# CP2K Developers Meeting

July 22nd, 2024 14:00-16:00 CEST
(see also https://www.cp2k.org/dev:meetings)

# CP2K Developers Meeting

1. Current Development Efforts (all)
2. MiMiC framework for multiscale modeling (Andrej Antalik)
3. DBCSR update
4. OpenCL based GPU support in DBCSR and DBM/DBT
5. Current Issues when running CP2K (all)
6. Feature Deprecation (all)
7. Current status of references & Ideas for optimization (all)
8. CP2K Release (all)
9. Open CP2K-Related Positions (all)
10. CP2K-related Events (all)

# Current Development Efforts

What are you currently working on or planning to work on?

- CP2K HFX ERI acceleration (FPGA, GPU) ongoing
- ERI on FPGA: Xilinx U280 implementation accepted at FPL conference
- Hutter: PASC Harris functional in CP2K
- Kühne: Discussion with NVIDIA on library for multicenter-integrals (AMD is also interested in ERIs)
- Brehm: On-the-fly force fields in CP2K (NHR project granted)

# CASUS

- support for DFT-D4
  - integrated the github package by Grimme
  - gradient / stress tensor working
- i-Pi master code committed to the trunk
- refactor FFT backend
  - GPU offloading with SpFFT (already used by SIRIUS)
  - introduce 2D-FFTs (building block for parallel FFTs)
  - later: MPI capabilities of FFTW/SpFFT
- Finite-Temperature RPA: currently testing energies
- Sigma-RPA by A. Görling
- planned: GFN2-xTB (Anna Hehn also working on GFN2-xTB, also Jürg Hutter is working on the non-SCF GFN0) + ev. “GFN3” & gTB
- future: k-points
- Interaction/Cooperation with Nvidia (start with generalized EVP), also the case for DLA-Futures from CSCS

# Regensburg

- real-time GW-BSE

  (analog to RT-TDDFT but with screened exchange self-energy instead of exchange-correlation potential, by Stepan Marek, first implementation gives correct numbers for molecules, todo: more benchmark calculations)

- GW with full k-point sampling for band structures for small cells (gives correct numbers, todo: more benchmark calculations)

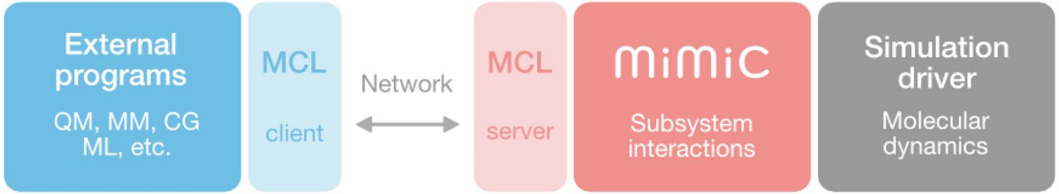
- RI basis sets for TZV(2)P-MOLOPT, whole periodic table

- to be started in August: RT-TDDFT with k-points for small cells

# What is MiMiC

Open-source framework for multiscale modelling

- Loosely couples external programs
- Computes interactions (currently QM/MM)
- Adopts MPMD model with MPI-based communication



J. Chem. Phys., 2024, 161, 022501

**Key features**

- Implemented functionalities easily transferable
- Avoids overheads
- Flexible in terms of resource allocation
- Each program can be optimized independently

