Using Colvars in NAMD

Jérôme Hénin

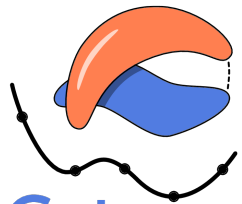
Laboratoire de Biochimie Théorique, IBPC

CNRS – Université Paris Cité

NAMD / VMD Workshop - Auburn

June 25, 2024

Colvars in the software ecosystem



Colvars

interfaces

back-ends

tinkerHP

NAMD
Scalable Molecular Dynamics

FAST. FLEXIBLE. FREE.
GROMACS

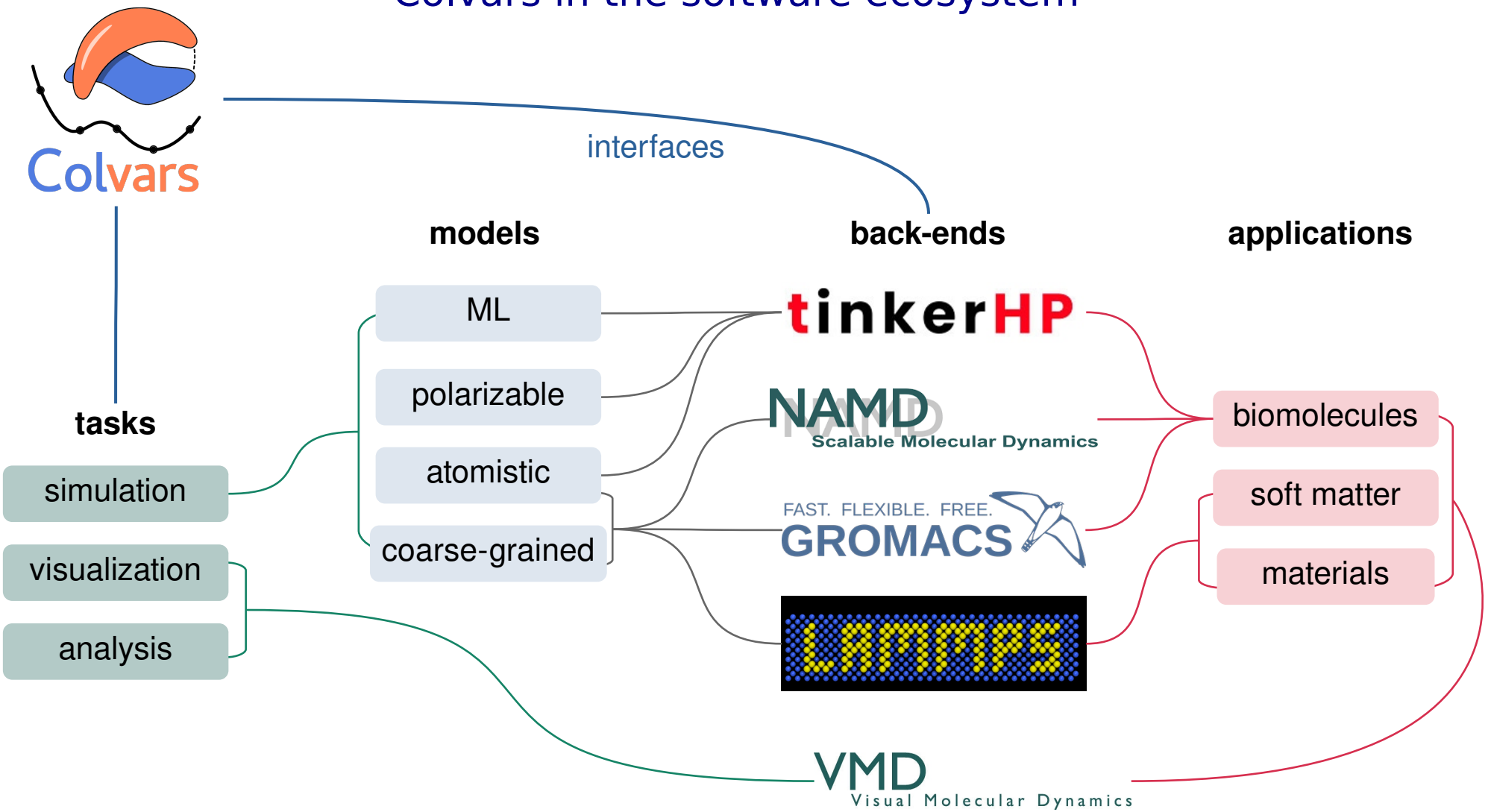
LAMMPS

tasks

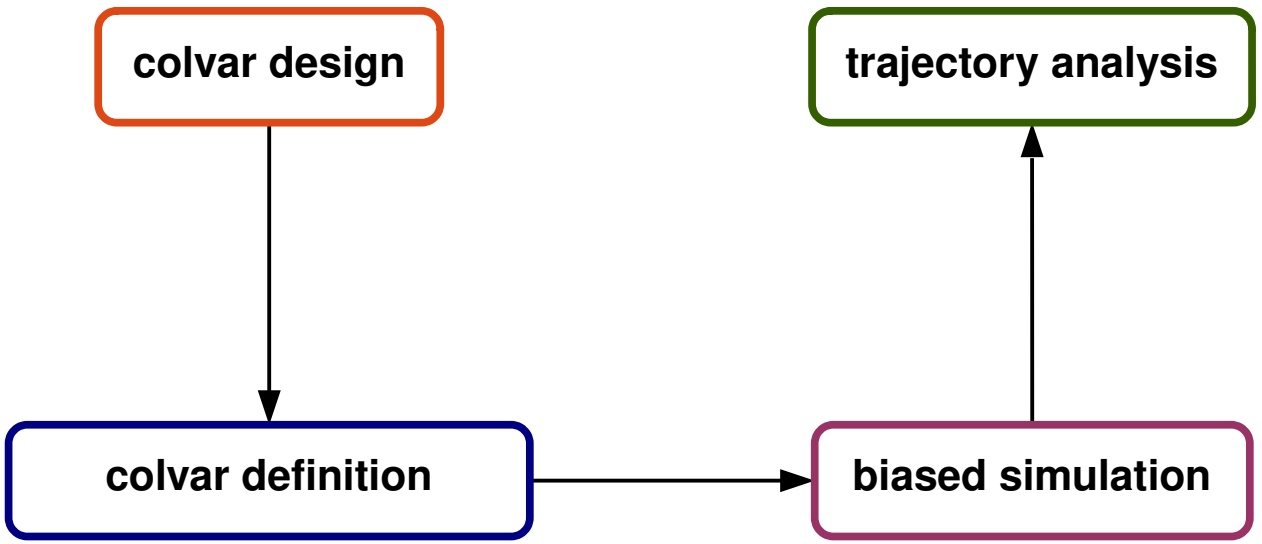
- simulation
- visualization
- analysis

VMD
Visual Molecular Dynamics

Colvars in the software ecosystem



Colvars workflow



1. Basic input
2. Preparation with VMD
3. Simulation with NAMD
4. Making the most of scripts
5. Visualization and troubleshooting with VMD

1

Basic input

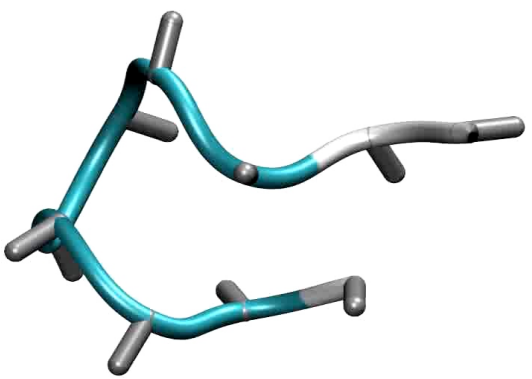
User input for Targeted MD

```
colvar {
  name RMSD

  rmsd {
    atoms {
      atomsFile  beta.pdb
      atomsCol   0
    }
    refPositionsFile beta.pdb
  }
}


harmonic {
  colvars RMSD

  centers 5.3
  targetCenters 0.0
  targetNumSteps 200000
  forceConstant 100.
}
```

