

Excited States and Nonadiabatic Dynamics *CyberTraining School/Workshop 2023*

Alexey Akimov

University at Buffalo, SUNY

June 11, 2023

General Workshop Goals
and
Overview of the CyberTraining Infrastructure

Objectives and Agenda

CyberTraining: Pilot: Modeling Excited State Dynamics in Solar Energy Materials



Workshop Objectives

- Get **familiar with a variety of software packages** relevant to modeling of excited states and nonadiabatic dynamics
- Get an overview of **theoretical background** for corresponding computational methods
- Get a **practical experience** with these tools and packages

Keywords and topics:

- nonadiabatic dynamics
- excited states
- quantum dynamics
- quantum-classical methods
- charge transfer
- excitation energy transfer
- trajectory surface hopping
- coupled trajectories
- exact factorization
- TD-DFT, CASSCF, GW/BSE
- algorithms and methods
- software, programming, Python
- best practices, Git, GitHub

This year

- Libra (Akimov)
- Quantics/MCTDH (Worth)
- GAMESS (Filatov)
- NWChem (Govind)
- TT-SOFT, TT-Chebyshev (Soley)

2022 (in person/virtual)

- pyUNIxMD (Min)
- CT-MQC (Ibele)
- SHARC (Mai)
- SHARC/COBRAMM (Avagliano)
- OpenMolcas (Mai, Avagliano)
- ORCA (Mai)
- Hefei-NAMD (Zhao, Chu)
- Quantum Espresso (Zhao, Chu)
- BerkeleyGW and paratec (Zhang)
- DynEMol (Rego)
- Libra (Akimov)
- DFTB+ (Shakiba)
- CP2K (Shakiba)
- TBD (Kilin)

2021 (virtual)

- Libra (Akimov)
- NEXMD (Tretiak)
- Newton-X (Barbatti)
- nano-qmflows (Infante, Zapata)
- CAT, auto-FOX (Infante, Zapata)
- COLUMBUS (Lischka)
- DFTB+
- CP2K
- Quantum Espresso
- ErgoSCF

The Plan & Resource

All the details are here:

[https://compchem-cybertraining.github.io/Cyber Training Workshop 2023](https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2023)

Join Slack:

- Members can invite new members
- Private and public channels, direct (private) messages, conversations
- Any time, no strings attached

https://join.slack.com/t/quantumdynamicshub/shared_invite/zt-mjbhjssx-GGhsbYHxeBMvhmumK_j7LA

VPN and Accounts:

- 2-factor authentication
- submit a ticket: <https://ubccr.freshdesk.com/support/home>



University at Buffalo, SUNY
June 11-23, 2023
9:00 am - 5:00 pm EDT

Instructors: Alexey Akimov, Graham Worth, Michael Filatov, Niri Govind, Daniel Mejia Rodríguez, Micheline Soley

Helpers: Qingxin Zhang, Mohammad Shakiba, Eryn Spinlove, Konstantin Komarov, Edoardo Apra

Excited States and Nonadiabatic Dynamics CyberTraining Workshop 2023

About the Summer School and Workshop

The CyberTraining workshop aims to educate graduate students, postdocs, researchers, and educators working in a broader field of nonadiabatic and excited-state dynamics as well as in computational material sciences in a variety of tools and methods for such types of calculations. The workshop will provide conceptual and practical hands-on training in a range of methods and cyberinfrastructure (software and platforms) for modeling excited state and nonadiabatic dynamics in abstract models and atomistic materials. We will also cover tools and

Topics and Instructors

June 12, 2023 (Day 2), **Monday**

Morning, 9 am - noon

- Workshop Kick Off: goals, logistics, details. Overview of the CCR CyberInfrastructure (30 min)
- Working with Git and GitHub. Theory and Hands on (60 min)
- General overview of Libra software (Lecture and Demo/Hands on)(90 min)

Noon - 1:30 pm Lunch break **Afternoon, 1:30 pm - 5:00 pm**

- Theory of adiabatic and nonadiabatic dynamics. Lecture (60 min)
- TSH and Ehrenfest dynamics with model Hamiltonians in Libra. Hands on (90 min)
- HEOM and QTAG in Libra. (60 min)

Alexey Akimov, Qingxin
Zhang, Mohammad
Shakiba

June 13, 2023 (Day 3), **Tuesday**

Morning, 9 am - noon

- Machine learning capabilities of Libra: Lecture, Demo, and Hands on (60 min)
- DVR in Libra. Lecture, Demo, and Hands on (90 min)
- TBD (30 min)

Noon - 1:30 pm Lunch break **Afternoon, 1:30 pm - 5:00 pm**

- NA-MD in finite and condensed matter systems with xTB and TD-DFT with Libra/CP2k code.
Lecture and Hands on

Alexey Akimov, Qingxin
Zhang, Mohammad
Shakiba

Libra



<https://github.com/Quantum-Dynamics-Hub/libra-code>

https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2023/_episodes/03-libra

https://github.com/compchem-cybertraining/Tutorials_Libra

https://github.com/compchem-cybertraining/Tutorials_DFTB_plus

https://github.com/compchem-cybertraining/Tutorials_CP2K



Alexey Akimov



Mohammad Shakiba



Qingxin Zhang

Topics and Instructors

June 14, 2023 (Day 4), Wednesday

Morning, 9 am - noon

- Theory and hands on with Quantics and MCTDH

Noon - 1:30 pm Lunch break **Afternoon, 1:30 pm - 5:00 pm**

- Theory and hands on with Quantics and MCTDH

Graham Worth, Eryn
Spinlove

June 15, 2023 (Day 5), Thursday

Morning, 9 am - noon

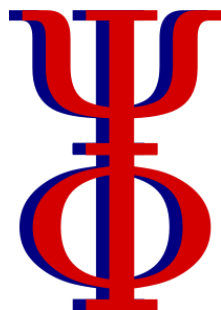
- Theory and hands on with Quantics and MCTDH

Noon - 1:30 pm Lunch break **Afternoon, 1:30 pm - 5:00 pm**

- Theory and hands on with Quantics and MCTDH

Graham Worth, Eryn
Spinlove

Quantics



<http://www2.chem.ucl.ac.uk/quantics/>

https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2023/_episodes/04-quantics

https://github.com/compchem-cybertraining/Tutorials_Quantics



Graham Worth



Eryn Spinlove

Topics and Instructors

June 16, 2023 (Day 6), Friday	Morning, 9 am - noon <ul style="list-style-type: none">Theory: Introduction in ensemble DFT and basic aspects of REKS method for ground electronic states Noon - 1:30 pm Lunch break Afternoon, 1:30 pm - 5:00 pm <ul style="list-style-type: none">Hands on: REKS implementation in GAMESS-US; Demos and practical exercises with REKS method for strongly correlated molecular ground states.	Michael Filatov, Konstantin Komarov
June 17, 2023 (Day 7), Saturday	On your own. Projects time	
June 18, 2023 (Day 8), Sunday	On your own. Projects time	
June 19, 2023 (Day 9), Monday	Morning, 9 am - noon <ul style="list-style-type: none">Theory: Ensemble DFT for excited states and its implementation in state-averaged REKS methodology Noon - 1:30 pm Lunch break Afternoon, 1:30 pm - 5:00 pm <ul style="list-style-type: none">Hands on: Practical exercises with GAMESS-US and NAMD simulations with GAMESS/pyUNI-xMD package	Michael Filatov, Konstantin Komarov

GAMESS US

<https://www.msg.chem.iastate.edu/gamess/>

https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2023/_episodes/05-gamess

https://github.com/compchem-cybertraining/Tutorials_GAMESS



Michael Filatov (Gulak)



Konstantin Komarov
(remotely)

Topics and Instructors

June 20, 2023 (Day 10), Tuesday

Morning, 9 am - noon

- [Theory and hands on with NWChem](#)

Noon - 1:30 pm Lunch break **Afternoon, 1:30 pm - 5:00 pm**

- [Theory and hands on with NWChem](#)

Daniel Mejia Rodriguez,
Edoardo Apra, Niri Govind

June 21, 2023 (Day 11), Wednesday

Morning, 9 am - noon

- [Theory and hands on with NWChem](#)

Noon - 1:30 pm Lunch break **Afternoon, 1:30 pm - 5:00 pm**

- [Theory and hands on with NWChem](#)

Daniel Mejia Rodriguez,
Edoardo Apra, Niri Govind

NWChem

<https://www.nwchem-sw.org/>



Daniel Mejia-Rodriguez



Niri
Govind
(remotely)



Edoardo
Apra
(remotely)

https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2023/_episodes/06-nwchem

https://github.com/compchem-cybertraining/Tutorials_NWChem

Topics and Instructors

June 22, 2023 (Day 12), Thursday

Morning, 9 am - noon

- Theory of quantum dynamics with TT-SOFT and TT-Chebyshev, IPA optimization.

Noon - 1:30 pm Lunch break **Afternoon, 1:30 pm - 5:00 pm**

- Hands on with TT-SOFT and TT-Chebyshev, IPA optimization.

