High-throughput T_2 calculations

Michael Y. Toriyama

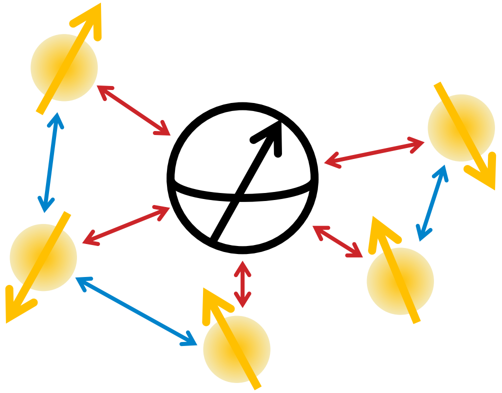
Argonne National Laboratory
Chicago, IL, USA

Cluster correlation expansion (CCE)

- Spin qubit dynamics are determined by the nuclear spin environment

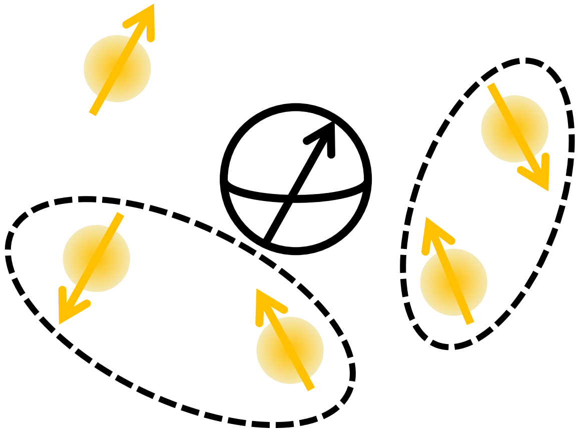
$$\widehat{H} = \boldsymbol{SDS} + \boldsymbol{B}\gamma_S\boldsymbol{S} + \sum_i \boldsymbol{B}\gamma_i\boldsymbol{I}_i + \boldsymbol{SA}_i\boldsymbol{I}_i + \boldsymbol{I}_i\boldsymbol{QI}_i + \sum_{j<i}\boldsymbol{I}_i\boldsymbol{J}_{ij}\boldsymbol{I}_j$$

Zero-field splitting	Zeeman (qubit)	Zeeman (nuclear)	Hyperfine	Quadrupole	Dipolar



- Coherence of the spin qubit can be simulated using the cluster correlation expansion (CCE) approach

$$\mathcal{L}(t) = \frac{\langle 1|\hat{\rho}_S(t)|0\rangle}{\langle 1|\hat{\rho}_S(0)|0\rangle} \approx \prod_i \tilde{\mathcal{L}}_{\{i\}}(t) \prod_{i,j} \tilde{\mathcal{L}}_{\{ij\}}(t) \ldots$$



