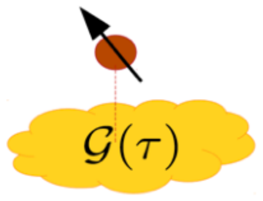
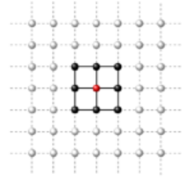
TRIQS Library & Applications

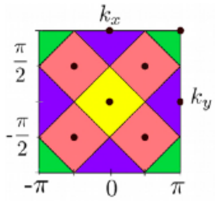
Nils Wentzell

SIMONS FOUNDATION

FLATIRON INSTITUTE
Center for Computational Quantum Physics







DMFT & Cluster Extensions

W I E N 2k

VASP

Abinit

WANNIER90

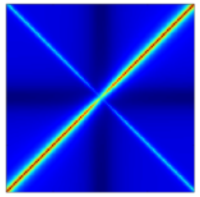
QUANTUMESPRESSO

DFT + DMFT

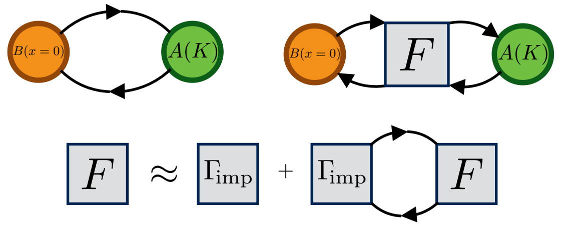
dft tools

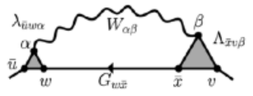
solid dmft

triqs



Vertex Methods





Impurity Solvers

ED

CTQMC

NRG

DMRG

PT

Non-Equilibrium

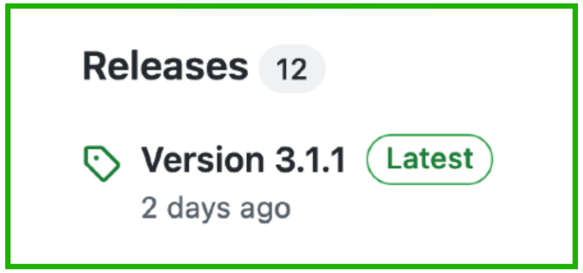
What is TRIQS?

- TRIQS - A **T**oolbox for **R**esearch on **I**nteracting **Q**uantum **S**ystems
  - TRIQS Library — Fundamental Building Blocks
  - Applications based on the TRIQS Library
- Open source (GPLv3 and Apache 2).
- High-level Interface in Python 3
- Low-level Backend in Modern C++

