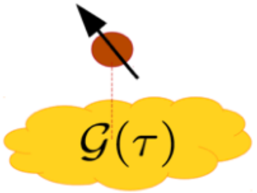
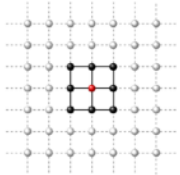
TRIQS

A Software Platform for Quantum Embedding

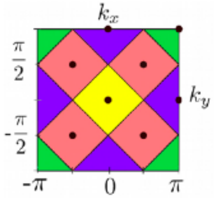
SIMONS FOUNDATION

FLATIRON INSTITUTE





O. Parcollet

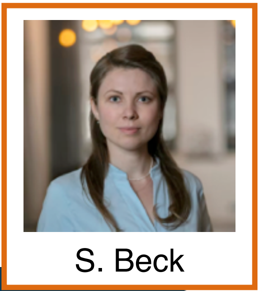


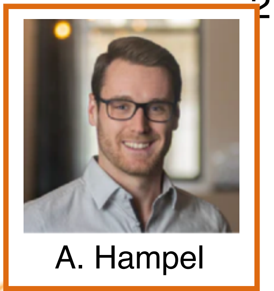
DMFT & Cluster Extensions

Abinit

WANNIER90

QUANTUMESPRESSO


S. Beck

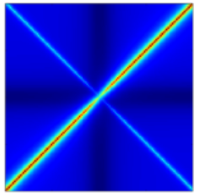

A. Hampel

DFT + DMFT

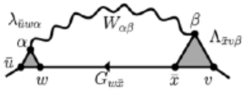
dft tools

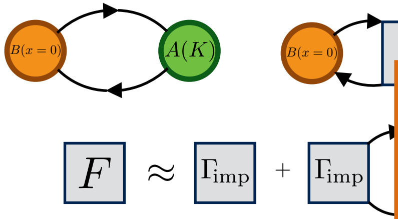
solid dmft

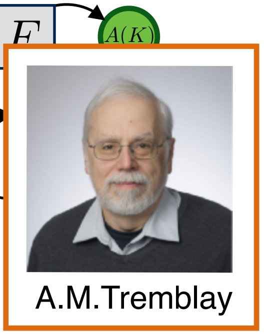
TRIQS



Vertex Methods






A.M.Tremblay

Impurity Solvers

ED

CTQMC

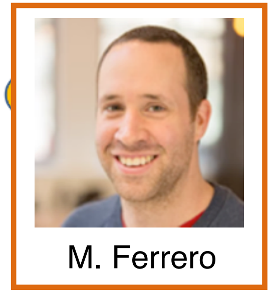
NRG

DMRG

DiagMC

PT

Non-Equilibrium

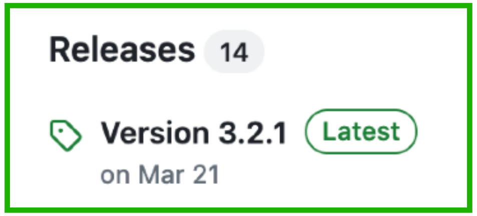

M. Ferrero

O. Parcollet et al. CPC ’15 ~ 420 Citations

TRIQS Library

- 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

triqs.github.io



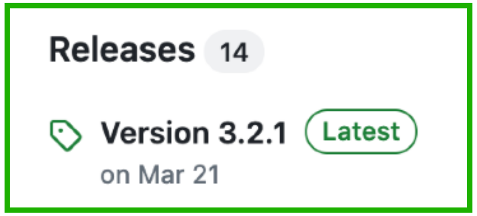
Releases 14

Version 3.2.1 Latest
on Mar 21

TRIQS Library

- 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

triqs.github.io



Releases 14

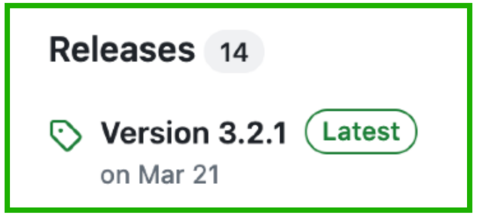
Version 3.2.1 Latest
on Mar 21

O. Parcollet et al. CPC '15 ~ 430 Citations

TRIQS Library

- 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++

triqs.github.io



Releases 14

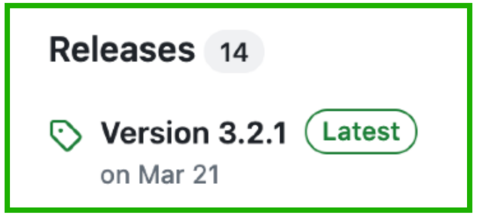
Version 3.2.1 Latest
on Mar 21