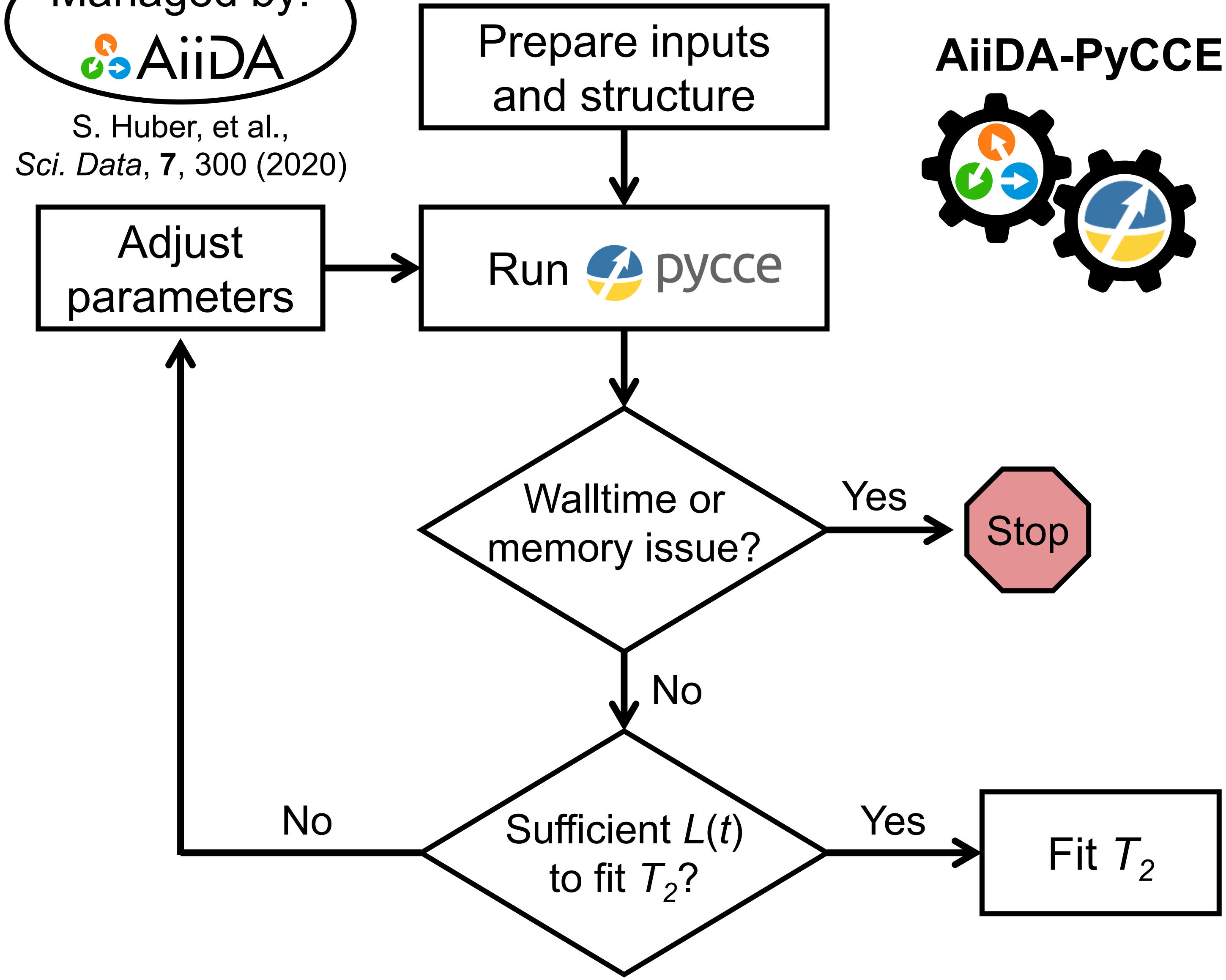
Open-source CCE code: pycce

M. Onizhuk and G. Galli, *Adv. Theory Simul.*, **4**, 2100254 (2021)
M. Onizhuk and G. Galli, *Rev. Mod. Phys.*, **97**, 021001 (2025)

Coherence time (T_2) calculation workflow

Managed by:
AiiDA

S. Huber, et al.,
Sci. Data, **7**, 300 (2020)


Prepare inputs and structure
Adjust parameters
Run pycce
Walltime or memory issue?
Yes
Stop
No
Sufficient $L(t)$ to fit T_2?
No
Yes
Fit T_2

AiiDA-PyCCE

High-throughput

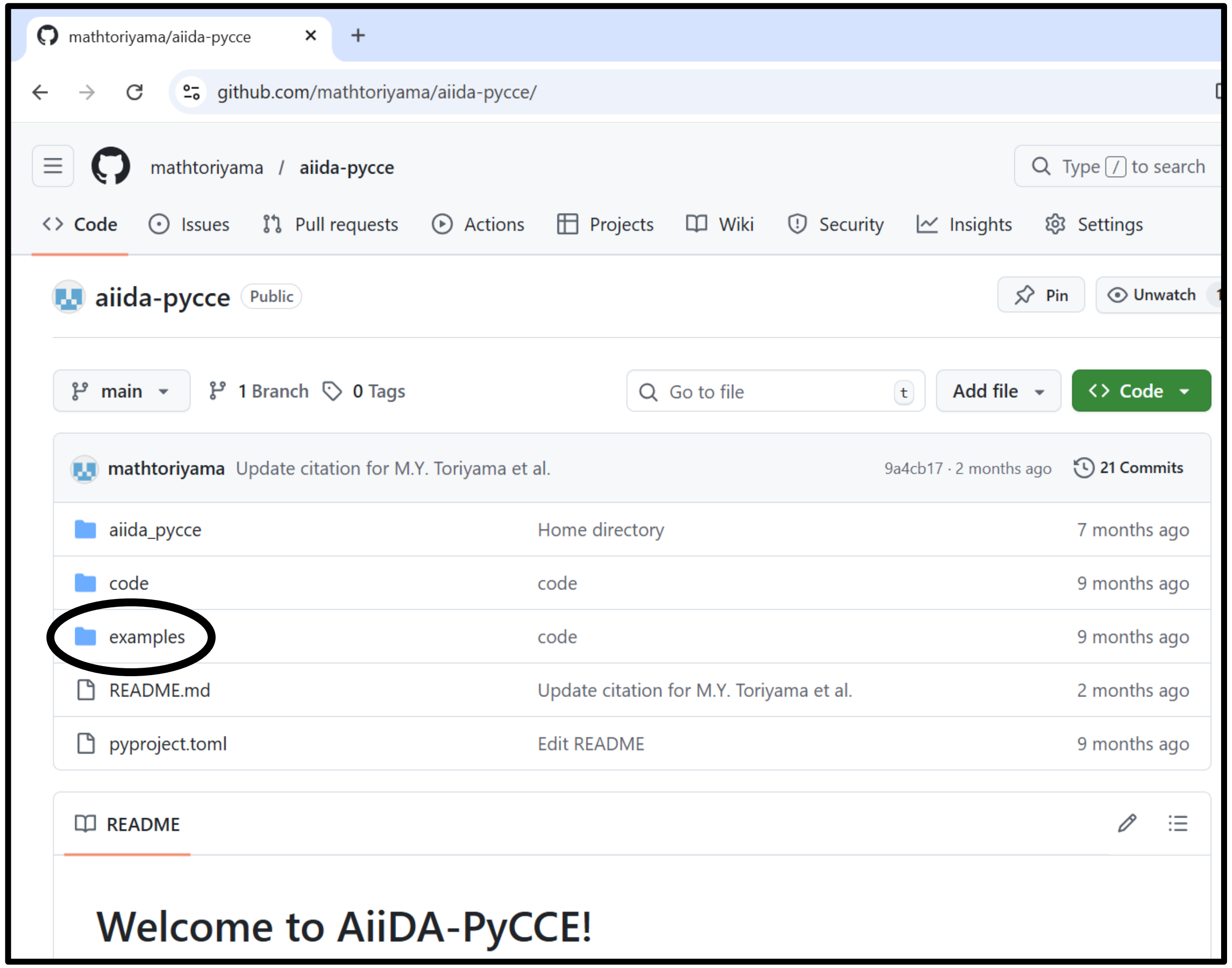
Tracking >1000s of calculations at once, for streamlined data management

Consistent and flexible template

Only need *structure* and *basic CCE inputs* (expansion order, nuclear spin bath size, etc.)