Micheline Soley

TT-SOFT and TT-Chebyshev

https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2023/_episodes/07-ttsoft

https://github.com/compchem-cybertraining/Tutorials_NWChem



Micheline Soley

Please Introduce Yourself

- Name, position, affiliation, research group
- Research interests and expertise
- Anything else you would like to share with us

More Resources

Codes: <https://github.com/Quantum-Dynamics-Hub>

Training: <https://github.com/compchem-cybertraining>

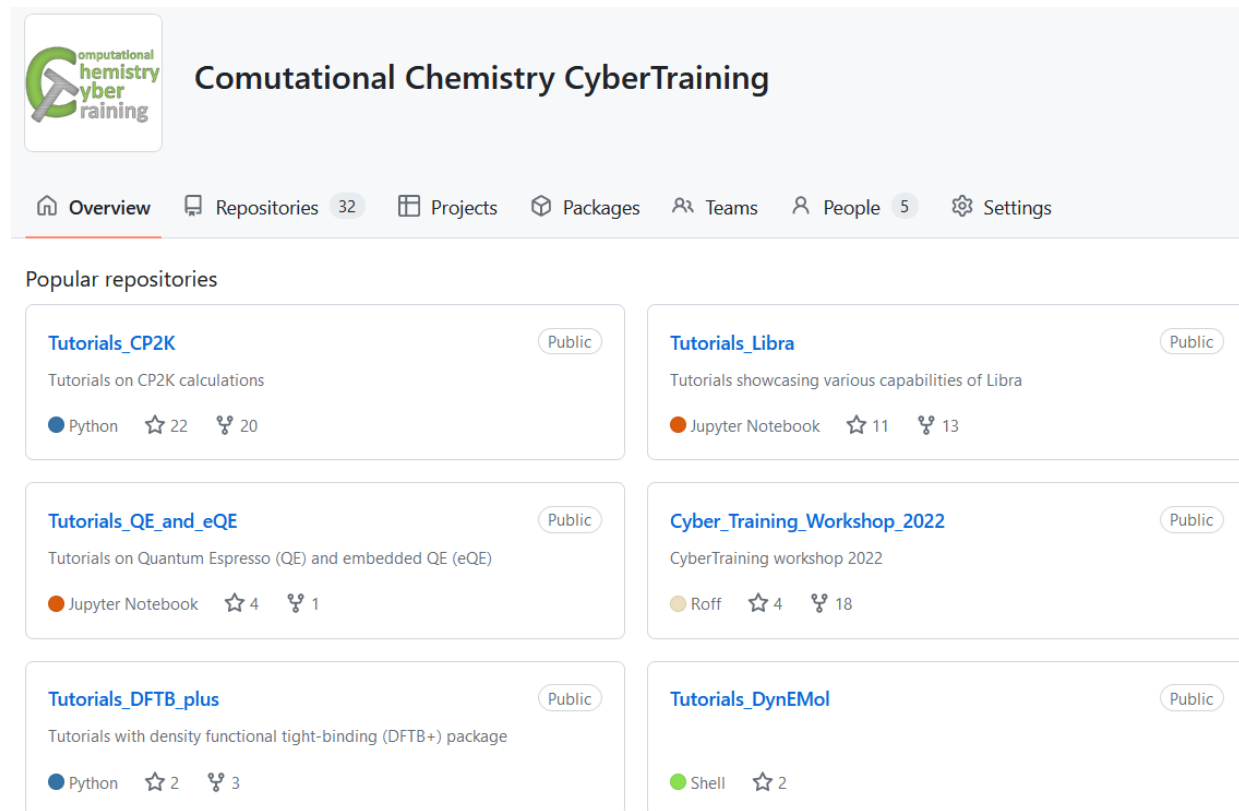
Quantum Dynamics Hub: <https://quantum-dynamics-hub.github.io/>

Summer 2021 workshop: [https://compchem-cybertraining.github.io/Cyber Training Workshop 2021/](https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2021/)

Libra Winter school: [https://compchem-cybertraining.github.io/Libra Winter School 2022/](https://compchem-cybertraining.github.io/Libra_Winter_School_2022/)

Summer 2022 workshop: [https://compchem-cybertraining.github.io/Cyber Training Workshop 2022/](https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2022/)

Summer 2023 (This!) workshop: [https://compchem-cybertraining.github.io/Cyber Training Workshop 2023/](https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2023/)



The screenshot shows the GitHub profile for 'Computational Chemistry CyberTraining'. The profile includes a repository count of 32, and tabs for Overview, Repositories, Projects, Packages, Teams, People (5), and Settings. Under the 'Popular repositories' section, six repositories are listed:

Repository Name	Description	Language	Stars	Forks	Public
Tutorials_CP2K	Tutorials on CP2K calculations	Python	22	20	Yes
Tutorials_Libra	Tutorials showcasing various capabilities of Libra	Jupyter Notebook	11	13	Yes
Tutorials_QE_and_eQE	Tutorials on Quantum Espresso (QE) and embedded QE (eQE)	Jupyter Notebook	4	1	Yes
Cyber_Training_Workshop_2022	CyberTraining workshop 2022	Roff	4	18	Yes
Tutorials_DFTB_plus	Tutorials with density functional tight-binding (DFTB+) package	Python	2	3	Yes
Tutorials_DynEMol		Shell	2		Yes

Daily Schedule

Daily

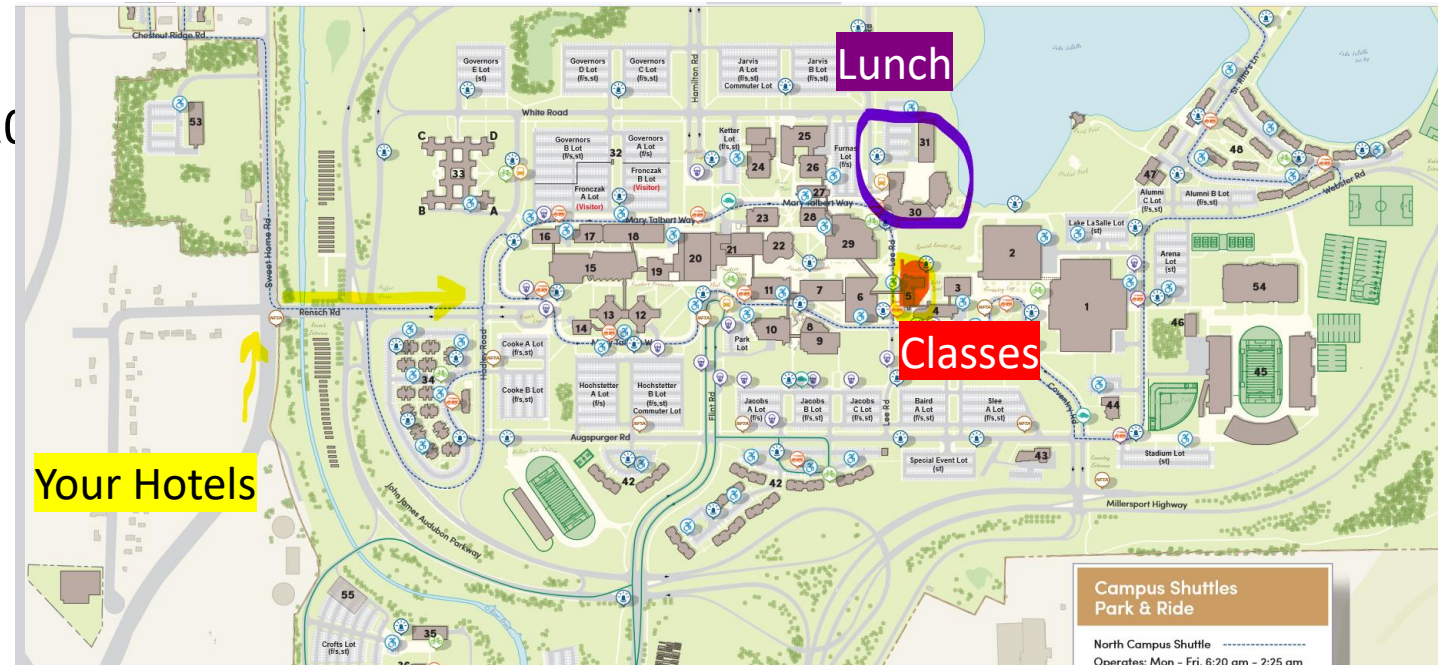
- Breakfast = hotel
- 9:00 am – 12:00 pm: Morning session (Recording)
- 12:00 – 1:30 pm Working lunch/rest – on your own at “Commons”, rest, discuss, collaborate
- 1:30 pm – 5:00 pm: Afternoon session (Recording)
- After 5:00 pm: collaborations/on your own, dinner on your own

Location

- Classes are @: June 12 – 22 Clemens 120

Campus Map:

<https://emergency.buffalo.edu/content/dam/www/parking/North%20Campus%20Parking%20Map%2011x17.pdf>
<https://emergency.buffalo.edu/content/dam/www/parking/North%20Campus%20Parking%20Map%2011x17.pdf>



- We cover your hotel stay (except for the local folks). Stipends also cover the rest of expenses, please keep your receipts just in case. Let me know your flight expenses via the Google form provided on Slack channel.
- Travel for the US participants to a reasonable amount, partially the international participants (as the funds allow), except for local/UB-affiliated folks.
- Paperwork: All trainees will need to fill in the **RF Participation Stipend** form and one of the other two forms: **W-9 - for the US residents** and **W-8BEN for the non-residents**. The forms are distributed to you **via Slack** – please **DON't sent them back via e-mail** – upload to the form provided or via Slack.
- A lot of paperwork later – likely it'll be just me handling most of the stuff
- Prizes: \$300 (1 first prize), \$200 (3 second prizes), \$100 (5 first prizes) – the project competition. Online and in-person participants are eligible. UB-affiliated participants are eligible too.
- Reimbursement/honoraria to the instructors – a separate paperwork. Will send you instructions via the Instructors Channel on Slack.