Doc: triqs.github.io

github.com/TRIQS/TRIQS

triqsworkspace.slack.com



Releases 12

Version 3.1.1 Latest

2 days ago

M. Ferrero

I. Krivenko

T. Ayral

Hugo Strand

P. Dumitrescu

A. Hampel

D. Simon

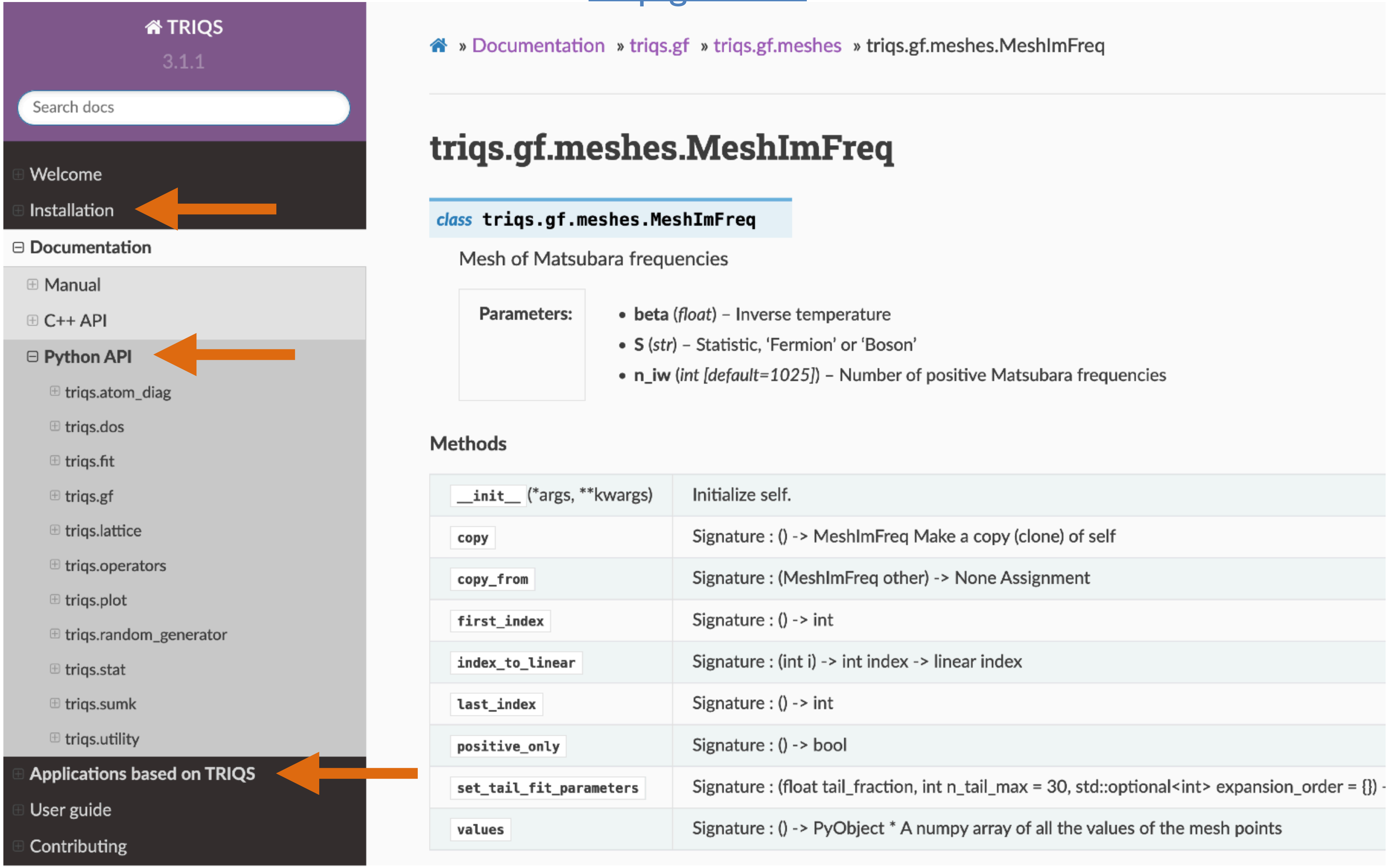
M. Zingl

A. Moutenet

O. Parcollet et al. CPC '15 ~ 360 Citations

What is TRIQS?

triqs.github.io



TRIQS
3.1.1
Search docs

- Welcome
- Installation
- Documentation
  - Manual
  - C++ API
  - Python API
    - triqs.atom_diag
    - triqs.dos
    - triqs.fit
    - triqs.gf
    - triqs.lattice
    - triqs.operators
    - triqs.plot
    - triqs.random_generator
    - triqs.stat
    - triqs.sumk
    - triqs.utility
- Applications based on TRIQS
- User guide
- Contributing

» Documentation » triqs.gf » triqs.gf.meshes » triqs.gf.meshes.MeshImFreq

triqs.gf.meshes.MeshImFreq

class `triqs.gf.meshes.MeshImFreq`

Mesh of Matsubara frequencies

Parameters:
- **beta** (*float*) – Inverse temperature
- **S** (*str*) – Statistic, 'Fermion' or 'Boson'
- **n_iw** (*int [default=1025]*) – Number of positive Matsubara frequencies

Methods

`__init__`(*args, **kwargs)	Initialize self.
`copy`	Signature : () -> MeshImFreq Make a copy (clone) of self
`copy_from`	Signature : (MeshImFreq other) -> None Assignment
`first_index`	Signature : () -> int
`index_to_linear`	Signature : (int i) -> int index -> linear index
`last_index`	Signature : () -> int
`positive_only`	Signature : () -> bool
`set_tail_fit_parameters`	Signature : (float tail_fraction, int n_tail_max = 30, std::optional<int> expansion_order = {}) -
`values`	Signature : () -> PyObject * A numpy array of all the values of the mesh points

TRIQS Library — Building Blocks

- Generic Green Function Objects, e.g. $G : (\mathbf{k}, i\omega) \rightarrow \mathbb{C}^{2\times 2}$
- Many Body Operators

```
print(n('up') + c('up') * c_dag('up'))
-> 1
```

