Efficient Open-Source Implementations of Linear-Scaling Polarizable Embedding: Use Octrees to Save the Trees

Maximilian Scheurer,[*,†,§] Peter Reinholdt,[*,‡,§] Jógvan Magnus Haugaard Olsen,[¶] Andreas Dreuw,[†] and Jacob Kongsted[‡]

†*Interdisciplinary Center for Scientific Computing, Heidelberg University, D-69120 Heidelberg, Germany*

‡*Department of Physics, Chemistry and Pharmacy, University of Southern Denmark, DK-5230 Odense M, Denmark*

¶*DTU Chemistry, Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark*

§*Contributed equally to this work*

E-mail: maximilian.scheurer@iwr.uni-heidelberg.de; reinholdt@sdu.dk

Abstract

We present open-source implementations of the linear-scaling Fast Multipole Method (FMM) within the Polarizable Embedding (PE) model for efficient treatment of large polarizable environments in computational spectroscopy simulations. The implementations are tested for accuracy, efficiency, and usability on model systems as well as more realistic biomolecular systems. We explain how FMM parameters affect the calculation of molecular properties and show that PE calculations employing FMM can be carried out in a black-box manner. The efficiency of the linear-scaling approach is demonstrated by simulating the UV/Vis spectrum of a chromophore in an environment of more than

1

one million polarizable sites. Our implementations are interfaced to several open-source quantum chemistry programs, making computational spectroscopy simulations within the PE model and FMM available to a large variety of methods and a broad user base.

1 Introduction

Polarizable embedding (PE) models are nowadays ubiquitously present in computational molecular sciences and are routinely used to address a variety of biochemical systems with regards to dynamics simulations and computational spectroscopy.[1–8] In this context, PE models are considered the future gold standard technique, as more and more implementations and applications are becoming available.[6] Leaving the coupling of the environment and the quantum region aside, all PE-related models require the solution of classical polarization response equations, where, for traditional schemes, the rate-limiting step scales quadratically with the number of polarizable sites in the environment. Spurred by this bottleneck, linear-scaling implementations relying on the Fast Multipole Method (FMM)[9] have been developed in recent years.[10,11] We especially want to point out the generalized linear-scaling implementation by Lipparini for a general overview.[11] The scaling of iteratively evaluated terms arising in PE models is illustrated in Figure 1. The quantum→classical polarization consists of density-dependent electric field expectation values $\mathbf{F}_{el}[\rho_{el}]$ evaluated at each polarizable site in the environment. The classical→quantum polarization is, in addition to the presence of the permanent multipoles in the environment, described by the induction operator $\hat{v}_{ind}$ taking into account the induced field at each environment site. Hence, both terms scale linearly with the number of polarizable sites. The induced fields $\mathbf{F}_{ind}$, required to solve the induced dipole polarization equations, are evaluated for each pair of polarizable sites, such that this term scales quadratically with the number of sites. Using FMM, this bottleneck can be mitigated, such that evaluation of $\mathbf{F}_{ind}$ scales linearly as well. Detailed derivations and working equations for all terms of the PE model can be found in the literature.[1–3,7] We will not delve deeply into the theory behind FMM, but rather refer to the
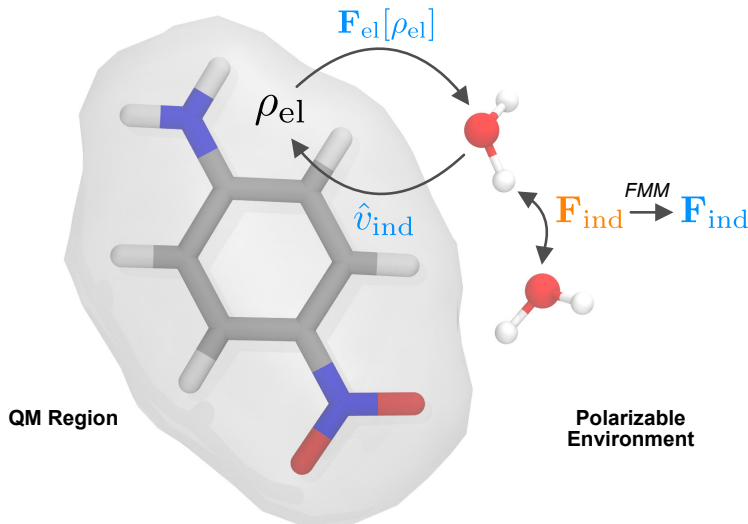
Figure 1: Illustration of iteratively evaluated interaction terms in PE models. Terms that scale linearly with the number of polarizable sites N_{sites} are shown in blue, whereas quadratically scaling terms are shown in orange. Using FMM, the quadratically scaling term to compute the induced fields $\mathbf{F}_{\text{ind}}$ at all polarizable sites becomes linearly scaling.

excellent descriptions in Refs. 12–14. Briefly, FMM consists of the following key steps:

- The system is (recursively) subdivided into hierarchical boxes using an octree.

- Multipole expansions of the boxes are computed up to a given expansion order p. This is done in a bottom-up manner: first, multipole expansions of the individual source distributions in the "leaf-level" boxes are calculated (P2M). The multipole distributions are then translated up the tree from child to parent until the root node is reached (M2M).

- Based on the angle-opening criterion, with the angle θ, interaction lists are generated to determine which cell-cell interactions should be evaluated directly (P2P) and which should be evaluated as far-field expansions.

- Short-ranged interactions (P2P) are evaluated identically to standard direct methods

- The potential (derivatives) of far-field boxes are evaluated (M2L) and propagated down the octree (L2L).

- The long-ranged part of the potential is evaluated using the local expansion (L2P).

With this hierarchical decomposition of the system, the potential (or field) can then be evaluated at a cost that scales asymptotically as $\mathcal{O}(N_{\text{sites}})$.

Previous implementations of FMM in the context of PE models have mainly been focused on polarizable molecular dynamics simulations, where the polarization equations usually need to be solved repeatedly for each time step (unless extended-Lagrangian[15–17] or truncated schemes[18] are applied). For the computation of molecular response properties within the PE framework, however, repeated solution of the polarization equations is also necessary, especially for high-order response properties, where the induction operator must be obtained for a large number of trial vectors (transition densities), and can become a rate-limiting step when large molecular environments are incorporated. Compared to the iterations needed to converge the PE self-consistent field (SCF) electronic ground state, the number of polarization equations that need to be solved to obtain, e.g., excitation energies and transition moments are much larger.

To the best of our knowledge, all previously presented FMM implementation in the context of PE models are found in closed-source codes and thus not publicly available. To this end, we present a linear-scaling formulation of the PE model that we have implemented in two open-source libraries, namely PElib and CPPE.[19,20] The implementation in PElib exposes PE-FMM to the Dalton program package,[21] and interfaces to LSDalton[21] and DIRAC[22] will be available soon. CPPE currently enables the linear-scaling functionalities in PySCF and Psi4,[23,24] and future releases of VeloxChem[25] and Q-Chem.[26]

The remainder of this article is structured as follows. First, a brief description of both implementations is given. Second, the implementations are tested for accuracy, performance, and scaling, and we provide suggestions for the selection of crucial model parameters. Third, the capabilities are demonstrated by investigating excited states of Nile Red in a sizeable biomolecular environment with more than a million polarizable sites.

