CONDA and CONDA-FORGE

for users and maintainers

+ How they fit into the larger picture

Package, dependency and environment management for any language—Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN, and more.

Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

docs.conda.io

Evgeny's autotools example

environment.yml

```yaml
name: autotools
channels:
  - conda-forge
dependencies:
  - compilers
  - pkg-config
  - automake
  - hdf5
variables:
  CFLAGS: "-isystem ${CONDA_PREFIX}/include"
  CXXFLAGS: "-fPIC -isystem ${CONDA_PREFIX}/include"
  CPPFLAGS: " "
  LDFLAGS: " "
  LIBS: " "
```

- `name`: leave out to let user provide
- `channels`: leave out to use channel `defaults` maintained by anaconda
- `dependencies`: can also use `pip` to install dependencies not available in conda (see later)
- `variables`: will be set when entering the environment

```
conda env create -f environment.yml
conda activate autotools
```

https://trex-coe.github.io/hackathon

Reproducible paper example

A fully automated approach to calculate the melting temperature of elemental crystals

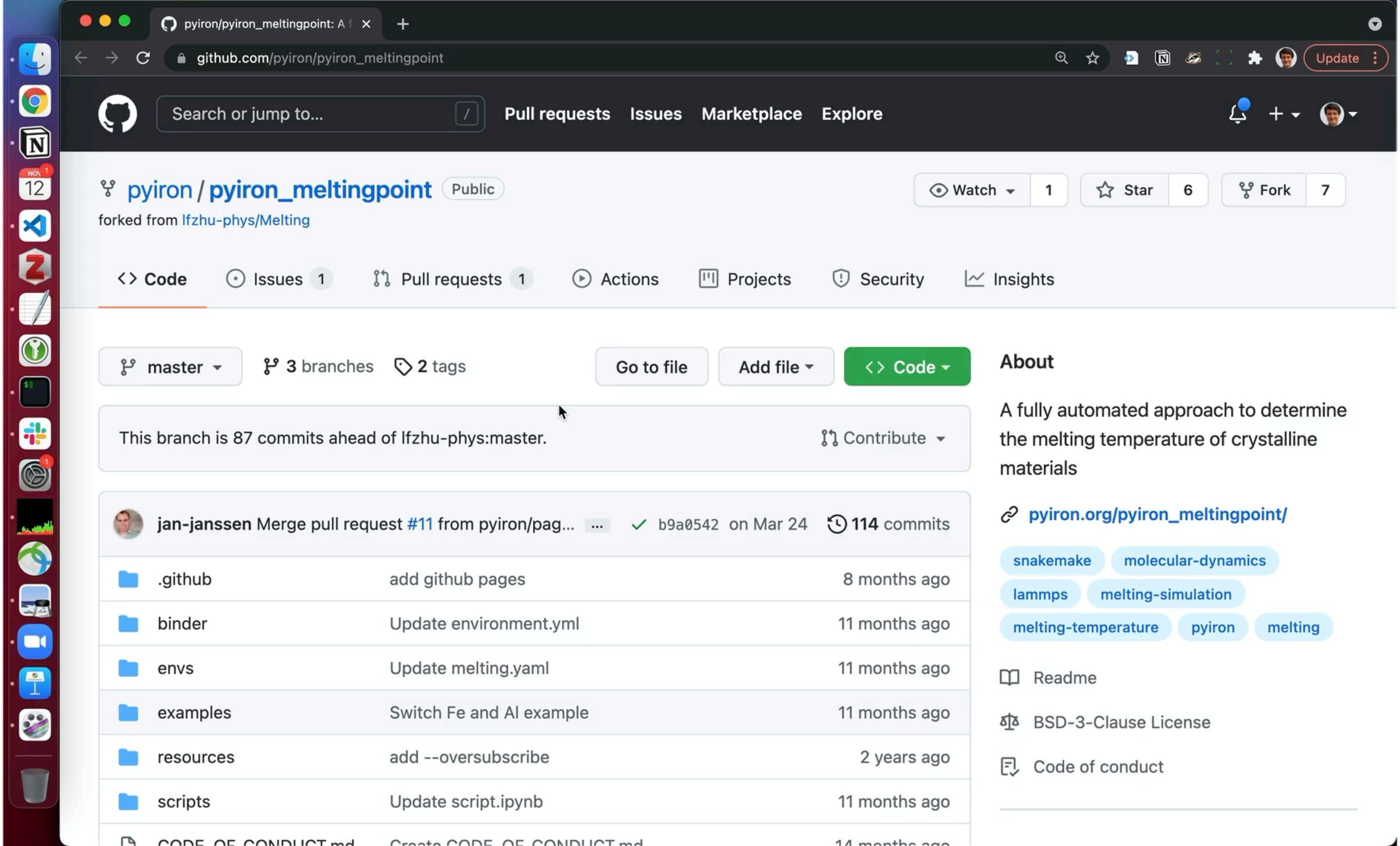
February 01, 2021

Read the paper

launch binder

DEMO

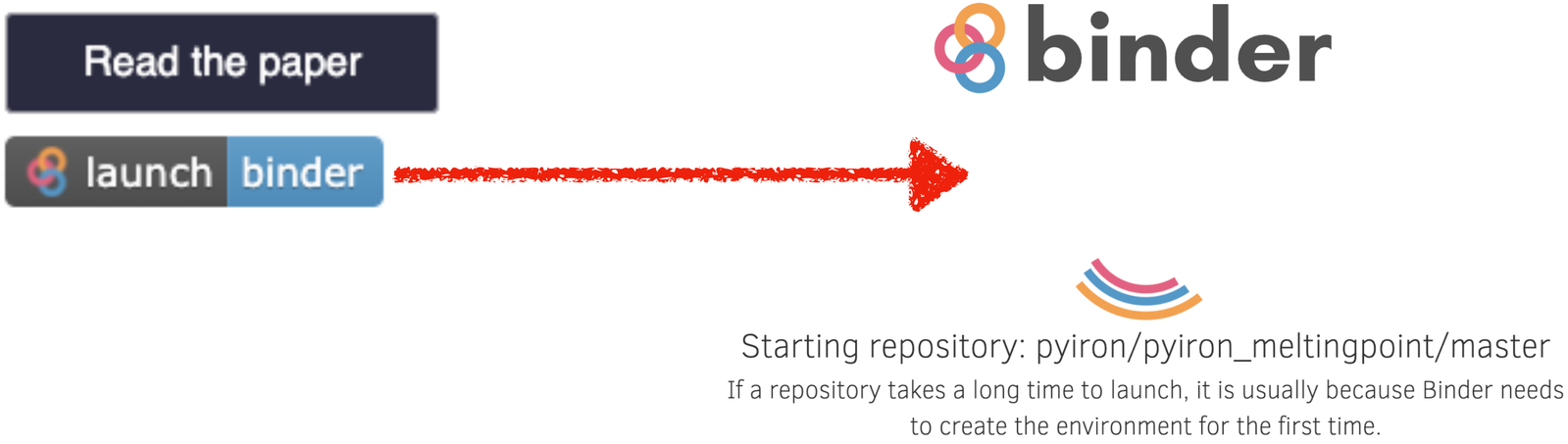
https://pyiron.org/publications/2021/02/01/melting.html


pyiron/pyiron_meltingpoint: A
github.com/pyiron/pyiron_meltingpoint
Update
Search or jump to...
Pull requests
Issues
Marketplace
Explore
pyiron / pyiron_meltingpoint
Public
forked from lfzhu-phys/Melting
Watch 1
Star 6
Fork 7
Code
Issues 1
Pull requests 1
Actions
Projects
