Breaking the exascale barrier for the electronic structure problem in ab-initio molecular dynamics

Robert Schade1,2, Tobias Kenter1,2, Hossam Elgabarty1,3, Michael Lass1,2, Thomas D. Kühne1,3 and Christian Plessl1,2

Abstract
The non-orthogonal local submatrix method applied to electronic structure–based molecular dynamics simulations is shown to exceed 1.1 EFLOP/s in FP16/FP32-mixed floating-point arithmetic when using 4400 NVIDIA A100 GPUs of the Perlmutter system. This is enabled by a modification of the original method that pushes the sustained fraction of the peak performance to about 80%. Example calculations are performed for SARS-CoV-2 spike proteins with up to 83 million atoms.

Keywords
Ab-initio molecular dynamics, Car–Parrinello molecular dynamics, tight-binding, density functional theory, linear-scaling electronic structure theory

Introduction
Electronic structure–based ab-initio molecular dynamics (AIMD) simulations (Car et al. 1985; Payne et al., 1992; Kühne 2014) are an important tool in solid-state physics, chemistry, and material science. The explicit treatment of quantum-mechanical effects in the electronic structure is required in situations, where empirical model potentials used in classical molecular dynamics fail to describe the relevant physical or chemical phenomena.

To derive the forces acting on the atoms, the electronic structure problem has to be solved in every time step during the propagation of the atoms. To make this possible, linear-scaling methods have been developed, where the computational complexity scales only linearly with the number of atoms in the system (Goedecker, 1999; Yang, 1991; Galli et al., 1992; Richters and Kühne, 2014; Niklasson et al., 2016). We have proposed the non-orthogonal local submatrix method (NOLSM, Schade et al., 2022) as a massively parallel method to solve the electronic structure problem via an approximate solution of the required matrix functions. The local nature of the method avoids inter-node communication in the solution phase and has been shown to scale extremely well to more than one thousand GPUs, while efficiently using the mixed-precision tensor cores for linear algebra operations.

This article is building on the implementation described in Schade et al. (2022), but since we are focusing on improvements to increase the sustained peak performance, aspects like the compensation of noise from numerical approximations with an appropriately modified Langevin-type equation to obtain accurate thermodynamical expectation values are not touched here, but have been discussed in previous work (Richters and Kühne 2014; Rengaraj et al., 2020). Instead, Section II summarizes the tackled problem, whereas Section III puts the achievement in relation to the performance of related large-scale electronic structure–based structure relaxations or AIMD simulations. The innovations beyond those presented in Schade et al. (2022) are described in Section IV. In Section V-B, we discuss our evaluation and define the performance measurements. Finally, Section VI discusses the achieved performance.

1Paderborn University, Paderborn Center for Parallel Computing, Germany
2Paderborn University, Department for Computer Science, Germany
3Paderborn University, Department for Chemistry, Germany

Corresponding author:
Thomas D. Kühne, Paderborn University, Warburger Str. 100, Paderborn 33098, Germany.
Email: thomas.kuehne@uni-paderborn.de
Overview of the problem

Molecular dynamics calculations simulate the movement of atoms in molecules, surfaces, or solids by integrating Newton’s equation of motion

$$M_i \ddot{R}_i = F_i,$$  \hspace{1cm} (1)

where $M_i$ is the mass of atom $i$, $R_i$ is its position in space, and $F_i$ is the force acting on it. Thus, the evaluation of the forces acting on the atoms is required in every time step. In electronic structure–based molecular dynamics, these forces are evaluated on-the-fly from the total energy $E$ of the system via

$$F_i = -\frac{\partial E}{\partial R_i}.$$  \hspace{1cm} (2)

In turn, the total energy is not obtained from an empirical model as in classical molecular dynamics, but directly from the quantum-mechanical problem of electrons in the electrostatic field of the nuclei. The total energy of a system can be written as

$$E = E_{\text{elec}} + E_{\text{dc}} + E_{\text{ion}},$$  \hspace{1cm} (3)

where $E_{\text{elec}}$ is the electronic energy, $E_{\text{dc}}$ are the double counting terms, and $E_{\text{ion}}$ is the nuclear Coulomb repulsion energy of the atoms. The bulk of the computational effort is required to obtain the electronic energy. While this can be done efficiently for small and medium-sized systems by solving a high-dimensional eigenvalue problem (Kühne et al., 2007; Kühne et al., 2020), very large systems require methods that scale at most linearly with the size of the system in terms of number of atoms. Such linear-scaling electronic structure methods have been developed (Goedecker, 1999; Yang, 1991; Galli et al., 1992; Richters and Kühne, 2014), for example, based on the one-particle reduced density matrix $D$ (McWeeny, 1960), which allows to obtain the electronic energy via

$$E_{\text{elec}} = \text{Tr}(D H_0).$$  \hspace{1cm} (4)