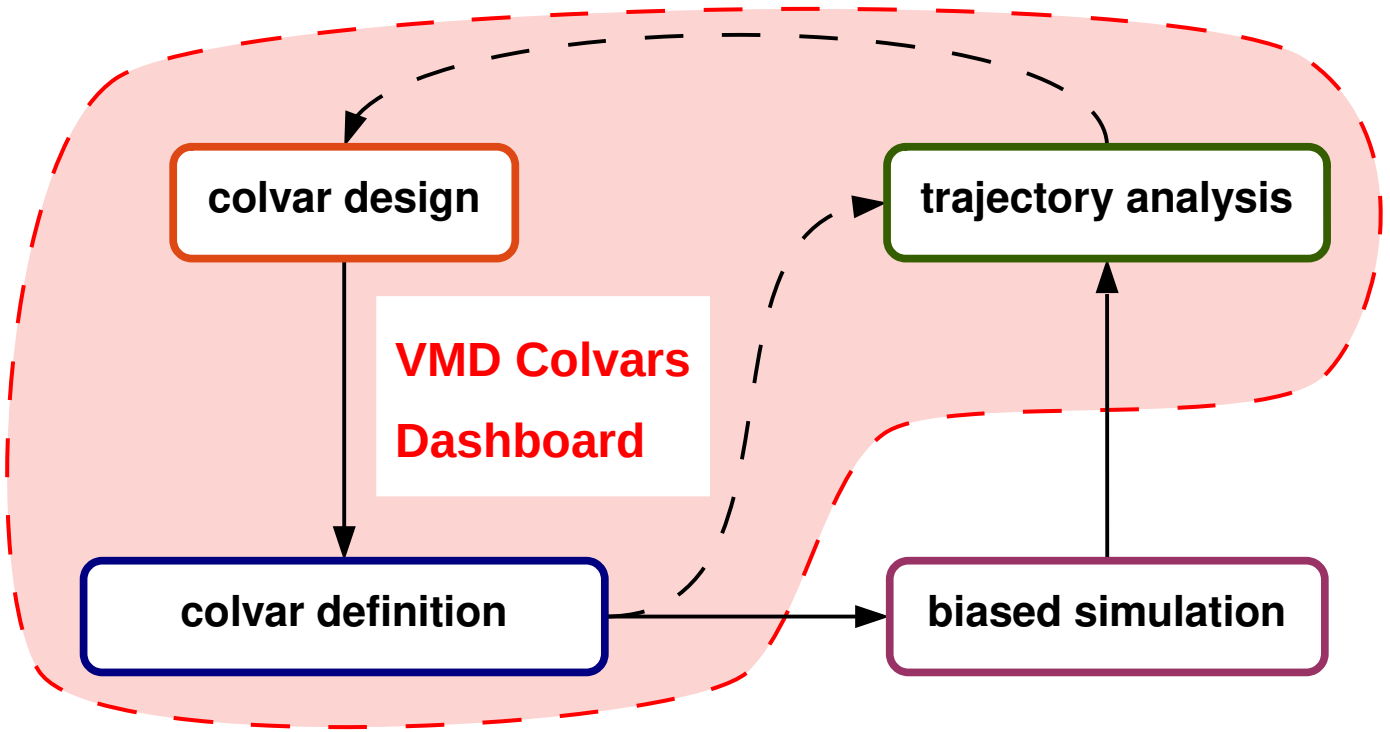


Colvars + NAMD simulation

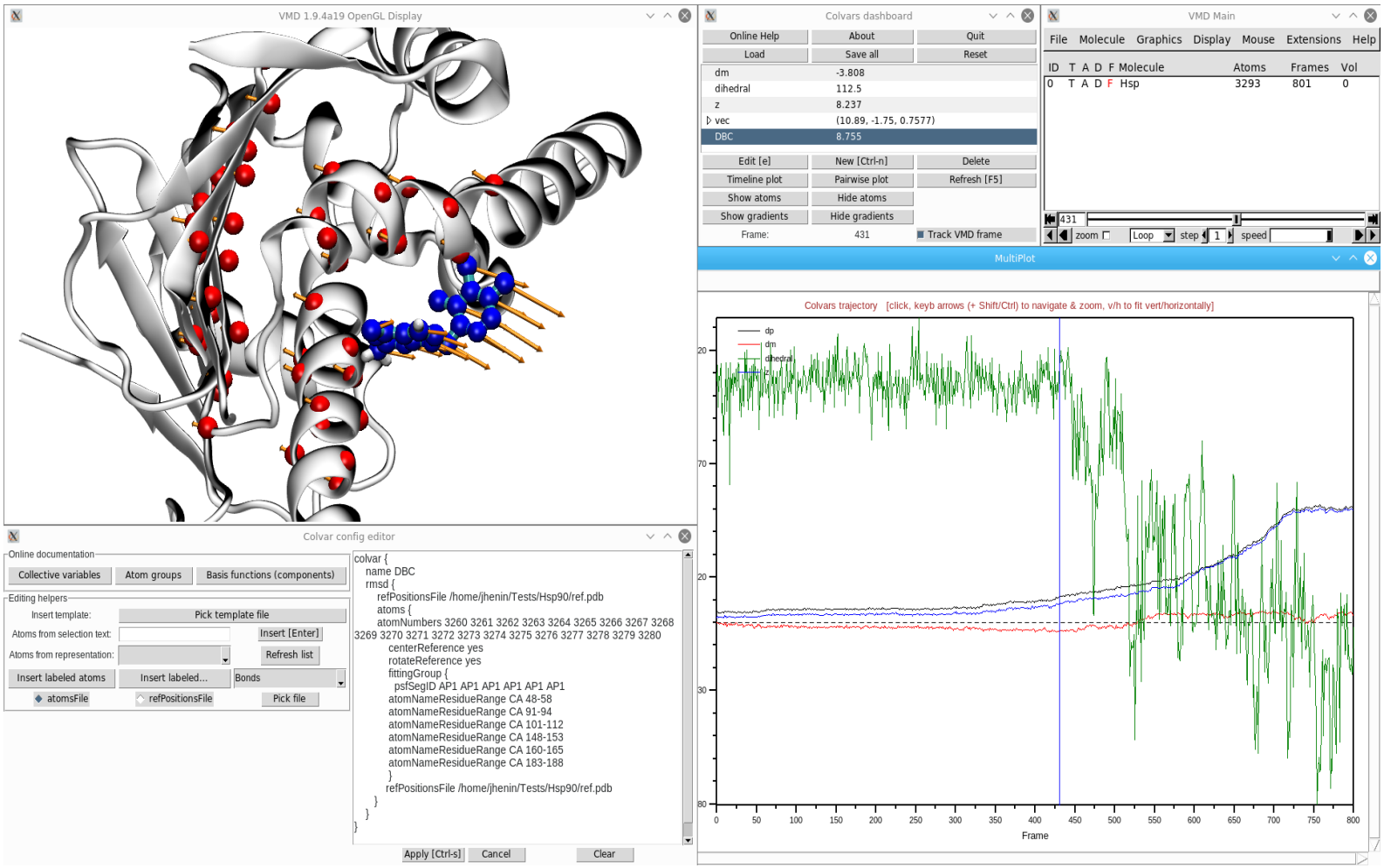
2

Preparation with VMD

Colvars workflow



Interactive Colvars Dashboard in VMD



5-minute tour: https://youtu.be/wZdcIVmf_yY

Hénin, Lopes, Fiorin, JCTC 2022

3

Simulation with NAMD

Running a Colvars/NAMD simulation

- activate Colvars in NAMD's config file:

```
colvars        on
colvarsConfig  my_cfg.colvars
colvarsInput   previous_run.colvars.state  # Restart
```

- run NAMD:

```
namd3 +p8 mysim.namd > mysim.log &
```

Reading the Colvars output

```
colvars: Initializing the collective variables module, version 2024-06-04.
colvars: Please cite Fiorin et al, Mol Phys 2013:
colvars:   https://doi.org/10.1080/00268976.2013.813594
colvars: as well as all other papers listed below for individual features used.
colvars: This version was built with the C++11 standard or higher.
colvars: Summary of compile-time features available in this build:
colvars:   - SMP parallelism: enabled (num. threads = 8)
colvars:   - Lepton custom functions: available
colvars:   - Tcl interpreter: available
colvars: Redefining the Tcl "cv" command to the new script interface.
colvars: Using NAMD interface, version "2023-12-05".
```

library version

Reading the Colvars output

```
colvars: Initializing the collective variables module, version 2024-06-04.
colvars: Please cite Fiorin et al, Mol Phys 2013:
colvars:   https://doi.org/10.1080/00268976.2013.813594
colvars: as well as all other papers listed below for individual features used.
colvars: This version was built with the C++11 standard or higher.
colvars: Summary of compile-time features available in this build:
colvars:   - SMP parallelism: enabled (num. threads = 8)
colvars:   - Lepton custom functions: available
colvars:   - Tcl interpreter: available
colvars: Redefining the Tcl "cv" command to the new script interface.
colvars: Using NAMD interface, version "2023-12-05".
```