M. Toriyama, et al., *npj 2D Mater. Appl.*, **9**, 108 (2025)

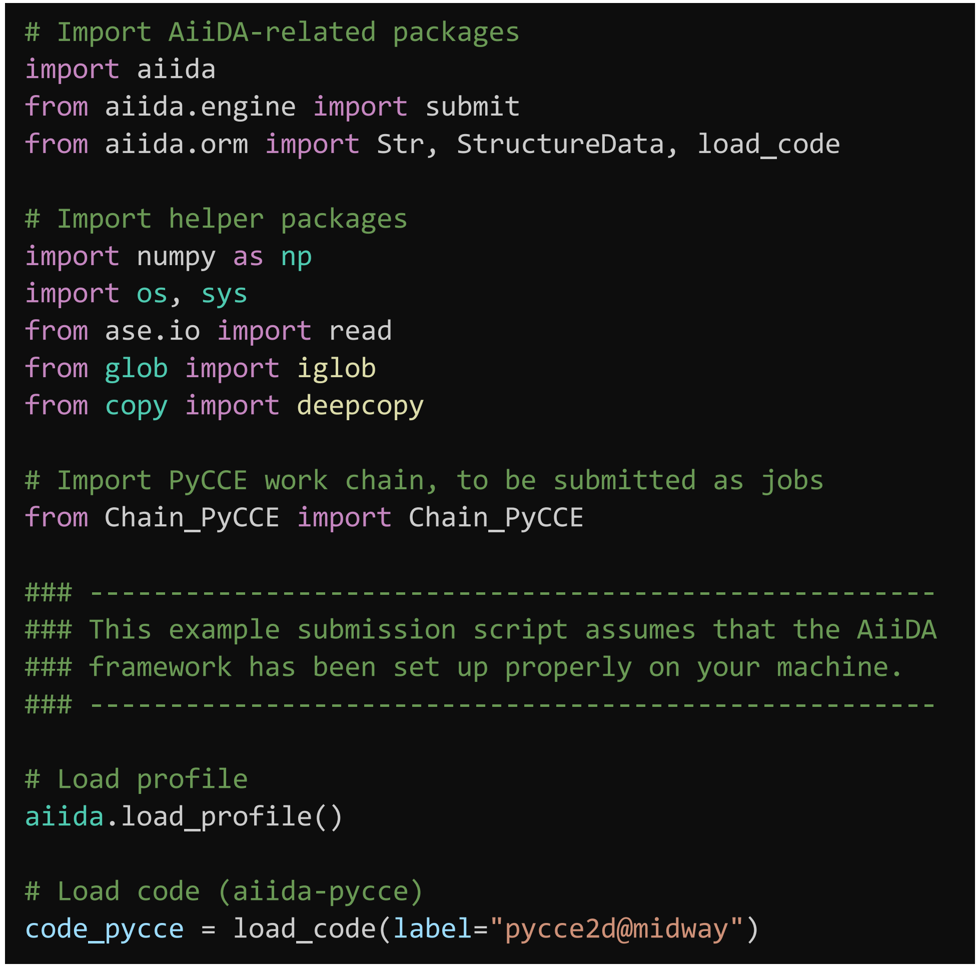
Example job submission script



mathtoriyama	Update citation for M.Y. Toriyama et al.	9a4cb17 · 2 months ago 21 Commits
aiida_pycce	Home directory	7 months ago
code	code	9 months ago
examples	code	9 months ago
README.md	Update citation for M.Y. Toriyama et al.	2 months ago
pyproject.toml	Edit README	9 months ago

Welcome to AiiDA-PyCCE!

