Exascale in der computergestützten Chemie: Neue Methode zur Elektronenstruktursimulation für Millionen Atome

Robert Schade
Tobias Kenter, Hossam Elgabartty, Michael Lass, Ole Schütt, Alfio Lazzaro, Hans Pabst, Stephan Mohr, Jürg Hutter, Thomas D. Kühne, Christian Plessl

Universität Paderborn, Germany
Paderborn Center for Parallel Computing, PC²

ZKI AK SC Frühjahr 2022
Intensive Scientific Support at PC2 for NHR projects:

- Physics and Chemistry, especially:
  - Atomistic Simulations
  - Optoelectronics and Quantum Photonics
  - Quantum Computing
  - High-Energy Physics
- Machine Learning
- Heterogeneous Computing with Accelerators

This project: Prof. Dr. Thomas Kühne, Dynamics of Condensed Matter (DCM), Paderborn University

pc2.de/go/millionatoms
Mixed Precision for Molecular Dynamics Simulations
Molecular Dynamics Simulations

Virtual Microscope

HIV-1 capsid in aqueous solution

⇒ study chemical and biological properties of large molecules like viruses

pc2.de/go/millionatoms
Mixed-Precision for Molecular Dynamics

Rough energy cost for operations (in 45 nm):

<table>
<thead>
<tr>
<th>precision</th>
<th>Addition</th>
<th>Multiplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP16</td>
<td>0.4 pJ</td>
<td>1.1 pJ</td>
</tr>
<tr>
<td>FP32</td>
<td>0.9 pJ</td>
<td>3.7 pJ</td>
</tr>
</tbody>
</table>

from M. Horowitz, "1.1 Computing’s energy problem (and what we can do about it),” 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014

⇒ use of **low and mixed precision** is key for energy-efficient simulations

NVIDIA A100 third-generation tensor core peak performances (with boost clock)

<table>
<thead>
<tr>
<th>Multiply</th>
<th>Accumulate</th>
<th>Peak performance in TFLOP/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP16</td>
<td>FP32</td>
<td>312</td>
</tr>
<tr>
<td>TF32</td>
<td>FP32</td>
<td>156</td>
</tr>
<tr>
<td>FP64</td>
<td>FP64</td>
<td>19.5</td>
</tr>
</tbody>
</table>

pc2.de/go/millionatoms
Mixed-Precision for Molecular Dynamics

Combine all Levels of the Computer Technology Stack:

- **Algorithm**
  - Non-orthogonal local submatrix (NOLSM) method compensation of numerical noise

- **Application**
  - CP2K: quantum chemistry and solid-state theory program

- **Libraries**
  - NVIDIA cuBLAS with streams and CUDA graphs

- **Computer Architecture**
  - Third generation NVIDIA tensor cores with mixed FP16/FP32 support

- **Semiconductor Technology**
  - NVIDIA A100 Tensor-core GPU (TSMC 7 nm FinFET)
Low/Mixed Precision can be compensated

- Trajectory $R_i(t)$ for every atom are not of interest
  - huge amount of data
  - depends on initial positions and velocities of atoms
- **Instead**: thermodynamical averages
  \[
  \langle A \rangle = \frac{1}{Z} \int d^{3N}R \int d^{3N}p \ e^{-E(R,p)/(k_BT)} \cdot A(R,p)
  \]
  - noise from numerical approximations can be compensated with a modified Langevin-type equation of motion for the atoms
  \[
  M_i \ddot{R}_i = F_i - \gamma_N M_i \dot{R}_i
  \]
  ⇒ **accurate thermodynamical averages** can be obtained!

Rengaraj, Lass, Plessl, and Kühne [2020]

pc2.de/go/millionatoms
How to obtain the Forces?

- Forces on the atoms are obtained from the total energy $E$ of the system
  \[
  F_i = -\frac{dE}{dR_i}
  \]

- **Important:** quantum-mechanical effects and long-range interactions
  ⇒ we use the GFN-xTB approach (Grimme, Bannwarth, and Shushkov [2017])
  ⇒ solve a **quantum-mechanical electronic-structure problem** in every step

- Conventional approaches: generalized eigenvalue problem
  \[
  H_0 c_i = \epsilon_i S c_i \quad \Rightarrow \quad \mathcal{O}(N^3)
  \]

- But: Hamiltonian matrix $H_0$ and overlap matrix $S$ are **sparse**
Solution of the Electronic-Structure Problem

