmft/qmc_prep_ctqmc.F90:

```
         !Location array in memory for C++ pointer args to pass
           g_iw_ptr       = C_LOC( gw_tmp_nd ) !C_LOC( g_iw )
           gtau_ptr       = C_LOC( gtmp_nd ) !C_LOC( gtau )
           gl_ptr         = C_LOC( gl_nd )
           fw1_nd_ptr     = C_LOC( fw1_nd_tmp )
           u_mat_ij_ptr   = C_LOC( u_mat_ij )
           u_mat_ijkl_ptr = C_LOC( u_mat_ijkl )
           levels_ptr     = C_LOC( levels_ctqmc )

         !Calling interfaced TRIQS solver subroutine from src/01_triqs_ext package
#if defined HAVE_TRIQS
           call Ctqmc_triqs_run (     rot_inv, leg_measure, hist, wrt_files, tot_not,  &
&           nflavor, nfreq, ntau , nleg, int(paw_dmft%dmftqmc_n/paw_dmft%nproc),       &
&           paw_dmft%dmftctqmc_meas*2*2*nflavor, paw_dmft%dmftqmc_therm,               &
&           verbosity_solver, paw_dmft%dmftqmc_seed,beta,                              &
&           levels_ptr,  u_mat_ij_ptr, u_mat_ijkl_ptr, fw1_nd_ptr,                     &
&           g_iw_ptr, gtau_ptr, gl_ptr, paw_dmft%spacecomm                             )
#endif
```

in "01_triqs_ext/triqs_cthyb_qmc.cpp":

```
void ctqmc_triqs_run( bool rot_inv, bool leg_measure, bool hist, bool wrt_files, bool tot_not,
                      int n_orbitals, int n_freq, int n_tau, int n_l, int n_cycles_, int cycle_length,
      int ntherm, int verbo, int seed, double beta_, double *epsi, double *umat_ij,
      double *umat_ijkl, std::complex<double> *f_iw_ptr, std::complex<double> *g_iw_ptr,
      double *g_tau, double *gl, MPI_Fint *MPI_world_ptr )
```

Work done by **Valentin Planes**.

Prerequisites

A complete list of requirements

TRIQS is built upon several python and C++ libraries, which, if not present already in your system, can be freely downloaded and installed. All the libraries and tools used by TRIQS are described below.

C++ compilers

TRIQS is written in C++, i.e. in the **C++** standard as defined by the ISO standardization committee, i.e. currently C++14. A recent compiler is therefore mandatory.

- Standard compliant C++ compilers (C++14).
  - g++ 4.9.1 and higher.
  - clang 3.4 and higher (in particular the default clang on OS X >= 10.8). The recommandation is to use the latest version of clang available.
  - Intel icc 16.0 seems to be fine too (it implements C++14). The intel icl on OS X (with the clang front end) seems to be working as well. In both cases, tests are still needed with these compilers, since none of the developers use them on a routine basis.
- C++98 and C++11 compilers are obsolete and will *never* be supported.
  - g++ before 4.9.1
  - Intel icc 15.0 and below

Libraries

Library/tool	Version	Comment
mpi	openmpi	Parallelism (1).
fftw	>= 3.2	Fourier transforms
boost	>= 1.49	C++ librairies
hdf5	>= 1.8.2	File storage system. Important: the *serial* version must be installed
python	>= 2.7	The Python interpreter
scipy	>= ?	Python mathematical library
numpy	>= ?	Python scientific library
h5py	>= ?	Python interface to hdf5 library
matplotlib	>= 0.99	Python 2D plotting library
mpi4py	>= ?	Python MPI
cmake	>= 2.8.7	CMake is used to control the software compilation process
mako	>= 0.9.1	mako templates are used to generate the C++/python wrapper
sphinx	>= 1.0.1	Python documentation tools (to compile documentation)
pyparsing	>= ?	Tool for sphinx (to compile documentation, for developers)
libclang	3.4	python bindings of the clang lib (to compile documentation, for developers)

(1) Since standard linux distributions (and macports on OS X) now provide openmpi, even on laptops, we avoid the unnecessary complication of maintaining a non-parallel version of TRIQS.