Course Project

Project rules

[https://compchem-cybertraining.github.io/Cyber Training Workshop 2023/CODE OF CONDUCT.html](https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2023/CODE_OF_CONDUCT.html)

- Consist of: a) short written report, b) presentation at the last day of workshop; c) set of input/output files deposited on the GitHub repository
 - Should actively involve one of the packages discussed over the workshop period
 - Preferably not something you have an extensive experience with
 - Doesn't have to be a full-scale research project, but can be a step towards this direction
 - Projects completed using local or home institution resources are eligible
 - Can be an application or a coding project
 - The consistency in your course work during this school will contribute to your chances to win the awards
 - The awards decisions will be made based on the committee evaluation. The awards will be: 1 first prize (\$300), 3 second
-
- Submit your project via GitHub by June 30
 - Oral presentation – tentatively July 1-2, via Zoom

Check out the past years' projects:

[https://github.com/compchem-cybertraining/Cyber Training Workshop 2021/tree/gh-pages/course work](https://github.com/compchem-cybertraining/Cyber_Training_Workshop_2021/tree/gh-pages/course_work)

[https://compchem-cybertraining.github.io/Cyber Training Workshop 2021/ episodes/13-projects](https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2021/episodes/13-projects)

[https://github.com/compchem-cybertraining/Cyber Training Workshop 2022/tree/gh-pages/course work](https://github.com/compchem-cybertraining/Cyber_Training_Workshop_2022/tree/gh-pages/course_work)

[https://compchem-cybertraining.github.io/Cyber Training Workshop 2022/ episodes/15-projects](https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2022/episodes/15-projects)

Getting Started on UB CCR

Accessing UB Computing Resources

Before the Workshop

https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2023/setup.html

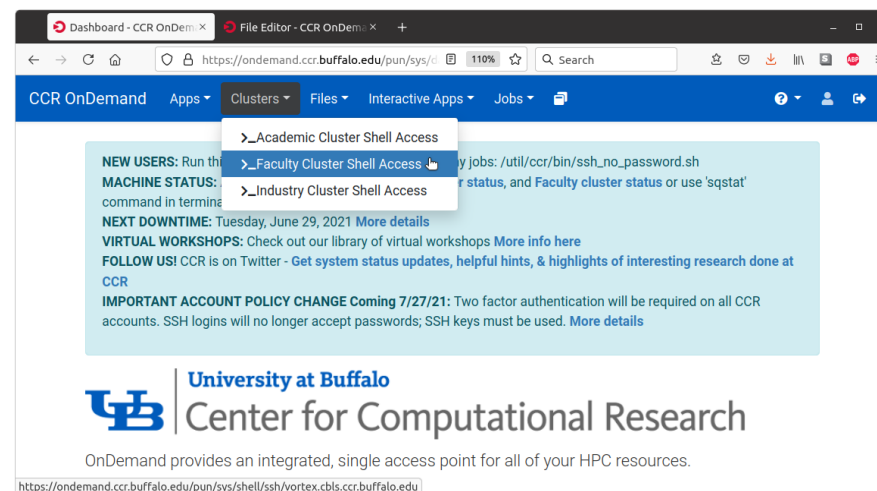
https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2023/_episodes/01-introduction

OnDemand (not Firefox, use Chrome)

On campus – nothing special;

Off-campus – use UB VPN

<https://ondemand.ccr.buffalo.edu/>

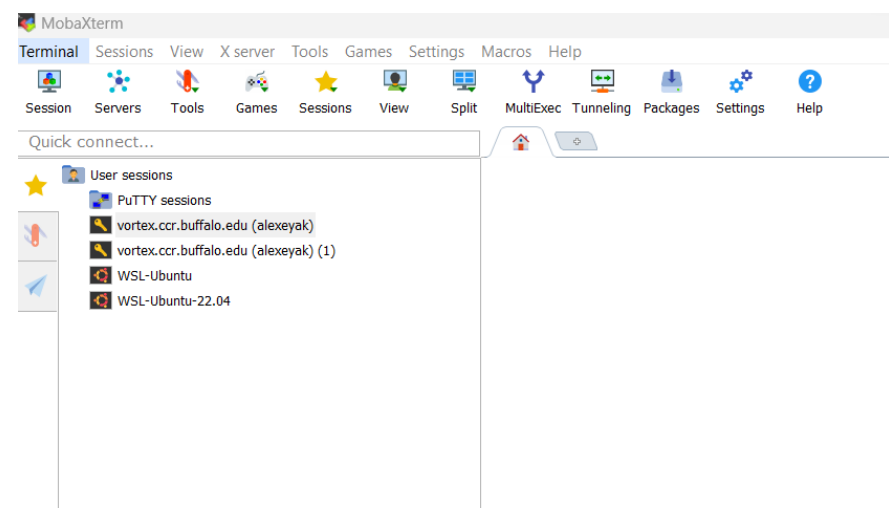


Moba Xterm

Generate the SSH public/private key.

- Use Moba Tools -> MobaKeyGen
- Coordinate it with your UB credentials

<https://docs.ccr.buffalo.edu/en/latest/portals/idm/>



Accessing UB Computing Resources

Your **.bashrc** file (in your home directory)

```
# .bashrc
```

```
# User specific aliases and functions
```

```
# Source global definitions
```

```
if [ -f /etc/bashrc ]; then
```

```
    . /etc/bashrc
```

```
fi
```

```
module use /projects/academic/cyberwksp21/Modules
```

```
eval "$(/projects/academic/cyberwksp21/SOFTWARE/Conda/bin/conda shell.bash hook)"
```

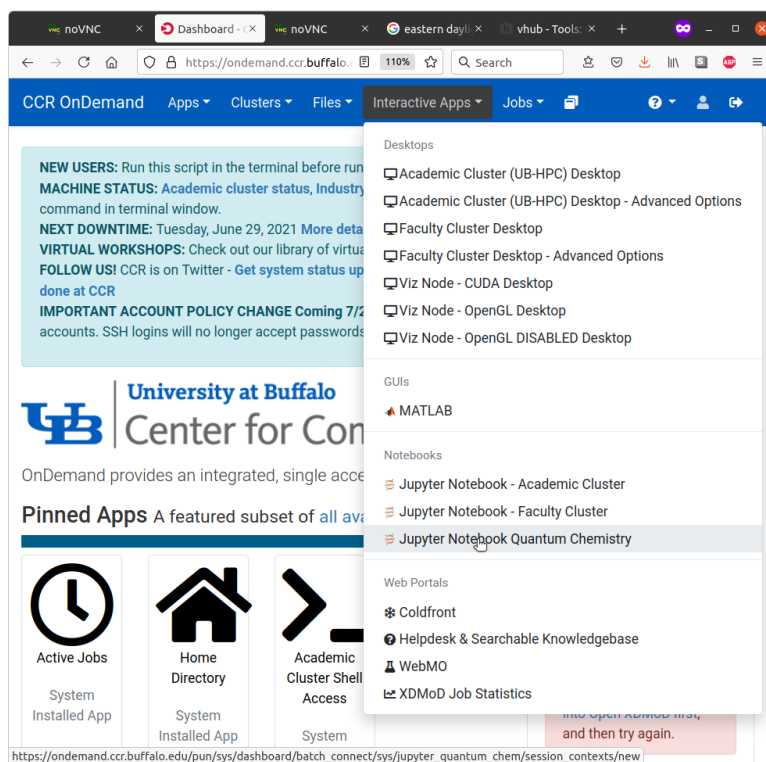
- Restart terminal or `source .bashrc`
- For terminal-based operations: Activate conda environment: `conda activate libra2`
- For Jupyter – just launch it

Projects directory: `/projects/academic/cyberwksp21` - slower, smaller, but permanent

Scratch space: `/panasas/scratch/grp-cyberwksp21` - faster, larger, but temporarily (unused files are cleaned up every 1 or 2 weeks)

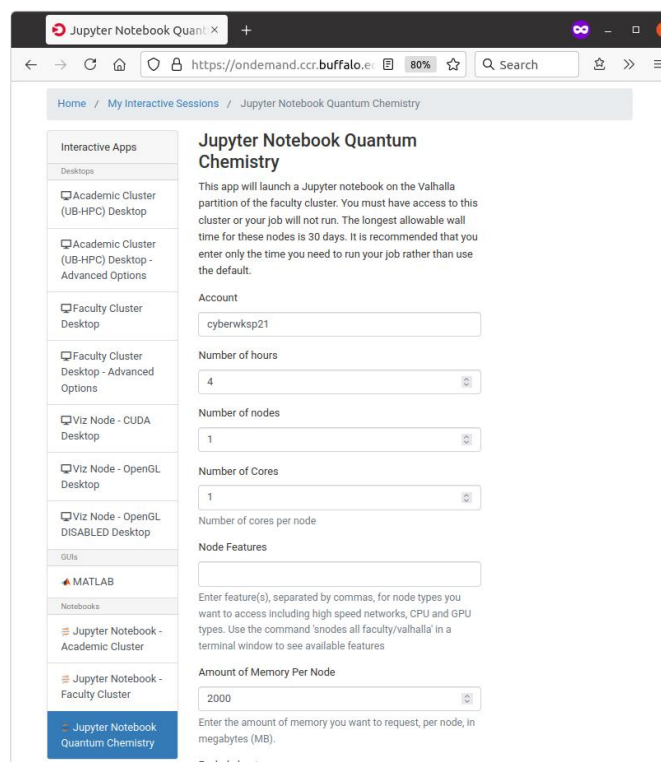
Using Jupyter Notebooks

[https://compchem-cybertraining.github.io/Cyber Training Workshop 2023/ episodes/01-introduction](https://compchem-cybertraining.github.io/Cyber_Training_Workshop_2023/episodes/01-introduction)



The screenshot shows the CCR OnDemand dashboard. The 'Interactive Apps' menu is open, displaying various desktop and notebook options. The 'Pinned Apps' section at the bottom highlights 'Jupyter Notebook - Quantum Chemistry'.

- Interactive Apps Menu:**
 - Desktops:
 - Academic Cluster (UB-HPC) Desktop
 - Academic Cluster (UB-HPC) Desktop - Advanced Options
 - Faculty Cluster Desktop
 - Faculty Cluster Desktop - Advanced Options
 - Viz Node - CUDA Desktop
 - Viz Node - OpenGL Desktop
 - Viz Node - OpenGL DISABLED Desktop
 - GUIs:
 - MATLAB
 - Notebooks:
 - Jupyter Notebook - Academic Cluster
 - Jupyter Notebook - Faculty Cluster
 - Jupyter Notebook Quantum Chemistry
 - Web Portals:
 - Coldfront
 - Helpdesk & Searchable Knowledgebase
 - WebMO
 - XDMoD Job Statistics
- Pinned Apps:**
 - Active Jobs
 - Home Directory
 - Academic Cluster Shell Access



This screenshot shows the configuration page for the 'Jupyter Notebook Quantum Chemistry' app. It includes fields for account, number of hours, number of nodes, number of cores, and node features.

