doi:10.2533/chimia.2025.220 Chimia 79 (2025) 220–223

Multiscale Molecular Dynamics Simulations with the MiMiC Framework

Andrea Levy[§a], Andrej Antalík[a], Jógvan Magnus Haugaard Olsen[b], and Ursula Rothlisberger[a*]

[§]SCS-dsm-firmenich Award for the best poster presentation in Computational Chemistry

Abstract: Multiscale simulations are essential techniques in computational chemistry, providing insights into complex phenomena across extended temporal and spatial scales. With a particular interest in the dynamics of such processes, we developed MiMiC, a framework for efficient multiscale molecular dynamics simulations suited for high-performance computing. One of its key characteristics is a flexible design where external specialized programs handle individual subsystems. This article reviews the core features and some recent advancements in MiMiC, particularly the integration of OpenMM and CP2K.

Keywords: Molecular dynamics · Multiscale modeling · Multiscale simulations · QM/MM

Andrea Levy graduated in physics at the University of Milano Bicocca. Since 2020, he has been a PhD student under the supervision of Prof. Ursula Rothlisberger. His research focuses on the development of multiscale methods and their applications to biological systems.

Andrej Antalík graduated in physics at the Charles University in Prague, where he also obtained a PhD under the supervision of Prof. Jiří Pittner. In 2021, he joined the group of Prof. Ursula Rothlisberger as a postdoctoral researcher. His research focuses on the development of electronic structure and multiscale methods, and he is one of the main developers of the MiMiC framework.

Jógvan Magnus Haugaard Olsen earned a degree in chemistry at the University of Copenhagen and a PhD at the University of Southern Denmark. He pursued postdoctoral research with Prof. Ursula Rothlisberger (EPFL) and continued as an independent researcher at the University of Southern Denmark and later at UiT Arctic University of Norway. He became an assistant professor at the Technical University of Denmark (DTU) in 2021, where he has been an associate professor since 2023. His research focuses on the development and application of multiscale methods to study photodynamic processes. He is one of the authors and main developers of the MiMiC framework.

Ursula Rothlisberger earned a degree in physical chemistry and a PhD at the University of Bern. She pursued postdoctoral research with Prof. Michael L. Klein (University of Pennsylvania) and later with Prof. Michele Parrinello (Max Planck Institute for Solid State Physics). She was appointed assistant professor at ETH Zurich and moved to EPFL in 2002 as an associate professor, becoming a full professor in 2009. Her research focuses on the development of electronic structure and multiscale methods, as well as their application in biochemistry and material science.

1. Introduction

Chemical and physical phenomena often span extended temporal and spatial scales. Typical examples in biology include the rearrangement of electrons during biochemical processes, proton transfer reactions, and photoexcitation. In such cases, a quantum mechanical (QM) description is required, but it is not applicable to the entire system considering the large size of biomolecules combined with the steep computational scaling of QM methods with system size. Multiscale methods address this problem by treating different regions of the systems at different resolutions and levels of theory.[1,2] In particular, the more accurate but computationally demanding methods are applied only where they are strictly needed.

Typically, multiscale methods are implemented by extending the functionalities of existing programs. However, this approach requires substantial implementation and maintenance efforts and potentially duplicates the functionalities already optimized in other programs. A different strategy consists of implementing well-defined interfaces among standalone programs. However, this approach can lack computational efficiency due to repeated start-ups and shutdowns of the programs and slow data exchange between them, as they do not necessarily share memory.

We introduced the MiMiC framework for multiscale molecular dynamics (MD) simulations,[3–5] which loosely couples external specialized programs in a highly flexible manner while preserving high computational efficiency. It employs a client-server approach with a multiple-program multiple-data (MPMD) model, where external programs run concurrently and compute different

*Correspondence: Prof. U. Rothlisberger, E-mail: ursula.roethlisberger@epfl.ch
[a]LCBC, SB ISIC EPFL, CH-1015 Lausanne, Switzerland; [b]TCBC, DTU Chemistry, 2800 Kongens Lyngby, Denmark

subsystem contributions while MiMiC handles the inter-program communication and computes the subsystem interactions.