2 Computational Methodology

2.1 Implementation Details

Both implementations are based on a Cartesian formulation of FMM with an adaptive octree algorithm.[13,14,27] They share the same tunable parameters, i.e., the angle opening criterion, θ, and expansion order, p, and both implementations have full support for the domain-specific requirements that embedding libraries impose, such as support for the exclusion of interactions and damped interaction kernels.[28,29]

The implementation in the CPPE library[20] is based on autogenerated C++ code using fmmgen.[14] The fmmgen library generates Cartesian operators and all kernels required for FMM based on symbolic algebra with SymPy.[30] The script we used to generate the FMM kernels for CPPE can be found in the CPPE GitHub repository (`https://github.com/maxscheurer/cppe`). We enabled common subexpression elimination (CSE) and let the code generator expand the power function `std::pow(b, n)` as products up to $n = 11$. To allow for damped P2P kernels and streamlined code for multiple source orders, the fmmgen code was slightly adapted (`https://github.com/fmmgen/maxscheurer`). For the latter, we make use of template meta-programming such that the code path for each source order is fixed at compile time. The generated code comes with loop-based OpenMP parallelism included. The tree code in CPPE was adapted from the fmmgen code examples and interfaced to the low-level routines which require electric field computations. For experimentation and testing, the routines are also conveniently available via the CPPE Python API. Host programs that interface with CPPE do not require any changes to adapt for the linear-scaling FMM code except for exposing the additional options (which is a trivial change). FMM is available in CPPE as of version 0.3.1.

The PElib FMM implementation was written from scratch as a self-contained Fortran 90 module. It features support for arbitrary-order multipole sources, unlimited expansion orders, and regular Coulomb as well as damped kernels, and computation of electric po-

tentials, fields, and arbitrary-order field derivatives. One of the central quantities in the implementation are the multipole interaction tensors, i.e., derivatives of $1/R$, which are used for both the P2P and M2L kernels.[31] For efficiency reasons, low-order tensors (up to 12th order) are computed using a combination of automatic code-generation and CSE, as explained above, while higher orders are computed using the open-ended formula described in Ref. 32. The implementation is parallelized using MPI. A standalone version is available at `https://github.com/peter-reinholdt/fmm`, with interfacing possible through either Fortran modules or Python via F2PY.[33]

2.2 Test System Setup

The implementations were tested on the pNA molecule placed inside a water solvation box. The pNA geometry was taken from Ref. 34 and solvated using the PACKMOL package.[35] Eleven systems with different box sizes were generated, ranging from 1440 atoms to 193596 atoms, as illustrated in Figure 2. The system with 114465 polarizable sites (38155 H_2O molecules) will be referred to as pNA/38k H_2O in the following, as it is used in several test cases. The generated systems were not processed further because PACKMOL generates configurations without steric clashes, such that they can directly be used for benchmarking purposes. The environment water molecules were parametrized using LoProp[36,37] at the CAM-B3LYP/6-31+G* level of theory.[38,39] The parametrization of a single water molecule was run with PyFraME[40] and Dalton,[21] and parameters for the remaining waters were then assigned by translation and rotation of the parameters from the reference molecule. Thus, each atom is assigned a charge, dipole moment, quadrupole moment, and anisotropic dipole–dipole polarizability, i.e., all atoms in the test systems are polarizable.

2.3 Benchmark and Test Calculations

To assess the correctness and accuracy of both implementations, initial testing was performed on the systems described above. A parameter study for the expansion order p and the opening
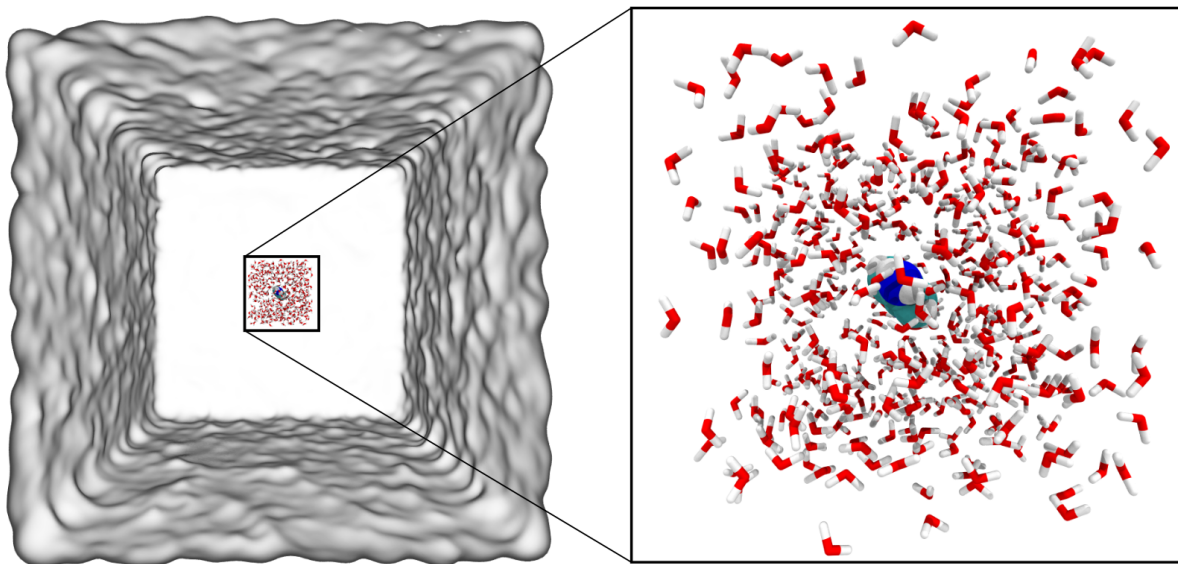
6

Figure 2: Size illustration of the generated pNA test systems. The smallest system (right) contains 1440 polarizable atoms, whereas the largest system in the test set contains 193596 polarizable atoms.

angle θ was run by evaluating the static electric field of the pNA/38k H_2O test system at all sites. The field evaluations from FMM were then compared to results from direct summation. Serial timings of the field evaluations were recorded to assess the performance for different FMM parameters. These calculations were run on the same hardware (Intel Xeon E5-2680 v3 processor), and both codes were compiled using the Intel 19 compiler. The choice of compiler can have quite a large impact on the performance of the fmmgen-generated code, and it was shown in the fmmgen publication that Intel compilers outperform, e.g., GCC in this case.[14]

The linear scaling in production PE-SCF and PE-TDDFT calculations, where a multitude of field evaluations is required, was then tested on all generated pNA systems. Calculations were performed with CPPE/PySCF and PElib/Dalton at the TD-CAM-B3LYP[38]/6-31G(d,p) level of theory. For PySCF, the Tamm-Dancoff approximation (TDA) was employed.[41] The five energetically lowest singlet states were computed. To test the MPI-parallelized PElib/Dalton FMM code, the test calculations were run both on a single node

(24 cores) and on ten nodes (240 cores). Note that the FMM code in CPPE is OpenMP-parallelized, such that all computations using this implementation are run on a single node with 24 OMP threads. Timings for the SCF and the linear response (LR) procedure were recorded separately. All calculations were run employing i) direct summation and ii) FMM with $p = 5$ and $\theta = 0.5$.

We computed the five lowest singlet excitation energies of pNA/38k H_2O with PE-ADC(2) and PE-TDDFT (CAM-B3LYP) and the 6-31G(d,p) basis set. PE-ADC(2) calculations were run with CPPE/PySCF in the adcc[42] toolkit employing perturbative corrections for the excitation energies, i.e., the induced moment equations are not solved during the ADC procedure itself.[43] The PE-TDDFT computations were run with PElib/Dalton as described above. The computations were run employing either i) direct summation and ii) FMM with $p = 5$ and $\theta = 0.5$.