Interactive Apps

Jupyter Notebook Quantum Chemistry

This app will launch a Jupyter notebook on the Valhalla partition of the faculty cluster. You must have access to this cluster or your job will not run. The longest allowable wall time for these nodes is 30 days. It is recommended that you enter only the time you need to run your job rather than use the default.

Account: cyberwks21

Number of hours: 4

Number of nodes: 1

Number of Cores: 1

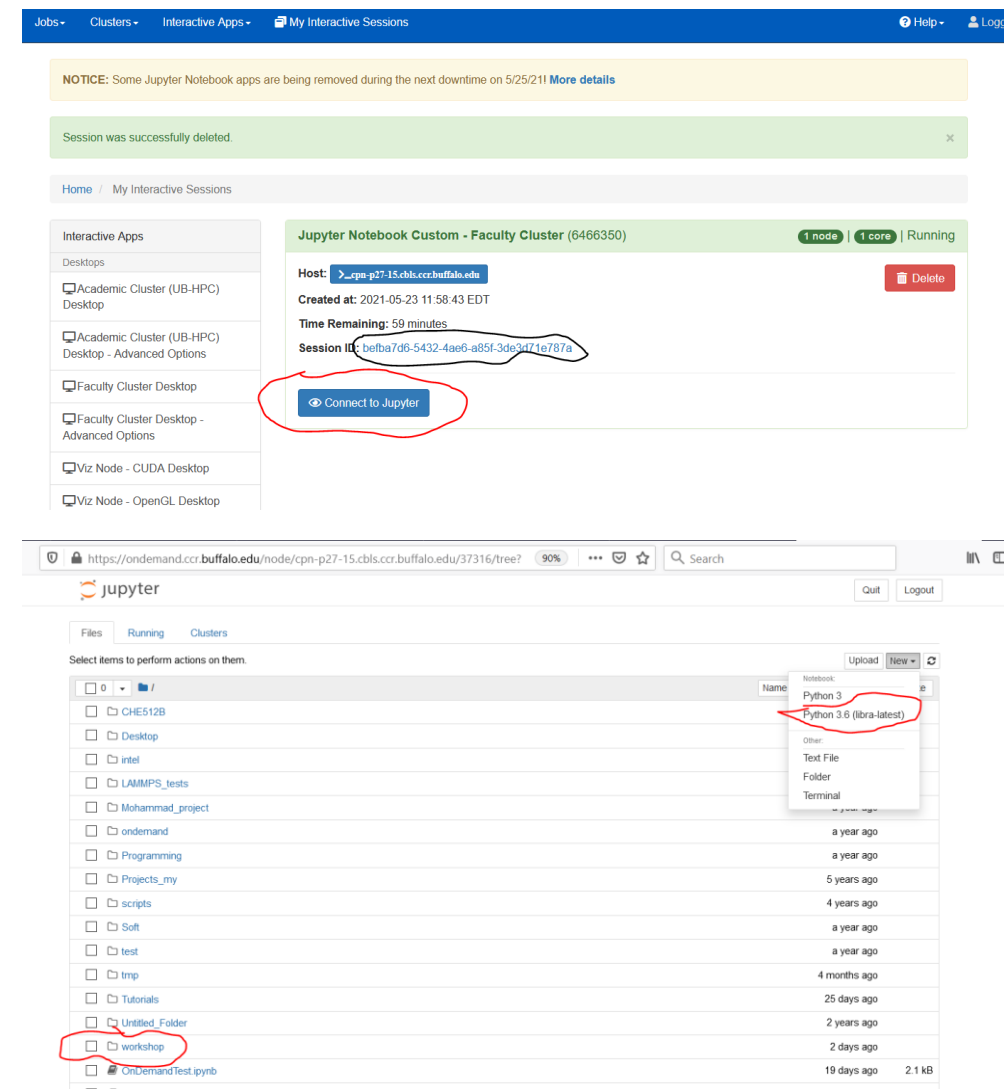
Number of cores per node: 1

Node Features:

Enter feature(s), separated by commas, for node types you want to access including high speed networks, CPU and GPU types. Use the command 'nodes all faculty/valhalla' in a terminal window to see available features

Amount of Memory Per Node: 2000

Enter the amount of memory you want to request, per node, in megabytes (MB).



This screenshot shows the Jupyter Notebook interface. The 'workshop' directory is selected in the file explorer, and the 'Python 3 (libra-latest)' kernel is chosen. A red circle highlights the 'workshop' directory in the file explorer.

Jobs **Clusters** **Interactive Apps** **My Interactive Sessions**

NOTICE: Some Jupyter Notebook apps are being removed during the next downtime on 5/25/21! [More details](#)

Session was successfully deleted.

Home / **My Interactive Sessions**

Interactive Apps

Jupyter Notebook Custom - Faculty Cluster (6466350) **1 node** **1 core** **Running**

Host: >cpn-p27-15.cbls.ccr.buffalo.edu

Created at: 2021-05-23 11:58:43 EDT

Time Remaining: 59 minutes

Session ID: be1ba7d6-5432-4ae6-a85f-3de3471a787a

[Connect to Jupyter](#)

Files **Running** **Clusters**

Select items to perform actions on them.

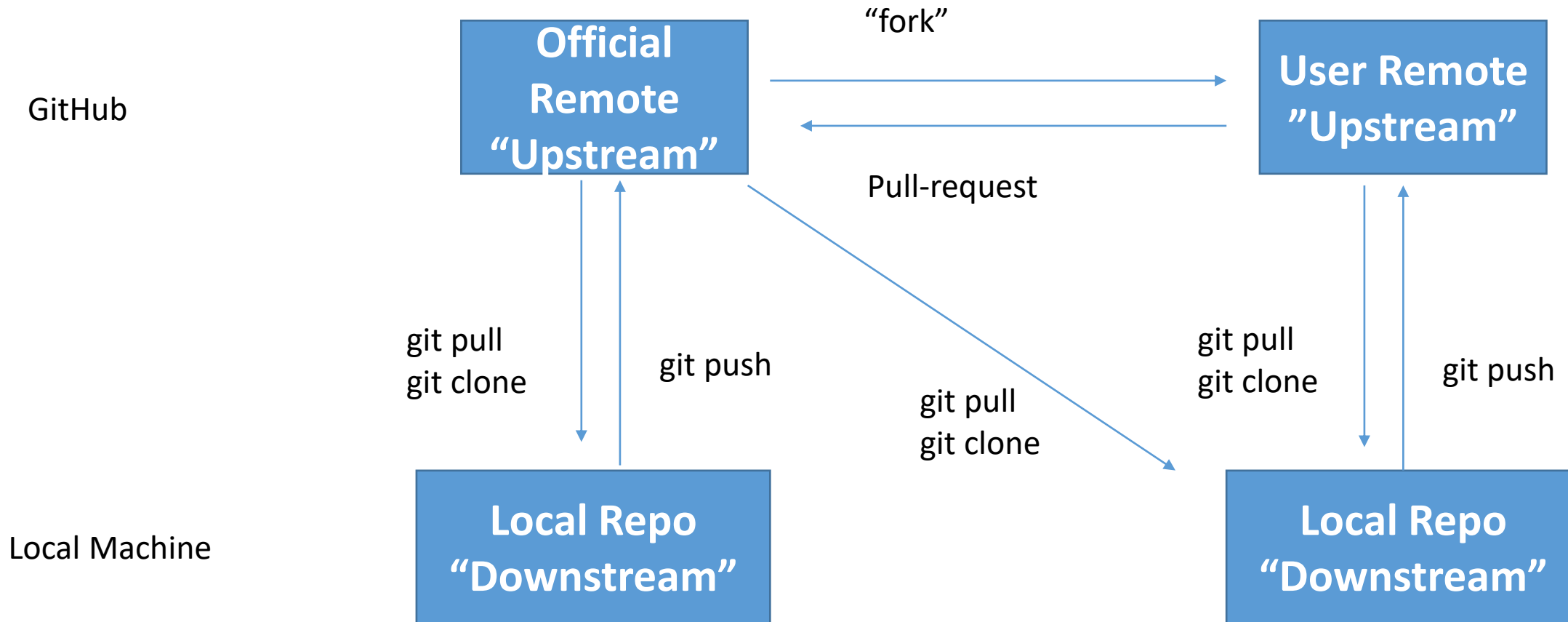
workshop

Python 3 (libra-latest)

GitHub & Git

Overview

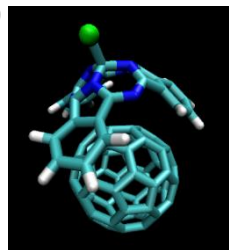
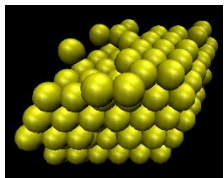
Git and GitHub Workflow



Libra overview

Libra History

Classical MD



Akimov, Prezhdo, *JCTC*, **2013**, 9, 4959.
Akimov, Prezhdo, *JCTC*, **2014**, 10, 789



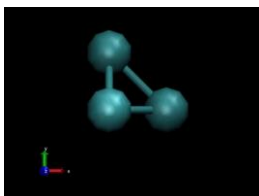
Akimov *JCC*, **2016**, 37, 1626

Libra-X (with Drs. Ryoji Asahi, Kosuke Sato, Ekadashi Pradhan)

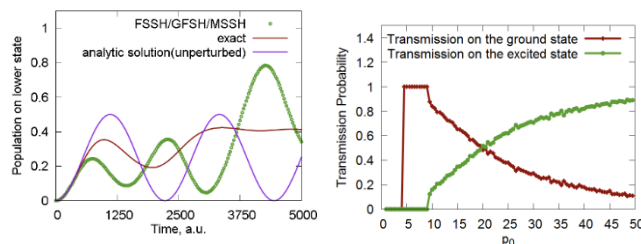
Sato, Pradhan, Asahi, Akimov *PCCP* **2018**, 20, 25275