2. Software Design

The MiMiC framework comprises two main components: the main MiMiC library and MCL, *i.e.* the MiMiC communication library (Fig. 1). The main library manages the communication between the client programs and computes the interactions between subsystems. It is linked to the simulation driver, which propagates the system by integrating the equations of motion. This interface between the library and the simulation driver is realized through a simple application programming interface (API), which consists of a few procedures that are incorporated into the MD simulation loop. MCL facilitates data transfer *via* network-based communication (currently utilizing message passing interface (MPI)), with minimal impact on the parallelization of the interfaced external programs.

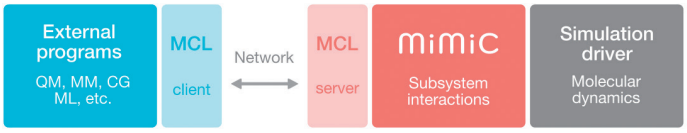

Fig. 1. Illustration of the strategy used by the MiMiC framework (reprinted from ref. [4], with the permission of AIP Publishing).

Once interfaced, a typical workflow on the client program side is as follows: after a normal initialization, the program enters the MiMiC loop, *i.e.* a conditional loop where at every iteration the program receives a request through MCL, and executes the corresponding action, which can be sending/receiving data, starting a calculation, or finalizing the execution at the end of the simulation. For more details on the design of the MiMiC framework, we refer you to ref. [4].

3. QM/MM MD with MiMiC

One of the most widespread multiscale methods consists of combining QM resolution, applied to a small subsystem, with molecular mechanics (MM) using classical force fields applied to the remainder.[6,7] This method, known as QM/MM, can be implemented using an additive scheme, where the time-independent Hamiltonian of the system,

$$H_{\mathrm{tot}} = H_{\mathrm{QM}} + H_{\mathrm{MM}} + H_{\mathrm{QM/MM}}, \quad (1)$$

describes the energies of the QM and MM subsystems, and their interactions. In QM/MM MD, the Born–Oppenheimer (BO) approximation is typically assumed for the QM subsystem, in combination with the approximation that both QM nuclei and MM particles are propagated according to the classical equations of motion. The QM/MM interaction term includes bonded terms (if bonds are cut at the QM–MM boundary), as well as non-bonded interactions corresponding to van der Waals and electrostatic interactions. In particular, in electrostatic embedding QM/MM, the electron density of the QM subsystem is polarized by fixed atom-centred charges (or multipoles) of the MM subsystem, and since the charges of the MM subsystem are fixed, there is no back-polarization from the QM subsystem.

MiMiC features a generalized version of the Hamiltonian electrostatic coupling scheme by Laio *et al.*,[8] where the electrostatic QM/MM interactions are split into short-range (SR) and long-range (LR) contributions. SR interactions are computed exactly, while the LR ones are approximated using a multipole expansion of the QM potential to an arbitrary order.[3] This allows users to employ a relatively small SR region, resulting in a less expensive calculation, yet without a significant loss in accuracy. For more theoretical details, we refer you to refs. [3, 4].

4. Recent Features Introduced in MiMiC

This section highlights recent features of the MiMiC framework, focusing on newly added client programs alongside the already available MM and QM ones, *i.e.* GROMACS and CPMD, respectively. In particular, we recently introduced OpenMM and CP2K into the MiMiC ecosystem. Thanks to the design of MiMiC, adding a new client program ensures compatibility with any other interfaced code for supported multiscale methods.

4.1 *OpenMM*

The OpenMM MD package shares a similar design philosophy with MiMiC: it focuses on extensibility and efficiency, especially on graphics processing unit (GPU) architectures, and it is also open-source.

We recently introduced a MiMiC interface to OpenMM and demonstrated its efficiency for QM/MM MD simulations.[9] This benefits both projects: it adds a highly flexible client program to the list of clients interfaced with the MiMiC framework and enables future development to be accessible by OpenMM with minimal changes needed. In particular, the OpenMM–MiMiC interface integrates the core OpenMM C++ library into a standalone C++ program that implements the MiMiC loop, with the same logic introduced in section 2.