O. Parcollet et al. CPC ’15 ~ 430 Citations

TRIQS Library

- 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++

triqs.github.io



Releases 14

Version 3.2.1 Latest

on Mar 21

O. Parcollet

A. Hampel

H. Strand

D. Kiese

P. Dumitrescu

J. Kaye

M. Ferrero

I. Krivenko

T. Ayral

D. Simon

M. Zingl

A. Moutenet

S. Beck

O. Parcollet et al. CPC ’15 ~ 430 Citations

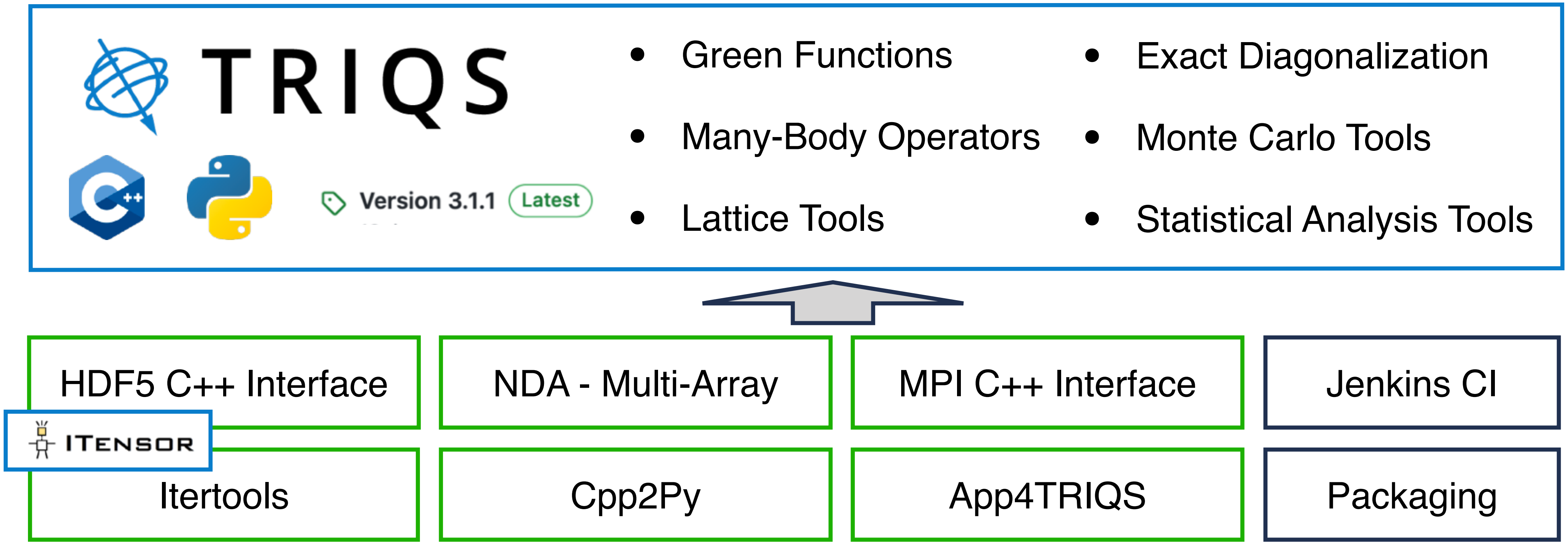
TRIQS — Software Stack

TRIQS

Version 3.1.1 Latest

- Green Functions
- Many-Body Operators
- Lattice Tools
- Exact Diagonalization
- Monte Carlo Tools
- Statistical Analysis Tools

TRIQS — Software Stack



TRIQS

Version 3.1.1 Latest

- Green Functions
- Many-Body Operators
- Lattice Tools
- Exact Diagonalization
- Monte Carlo Tools
- Statistical Analysis Tools

HDF5 C++ Interface

NDA - Multi-Array

MPI C++ Interface

Jenkins CI

ITensor

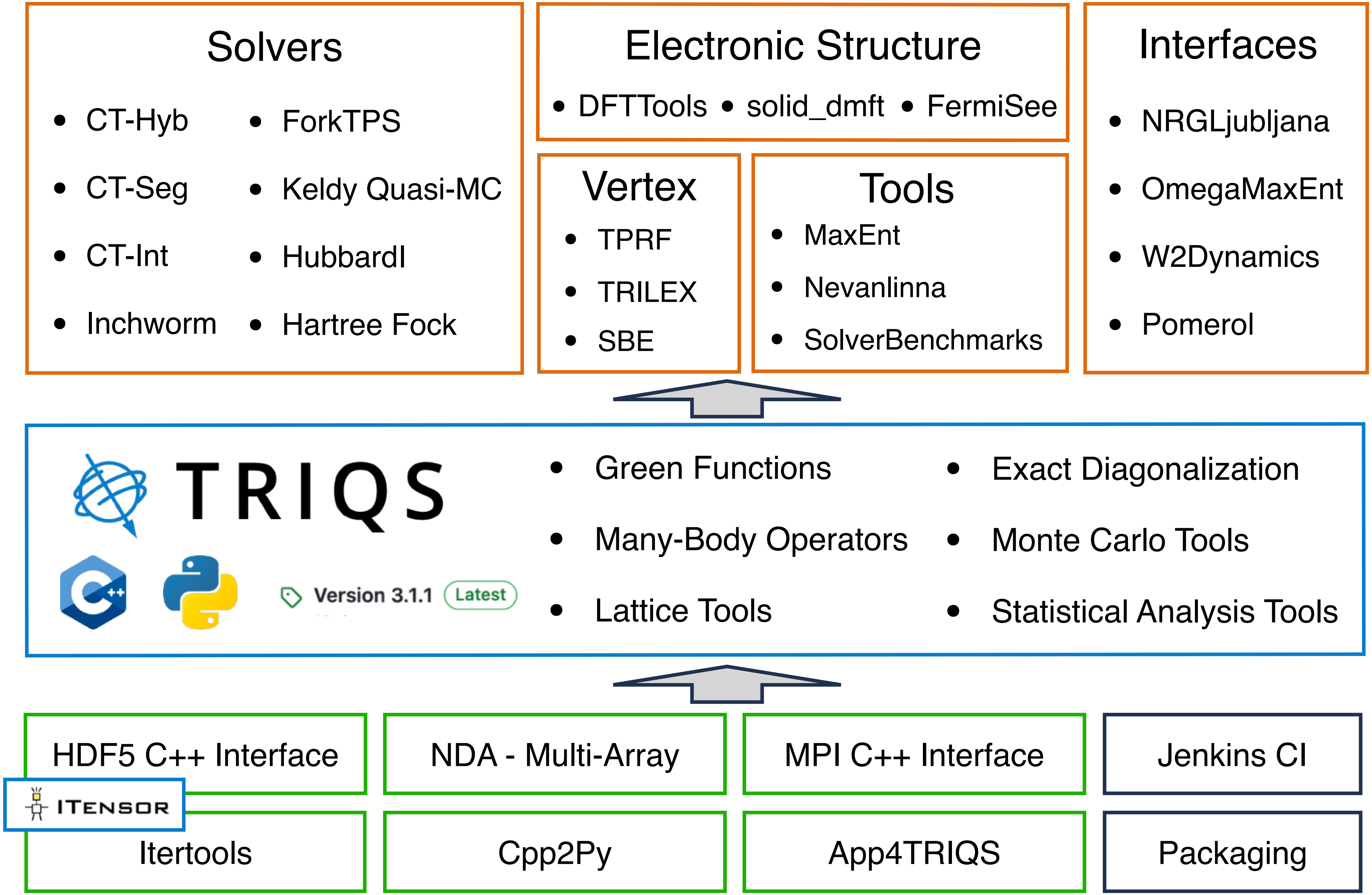
Itertools

Cpp2Py

App4TRIQS

Packaging

TRIQS — Software Stack



Solvers

- CT-Hyb
- CT-Seg
- CT-Int
- Inchworm
- ForkTPS
- Keldy Quasi-MC
- HubbardI
- Hartree Fock

Electronic Structure

- DFTTools
- solid_dmft
- FermiSee

Vertex

- TPRF
- TRILEX
- SBE

Tools

- MaxEnt
- Nevanlinna
- SolverBenchmarks

Interfaces

- NRGLjubljana
- OmegaMaxEnt
- W2Dynamics
- Pomerol

TRIQS

Version 3.1.1 Latest

- Green Functions
- Many-Body Operators
- Lattice Tools
- Exact Diagonalization
- Monte Carlo Tools
- Statistical Analysis Tools