Pradhan, Sato, Akimov *J. Phys.: Condens. Matter*, **2018**, 30, 484002

Rigid body MD

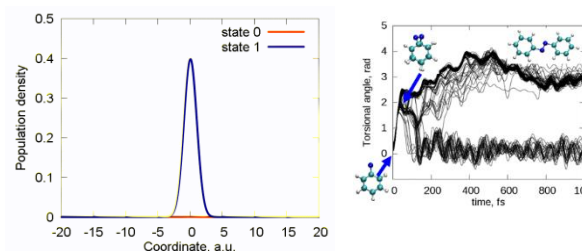


Ehrenfest & TSH



DVR

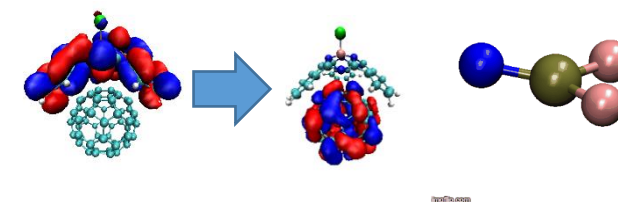
Back-reaction



- Simplectic integrators for classical DOFs
- Thermostats
- Barostats
- Force fields
- ANN
- Chemical object representation
- Simplectic integrators for TDSE
- TSH, Ehrenfest, stand-alone scripts
- Decoherence methods
- Model problems
- Added HF and EHT to LCCCS
- Interface with VASP, then QE

- Modularization and revision
- DVR methods
- Semiempirical Hamiltonians
- Molecular integrals
- Decoherence methods, TSH

- Interfaces with GAMESS, QE
- Added back-reaction for QE
- More modularization



Pyxaid2 (with Prof. Wei Li)

Li, Zhou, Prezhdo, Akimov *ACS Energy Lett*, **2018**, 3, 2159

- SOC, multiple k-points, etc.

2007-2011
(LCCCS)

2011-2015
(Pyxaid)

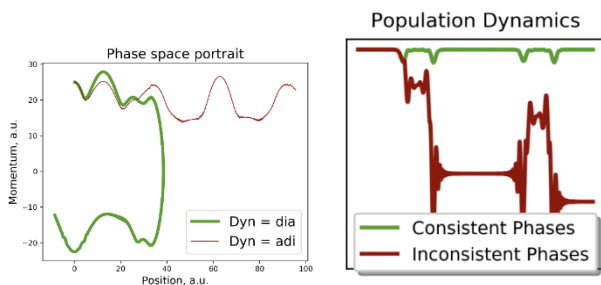
2015/2016
(Libra)

2018
(Pyxaid2, Libra-X)

Libra History

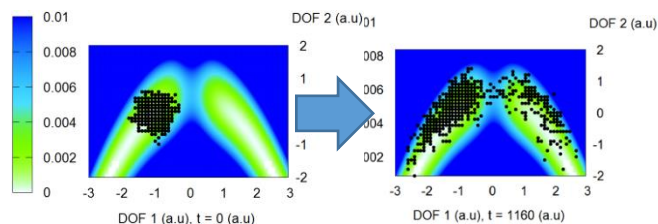
Phase correction for NACs

Akimov *JPCL* **2018** 9, 6096-6102



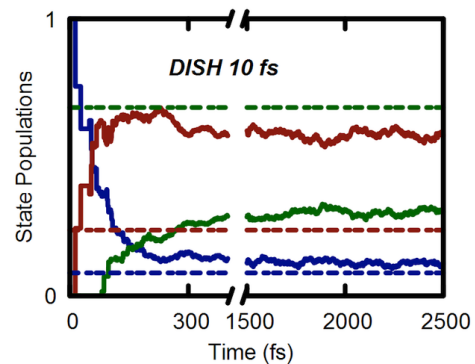
Entangled trajectories

Smith, Akimov *JCP* **2018**, 148, 144106



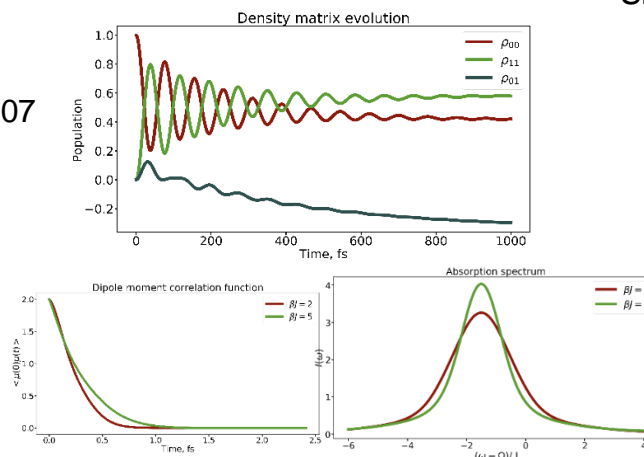
Bastida's Boltzmann-corrected Ehrenfest, mSDM

Smith; Akimov *JCP* **2019**, 151, 124107



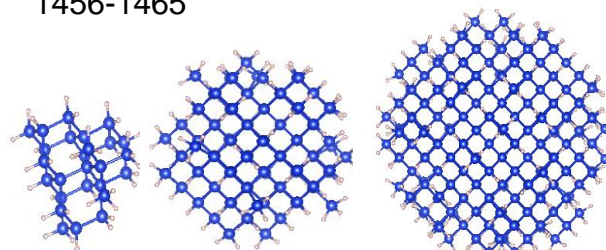
HEOM

Temen, Jain, Akimov *Int. J. Quant. Chem.*, **2020**, 120, e26373



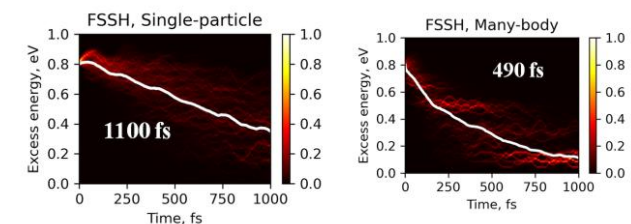
Belyaev-Lebedev LZ method

Smith, B.; Akimov, A. V *JPCL* **2020**, 11, 1456-1465



Many-body NA-MD

Smith, B.; Shakiba, M.; AVA *JCTC* **2021**, 17, 678
Smith, B.; Shakiba, M.; AVA *JPCL* **2021**, 12, 2444

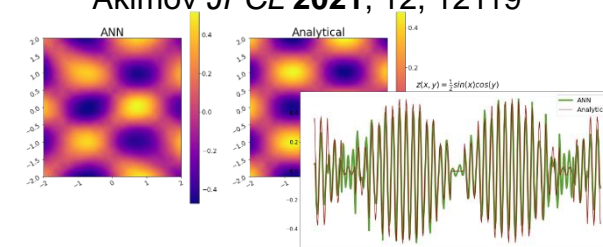


Revised DISH, new workflows

Akimov *JCP* **2021**, 155, 134106

Machine Learning revised. TD-ML approach

Akimov *JPCL* **2021**, 12, 12119



2018

2019

2020

2021

- **modular**
- **versatile**
- **“methodology discovery”**
(prototyping)
- **practical**
- **user-friendly & documented**
- **community tool**

Maximize and simplify the re-use, OOP

linear algebra, integrals,
quantum and classical mechanics/dynamics,
nonadiabatic methods, surface hopping,
IO utilities, model preparation and analysis

- Use with model problems and atomistic simulations
- Python – for convenience, C++ - for efficiency

Fully-functional tool that can be applied to real
(atomistic) systems to study materials

The code is convenient to users and they have plenty
resources – examples and documentation

- A platform to adopt the past and latest developments
- The developers can understand and contribute to the code

- **Many codes** (Newton-X, SHARC, NEXMD, FIERBALL, JADE, MOLPRO, PYXAID, PYUNIXMD, ...)
 - Black-box. Difficult to re-use to formulate other methods, etc.
 - Limited functionality (high focus, e.g. atomistic of special kind)
- **Many methods** (FSSH, DISH, A-FSSH, QTAG, QTSH, etc.)
 - Not always available
 - Not always user-friendly (e.g. my experience with PYXAID prototype)
 - Not always portable/modular, lack of best coding standards, no version control, etc.
 - Limited consistency of different codes
 - Possible redundancies even in the same code

- Adopt the best practices
 - Modularity (e.g. PySCF, Psi4NumPy, PyQuante, HORTON)
 - Language standards (Python, C++ vs. Fortran? Hybrid programming)
 - Testing & Documentations (pytest, unittest, Doxygen/Sphinx)
 - User/developer training (Workshops, Summer/Winter schools)
- Focus on the community
 - Every group has expertise in their field – rely on that
 - Community contributions – PR on GitHub
 - Use version control and collaborative workflows via GitHub, Issues
 - Frequent communication and close collaboration e.g. via Slack

Community Tool: Code Contributions/Integration

Amber Jain – Hierarchical Equations of Motion (HEOM)
<https://github.com/amber-jain-group-iitb/heom> **amber**

src/dyn/heom

Xiang Sun – (Non)-equilibrium Fermi Golden Rule (FGR)
<https://github.com/tsiangsun/FGR>

src/fgr

Nandini Ananth – Initial value representation (IVR)
<https://github.com/AnanthGroup/SC-IVR-Code-Package>

src/ivr

Sophya Garaschchuk – quantum trajectory guided Gaussians (QTAG)

src/libra_py/dynamics/qtag

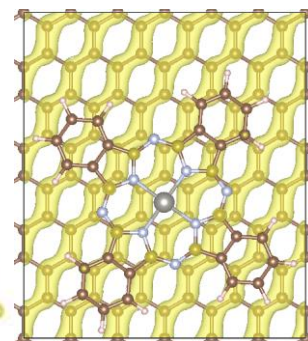
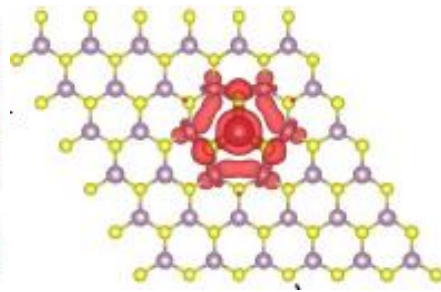
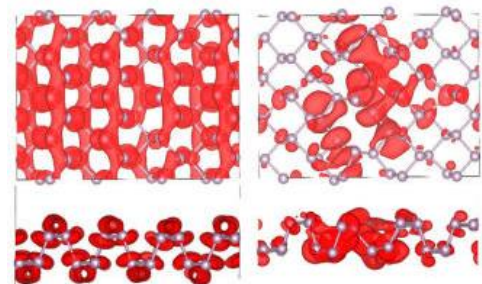
Craig Martens – quantum trajectory surface hopping (QTSH) in progress

... and more

Practical: Libra in Materials Research

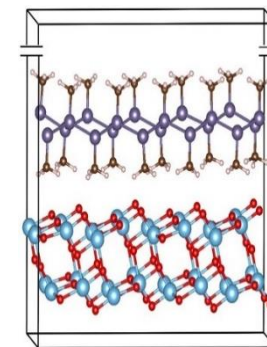
2D systems

Long et al. *JPCL* **2016**, 7, 653.