2.4 Setup for Nile Red

A snapshot extracted from a QM/MM MD trajectory of the Nile Red molecule bound to a Beta-lactoglobulin protein (BLG) was taken from a previous study.[7] The coordinates of the Nile Red molecule were extracted directly and used as the QM region in the following. Embedding parameters of the protein, water, and ions were assigned using the PyFraME package. For the protein, we used the parameters described in Ref. 44, while parameters for the water molecules and ions were taken from Ref. 45. In addition to the primary simulation cell, we also created larger 3x1x1, 3x3x1 and 3x3x3 replicated cells with 3, 9, and 27 copies of the primary unit cell, respectively, which for the largest system yielded a total of 1,664,118 environment sites, as summarized in Table 1. For the PE calculations, we used FMM with $p = 5$ and $\theta = 0.5$. Thole-style damping was used for all systems using standard damping parameters. Three singlet excited states were computed using PE-ADC(2)/6-31G(d) with CPPE/PySCF in adcc[42] including perturbative corrections of the excitation energies,[43] whereas five singlet excited states were obtained with PE-TDDFT at

8

Table 1: Nile Red/BLG Systems

System Name	Unit Cells	N_{sites}
1x1x1	1	61,634
3x1x1	3	184,902
3x3x1	9	554,706
3x3x3	27	1,664,118

the CAM-B3LYP/6-31G(d) level of theory with PElib/Dalton. Local field effects were taken into account in the PE-TDDFT calculations via effective external fields (EEF).[46] The ADC calculations were run on a single node using 24 OpenMP threads, and the TDDFT jobs were run on 40 CPU cores on one node with MPI. The resulting stick spectra were convoluted with a Lorentzian broadening function with a half-width at half maximum value of 0.124 eV.

The code used to generate the pNA test systems and to produce all plots contained in this paper, together with example Python scripts, can be found on GitHub (`https://github.com/maxscheurer/pe_fmm`) and is archived on Zenodo under the DOI `10.5281/zenodo.4577182`.

3   Results and Discussion

3.1   Errors and Timings of Electric Field Evaluations

To test the accuracy of electric field evaluations by our two implementations, we conducted a parameter study varying the expansion order p and the opening angle θ for the pNA/38k H_2O test system. The error per site i is given by

$$F_i^{\text{err}} = \frac{\left\| \mathbf{F}_i^{\text{direct}} - \mathbf{F}_i^{\text{FMM}} \right\|}{\left\| \mathbf{F}_i^{\text{direct}} \right\|}. \tag{1}$$

The normalized error distributions for $\theta = 0.2, 0.3, 0.5, 0.7, 0.99$ and $p = 3, 5, 7$ for each implementation are shown in Figure 3. For small opening angles, both implementations show rather similar error distributions, whereas for large angles ($\theta = 0.99$), CPPE shows
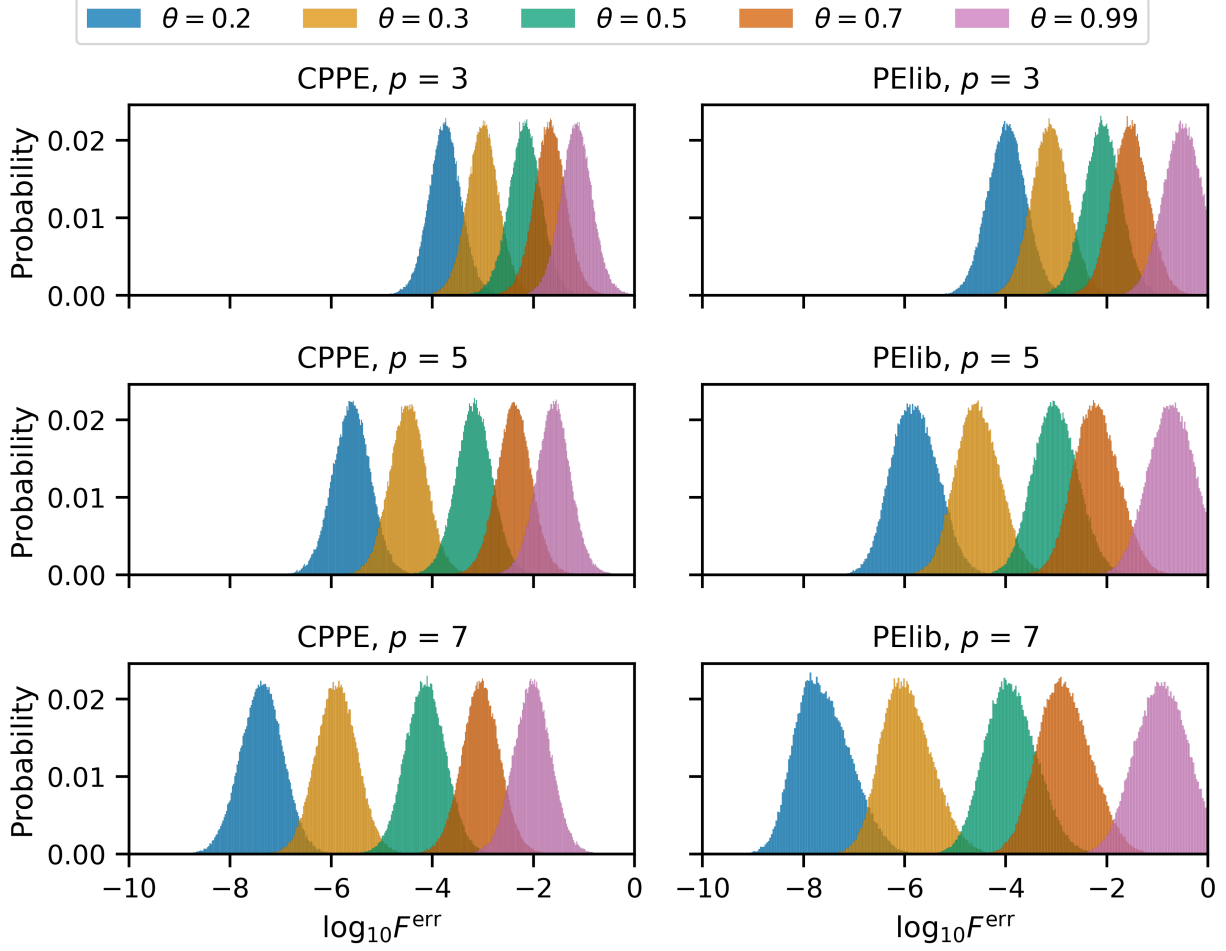
Figure 3: Accuracy of FMM fields with the CPPE (left column) and PElib (right column) implementations. The histograms show probability densities over the error defined in eq. (1). The errors are shown for different values of the expansion order p and opening angle θ for pNA/38k H_2O .

smaller field evaluation errors than PElib. As a matter of fact, an opening angle close to 1.00 should not be used in practice, such that the error differences are not concerning. Furthermore, it can be seen that the average error can be easily controlled using both the opening angle and the expansion order in a well-defined manner, comparable to previous implementations.[10] Since induced fields are evaluated iteratively during the SCF or LR procedures, the overall errors due to FMM field evaluations will, of course, accumulate. We will analyze this behavior in Section 3.3. The presented error distributions show that the accuracy of the implementations are well controllable and approach the result from direct

summation for large expansion orders and/or small opening angles.

Another aspect of ensuring the correctness of the two implementations is the time spent on the field evaluations. The tradeoff between computational effort and accuracy is, of course, also a significant factor in the choice of p and θ for practical simulations. The timings for the field evaluations in Figure 3 are summarized in Table 2, obtained without using any parallelization scheme. The performance of the two implementations is overall very similar.