- Lattice Tools
- Tools for Exact Diagonalization
- Monte Carlo Tools (Metropolis Hastings, Determinant Manipulations)
- Statistical Analysis Tools

```python
from triqs.gf import Gf, MeshImFreq

beta = 10.0   # Inverse temperature
n_iw = 200    # Number of pos. Matsubara frequencies
eps  = 1.0    # Energy

#Construct and initialize Green Function
iw_mesh = MeshImFreq(beta, 'Fermion', n_iw)
G = Gf(mesh = iw_mesh, target_shape=())

for iw in iw_mesh:
    G[iw] = 1.0 / (iw - eps)
```

```python
from triqs.operators import n
from triqs.atom_diag import AtomDiag

mu = 1.0   # Chemical potential
U  = 4.0   # Interaction

# Define Hamiltonian
H = U * n('up') * n('dn') + mu * (n('up') + n('dn'))

# Calculate Ground State Energy
ad = AtomDiag(H, [('up',), ('dn',)])
e_gs = ad.gs_energy
```

Basic Libraries — Standalone

HDF5 C++ Interface
github.com/TRIQS/h5

NDA - Multi-Array
github.com/TRIQS/nda

MPI C++ Interface
github.com/TRIQS/mpi

Itertools
github.com/TRIQS/itertools

Cpp2Py
github.com/TRIQS/cpp2py

App4TRIQS
github.com/TRIQS/app4triqs

ITENSOR

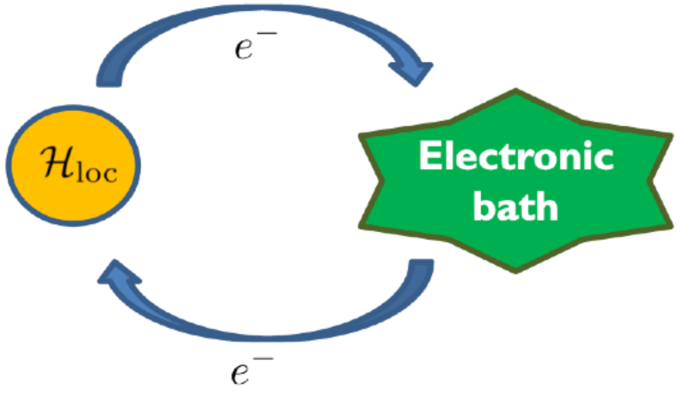
```
// Create array of shape (4,4)
array<int, 2> A(4, 4);

// Assign
A() = 0;
A(0, 1) = 40;
A(range(0, 2), 0) = 20;

// Algorithms
auto s = sum(abs(A));
```

```
// write to file
{
  h5::file f("dat.h5", 'w');
  h5::write(f, "A", A);
}

// read from file
array<int, 2> R;
{
  h5::file f("dat.h5", 'r');
  h5::read(f, "A", R);
}
```

TRIQS Applications — Impurity Solvers

DMFT Tutorial

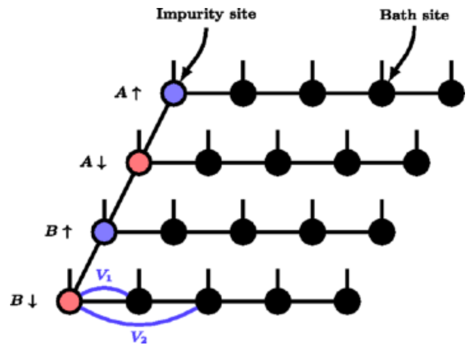
- CT-Hyb — Hybridization-Expansion QMC Solver
  triqs.github.io/cthyb   *P. Seth et al. CPC '16 ~ 250 Citations*
- CT-Seg — Segment-Picture Hybrid.-Exp. QMC Solver (unpublished)
- CT-Int — Interaction-Expansion QMC Solver (unpublished)
- HubbardI Solver
  triqs.github.io/hubbardI
- Hartree Fock
  triqs.github.io/hartree_fock