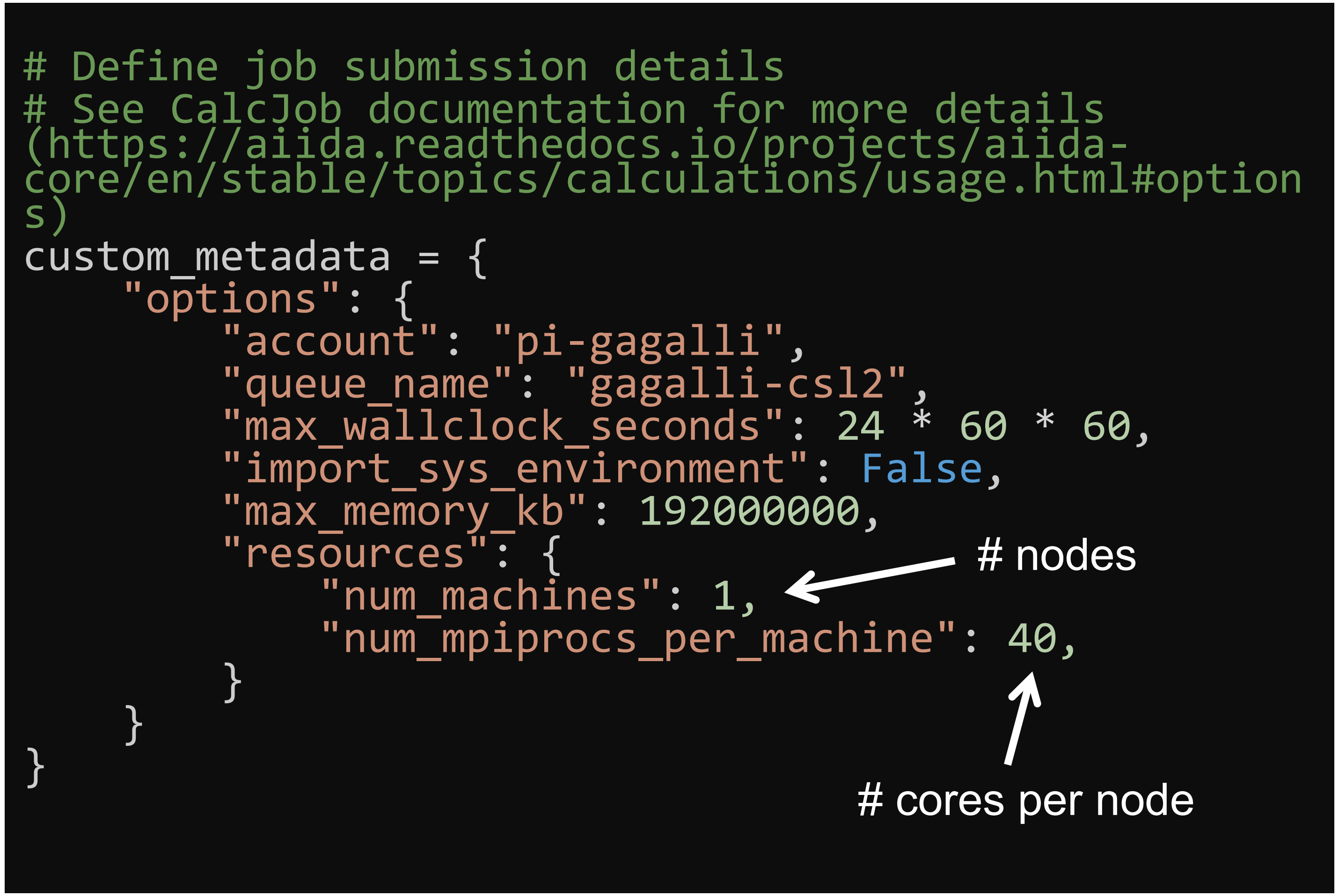
examples/HT_2D_Hosts.py



```python
# Import AiiDA-related packages
import aiida
from aiida.engine import submit
from aiida.orm import Str, StructureData, load_code

# Import helper packages
import numpy as np
import os, sys
from ase.io import read
from glob import iglob
from copy import deepcopy

# Import PyCCE work chain, to be submitted as jobs
from Chain_PyCCE import Chain_PyCCE

### -----------------------------------------------------
### This example submission script assumes that the AiiDA
### framework has been set up properly on your machine.
### -----------------------------------------------------

# Load profile
aiida.load_profile()

# Load code (aiida-pycce)
code_pycce = load_code(label="pycce2d@midway")
```

https://github.com/mathtoriyama/aiida-pycce

Define job submission details



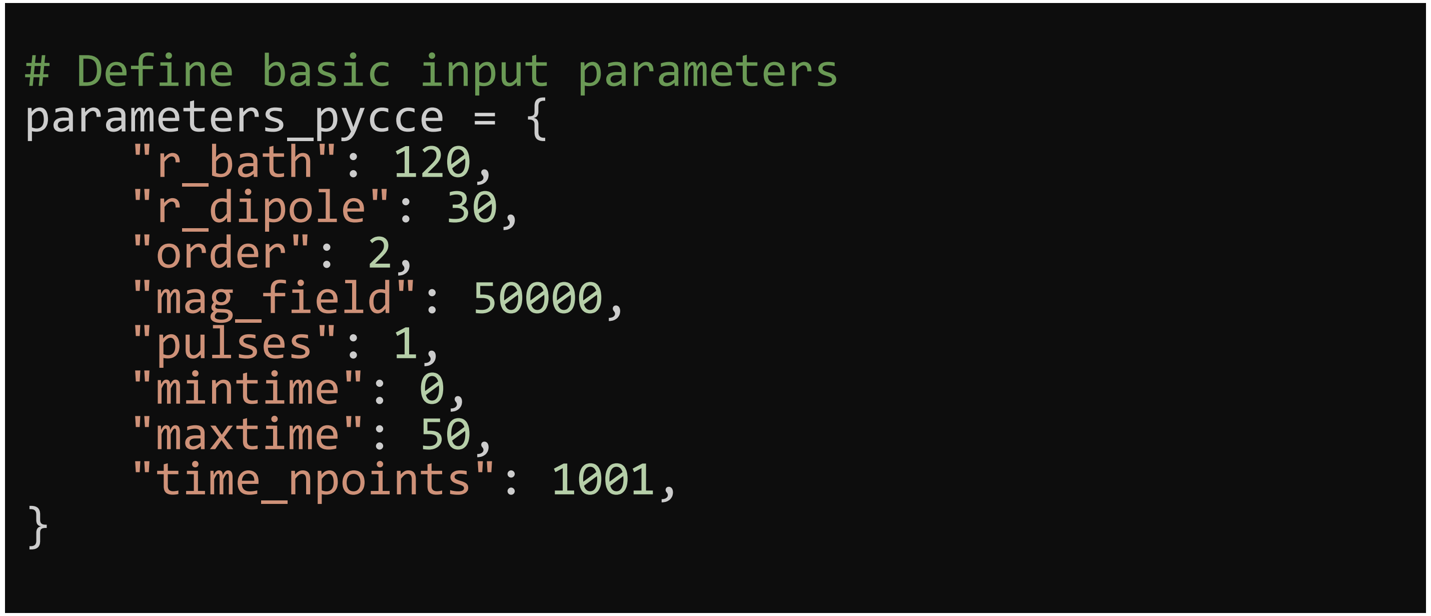
```python
# Define job submission details
# See CalcJob documentation for more details (https://aiida.readthedocs.io/projects/aiida-core/en/stable/topics/calculations/usage.html#options)
custom_metadata = {
    "options": {
        "account": "pi-gagalli",
        "queue_name": "gagalli-csl2",
        "max_wallclock_seconds": 24 * 60 * 60,
        "import_sys_environment": False,
        "max_memory_kb": 192000000,
        "resources": {
            "num_machines": 1,
            "num_mpiprocs_per_machine": 40,
        }
    }
}
```