Density-matrix-based approach with matrix-sign function:
VandeVondele, Borstnik, and Hutter [2012]

\[ E_{\text{elec}} = \text{Tr}(H_0 D) \quad \text{with} \quad D = \frac{1}{2}(I - \text{sign}(S^{-1}H_0 - \mu I))S^{-1} \]

- matrix sign function and inversion can be evaluated iteratively

\[
\begin{align*}
\text{sign}(x) &= \frac{x}{|x|} = \frac{x}{\sqrt{x^2}} \\
\text{sign}(A) &= A \cdot (A^2)^{-1/2}
\end{align*}
\]

\[
\begin{align*}
X_0 &= A \\
X_{i+1} &= \frac{1}{2}X_i \cdot (3I - X_i^2) \\
\text{sign}(A) &= \lim_{i \to \infty} X_i
\end{align*}
\]

- linear-scaling approach \( O(N) \)
- only **multiplications** of distributed sparse matrices are required
- but usually bound by inter-node communication!

pc2.de/go/millionatoms
Electronic-structure based molecular dynamics:

1. positions of atoms $R_i$
2. $\Rightarrow$ Hamiltonian matrix $H_0$ and overlap matrix $S$
3. (...heavy linear algebra (matrix function) ...)
4. $\Rightarrow$ density matrix $D$
5. $\Rightarrow$ total energy $E$
6. $\Rightarrow$ Forces $F_i$
7. $\Rightarrow$ new positions $R_i$
Submatrix Method
Submatrix Method: General Idea

Purpose: Estimate matrix function (e.g. sign or inversion) of a large sparse matrix

Lass, Mohr, Wiebeler, Kühne, and Plessl [2018]
Lass, Schade, Kühne, and Plessl [2020]
Submatrix Method: General Idea

Step 1: Identify nonzero values in every column

Lass, Mohr, Wiebeler, Kühne, and Plessl [2018]
Lass, Schade, Kühne, and Plessl [2020]
Submatrix Method: General Idea

Step 2: Build submatrix $T_i(A)$ for every column $i$ with only the rows that have non-zero elements.
Submatrix Method: General Idea

Step 3: Apply matrix function $f$ to submatrices $\mathcal{T}_i(A)$

Lass, Mohr, Wiebeler, Kühne, and Plessl [2018]
Lass, Schade, Kühne, and Plessl [2020]
Submatrix Method: General Idea

Step 4: copy resulting columns to result matrix

Lass, Mohr, Wiebeler, Kühne, and Plessl [2018]

Lass, Schade, Kühne, and Plessl [2020]
Submatrix Method: General Idea

Properties of the Submatrix method:
- large distributed sparse matrix ⇒ many small dense matrices
- suitable for dense linear algebra
- massively parallel
- linear-scaling approach

Lass, Mohr, Wiebeler, Kühne, and Plessl [2018]
Lass, Schade, Kühne, and Plessl [2020]
Non-Orthogonal Local Submatrix Method (NOLSM)
Non-Orthogonal Local Submatrix Method (NOLSM)

- view density matrix $D(H_0, S)$ as a matrix function
  
  $$f = D(H_0, S) = \frac{1}{2}(I - \text{sign}(S^{-1}H_0 - \mu I))S^{-1}$$

- **Non-Orthogonal**: submatrix construction is applied to the non-orthogonal matrices $H_0$ and $S$

- **Local**: the submatrix construction is performed locally, no matrix entries are transferred
  
  $\Rightarrow$ communication-avoiding

Schade, Kenter, Elgabarty, Lass, Schütt, Lazzaro, Pabst, Mohr, Hutter, Kühne, and et al. [2022]
Non-Orthogonal Local Submatrix Method (NOLSM)

Implementation with GPUs:

⇒ minimal communication between nodes and CPU-to-GPU

Schade, Kenter, Elgabarty, Lass, Schütt, Lazzaro, Pabst, Mohr, Hutter, Kühne, and et al. [2022]

pc2.de/go/millionatoms
Non-Orthogonal Local Submatrix Method (NOLSM)

1. transfer of data for a submatrix of dimension $n$
   - atomic positions, atom species
   - intermediate quantities of xTB approach
   \[\mathcal{O}(n)\] data

2. construction of matrix elements of the submatrices
   - Overlap matrix $S$: contracted Gaussian-type orbitals
     - complete loop unrolling with automatic code generation
     - use single-precision floating-point transcendental functions in NVIDIA GPUs
   - Hamiltonian matrix $H_0$ in GFN-xTB
     Grimme, Bannwarth, and Shushkov [2017]
   \[\mathcal{O}(n^2)\] computation

pc2.de/go/millionatoms
Non-Orthogonal Local Submatrix Method (NOLSM)

