![JOSS - The Journal of Open Source Software]

ELECTRIC: Electric fields Leveraged from multipole Expansion Calculations in Tinker Rapid Interface Code

Jessica Nash[1, 2]**, Taylor Barnes**[1, 2]**, and Valerie Vaissier Welborn**[*2]

1 The Molecular Sciences Software Institute, 1880 Pratt Drive, Suite 1100, Blacksburg, VA 24060 **2** Department of Chemistry, Davidson Hall, Virginia Tech, 1040 Drillfield Drive, Blacksburg, VA 24061

Summary

Polarizable force fields have changed the landscape of biomolecular simulation, mostly by featuring improved electrostatic potential energy terms (Jing et al., 2019). These novel energy functions allow environment-driven changes in charge distribution, which yield simulations with improved geometries and molecular properties. In particular, the AMOEBA polarizable force field exhibits two fundamental changes compared to more traditional fixed charge force fields (Shi et al., 2013; Zhang et al., 2018). The first one relates to permanent electrostatics, expressed in AMOEBA in terms of atomic multipoles (truncated at quadrupoles) that account for anisotropy in the computed charge distributions. The second one represents polarizability through an induced dipole term that can respond to the chemical environment. These modified terms make the AMOEBA force field more physically grounded than other force fields and is the basis for more realistic simulations of biomolecular systems.

Improved electrostatics with AMOEBA give a unique opportunity to accurately compute electric fields, powerful metrics of catalytic activity in enzymes and other systems (Bhowmick, Sharma, & Head-Gordon, 2017; Vaissier, Sharma, Schaettle, Zhang, & Head-Gordon, 2018; Vaissier Welborn, Ruiz Pestana, & Head-Gordon, 2018). Electric fields projected onto specific bonds report on the effect of the surroundings (interacting via coulombic interactions, solvent effects, hydrogen bonding or other forces, all mostly electrostatic in nature) on the flow of electrons along these bonds. Therefore, projected electric fields are correlated to the probability of breaking these bonds, making them a useful probe of chemical reactivity.

ELECTRIC (J. Nash & Welborn, 2020) is a MolSSI Driver Interface (MDI) (Taylor A. Barnes, 2020; Taylor Arnold Barnes, 2020) driver that utilizes Tinker (Rackers et al., 2018) to analyze specific components of electric fields that are modeled using the AMOEBA force field. `ELECTRIC` parses Tinker trajectories and orchestrates additional Tinker calculations in order to project components of the electric fields onto user-defined bonds (specified by two atoms). It outputs the field in MV/cm, which is the sum of the direct field (from permanent electrostatics) and the induced field (from the induce dipole term), projected onto the bond unit vector (i.e., normalized by the bond length). `ELECTRIC` enables splitting of the total field into contributions from different components of the system, by molecules or by residues as specified in a reference PDB file. In summary, `ELECTRIC` was designed to expand quantitative system characterization via the computation of electric fields with user-friendly processing tools of Tinker-AMOEBA simulations.

In practice, the user needs a Tinker trajectory file (i.e., `filename.arc`), a Tinker input file (i.e., `tinker.key`) stripped from any keywords corresponding to periodic boundary conditions and a Tinker reference snapshot (`filename.xyz` - this can be the first frame in `filename.arc`) where the dimensions of the box, usually printed on line 2, has been deleted. Tinker is then

---

*Corresponding author

launched as an MDI engine before the `ELECTRIC` driver. Options regarding the electric fields calculations, such as probe index number, number of frames at the beginning of the trajectory file to skip (equilibration procedure), reference PDB file, etc. are specified as command line arguments when launching the driver. A complete usage procedure is provided in the `README` file.

Statement of Need

Since electric fields assist the motion of charges such as ions or electrons, they can link structure to function in molecular dynamics simulations. By taking advantage of their additivity property, we can decompose the total electric field into contributions from each system component (protein residues, solvent molecules, etc.). While this approach has already been used to probe electron flow along the bonds that break and form during a catalyzed reaction (Bhowmick et al., 2017; Vaissier et al., 2018), its applicability reaches many other research areas as it can also be used to probe ion transport or electron reorganization upon molecular binding, for example.

Mathematics

The electric field at atom i, $\vec{E}^i$, has components defined as:

$$E_x^i = \sum_j E_x^{j \to i} = \sum_j \left(E_{x,\mathrm{perm}}^{j \to i} + E_{x,\mathrm{ind}}^{j \to i} \right) \tag{1}$$

where E_x^i is the x-component of the electric field on atom i, $E_x^{j \to i}$ is the x-component of the electric field on atom i due to atom j and "perm" and "ind" refer to permanent and induced fields, respectively.