slurm
workload manager

Submit 1 CCE simulation per core.

https://github.com/mathtoriyama/aiida-pycce

Set up PyCCE inputs



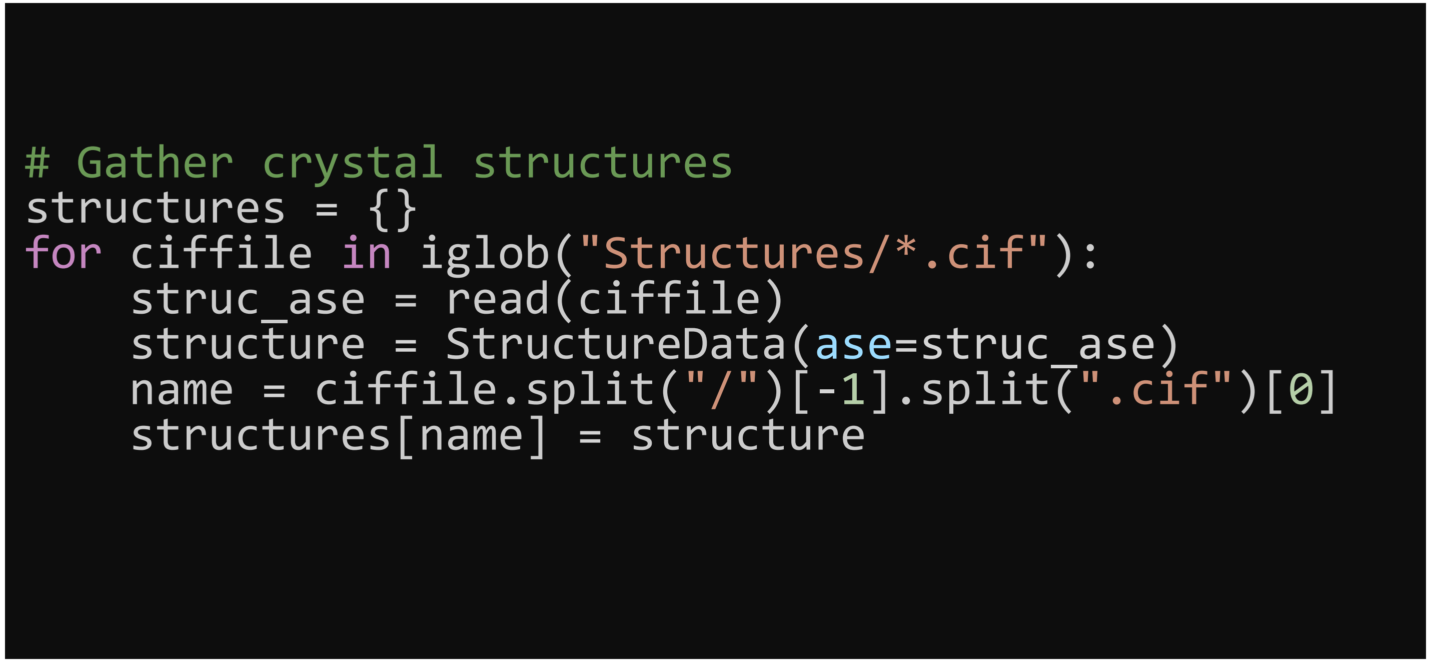
```
# Define basic input parameters
parameters_pycce = {
    "r_bath": 120,
    "r_dipole": 30,
    "order": 2,
    "mag_field": 50000,
    "pulses": 1,
    "mintime": 0,
    "maxtime": 50,
    "time_npoints": 1001,
}
```

Consistent and flexible template

Only need *structure* and *basic CCE inputs* (expansion order, nuclear spin bath size, etc.)

