M-Chem: a Modular Software Package for Molecular Simulation that Spans Scientific Domains

Jagna Witek[1], Joseph P. Heindel[1,2], Xingyi Guan[1,2], Itai Leven[1], Hongxia Hao[1], Pavithra Naullage[1], Allen LaCour[1,2], Selim Sami[1], M.F.S.J. Menger[4], D. Vale Cofer-Shabica[5], Eric Berquist[6], Shirin Faraji[4], Evgeny Epifanovsky[6], Teresa Head-Gordon[1,2,3]

[1]Kenneth S. Pitzer Theory Center and Department of Chemistry
[2]Chemical Sciences Division, Lawrence Berkeley National Laboratory
[3]Departments of Bioengineering and Chemical and Biomolecular Engineering
University of California, Berkeley, CA, USA
[4]Stratingh Institute for Chemistry, University of Groningen, 9747 AG Groningen, The Netherlands
[5]Department of Chemistry, University of Pennsylvania, Philadelphia, PA, 19128 USA
[6]Q-Chem, Inc., 6601 Owens Drive, Suite 105, Pleasanton, California 94588, USA
corresponding author: epif@q-chem.com, thg@berkeley.edu

Abstract

We present a new software package called M-Chem that is designed from scratch in C++ and parallelized on shared-memory multi-core architectures to facilitate efficient molecular simulations. Currently, M-Chem is a fast molecular dynamics (MD) engine that supports the evaluation of energies and forces from two-body to many-body all-atom potentials, reactive force fields, coarse-grained models, combined quantum mechanics molecular mechanics (QM/MM) models, and external force drivers from machine learning, augmented by algorithms that are focused on gains in computational simulation times. M-Chem also includes a range of standard simulation capabilities including thermostats, barostats, multi-timestepping, and periodic cells, as well as newer methods such as fast extended Lagrangians and high quality electrostatic potential generation. At present M-Chem is a developer friendly environment in which we encourage new software contributors from diverse fields to build their algorithms, models, and methods in our modular framework. The long-term objective of M-Chem is to create an interdisciplinary platform for computational methods with applications ranging from biomolecular simulations, reactive chemistry, to materials research.

1 Introduction

Computer simulations are undoubtedly established as one of the main pillars of modern research, accelerating scientific progress in the molecular sciences including biomolecular modeling, computational quantum chemistry, and materials science research. Each of these scientific domains are supported by a variety of codes that incorporate a combination of software efficiencies and new algorithms that avoid bottlenecks and/or reduce unfavorable scaling, thereby allowing for exploration of ever increasing system sizes and longer timescales. This opens up the question - why is there a need for another simulation software package?

We believe there are three primary reasons. The first is that simulation codes are siloed into scientific domains that insufficiently bridge across different simulation communities. Software packages such as Amber[1], Charmm[2], NAMD[3], and GROMACS[4] are popular in the biomolecular simulation community, whereas LAMMPS is viewed primarily as a materials research code[5], and many quantum chemistry codes[6?] are not particularly compatible with classical simulations and/or require plane wave implementations to describe electronic structure. A second reason is that software implementations for advanced force fields comprising many-body potentials or reactive force field models are insufficiently optimized, thereby limiting their wider adoption by consumers of molecular simulation software.[7] Finally it is well appreciated that large software packages benefit greatly from ground up rewrites, creating new opportunities for software efficiency, ease of development and use, exploiting new hardware paradigms, and seamless software integration and interoperability.

In that spirit we present a new software package called M-Chem that is designed to facilitate efficient molecular simulations on shared-memory multi-core architectures to address these issues. M-Chem, like all new software packages, is incomplete and yet is rapidly expanding its simulation capabilities. Hence some users may find that their research objectives may well be met with many of the features that we describe here. At present M-Chem enables users to run efficient MD simulations for molecular liquids using pairwise to many-body non-reactive force fields[8–10], ReaxFF simulations[11,12], coarse-grained electron (C-GeM)[13] and monoatomic-water (mW) simulations[14], and forces derived from machine learning models created with PyTorch by utilizing the Torch C++ library. Standard, but critical, MD algorithms include periodic boundary conditions, particle mesh Ewald[15], targeted ensembles such as NVT and NPT using velocity scaling Berendsen thermostats, as well as extended system Nose-Hoover thermostats and barostats.[16] There is an emergent analysis toolkit of simulated properties, including the ability to generate electrostatic potentials at protein surfaces using C-GeM, whose accuracy is comparable to some of the best ab initio charge partitioning methods at orders of magnitude less cost.[17] Furthermore, we introduce Q-Force as an automated approach to small molecule force field development for M-Chem utilizing ab initio quantum chemistry calculations.[18] Both standard conjugate gradient self-consistent field (CG-SCF)[19] and extended Lagrangian formulations[20] are available to solve the many-body energy and force solutions, including iEL/0-SCF[21] and SC-XLMD[22] that are iteration-free algorithms for classical polarizable and ReaxFF models, respectively. M-Chem also has multi-scale capabilities with mechanical and simple electrostatic (point charge) embedding, exploiting some of the common libraries between M-Chem and Q-Chem such that the interface is internal to M-Chem. All of these features are described in some detail below.

Finally, M-Chem is an attractive developer platform and environment. The M-Chem package is entirely written in C++, with an overall architecture that facilitates excellent transferability and code readability. M-Chem is composed of independent modules (libraries) and a developer can decide which libraries they want to include in their project, which makes the code cleaner and reduces the compilation and linking time. From the computational software engineering perspective, the M-Chem project follows software best practices including extensive unit testing and full documentation. M-Chem has been optimized for excellent on-node parallel performance on modern hardware architectures using OpenMP, even on desktop machines, and thus provides a solid basis for extending the paradigm to efficient Graphical Processing Units (GPUs) and multi-node parallelization with the Message Passing Interface (MPI) in the future. As we show, M-Chem strives for both accuracy and greater tractability, utilizing novel methodology alongside code optimization that provides faster turnaround and greater interoperability capabilities compared to other software packages. This manuscript serves the primary purpose as an open invitation to software developers who will receive M-Chem source code in exchange for software contributions that builds up capabilities that further benefit developers and users across multiple scientific domains.

2 Biomolecular, Chemical, and Materials Force Fields in M-Chem

The vast majority of all-atom MD simulation studies use classical pairwise fixed-charge electrostatics because the MD engine is so well-tuned and optimized for such force fields. M-Chem fully supports such partial charge models for liquids such as water or chain molecules such as proteins. Even so, the failures of pairwise additivity have been made clear in a number of simulation studies.[23–26] Therefore classical potential energy surfaces are evolving towards more intricate models based on the following energy (and force) contributions

$$E_{\text{int}} = E_{\text{valence}} + E_{\text{elec}} + E_{\text{Pauli}} + E_{\text{disp}} + E_{\text{pol}} \tag{1}$$

where E_{valence} corresponds to geometric terms that are designed to capture chemical connectivity but not reactivity, and the E_{elec}, E_{Pauli}, E_{disp}, and E_{pol} terms correspond to the non-bonded permanent electrostatics, Pauli repulsion, dispersion, and polarization energies, respectively. These terms are the accepted areas for accuracy improvements of advanced force fields.[9,26] Furthermore, most many-body force fields have neglected certain interactions such as charge penetration and charge transfer, and thus they rely on, but do not always demonstrate, how cancellation of errors occurs among the remaining molecular interactions accounted for such as exchange repulsion, electrostatics, and polarization. Our recent (many-body) MB-UCB force field for water[10] and ions[27] explicitly accounts for the decomposed molecular interactions commensurate with the variational absolutely-localized molecular orbitals (ALMO) analysis[28,29], including charge penetration and transfer, and makes force field design choices that reduce the computational expense while remaining accurate. We have completed a bottom to top software creation of a MD engine in C++ in M-Chem that encompasses both simple partial charge and advanced non-reactive force fields with point multipole electrostatics and many-body polarization as developed in force fields such as AMOEBA[9,30,31] and AMOEBA+[32], EFP[33], GEM[34,35], HIPPO[36], SIBFA[37], MB-Pol[38,39] and the MB-UCB force-field[10,27,40].

Another addition to the model suite in M-Chem is the reactive force-field, ReaxFF.[11,12] We have implemented the ReaxFF model based on the 2008 description of the force field[41], and it is thoroughly validated against the implementation of the same force field in LAMMPs[42]. We also verified the integrity of the implementation through extensive testing of all gradient terms on a wide variety of test systems. Originally, ReaxFF utilized the electronegativity equalization method (EEM) method[43–45] which manifests charge rearrangements by adopting atomic electronegativity and atomic hardness as fitting parameters to DFT derived Mulliken charges.[44,45] The 2008 ReaxFF model in M-Chem supports the EEM extension developed by Rappé and Goddard by replacing the standard Coulomb potential with a shielded electrostatic term, and using experimental atomic ionization potentials, electron affinities, and atomic radii as the input data for optimizing the charge rearrangements in response to nuclear displacements.[46] We have also developed a ReaxFF model that replaces EEM with the novel Coarse-Grained Electron Model (C-GeM).[13,47] The C-GeM model represents atoms in terms of a core-shell model, with nuclei being positive cores surrounded by separable electronic shells, with both described by Gaussian charge distributions. This greatly improves the accuracy of the ReaxFF force field without any significant increase in computational time. At this point in time, the ReaxFF energy and gradient terms are all complete, but the code is being optimized, so we have not extensively timed it against other implementations. Additionally, we wish to incorporate recently introduced corrections[48] which smooth slight discontinuities in ReaxFF, thereby greatly improving energy conservation in the microcanonical ensemble.