At zero electronic temperature, the density matrix can be written as a matrix function

$$D = \frac{1}{2} \left( I - \text{sign}(S^{-1} H_0 - \mu I) \right) S^{-1},$$  \hspace{1cm} (5)

where $H_0$ is the one-particle Hamiltonian matrix of the system and $S$ is the overlap matrix between the basis functions that are used to describe the wave functions of the electrons. Furthermore, $\mu$ denotes the chemical potential.

The evaluation of the matrix-sign function in equation (5) can be performed iteratively, for example, with the Newton–Schulz iteration (Schulz 1933)

$$X_0 = A, X_{k+1} = \frac{1}{2} X_k (3I - X_k^2)$$  \hspace{1cm} (6)

or other iteration schemes (Richters, Lass, et al. 2019) that only require scalings, additions, and multiplications of matrices.

The submatrix method (Lass, Mohr, et al. 2018; Lass, Schade, et al. 2020) instead views the density matrix as a matrix function to be evaluated. Therein, the evaluation of a matrix function $f(A)$ is performed in three steps. The steps are performed for every column $i$ of the matrix $A$ independently and are schematically shown in Figure 1:

1. In the first step, a submatrix is generated for column $i$ of the input matrix $A$ by removing all rows and corresponding columns for which the matrix $A$ has vanishing or negligible elements in column $i$. The result is a smaller and much denser submatrix $A_{T_i}$.

2. The matrix function is applied to the submatrix $T_i(A)$, that is, $f(T_i(A))$ is evaluated.

3. The matrix elements of $f(T_i(A))$ that correspond to the column $i$ are written to the matrix $B$ in the sense that the submatrix construction of step 1 is applied in a reverse way.

The resulting matrix $B$ is an approximation of the matrix function $f(A)$ and by construction has the same sparsity pattern as $A$. The non-orthogonal submatrix method, where $D = D(H_0, S)$ is a matrix function of two matrices, combines the sparsity patterns $H_0$ and $S$ before the submatrix construction. Thus, it builds two submatrices, $T_i(H_0)$ and $T_i(S)$, and in step 2, the matrix function is

$$T_i(D) = \frac{1}{2} \left( I - \text{sign}(T_i(S)^{-1}T_i(H_0) - \mu I) \right) T_i(S)^{-1}.$$  \hspace{1cm} (8)

Note that the efficiency and accuracy of the submatrix method can be improved by generating one submatrix for multiple columns instead of just a single column as described in detail in Schade et al. (2022) together with the GPU implementation. Moreover, the use of the submatrix approximation and low-precision numerics can be compensated by a modified Schade and Kühne 2014; Rengaraj et al., 2020).

Current state of the art

Table 1 lists previous attempts to extend the boundaries of electronic structure–based structure relaxation or molecular dynamics simulations.
Innovations realized

Summary of contributions

This work uses the previously reported algorithmic innovations like the use of approximate computing techniques, the non-orthogonal local submatrix method and its realization with GPUs, while minimizing the communication, as well as the heuristic combination of columns in the submatrix creation described in Schade et al. (2022). A new development beyond the implementation innovations already shown in Schade et al. (2022) like the efficient iterative evaluation of matrix functions for dense matrices on GPU tensor cores is introduced in Section IV-B: The matrix-size dependency of the GPU performance is now also considered for the combination of submatrices and yields an additional speedup.

Implementation innovations

1. Submatrix Combination Heuristics:

   The combination of columns for the generation of submatrices introduced in Schade et al. (2022) used a cubic metric, that is, the combination of two columns yields an improvement if and only if
   \[
   (n_i + n_j - n_{ij})^3 < n_i^3 + n_j^3,
   \]
   where \( n_i \) is the dimension of the submatrix for column \( i \) and \( n_j \) for column \( j \), and \( n_{ij} \) is the number of overlapping columns. This cubic metric represents the number of floating-point operations during the evaluation of the matrix function for the submatrices, which is based on dense matrix multiplications. We propose to modify the cubic metric by including the performance characteristic of the used GPUs. For this purpose, the matrix multiplication performance \( p(n) \) of the GPUs is measured for different matrix sizes \( n \) and interpolated. The combination criterion then compares the predicted runtime \( p(n_i) \times n_i^3 \) for the matrix functions of the submatrix instead of the number of floating-point operations, that is,
   \[
   p(n_i + n_j - n_{ij}) \times (n_i + n_j - n_{ij})^3 < p(n_i) \times n_i^3 + p(n_j) \times n_j^3.
   \]
   (10)