https://github.com/mathtoriyama/aiida-pycce

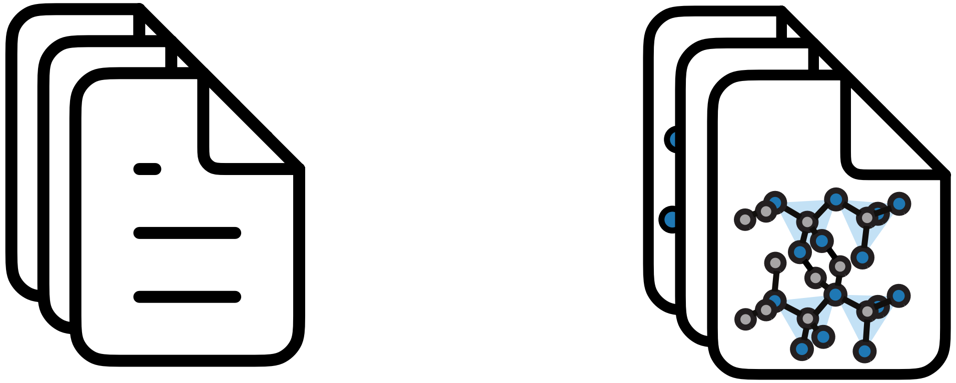
Gather and “codify” crystal structures



```python
# Gather crystal structures
structures = {}
for ciffile in iglob("Structures/*.cif"):
    struc_ase = read(ciffile)
    structure = StructureData(ase=struc_ase)
    name = ciffile.split("/")[-1].split(".cif")[0]
    structures[name] = structure
```

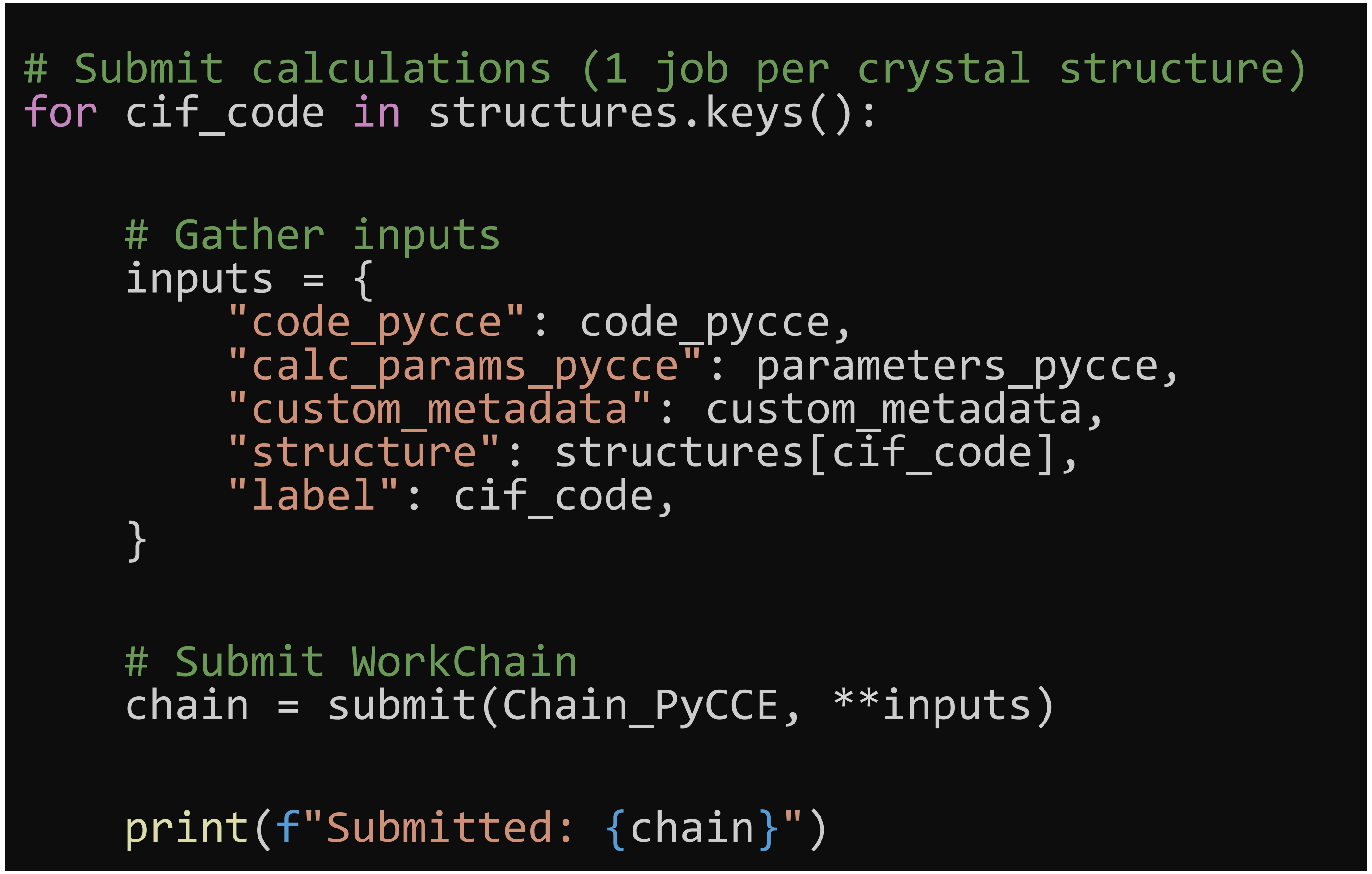
Consistent and flexible template

Only need *structure* and *basic CCE inputs* (expansion order, nuclear spin bath size, etc.)