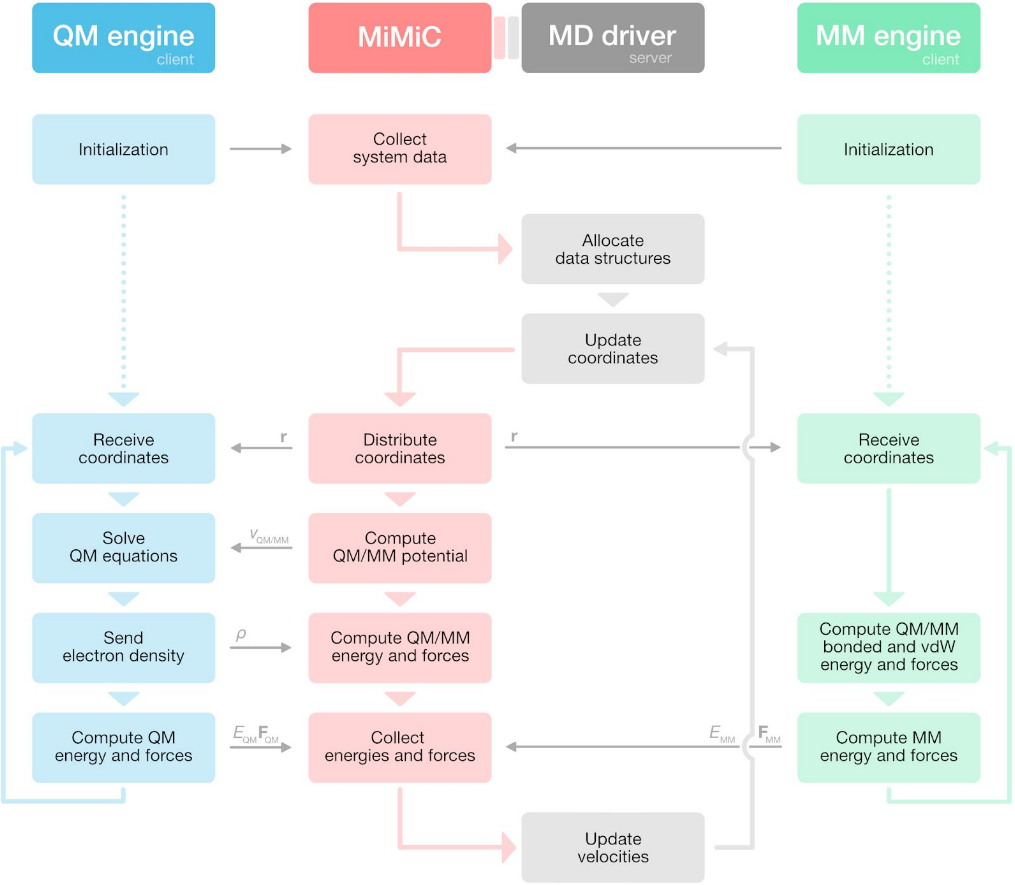
# MiMiC
# How does it work

## Launch

```
mpirun -np 4 server_exec  : \
       -np 4 client1_exec : \
       -np 8 client2_exec
```

## Client implementation

```
while (is_last_step) do
    command = mimic_receive_command()
    if (command == MCL_RUN_ENERGY_FORCES) then
        energy, forces = comp_energy_forces()
    else if (command == MCL_SEND_FORCES) then
        mimic_send_forces(forces)
    else if (command == MCL_...) then
        ...
    else if (command == MCL_EXIT) then
        is_last_step = true
    else
        Abort("Unrecognized MiMiC command!")
    end
end
```