New variables for .ac file

C++

```
# Forced C++ compiler flags
# Note: will override build-system configuration - USE AT YOUR OWN RISKS!
#
CXXFLAGS="-std=c++1y"
```

HDF5

```
# Include flags for the NetCDF library (default is unset)
#
with_netcdf_incs="-I/usr/local/netcdf/include -I/usr/local/hdf5_mpich/include"

# Link flags for the NetCDF library (default is unset)
#
with_netcdf_libs="-L/usr/local/netcdf/lib -lnetcdff -lnetcdf -L/usr/local/hdf5_mpich/lib -lhdf5_hl -lhdf5'
#
```

TRIQS CTHYB boost

```
# Link with TRIQS library (for CTQMC and rotationally invariant interaction)
#    (default is no)
#
enable_triqs="yes"
#
# Include flags for the TRIQS library (default is unset)
#
with_triqs_incs="-I/usr/local/src/TRIQS/cthyb-1.3/c++ -I${BOOST_PATH}/include
 -I${TRIQS_PATH}/include -I/usr/include/python2.7 -I/   usr/include/hdf5/serial"

# Link flags for the TRIQS library (default is unset)
#
with_triqs_libs="-L${TRIQS_PATH}/lib -lcthyb_c -ltriqs -L/usr/local/boost_gcc/lib
-lboost_mpi -lboost_mpi_python -lboost_python -L/  usr/lib/x86_64-linux-gnu -lpython2.7
-lmpicxx -lstdc++"
```

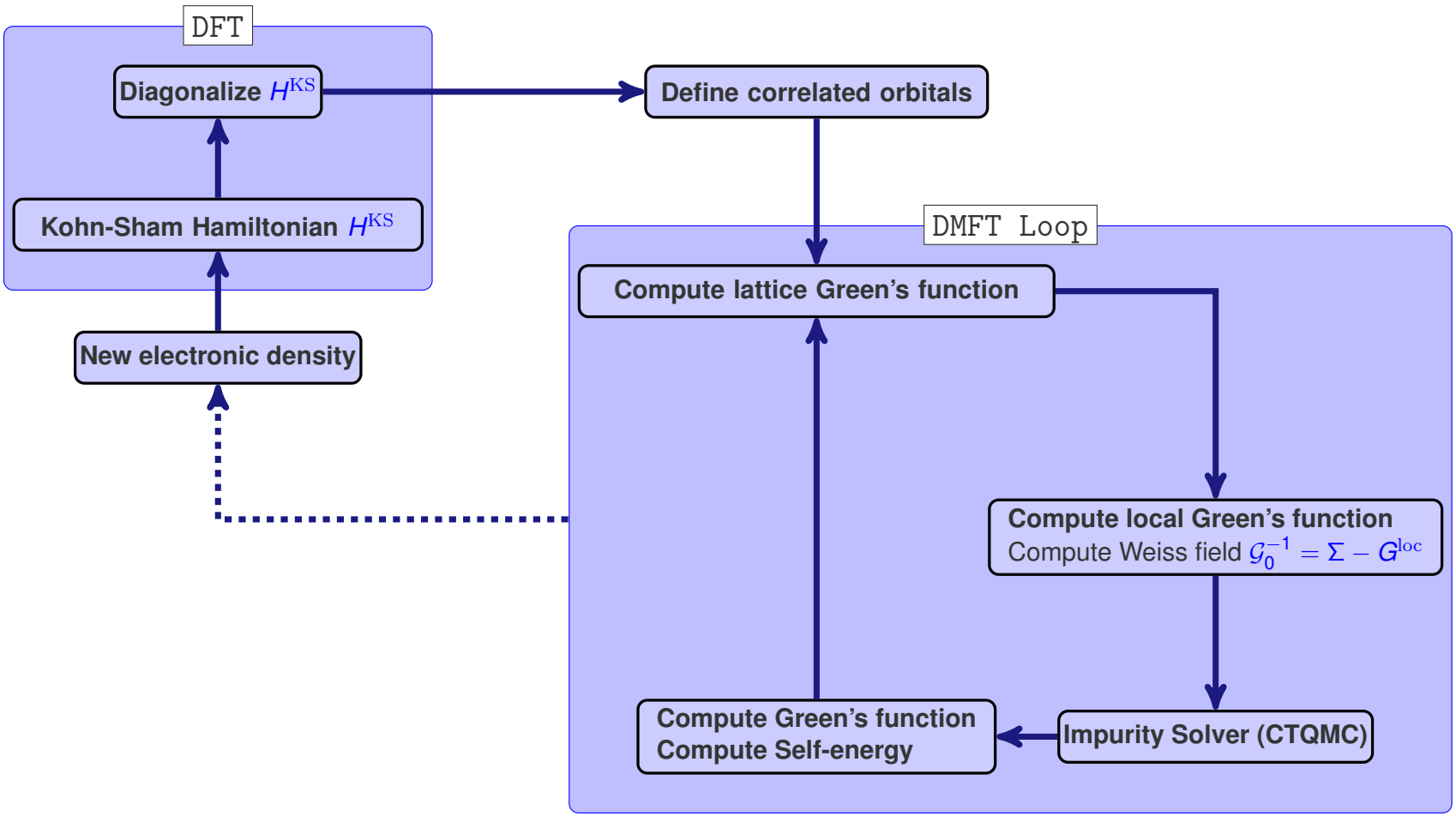
Use mpich instead of openmpi !!
Work done by **Valentin Planes**, special thanks to **Yann Pouillon** for the help with the built system and to **Jean Michel Beuken** to install TRIQS in the test farm and for the help to set-up an automatic test.