HDF5 C++ Interface

NDA - Multi-Array

MPI C++ Interface

Jenkins CI

ITENSOR

Itertools

Cpp2Py

App4TRIQS

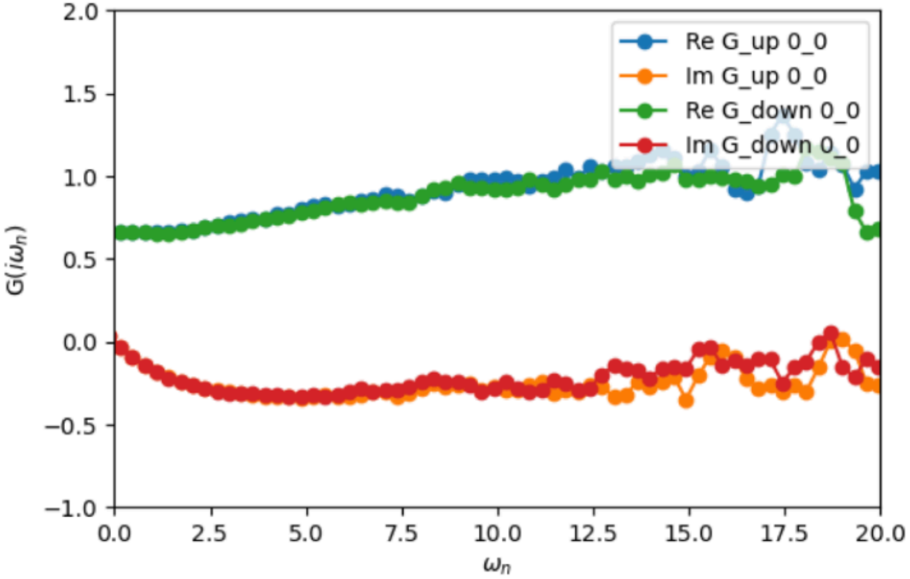
Packaging

TRIQS Applications — CT-Hyb QMC

triqs.github.io/cthyb

- Quantum Impurity Solver
- Hybridization Expansion
- Generic Multi-band/orbital Interactions
- Complex Interactions $\sum_{ijkl}\sum_{\sigma\sigma'} U^{\sigma\sigma'}_{ijkl} c^\dagger_{\sigma i} c^\dagger_{\sigma' j} c_{\sigma' k} c_{\sigma l}$

What can we measure?
$\langle \mathcal{T} c_{\sigma i}(\tau) c^\dagger_{\sigma j}\rangle$
$\langle \mathcal{T} c^\dagger_{\sigma i}(i\omega) c_{\sigma j}(i\omega') c^\dagger_{\sigma' k}(i\omega'') c_{\sigma' l}(0)\rangle$
$\langle \mathcal{T} A(\tau) B(0)\rangle$



P. Seth

I. Krivenko

M. Ferrero

H. Strand

O. Parcollet

A. Hampel

H. LaBollita

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 ~ 140 Citations

WIEN2k

WANNIER90

A. Hampel

S. Beck

M. Aichhorn

L. Pourovskii

V. Vildosola

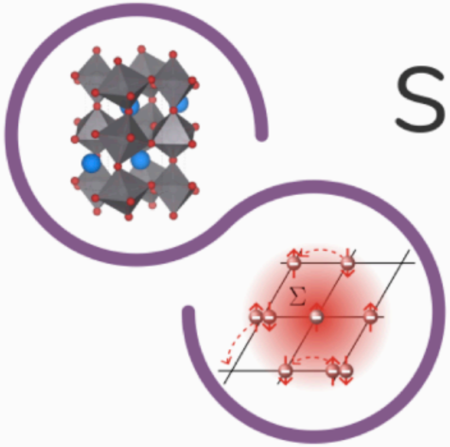
O. Peil

M. Zingl

M. Ferrero

G. Kraberger

J. Karp



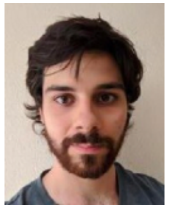
solid_dmft

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

triqs.github.io/solid_dmft/

M. Merkel et al. JoSS '22

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

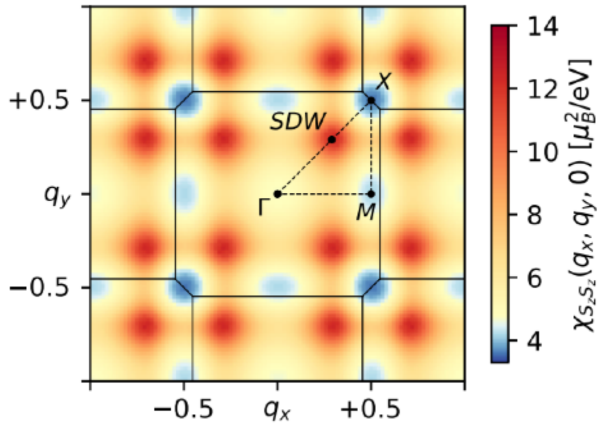
Y. in't Veld

M. Rösner

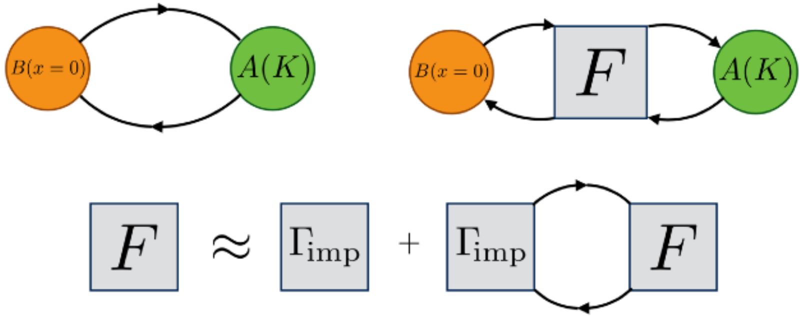
S. Kaeser

P. Hansmann

E. van Loon



H. Strand et al. PRB '19



- Lindhard Susceptibilities
- Random-phase Approximation
- GW Approximation
- Generalized Susceptibilities
- Bethe-Salpeter Equation Solver
- Vertex-Corrected Lattice Susceptibilities

TRIQS — Packaging

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

- Anaconda `conda install -c conda-forge triqs`
- Debian Packages for Ubuntu 22.04 and 24.04 `apt-get install triqs`
- Binder Notebook triqs.github.io/notebook
- Docker Image

```
docker pull flatironinstitute/triqs
docker run -p 8888:8888 flatironinstitute/triqs
```