   This criterion effectively increases the dimension of the submatrices and the achievable portion of the peak performance. Results are shown for the SARS-CoV-2 spike protein in aqueous solution with approx. 1.7 mio. atoms in Table 2 and histograms for the submatrix dimensions in Figure 2. The criterion of equation (10) approximately doubles the average submatrix dimension and slightly increases the total number of floating-point operations while drastically increasing the estimated floating-point throughput and leading to an overall estimated speedup compared to the previous criterion of equation (9).

How performance was measured

Computational details

1. SARS-CoV-2 Spike Protein in Aqueous Solution:

   As our benchmark system, we have used the full-length SARS-CoV-2 spike protein in the open state, anchored in a lipid bilayer (Reference PDB structure: 6VSB) and pre-equilibrated with all-atom MD using NAMD (Wrapp et al., 2020; Casalino et al., 2021). The system was solvated in aqueous solution in a simulation cell with dimensions 204.7 × 199.5 × 408.5 Å, and including 1,693,134 atoms. The single cell shown in Figure 3 can be easily repeated in a two-dimensional grid of spike proteins as a scalable benchmark system.

2. Simulation Details:

   The electronic structure is simulated with the GFN-xTB approach in conjunction with a London dispersion correction based on the rational Becke–Johnson damping function (Grimme et al., 2011). Further details can be found in Schade et al. (2022).

   For the sake of benchmark resources, we have restricted each simulation run to one SCF iteration in the spirit of the second-generation Car–Parrinello AIMD method (Kühne et al., 2007; Kühne and Prodan, 2018), but included the iterations for finding an appropriate chemical potential that produces a charge-neutral system.
Measurements

The main measurements presented here are as follows:

1. Wall clock time of the NOLSM method \( T_{\text{NOLSM}} \):

   The wall clock time \( T_{\text{NOLSM},i} \) of the NOLSM method on node \( i \) is measured for each iteration of the chemical potential. Each iteration includes all transfers between host and GPU. The overall wall clock time \( T_{\text{NOLSM}} \) is defined as the maximum over all node wall clock times.

2. FLOPs in the NOLSM method \( \text{FLOPs}_{\text{NOLSM},i} \):

   The per-node floating-point operations \( \text{FLOPs}_{\text{NOLSM}} \) in the FP16/FP32-mixed-precision matrix iterations in the NOLSM method are estimated as \( 2n^3 \) for a gemm-operation \( C = \alpha A \cdot B + \beta C \) with \( A, B, C \in \mathbb{R}^{n \times n} \) for each
The construction of the matrix elements of the submatrices and other operations scaling like $O(n^2)$ are neglected in the count.

### 3. Node performance of NOLSM method $P_{\text{NOLSM},i}$

The node performances of the NOLSM method are defined as $P_{\text{NOLSM},i} = \frac{\text{FLOPs}_{\text{NOLSM},i}}{T_{\text{NOLSM},i}}$ for each node $i$.

### 4. Performance of the NOLSM method $P_{\text{NOLSM}}$

The performance of the NOLSM method is defined as the sum of the node performances.

### HPC system and environment

The benchmark runs presented here have been performed on the Perlmutter system at the National Energy Research Scientific Computing Center (NERSC). The Perlmutter system consists of 1536 GPU nodes with one AMD EPYC 7763 64-core CPU with 256 GB DDR4 memory and four NVIDIA A100 GPUs with 40 GB of HBM2 memory each. The peak performance of the tensor cores in one NVIDIA A100 GPU is 312 TFLOP/s in FP16 with FP32-based accumulate (NVIDIA Corporation, 2021). The system uses HPE Cray Slingshot as node interconnect.

The software environment used in this work consisted of GCC 11.2.0, Cray-MPICH 8.1.10, CUDA NVCC 11.5.119, and CUBLAS 11.5. One MPI rank per node and 64 CPU-threads per rank were used as well as four CUDA streams per GPU. Each stream was controlled by a single CPU-thread.