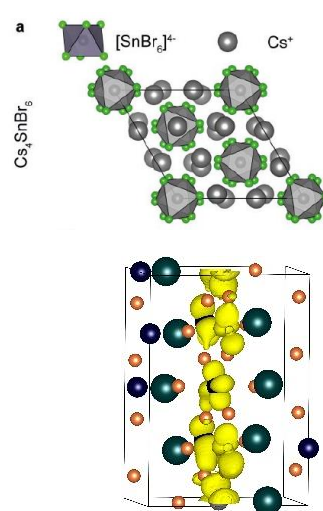
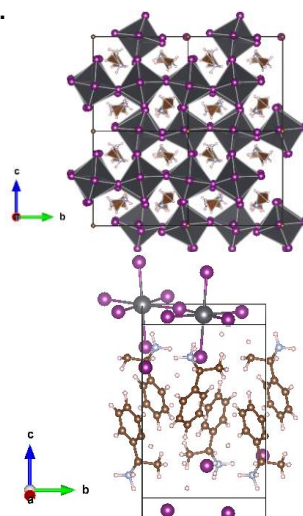
2D heterojunctions

Nijamudheen, A.; *AVA JPCC*, **2017**, 121, 6520



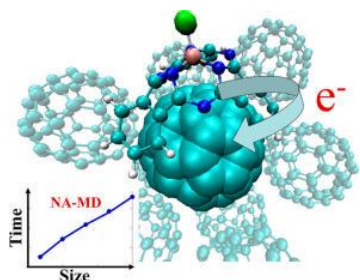
Perovskites

Nijamudheen, A.; *AVA JPCL* **2018**, 9, 248



Organic heterojunctions

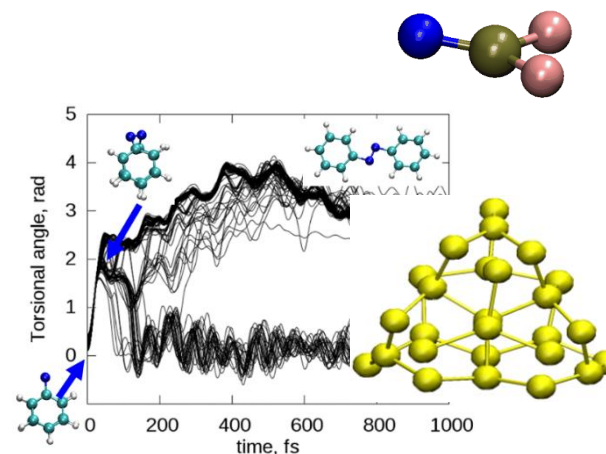
Sato et al. *PCCP*, **2018**, 20, 25275.



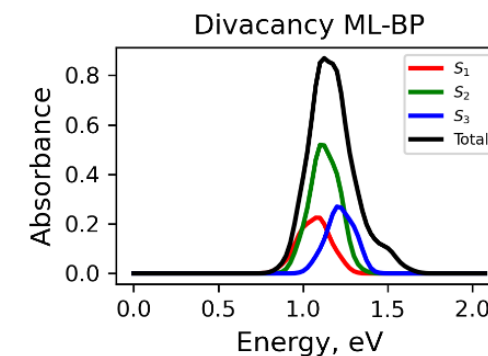
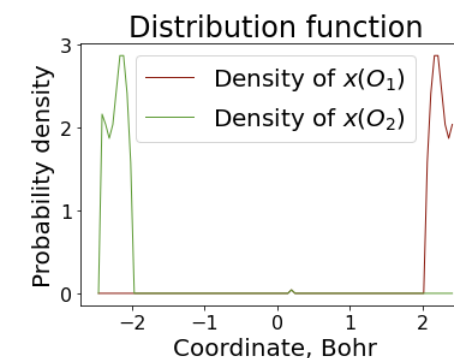
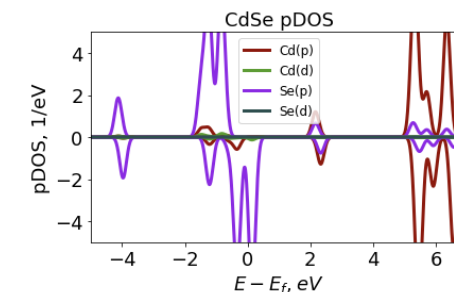
Quantum Dots & Molecules

Lin, Y.; *AVA JPCC*, **2016**, 120, 9028

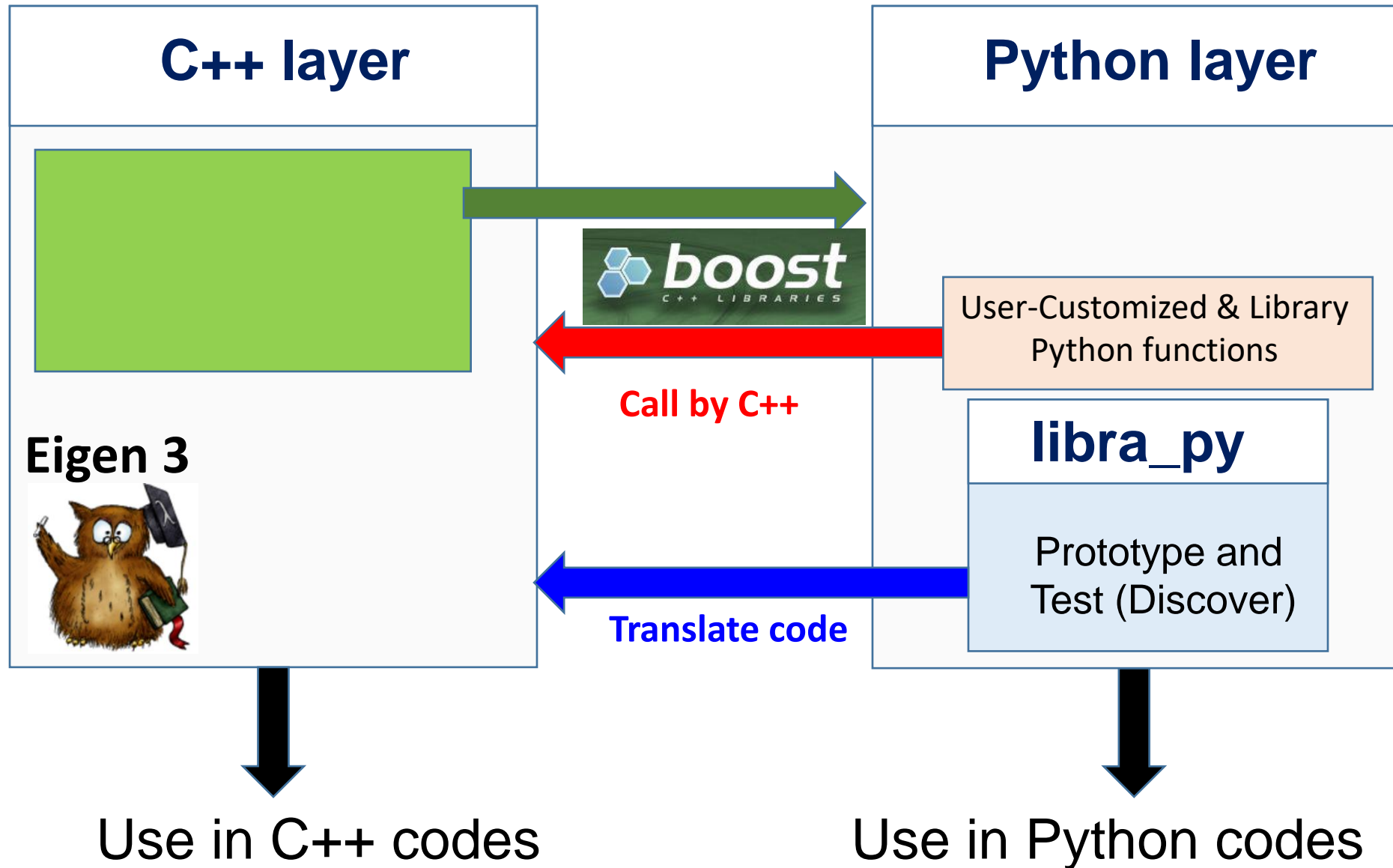
Pradhan et al. *JPCM*, **2018**, 30, 484002



Auxiliary Analysis Tools



C++/Python Interoperability



Modularity: API Diversity

- The goal is to suite the needs of the **users of various levels**
- Find a balance between **simplicity** and **flexibility**

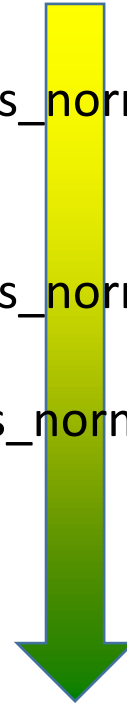
Developer/Efficiency

double **gaussian_overlap**(AO* AOa, AO* AOb,int is_normalize, int is_derivs, VECTOR& dIdA, VECTOR& dIdB, vector<double*>& auxd,int n_aux);

double **gaussian_overlap**(AO* AOa, AO* AOb,int is_normalize, int is_derivs, VECTOR& dIdA, VECTOR& dIdB);

double **gaussian_overlap**(AO* AOa, AO* AOb,int is_normalize);

double **gaussian_overlap**(AO* AOa, AO* AOb);