Security
Insights
master
3 branches
2 tags
Go to file
Add file
Code
This branch is 87 commits ahead of lfzhu-phys:master.
Contribute
jan-janssen Merge pull request #11 from pyiron/pag...
b9a0542 on Mar 24
114 commits
.github add github pages 8 months ago
binder Update environment.yml 11 months ago
envs Update melting.yaml 11 months ago
examples Switch Fe and Al example 11 months ago
resources add --oversubscribe 2 years ago
scripts Update script.ipynb 11 months ago
About
A fully automated approach to determine the melting temperature of crystalline materials
pyiron.org/pyiron_meltingpoint/
snakemake
molecular-dynamics
lammps
melting-simulation
melting-temperature
pyiron
melting
Readme
BSD-3-Clause License
Code of conduct

Reproducible paper example

A fully automated approach to calculate the melting temperature of elemental crystals

February 01, 2021



https://pyiron.org/publications/2021/02/01/melting.html

conda & conda-forge

conda: package manager

- cross-platform: Linux, MacOS, Windows
- dependency resolution
- binaries, python packages

conda-forge: conda "channel"

- Open, community-led project
- Pre-configured CI setup for building new packages
  (managed through GitHub repositories)
- Easy to keep packages updated
  (automatic triggers for new releases)
- Many mat. sci. codes already there

MaX code/library	conda package added
Quantum Espresso	11/2020
Siesta	02/2018
Yambo	06/2021
fleur	in preparation
cp2k	01/2019
BigDFT	-
AiiDA	02/2019
CheSS	-
LaXLib & FFTXLib	-
Sirius	04/2021
DBCSR	-
COSMA	-
COSTA	10/2021
SpFFT & spla	04/2021