1. transfer of data for a submatrix of dimension $n$
   - atomic positions, atom species
   - intermediate quantities of xTB approach
   \[ \Rightarrow \mathcal{O}(n) \text{ data} \]

2. construction of matrix elements of the submatrices $T_j(S)$, $T_j(H_0)$ \[ \Rightarrow \mathcal{O}(n^2) \text{ computation} \]

3. computation of the density matrix $T_j(D)$ and energy-weighted density matrix $T_j(W)$
   \[ X_0 = A \]
   \[ X_{i+1} = \frac{1}{2} X_i (3I - X_i^2) \]
   \[ \text{sign}(A) = \lim_{i \to \infty} X_i \Rightarrow \mathcal{O}(n^3) \text{ computation} \]
Matrix-Iterations on NVIDIA A100 in FP16/FP32

Customized "batching" for matrix iterations:

- Kernel from cuBLAS
- +CUDA streams: to load the full GPU
- +CUDA graphs: reduce kernel launch latency

⇒ for intermediate sized matrices ($\geq 1000$) about 60%-85% of peak can be reached

pc2.de/go/millionatoms
Non-Orthogonal Local Submatrix Method (NOLSM)

1. transfer of data for a submatrix of dimension $n$
   - atomic positions, atom species
   - intermediate quantities of xTB approach
   \[\mathcal{O}(n)\] data

2. construction of matrix elements of the submatrices $T_j(S), T_j(H_0)$
   \[\mathcal{O}(n^2)\] computation

3. computation of the density matrix $T_j(D)$ and energy weighted density matrix $T_j(W)$
   \[\mathcal{O}(n^3)\] computation

4. transfer of result columns to host
   \[\mathcal{O}(n)\] data

pc2.de/go/millionatoms
Non-Orthogonal Local Submatrix Method (NOLSM)

- implemented in the quantum-chemistry and solid-state physics code **CP2K** (cp2k.org/)
  Kühne et al. [2020]

- using **DBCSR** (https://github.com/cp2k/dbcsr) for sparse matrix storage

- **Julia/CUDA C++** for matrix elements,

- **cuBLAS** for matrix operations on NVIDIA A100

Schade, Kenter, Elgabarty, Lass, Schütt, Lazzaro, Pabst, Mohr, Hutter, Kühne, and et al. [2022]

pc2.de/go/millionatoms
Scaling, Efficiency and Performance
936 GPU-nodes with each:
- CPU: 2xAMD EPYC 7402
- Memory: 512 GB DDR4-3200 RAM
- GPU: 4 × NVIDIA A100, 40 GB, NVLink3
- Network: 4 × Mellanox HDR200 InfiniBand ConnectX 6 (200 Gbit/s each)

Peak TC Performance:
- FP64: 73 PFLOP/s
- FP16/FP32: 1170 PFLOP/s
Submatrix Method: NOLSM-Scaling

Weak scaling of bulk water for 1 million atoms per node:

⇒ near perfect weak scaling
Submatrix Method: NOLSM-Scaling

Strong scaling of bulk water with 102 million atoms:

Wall time
Parallel efficiency

 wall time

⇒ near perfect strong scaling
Submatrix Method: Combination of Submatrices

Strong scaling of bulk water with 102 million atoms:

- Performance of NOLSM for bulk water
  - Peak multiply performance (312 TFLOP/s/GPU)

Average submatrix dimension $\approx 660$

$\Rightarrow \approx 43\%$ of peak reached

$\Rightarrow$ small matrix sizes limit achievable performance
Submatrix Method: Combination of Submatrices

Idea: use flexibility of submatrix method and combine submatrices with similar columns

$\Rightarrow$ fewer but larger submatrices!
Submatrix Method: Combination of Submatrices

average submatrix dimension $654 \rightarrow 1202$

and reduction of floating-point operations by factor 1.65!

pc2.de/go/millionatoms
Submatrix Method: Combination of Submatrices

Strong scaling of HIV-1 with 62.5 million atoms:

- Peak Performance
- NOLSM

⇒ ≈ 68% of peak reached

average submatrix dimension ≈ 1200

⇒ 324 PFlops FP16/FP32 for 384 nodes (68% of TC Peak)

pc2.de/go/millionatoms
Submatrix Method: Strong Scaling for HIV-1

Wall time for HIV-1 (62M) vs. Number of nodes

Efficiency

Parallel efficiency

HIV-1 capsid in aqueous solution with 62 million atoms

⇒ near perfect strong scaling

pc2.de/go/millionatoms
Can we go beyond one Exaflop? (in FP16/FP32)
Perlmutter at NERSC