Finally, M-Chem aims to be an open platform development for coarse-grained models. To illustrate, we have implemented the monoatomic-water (mW) model develop by Molinero and co-workers, which is a single site water model that exhibits both cooperativity and directional hydrogen-bonding of water in various phases[14,49–51] based on the Stillinger Weber Potential for tetrahedral systems[52]. The mW model is found to be in good agreement with experimental data on water energetics, density, structure, and phase transitions[14], and in the limit of high cooperativity phase separation of liquid water itself was observed in the supercooled state. Recently the mW water model was combined with a CG protein model to study antifreeze proteins.[53,54] In this case the coarse-grained model is comprised of an attractive 2-body potential with a ($1/\mathrm{r}^4$) distance dependence screened with an exponential, and a repulsive 3-body potential with a cosine squared angular dependence that penalizes non-tetrahedral configurations, and a product of two scaled exponentials with $\exp(1/\mathrm{r}^2)$ distance dependence. This approach of coarse-graining to effective potentials has been shown to agree well with all-atom potentials for force matching.[55–63]

3 Performance of Non-Reactive Force Fields on Multicore Nodes

Software implementations of models such as AMOEBA and MB-UCB, when insufficiently optimized, limit their greater adoption by consumers of molecular simulation software.[7] This is because the fastest MD codes for fixed charge models such as CHARMM[64], NAMD[3] and GROMACS[4] efficiently exploit heterogeneous computing, and hence define the aspirational goals for M-Chem to improve scaling performance for advanced many-body potentials and forces. Shared-memory CPUs are an obvious target for a new software platform given widely available computing resources ranging from mid-range compute clusters, cloud computing, and even supercomputer centers. This is recognized by the NAMD[3] and GROMACS[4] teams, whose software packages are leaders in performance in the 10–200-core regime as demonstrated in a recent comparison[64].

However, both packages lack the optimization for advanced force fields that utilize multipolar electrostatics and point dipole polarization, core-shell models such as C-GeM, and overall advanced force fields often scale poorly on multicore architectures. This limitation is significant since recent hardware offerings that arrived in late 2019, the 2nd generation AMD EPYC processors, can now support up to 128 CPU cores per node. With two-way simultaneous multithreading, with each core doubling the current peak rate to 33.6 GFLOPS, the tripling of the core count of previous generation CPU configurations is a great opportunity for modernized parallel software.

Parallel efficiency of codes is most commonly measured in terms of weak scaling. But as recently stated by Abraham and co-workers for CPU optimization of European-based GROMACS: "When studying a protein system with 30,000 atoms, it is not relevant that a virus comprising 10 million atoms would scale better. Therefore, weak scaling performance is typically not of primary concern."[4] We therefore consider not just weak but strong scaling performance as well for the two most time-consuming parts of evaluation of an MD step in the many-body models: the fixed-radius nearest neighbor search and force field evaluation whose kernel is the non-bonded forces. In this section we compare parallel scaling benchmarks for the MD engine of M-Chem utilizing OpenMP on a single node and compared to the high-performance TINKER-HP software package developed by the European group of Piquemal and co-workers[65], and written in Fortran with MPI parallelization.

OpenMP parallelization of fixed-radius nearest neighbor evaluations. Due to the computational complexity of the nonbonded terms in force fields, neighbor lists are essential to attain them with linear cost scaling by only evaluating a subset of atom pairs located within a defined cutoff radius. In particular, the real-space Ewald and van der Waals pair list need to be evaluated with a reasonable frequency to account for atoms' movement during updates through integration of the equations of motion. When the pair list is evaluated for the whole system, the computational complexity becomes significant and therefore well

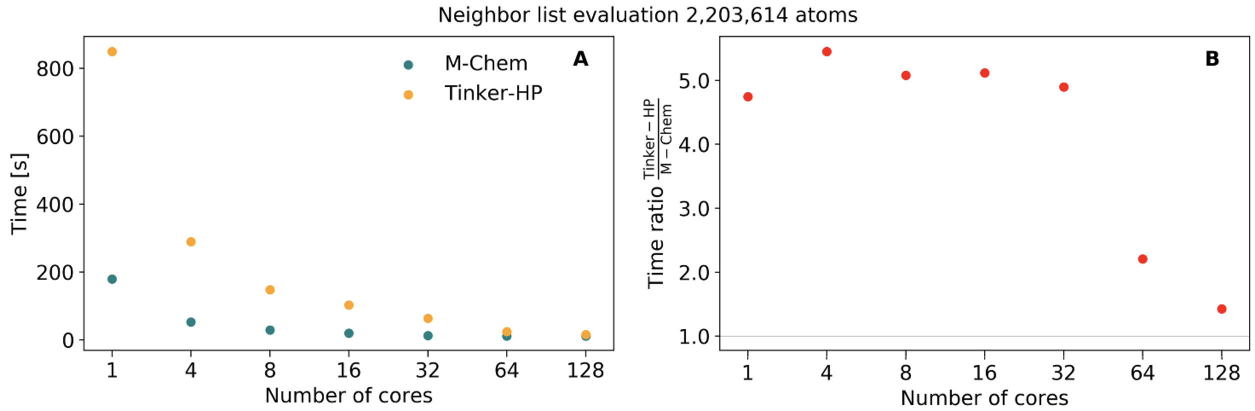

Figure 1: Strong scaling performance of neighbor list evaluation comparing M-Chem and Tinker-HP. A: Total execution time of a single fixed-radius neighbor list evaluation per number of cores on a large water system. B: Tinker-HP/M-Chem execution time ratio.

scaling algorithms become critical. Additionally, due to the nature of the AMOEBA force field, positions of atomic centers for certain atoms evaluated with the van der Waals interaction are scaled with respect to the atomic centers, which effectively means that two pair lists need to be evaluated.

Designing an efficient parallel fixed-radius nearest neighbor (fRNN) algorithm with $O(n)$ memory cost scaling and $O(nlogn)$ computational scaling is difficult, however we are able to lean on recent developments for similar algorithms for image and volumetric data processing that have not been used in molecular simulation software previously.[66] The execution time of a single neighbor list evaluation for a system of over 2 million atoms is shown in Figure 1, comparing M-Chem's octree fRNN algorithm with the linked cells fRNN method available in Tinker-HP. The execution time demonstrates that M-Chem's implementation is approximately 4-5 times faster than Tinker-HP's for serial execution on a single core, and outperforming TINKER-HP albeit with increasingly smaller gaps as core count increases. Even so, M-Chem yields a factor of 2 and 1.4 improvement over Tinker-HP for 64 and 128 cores, respectively (numerical values are provided in Table S1).

OpenMP parallelization of force field evaluations per MD step. Next we consider the parallelization of the MD step, equivalent to the force evaluations based on Eq. (1). Provided that the topology of the molecules are known, parallel algorithms that run in linear time are straightforward to implement for the valence terms. Instead the rate-limiting step of Eq. (1) are the non-bonded interactions, especially the long-range electrostatic terms that require computing multipole and induced dipole moments on a grid and performing fast Fourier transforms (FFTs) to complete the Ewald model.[15] While FFTs are also relatively straightforward with existing high-performance software libraries[67,68], the electrostatic multipole moments and inducible dipoles present several difficulties that requires a carefully designed algorithm. Our approach is similar to other codes by using spatial localization to form batches of multipoles, and the grid assigned to processors followed by parallel accumulation on the final grid.[4,69–71]

The M-Chem strong scaling parallel performance of force-field evaluation using Eq. (1) is presented in Table 1 in which we compare the timings of 20 iterations of energy and gradient evaluations against Tinker-HP for methanol (978,000 atoms) and water (2,203,614 atoms). The liquid methanol system invokes two sets of induced dipoles (treated with p- and d-scaling to separately evaluate intra- from intermolecular polarization[30]), and gives us a sense of what performance looks like for the proteinaceous component of a protein-solvent system. We have found it worthwhile to optimize the water force field separately, which only invokes p-scaling (i.e. no intramolecular polarization is operative), as it is almost always the relevant solvent in simulation studies. M-Chem utilizes the fact that systems where all molecules belong to a single polarization group can be evaluated in a significantly simplified and more efficient manner.

Table 1: Strong scaling execution time [s] of 20 iterations of the AMOEBA force field energy and gradient evaluation across different systems and algorithms. The cutoffs for the van der Waals and real-space Ewald terms were set to 12.0 and 7.0 Å respectively. The induced dipole convergence was set to 10^{-5} D.