QM engine client

MiMiC

MD driver server

MM engine client

Initialization

Collect system data

Initialization

Allocate data structures

Update coordinates

Receive coordinates

Distribute coordinates

Receive coordinates

Solve QM equations

Compute QM/MM potential

Send electron density

Compute QM/MM energy and forces

Compute QM/MM bonded and vdW energy and forces

Compute QM energy and forces

Collect energies and forces

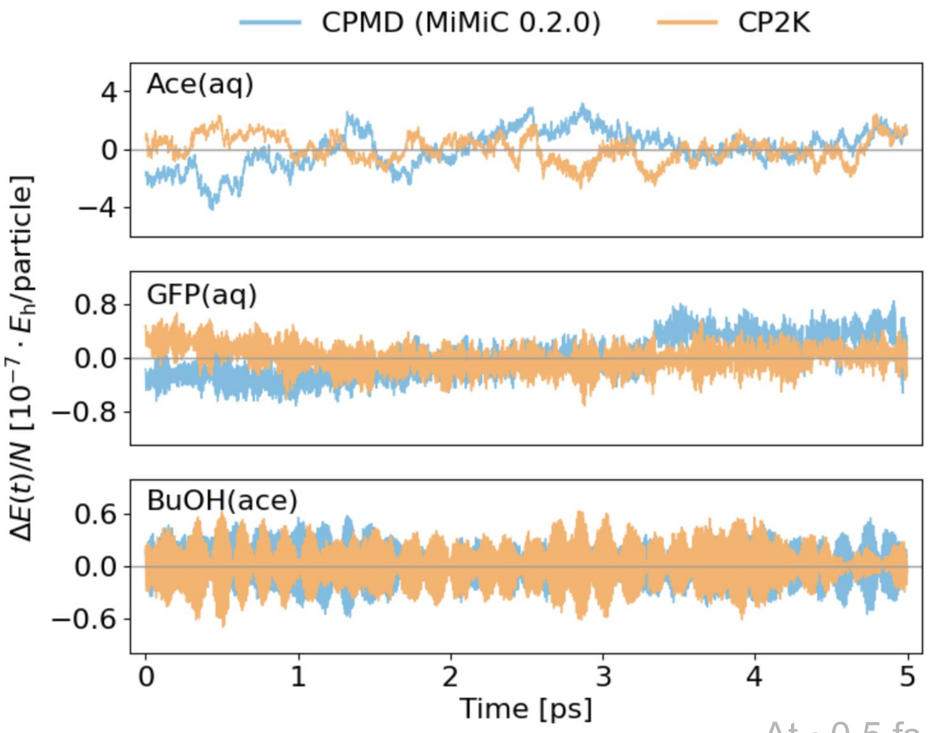
Compute MM energy and forces

Update velocities

MiMiC

# Current status (in CP2K)

**It works with GPW!**



# atoms

3k

30k

140k

Δt ~0.5 fs

(...and in MiMiC)

**Currently implemented**

- Electrostatic embedding QM/MM with long-range via multipole expansions
- Clients
  - GROMACS
  - CPMD
  - OpenMM

**In progress**

- Polarizable embedding
- GPU Offloading
- DRESP charges
- Clients
  - CP2K
  - TinkerHP
  - QuantumESPRESSO
  - …and more

# DBCSR Update (Alfio)

- Current workflow:
  - GPU kernels require autotuning, otherwise fall-back to CPU if kernels are not available (but we do move data to the GPU!)
  - Users should run the autotuning procedure to add more kernels, and then recompile CP2K. This is not feasible for a centralized CP2K installation.
- New workflow (intermediate solution):
  - Provide a generic kernel for any missing tuned kernel
  - Print a warning to inform users that the generic (not optimized) kernel is used
  - Waiting CSCS people (thanks a lot to Augustin and Rocco) for testing it on Alps (H100), then release a new DBCSR version for the next CP2K release
  - Thanks to everyone who have contributed to DBCSR!
- Next workflow:
  - Rely on vendor batched optimized libraries, possible drop autotuning entirely

# OpenCL based GPU support in DBCSR and DBM/DBT

**DBCSR**: benchmarks/QS_DM_LS/H2O-dft-ls.NREP4.inp

| Systems | 1 | 2 | 4 |
|---|---|---|---|
| 4xPVC (8T) | 177 | 110 | 64 |
| 2xPVC (4T) | 231 | 135 | 80 |
| 2xH100pci* | 247 | | |
| 2xSPR-56c | 393 | 225 | 121 |

CPU baseline also represents the host system for any of the GPU results. No matter of the individual best case, always 16 MPI-ranks have been used. CPU relied on DDR5 DIMMs. Superscaling, e.g., from 110s to 80s can be due to fit in LLC.

*H100 based on CUDA/HIP in DBCSR&DBM (OCL result in braces). Results for DBCSR use parameters tuned for A100.

**DBM/DBT**: benchmarks/QS_low_scaling_GW/GW.inp

| Systems | 1 | 2 | 4 |
|---|---|---|---|
| 4xPVC (8T) | 307 | 166 | 107 |
| 2xPVC (4T) | 398 | 212 | 132 |
| 2xH100pci* | 491 (510) | | |
| 2xSPR-56c | 464 | 229 | 138 |

HPC fabric (beside of GPU2GPU) was HDR200 in all cases like for multi-node result (2 or 4 systems).

All results are total time to solution [s].