Automatic test v8/t01

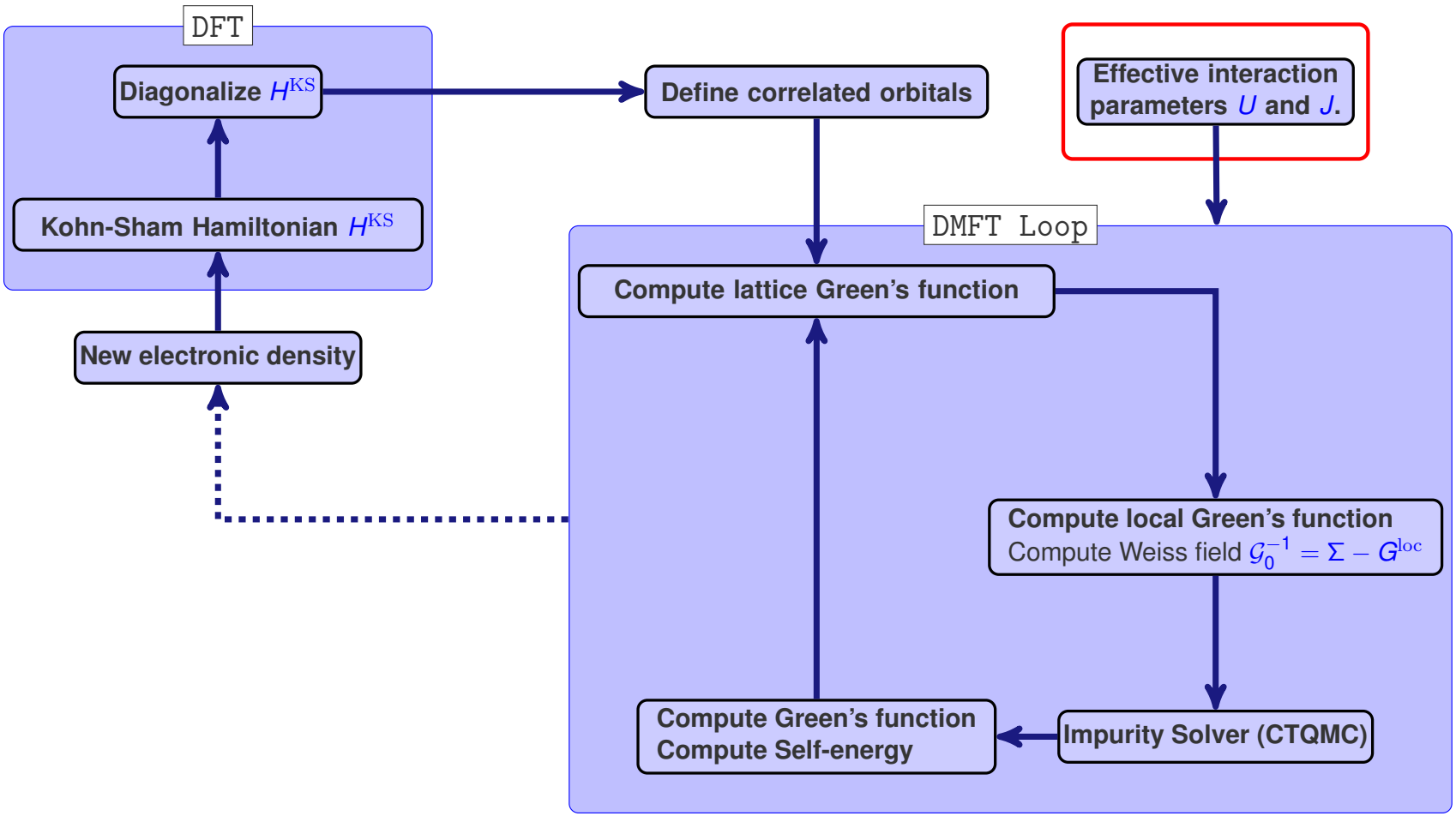
```
# == LDA+DMFT
usedmft1 0
usedmft2 1
dmftbandi 21 # t2g bands only
dmftbandf 23 # t2g bands only
#dmft_nwlo 200 # log frequency cannot be used (yet) with TRIQS
dmft_nwli 100
dmft_iter 1
dmftcheck 0
dmft_rslf 0
dmft_mxsf 0.7
dmft_dc 1
dmft_t2g 1   # special value for t2g only calculation.

# == CTQMC
dmft_solv 7 # Rotationally invariant interaction with TRIQS/CTHYB Monte Carlo code.
dmftqmc_l 201  # dmftqmc_l must be larger than 2*dmft_nwli
dmftqmc_n 2.d5
dmftqmc_therm 10000
dmft_tolfreq 0.01 # As log grid is not used, required precision is reduced.
dmftctqmc_triqs_nleg 30 # Number of Legendre polynomial to describe the Green's function.
```