For further details, we refer to our recent work.[9] Here, we highlight one key result from that study, where BO QM/MM MD simulations with MiMiC were compared using GROMACS or OpenMM as alternative MM client programs. The simulated system (Fig. 2a) is the zinc-binding site of a GB1 protein mutant solvated in water (~25,000 atoms in total), where 40 atoms are modelled at the QM level using density functional theory (DFT). A simulation with the GROMACS–MiMiC interface served as a reference, and we compared the difference in the total energy per particle from two simulations starting from the identical initial configuration, using GROMACS or OpenMM as the MM client program. In the case of GROMACS, for this comparison, we did not use the 'reprod' keyword, which removes all sources of non-reproducibility. In both cases, CPMD was the QM client program. As shown in Fig. 2b, no deviation in the total energy per particle is apparent for the first ~0.7ps with both MM client programs, and the magnitude of subsequent fluctuations is comparable.

4.2 *GROMACS Updated Release*

GROMACS was the first MM client program interfaced with MiMiC in its initial release,[3] with the highest supported version v2021.6. We recently updated the interface to support the latest GROMACS release (v2025.0), and this updated interface will become available in one of next GROMACS releases. Although these updates introduce no conceptual changes to the original interface, they are essential for maintaining compatibility with newer GROMACS versions and accessing their latest features.

To validate this update, we performed the same validation tests presented in section 4.1. on the GB1 mutant system solvated in water, now comparing GROMACS v2021.6 and v2025.0 as MM client programs. As in the previous case, Fig. 2c shows no substantial difference in the total energy per particle fluctuations in both cases.

4.3 *CP2K*

Recently, we also extended the MiMiC ecosystem with CP2K,[10] which offers a wide range of electronic-structure methods and excellent performance on high-performance computing systems, including progressively wider support for GPU architec-

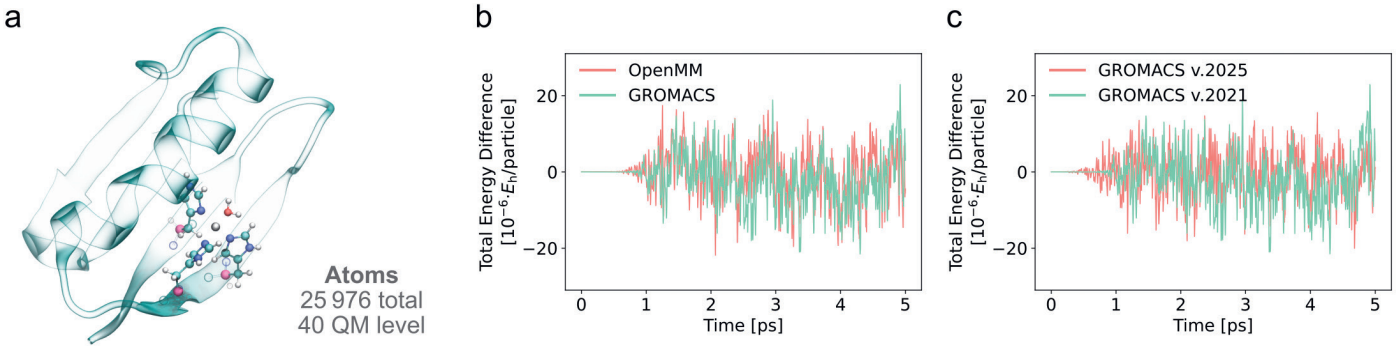

Fig. 2. Validation for the new OpenMM–MiMiC interface (data from ref. [9]). a) System simulated: GB1 protein in water (not represented for clarity). MM subsystem transparent, QM atoms in ball-and-stick representation, and boundary QM–MM atoms highlighted in pink. b) Difference in the total energy per particle between a reference QM/MM MD simulation using GROMACS as MM client program (without the 'reprod' keyword activated), and a simulation from the same configuration using OpenMM (red) or GROMACS (green) as an MM client program. In both cases, CPMD was the QM client program. c) Same as b), using GROMACS v.2025.0 (red), and v.2021.6 (green) as an MM client program.

tures. Owing to its modular design, we were able to introduce the interface to MiMiC with minimal interference to the rest of the code and in fact, it alters only a single-core CP2K routine. The rest of the interface is contained in independent modules.