TRIQS Applications — Next-Generation Solvers

- ForkTPS DMRG Solver

triqs + ITENSOR

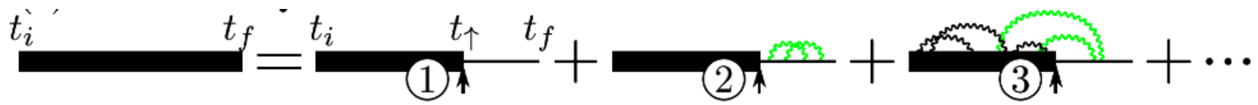


D. Bauernfeind et al. PRX '17

X. Cao

D. Bauernfeind

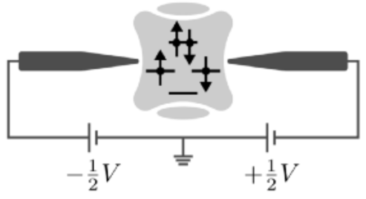
- Inchworm CTQMC Solver

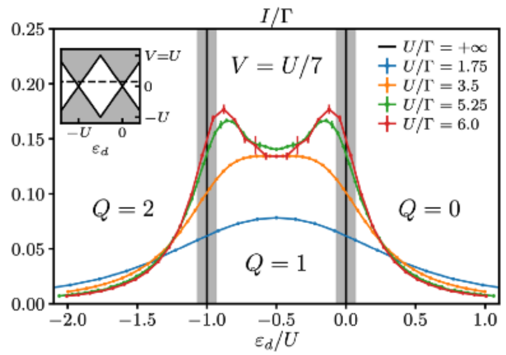


G. Cohen et al. PRL '15

M. Charlebois

- Keldysh Quasi-Monte-Carlo Solver





Marjan Maček et al. PRL '20

C. Bertrand

P. Dumitrescu

TRIQS Applications — Connection to Electronic Structure

- DFT Tools — Toolbox for Ab-Initio Calculations of Correlated Materials

  triqs.github.io/dft_tools

  M. Aichhorn et al. CPC ’16 ~ 150 Citations

A. Hampel

S. Beck

M. Aichhorn

L. Pourovskii

V. Vildosola

O. Peil

M. Zingl

M. Ferrero

G. Kraberger

J. Karp

TRIQS Applications — Connection to Electronic Structure

- DFT Tools — Toolbox for Ab-Initio Calculations of Correlated Materials

triqs.github.io/dft_tools

M. Aichhorn et al. CPC '16 ~ 120 Citations

WANNIER90

A. Hampel

S. Beck

M. Aichhorn

L. Pourovskii

V. Vildosola

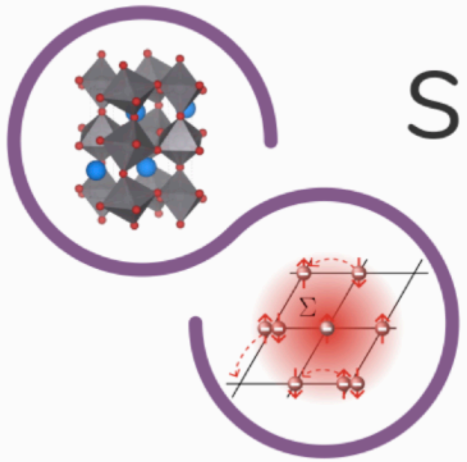
O. Peil

M. Zingl

M. Ferrero

G. Kraberger

J. Karp

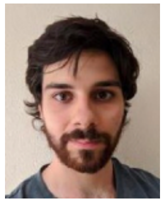


solid_dmft

A versatile python wrapper to perform DFT + DMFT calculations utilizing the TRIQS software library.

flatironinstitute.github.io/solid_dmft/

A. Hampel


A. Carta

S. Beck

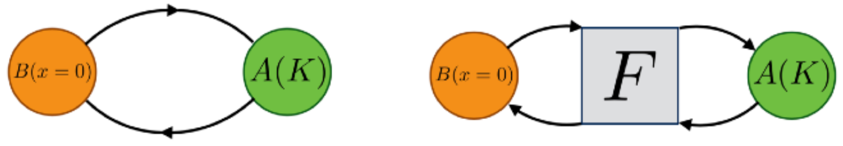
M. Merkel

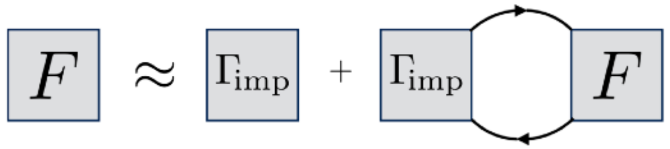
TRIQS Applications — Vertex Calculations

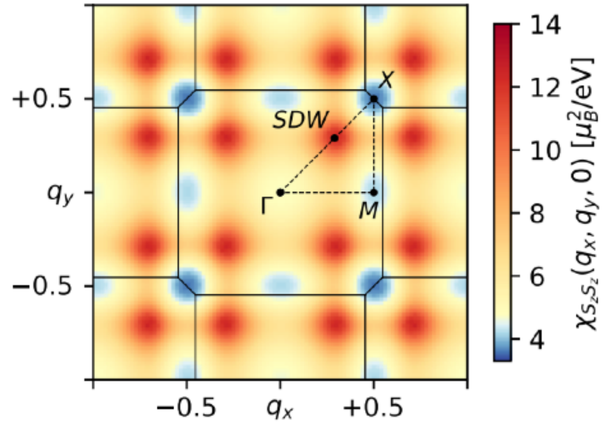
- TPRF — The Two-particle Response Function Tool Box