Data collected Nov 11, 2021

conda-forge statistics

- ~15k packages ("feedstocks") overall
  - cf. Guix: ~19k packages[1], many of which automatically imported
- Significant efforts in OSX + Windows + different CPU architectures
- 3.4k members of the conda-forge github team (mostly feedstock maintainers)
- 29 members of staged-recipes team reviewing creation of new feedstocks

[1] https://twitter.com/GuixHpc/status/1458796042074443783?s=20

Architecture	Package versions	Repo size [GB]
noarch	74k	87
linux-64 (x86)	221k	1573
linux-aarch64	38k	264
linux-ppc64le	36k	234
osx-64 (x86)	206k	878
osx-arm64	13k	54
win-64 (x86)	144k	602
TOTAL		**3691**

Data collected Nov 11, 2021

Your package on conda-forge

To submit a package to the `conda-forge` channel, add its `recipe` and licence to the `staged-recipes` repository and create a pull request. Once the pull request is merged, the package becomes available on the `conda-forge` channel. Note that contributing a package makes you the `maintainer` of that package.

Let's have a quick look at

Staged recipes: https://github.com/conda-forge/staged-recipes
PR for fleur code: https://github.com/conda-forge/staged-recipes/pull/14726

https://conda-forge.org/docs/maintainer/adding_pkgs.html

conda & conda-forge

Pros

- Specify environment (& to a lesser degree build recipes) in simple, intuitive yaml format
- packages are precompiled => installation is fast
- packages are typically more up to date than those from the OS
- large & friendly community
- Default compiler flags on conda-forge set rpath
  => executables will typically also work when called outside environment

Cons

- No "build from source" as a user if package is missing for your OS
- Less reproducibility guarantees than `nix` - e.g. builds can in principle access the internet (even if they shouldn't need to)
- Focus on portability over performance (e.g. default microarchitecture on linux-64: `nocona` from 2004, no AVX)
- Dependency resolution can be slow;
  Unhelpful error messages when dependency resolution fails

package managers vs build systems

Abstract definition of a build system [1]
A *build system* takes a *task description*, a *target key*, and a *key-value store* and returns a new store, in which the target key (and all its dependencies) have up-to-date values.

- `make`: store=file system, key=filename, value=file content,
  task description=Makefile rules
- `Excel`: store=worksheet, key=cell index (A1), value=number/string,
  task description=formulae in the worksheet
- `conda`: store=environment, key=package name, value=package content,
  task description=environment specification (including package versions, variants, ...)

[1] Mokhov, A.; Mitchell, N.; Jones, S. P. Build Systems à La Carte: Theory and Practice.
Journal of Functional Programming **2020**, *30*. https://doi.org/10.1017/S0956796820000088.

build systems

Table 1. *Build system differences*

Build system	Persistent build information	Scheduler	Dependencies	Minimal	Cutoff	Cloud
MAKE	File modification times	Topological	Static	Yes	No	No
EXCEL	Dirty cells, calc chain	Restarting	Dynamic	No	No	No
SHAKE	Previous dependency graph	Suspending	Dynamic	Yes	Yes	No
BAZEL	Cloud cache, command history	Restarting	Dynamic*	No	Yes	Yes

*At present, user-defined build rules cannot have dynamic dependencies.

[1] Mokhov, A.; Mitchell, N.; Jones, S. P. Build Systems à La Carte: Theory and Practice.
Journal of Functional Programming **2020**, *30*. https://doi.org/10.1017/S0956796820000088.

package managers

	Supported operating systems	Public binary catalogues	Reproducibility guarantees	FOSS packages for atomistic simulation [1]
easybuild	Linux	No	Weak	21/24
Spack	Linux, MacOS	No	Weak	19/24
CONDA-FORGE	Linux, MacOS, Windows	Yes	Medium	17/24
Nix	Linux, MacOS	Yes	High	6/24
Guix	Linux	Yes	High	1/24

As of 10/2021

[1] 24 FOSS simulation codes from http://atomistic.software

Questions

- Experience to share with package managers in HPC?
- Which are the trends you see from your perspective?
  (also: packages vs containers)
- Requirements a package manager for HPC must fulfil?
- Can we identify 1-2 promising solutions where it makes sense to focus our efforts (quantum chemistry + materials science)?

  I.e. by providing up-to-date build recipes, optimized for different platforms, ...