Each atom i is characterized by permanent atomic multipoles, including a monopole (charge) q^i, a dipole $\{\mu_x^i, \mu_y^i, \mu_z^i\}$ and a quadrupole $\{Q_{xx}^i, Q_{xx}^i, Q_{xy}^i, Q_{xz}^i ... Q_{zz}^i\}$, such that

$$E_{x,\mathrm{perm}}^{j \to i} = -T_x q^j + \sum_{m=y,z} T_{xm} \mu_m^j - \frac{1}{3} \sum_{m=y,z} \sum_{n=y,z} T_{xmn} Q_{mn}^j, \tag{2}$$

and

$$E_{x,\mathrm{ind}}^{j \to i} = \sum_{m=y,z} = T_{xm} \mu_{\mathrm{ind},m}^j, \tag{3}$$

where

$$T_{xy...} = \frac{1}{4\pi\epsilon_0} \nabla_x \nabla_y ... \frac{1}{r_{ij}}, \tag{4}$$

as also defined in Bhowmick et al. (2017).

Similar equations can be written for the y- and z-components of the field. The field projected onto a specific bond, say bond ij between atoms i and j, is then calculated as:

$$E_{\mathrm{proj}}^{ij} = \left(\frac{\vec{E}^i + \vec{E}^j}{2} \right) . \vec{u}^{ij}, \tag{5}$$

where E_{proj}^{ij} is the electric field projected onto bond ij and $\vec{u}^{ij}$ the unitary vector defining bond ij.

Acknowledgements

References

Barnes, T. A. (2020). The molssi driver interface library. *GitHub repository*. GitHub. Retrieved from https://github.com/MolSSI-MDI/MDI_Library.git

Barnes, T. A. (2020). *MolSSI driver interface library*. Zenodo. doi:10.5281/zenodo.3659285

Bhowmick, A., Sharma, S. C., & Head-Gordon, T. (2017). The importance of the scaffold for de novo enzymes: A case study with kemp eliminase. *J. Am. Chem. Soc.*, *139*(16), 5793–5800. doi:10.1021/jacs.6b12265

Jing, Z., Liu, C., Cheng, S. Y., Qi, R., Walker, B. D., Piquemal, J.-P., & Ren, P. (2019). Polarizable force fields for biomolecular simulations: Recent advances and applications. *Annu. Rev. Biophys.*, *48*(1), 371–394. doi:10.1146/annurev-biophys-070317-033349

J. Nash, T. B., & Welborn, V. V. (2020). ELECTRIC: Electric fields leveraged from multipole expansion calculations in tinker rapid interface code. *GitHub repository*. GitHub. Retrieved from https://github.com/WelbornGroup/ELECTRIC

Rackers, J. A., Wang, Z., Lu, C., Laury, M. L., Lagardĕre, L., Schnieders, M. J., Piquemal, J.-P., et al. (2018). Tinker 8: Software tools for molecular design. *Journal of Chemical Theory and Computation*, *14*(10), 5273–5289. doi:10.1021/acs.jctc.8b00529

Shi, Y., Xia, Z., Zhang, J., Best, R., Wu, C., Ponder, J. W., & Ren, P. (2013). Polarizable atomic multipole-based amoeba force field for proteins. *J. Chem. Theory Comput.*, *9*(9), 4046–4063. doi:10.1021/ct4003702

Vaissier, V., Sharma, S. C., Schaettle, K., Zhang, T., & Head-Gordon, T. (2018). Computational optimization of electric fields for improving catalysis of a designed kemp eliminase. *ACS Catal.*, *8*(1), 219–227. doi:10.1021/acscatal.7b03151

Vaissier Welborn, V., Ruiz Pestana, L., & Head-Gordon, T. (2018). Computational optimization of electric fields for better catalysis design. *Nat. Catal.*, 649–655. doi:10.1038/s41929-018-0109-2

Zhang, C., Lu, C., Jing, Z., Wu, C., Piquemal, J.-P., Ponder, J. W., & Ren, P. (2018). AMOEBA polarizable atomic multipole force field for nucleic acids. *J. Chem. Theory Comput.*, *14*(4), 2084–2108. doi:10.1021/acs.jctc.7b01169