Performance is only representative for above workloads (not for GPU vs CPU or in general)

# OpenCL based GPU support, next steps, etc.

Status

- On-par or better than CUDA based implementation on Nvidia GPUs
- Fully interoperable/tested with CUDA code on Nvidia GPUs (similarly for HIP based code-path; not tested)
- DBCSR: hard-coded rules for default kernels and comprehensive set of tuned parameters (P100, V100, A100, H100, Mi250, PVC); tuning ~3min. per kernel)
- OpenCL BE in DBCSR: GPL → BSD-3
- DBM/DBT: perf. on par with CUDA based implementation

Next steps

- CMake based toggles/build, and ARCH-file for LRZ SMUCp2
- OpenCL BE from DBCSR to be recycled (separate hfp/libxstream repository)
- Support for SpLA, ELPA, local GEMM, and CP2K's GRID component
- Tuned parameters for DBM/DBT?

Other

- LIBXSMM for ARM64 ready (SHA can be shared; release model changed)
- LIBINT code-gen flow (script)

OpenCL is an industry standard with broad/native vendor support (AMD, ARM, Intel, Nvidia). Major diff. with CUDA is it is practically JIT-only (but no special compiler needed), also C++ support was only added in OpenCL 3.0 (as an extension); otherwise C99/C11 standard.

# Current Issues when Running CP2K

- mpiwrap: get number of processes per system (issue #3565)
  - Enables hints on perf/comm. constraints like "total number of ranks to be square number", etc.
  - Enables MAX_MEMORY=auto (-1) in various input sections aka max. memory per process
- Improve assertions
  - Require reference to documentation for typical issues (aka landing page)
    - Example: Cholesky decomposition failed → typical causes can be documented
    - Collect issues occuring relatively often (according to Google groups/forum)
  - Introduce backtrace plus "phone home", etc. (even online service for crashes)
  - Distinction between "cannot continue" vs. "development expectation not met" is not tested (NDEBUG or CPABORT vs CPASSERT)
    - Ban CPASSERT from innermost loops?
- Embrace Open Source
  - Reward contributions to amplify CP2K's importance (beyond GitHub stats)
  - Revive https://www.cp2k.org/performance?

# Feature Deprecation

- PW implementation of HFX: deprecate in a later release
- debug code XC (not tested anymore): deprecate in a later release
- -D__SCALAPACK vs -D__parallel: check if redundant, remove if redundant, prefer -D__parallel
- PEXSI: mark as deprecated in this release, then drop
- QUIP: mark as deprecated in this release, then drop

# Optimization Ideas: A Glimpse into AMD’s MI300A APU and its benefit for CP2K

- A brief presentation by Gina Sitaraman, AMD

# How CP2K can benefit from Unified Memory Programming Model in AMD's MI300A GPUs

Gina Sitaraman, AMD DC GPU team

AMD @ CP2K Developer Meeting
July 22nd, 2024

AMD
together we advance_

# Solving Porting Challenges

1. CP2K: Current GPU implementation status
2. M300A: What is an APU?
3. Managed Memory and APU Programming Model

# CP2K: GPU Porting Status

- Several GPU backends in CP2K (PW, GRID, …) and dependencies (DBCSR, COSMA, ELPA, etc.)

- In H2O-RPA-* benchmarks, 65% of runtime is still on CPU
  - Integral calculations, multi-threaded on CPU for efficient scaling within MPI rank
  - MPI Communication that becomes a bottleneck for large systems

- Can the integral computations be offloaded to GPU?

- On discrete GPUs, moving data back and forth is expensive
- On APUs, data movement cost is zero or nearly so
  - An opportunity to port much more of the computation of the RPA benchmark to GPUs
  - For existing GPU backends, just add unified memory support (Mathieu T. helped with CP2K's GRID and COSMA)

[Public]

# AMD Instinct™ MI300 APU

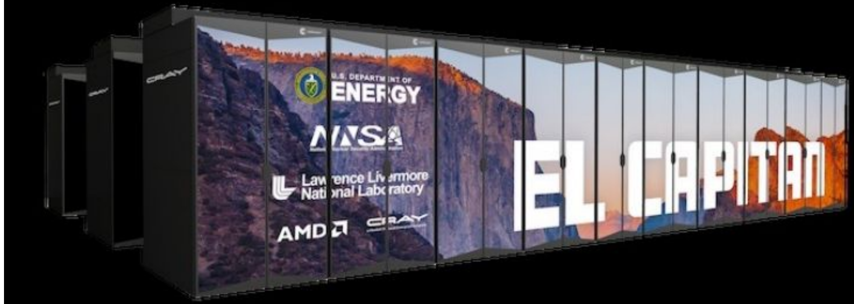
LLNL's El Capitan Exascale supercomputer will be powered by the AMD Instinct™ MI300 APU: "MI300A"