- Singularity

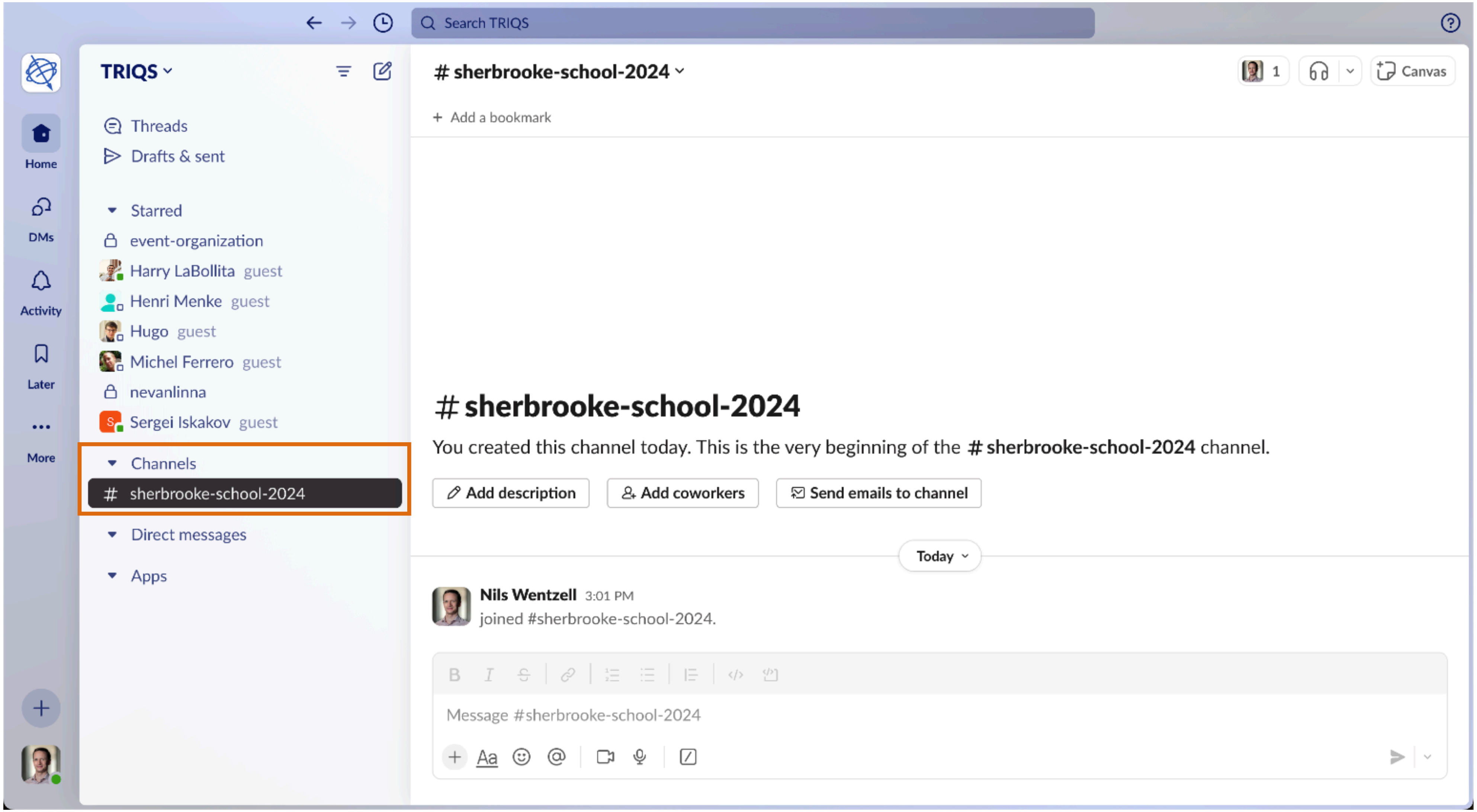
```
singularity pull docker://flatironinstitute/triqs
singularity exec triqs.sif python myscript.py
```

- EasyBuild `eb -r --software-name=TRIQS`

Versions 3.3
Soon!

TRIQS — Slack Channel

triqs.github.io/slack



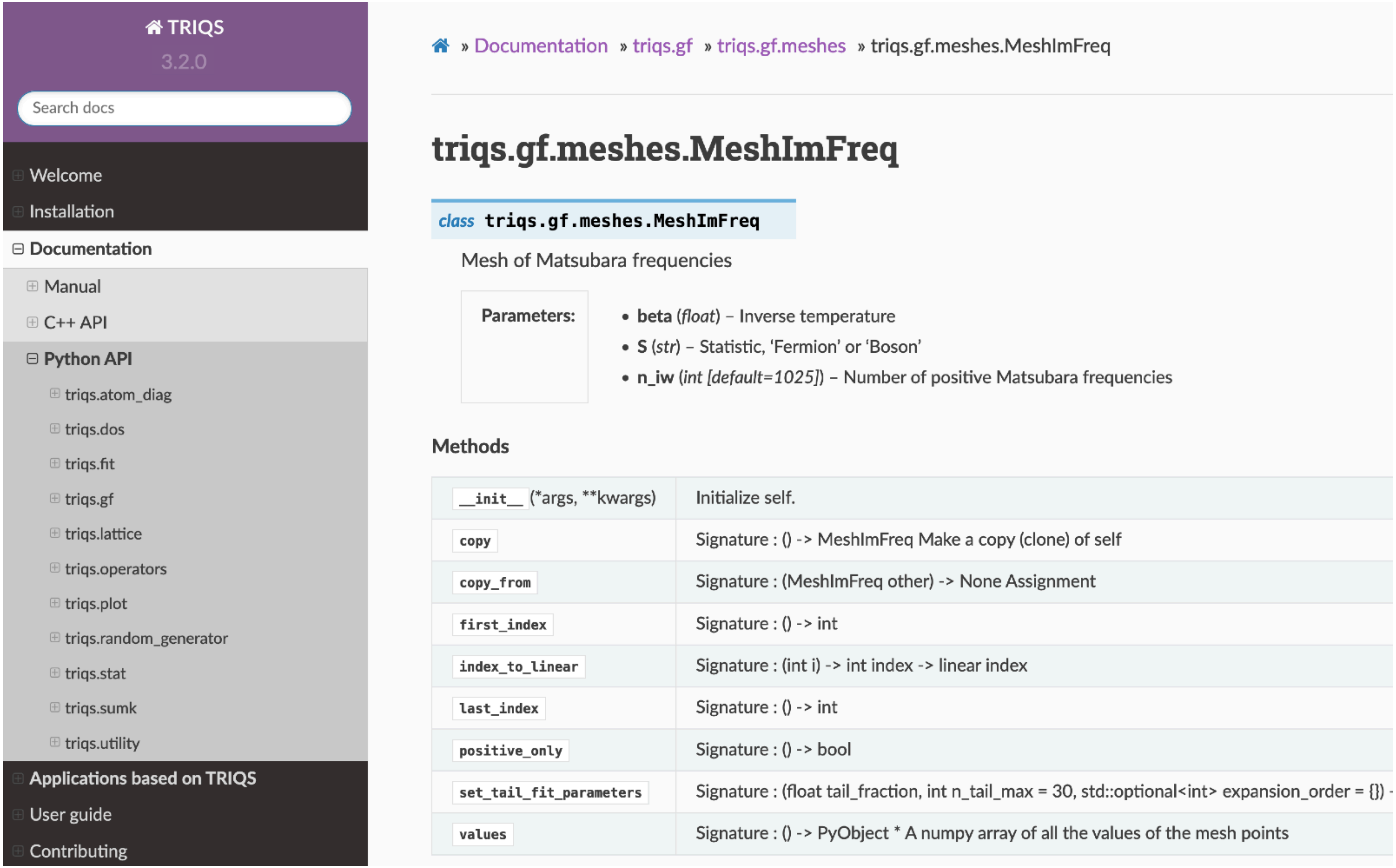
sherbrooke-school-2024

You created this channel today. This is the very beginning of the **# sherbrooke-school-2024** channel.

Nils Wentzell 3:01 PM
joined #sherbrooke-school-2024.