978,000 atoms methanol box				$\mathrm{N}_{\mathrm{CPUs}}$			
	1	4	8	16	32	64	128
M-Chem iEL/0-SCF	2330.0	643.0	346.0	200.0	124.0	89.0	94.0
M-Chem CG-SCF	3054.0	834.0	447.0	259.0	150.0	110.0	108.0
Tinker-HP DC/JI DIIS	3478.0	1010.0	545.0	288.0	180.0	123.0	120.0
M-Chem Direct	2268.0	620.0	324.0	186.0	105.0	74.0	66.0
Tinker-HP Direct	2713.0	794.0	422.0	221.0	136.0	92.0	90.0
61,000 atoms methanol box				$\mathrm{N}_{\mathrm{CPUs}}$			
	1	4	8	16	32	64	128
M-Chem iEL/0-SCF	141.0	41.0	22.0	13.0	8.0	7.0	8.0
M-Chem CG-SCF	181.0	52.0	28.0	17.0	10.0	8.0	11.0
Tinker-HP DC/JI DIIS	183.0	49.0	30.0	15.0	10.0	7.0	-
M-Chem Direct	141.0	38.0	20.0	11.0	7.0	5.0	6.0
Tinker-HP Direct	144.0	40.0	22.0	12.0	7.0	7.0	-
2,203,614 atoms water box				$\mathrm{N}_{\mathrm{CPUs}}$			
	1	4	8	16	32	64	128
M-Chem iEL/0-SCF	5605.0	1529.0	791.0	438.0	250.0	167.0	162.0
M-Chem CG-SCF	6753.0	1855.0	972.0	545.0	301.0	213.0	215.0
Tinker-HP DC/JI DIIS	11008.0	3610.0	1692.0	713.0	522.0	369.0	388.0
M-Chem Direct	5095.0	1401.0	723.0	398.0	234.0	152.0	153.0
Tinker-HP Direct	8760.0	2899.0	1488.0	698.0	462.0	289.0	256.0
96,000 atoms water box				$\mathrm{N}_{\mathrm{CPUs}}$			
	1	4	8	16	32	64	128
M-Chem iEL/0-SCF	219.0	61.0	32.0	17.0	10.0	7.0	7.0
M-Chem CG-SCF	254.0	70.0	37.0	21.0	13.0	10.0	11.0
Tinker-HP DC/JI DIIS	282.0	77.0	43.0	22.5	14.0	11.0	-
M-Chem Direct	206.0	57.0	29.0	16.0	9.0	7.0	7.0
Tinker-HP Direct	225.0	61.0	34.0	18.0	11.0	7.0	-

It is important to note in this comparison that the SCF solution to polarization is evaluated differently in the two codes. M-Chem is using conjugate gradients SCF (CG-SCF)[19] with the electric field from fixed multipoles as an initial guess, while Tinker-HP is evaluated with a divide-and-conquer Jacobi iterations with direct inversion in the iterative subspace extrapolation (DC/JI-DIIS)[72] which is more efficient than the CG-SCF method. Even so, and regardless of the number of cores used, M-Chem performs better. But to consider a more direct comparison between the codes, we consider the evaluation of the AMOEBA force field without mutual polarization, but just direct polarization as per the iAMOEBA model[73], such that the rate-limiting step is the 2-body multipolar permanent electrostatics. Table 1 shows that the multipolar electrostatics and corresponding evaluation of electric fields is ∼30% faster for the methanol system and ∼90-100% faster for the large water box.

Weak scaling performance for force field evaluations. The comparison of weak scaling between M-Chem and Tinker-HP is presented in Figure 2 and Table S2, where the size of the system per CPU is approximately 15,000 atoms for methanol and 17,000 atoms for water. In the top panel of Figure 2 we first consider the evaluation of AMOEBA with direct polarization (direct SCF in Table S2), such that the rate-limiting step is again the 2-body multipolar permanent electrostatics. It is evident that the parallel efficiency for M-Chem is superior to TINKER-HP with performance gains increasing with increasing core count, with the best performance gains of 60% (methanol) and 90% (water) using 128 cores. For the timings with polarization in the bottom panel of Figure 2, M-Chem outperforms Tinker-HP for all core counts, with the biggest gap of 33 seconds on 128 cores for the ∼1M atom methanol box, and saving 193 seconds and approximately 90% of the time used by TINKER-HP for the 2 million atom water system (numerical values are given in Table S2).

Often the overall performance of an MD code is measured in terms of number of ns of simulation executed per day. The standard system used for such benchmarking is the prototypic dihydrofolate reductase (DHFR) solvated in explicit water from the joint AMBER/CHARMM benchmark. Since a full protein front-end features have not yet been fully implemented in M-Chem, we performed the benchmark on a methanol box (23,556 atoms), and the best performance results using OpenMP were obtained on 64 cores, yielding the same performance of the MPI version of Tinker-HP using 64 cores of ∼1.2 ns/day, and a factor of 6X compared to the TINKER reference code. Even so, the next step for M-Chem is to pursue hybrid OpenMP/MPI to improve on node performance and to advance to cross-node MPI parallelization that will benefit large systems.

One of the goals of M-Chem is to also support a diverse set of models and algorithms to expand the timescales and lengthscales of MD simulation and to provide necessary statistical mechanics capabilities. The M-Chem MD engine is thus already imbued with standard velocity scaling including Nose Hoover thermostats and barostats[16], periodic boundary conditions using cubic cells, and conjugate gradients for minimization and appropriate pre-conditioners for conjugate gradients used in the self-consistent field step for solving sets of linear equations (such as for polarization).

The choice of the algorithm to treat polarization or charge equilibration is of paramount importance to overall efficiency of force field evaluation for non-reactive and reactive force fields. Albaugh and co-workers have introduced a number of extended Lagrangian approaches that reduces SCF cycles by half (iEL/SCF)[20] or eliminates the self-consistent field step altogether (iEL/0-SCF)[21] for non-reactive force fields such as AMOEBA or MB-UCB. Most recently we combined iEL/0-SCF with a stochastic integration scheme that allows for a longer time step using a multi-time stepping algorithm, SIN(R), developed by Tuckerman and co-workers.[74,75] Depending on system and desired accuracy, the iEL/0-SCF and SIN(R) combination yields lower bound computational speed-ups of 6-8 relative to a standard Verlet integration step using a standard SCF solver.[75] This benefits not only polarization, but the simple induction-like model for charge transfer of the MB-UCB model. The results in Figure 2 show that, for the cases of both methanol and water, M-Chem's default extended Lagrangian integrator decreases execution time further (black dots) than the SCF solver. Furthermore, the EEM or charge equilibration method (CEM or qEq) electrostatics of the ReaxFF model in M-Chem can be solved using either a pre-conditioned conjugate gradient method, and we are in the process of implementing the iEL/SCF[76] and SC-XLMD[22] extended Lagrangian approaches that should increase simulation timescales dramatically for ReaxFF simulations.

4 Interoperability in M-Chem: PyTorch and Q-Chem

Machine learning interfaces. The rise of machine-learned models in biology, chemistry, and materials science[77,78] is tackling everything from materials property prediction to learning the energies and forces of complex system with good to excellent ab initio quality. This exciting and burgeoning area has resulted in many different neural network-based force field models[79–82], and hence we have defined a generic interface

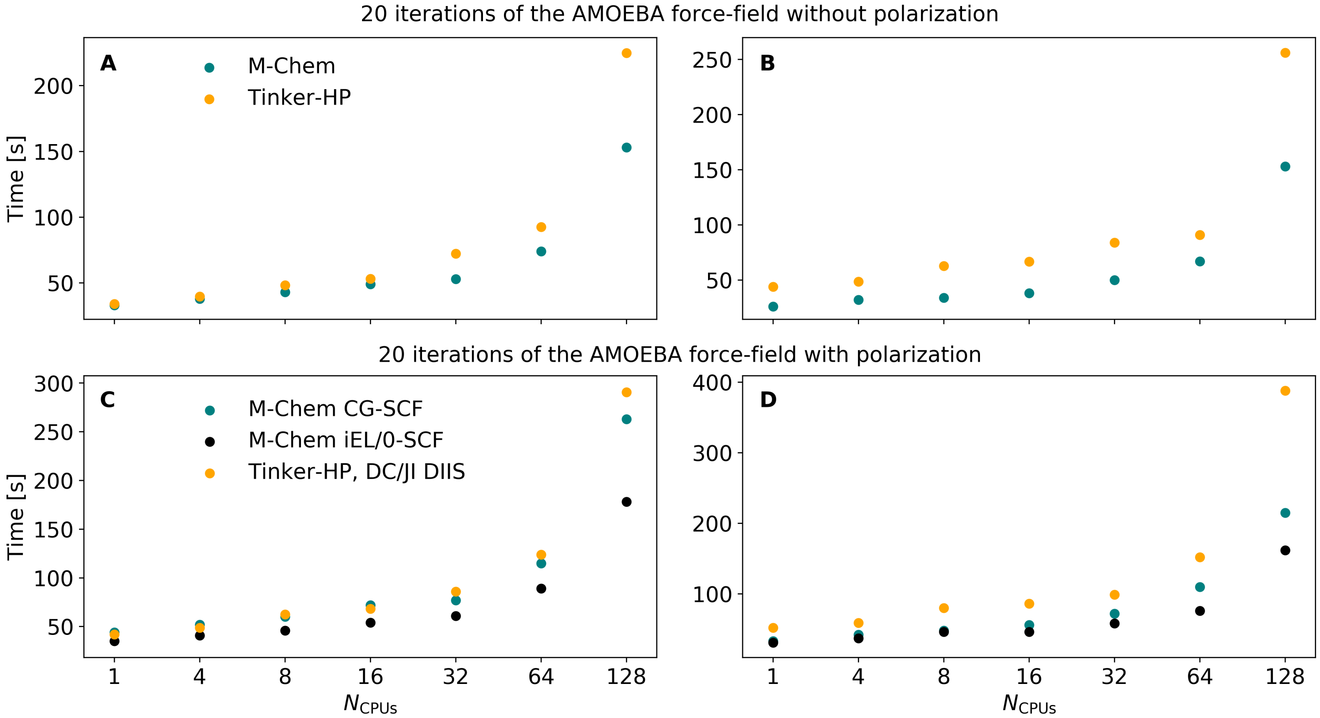

Figure 2: Weak scaling performance of AMOEBA force field components comparing M-Chem and Tinker-HP. Total execution time per number of cores for 20 iterations of force-field evaluations for a (A) methanol box (approx. 15,000 atoms per CPU) and (B) water box (approx. 17,000 atoms per CPU), both with only direct polarization[73], and largely probing timings for multipolar electrostatics. Total execution time per number of cores for 20 iterations of the complete AMOEBA force-field (including polarization) for a (C) methanol box (approx. 15,000 atoms per CPU) and (D) water box (approx. 17,000 atoms per CPU). The cutoffs for the van der Waals and real-space Ewald terms were set to 12.0 and 7.0 Å respectively. The induced dipole convergence was set to 10^{-5} D. This figure also compares different SCF algorithms: CG-SCF and iEL/0-SCF for M-Chem and DC/JI DIIS for Tinker-HP.

for ML FF models built with the PyTorch framework.[83] This is made possible by the ability of PyTorch to compile its own models into a byte-code representation which can then be loaded and evaluated by libtorch, which is a full-featured front-end API for PyTorch written in C++. This means that all we need to do is link against the libtorch library and we can evaluate ML models on CPUs, GPUs, or even mutliple GPUs with minimal programming effort. Our aim is to allow users to write code within the PyTorch ecosystem and use their models in M-Chem without having to write any C++ code.