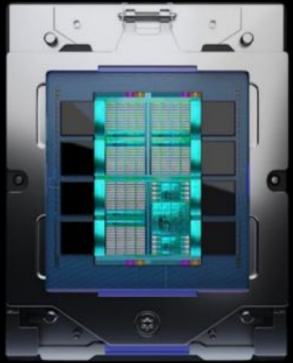
MI300A is an APU, with AMD CDNA™ 3 GPUs, Zen 4 CPUs, cache memory, and HBM chiplets in a single package

24 Zen4 CPU cores

128 GiB of HBM3

*"It's much easier to program"*


U.S. DEPARTMENT OF ENERGY
NNSA
Lawrence Livermore National Laboratory
AMD
CRAY
EL CAPITAN



The world's first integrated data center CPU + GPU

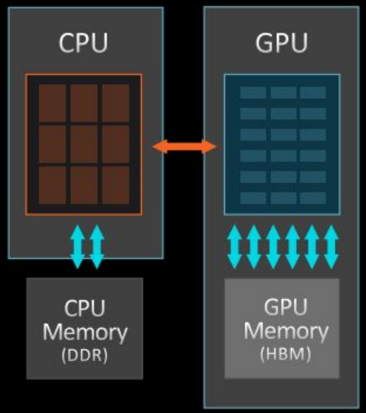
AMD INSTINCT™

MI300

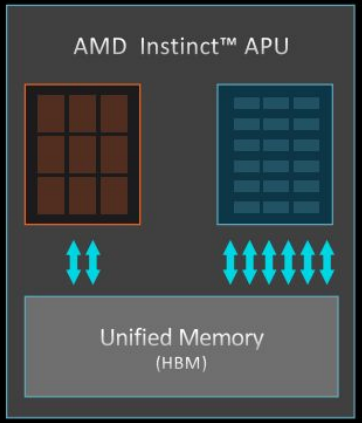
Breakthrough architecture to power the exascale AI era

# AMD MI300A UNIFIED MEMORY APU ARCHITECTURE BENEFITS

AMD CDNA™ 2 Coherent Memory Architecture → AMD CDNA™ 3 Unified Memory APU Architecture



- **Eliminate Redundant Memory Copies**
- **No programming distinction between host and device memory spaces**
- **High performance, fine-grained sharing between CPU and GPU processing elements**
- **Single process can address all memory, compute elements on a socket**



# APU PROGRAMMING MODEL

## CPU CODE

```
double* in_h = (double*)malloc(Msize);
double* out_h = (double*)malloc(Msize);


for (int i=0; i<M; i++) // initialize
   in_h[i] = …;

cpu_func(in_h, out_h, M);


for (int i=0; i<M; i++) // CPU-process
  … = out_h[i];
```

## GPU CODE

```
double* in_h = (double*)malloc(Msize);
double* out_h = (double*)malloc(Msize);
hipMalloc(&in_d, Msize);
hipMalloc(&out_d, Msize);

for (int i=0; i<M; i++) // initialize
   in_h[i] = …;
hipMemcpy(in_d,in_h,Msize);
gpu_func<< >>(in_d, out_d, M);
hipDeviceSynchronize();
hipMemcpy(out_h,out_d,Msize);

for (int i=0; i<M; i++) // CPU-process
  … = out_h[i];
```

## APU CODE

```
double* in_h = (double*)malloc(Msize);
double* out_h = (double*)malloc(Msize);


for (int i=0; i<M; i++) // initialize
   in_h[i] = …;

gpu_func<< >>(in_h, out_h, M);
hipDeviceSynchronize();


for (int i=0; i<M; i++) // CPU-process
  … = out_h[i];
```