TRIQS Documentation

triqs.github.io



triqs.gf.meshes.MeshImFreq

class `triqs.gf.meshes.MeshImFreq`

Mesh of Matsubara frequencies

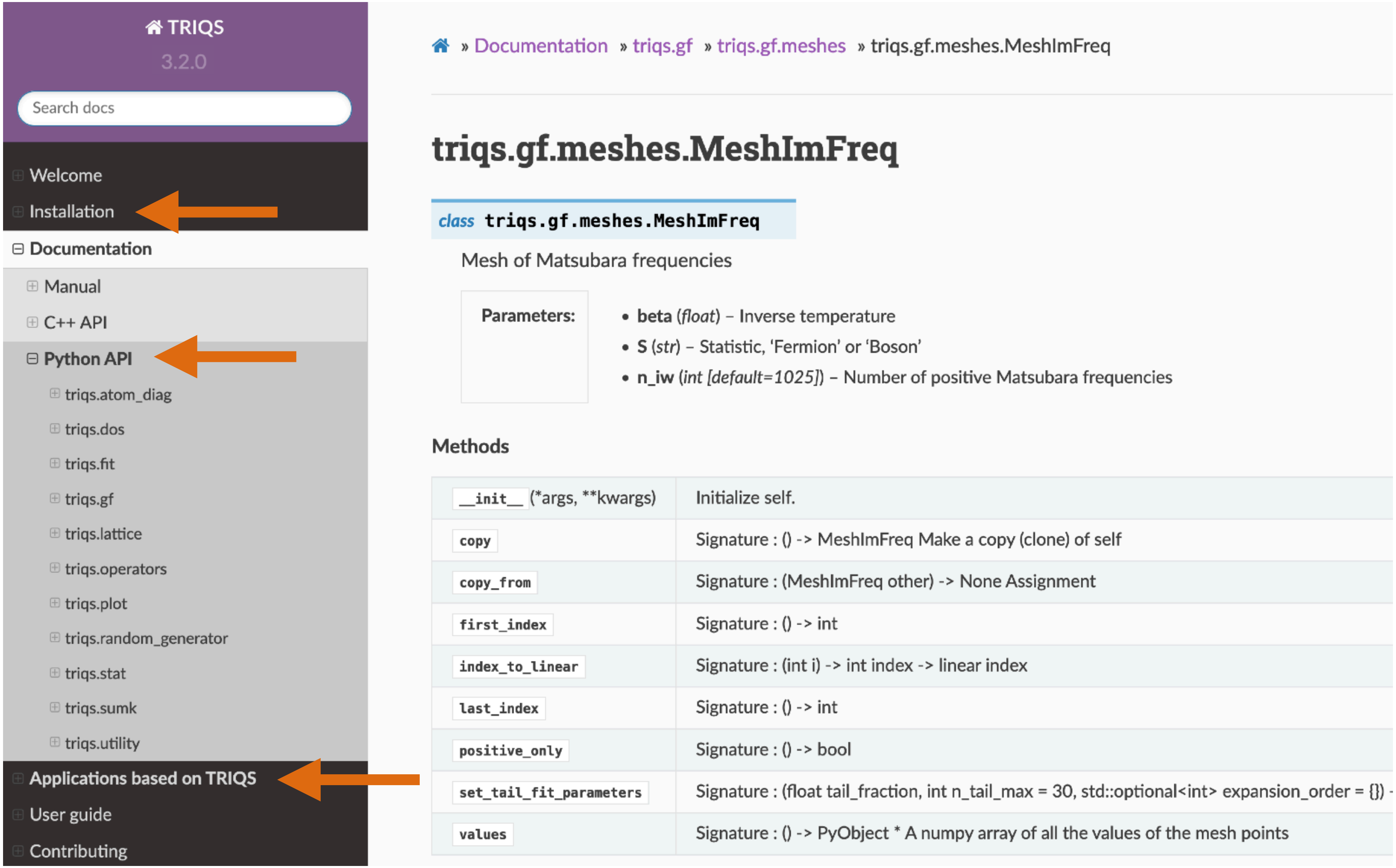
Parameters:
- **beta** (*float*) – Inverse temperature
- **S** (*str*) – Statistic, ‘Fermion’ or ‘Boson’
- **n_iw** (*int [default=1025]*) – Number of positive Matsubara frequencies

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 Documentation

triqs.github.io



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

triqs.gf.meshes.MeshImFreq

class `triqs.gf.meshes.MeshImFreq`

Mesh of Matsubara frequencies

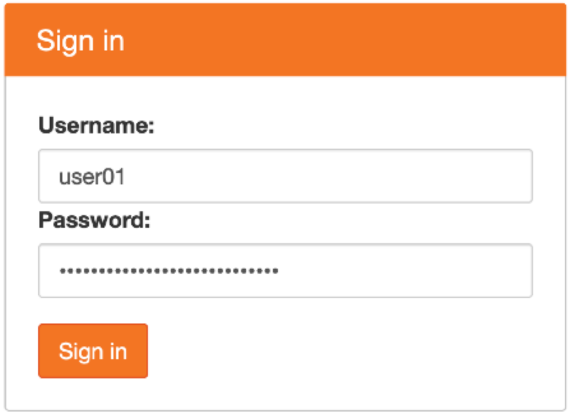
Parameters:
- **beta** (*float*) – Inverse temperature
- **S** (*str*) – Statistic, 'Fermion' or 'Boson'
- **n_iw** (*int [default=1025]*) – Number of positive Matsubara frequencies