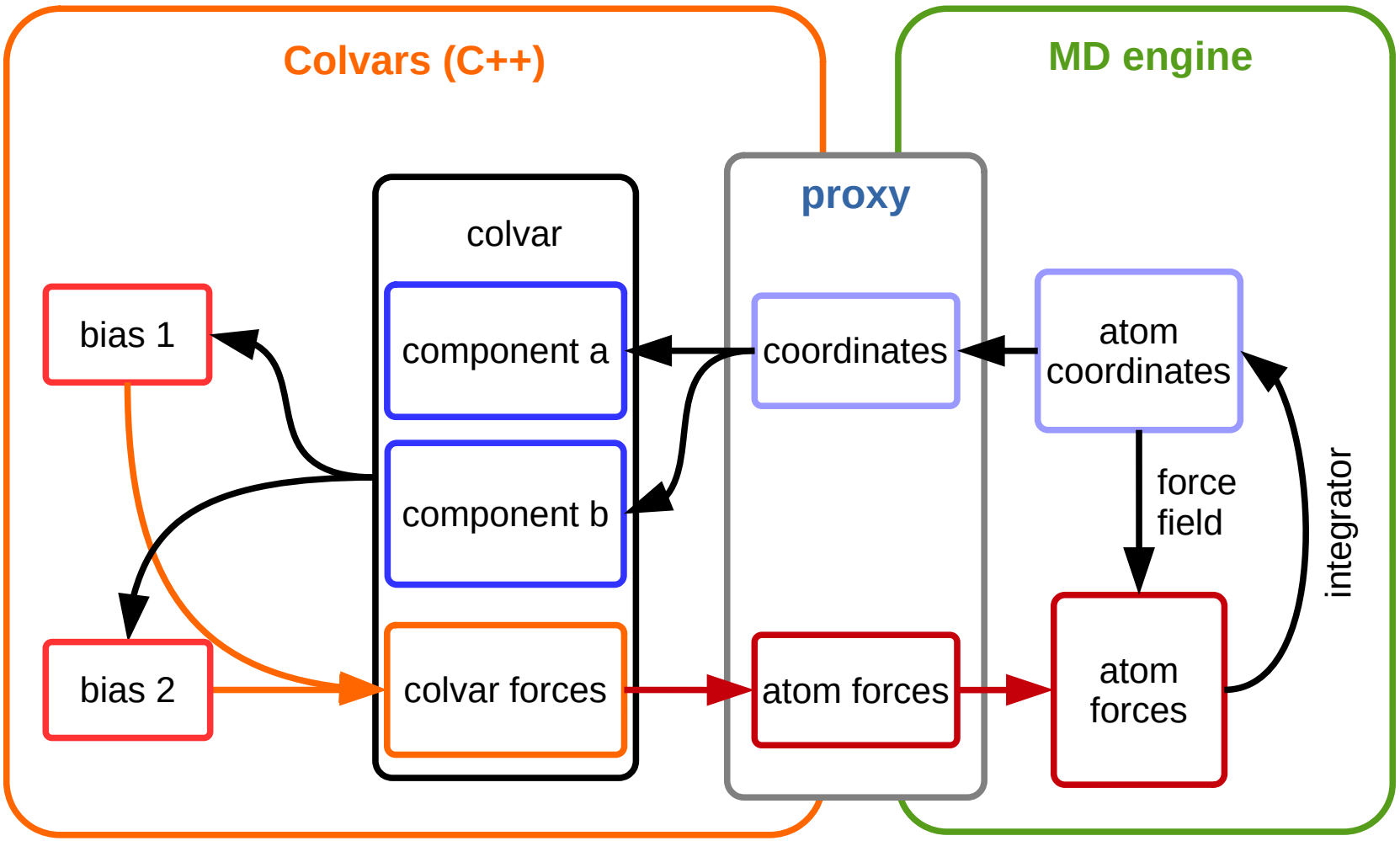
references to cite

Reading the Colvars output

```
colvars: Initializing the collective variables module, version 2024-06-04.
colvars: Please cite Fiorin et al, Mol Phys 2013:
colvars:   https://doi.org/10.1080/00268976.2013.813594
colvars: as well as all other papers listed below for individual features used.
colvars: This version was built with the C++11 standard or higher.
colvars: Summary of compile-time features available in this build:
colvars:   - SMP parallelism: enabled (num. threads = 8)
colvars:   - Lepton custom functions: available
colvars:   - Tcl interpreter: available
colvars: Redefining the Tcl "cv" command to the new script interface.
colvars: Using NAMD interface, version "2023-12-05".
```

features

Colvars Module under the hood



Performance issues

- performance bottlenecks:
  - computation
  - communication
- Colvars always runs on the CPU
  - takes advantage of multi-core CPUs (smp option)
- NAMD3 can run GPU-resident
  - requires data transfers from GPU to main memory
  - faster simulations → more probable that colvar calculation is a bottleneck

Performance tuning

- most computationally expensive colvar: coordNum (n x m)
  - can be improved with pair lists
- communication-intensive colvars:
  - anything with many atoms!