- GPU memory allocation on Device
- Explicit memory management between CPU & GPU
- Synchronization Barrier

https://github.com/amd/HPCTrainingExamples/tree/main/ManagedMemory

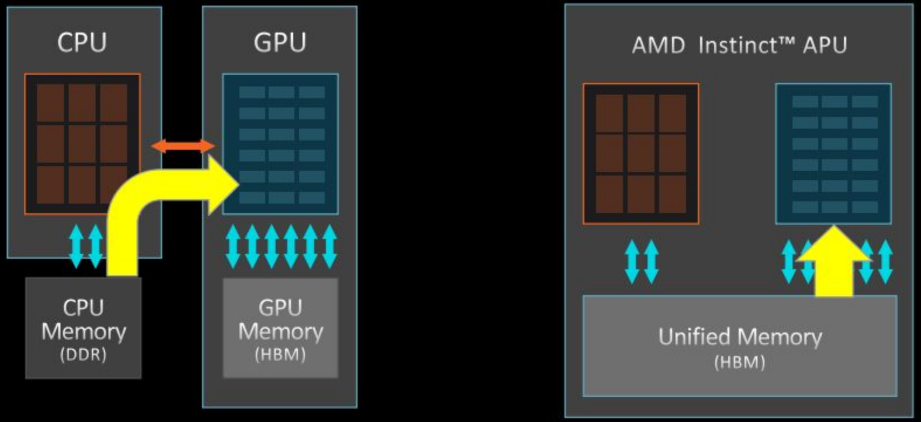
# APU PROGRAMMING: PERFORMANCE IMPLICATIONS

## GPU CODE

```
double* in_h = (double*)malloc(Msize);
double* out_h = (double*)malloc(Msize);
hipMalloc(&in_d, Msize);
hipMalloc(&out_d, Msize);


for (int i=0; i<M; i++) //initialize
   in_h[i] = …;
hipMemcpy(in_d,in_h,Msize);
gpu_func<< >>(in_d, out_d, M);
hipDeviceSynchronize();
hipMemcpy(out_h,out_d,Msize);

for (int i=0; i<M; i++) // CPU-process
  … = out_h[i];
```



| Operation | MI250X (MCM) | MI300A |
|---|---|---|
| **H2D Copy** | O(10) GB/s | O(TB/s) |

- GPU memory allocation on Device
- Explicit memory management between CPU & GPU
- Synchronization Barrier

# APU PROGRAMMING: PERFORMANCE IMPLICATIONS

## APU CODE

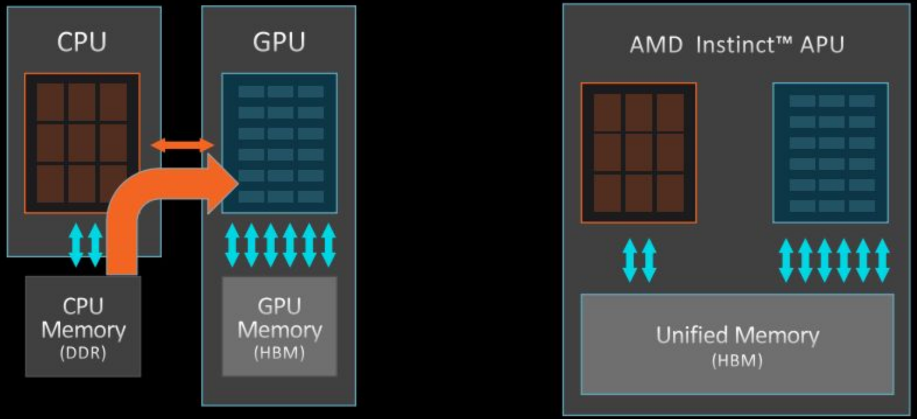
```
double* in_h = (double*)malloc(Msize);
double* out_h = (double*)malloc(Msize);



for (int i=0; i<M; i++) //initialize
   in_h[i] = …;

gpu_func<< >>(in_h, out_h, M);
hipDeviceSynchronize();


for (int i=0; i<M; i++) // CPU-process
  … = out_h[i];
```



| Operation | MI250X (MCM) | MI300A |
|---|---|---|
| Coherent Access | O(10) GB/s | N/A |

- ~~GPU memory allocation on Device~~
- ~~Explicit memory management between CPU & GPU~~
- Synchronization Barrier

# Conclusions

- MI300A is an APU with a unified memory space between CPUs and GPUs

- The APU's unified memory architecture offers great benefits:
  - It's much easier to program!
  - HIP code should run "out of the box"
  - HIP code can be further optimized to remove redundant copies

- Code with unified memory access can be tested on MI200 GPUs
  - `export HSA_XNACK=1`

- Could we work together to offload more compute from CPU intensive regions of CP2K?
  - Are there good candidates to write kernels for?
  - Are there experts who we could work with?

# Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Third-party content is licensed to you directly by the third party that owns the content and is not licensed to you by AMD. ALL LINKED THIRD-PARTY CONTENT IS PROVIDED "AS IS" WITHOUT A WARRANTY OF ANY KIND. USE OF SUCH THIRD-PARTY CONTENT IS DONE AT YOUR SOLE DISCRETION AND UNDER NO CIRCUMSTANCES WILL AMD BE LIABLE TO YOU FOR ANY THIRD-PARTY CONTENT. YOU ASSUME ALL RISK AND ARE SOLELY RESPONSIBLE FOR ANY DAMAGES THAT MAY ARISE FROM YOUR USE OF THIRD-PARTY CONTENT.

# manual.cp2k.org

Input Reference

- Input descriptions support Latex-formulas, Markdown, and unicode incl. emoji.
- Special treatment for XC_SECTIONS.
- Highlighting of mentioned keywords ⭐.
- New precommit check for missing spaces in multi-line descriptions.

Methods Section

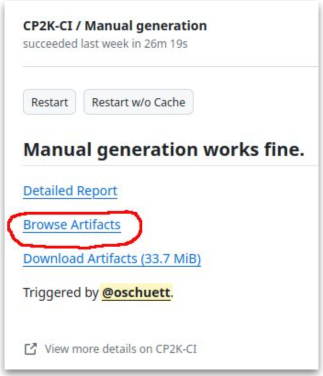
- Curated structure, see e.g. the section on X-Ray or HFX.
- Lot's of great new pages, e.g. about metadynamics and GW.
- Getting started: README.md
- **Still missing: (contributions needed)**
  - **DFTB, Implicit Solvation, NMR, Raman, Nudged elastic band, Path Integrals**
  - **Overview pages that, e.g. Optical Spectroscopy**

Example Inputs

- Currently: moving from wiki to GitHub

Search engine:

- **either improve sphinx internal search or hook in external search, e.g. Algolia (contribution needed). See also #343.**



**CP2K-CI / Manual generation**

succeeded last week in 26m 19s

**Manual generation works fine.**

Detailed Report

Browse Artifacts

Download Artifacts (33.7 MiB)

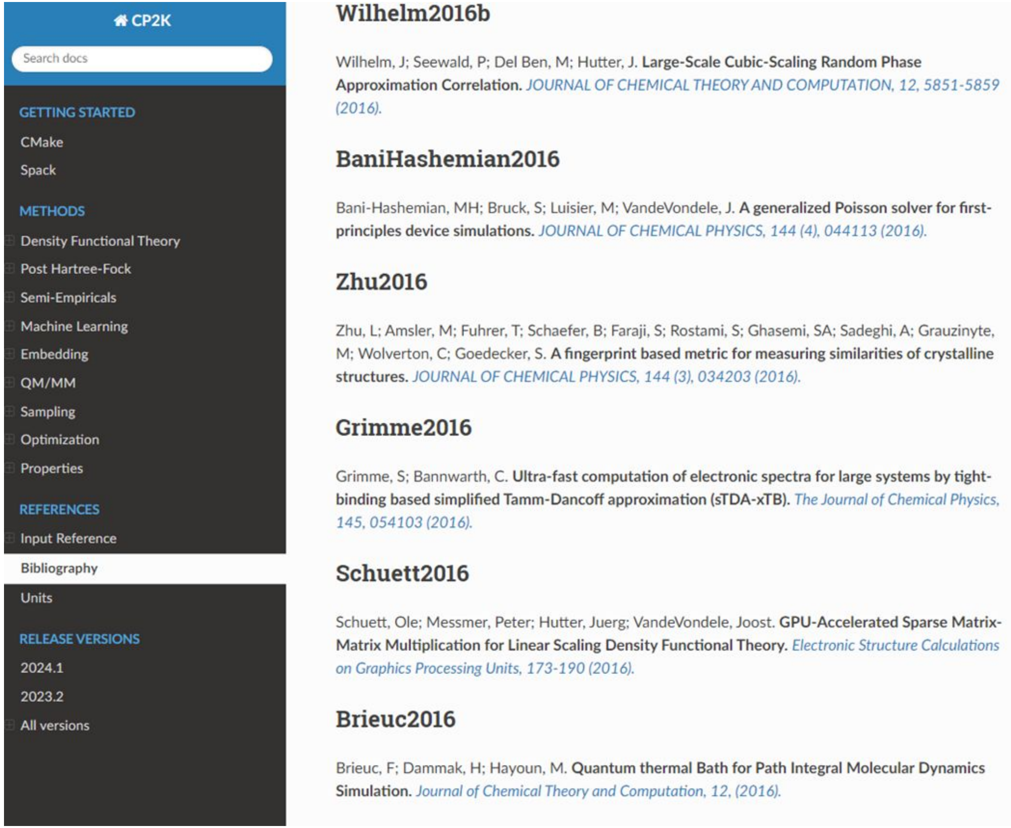
Triggered by **@oschuett**.