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 — Getting Started

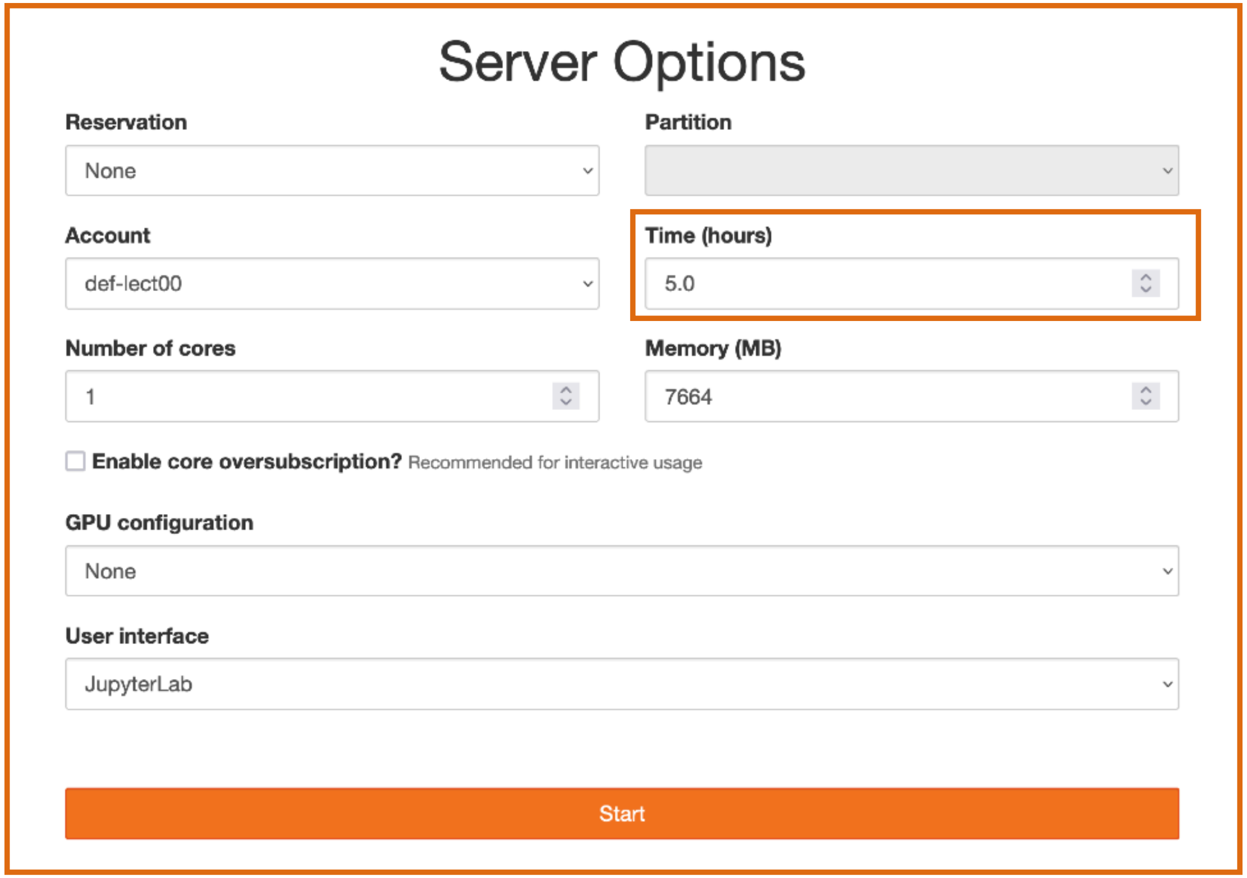
jupyter.quantum2024.ccs.usherbrooke.ca



Sign in

Username: user01

Password: ••••••••••••••••••••••••••••



Server Options

Reservation: None

Partition:

Account: def-lect00

Time (hours): 5.0

Number of cores: 1

Memory (MB): 7664

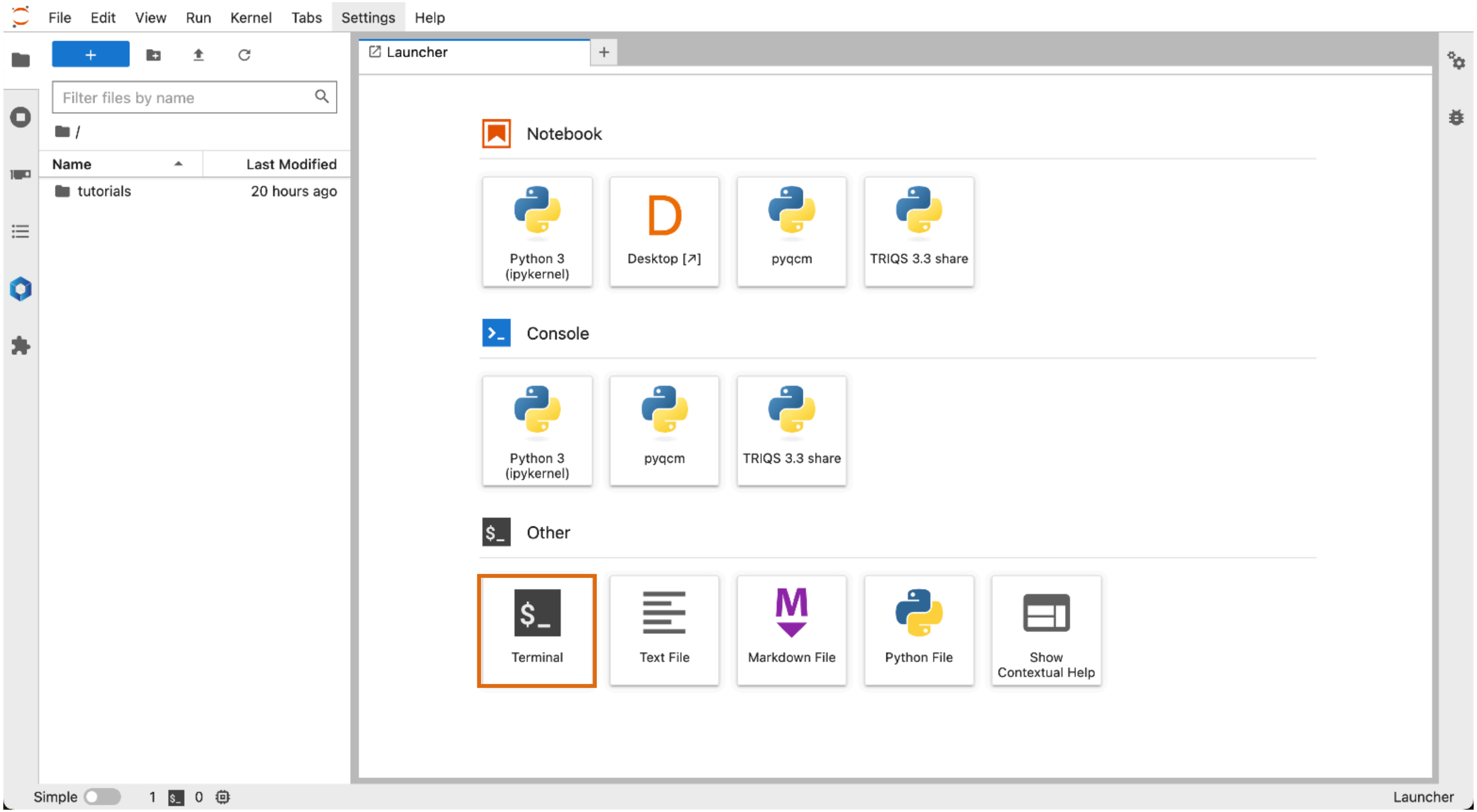
□ **Enable core oversubscription?** Recommended for interactive usage