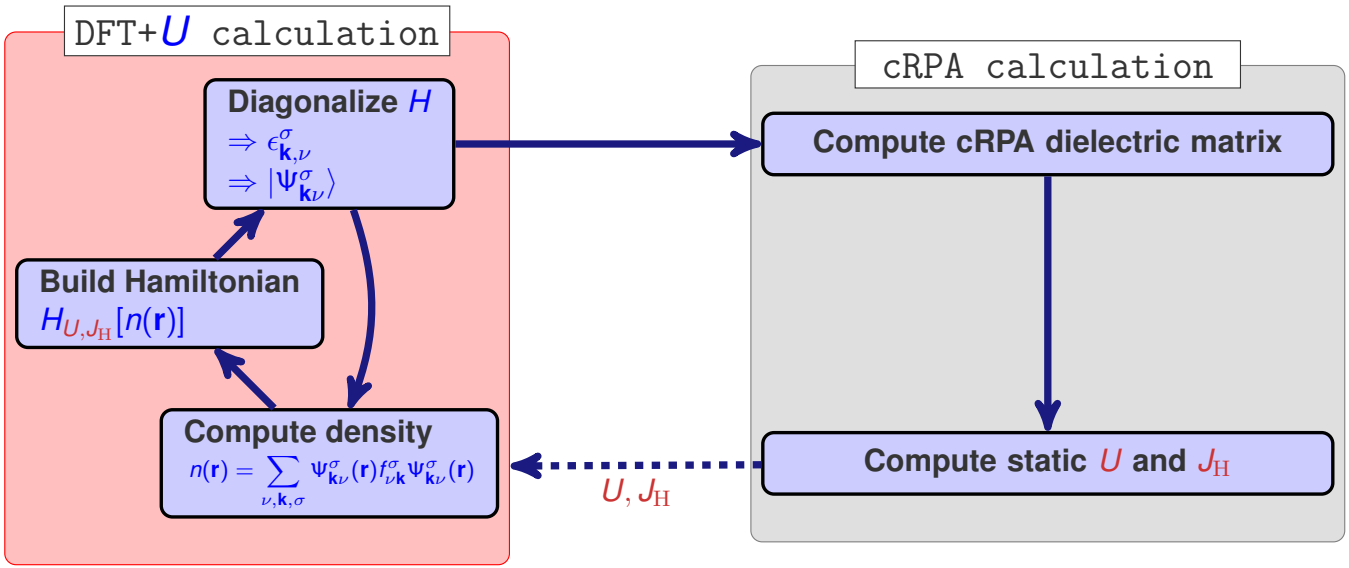
DFT+DMFT scheme in ABINIT



DFT+DMFT scheme in ABINIT



Self-consistent calculation of U in ABINIT



A simplified scheme to compute effective interactions.
K. Karlsson, F. Aryasetiawan, and O. Jepsen Phys. Rev. B 81, 245113 (2010)

Implemented in ABINIT(B. Amadon, T. Applencourt et F. Bruneval PRB (2014))

Coherence between correlated orbitals in DFT+U (truncated atomic orbitals) and cRPA (Wannier functions) ?