To unlock the possibility of introducing new QM clients, MiMiC underwent major refactoring to enable its intended flexibility and facilitate the integration of new QM client programs. More details are reported in our recent work.[10] Here, we highlight two key results from that study. For validation, we compared BO QM/MM MD simulations using CPMD or CP2K as the QM client program, with the original and the refactored MiMiC versions, respectively. In both cases, the MM client program was GROMACS, with a system consisting of a green fluorescent protein (GFP) solvated in water (~28,000 atoms in total, with 22 atoms treated at the QM level with DFT, Fig. 3a). We monitored the fluctuations of the total energy per particle. The plot in Fig. 3b demonstrates that the simulations are stable in both cases, with the energy fluctuating around the mean value.

Combining these two latest developments,[9,10] we were able to take advantage of the modular design of MiMiC and replace one simulation component, namely GROMACS, with another one, OpenMM, with no additional effort. Using the same GFP system introduced in this section, we demonstrated the stability of QM/MM MD simulations, now using CP2K to treat the QM subsystem, and OpenMM for the MM one (Fig. 3c).

5. Conclusions and Outlook

In the presented work, we have briefly reviewed the main features of the MiMiC framework for flexible and efficient multiscale MD simulations, focusing on recent developments, namely the interfaces with OpenMM and GROMACS (v2025.0) as MM client programs and CP2K as a QM client.

The list of new client programs will continue to grow, with development efforts for Quantum ESPRESSO and Tinker-HP interfaces underway. We also plan to extend MCL, as well as the MiMiC main library, with a Python API, which will expand the pool of accessible programs even further. Additionally, we intend to go beyond electrostatic embedding QM/MM, by introducing polarizable embedding,[11] as well as fragment-based methods and multi-layered QM/QM(/MM) models and machine learning accelerated dynamics[12] to treat larger QM systems.

Associated Content

The source code of the MiMiC framework is free and open-source and is hosted on GitLab.[5] The code of the new interfaces highlighted in this paper will soon be available on the same GitLab page after code review.

Acknowledgements

The authors gratefully acknowledge all the members of the MiMiC consortium. Funding was provided by the Swiss National Science Foundation (Grant. Nos. 200020-185092 and 200020-21944), and computing resources by the Swiss National Supercomputing Centre CSCS. JMHO gratefully acknowledges financial support from VILLUM FONDEN (Grant No. VIL29478).