View more details on CP2K-CI

Live preview for pull requests

# Current status of references

https://manual.cp2k.org/trunk/bibliography.html



### Wilhelm2016b

Wilhelm, J; Seewald, P; Del Ben, M; Hutter, J. **Large-Scale Cubic-Scaling Random Phase Approximation Correlation.** *JOURNAL OF CHEMICAL THEORY AND COMPUTATION, 12, 5851-5859 (2016).*

### BaniHashemian2016

Bani-Hashemian, MH; Bruck, S; Luisier, M; VandeVondele, J. **A generalized Poisson solver for first-principles device simulations.** *JOURNAL OF CHEMICAL PHYSICS, 144 (4), 044113 (2016).*

### Zhu2016

Zhu, L; Amsler, M; Fuhrer, T; Schaefer, B; Faraji, S; Rostami, S; Ghasemi, SA; Sadeghi, A; Grauzinyte, M; Wolverton, C; Goedecker, S. **A fingerprint based metric for measuring similarities of crystalline structures.** *JOURNAL OF CHEMICAL PHYSICS, 144 (3), 034203 (2016).*

### Grimme2016

Grimme, S; Bannwarth, C. **Ultra-fast computation of electronic spectra for large systems by tight-binding based simplified Tamm-Dancoff approximation (sTDA-xTB).** *The Journal of Chemical Physics, 145, 054103 (2016).*

### Schuett2016

Schuett, Ole; Messmer, Peter; Hutter, Juerg; VandeVondele, Joost. **GPU-Accelerated Sparse Matrix-Matrix Multiplication for Linear Scaling Density Functional Theory.** *Electronic Structure Calculations on Graphics Processing Units, 173-190 (2016).*

### Brieuc2016

Brieuc, F; Dammak, H; Hayoun, M. **Quantum thermal Bath for Path Integral Molecular Dynamics Simulation.** *Journal of Chemical Theory and Computation, 12, (2016).*

# Ideas for optimization

Unify reference format in Phys. Rev. style:

A. Author, B. Author, *Title of the paper*, Abbrev. J. Name **VOL**, pages (year)

Every entry could provide exclusively:

- Authors
- Title
- Abbreviated journal name, large & small caps
- Volume
- Pages (one entry)
- Year
- doi

Ole might adjust common/bibliography.F and routines to generate the bibliography webpage

Jan can volunteer to clean old references and to bring them to updated format

# Regtest Improvements

- Done: No more false positive slow test (#3534)
- Ongoing: Names instead of numbers for test types (please help)
- Planned: Check other outputs besides the stdout
- Planned: Allow for multiple checks after running an input file
- Discussion: #2996

# CP2K-Release

- waiting for dbcsr-alps validation (1-2 weeks)

# Open CP2K-Related Positions

- position at CASUS: see slide CP2K@CASUS
- two positions at PC2:
  - Postdoc as HPC expert (f/m/d) for atomistic simulation - Focus on Plane-Wave DFT (https://pc2.uni-paderborn.de/fileadmin/pc2/job_offers/6463.pdf)
  - Postdoc as HPC expert (f/m/d) for quantum chemistry (https://pc2.uni-paderborn.de/fileadmin/pc2/job_offers/6462.pdf)

# CP2K-Related Events:

Plans:

- Paderborn+CASUS/HZDR:
  - ~Q1/25: Gromacs & CP2K on QM/MM (school with tutorial, 3-4 days in person)