  triqs.github.io/tprf

H. Strand




$F \approx \Gamma_{\mathrm{imp}} + \Gamma_{\mathrm{imp}}$ F

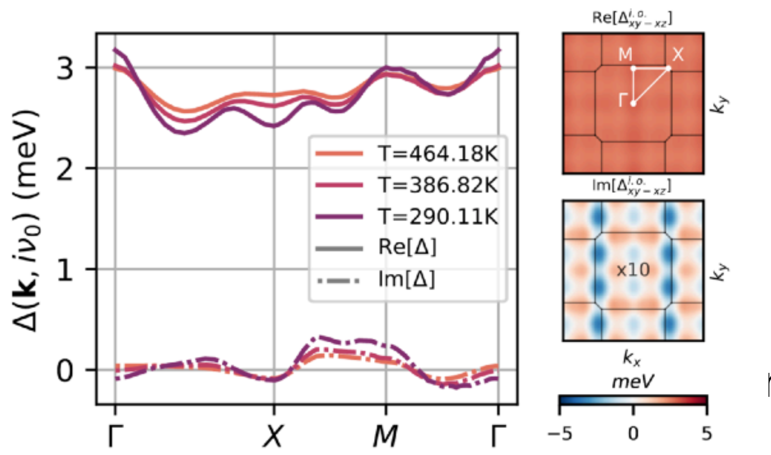


H. Strand et al. PRB '19

- Linearized Eliashberg
[illegible] Susceptibilities

S. Kaeser

P. Hansmann



(a) The $xy - xz$ component of the inter-orbital singlet gap function.

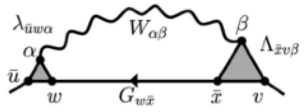
- Vertex-Corrected Lattice Susceptibilities

S. Kaeser et al. '21

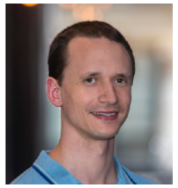
TRIQS Applications

- TRILEX — Triply-irreducible local expansion (private)

  Reach out to us if you are interested!



T. Schäfer



M. Richter

- MaxEnt — Analytic Continuation

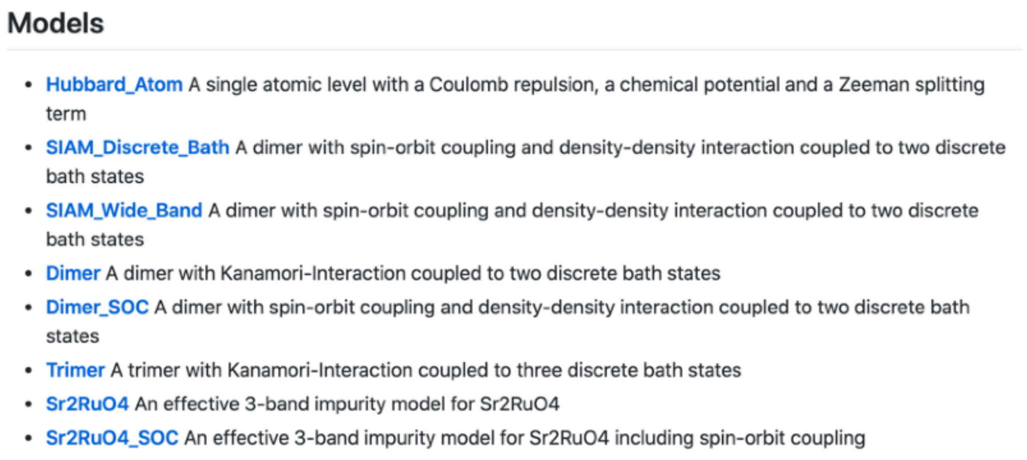
  triqs.github.io/maxent

G. Kraberger

M. Zingl

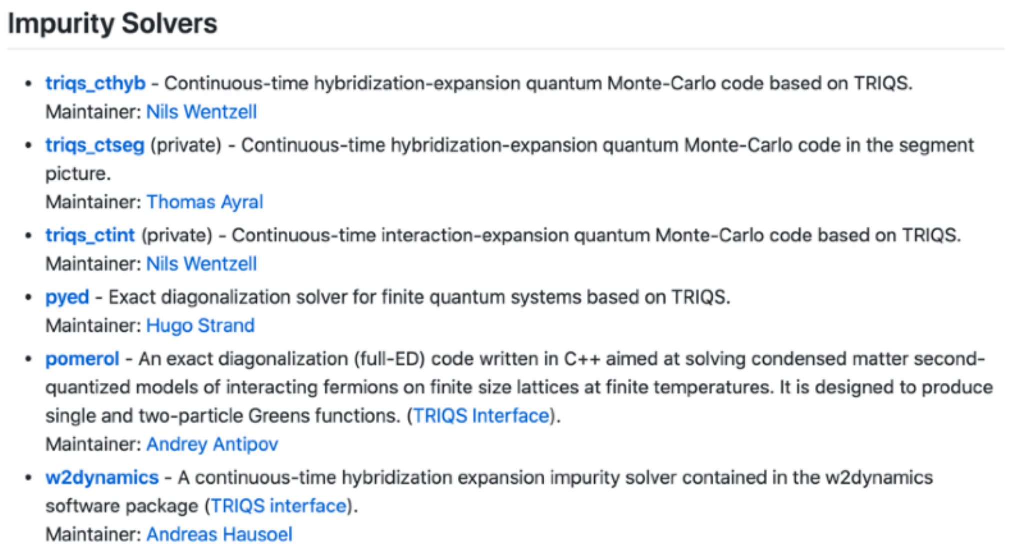
- Solver Benchmarks — A Set of Reference Impurity Models