GPU configuration: None

User interface: JupyterLab

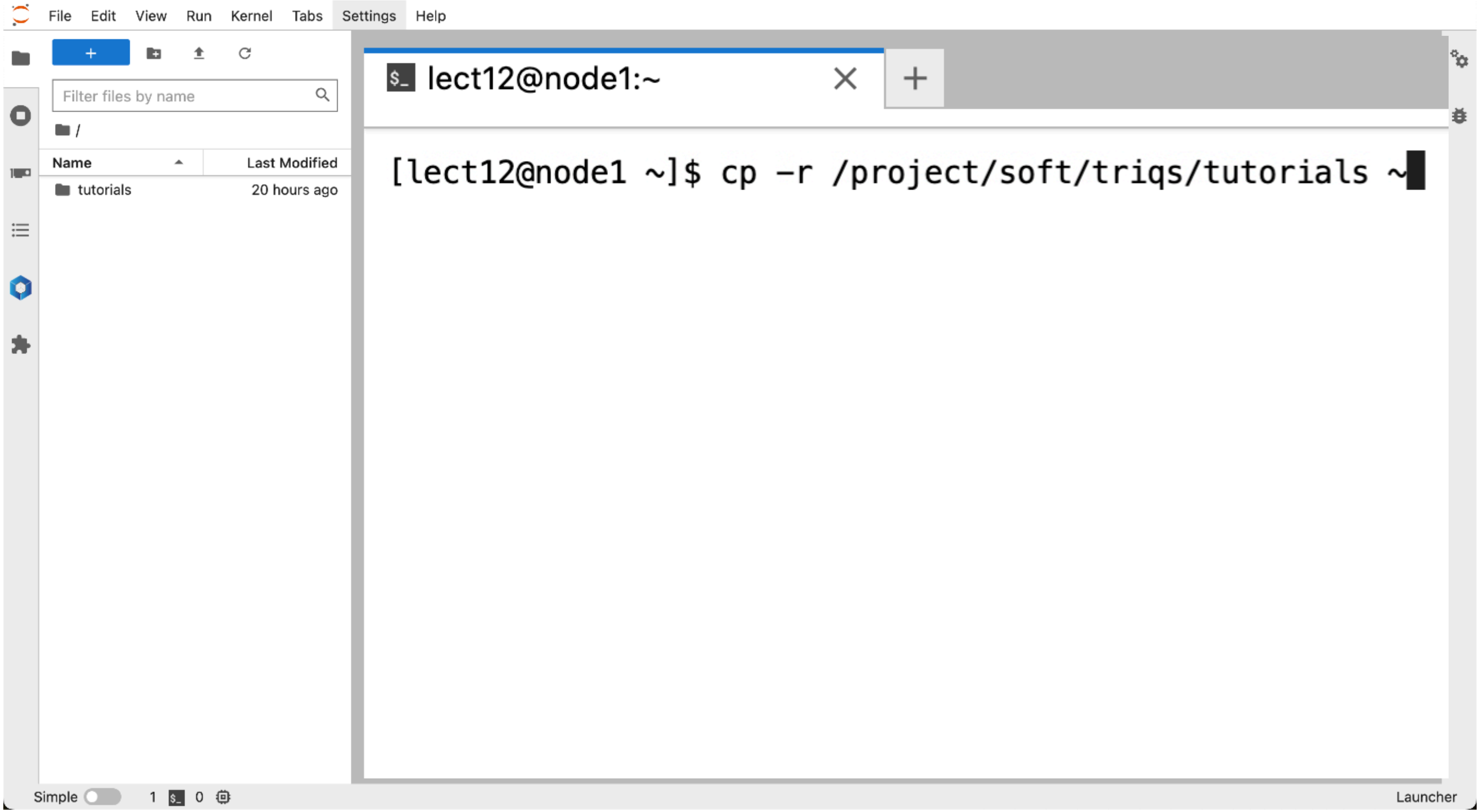
TRIQS — Getting Started

jupyter.quantum2024.ccs.usherbrooke.ca



TRIQS — Getting Started

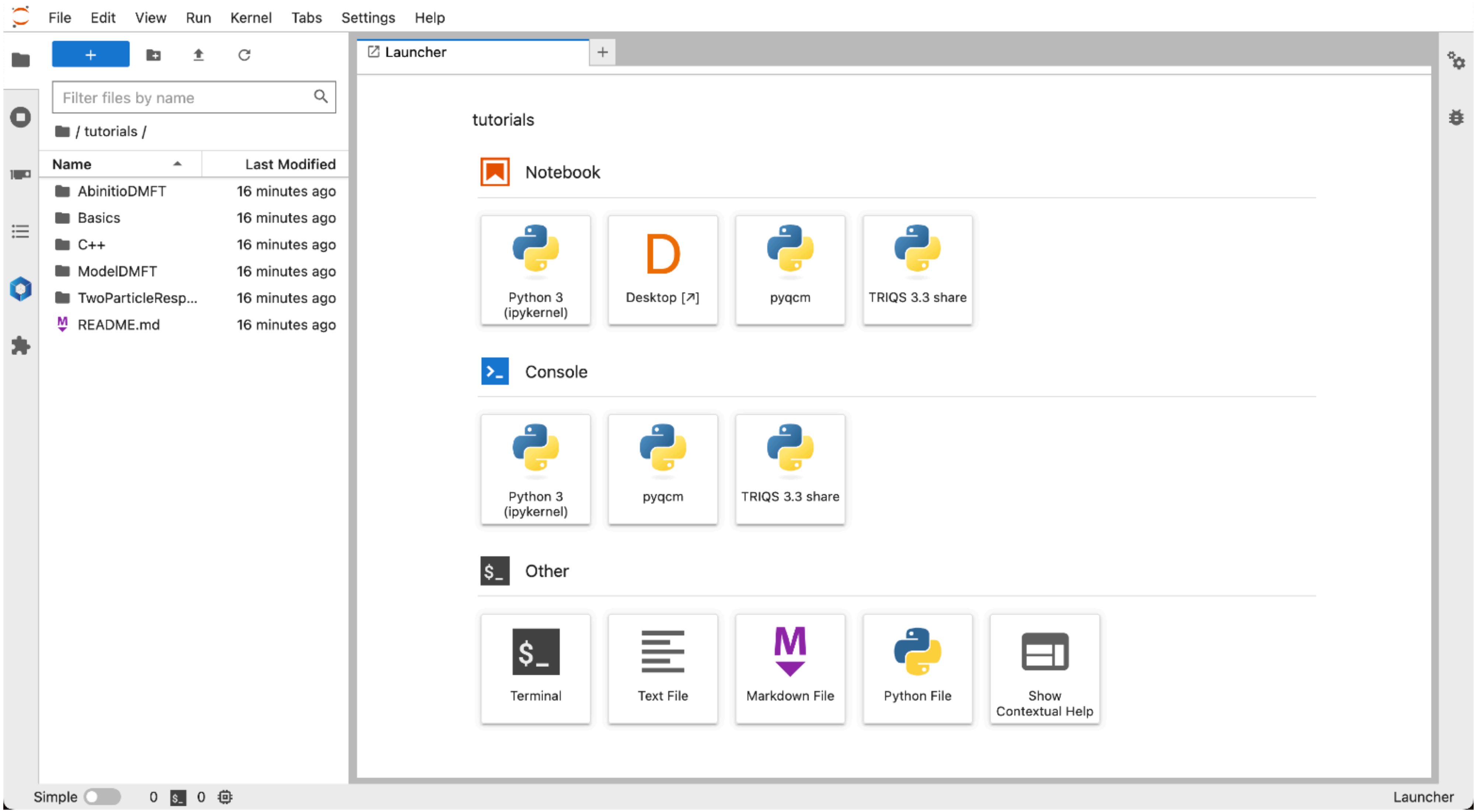
jupyter.quantum2024.ccs.usherbrooke.ca



```
[lect12@node1 ~]$ cp -r /project/soft/triqs/tutorials ~
```