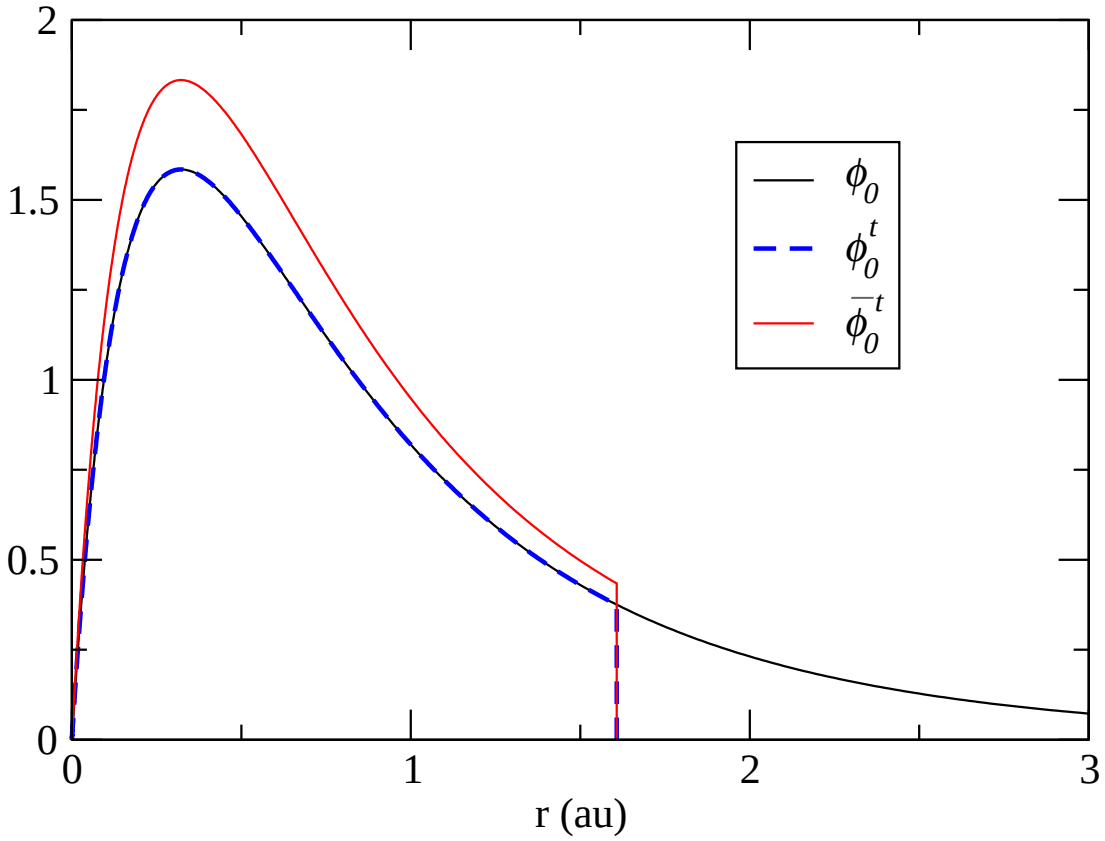


FIGURE: Comparison of ϕ_0, ϕ_0^t and $\bar{\phi}_0^t$ for the oxygen-*p* orbital

Choice of Wannier function that are coherent with localized orbitals

Orbital	Norm	v (eV)
$\phi_{\bar{0}}^{t}$	0.74	14.6
ϕ_{0}^{t}	1.00	26.7
ϕ_{0}	1.00	20.2
Wannier	1.00	20.2

Tutorial cRPA

- $SrVO_3$.
- Presentation of different ways to compute effective interactions, according to standart models of the litterature.
- Discussion about the localization of Wannier functions and comparison to atomic wavefunctions.
- Calculation of frequency dependent interaction.

Tutorial DFT+DMFT

- $SrVO_3$.
- Calculation of Green's function, and self-energy and renormalization weights, as well as total energy.
- Discussion about the localization of Wannier functions.
- Calculation of spectral function using an external Maximum Entropy code.

Perspectives and in progress

- Calculation of magnetic susceptibility into the public version.
- Spin orbit coupling into the public version.
- Calculation of forces.
- Calculation of **k**-resolved spectral function ?
- Towards GW+DMFT ?