- Perlmutter Phase 1: 1536 GPU-nodes with each
  - CPU: AMD EPYC 7763
  - Memory: 256 GB DDR4-3200 RAM
  - GPU: 4 × NVIDIA A100, 40 GB, NVLink3
  - Network: 2 × HPE Cray Slingshot

- Peak TC Performance:
  - FP64: 120 PFLOP/s
  - FP16/FP32: 1916 PFLOP/s
(Nearly) Full System Run on Perlmutter

- SARS-CoV-2 Spike protein in aqueous solution
- approx. 1.7 mio. atoms
- Wrapp et al., Science, 367(6483), 2020; Casalino et al., Int J High Perform Comput Appl, 35(5), 2021
- replicated in x- and y-direction
  i.e. 7x7 gives \( \approx \) 83 mio. atoms
  \( \Rightarrow \) scalable benchmark system

Spike protein in aqueous solution (right without H and O)

pc2.de/go/millionatoms
(Nearly) Full System Run on Perlmutter

- run with 1100 nodes
- 4400 NVIDIA A100 GPUs
- Three system sizes:
  - 6x5: 51 mio. atoms
  - 6x6: 61 mio. atoms
  - 7x7: 83 mio. atoms

Wall time for the submatrix method

pc2.de/go/millionatoms
(Nearly) Full System Run on Perlmutter

- run with 1100 nodes
- 4400 NVIDIA A100 GPUs
- Three system sizes:
  - 6x5: 51 mio. atoms
  - 6x6: 61 mio. atoms
  - 7x7: 83 mio. atoms
- 0.312 PFLOP/s per NVIDIA A100 theoretical peak (with boost)
- 1.248 PFLOP/s per node theoretical peak

⇒ about 1 PFLOP/s per node

pc2.de/go/millionatoms
(Nearly) Full System Run on Perlmutter

- run with 1100 nodes
- 4400 NVIDIA A100 GPUs
- Three system sizes:
  - 6x5: 51 mio. atoms
  - 6x6: 61 mio. atoms
  - 7x7: 83 mio. atoms

⇒ maximal performance of ≈ 1.1 EFLOP/s in FP16/FP32 reached

c2.de/go/millionatoms
Outlook

- extension from GFN2-xTB to DFT ⇒ more universally applicable
- exploration of other applications of the submatrix method for approximate matrix functions in physics and chemistry
Non-orthogonal local submatrix (NOLSM) method:

- for matrix functions of large sparse matrices
- massively parallel communication-avoiding method
- GPU-accelerated for NVIDIA GPUs
  - minimal transfer between host and GPUs
  - matrix construction on GPUs
  - mixed-precision on NVIDIA tensor cores
- compensation schema for numerical noise

Extended electronic-structure based molecular dynamics simulations to
- more than 100 million atoms
- beyond 1 EFLOP/s
Acknowledgments: HPC Systems Used

- **Noctua 1/2@PC²**: The authors gratefully acknowledge the funding of this project by computing time provided by the Paderborn Center for Parallel Computing (PC²).

- **JUWELS Booster@JSC**: The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS Booster at Jülich Supercomputing Centre (JSC).

- **Perlmutter@NERSC**: This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award DDR-ERCAP0022240.
Acknowledgments

- Federal Ministry of Education and Research (BMBF) and the state of North Rhine-Westphalia as part of the NHR Program.
- European Research Council (ERC) Grant under the European Union’s Horizon 2020 research and innovation program (Grant Agreement No. 716142).
- Funding from Paderborn University’s research award for ”GreenIT”
- Thanks to Thomas Muller (JSC), Paul F. Baumeister (JSC), and Markus Hrywniak (NVIDIA) for valuable discussions during the JUWELS Booster porting workshop.
Intensive Support Projects for Physics/Chemistry

Examples of current projects:

- extensions of the submatrix method
- Optimization of small and mid-sized molecular dynamics calculations
- Acceleration of electron repulsion integrals for quantum chemistry
- Functional renormalization group with Julia
- SIMULATeQCD (lattice QCD)
- new algorithms for quantum computing for chemistry and interfaces to codes (https://arxiv.org/abs/2202.02417)

pc2.de/go/millionatoms


pc2.de/go/millionatoms