### Performance measurements and results

**Performance of the NOLSM method for the spike protein**

We have performed calculations for three different grid sizes of spike proteins: $6 \times 5$ (51 mio. atoms), $6 \times 6$ (61 mio. atoms), and $7 \times 7$ (83 mio. atoms). All three
Example calculations have been performed with 1100 nodes of the Perlmutter system, that is, 4400 NVIDIA A100 GPUs.

The wall clock time of the NOLSM method $T_{\text{NOLSM}}$ is shown in Figure 4. The distribution of the performances of individual nodes is shown in Figure 5 for $7 \times 7$ spike proteins (83 mio. atoms) in relation to the peak performance of the GPUs. The performances of the nodes with 4 NVIDIA A100 GPUs mainly fall in the range between 1 PFLOP/s and 1.07 PFLOP/s with an average of 1.03 PFLOP/s. This represents about 80% of the peak performance of $1.248 \text{ PFLOP/s} = 4 \cdot 0.312 \text{ PFLOP/s per node.}$

A floating-point throughput of 1.106 to 1.127 EFLOP/s with 4400 NVIDIA A100 GPUs is achieved representing about 80% of the theoretical peak performance of the tensor cores.

**Conclusion**

To the best of our knowledge, the achieved $\sim 1.1 \text{ EFLOP/s}$ in FP16/FP32 floating-point arithmetic positions electronic structure–based molecular dynamics calculations with the non-orthogonal local submatrix method in CP2K (T. Kühne et al., 2020) among one of the first algorithms in computational natural science that has broken the exaflop barrier within a scientific application (Kurth et al., 2018; Joubert et al., 2018; Liu et al., 2021). The massively parallel nature of the method allows for an efficient use of many thousand GPUs. The method can not only be applied to electronic structure–based molecular dynamics, but also in other situations where a matrix function needs to be evaluated for a large sparse matrix or problems that can be transformed to such an operation.

**Acknowledgements**

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.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article:
Additionally, we would like to thank for funding of this project by computing time provided by the Paderborn Center for Parallel Computing (PC2). This work is partially funded by Paderborn University’s research award for “GreenIT,” as well as the Federal Ministry of Education and Research (BMBF) and the state of North Rhine-Westphalia as part of the NHR Program. T.D.K. received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (Grant Agreement No. 716142).

**REFERENCES**

Andermatt S, Cha J, Schüffmann F, et al. (2016) Combining linear-scaling DFT with subsystem DFT in Born–Oppenheimer and Ehrenfest molecular dynamics Simulations: from molecules to a virus in solution. *Journal of Chemical Theory and Computation* 12(7): 3214–3227.

Arita M, Arapan S, Bowler DR, et al. (2014) Large-scale DFT simulations with a linear-scaling DFT code CONQUEST on K-computer. *Journal of Advanced Simulation in Science and Engineering* 1(1): 87–97.

Bowler DR and Miyazaki T (2010) Calculations for millions of atoms with density functional theory: linear scaling shows its potential. *Journal of Physics: Condensed Matter* 22(7): 074207.

Car R and Parrinello M (1985) Unified approach for molecular dynamics and density-functional theory. *Physical Review Letters* 55: 2471–2474.

Casalino L, Dommer AC, Gaieb Z, et al. (2021) “AI-driven multiscale simulations illuminate mechanisms of SARS-CoV-2 spike dynamics”. In: *The International Journal of High Performance Computing Applications* 35.5, pp. 432–451. DOI: 10.1177/109434202110086452.

Das S, et al. (2019) Fast, scalable and accurate finite-element based ab initio calculations using mixed precision computing: 46 PFLOPS simulation of a metallic dislocation system In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–11.

FattebertJean-Luc, et al. (2016) Modeling dilute solutions using first-principles molecular dynamics: computing more than a million atoms with over a million cores In: *SC16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, pp. 12–22.

Galli G and Parrinello M (1992) “Large scale electronic structure calculations”. *Physical Review Letters* 69(24): 3547–3550.

Goedecker S (1999) “Linear scaling electronic structure methods”. In: *Reviews of Modern Physics* 71(4): 1085–1123.

Grimme S, Ehrlich S and Goerigk L (2011) Effect of the damping function in dispersion corrected density functional theory. *Journal of Computational Chemistry* 32(7): 1456–1465.

Gygi F, et al. (2006) Large-scale electronic structure calculations of high-Z metals on the BlueGene/L platform In: *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, p. 45.