User/Convenience

Example

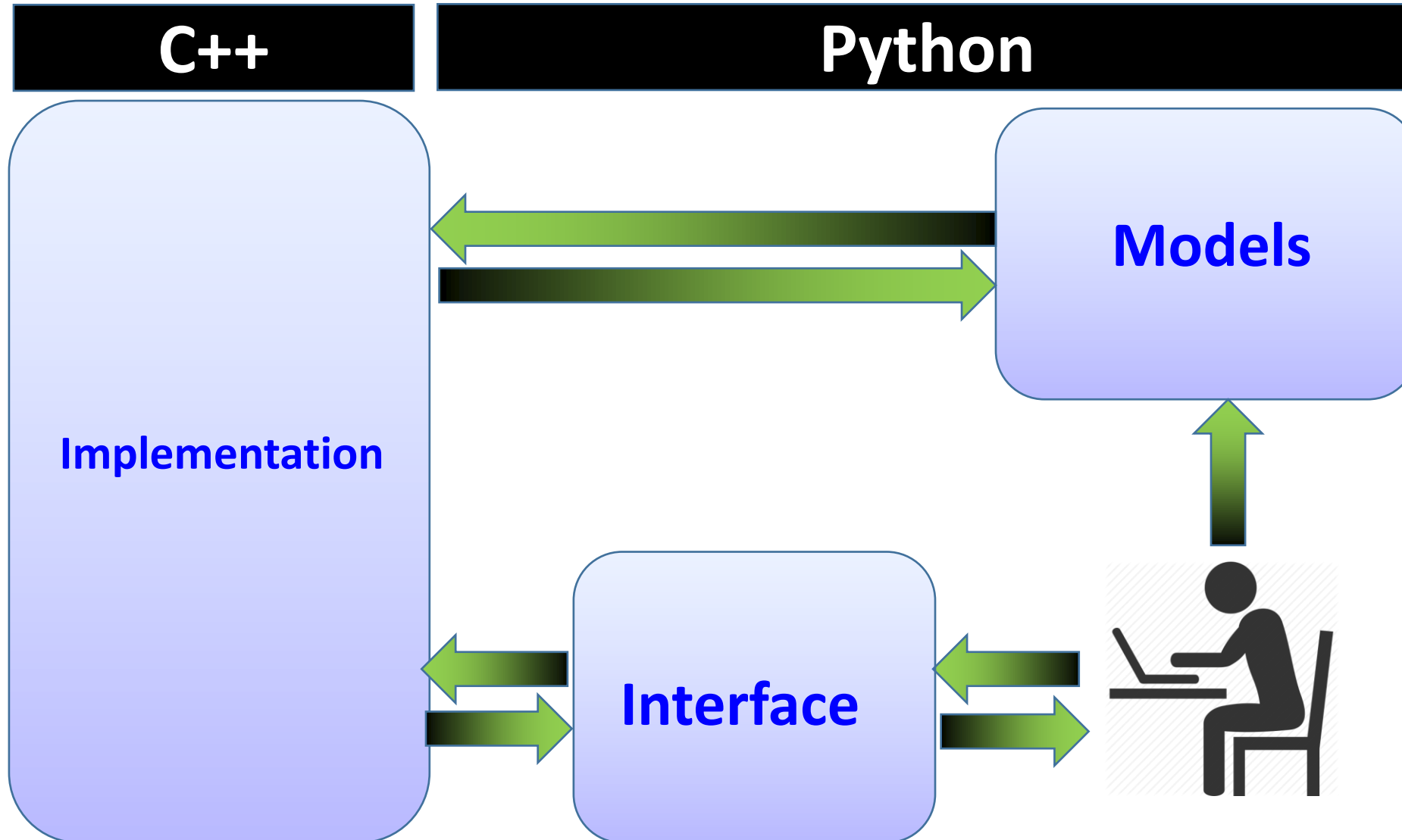
Computing kinetic energy between Gaussians

```
g1 = PrimitiveG()  
g2 = PrimitiveG()  
g1.init(n1,m1,k1, a1, VECTOR(x1, y1, z1))  
g2.init(n2,m2,k2, a2, VECTOR(x2, y2, z1))  
  
kin = kinetic_integral(g1,g2)
```

Benchmarked against PyQuante

```
p1 = PyQuante.PGBF.PGBF(a1,(R1.x,R1.y,R1.z),(n1,m1,k1))  
p2 = PyQuante.PGBF.PGBF(a2,(R2.x,R2.y,R2.z),(n2,m2,k2))  
  
val_ref = p1.kinetic(p2)
```

Passing Python functions



How it works with Sampling

```
vector<MATRIX> metropolis_gau  
(Random& rnd, bp::object target_distribution,  
MATRIX& dof, bp::object distribution_params,  
int sample_size, int start_sampling, double gau_var){
```

Metropolis Algorithm

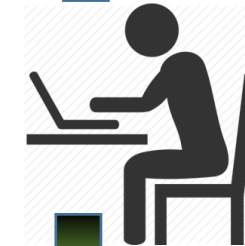
```
double p_old =  
bp::extract<double>( target_distribution(s_old,  
distribution_params) );  
  
...  
}
```

C++

```
def test():  
    q = MATRIX(ndof, 1)  
    output = metropolis_gau( piab, q, params, ...)
```

User calls the sampling

Output



```
def piab(q, params):
```

User defines the
probability density

Python

Example

User defines how to
run the MC sampling

```
q = MATRIX(1,1); q.set(0, 0.5)
params = {"k":1.0, "m":2000.0, "states":[0], "coeffs":[1.0]}
Nsamp = 1000000; Nstart = 50000
sampling = metropolis_gau(rnd, HO_sup, q, params, Nsamp, Nstart, 0.05)
bin(sampling, -1.5, 2.0, 0.01, 0, 0, "_distrib-1.txt")
```

```
def HO_sup(q, params):
    k = params["k"]; m = params["m"];
    states = params["states"]; coeffs = params["coeffs"]
    x = q.get(0)
    sz = len(states)
    p = 0.0
    for n in xrange(sz):
        p = p + coeffs[n] * ket_n(x, states[n], k, m)
    p = p * p
    return p
```

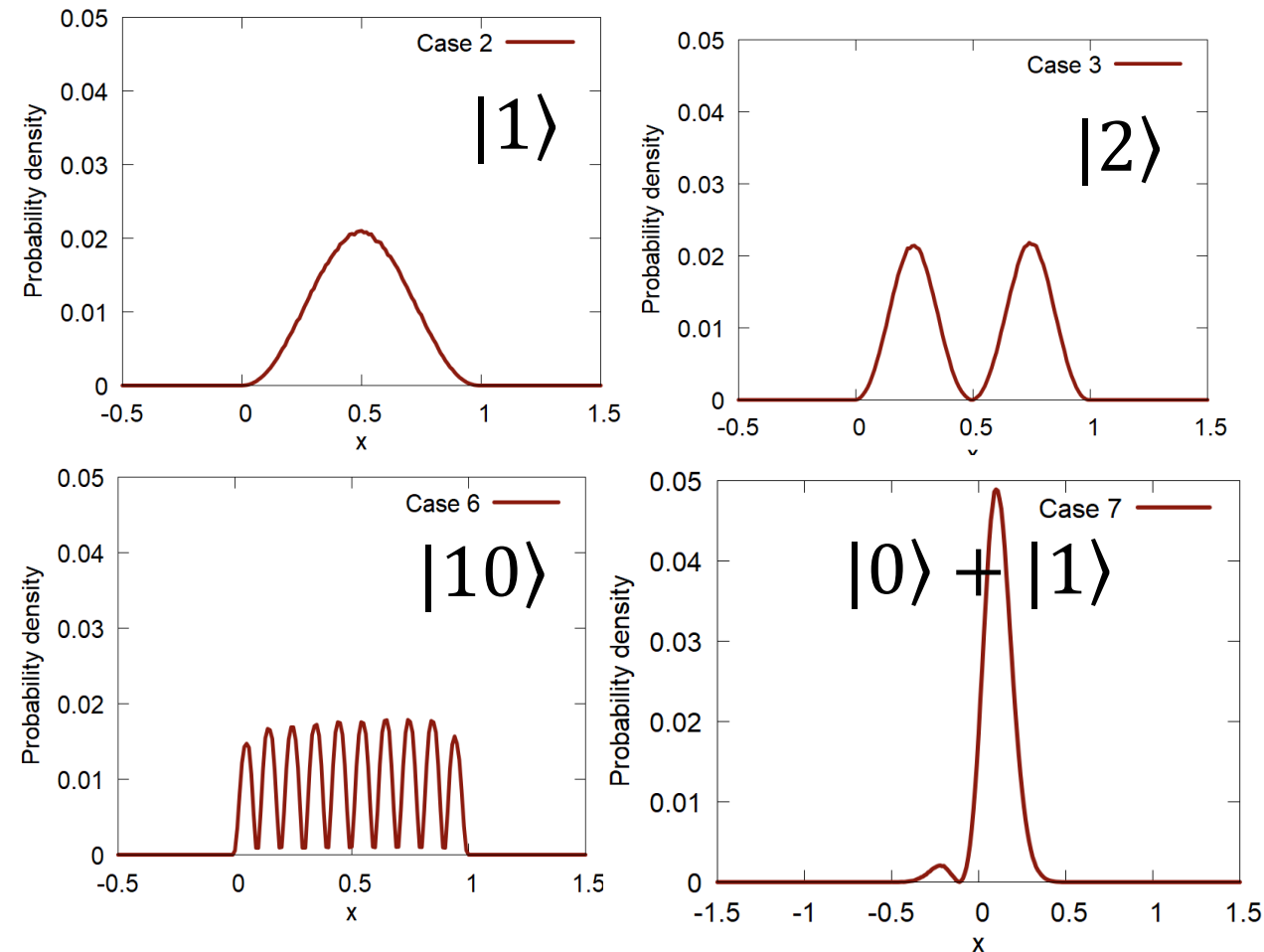
User defines what
probability distribution
function is to be sampled