Contact leopold.talirz@gmail.com

Acknowledgements

Supported by

MARVEL NCCR
nccr-marvel.ch

MaX Centre
max-centre.eu

H2020 Dome 4.0 (starting 2021)

H2020 OpenModel (starting 2021)

SNSF
snf.ch

EPFL
epfl.ch

H2020 NEP (starting 2021)

swissuniversities
swissuniversities.ch
Go to project page

BIG-MAP
big-map.eu

H2020 MarketPlace
the-marketplace-project.eu

H2020 Intersect
intersect-project.eu

OSSCAR
osscar.org, EPFL Open Science Fund

ERC
erc.europa.eu

Partners - Supercomputing Centers

CSCS
cscs.ch

CINECA
cineca.it

PASC
pasc-ch.org

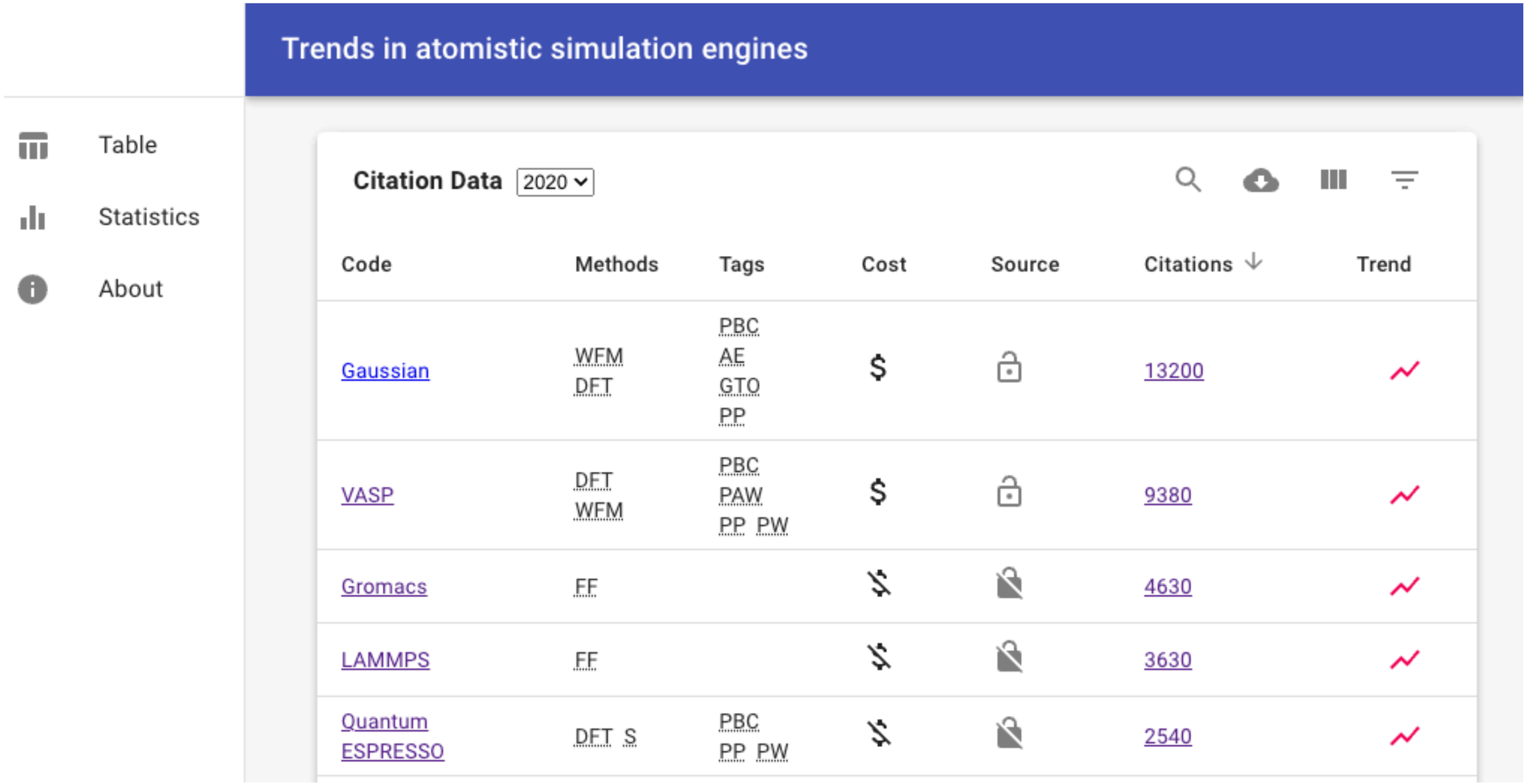
PRACE
prace-ri.eu

FZ-Jülich
fz-juelich.de

NFFA
nffa.eu

EMMC
emmc.info

atomistic.software



Swiss National Science Foundation
MARVEL NATIONAL CENTRE OF COMPETENCE IN RESEARCH
MAX DRIVING THE EXASCALE TRANSITION
swissuniversities
EPFL