Hasegawa Y, Iwata JI, Tsuji M, et al. (2014) Performance evaluation of ultra-large-scale first-principles electronic structure calculation code on the K computer. *The International Journal of High Performance Computing Applications* 28(3): 335–355.

Hutter J and Curioni A (2005) “Dual-level parallelism for ab initio molecular dynamics: reaching teraflop performance with the CPMD code”. *Parallel Computing* 31(1): 1–17.

Jain N, et al. (2016) Openatom: scalable ab-initio molecular dynamics with diverse capabilities In: *International Conference on High Performance Computing*. Springer, pp. 139–158.

Joubert W, et al. (2018) Attacking the Opioid Epidemic: Determining the Epistatic and Pleiotropic Genetic Architectures for Chronic Pain and Opioid Addiction In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis. SC ’18*. Dallas, TX: IEEE Press. DOI: 10.1109/SC.2018.00060.

Kühne TD, Heske J and Prodan E (2020) Disordered crystals from first principles II: transport coefficients. *Annals of Physics* 421: 168290.

Kühne TD and Prodan E (2018) Disordered crystals from first principles I: quantifying the configuration space. *Annals of Physics* 391: 120–149.

Kühne TD (2014) Second generation Car–Parrinello molecular dynamics. *WIREs Comput 4*: 391–406.

Kühne TD, Iannuzzi M, Del Ben M, et al. (2020) “CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations.” *The Journal of Chemical Physics* 152(19): 194103.

Kühne TD, Krack M, Mohamed FR, et al. (2007) “Efficient and Accurate Car-Parrinello-like Approach to Born-Oppenheimer Molecular Dynamics”. In: *Physical Review Letters* 98.6, p. 066401.

Kurth T, et al. (2018) Exascale deep learning for climate analytics In: *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 649–660. DOI: 10.1109/SC.2018.00054.

Lass M, Mohr S, et al. (2018) A massively parallel algorithm for the approximate calculation of inverse p-th roots of large sparse matrices In: *Proc. Platform for Advanced Scientific Computing (PASC) Conference*. ACM.

Lass M, Robert Schade, et al. (2020) A submatrix-based method for approximate matrix function evaluation in the quantum
chemistry code CP2K. Proc. International Conference for High Performance Computing, Networking, Storage and Analysis (SC). IEEE Computer Society, pp. 1127–1140.

Liu YA, et al. (2021) Closing the "quantum supremacy" gap: achieving real-time simulation of a random quantum circuit using a new sunway supercomputer In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. SC ’21. St. Louis, Missouri: Association for Computing Machinery. DOI: 10.1145/3458817.3487399.

McWeeny R (1960) Some recent advances in density matrix theory. Reviews of Modern Physics 32(2): 335–369.

Niklasson AMN, Mniszewski SM, Negre CFA, et al. (2016) “Graph-based linear scaling electronic structure theory”. In: The Journal of Chemical Physics 144(23): 234101. DOI: 10.1063/1.4952650.

Nomura K, et al. (2014) Metascaleable quantum molecular dynamics simulations of hydrogen-on-demand In: SC ’14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, pp. 661–673.

Corporation NVIDIA (2021) URL: https://www.nvidia.com/content/dam/enzp/Solutions/Data-Center/nvidia-ampere-architecture-whitepaper.pdf.

Payne MC, Teter MP, Allan DC, et al. (1992) Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients. Reviews of Modern Physics 64(4): 1045–1097.

Rengaraj V, Lass M, Plessl C, et al. (2020) “Accurate Sampling with Noisy Forces from Approximate Computing.” Computation 8: 39.

Richters D and Kühne TD (2014) “Self-consistent field theory based molecular dynamics with linear system-size scaling.” The Journal of Chemical Physics 140(13): 134109.

Richters D, Lass M, Walther A, et al. (2019) A general algorithm to calculate the inverse principal p-th root of symmetric positive definite matrices. Communications in Computational Physics 25(2): 564–585.

Schade R, Kenter T, Elgabarty H, et al. (2022) “Towards electronic structure-based ab-initio molecular dynamics simulations with hundreds of millions of atoms.” Parallel Computing 111: 102920. DOI: 10.1016/j.parco.2022.102920.

Schulz G (1933) Iterative berechnung der reziproken matrix. ZAMM - Zeitschrift für Angewandte Mathematik und Mechanik 13(1): 57–59.

VandeVondele J, Borštnik U and Hutter J (2012) Linear scaling self-consistent field calculations with millions of atoms in the condensed phase. Journal of Chemical Theory and Computation 8(10): 3565–3573.