https://github.com/mathtoriyama/aiida-pycce

Submit jobs

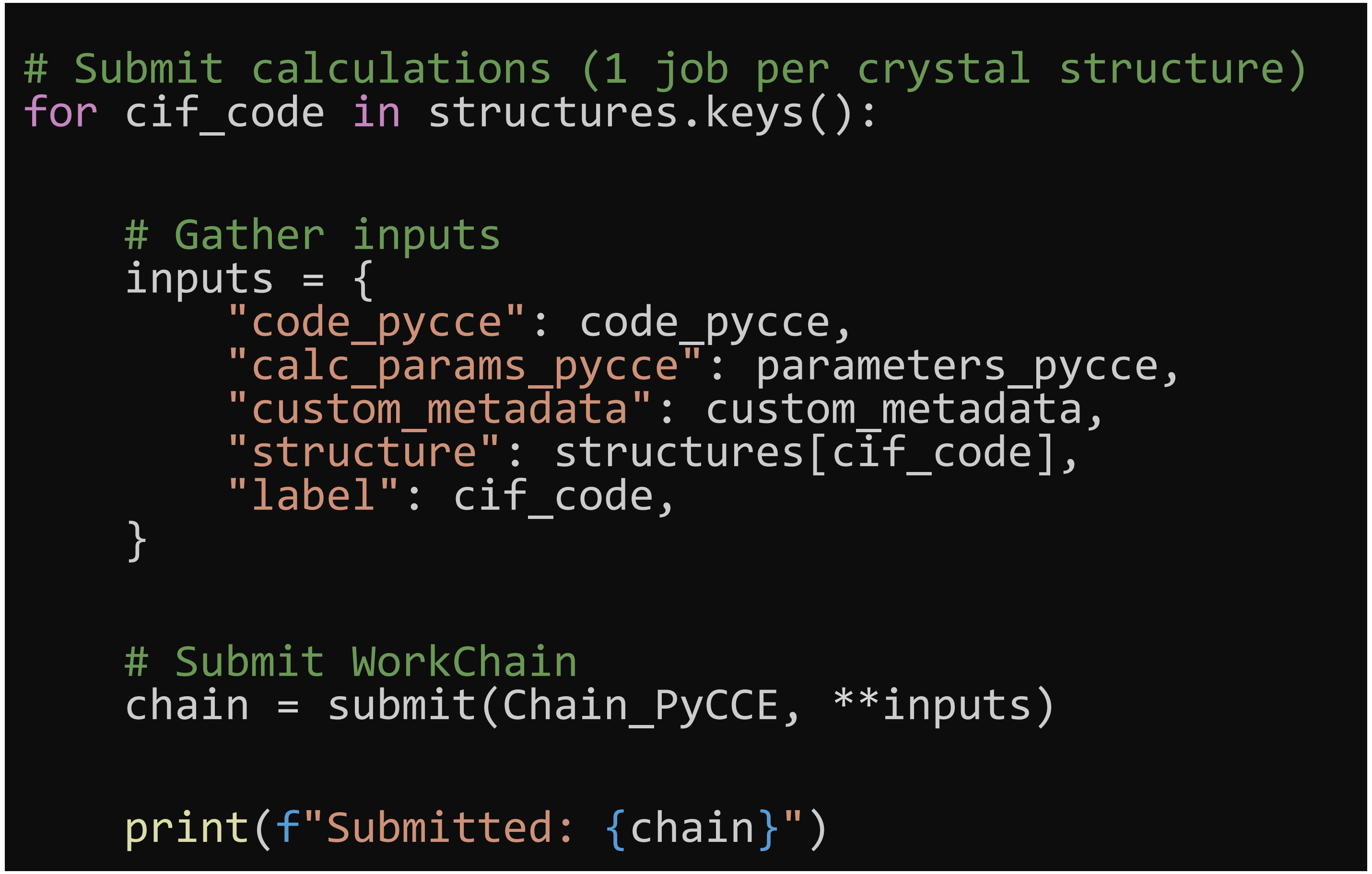


```python
# Submit calculations (1 job per crystal structure)
for cif_code in structures.keys():

    # Gather inputs
    inputs = {
        "code_pycce": code_pycce,
        "calc_params_pycce": parameters_pycce,
        "custom_metadata": custom_metadata,
        "structure": structures[cif_code],
        "label": cif_code,
    }

    # Submit WorkChain
    chain = submit(Chain_PyCCE, **inputs)

    print(f"Submitted: {chain}")
```

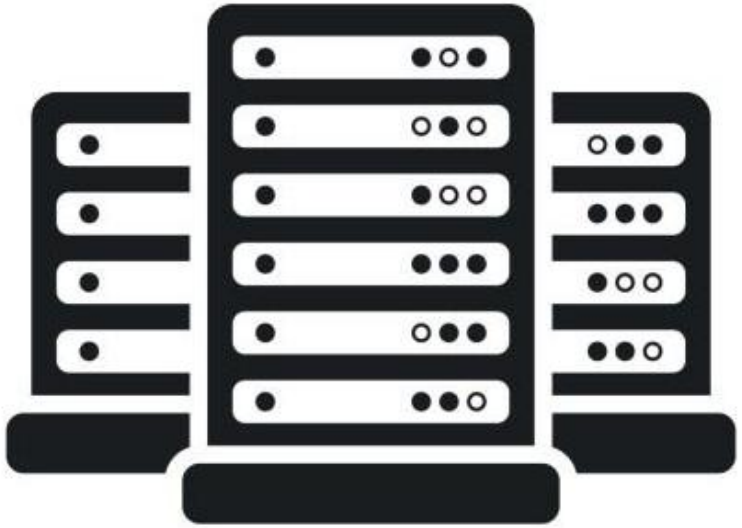
Computing cluster

https://github.com/mathtoriyama/aiida-pycce

Submit jobs



```
# Submit calculations (1 job per crystal structure)
for cif_code in structures.keys():


    # Gather inputs
    inputs = {
        "code_pycce": code_pycce,
        "calc_params_pycce": parameters_pycce,
        "custom_metadata": custom_metadata,
        "structure": structures[cif_code],
        "label": cif_code,
    }


    # Submit WorkChain
    chain = submit(Chain_PyCCE, **inputs)


    print(f"Submitted: {chain}")
```