  github.com/triqs/benchmarks



Models

- **Hubbard_Atom** A single atomic level with a Coulomb repulsion, a chemical potential and a Zeeman splitting term
- **SIAM_Discrete_Bath** A dimer with spin-orbit coupling and density-density interaction coupled to two discrete bath states
- **SIAM_Wide_Band** A dimer with spin-orbit coupling and density-density interaction coupled to two discrete bath states
- **Dimer** A dimer with Kanamori-Interaction coupled to two discrete bath states
- **Dimer_SOC** A dimer with spin-orbit coupling and density-density interaction coupled to two discrete bath states
- **Trimer** A trimer with Kanamori-Interaction coupled to three discrete bath states
- **Sr2RuO4** An effective 3-band impurity model for Sr2RuO4
- **Sr2RuO4_SOC** An effective 3-band impurity model for Sr2RuO4 including spin-orbit coupling



Impurity Solvers

- **triqs_cthyb** - Continuous-time hybridization-expansion quantum Monte-Carlo code based on TRIQS.
  Maintainer: Nils Wentzell
- **triqs_ctseg** (private) - Continuous-time hybridization-expansion quantum Monte-Carlo code in the segment picture.
  Maintainer: Thomas Ayral
- **triqs_ctint** (private) - Continuous-time interaction-expansion quantum Monte-Carlo code based on TRIQS.
  Maintainer: Nils Wentzell
- **pyed** - Exact diagonalization solver for finite quantum systems based on TRIQS.
  Maintainer: Hugo Strand
- **pomerol** - An exact diagonalization (full-ED) code written in C++ aimed at solving condensed matter second-quantized models of interacting fermions on finite size lattices at finite temperatures. It is designed to produce single and two-particle Greens functions. (TRIQS Interface).
  Maintainer: Andrey Antipov
- **w2dynamics** - A continuous-time hybridization expansion impurity solver contained in the w2dynamics software package (TRIQS interface).
  Maintainer: Andreas Hausoel

TRIQS Interfaces to External Codes

- Interface to NRGLjubljana Code

  triqs.github.io/nrgljubljana_interface

Rok Zitko

- Interface to the Pomerol Exact Diagonalization Code

  github.com/krivenko/pomerol2triqs

I. Krivenko

A. Antipov

- Interface to OmegaMaxEnt (Sherbrooke code)

  triqs.github.io/omegamaxent_interface

D. Bergeron

- Interface to w2dynamics CTHyb Code

  triqs.github.io/w2dynamics_interface

A. Hausoel

A. Kowalski

External Applications

- SOM — Stochastic Optimization Method for Analytic Continuation
  krivenko.github.io/som

I. Krivenko

- DCore — Toolbox for Ab-Initio DMFT Calculations
  github.com/issp-center-dev/DCore

H. Shinaoka

J. Otsuki

- PyED — Exact Diagonalization for finite Quantum Systems
  github.com/hugostrand/pyed/

Hugo Strand

- Dualfermion — Second order dual fermion implementation
  github.com/egcpvanloon/dualfermion