We have done this successfully with NewtonNet[82], a message passing equivariant NN model, demonstrating the viability of this approach. In order to ensure the interface has been implemented properly, we compared timings of evaluating the force field 2000 times, including calculation of both energies and gradients using the PyTorch version and the same model exported to C++ and run in MChem. To do this, we used a version of NewtonNet trained on a hydrogen-combustion dataset[84] and evaluated the model on many configurations of $H_2O_2(OH^\bullet)$. We find that evaluating energies and gradients with the model on the CPU takes an average of 6.4 ms using libtorch as integrated into M-Chem as opposed to the 7.7 ms using PyTorch. Similarly, on the GeForce GTX 1650 mobile GPU, which is not intended for fast double precision arithmetic as required here, evaluation with libtorch/M-Chem takes 8.7 ms vs 10.2 ms using PyTorch. Overall, by simply evaluating a neural network through the libtorch C++ API, we can evaluate the model ∼20% faster using the PyTorch Python API. While these relative timings will depend on the structure of the model being used and the system being evaluated, we emphasize that it is reasonable to expect moving your model to MChem is likely to slightly speed up a model rather than slow it down.

Furthermore, this interface opens many interesting possibilities. Indeed, one could implement an entire force field of any kind using PyTorch (in python), serialize the model, load it into M-Chem, and use this force field in MD simulations without having to write any C++. The refinement of a neural network force field could also be accomplished on the fly, by backpropagating a NN model and updating the weights on the C++ side. One can hence envision advanced QM/ML simulations in which the QM method can be dynamically swapped in or out when a NN has low confidence in its prediction, allowing the ML model to be updated based on the results of the QM calculation without stopping the simulation. This complex type of simulation is still a work in progress but is the type of ML interface capabilities that MChem aims to make routine, thereby creating an easy interface to create reference or prototype code, and would be an excellent opportunity for a first contribution. In the future, we may explore similar interfaces with TensorFlow, another popular ML library.

QM/MM mechanical and electrostatic embedding. Biologically or technologically relevant systems are typically very large and thus computationally too demanding to be treated at any level of quantum mechanical method that accounts for the explicit role of the electrons and have high predictive power. At the same time, commonly used molecular mechanical force fields are not sufficiently flexible to model processes that involve electronic rearrangement, e.g. bond formation/breaking and electronic excitations. Fortunately, many chemical processes of interest occur in spatially localized regions and only a small region must be treated at the quantum mechanical level. Therefore, one can overcome the limitations of both methods by constructing an approximate reduced-dimensionality Hamiltonian. In such a hybrid scheme, the system is divided into a small region of interest, which undergoes electronic rearrangement and is treated quantum mechanically, and the remaining part, the environment, which is treated classically; the so-called quantum mechanics/molecular mechanics framework. Utilizing such model Hamiltonians not only makes the calculations computationally feasible, but also leads to a clear distinction between essential and non-essential parts of the system. As a practical matter, for a successful QM/MM simulation, a molecular dynamics software (here M-Chem) must be combined with an electronic structure software (here Q-Chem) that requires coordination between many different pieces of code to enable various flavour of QM/MM schemes. There are three QM/MM schemes, which describe the coupling between the QM and MM regions to different extends; i) mechanical embedding, ii) electrostatic embedding, and iii) polarizable embedding. Currently our QM/MM interface supports mechanical embedding. Work in the direction to more elaborate embedding such as multi-polar electrostatic embedding and polarizable embedding are in progress in our team.

5 Simulation Utilities in M-Chem: ESPs, FF Parameterization, and File Formats

It is desirable for a single biomolecular simulation software package to support diverse models, methods, and analysis tools beyond the MD engine itself. An important utility relevant for all kinds of chemical applications is the ability to accurately create an electrostatic potential (ESP) on a molecular surface. ESPs are used to help understand sites of biomolecular association and docking, drug binding, interfacial surface potentials, and visualization of noncovalent interactions such as hydrogen bonding, halogen bonding, and pi-pi interactions. Another example is that the point of transferable force field models is that protocols must be developed that allow a systematic search for a consistent set of force field parameters for new molecule chemistries. These software libraries associated with the M-Chem code are described here.

Electrostatic potentials using C-GeM. Although C-GeM is already part of the ReaxFF force field implementation in M-Chem, it can also be run standalone for the fast generation of electrostatic potentials. We have tested C-GeM against 600 molecules each with 10 separate configurations that are being used for benchmarking empirical approaches for ESP rendering of drug molecules containing 30-65 atoms evaluated at the B3LYP/6-311G** level of theory. C-Gem performed quite accurately across the data set with mean absolute error (MAE) of 2.8 kcal/mol, compared to 7.2 kcal/mol MAE for the default EEM method used in the docking software AutoDock, and slightly better than new machine learning approaches generated by the Forli group. Compared to the original C-GeM model[17] which works in gas phase, in the M-Chem implementation we added dielectric settings to allow for electrostatics predictions in solvent. Figure 3 shows the C-GeM electrostatic potential surface for crambin in solvent(a) and in gas phase(b).

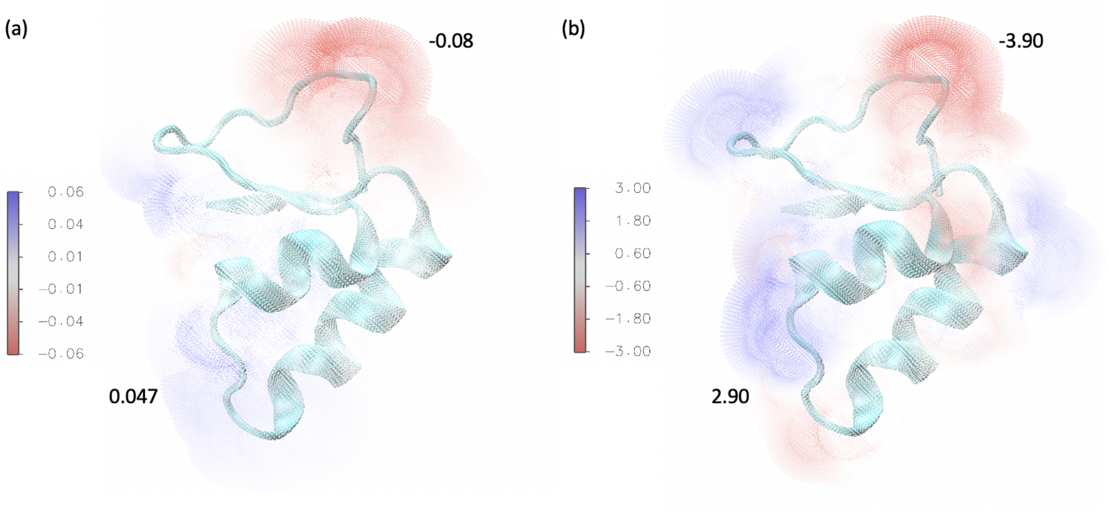

Figure 3: Electrostatic potential surface predicted with C-GeM on crambin(1CRN) (a) with dielectric constant for solution $\epsilon_s = 78.54$ and (b) in gas phase $\epsilon_s = 1$

It is our hope that, given the interface with Q-Chem, we can support alternative approaches such as the rendering of fitted charges (ESPC) in which the partial charges on atoms are derived from a partitioning scheme of an expensive electronic wave functions such as Mulliken[85], Hirshfeld-I[86], DDEC/c3[87], or natural population analysis[88]. ESPC models are the main methods used in all the major biomolecular force fields including AMBER[1] and CHARMM[89], and thus should also be supported here in M-Chem. Electrostatic potential surfaces rendered from Poisson Boltzmann models are also possible, and the Head-Gordon lab has developed PB-AM and PB-SAM in APBS[90], but will be brought in-house in future versions of M-Chem.

QM-based force fields with Q-Force. Fully QM-based force fields provide an alternative to atom-types-based force fields with molecule-specific parameters that result in a good match between QM and MM potential energy surfaces. The Q-Force toolkit[18], which derives QM-based force fields through an automated pipeline, is an open-source software that has been interfaced with M-Chem and Q-Chem. Q-Force automatically derives standard MD models for any (novel) molecule based on QM Hessians, the electrostatic potential, torsional scans and atomic bond orders used to parametrize these potentials. It has been shown that such an approach can drastically improve the match between QM and MD potential energy surfaces and consequently increase the predictive power of standard MD simulations without additional computational cost after the initial parameterization. Additionally, the implemented molecular fragmentation procedure allows parameterizing large molecules with significantly reduced computational cost, which allowed Q-Force to be easily applied to a variety of technologically relevant applications. Extending Q-Force to advanced force fields is currently an active research line in the Head-Gordon lab and will be developed in M-Chem.