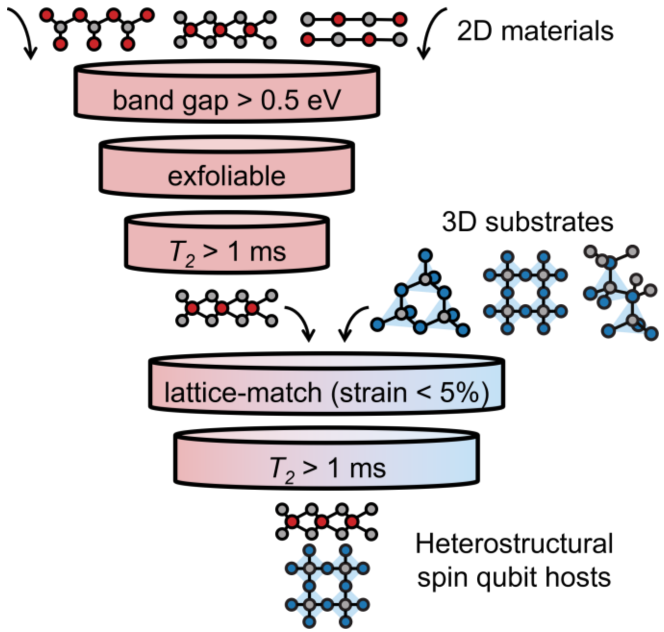
Computing cluster

https://github.com/mathtoriyama/aiida-pycce

Discovery of 2D qubit host materials

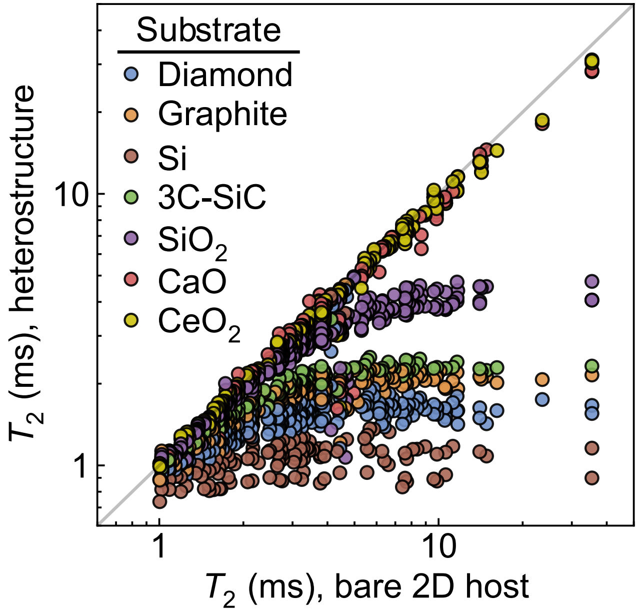
Software

Developed a computational strategy to predict T_2 in a high-throughput manner



Physics

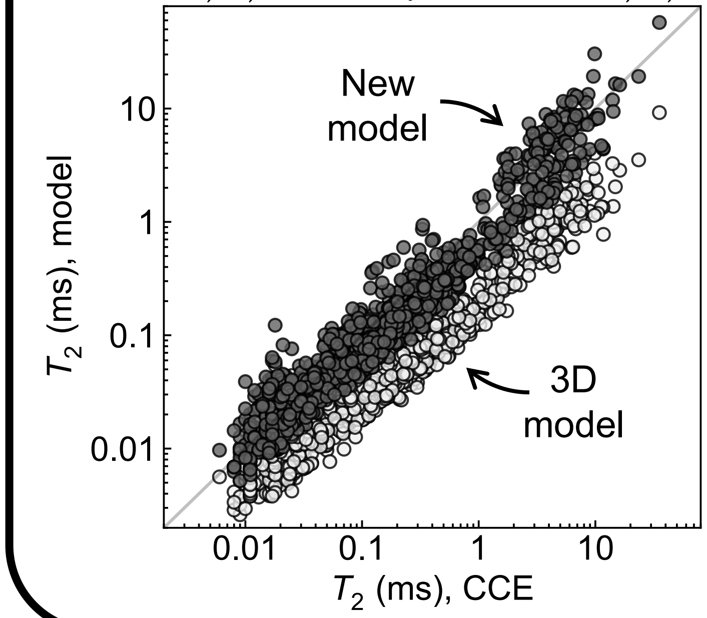
Identified design rules for 2D materials and substrates to sustain robust spin coherence



Data

Fitted an analytical formula to predict T_2 for 2D materials rapidly and accurately

$$T_{2,2D,i} \approx 0.94 n_i^{-0.28} w^{-0.95} \, T_{2,3D,i}$$



M. Toriyama, et al., *npj 2D Mater. Appl.*, **9**, 108 (2025)