E.Van Loon

TRIQS — Packaging

triqs.github.io/triqs/latest/install.html

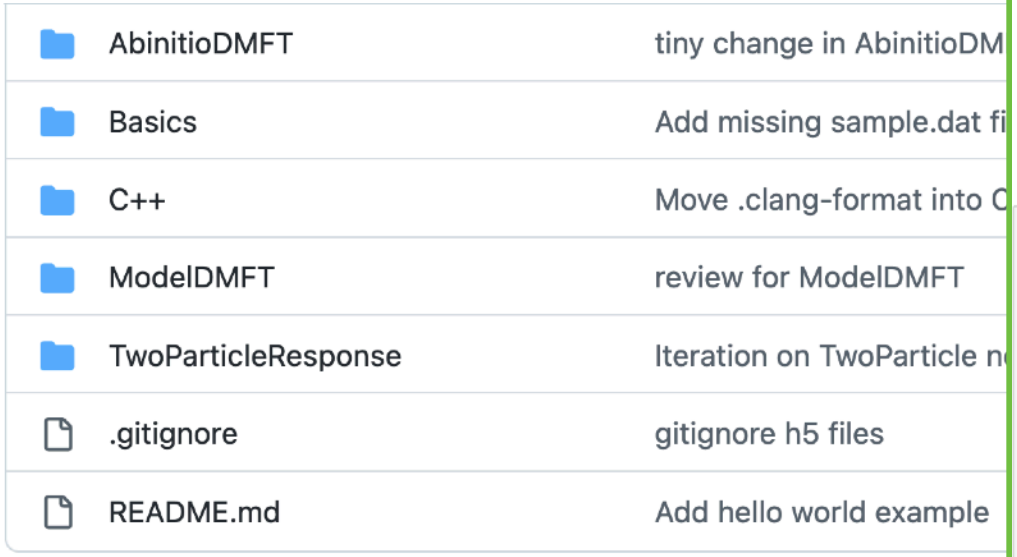
- Anaconda `conda install -c conda-forge triqs`
- Debian Packages for Ubuntu 20.04 and 22.04 `apt-get install triqs`
- Binder Notebook triqs.github.io/notebook
- Docker Image
```
docker pull flatironinstitute/triqs
docker run -p 8888:8888 flatironinstitute/triqs
```
- Singularity
```
singularity pull docker://flatironinstitute/triqs
singularity exec triqs.sif python myscript.py
```
- EasyBuild `eb -r --software-name=TRIQS`

TRIQS Install-Session after Dinner!

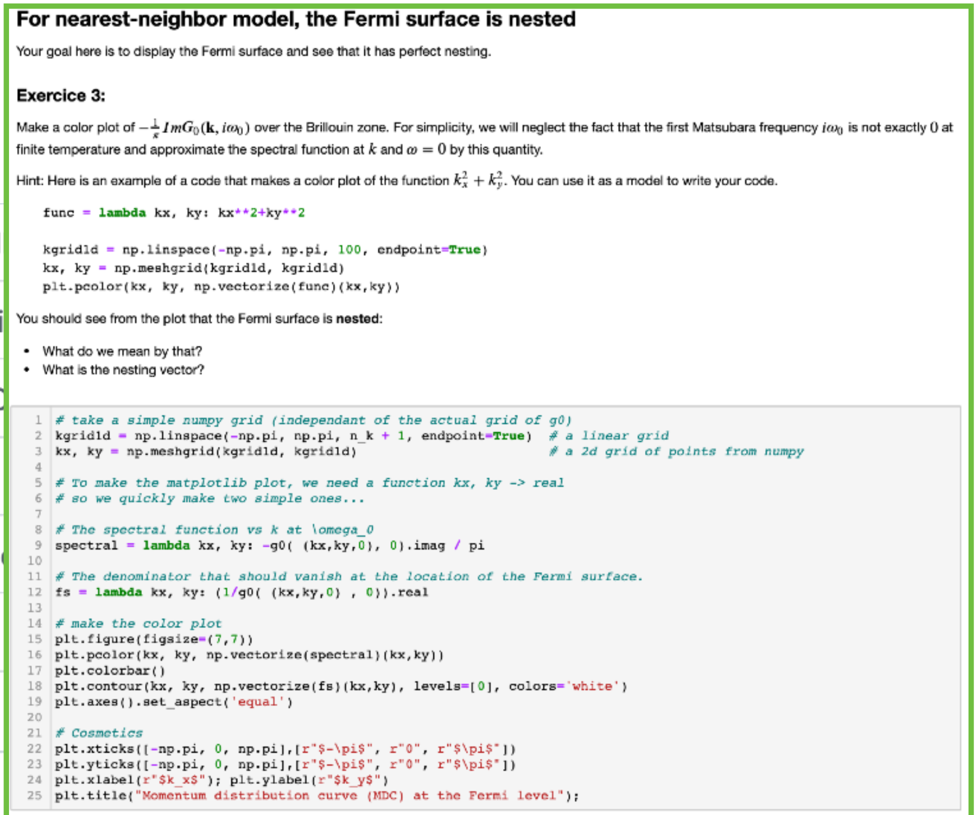
TRIQS — Getting Started

github.com/TRIQS/tutorials

- Set of IPython Notebook Tutorial



AbinitioDMFT	tiny change in AbinitioDM
Basics	Add missing sample.dat fi
C++	Move .clang-format into C
ModelDMFT	review for ModelDMFT
TwoParticleResponse	Iteration on TwoParticle n
.gitignore	gitignore h5 files
README.md	Add hello world example



For nearest-neighbor model, the Fermi surface is nested

Your goal here is to display the Fermi surface and see that it has perfect nesting.