File Formats. For saving trajectories, we use a storage mechanism based on the Hierarchical Data Format version 5 (HDF5) standard. It is similar in spirit to the MDTraj HDF5 layout, in particular with storing the topology alongside time-ordered information, such as coordinates and optional velocities. However, due to the data requirements of non-traditional force fields such as ReaxFF and machine-learned potentials, it is not a direct reimplementation of the MDTraj format. We instead define the data layout in an extensible JSON Schema-like format called qarchive for which compile-time invariants on the layout are enforced, with bindings to other languages forthcoming.

6 Discussion and Conclusions

Typically, and over time, software components and algorithms become tightly coupled into a monolithic code that is difficult to port to a new platform or do not offer ease of entry for novel development work. It is now understood that much better portability can be attained at a very modest performance trade-off, with attention to modularity. Furthermore, present-day computers require careful algorithm design and performance tuning to harness the power of multiple levels of parallelism: data-parallel instructions at the lowest level, multitasking at the shared-memory multi-core level, and across-node distributed-memory parallel code execution.

In summary, we have presented a current snapshot of the capabilities of the M-Chem software package, a modular and feature diverse molecular simulation code written from the ground up in C++. M-Chem has already optimized serial and on-node performance on multicore CPUs, but the resulting software design is extensible to cross-node and GPU parallelization which defines the next phase of code optimization. The M-Chem infrastructure has also led to software hooks and APIs that permit integration of diverse models and methods such as a range of force fields, ML, and QM/MM capabilities. Our hope is that the ease with which we can change the many-body electrostatic model associated with any of the above force fields, or to interface or embed alternative ML and physics-based models, demonstrates our goal of making the physics modular where possible without sacrificing accuracy or speed.

Of course M-Chem as a user-friendly code is far from complete in many ways including front-end user interfaces to set up interfacial systems, large macromolecules, and extremely large all-atom simulations (∼500 million atoms) or back-end analysis tools that address the needs of diverse communities. M-Chem is missing critical algorithms such as free energy calculations, Poisson Boltzmann solvers, Monte-Carlo methods, other coarse-grained models, or advanced sampling methods. Even so, enough code capability is present and useable such that M-Chem is a software platform ready for bringing in a completely new set of developers of academic researchers and setting the stage for better informed partnerships with academic users and industrial consumers of M-Chem.

7 Acknowledgements

As Prof. Peter Gill is one of the founders of Q-Chem, which was also built from scratch over 25 years ago, we are happy to honor him with this debut of the M-Chem software package (of which we hope Peter becomes a developer!). We also thank the National Science Foundation under grant CHE-1955643. M-Chem software development was also supported under a SBIR grant from the National Institutes of Health 2R44GM128480-02. We thank Farnaz Heider-Zadeh for the dihedral angle and C-GeM reference codes.

CONFLICT OF INTEREST

E.E. is a part-owner of Q-Chem, Inc.

DATA AVAILABILITY

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

CODE AVAILABILITY

The source code is available to all developers.

References

[1] Romelia Salomon-Ferrer, David A. Case, and Ross C. Walker. An overview of the amber biomolecular simulation package. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2013. doi: 10.1002/wcms.1121.

[2] B. R. Brooks, C. L. Brooks III, A. D. Mackerell Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus. Charmm: The biomolecular simulation program. *Journal of Computational Chemistry*, 30(10):1545–1614, 2009. doi: 10.1002/jcc.21287. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.21287`.

[3] James C. Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D. Skeel, Laxmikant Kalvo, and Klaus Schulten. Scalable molecular dynamics with namd. *Journal of Computational Chemistry*, 26(16):1781–1802, 2005. doi: 10.1002/jcc.20289. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.20289`.

[4] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C. Smith, Berk Hess, and Erik Lindahl. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1-2:19 – 25, 2015. ISSN 2352-7110. doi: https://doi.org/10.1016/j.softx.2015.06.001. URL `http://www.sciencedirect.com/science/article/pii/S2352711015000059`.

[5] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.*, 117(1):1 – 19, 1995. URL `http://lammps.sandia.gov`.

[6] Evgeny Epifanovsky, Andrew T. B. Gilbert, Xintian Feng, Joonho Lee, Yuezhi Mao, Narbe Mardirossian, Pavel Pokhilko, Alec F. White, Marc P. Coons, Adrian L. Dempwolff, Zhengting Gan, Diptarka Hait, Paul R. Horn, Leif D. Jacobson, Ilya Kaliman, Jörg Kussmann, Adrian W. Lange, Ka Un Lao, Daniel S. Levine, Jie Liu, Simon C. McKenzie, Adrian F. Morrison, Kaushik D. Nanda, Felix Plasser, Dirk R. Rehn, Marta L. Vidal, Zhi-Qiang You, Ying Zhu, Bushra Alam, Benjamin J. Albrecht, Abdulrahman Aldossary, Ethan Alguire, Josefine H. Andersen, Vishikh Athavale, Dennis Barton, Khadiza Begam, Andrew Behn, Nicole Bellonzi, Yves A. Bernard, Eric J. Berquist, Hugh G. A. Burton, Abel Carreras, Kevin Carter-Fenk, Romit Chakraborty, Alan D. Chien, Kristina D. Closser, Vale Cofer-Shabica, Saswata Dasgupta, Marc de Wergifosse, Jia Deng, Michael Diedenhofen, Hainam Do, Sebastian Ehlert, Po-Tung Fang, Shervin Fatehi, Qingguo Feng, Triet Friedhoff, James Gayvert, Qinghui Ge, Gergely Gidofalvi, Matthew Goldey, Joe Gomes, Cristina E. González-Espinoza, Sahil Gulania, Anastasia O. Gunina, Magnus W. D. Hanson-Heine, Phillip H. P. Harbach, Andreas Hauser, Michael F. Herbst, Mario Hernández Vera, Manuel Hodecker, Zachary C. Holden, Shannon Houck, Xunkun Huang,

Kerwin Hui, Bang C. Huynh, Maxim Ivanov, Ádám Jász, Hyunjun Ji, Hanjie Jiang, Benjamin Kaduk, Sven Kähler, Kirill Khistyaev, Jaehoon Kim, Gergely Kis, Phil Klunzinger, Zsuzsanna Koczor-Benda, Joong Hoon Koh, Dimitri Kosenkov, Laura Koulias, Tim Kowalczyk, Caroline M. Krauter, Karl Kue, Alexander Kunitsa, Thomas Kus, István Ladjánszki, Arie Landau, Keith V. Lawler, Daniel Lefrancois, Susi Lehtola, Run R. Li, Yi-Pei Li, Jiashu Liang, Marcus Liebenthal, Hung-Hsuan Lin, You-Sheng Lin, Fenglai Liu, Kuan-Yu Liu, Matthias Loipersberger, Arne Luenser, Aaditya Manjanath, Prashant Manohar, Erum Mansoor, Sam F. Manzer, Shan-Ping Mao, Aleksandr V. Marenich, Thomas Markovich, Stephen Mason, Simon A. Maurer, Peter F. McLaughlin, Maximilian F. S. J. Menger, Jan-Michael Mewes, Stefanie A. Mewes, Pierpaolo Morgante, J. Wayne Mullinax, Katherine J. Oosterbaan, Garrette Paran, Alexander C. Paul, Suranjan K. Paul, Fabijan Pavošević, Zheng Pei, Stefan Prager, Emil I. Proynov, Ádám Rák, Eloy Ramos-Cordoba, Bhaskar Rana, Alan E. Rask, Adam Rettig, Ryan M. Richard, Fazle Rob, Elliot Rossomme, Tarek Scheele, Maximilian Scheurer, Matthias Schneider, Nickolai Sergueev, Shaama M. Sharada, Wojciech Skomorowski, David W. Small, Christopher J. Stein, Yu-Chuan Su, Eric J. Sundstrom, Zhen Tao, Jonathan Thirman, Gábor J. Tornai, Takashi Tsuchimochi, Norm M. Tubman, Srimukh Prasad Veccham, Oleg Vydrov, Jan Wenzel, Jon Witte, Atsushi Yamada, Kun Yao, Sina Yeganeh, Shane R. Yost, Alexander Zech, Igor Ying Zhang, Xing Zhang, Yu Zhang, Dmitry Zuev, Alán Aspuru-Guzik, Alexis T. Bell, Nicholas A. Besley, Ksenia B. Bravaya, Bernard R. Brooks, David Casanova, Jeng-Da Chai, Sonia Coriani, Christopher J. Cramer, György Cserey, A. Eugene DePrince, Robert A. DiStasio, Andreas Dreuw, Barry D. Dunietz, Thomas R. Furlani, William A. Goddard, Sharon Hammes-Schiffer, Teresa Head-Gordon, Warren J. Hehre, Chao-Ping Hsu, Thomas-C. Jagau, Yousung Jung, Andreas Klamt, Jing Kong, Daniel S. Lambrecht, WanZhen Liang, Nicholas J. Mayhall, C. William McCurdy, Jeffrey B. Neaton, Christian Ochsenfeld, John A. Parkhill, Roberto Peverati, Vitaly A. Rassolov, Yihan Shao, Lyudmila V. Slipchenko, Tim Stauch, Ryan P. Steele, Joseph E. Subotnik, Alex J. W. Thom, Alexandre Tkatchenko, Donald G. Truhlar, Troy Van Voorhis, Tomasz A. Wesolowski, K. Birgitta Whaley, H. Lee Woodcock, Paul M. Zimmerman, Shirin Faraji, Peter M. W. Gill, Martin Head-Gordon, John M. Herbert, and Anna I. Krylov. Software for the frontiers of quantum chemistry: An overview of developments in the q-chem 5 package. *The Journal of Chemical Physics*, 155(8):084801, 2021. doi: 10.1063/5.0055522. URL `https://doi.org/10.1063/5.0055522`.