Table 2: Serial timings for the field evaluations in Figure 3[a)]

θ	CPPE			PElib		
	$p = 3$	$p = 5$	$p = 7$	$p = 3$	$p = 5$	$p = 7$
0.20	668.20	666.66	692.21	780.89	795.43	851.89
0.30	255.36	312.30	272.18	215.26	219.74	243.22
0.50	69.00	76.86	74.62	55.41	59.76	72.75
0.70	33.26	34.01	35.99	20.81	24.65	37.26
0.99	12.17	12.36	13.52	5.81	9.44	21.69
Direct	1574.63			1522.91		

[a)] Timings are reported in seconds.

The run time of the direct summation evaluation is about 1500 seconds. The execution time is reduced by a factor of two with the most conservative (and slow) FMM parameters tested, whereas the fastest parameter set is up to 130–260 times faster. The run time increases with increasing expansion orders and decreasing opening angle, as expected. To put the timings in perspective with respect to the entailed errors, we observe that for $(p = 5, \theta = 0.50)$, the evaluation time decreases by approximately a factor of 20–25 for both implementations, while the average F^{err} is only 10^{-3}. In summary, the presented errors and timings show the anticipated behavior and allow us to efficiently evaluate electric fields with the expected error-performance trade-off. Based on both the timings and the field evaluation accuracy, we adopted the combination of $(p = 5, \theta = 0.50)$ in the following, since it gives a reasonable compromise between speed and tolerable errors.

3.2 Linear Scaling Test for SCF and Linear Response

Next, we will assess that our implementations show the expected linear scaling behavior in practical quantum chemical calculations. The use of FMM is beneficial for systems where the iterative evaluation of induced fields becomes the rate-limiting step, i.e., the classical part of the calculation cost dominates the quantum part. To this end, we recorded timings for SCF and LR computations on the pNA systems with direct summation and with FMM ($p = 5, \theta = 0.5$). We expect that for these environment-dominated systems, the wall times for SCF and LR procedures increase quadratically with the number of sites for direct summation, and FMM should show an asymptotic linear scaling. The recorded timings for both implementations are shown in Figure 4. The calculations with CPPE were run in PySCF
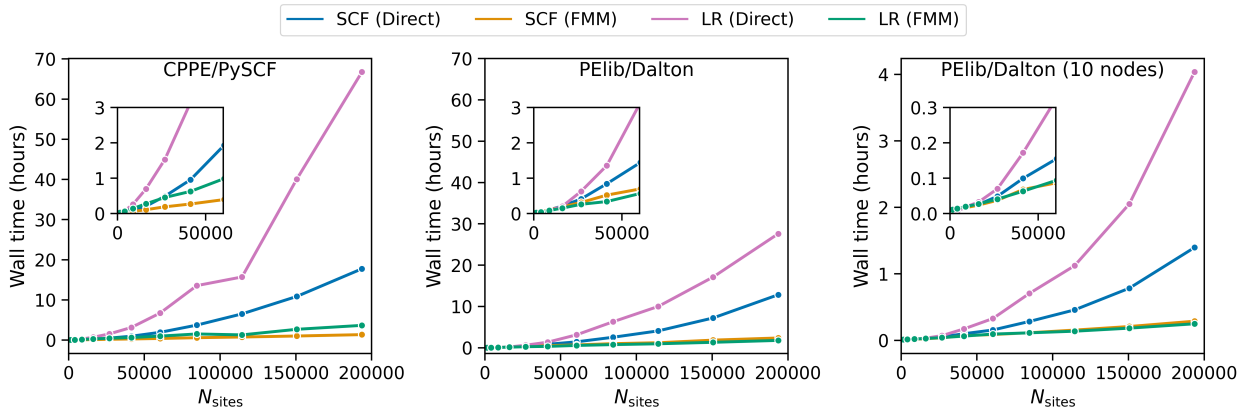

Figure 4: Scaling of direct and FMM summation schemes. Wall times for CAM-B3LYP/6-31G(d,p) SCF and linear response computations are shown. The systems consist of box with a single pNA molecule surrounded by water molecules, consisting of N_{sites} atoms. In case of CPPE/PySCF, the Tamm-Dancoff approximation (TDA) was used.

on a single compute node, whereas for PElib/Dalton, results for a single node and ten nodes are shown. First, for SCF and LR with direct summation schemes, a quadratic increase in wall time is observed. The prefactor for CPPE/PySCF in the LR computation is larger than for PElib/Dalton due to the number of iterations taken in the eigensolver procedure. Also, different efficiency of the linearly scaling parts of the host programs, such as the evaluations of density-dependent electric fields, will lead to slightly different performance characteristics

for the two codes. Nevertheless, the SCF timings are on the same order of magnitude for both setups. Second, employing FMM mitigates the quadratic scaling and reduces the run time for larger systems significantly. A robust linear scaling is observed for environments with up to 200,000 polarizable sites. It is interesting to see the early onset of the quadratic function for SCF and LR jobs for systems with less than 100,000 atoms. This shows that the effect of FMM can be already beneficial for environments that are commonly used in PE computations. We observe cross-over points between the total run time of the direct and FMM implementations at 1440 and 4083 sites for CPPE and PElib, respectively. For the systems with more than 100,000 atoms, the SCF takes at least three times longer with direct summation than with FMM. It is thus advisable to enable FMM for efficient PE computations already when moderately sized environments are used (perhaps from about 10000 sites). Such systems sizes (and larger ones) are frequently encountered in practical settings. Third, the setup with PElib/Dalton on ten nodes shows that the FMM implementation in PElib efficiently scales across several nodes. Of course, the benefit of using more CPU cores increases with a larger computational load. This can, however, be of importance when tackling systems with larger QM regions, and the implementation integrates well with the general parallelization strategy employed in Dalton (pure MPI). Since the majority of host programs for CPPE do not use MPI, we refrained from an MPI-parallel FMM implementation in CPPE.