- solutions:
  - tune the performance of NAMD first (GPU-resident, etc.)
  - use as few atoms as possible for colvars
  - enable smp, reserve several cores
  - for slow colvars, enable timeStepFactor (MTS)

4

Making the most of scripts

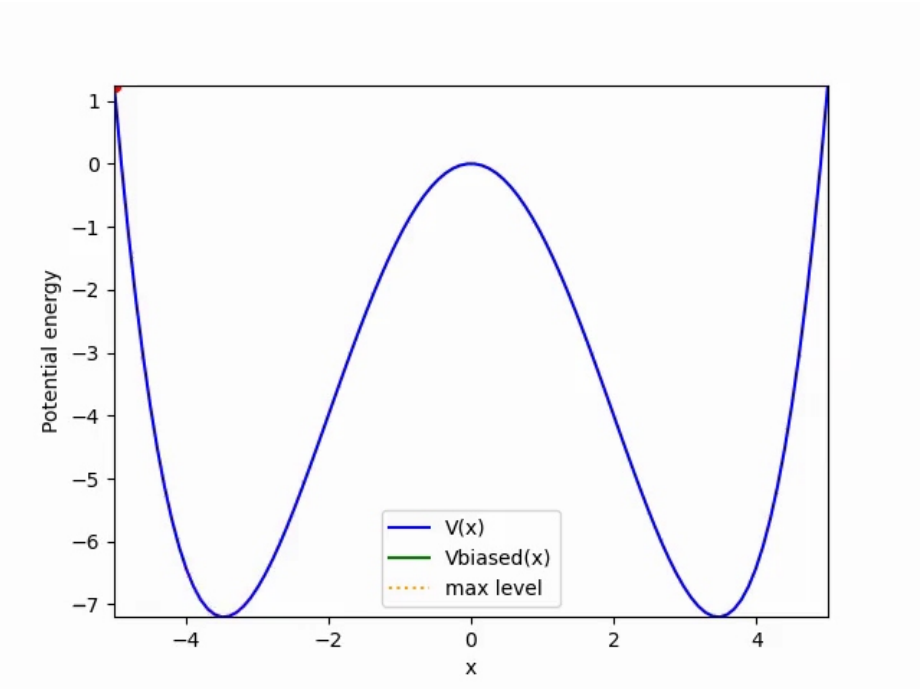
User-facing flexibility

- new variables can be created **at run-time** as combinations of basis functions:
  - polynomial
  - symbolic expressions (Lepton)
  - Tcl scripts
- new biasing algorithms can be written as Tcl scripts
- the NAMD config file is a Tcl script

Adiabatic Bias MD: selecting forward fluctuations

M. Marchi & P. Ballone JCP 1999

- one-sided **harmonic restraint** with constant **k**
- **ratchet** mechanism: restraint follows maximum level



ABMD: a quick implementation

Colvars / Tcl script:

```
proc calc_colvar_forces {ts} {
  set z [cv colvar z value]
  if {$z > $max} {                                  ;# above high-water mark?
    if {$z <= $target} {set max $z}                 ;# raise it until target
  } else {
    cv colvar z addforce [expr $k * ($max - $z)]    ;# else apply bias
  }
}                                                   ;# that's it
```

In Colvars config:

```
scriptedColvarForces on

colvar {
    name z
    ...
```

Scripting NAMD with Colvars

```
set terminate false


while { ! $terminate } {
    set cv_value [cv colvar $myCV value]
    cv config [new_cfg $cv_value]   ;# change Colvars config
    if [criterion $cv_value] {
        set terminate true          ;# terminate depending on CV
    }
    run $nSteps                     ;# advance simulation
}
```

5

Visualization and troubleshooting with VMD

Demo

- visualize
  - atoms
  - forces
- check PBCs
  - broken up atom groups?
- compare colvar trajectories

back to Giacomo