[7] A. Krylov, T. L. Windus, T. Barnes, E. Marin-Rimoldi, J. A. Nash, B. Pritchard, D. G. A. Smith, D. Altarawy, P. Saxe, C. Clementi, T. D. Crawford, R. J. Harrison, S. Jha, V. S. Pande, and T. Head-Gordon. Perspective: Computational chemistry software and its advancement as illustrated through three grand challenge cases for molecular science. *J Chem Phys*, 149(18): 180901, 2018. ISSN 1089-7690 (Electronic) 0021-9606 (Linking). doi: 10.1063/1.5052551. URL `https://www.ncbi.nlm.nih.gov/pubmed/30441927`.

[8] P. Y. Ren and J. W. Ponder. Consistent treatment of inter- and intramolecular polarization in molecular mechanics calculations. *J. Comp. Chem.*, 23(16):1497–1506, 2002.

[9] Jay W. Ponder, Chuanjie Wu, Pengyu Ren, Vijay S. Pande, John D. Chodera, Michael J. Schnieders, Imran Haque, David L. Mobley, Daniel S. Lambrecht, Jr Robert A. DiStasio, Martin Head-Gordon, Gary N. I. Clark, Margaret E. Johnson, and Teresa Head-Gordon. Current status of the amoeba polarizable force field. *J. Phys. Chem. B*, 114(8):2549–2564, 2010.

[10] Akshaya K. Das, Lars Urban, Itai Leven, Matthias Loipersberger, Abdulrahman Aldossary, Martin Head-Gordon, and Teresa Head-Gordon. Development of an advanced force field for water using variational energy decomposition analysis. *Journal of Chemical Theory and Computation*, 15(9):5001–5013, 2019. ISSN 1549-9618. doi: 10.1021/acs.jctc.9b00478. URL `https://doi.org/10.1021/acs.jctc.9b00478`.

[11] A. C. T. van Duin, S. Dasgupta, F. Lorant, and W. A. Goddard. Reaxff: A reactive force field for hydrocarbons. *J. Phys. Chem. A*, 105:9396–9409, 2001.

[12] Itai Leven, Hongxia Hao, Songchen Tan, Xingyi Guan, Katheryn A. Penrod, Dooman Akbarian, Benjamin Evangelisti, Md Jamil Hossain, Md Mahbubul Islam, Jason P. Koski, Stan Moore, Hasan Metin Aktulga, Adri C. T. van Duin, and Teresa Head-Gordon. Recent advances for improving the accuracy, transferability, and efficiency of reactive force fields. *Journal of Chemical Theory and Computation*, 17(6):3237–3251, 2021. doi: 10.1021/acs.jctc.1c00118. URL `https://doi.org/10.1021/acs.jctc.1c00118`. PMID: 33970642.

[13] Itai Leven and Teresa Head-Gordon. C-gem: Coarse-grained electron model for predicting the electrostatic potential in molecules. *The journal of physical chemistry letters*, 10(21):6820–6826, 2019.

[14] V. Molinero and E. B. Moore. Water modeled as an intermediate element between carbon and silicon. *J Phys Chem B*, 113(13):4008–16, 2009. ISSN 1520-6106 (Print) 1520-5207 (Linking). doi: 10.1021/jp805227c. URL `https://www.ncbi.nlm.nih.gov/pubmed/18956896`.

[15] C. Sagui, L. G. Pedersen, and T. A. Darden. Towards an accurate representation of electrostatics in classical force fields: Efficient implementation of multipolar interactions in biomolecular simulations. *J. Chem. Phys.*, 120:73–87, 2004.

[16] M. Tuckerman, B. J. Berne, and G. J. Martyna. Reversible multiple time scale molecular dynamics. *J. Chem. Phys.*, 97:1990–2001, 1992.

[17] X. Guan, I. Leven, F. Heidar-Zadeh, and T. Head-Gordon. Protein c-gem: A coarse-grained electron model for fast and accurate protein electrostatics prediction. *J. Chem. Inf. Model.*, 61:4357 – 4369, 2021.

[18] Selim Sami, Maximilian F.S.J Menger, Shirin Faraji, Ria Broer, and Remco W. A. Havenith. Q-force: Quantum mechanically augmented molecular force fields. *J. Chem. Theory Comput.*, 17(8):4946–4960, 2021. doi: 10.1021/acs.jctc.1c00195.

[19] W. Wang and R. D. Skeel. Fast evaluation of polarizable forces. *J. Chem. Phys.*, 123:164107–164107, 2005.

[20] Alex Albaugh, Omar Demerdash, and Teresa Head-Gordon. An efficient and stable hybrid extended lagrangian/self-consistent field scheme for solving classical mutual induction. *The Journal of Chemical Physics*, 143(17):174104, 2015. doi: http://dx.doi.org/10.1063/1.4933375. URL `http://scitation.aip.org/content/aip/journal/jcp/143/17/10.1063/1.4933375`.

[21] A. Albaugh and T. Head-Gordon. A new method for treating drude polarization in classical molecular simulation. *J Chem Theory Comput*, 13(11):5207–5216, 2017. ISSN 1549-9626 (Electronic) 1549-9618 (Linking). doi: 10.1021/acs.jctc.7b00838. URL `https://www.ncbi.nlm.nih.gov/pubmed/28965397`.

[22] Songchen Tan, Itai Leven, Dong An, Lin Lin, and Teresa Head-Gordon. Stochastic Constrained Extended System Dynamics for Solving Charge Equilibration Models. *J. Chem. Theo. Comp.*, 16(10):5991–5998, oct 2020. ISSN 1549-9618. doi: 10.1021/acs.jctc.0c00514. URL `https://dx.doi.org/10.1021/acs.jctc.0c00514`.

[23] Richard C. Remsing, Marcel D. Baer, Gregory K. Schenter, Christopher J. Mundy, and John D. Weeks. The role of broken symmetry in solvation of a spherical cavity in classical and quantum water models. *J. Phys. Chem. Lett.*, 5(16):2767–2774, 2014. URL `<Go to ISI>://WOS:000340807100001`.

[24] Mario Vazdar, Eva Pluharova, Phil E. Mason, Robert Vacha, and Pavel Jungwirth. Ions at hydrophobic aqueous interfaces: Molecular dynamics with effective polarization. *J. Phys. Chem. Lett.*, 3(15):2087–2091, 2012. URL `<Go to ISI>://WOS:000309691500029`.

[25] P. S. Nerenberg and T. Head-Gordon. New developments in force fields for biomolecular simulations. *Curr Opin Struct Biol*, 49:129–138, 2018. ISSN 1879-033X (Electronic) 0959-440X (Linking). doi: 10.1016/j.sbi.2018.02.002. URL `https://www.ncbi.nlm.nih.gov/pubmed/29477047`.

[26] O. Demerdash, E. H. Yap, and T. Head-Gordon. Advanced potential energy surfaces for condensed phase simulation. *Annu. Rev. Phys. Chem.*, 65:149–174, 2014.

[27] Akshaya Kumar Das, Meili Liu, and Teresa Head-Gordon. Development of a many-body force field for aqueous alkali metal and halogen ions: An energy decomposition analysis guided approach. *Journal of Chemical Theory and Computation*, 18(2):953–967, 2022. doi: 10.1021/acs.jctc.1c00537. URL `https://doi.org/10.1021/acs.jctc.1c00537`. PMID: 35072483.

[28] Rustam Z. Khaliullin, Erika A. Cobar, Rohini C. Lochan, Alexis T. Bell, and Martin Head-Gordon. Unravelling the origin of intermolecular interactions using absolutely localized molecular orbitals. *The Journal of Physical Chemistry A*, 111(36):8753–8765, 2007. doi: 10.1021/jp073685z. URL `https://doi.org/10.1021/jp073685z`. PMID: 17655284.

[29] Paul R. Horn, Yuezhi Mao, and Martin Head-Gordon. Probing non-covalent interactions with a second generation energy decomposition analysis using absolutely localized molecular orbitals. *Phys. Chem. Chem. Phys.*, 18:23067–23079, 2016. doi: 10.1039/C6CP03784D. URL `http://dx.doi.org/10.1039/C6CP03784D`.

[30] P. Y. Ren and J. W. Ponder. Polarizable atomic multipole water model for molecular mechanics simulation. *J. Phys. Chem. B*, 107(24):5933–5947, 2003.

[31] Jay W. Ponder and David A. Case. *Force Fields for Protein Simulations*, volume 66, pages 27–85. Academic Press, 2003.

[32] Chengwen Liu, Jean-Philip Piquemal, and Pengyu Ren. Amoeba+ classical potential for modeling molecular interactions. *Journal of Chemical Theory and Computation*, 15(7):4122–4139, 2019. ISSN 1549-9618. doi: 10.1021/acs.jctc.9b00261. URL `https://doi.org/10.1021/acs.jctc.9b00261`.

[33] Lyudmila V. Slipchenko and Mark S. Gordon. Damping functions in the effective fragment potential method. *Mol. Phys.*, 107(8-12, SI):999–1016, 2009.

[34] G. A. Cisneros, D. Elking, J. P. Piquemal, and T. A. Darden. Numerical fitting of molecular properties to hermite gaussians. *Journal of Physical Chemistry A*, 111(47):12049–12056, 2007. ISSN 1089-5639. doi: 10.1021/jp074817r. URL `<Go to ISI>://WOS:000251140700015`.

[35] G. A. Cisneros, K. T. Wikfeldt, L. Ojamae, J. B. Lu, Y. Xu, H. Torabifard, A. P. Bartok, G. Csanyi, V. Molinero, and F. Paesani. Modeling molecular interactions in water: From pairwise to many body potential energy functions. *Chemical Reviews*, 116(13):7501–7528, 2016. ISSN 0009-2665. doi: 10.1021/acs.chemrev.5b00644. URL `<Go to ISI>://WOS:000379794000003`.