3.3 Accuracy of Electronic Excitation Spectra

Until now, we have only presented the accuracy of single field evaluations and performance associated with FMM in practical quantum chemical calculations. The actual errors of individual field evaluations are, however, only of minor interest for such calculations. More important is the propagation of the FMM error to molecular properties, such as excitation energies and transition moments. Since CPPE, PElib, and their host programs have a strong focus on computational spectroscopy, we analyzed how the FMM error affects PE-ADC(2)
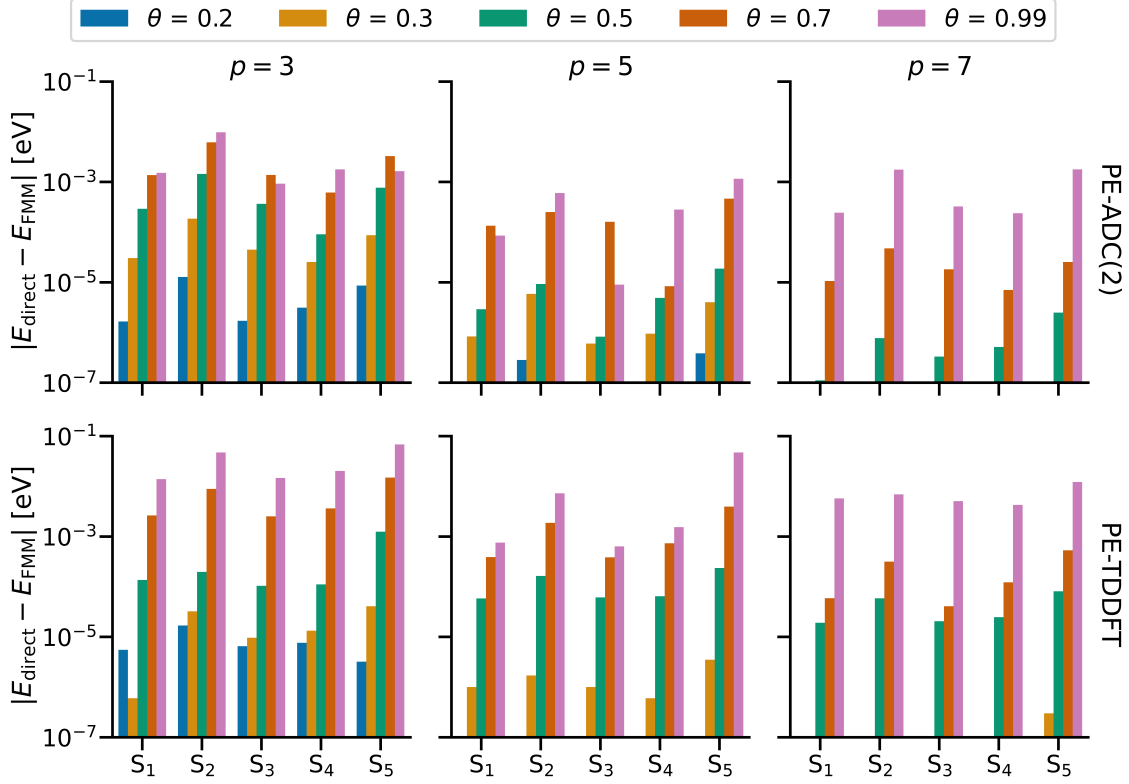
13

Figure 5: Errors in the excitation energies for pNA/38k H_2O computed with PE-ADC(2) and PE-TDDFT. Note that with some parameter combinations, the FMM result becomes identical to the direct reference to all printed digits, in which case no bar is shown.

and PE-TDDFT excitation energies. For this task, we used the pNA/38k H_2O test system again, and the errors of the excitation energies of the five energetically lowest singlet states are shown in Figure 5. Since the energy convergence threshold in the response solver for these computations is on the order of 10^{-5} au, all errors below this threshold are numerically equal to zero in this error analysis. For $p = 3$, mean errors are on the order of 0.5 meV and the maximum error is 9.7 meV for PE-ADC(2). For PE-TDDFT, the mean error is 3.9 meV and the maximum error amounts to approximately 0.07 eV. These large errors are only found when $\theta = 0.99$ is used, which is thus not advisable in practice. A good trade-off in the accuracy is found for $p = 5$ and $\theta = 0.5$, where the excitation energy errors in this case are in the range between 10^{-5} and 10^{-3} eV. For practical calculations, i.e., analysis and plotting of excitation spectra, these errors are negligibly small. The most important conclusion from this error analysis on practical calculations is that *all* observed errors are well below the

intrinsic error of the employed quantum chemical method.[47–49] This is encouraging because it allows one to use PE in combination with FMM almost in a black-box manner for excited-state calculations when reasonable defaults for the tree parameters are set (e.g., $p = 5$ and $\theta = 0.5$ in our codes). As such, FMM can also be employed by users not familiar with the underlying principles without reducing the quality of the obtained results. How FMM affects the accuracy of higher-order response properties would, however, require further analysis.

3.4 Sizeable Biomolecular Systems: Nile Red in BLG

To showcase the robustness of the presented FMM implementations, we ran excited state computations with PE-ADC(2) and PE-TDDFT on the Nile Red molecule embedded in BLG and solvated by water. An overview of the systems is presented in Figure 6. The system size was artificially increased for demonstration purposes by replicating the unit cell of the MD simulation box successively in all directions (Tab. 1). The timings for individual computational tasks and resulting spectra are presented in Figure 7.
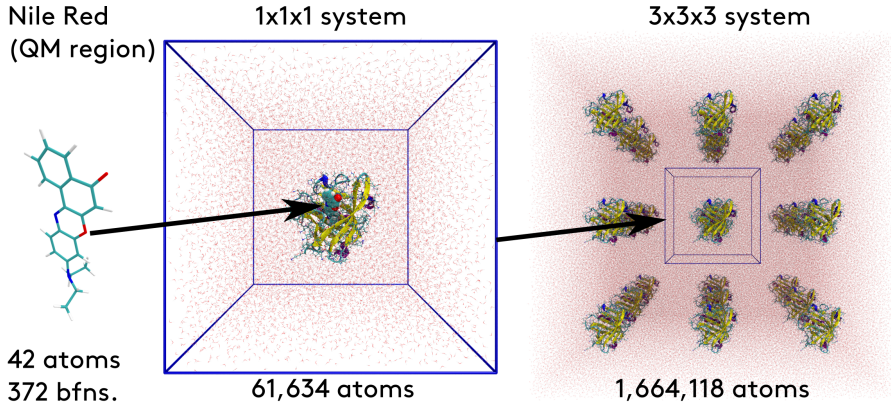

Figure 6: Illustration of Nile Red/BLG systems. Left: Nile Red molecule (QM region). Center: single unit cell containing water, protein, and ions. Right: 27 unit cells (3x3x3) replicated environment.

The largest system (3x3x3) contains more than 1.6 million polarizable sites in the environment. In the PE-ADC(2) calculations, the number of polarizable sites only affects the run time of the SCF and the perturbative corrections (PTC). The ADC eigenvalue problem
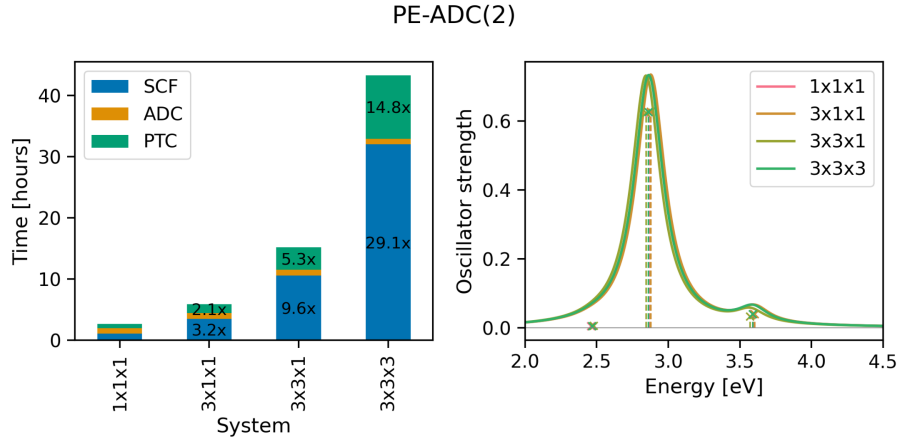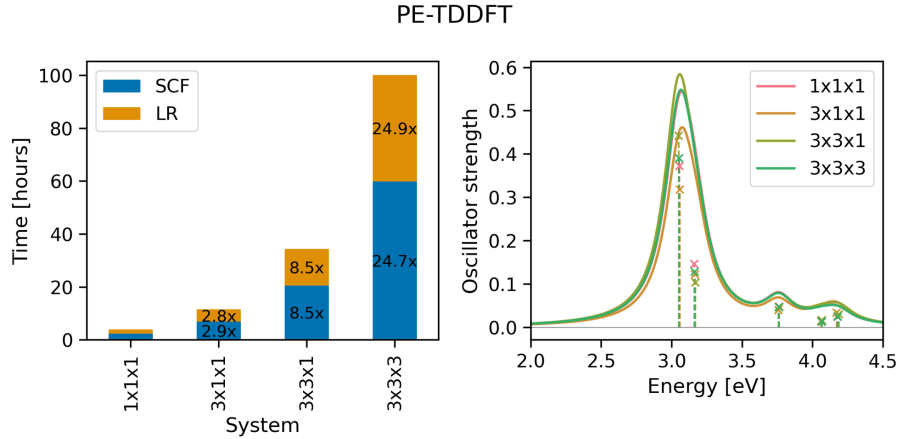
Figure 7: Timings and excitation spectra for Nile Red/BLG computed with a) PE-ADC(2) and b) PE-TDDFT employing FMM.