Exercice 3:

Make a color plot of $-\frac{1}{\pi} Im G_0(\mathbf{k}, i\omega_0)$ over the Brillouin zone. For simplicity, we will neglect the fact that the first Matsubara frequency $i\omega_0$ is not exactly 0 at finite temperature and approximate the spectral function at k and $\omega = 0$ by this quantity.

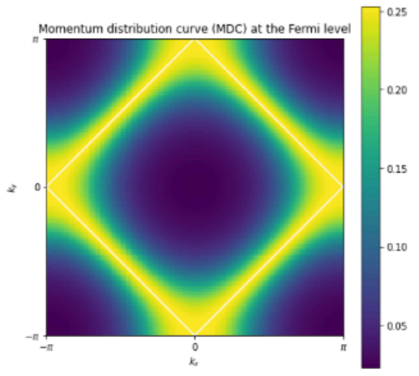
Hint: Here is an example of a code that makes a color plot of the function $k_x^2 + k_y^2$. You can use it as a model to write your code.

```
func = lambda kx, ky: kx**2+ky**2

kgrid1d = np.linspace(-np.pi, np.pi, 100, endpoint=True)
kx, ky = np.meshgrid(kgrid1d, kgrid1d)
plt.pcolor(kx, ky, np.vectorize(func)(kx,ky))
```

You should see from the plot that the Fermi surface is **nested**:

- What do we mean by that?
- What is the nesting vector?

```
# take a simple numpy grid (independant of the actual grid of g0)
kgrid1d = np.linspace(-np.pi, np.pi, n_k + 1, endpoint=True)  # a linear grid
kx, ky = np.meshgrid(kgrid1d, kgrid1d)                        # a 2d grid of points from numpy

# To make the matplotlib plot, we need a function kx, ky -> real
# so we quickly make two simple ones...

# The spectral function vs k at \omega_0
spectral = lambda kx, ky: -g0( (kx,ky,0), 0).imag / pi

# The denominator that should vanish at the location of the Fermi surface.
fs = lambda kx, ky: (1/g0( (kx,ky,0) , 0)).real

# make the color plot
plt.figure(figsize=(7,7))
plt.pcolor(kx, ky, np.vectorize(spectral)(kx,ky))
plt.colorbar()
plt.contour(kx, ky, np.vectorize(fs)(kx,ky), levels=[0], colors='white')
plt.axes().set_aspect('equal')

# Cosmetics
plt.xticks([-np.pi, 0, np.pi],[r"$-\pi$", r"0", r"$\pi$"])
plt.yticks([-np.pi, 0, np.pi],[r"$-\pi$", r"0", r"$\pi$"])
plt.xlabel(r"$k_x$"); plt.ylabel(r"$k_y$")
plt.title("Momentum distribution curve (MDC) at the Fermi level");
```



TRIQS — Getting Started

github.com/TRIQS/tutorials

TRIQS tutorial: getting started

Setting up JupyterLab:

1. log in to `jupyter.c2.quantum.ccs.usherbrooke.ca`
2. set # cores = 6, `mem = 4096`, `User Interface = JupyterLab`, `Duration 4h`
3. check that the jupyter kernel is set to `py3-triqs`
4. copy the tutorials to your home `cp -r /project/triqs/tutorials ~/.`
5. in the File Browser of JupyterLab navigate to `Basics;TwoParticleResponse (Day 1) or ModelDMFT (Day 2)`
6. start with the first notebook

If you are prompted to go to the terminal:

1. in JupyterLab use `New Launcher`, open `Other/Terminal` and type `source /project/triqs/load_triqs.sh`

🤔 for troubleshooting use search on: `triqs.github.io`

Setting up JupyterLab:

1. log in to `jupyter.c2.quantum.ccs.usherbrooke.ca`
2. set # cores = 6, `mem = 4096`, `User Interface = JupyterLab`, `Duration 4h`
3. check that the jupyter kernel is set to `py3-triqs`
4. copy the tutorials to your home `cp -r /project/triqs/tutorials ~/.`
5. in the File Browser of JupyterLab navigate to `Basics;TwoParticleResponse (Day 1) or ModelDMFT (Day 2)`
6. start with the first notebook

If you are prompted to go to the terminal:

1. in JupyterLab use `New Launcher`, open `Other/Terminal` and type `source /project/triqs/load_triqs.sh`

🤔 for troubleshooting use search on: `triqs.github.io`