TRIQS — Getting Started

jupyter.quantum2024.ccs.usherbrooke.ca



Name	Last Modified
AbinitioDMFT	16 minutes ago
Basics	16 minutes ago
C++	16 minutes ago
ModelDMFT	16 minutes ago
TwoParticleResp...	16 minutes ago
README.md	16 minutes ago

tutorials

Notebook

- Python 3 (ipykernel)
- Desktop [↗]
- pyqcm
- TRIQS 3.3 share

Console

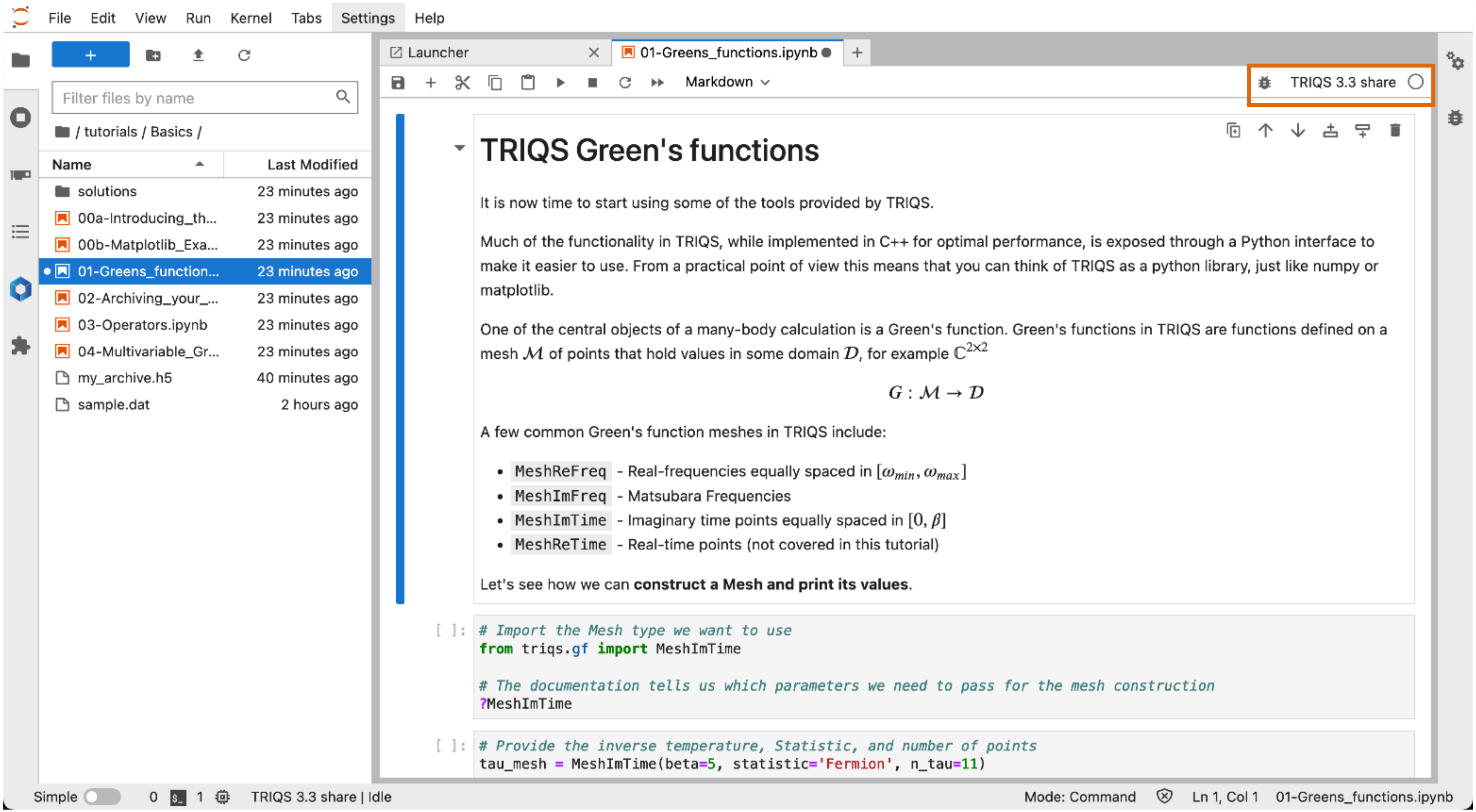
- Python 3 (ipykernel)
- pyqcm
- TRIQS 3.3 share

Other

- Terminal
- Text File
- Markdown File
- Python File
- Show Contextual Help

TRIQS — Getting Started

jupyter.quantum2024.ccs.usherbrooke.ca


File
Edit
View
Run
Kernel
Tabs
Settings
Help
Filter files by name
/ tutorials / Basics /
Name
Last Modified
solutions 23 minutes ago
00a-Introducing_th... 23 minutes ago
00b-Matplotlib_Exa... 23 minutes ago
01-Greens_function... 23 minutes ago
02-Archiving_your_... 23 minutes ago
03-Operators.ipynb 23 minutes ago
04-Multivariable_Gr... 23 minutes ago
my_archive.h5 40 minutes ago
sample.dat 2 hours ago
Launcher
01-Greens_functions.ipynb
Markdown
TRIQS 3.3 share
Simple
0
1
TRIQS 3.3 share | Idle
Mode: Command
Ln 1, Col 1
01-Greens_functions.ipynb

TRIQS Green's functions

It is now time to start using some of the tools provided by TRIQS.

Much of the functionality in TRIQS, while implemented in C++ for optimal performance, is exposed through a Python interface to make it easier to use. From a practical point of view this means that you can think of TRIQS as a python library, just like numpy or matplotlib.

One of the central objects of a many-body calculation is a Green's function. Green's functions in TRIQS are functions defined on a mesh $\mathcal{M}$ of points that hold values in some domain $\mathcal{D}$, for example $\mathbb{C}^{2x2}$

$$G : \mathcal{M} \rightarrow \mathcal{D}$$

A few common Green's function meshes in TRIQS include:

- `MeshReFreq` - Real-frequencies equally spaced in $[\omega_{min}, \omega_{max}]$
- `MeshImFreq` - Matsubara Frequencies
- `MeshImTime` - Imaginary time points equally spaced in $[0, \beta]$
- `MeshReTime` - Real-time points (not covered in this tutorial)

Let's see how we can **construct a Mesh and print its values**.

[ ]:
```
# Import the Mesh type we want to use
from triqs.gf import MeshImTime

# The documentation tells us which parameters we need to pass for the mesh construction
?MeshImTime
```

[ ]:
```
# Provide the inverse temperature, Statistic, and number of points
tau_mesh = MeshImTime(beta=5, statistic='Fermion', n_tau=11)
```

TRIQS Tutorials Summary

- Notebook Server

  jupyter.quantum2024.ccs.usherbrooke.ca

- TRIQS Tutorials

```
cp -r /project/soft/triqs/tutorials ~
```

- TRIQS Slack Workspace

  triqs.github.io/slack

- TRIQS Documentation

  triqs.github.io