is solved without direct environment coupling, such that it always requires the same computational cost. Theoretical time increases would be 3-fold, 9-fold, and 27-fold for the 3x1x1, 3x3x1, and 3x3x3 system, respectively, when compared to the single unit cell 1x1x1 system. This trend is present for the SCF timings with CPPE/PySCF. Even though the observed relative timing factors are a bit larger than the theoretical estimates, the implementation is still capable of treating the largest systems in a linearly scaling manner. For PTC, the relative timing factors are below the theoretical estimate due to the fact that the number of iterations needed to solve for the induced moments varies a bit, especially when compar-

ing to the smallest system. The relative timings observed for PE-TDDFT are rather close to theoretical estimates, showing almost ideal linear scaling behavior. In contrast to the PE-ADC(2) calculations, the relative timing factors are all a bit smaller than the estimated value, which could be caused by the different underlying quantum chemical programs and the employed parallelization strategy. Most importantly, both implementations, when used in a real-world computation, are shown to be robust with respect to system size, and no unexpected bottlenecks arise when treating such large systems. Note that for these systems, the compute time using direct summation would be almost unbearable and thus inefficient for production calculations. Timing analyses clearly show that our implementations are ready to handle polarizable environments of any practically relevant size without any difficulties.

The observed eigenstates of the systems are expected to vary only slightly due to the long-range nature of the interactions with additional unit cells. This is indeed the case for all presented computations on Nile Red/BLG. For PE-ADC(2), no direct coupling to the polarizable environment is taken into account in the ADC procedure, such that intensities are virtually identical in each system. On the contrary, the intensity of the energetically lowest transition increases with system size for PE-TDDFT, where direct coupling to the environment is present in the linear response procedure. For both methods, excitation energies are almost independent of the environment size. This is expected because the excitation energies should already be well converged for the smallest system. The additional unit cells are just artificially increasing the computational cost of computing induced dipoles, and one can clearly see that the long-range nature of the polarization interactions has only negligible effects on the excitation energies. This test is, however, crucial to assess that both implementations behave reasonably also for large environments. Consequently, both implementations can be employed for arbitrarily large polarizable environments, and yield both efficient linear scaling and physically sound molecular properties.

17

4   Conclusion

We have presented the performance and accuracy of two open-source FMM implementations to accelerate the classical parts of PE calculations. Using FMM, the asymptotic scaling of the PE calculations with the number of sites in the environment is linear, and we observe speed-ups compared to direct-summation techniques already for systems with a few thousand polarizable sites. Further, we have demonstrated the correct linear scaling of the method, which allowed for treating systems with more than a million polarizable sites. We demonstrate that the method allows for an accurate evaluation of the electric fields needed for the solution of the induced dipole polarization equations. More importantly, we show how FMM can be safely applied with essentially no degradation to the accuracy of practical calculations of excitation energies. Our open-source implementations of FMM for PE are coupled to several quantum chemistry packages, such as PySCF, Psi4, and Dalton, which makes it affordable to apply PE for larger environments than ever before with a wide selection of wave-function or density-functional approaches, targeting a wealth of molecular properties.

Acknowledgement

References

(1) Olsen, J. M.; Aidas, K.; Kongsted, J. Excited States in Solution through Polarizable Embedding. *J. Chem. Theory Comput.* **2010**, *6*, 3721–3734.

(2) Olsen, J. M. H.; Kongsted, J. *Adv. Quantum Chem.*; 2011; Vol. 61; pp 107–143.

(3) List, N. H.; Olsen, J. M. H.; Kongsted, J. Excited states in large molecular systems through polarizable embedding. *Phys. Chem. Chem. Phys.* **2016**, *18*, 20234–20250.

(4) Steinmann, C.; Reinholdt, P.; Nørby, M. S.; Kongsted, J.; Olsen, J. M. H. Response properties of embedded molecules through the polarizable embedding model. *Int. J. Quantum Chem.* **2019**, *119*, e25717.

(5) Mennucci, B.; Corni, S. Multiscale modelling of photoinduced processes in composite systems. *Nat. Rev. Chem.* **2019**,

(6) Bondanza, M.; Nottoli, M.; Cupellini, L.; Lipparini, F.; Mennucci, B. Polarizable embedding QM/MM: the future gold standard for complex (bio) systems? *Phys. Chem. Chem. Phys.* **2020**, *22*, 14433–14448.

(7) Hartmann, P.; Reinholdt, P.; Kongsted, J. In *QM/MM Studies of Light-responsive Biological Systems*; Andruniów, T., Olivucci, M., Eds.; Springer International Publishing: Cham, 2021; pp 143–195.

(8) Nottoli, M.; Cupellini, L.; Lipparini, F.; Granucci, G.; Mennucci, B. Multiscale Models for Light-Driven Processes. *Annu. Rev. Phys. Chem.* **2021**,

(9) Greengard, L.; Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.* **1987**, *73*, 325–348.

(10) Coles, J. P.; Masella, M. The fast multipole method and point dipole moment polarizable force fields. *J. Chem. Phys.* **2015**, *142*, 024109.

(11) Lipparini, F. General Linear Scaling Implementation of Polarizable Embedding Schemes. *J. Chem. Theory Comput.* **2019**, *15*, 4312–4317.

(12) Dehnen, W. A very fast and momentum-conserving tree code. *Astrophys. J. Lett.* **2000**, *536*, L39.

(13) Visscher, P.; Apalkov, D. Simple recursive implementation of fast multipole method. *J. Magn. Magn. Mater.* **2010**, *322*, 275–281.

(14) Pepper, R. A.; Fangohr, H. fmmgen: Automatic Code Generation of Operators for Cartesian Fast Multipole and Barnes-Hut Methods. **2020**,

(15) Toukmaji, A.; Sagui, C.; Board, J.; Darden, T. Efficient particle-mesh Ewald based approach to fixed and induced dipolar interactions. *J. Chem. Phys.* **2000**, *113*, 10913–10927.

(16) Albaugh, A.; Demerdash, O.; Head-Gordon, T. An efficient and stable hybrid extended Lagrangian/self-consistent field scheme for solving classical mutual induction. *J. Chem. Phys.* **2015**, *143*, 174104.

(17) Albaugh, A.; Niklasson, A. M. N.; Head-Gordon, T. Accurate classical polarization solution with no self-consistent field iterations. *J. Phys. Chem. Lett.* **2017**, *8*, 1714–1723.

(18) Aviat, F.; Lagardère, L.; Piquemal, J.-P. The truncated conjugate gradient (TCG), a non-iterative/fixed-cost strategy for computing polarization in molecular dynamics: Fast evaluation of analytical forces. *J. Chem. Phys.* **2017**, *147*, 161724.