Wilkinson KA, Hine NDM and Skylaris CK (2014) Hybrid MPI-OpenMP parallelism in the ONETEP linear-scaling electronic structure code: application to the delamination of cellulose nanofibris. Journal of Chemical Theory and Computation 10(11): 4782–4794.

Wrapp D, Wang N, Corbett KS, et al. (2020) “Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation.” Science 367(6483), pp. 1260–1263. DOI: 10.1126/science.abb2507.

Yang W (1991) “Direct calculation of electron density in density-functional theory.” Physical Review Letters 66(11), pp. 1438–1441.

Zhao Z, et al. (2009) The linearly scaling 3D fragment method for large scale electronic structure calculations. Journal of Physics: Conference Series 180(1): 012079.

Author biographies

Robert Schade received his Ph.D. from Georg-August University Göttingen in 2019 for the development of new methods for the ab-initio simulation of materials with strong local electronic correlations. In his Ph.D. work, Robert Schade developed a novel method for the description of strong local electronic correlations in solids based on reduced density matrix functional theory (rDMFT). At PC2, Dr Schade is engaged in HPC-consulting for scientific users as well as general training of users to empower them to use the available computing resources in an efficient way. His main research focus is on the design and implementation of novel algorithms for enabling quantum chemistry on quantum computers as well as new highly parallel algorithms for quantum chemistry and their acceleration with FPGAs und GPUs.

Tobias Kenter received his PhD from Paderborn University in 2016 on the topic of productivity for FPGAs through overlays, compilation approaches, and tight coupling between FPGAs and CPUs. Since then, he focused on the acceleration of scientific applications on FPGAs using high-level synthesis–based development flows from Xilinx (now part of AMD) and Intel. As scientific advisor for FPGA acceleration at the Paderborn Center for Parallel Computing (PC2), he strives to bring more applications to this exciting technology and is involved in planning and operation of production systems with FPGAs.

Hossam Elgabarty graduated with a PhD in Chemistry in 2013 from the group of Daniel Sebastiani at the FU-Berlin. Later, he did two postdoctoral fellowships: At the University of Mainz with Prof. T. D. Kühne and at the University of Halle-Wittenberg with Prof. Daniel Sebastiani. In 2017, he moved to the University of Paderborn where he is currently a group leader at the Chair of Theoretical Chemistry. His group focuses on ab-initio MD simulations and theoretical condensed phase spectroscopy.

Michael Lass received his PhD from Paderborn University in 2022. In his dissertation, he dealt with the acceleration of a quantum chemistry code using accelerator devices, in particular GPUs and FPGAs, by exploiting algorithmic approximations and low-precision arithmetic. Since then, he
works as a scientific advisor at the Paderborn Center for Parallel Computing (PC2), assisting users in the acceleration of their HPC codes and further developing the FPGA infrastructure.

**Thomas D. Kühne** studied computer science (B. Sc. ETH in 2003) and computational science and engineering (Dipl.-Rech. Wiss. ETH in 2005) with a focus in theoretical chemistry and computational astrophysics at ETH Zürich. Thereafter, he worked under the mentorship of Prof. Michele Parrinello in Lugano, where he obtained his Doctor of Science degree in theoretical physics in 2008, also from ETH Zürich. After postdoctoral research on multiscale simulation methods within the theoretical condensed matter group at Harvard University, he joined the University of Mainz as an assistant professor in theoretical chemistry in 2010. In 2014, he then moved to Paderborn University, as a tenured associate professor in Theoretical Interface Chemistry, where he was promoted to full professor in 2018 as the Chair of Theoretical Chemistry. Since May 2023, Prof. Kühne is the Founding Director of the Center for Advanced Systems Understanding (CASUS) at the Helmholtz-Zentrum Dresden-Rossendorf and Full Professor of Computational Systems Science at the TU Dresden. His research interests include the development of novel computational methods for ab-initio molecular dynamics and electronic structure theory, as well as the application of these techniques to study a large variety of different systems within Chemistry, Biophysics, and Materials Science.

**Christian Plessl** is professor for High-Performance Computing at the Department of Computer Science of Paderborn University. He is also the managing director of the Paderborn Center for Parallel Computing and member of the board of directors of the German National High-Performance Computing Alliance (NHR). He earned a PhD degree (2006) and MSc degree (2001), both from ETH Zurich. His research interests include architecture and tools for high-performance parallel and reconfigurable computing, hardware-software codesign, and adaptive computing systems.