[36] Joshua A. Rackers, Roseane R. Silva, Zhi Wang, and Jay W. Ponder. Polarizable water potential derived from a model electron density. *Journal of Chemical Theory and Computation*, 17(11):7056–7084, 2021. doi: 10.1021/acs.jctc.1c00628. URL `https://doi.org/10.1021/acs.jctc.1c00628`. PMID: 34699197.

[37] N. Gresh, G. A. Cisneros, T. A. Darden, and J. P. Piquemal. Anisotropic, polarizable molecular mechanics studies of inter- and intramoecular interactions and ligand-macromolecule complexes. a bottom-up strategy. *Journal of Chemical Theory and Computation*, 3(6):1960–1986, 2007. ISSN 1549-9618. doi: Doi 10.1021/Ct700134r. URL `<Go to ISI>://WOS:000251024200008`.

[38] V. Babin, C. Leforestier, and F. Paesani. Development of a "first principles" water potential with flexible monomers: Dimer potential energy surface, vrt spectrum, and second virial coefficient. *J Chem Theory Comput*, 9(12):5395–5403, 2013. URL `<Go to ISI>://WOS:000328437500020`.

[39] V. Babin, G. R. Medders, and F. Paesani. Toward a universal water model: First principles simulations from the dimer to the liquid phase. *J Phys Chem Lett*, 3(24):3765–3769, 2012. URL `<Go to ISI>://WOS:000312762900015`.

[40] A. K. Das, O. N. Demerdash, and T. Head-Gordon. Improvements to the amoeba force field by introducing anisotropic atomic polarizability of the water molecule. *J Chem Theory Comput*, 14(12): 6722–6733, 2018. ISSN 1549-9626 (Electronic) 1549-9618 (Linking). doi: 10.1021/acs.jctc.8b00978. URL `https://www.ncbi.nlm.nih.gov/pubmed/30428257`.

[41] Kimberly Chenoweth, Adri CT Van Duin, and William A Goddard. Reaxff reactive force field for molecular dynamics simulations of hydrocarbon oxidation. *The Journal of Physical Chemistry A*, 112 (5):1040–1053, 2008.

[42] Sudhir B Kylasa, Hasan Metin Aktulga, and Ananth Y Grama. Puremd-gpu: A reactive molecular dynamics simulation package for gpus. *Journal of Computational Physics*, 272:343–359, 2014.

[43] Johann Gasteiger and Mario Marsili. A new model for calculating atomic charges in molecules. *Tetrahedron Letters*, 19(34):3181–3184, 1978. ISSN 0040-4039.

[44] Wilfried J. Mortier, Karin Van Genechten, and Johann Gasteiger. Electronegativity equalization: application and parametrization. *J. Am. Chem. Soc.*, 107(4):829–835, 1985. doi: 10.1021/ja00290a017. URL `https://doi.org/10.1021/ja00290a017`.

[45] Wilfried J. Mortier, Swapan K. Ghosh, and S. Shankar. Electronegativity Equalization Method for the Calculation of Atomic Charges in Molecules. *J. Am. Chem. Soc.*, 108(15):4315–4320, 1986. ISSN 15205126. doi: 10.1021/ja00275a013.

[46] Anthony K Rappe and William A Goddard III. Charge equilibration for molecular dynamics simulations. *The Journal of Physical Chemistry*, 95(8):3358–3363, 1991.

[47] Itai Leven, Hongxia Hao, Akshaya Kumar Das, and Teresa Head-Gordon. A reactive force field with coarse-grained electrons for liquid water. *J. Phys Chem. Lett.*, 11:9240–9247, oct 2020. ISSN 1948-7185. doi: 10.1021/acs.jpclett.0c02516. URL `https://dx.doi.org/10.1021/acs.jpclett.0c02516`.

[48] David Furman and David J Wales. Transforming the accuracy and numerical stability of reaxff reactive force fields. *The journal of physical chemistry letters*, 10(22):7215–7223, 2019.

[49] F. Romano, E. Sanz, and F. Sciortino. Crystallization of tetrahedral patchy particles in silico. *J Chem Phys*, 134(17):174502, 2011. ISSN 1089-7690 (Electronic) 0021-9606 (Linking). doi: 10.1063/1.3578182. URL `https://www.ncbi.nlm.nih.gov/pubmed/21548694`.

[50] E. B. Moore and V. Molinero. Structural transformation in supercooled water controls the crystallization rate of ice. *Nature*, 479(7374):506–8, 2011. ISSN 1476-4687 (Electronic) 0028-0836 (Linking). doi: 10.1038/nature10586. URL `https://www.ncbi.nlm.nih.gov/pubmed/22113691`.

[51] F. Smallenburg and F. Sciortino. Tuning the liquid-liquid transition by modulating the hydrogen-bond angular flexibility in a model for water. *Phys. Rev. Lett.*, 115(1):015701, 2015.

[52] F. H. Stillinger and T. A. Weber. Computer simulation of local order in condensed phases of silicon. *Phys Rev B Condens Matter*, 31(8):5262–5271, 1985. ISSN 0163-1829 (Print) 0163-1829 (Linking). doi: 10.1103/physrevb.31.5262. URL `https://www.ncbi.nlm.nih.gov/pubmed/9936488`.

[53] A. Hudait, N. Odendahl, Y. Qiu, F. Paesani, and V. Molinero. Ice-nucleating and antifreeze proteins recognize ice through a diversity of anchored clathrate and ice-like motifs. *J Am Chem Soc*, 140(14): 4905–4912, 2018. ISSN 1520-5126 (Electronic) 0002-7863 (Linking). doi: 10.1021/jacs.8b01246. URL `https://www.ncbi.nlm.nih.gov/pubmed/29564892`.

[54] A. Hudait, Y. Qiu, N. Odendahl, and V. Molinero. Hydrogen-bonding and hydrophobic groups contribute equally to the binding of hyperactive antifreeze and ice-nucleating proteins to ice. *J Am Chem Soc*, 141(19):7887–7898, 2019. ISSN 1520-5126 (Electronic) 0002-7863 (Linking). doi: 10.1021/jacs.9b02248. URL `https://www.ncbi.nlm.nih.gov/pubmed/31020830`.

[55] G. S. Ayton and G. A. Voth. Hybrid coarse-graining approach for lipid bilayers at large length and time scales. *J Phys Chem B*, 113(13):4413–24, 2009. ISSN 1520-6106 (Print) 1520-5207 (Linking). doi: 10.1021/jp8087868. URL `https://www.ncbi.nlm.nih.gov/pubmed/19281167`.

[56] S. Izvekov and G. A. Voth. Mixed resolution modeling of interactions in condensed-phase systems. *J Chem Theory Comput*, 5(12):3232–44, 2009. ISSN 1549-9618 (Print) 1549-9618 (Linking). doi: 10.1021/ct900414p. URL `https://www.ncbi.nlm.nih.gov/pubmed/26602507`.

[57] V. Krishna, W. G. Noid, and G. A. Voth. The multiscale coarse-graining method. iv. transferring coarse-grained potentials between temperatures. *J Chem Phys*, 131(2):024103, 2009. ISSN 1089-7690 (Electronic) 0021-9606 (Linking). doi: 10.1063/1.3167797. URL `https://www.ncbi.nlm.nih.gov/pubmed/19603966`.

[58] Y. Wang, W. G. Noid, P. Liu, and G. A. Voth. Effective force coarse-graining. *Phys Chem Chem Phys*, 11(12):2002–15, 2009. ISSN 1463-9076 (Print) 1463-9076 (Linking). doi: 10.1039/b819182d. URL `https://www.ncbi.nlm.nih.gov/pubmed/19280011`.

[59] L. Lu, S. Izvekov, A. Das, H. C. Andersen, and G. A. Voth. Efficient, regularized, and scalable algorithms for multiscale coarse-graining. *J Chem Theory Comput*, 6(3):954–65, 2010. ISSN 1549-9618 (Print) 1549-9618 (Linking). doi: 10.1021/ct900643r. URL `https://www.ncbi.nlm.nih.gov/pubmed/26613319`.

[60] I. F. Thorpe, D. P. Goldenberg, and G. A. Voth. Exploration of transferability in multiscale coarse-grained peptide models. *J Phys Chem B*, 115(41):11911–26, 2011. ISSN 1520-5207 (Electronic) 1520-5207 (Linking). doi: 10.1021/jp204455g. URL `https://www.ncbi.nlm.nih.gov/pubmed/21905717`.

[61] A. Davtyan, J. F. Dama, G. A. Voth, and H. C. Andersen. Dynamic force matching: A method for constructing dynamical coarse-grained models with realistic time dependence. *J Chem Phys*, 142 (15):154104, 2015. ISSN 1089-7690 (Electronic) 0021-9606 (Linking). doi: 10.1063/1.4917454. URL `https://www.ncbi.nlm.nih.gov/pubmed/25903863`.

[62] J. W. Wagner, T. Dannenhoffer-Lafage, J. Jin, and G. A. Voth. Extending the range and physical accuracy of coarse-grained models: Order parameter dependent interactions. *J Chem Phys*, 147 (4):044113, 2017. ISSN 1089-7690 (Electronic) 0021-9606 (Linking). doi: 10.1063/1.4995946. URL `https://www.ncbi.nlm.nih.gov/pubmed/28764380`.