(19) Olsen, J. M. H.; List, N. H.; Steinmann, C.; Steindal, A. H.; Nørby, M. S.; Reinholdt, P. PElib: The Polarizable Embedding library. 2020; `https://doi.org/10.5281/zenodo.3967017`.

(20) Scheurer, M.; Reinholdt, P.; Kjellgren, E. R.; Olsen, J. M. H.; Dreuw, A.; Kongsted, J. CPPE: An open-source C++ and Python library for polarizable embedding. *J. Chem. Theory Comput.* **2019**, *15*, 6154–6163.

(21) Aidas, K.; Angeli, C.; Bak, K. L.; Bakken, V.; Bast, R.; Boman, L.; Christiansen, O.; Cimiraglia, R.; Coriani, S.; Dahle, P.; Dalskov, E. K.; Ekström, U.; Enevoldsen, T.; Eriksen, J. J.; Ettenhuber, P.; Fernández, B.; Ferrighi, L.; Fliegl, H.; Frediani, L.; Hald, K.; Halkier, A.; Hättig, C.; Heiberg, H.; Helgaker, T.; Hennum, A. C.; Hettema, H.; Hjertenæs, E.; Høst, S.; Høyvik, I.-M.; Iozzi, M. F.; Jansík, B.; Jensen, H. J. A.; Jonsson, D.; Jørgensen, P.; Kauczor, J.; Kirpekar, S.; Kjærgaard, T.; Klopper, W.; Knecht, S.; Kobayashi, R.; Koch, H.; Kongsted, J.; Krapp, A.; Kristensen, K.; Ligabue, A.; Lutnæs, O. B.; Melo, J. I.; Mikkelsen, K. V.; Myhre, R. H.; Neiss, C.; Nielsen, C. B.; Norman, P.; Olsen, J.; Olsen, J. M. H.; Osted, A.; Packer, M. J.; Pawlowski, F.; Pedersen, T. B.; Provasi, P. F.; Reine, S.; Rinkevicius, Z.; Ruden, T. A.; Ruud, K.; Rybkin, V. V.; Sałek, P.; Samson, C. C. M.; de Merás, A. S.; Saue, T.; Sauer, S. P. A.; Schimmelpfennig, B.; Sneskov, K.; Steindal, A. H.; Sylvester-Hvid, K. O.; Taylor, P. R.; Teale, A. M.; Tellgren, E. I.; Tew, D. P.; Thorvaldsen, A. J.; Thøgersen, L.; Vahtras, O.; Watson, M. A.; Wilson, D. J. D.; Ziolkowski, M.; Agren, H. The Dalton quantum chemistry program system. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2014**, *4*, 269–284.

(22) Saue, T.; Bast, R.; Gomes, A. S. P.; Jensen, H. J. A.; Visscher, L.; Aucar, I. A.; Di Remigio, R.; Dyall, K. G.; Eliav, E.; Fasshauer, E., et al. The DIRAC code for relativistic molecular calculations. *J. Chem. Phys.* **2020**, *152*, 204104.

(23) Sun, Q.; Zhang, X.; Banerjee, S.; Bao, P.; Barbry, M.; Blunt, N. S.; Bogdanov, N. A.; Booth, G. H.; Chen, J.; Cui, Z.-H.; Eriksen, J. J.; Gao, Y.; Guo, S.; Hermann, J.; Hermes, M. R.; Koh, K.; Koval, P.; Lehtola, S.; Li, Z.; Liu, J.; Mardirossian, N.; Mc-Clain, J. D.; Motta, M.; Mussard, B.; Pham, H. Q.; Pulkin, A.; Purwanto, W.; Robin-

son, P. J.; Ronca, E.; Sayfutyarova, E. R.; Scheurer, M.; Schurkus, H. F.; Smith, J. E. T.; Sun, C.; Sun, S.-N.; Upadhyay, S.; Wagner, L. K.; Wang, X.; White, A.; Whitfield, J. D.; Williamson, M. J.; Wouters, S.; Yang, J.; Yu, J. M.; Zhu, T.; Berkelbach, T. C.; Sharma, S.; Sokolov, A. Y.; Chan, G. K.-L. Recent developments in the PySCF program package. *J. Chem. Phys.* **2020**, *153*, 024109.

(24) Smith, D. G. A.; Burns, L. A.; Simmonett, A. C.; Parrish, R. M.; Schieber, M. C.; Galvelis, R.; Kraus, P.; Kruse, H.; Di Remigio, R.; Alenaizan, A.; James, A. M.; Lehtola, S.; Misiewicz, J. P.; Scheurer, M.; Shaw, R. A.; Schriber, J. B.; Xie, Y.; Glick, Z. L.; Sirianni, D. A.; O'Brien, J. S.; Waldrop, J. M.; Kumar, A.; Hohenstein, E. G.; Pritchard, B. P.; Brooks, B. R.; Schaefer, H. F., 3rd; Sokolov, A. Y.; Patkowski, K.; DePrince, A. E., 3rd; Bozkaya, U.; King, R. A.; Evangelista, F. A.; Turney, J. M.; Crawford, T. D.; Sherrill, C. D. Psi4 1.4: Open-source software for high-throughput quantum chemistry. *J. Chem. Phys.* **2020**, *152*, 184108.

(25) Rinkevicius, Z.; Li, X.; Vahtras, O.; Ahmadzadeh, K.; Brand, M.; Ringholm, M.; Scheurer, N. H. L. M.; Scott, M.; Dreuw, A.; Norman, P. VeloxChem: A Python-driven density-functional theory program for spectroscopy simulations in high-performance computing environments. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2020**, *10*, e1457.

(26) Shao, Y.; Gan, Z.; Epifanovsky, E.; Gilbert, A. T. B.; Wormit, M.; Kussmann, J.; Lange, A. W.; Behn, A.; Deng, J.; Feng, X.; Ghosh, D.; Goldey, M.; Horn, P. R.; Jacobson, L. D.; Kaliman, I.; Khaliullin, R. Z.; Kús, T.; Landau, A.; Liu, J.; Proynov, E. I.; Rhee, Y. M.; Richard, R. M.; Rohrdanz, M. A.; Steele, R. P.; Sundstrom, E. J.; Woodcock III, H. L.; Zimmerman, P. M.; Zuev, D.; Albrecht, B.; Alguire, E.; Austin, B.; Beran, G. J. O.; Bernard, Y. A.; Berquist, E.; Brandhorst, K.; Bravaya, K. B.; Brown, S. T.; Casanova, D.; Chang, C.-M.; Chen, Y.; Chien, S. H.; Closser, K. D.; Crittenden, D. L.; Diedenhofen, M.; DiStasio Jr., R. A.; Dop, H.; Dutoi, A. D.; Edgar, R. G.; Fatehi, S.; Fusti-Molnar, L.; Ghysels, A.; Golubeva-Zadorozhnaya, A.; Gomes, J.;