The dynamical algorithm is in C++, but...
Don't need to implement the model in C++

Initial Conditions: Metropolis Sampling

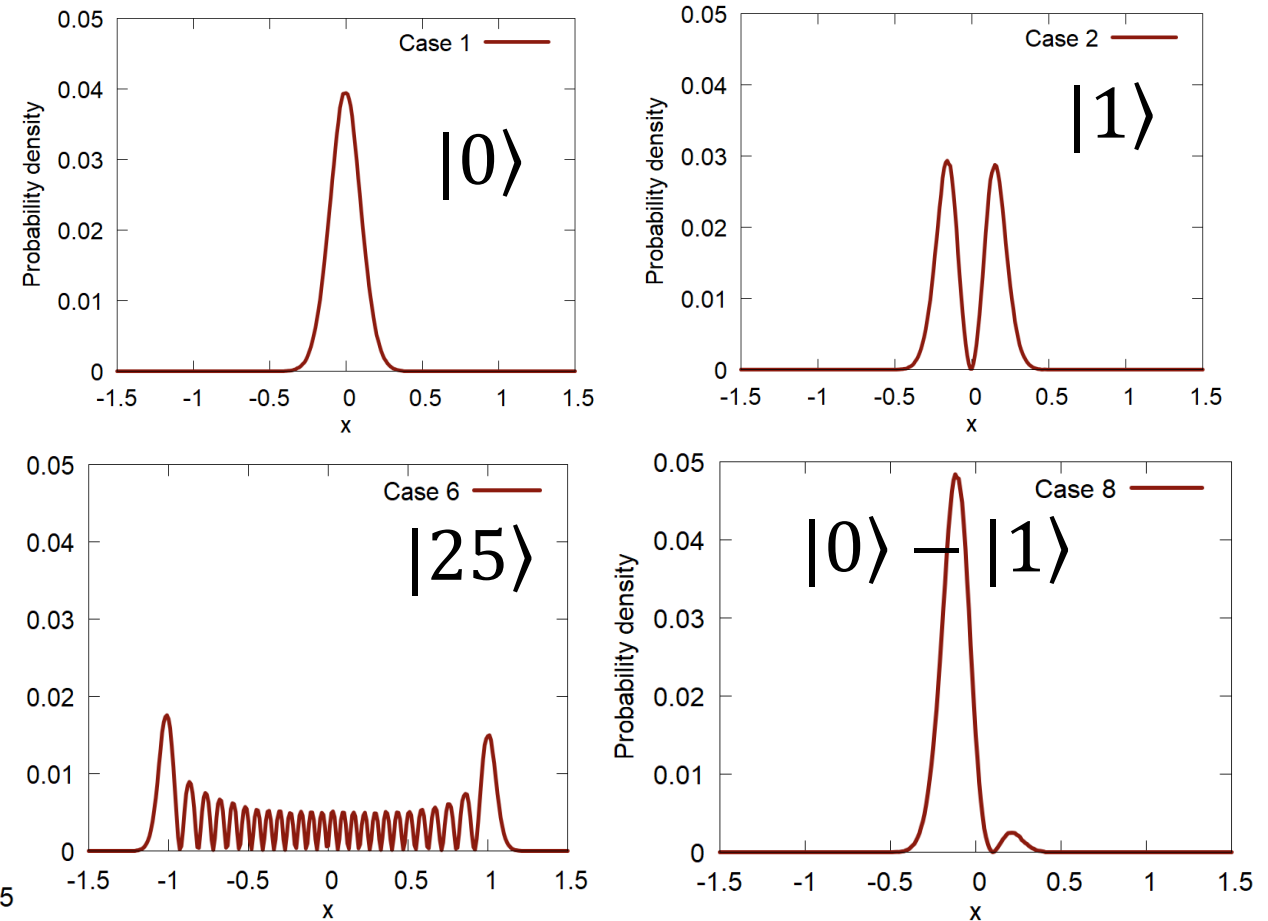
Particle in a box

$$\psi_n(q) \sim \sin\left(\frac{\pi n q}{L}\right)$$



Harmonic oscillator

$$\psi_n(q) \sim H_n(q\sqrt{\alpha}) \exp\left(-\frac{\alpha q^2}{2}\right)$$



Why Contribute?

- Make your methods broadly available (mutually advertise)
- Be listed on the developers/contributors lists, get credentials, get citations
- Make your methods compatible with other methods, enable easier interfacing – this facilitates new methods developments
- Take advantage of other methods/functions/data types available in the same code – learn once then swim
- Take advantage of improvements of other parts of the code
- Mutually ensure best standards and facilitate bug discovery/testing

How to Contribute?

Received: 18 April 2020 | Revised: 19 May 2020 | Accepted: 8 June 2020
DOI: 10.1002/qua.26373

SOFTWARE NEWS & UPDATES



Hierarchical equations of motion in the Libra software package

Story Temen¹  | Amber Jain² | Alexey V. Akimov¹ 

¹Department of Chemistry, University at Buffalo, The State University of New York, Buffalo, New York, USA

Abstract

We report the implementation of a hierarchical equations of motion (HEOM) module

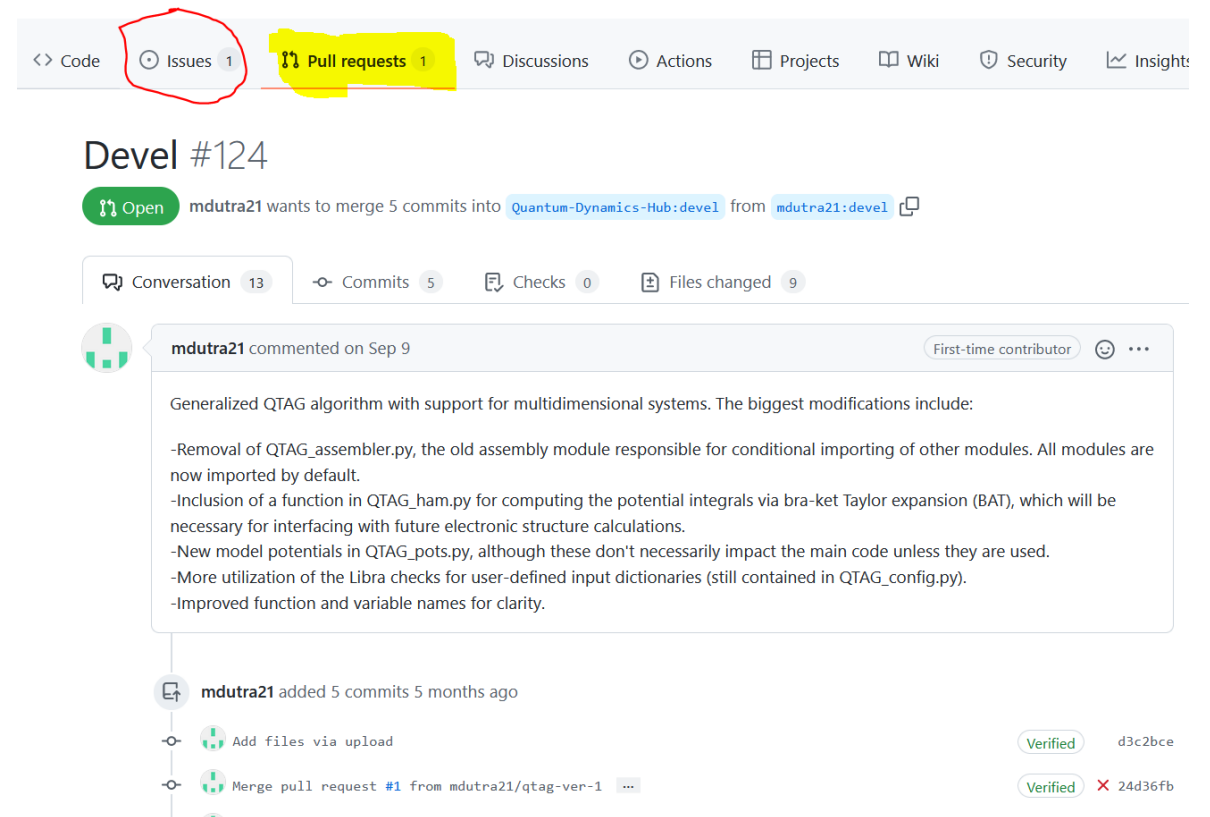
https://github.com/amber-jain-group-iitb/heom_amber

Before:

- hard-coded inputs (recompile for all parameters)
- Fortran into executable

After:

- General-purpose code, any inputs
- Python/C++, integrate with the plotting scripts, etc.



The screenshot shows a GitHub pull request interface. At the top, the navigation bar includes 'Code', 'Issues' (circled in red), 'Pull requests' (highlighted in yellow), 'Discussions', 'Actions', 'Projects', 'Wiki', 'Security', and 'Insights'. The main heading is 'Devel #124'. Below it, a green 'Open' button is followed by the text 'mdutra21 wants to merge 5 commits into Quantum-Dynamics-Hub:devel from mdutra21:devel'. A summary bar shows 'Conversation 13', 'Commits 5', 'Checks 0', and 'Files changed 9'. The pull request description, by user 'mdutra21' (marked as a 'First-time contributor'), details 'Generalized QTAG algorithm with support for multidimensional systems' and lists four modifications: removal of QTAG_assembler.py, inclusion of a function in QTAG_ham.py, new model potentials in QTAG_pots.py, and more utilization of Libra checks. The commit history shows 'mdutra21 added 5 commits 5 months ago', including 'Add files via upload' (verified, d3c2bce) and 'Merge pull request #1 from mdutra21/qqag-ver-1' (verified, 24d36fb).

- create a pull-request
- open an issue
- start a discussion (haven't tried yet)