[63] J. Jin and G. A. Voth. Ultra-coarse-grained models allow for an accurate and transferable treatment of interfacial systems. *J Chem Theory Comput*, 14(4):2180–2197, 2018. ISSN 1549-9626 (Electronic) 1549-9618 (Linking). doi: 10.1021/acs.jctc.7b01173. URL `https://www.ncbi.nlm.nih.gov/pubmed/29481754`.

[64] Antti-Pekka Hynninen and Michael F. Crowley. New faster charmm molecular dynamics engine. *Journal of Computational Chemistry*, 35(5):406–413, 2014. doi: 10.1002/jcc.23501. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.23501`.

[65] L. Lagardère, L.-H. Jolly, F. Lipparini, Félix Aviat, B. Stamm, Z. F. Jing, M. Harger, H. Torabifard, G. A. Cisneros, M. J. Schnieders, N. Gresh, Y. Maday, P. Y. Ren, J. W. Ponder, and J.-P. Piquemal. Tinker-hp: a massively parallel molecular dynamics package for multiscale simulations of large complex systems with advanced point dipole polarizable force fields. *Chem. Sci.*, 9:956–972, 2018.

[66] G. Schrack. Finding neighbors of equal size in linear quadtrees and octrees in constant time. *CVGIP: Image Understanding*, 55:221–230, 1992.

[67] M. Frigo and S.-G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93: 216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation.

[68] M. Frigo and S. G. Johnson. *FFTW tutorial for version 3.3.8*, 2018.

[69] Szilárd Páll and Berk Hess. A flexible algorithm for calculating pair interactions on simd architectures. *Computer Physics Communications*, 184(12):2641 – 2650, 2013. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2013.06.003. URL `http://www.sciencedirect.com/science/article/pii/S0010465513001975`.

[70] Jaewoon Jung, Takaharu Mori, Chigusa Kobayashi, Yasuhiro Matsunaga, Takao Yoda, Michael Feig, and Yuji Sugita. Genesis: a hybrid-parallel and multi-scale molecular dynamics simulator with enhanced sampling algorithms for biomolecular and cellular simulations. *WIREs Computational Molecular Science*, 5(4):310–323, 2015. doi: 10.1002/wcms.1220. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1220`.

[71] L. Lagardere, L. H. Jolly, F. Lipparini, F. Aviat, B. Stamm, Z. F. Jing, M. Harger, H. Torabifard, G. A. Cisneros, M. J. Schnieders, N. Gresh, Y. Maday, P. Y. Ren, J. W. Ponder, and J. P. Piquemal. Tinker-hp: a massively parallel molecular dynamics package for multiscale simulations of large complex systems with advanced point dipole polarizable force fields. *Chem Sci*, 9 (4):956–972, 2018. ISSN 2041-6520 (Print) 2041-6520 (Linking). doi: 10.1039/c7sc04531j. URL `https://www.ncbi.nlm.nih.gov/pubmed/29732110`.

[72] D. Nocito and G. J. O. Berana. Fast divide-and-conquer algorithm for evaluating polarization in classical force fields. *J. Chem. Phys.*, 146:114103, 2017.

[73] L. P. Wang, T. Head-Gordon, J. W. Ponder, P. Ren, J. D. Chodera, P. K. Eastman, T. J. Martinez, and V. S. Pande. Systematic improvement of a classical molecular model of water. *J Phys Chem B*, 117(34):9956–9972, 2013. URL `http://www.ncbi.nlm.nih.gov/pubmed/23750713`.

[74] D. T. Margul and M. E. Tuckerman. A stochastic, resonance-free multiple time-step algorithm for polarizable models that permits very large time steps. *J. Chem. Theo. Comput.*, 12:2170–2180, 2016.

[75] Alex Albaugh, Mark E. Tuckerman, and Teresa Head-Gordon. Combining iteration-free polarization with large time step stochastic-isokinetic integration. *Journal of Chemical Theory and Computation*,

0(0):null, 0. doi: 10.1021/acs.jctc.9b00072. URL https://doi.org/10.1021/acs.jctc.9b00072. PMID: 30830768.

[76] I. Leven and T. Head-Gordon. Inertial extended-lagrangian scheme for solving charge equilibration models. *Phys Chem Chem Phys*, 21(34):18652–18659, 2019. ISSN 1463-9084 (Electronic) 1463-9076 (Linking). doi: 10.1039/c9cp02979f. URL https://www.ncbi.nlm.nih.gov/pubmed/31460521.

[77] Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.

[78] Michele Ceriotti, Cecilia Clementi, and O Anatole von Lilienfeld. Introduction: Machine learning at the atomic scale, 2021.

[79] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[80] Xiang Gao, Farhad Ramezanghorbani, Olexandr Isayev, Justin S Smith, and Adrian E Roitberg. Torchani: a free and open source pytorch-based deep learning implementation of the ani neural network potentials. *Journal of chemical information and modeling*, 60(7):3408–3415, 2020.

[81] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24): 241722, 2018.

[82] Mojtaba Haghighatlari, Jie Li, Xingyi Guan, Oufan Zhang, Akshaya Das, Christopher J. Stein, Farnaz Heidar-Zadeh, Meili Liu, Martin Head-Gordon, Luke Bertels, Hongxia Hao, Itai Leven, and Teresa Head-Gordon. Newtonnet: a newtonian message passing network for deep learning of interatomic potentials and forces. *Digital Discovery*, pages –, 2022. doi: 10.1039/D2DD00008C. URL http://dx.doi.org/10.1039/D2DD00008C.

[83] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[84] Xingyi Guan, Akshaya Das, Christopher J Stein, Farnaz Heidar-Zadeh, Luke Bertels, Meili Liu, Mojtaba Haghighatlari, Jie Li, Oufan Zhang, Hongxia Hao, Itai Leven, Martin Head-Gordon, and Teresa Head-Gordon. A benchmark dataset for hydrogen combustion. *Scientific Data*, 9(1):1–7, 2022.

[85] R. S. Mulliken. Electronic population analysis on lcao-mo molecular wave functions. i. *The Journal of Chemical Physics*, 23:1833–1840, 1955. doi: 10.1063/1.1740588.

[86] Patrick Bultinck, Christian Van Alsenoy, Paul W. Ayers, and Ramon Carbó-Dorca. Critical analysis and extension of the hirshfeld atoms in molecules. *Journal of Chemical Physics*, 126, 2007. doi: 10.1063/1.2715563.

[87] Thomas A. Manz and David S. Sholl. Improved atoms-in-molecule charge partitioning functional for simultaneously reproducing the electrostatic potential and chemical states in periodic and nonperiodic materials. *Journal of Chemical Theory and Computation*, 8:2844–2867, 2012. doi: 10.1021/ct3002199.

[88] Alan E. Reed, Robert B. Weinstock, and Frank Weinhold. Natural population analysis. *The Journal of Chemical Physics*, 83:735–746, 1985. doi: 10.1063/1.449486.

[89] A. D. Mackerell. Empirical force fields for biological macromolecules: Overview and issues. *J Comput Chem*, 25(13):1584–1604, 2004.

[90] E. Jurrus, D. Engel, K. Star, K. Monson, J. Brandi, L. E. Felberg, D. H. Brookes, L. Wilson, J. Chen, K. Liles, M. Chun, P. Li, D. W. Gohara, T. Dolinsky, R. Konecny, D. R. Koes, J. E. Nielsen, T. Head-Gordon, W. Geng, R. Krasny, G. W. Wei, M. J. Holst, J. A. McCammon, and N. A. Baker. Improvements to the apbs biomolecular solvation software suite. *Protein Sci*, 27(1): 112–128, 2018. ISSN 1469-896X (Electronic) 0961-8368 (Linking). doi: 10.1002/pro.3280. URL `https://www.ncbi.nlm.nih.gov/pubmed/28836357`.

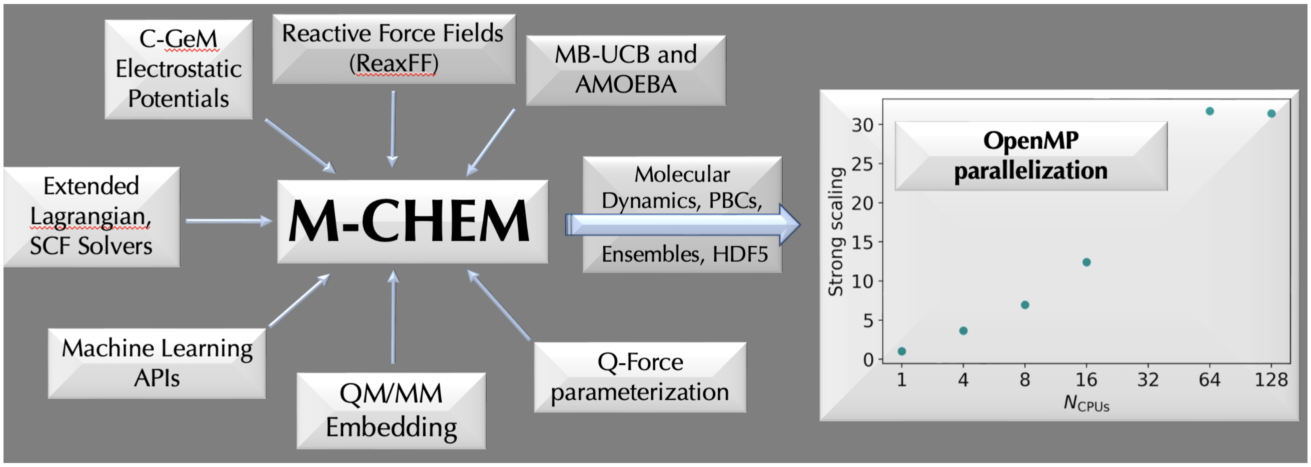

Figure 4: TOC graphic.