Hanson-Heine, M. W. D.; Harbach, P. H. P.; Hauser, A. W.; Hohenstein, E. G.; Holden, Z. C.; Jagau, T.-C.; Ji, H.; Kaduk, B.; Khistyaev, K.; Kim, J.; Kim, J.; King, R. A.; Klunzinger, P.; Kosenkov, D.; Kowalczyk, T.; Krauter, C. M.; Lao, K. U.; Laurent, A.; Lawler, K. V.; Levchenko, S. V.; Lin, C. Y.; Liu, F.; Livshits, E.; Lochan, R. C.; Luenser, A.; Manohar, P.; Manzer, S. F.; Mao, S.-P.; Mardirossian, N.; Marenich, A. V.; Maurer, S. A.; Mayhall, N. J.; Oana, C. M.; Olivares-Amaya, R.; O'Neill, D. P.; Parkhill, J. A.; Perrine, T. M.; Peverati, R.; Pieniazek, P. A.; Prociuk, A.; Rehn, D. R.; Rosta, E.; Russ, N. J.; Sergueev, N.; Sharada, S. M.; Sharmaa, S.; Small, D. W.; Sodt, A.; Stein, T.; Stück, D.; Su, Y.-C.; Thom, A. J. W.; Tsuchimochi, T.; Vogt, L.; Vydrov, O.; Wang, T.; Watson, M. A.; Wenzel, J.; White, A.; Williams, C. F.; Vanovschi, V.; Yeganeh, S.; Yost, S. R.; You, Z.-Q.; Zhang, I. Y.; Zhang, X.; Zhou, Y.; Brooks, B. R.; Chan, G. K. L.; Chipman, D. M.; Cramer, C. J.; Goddard III, W. A.; Gordon, M. S.; Hehre, W. J.; Klamt, A.; Schaefer III, H. F.; Schmidt, M. W.; Sherrill, C. D.; Truhlar, D. G.; Warshel, A.; Xua, X.; Aspuru-Guzik, A.; Baer, R.; Bell, A. T.; Besley, N. A.; Chai, J.-D.; Dreuw, A.; Dunietz, B. D.; Furlani, T. R.; Gwaltney, S. R.; Hsu, C.-P.; Jung, Y.; Kong, J.; Lambrecht, D. S.; Liang, W.; Ochsenfeld, C.; Rassolov, V. A.; Slipchenko, L. V.; Subotnik, J. E.; Van Voorhis, T.; Herbert, J. M.; Krylov, A. I.; Gill, P. M. W.; Head-Gordon, M. Advance in molecular quantum chemistry contained in the Q-Chem 4 program package. *Mol. Phys.* **2015**, *113*, 184–215.

(27) Yokota, R.; Barba, L. A. A tuned and scalable fast multipole method as a preeminent algorithm for exascale systems. *Int. J. High Perform. Comput. Appl.* **2012**, *26*, 337–346.

(28) Thole, B. T. Molecular polarizabilities calculated with a modified dipole interaction. *Chem. Phys.* **1981**, *59*, 341–350.

(29) van Duijnen, P. T.; Swart, M. Molecular and Atomic Polarizabilities: Thole's Model Revisited. *J. Phys. Chem. A* **1998**, *102*, 2399–2407.

(30) Meurer, A.; Smith, C. P.; Paprocki, M.; Čertík, O.; Kirpichev, S. B.; Rocklin, M.; Kumar, A.; Ivanov, S.; Moore, J. K.; Singh, S.; Rathnayake, T.; Vig, S.; Granger, B. E.; Muller, R. P.; Bonazzi, F.; Gupta, H.; Vats, S.; Johansson, F.; Pedregosa, F.; Curry, M. J.; Terrel, A. R.; Roučka, v.; Saboo, A.; Fernando, I.; Kulal, S.; Cimrman, R.; Scopatz, A. SymPy: symbolic computing in Python. *PeerJ Comput. Sci.* **2017**, *3*, e103.

(31) Stone, A. *The Theory of Intermolecular Forces*; OUP Oxford, 2013.

(32) Dykstra, C. E. Efficient calculation of electrically based intermolecular potentials of weakly bonded clusters. *J. Comput. Chem.* **1988**, *9*, 476–487.

(33) Peterson, P. F2PY: a tool for connecting Fortran and Python programs. *Int. J. Comput. Sci. Eng.* **2009**, *4*, 296–305.

(34) Slipchenko, L. V. Solvation of the excited states of chromophores in polarizable environment: Orbital relaxation versus polarization. *J. Phys. Chem. A* **2010**, *114*, 8824–8830.

(35) Martinez, L.; Andrade, R.; Birgin, E. G.; Martínez, J. M. PACKMOL: A package for building initial configurations for molecular dynamics simulations. *J. Comput. Chem.* **2009**, *30*, 2157–2164.

(36) Gagliardi, L.; Lindh, R.; Karlström, G. Local properties of quantum chemical systems: the LoProp approach. *J. Chem. Phys.* **2004**, *121*, 4494–4500.

(37) Vahtras, O. LoProp for Dalton. 2014.

(38) Yanai, T.; Tew, D. P.; Handy, N. C. A New Hybrid Exchange–Correlation Functional Using the Coulomb-Attenuating Method (CAM-B3LYP). *Chem. Phys. Lett.* **2004**, *393*, 51–57.

(39) Hehre, W. J.; Ditchfield, R.; Pople, J. A. Self—Consistent Molecular Orbital Methods.

XII. Further Extensions of Gaussian—Type Basis Sets for Use in Molecular Orbital Studies of Organic Molecules. *J. Chem. Phys.* **1972**, *56*, 2257–2261.

(40) Olsen, J. M. H. PyFraME: Python framework for Fragment-based Multiscale Embedding. 2020; `https://doi.org/10.5281/zenodo.3820471`.

(41) Hirata, S.; Head-Gordon, M. Time-dependent density functional theory within the Tamm–Dancoff approximation. *Chem. Phys. Lett.* **1999**, *314*, 291–299.

(42) Herbst, M. F.; Scheurer, M.; Fransson, T.; Rehn, D. R.; Dreuw, A. adcc: A versatile toolkit for rapid development of algebraic-diagrammatic construction methods. *Wires Comput Mol Sci* **2020**, *10*, e1462.

(43) Scheurer, M.; Herbst, M. F.; Reinholdt, P.; Olsen, J. M. H.; Dreuw, A.; Kongsted, J. Polarizable Embedding Combined with the Algebraic Diagrammatic Construction: Tackling Excited States in Biomolecular Systems. *J. Chem. Theory Comput.* **2018**, *14*, 4870–4883.

(44) Reinholdt, P.; Kjellgren, E. R.; Steinmann, C.; Olsen, J. M. H. Cost-effective potential for accurate polarizable embedding calculations in protein environments. *J. Chem. Theory Comput.* **2019**, *16*, 1162–1174.

(45) Beerepoot, M. T.; Steindal, A. H.; List, N. H.; Kongsted, J.; Olsen, J. M. H. Averaged solvent embedding potential parameters for multiscale modeling of molecular properties. *J. Chem. Theory Comput.* **2016**, *12*, 1684–1695.

(46) List, N. H.; Jensen, H. J. A.; Kongsted, J. Local electric fields and molecular properties in heterogeneous environments through polarizable embedding. *Phys. Chem. Chem. Phys.* **2016**, *18*, 10070–10080.

(47) Dreuw, A.; Wormit, M. The algebraic diagrammatic construction scheme for the polar-

ization propagator for the calculation of excited states. *Wires Comput Mol Sci* **2015**, *5*, 82–95.

(48) Loos, P.-F.; Scemama, A.; Blondel, A.; Garniron, Y.; Caffarel, M.; Jacquemin, D. A mountaineering strategy to excited states: Highly accurate reference energies and benchmarks. *J. Chem. Theory Comput.* **2018**, *14*, 4360–4379.

(49) Suellen, C.; Freitas, R. G.; Loos, P.-F.; Jacquemin, D. Cross-comparisons between experiment, TD-DFT, CC, and ADC for transition energies. *J. Chem. Theory Comput.* **2019**, *15*, 4581–4590.

Graphical TOC Entry