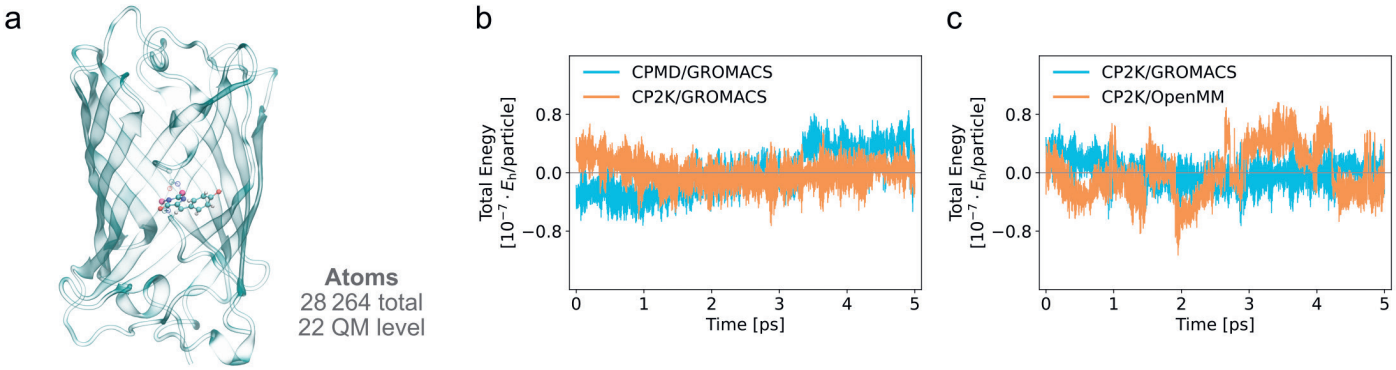

Fig. 3. Validation for the new CP2K–MiMiC interface (data from ref. [10]). a) System simulated: GFP in water (not represented for clarity). MM subsystem transparent, QM atoms in ball-and-stick representation, and boundary QM–MM atoms highlighted in pink. b) Fluctuations of the total energy per particle relative to the average energy for QM/MM MD simulations performed with the released MiMiC version with CPMD as a QM client program (blue) and the refactored one with CP2K (orange). c) Same as b), using CP2K as a QM client program, together with GROMACS (blue) or OpenMM (orange) as an MM client program.

Author Contributions

A. L: Software, Visualization, Writing – original draft, Writing – review & editing. A. A: Conceptualization, Software, Visualization, Supervision, Writing – review & editing. J. M. H. O: Conceptualization, Software, Funding acquisition, Project administration, Supervision, Writing – review & editing. U. R: Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing. text.

Received: February 27, 2025

[1] A. Warshel, M. Levitt, *J. Mol. Biol*. **1976**, *103*, 227, https://doi.org/10.1016/0022-2836(76)90311-9.
[2] M. J. Field, P. A. Bash, M. Karplus, *J. Comput. Chem*. **1990**, *11*, 700, https://doi.org/10.1002/jcc.540110605.
[3] J. M. H. Olsen, V. Bolnykh, S. Meloni, E. Ippoliti, M. P. Bircher, P. Carloni, U. Rothlisberger, *J. Chem. Theory Comput*. **2019**, *15*, 3810, https://doi.org/10.1021/acs.jctc.9b00093.
[4] A. Antalík, A. Levy, S. Kvedaravićiūté, S. K. Johnson, D. Carrasco-Busturia, B. Raghavan, F. Mouvet, A. Acocella, S. Das, V. Gavini, D. Mandelli, E. Ippoliti, S. Meloni, P. Carloni, U, Rothlisberger, J. M. H. Olsen, *J. Chem. Phys*. **2024**, *161*, 022501, https://doi.org/10.1063/5.0211053.
[5] The source code of the MiMiC framework is free and open-source and is hosted on GitLab: https://gitlab.com/mimic-project.
[6] E. Brunk, U. Rothlisberger, *Chem. Rev*. **2015**, *115*, 6217, https://doi.org/10.1021/cr500628b.
[7] F. Lipparini, B. Mennucci, *Chem. Phys. Rev*. **2021**, *2*, 041303, https://doi.org/10.1063/5.0064075.
[8] A. Laio, J. VandeVondele, U. Rothlisberger, *J. Chem. Phys*. **2002**, *116*, 6941, https://doi.org/10.1063/1.1462041.
[9] A. Levy, A. Antalík, J. M. H. Olsen, U. Rothlisberger, *arXiv* **2025**, arXiv:2502.06539.
[10] A. Antalík, A. Levy, S. K. Johnson, J. M. H. Olsen, U. Rothlisberger, *arXiv* **2025**, arXiv:2502.16253
[11] S. Kvedaravićiūté, A. Antalík, O. Adjoua, T. Plé, L. Lagardère, U. Rothlisberger, J. P. Piquemal, J. M. H. Olsen, *ChemRxiv* **2025**, ChemRxiv:10.26434/chemrxiv-2024-6sp7h.
[12] F. Mouvet., J. Villard, V. Bolnykh, U. Rothlisberger, *Acc. Chem. Res*. **2022**, *55*, 221, https://doi.org/10.1021/acs.accounts.1c00503.

License and Terms

The definitive version of this article is the electronic one that can be found at https://doi.org/10.2533/chimia.2025.220