Received: 24 June 2019 | Revised: 1 February 2021 | Accepted: 16 February 2021

DOI: 10.1002/wcms.1528

SOFTWARE FOCUS

WIREs COMPUTATIONAL MOLECULAR SCIENCE

WILEY

Gator: A Python-driven program for spectroscopy simulations using correlated wave functions

Dirk R. Rehn[1] | Zilvinas Rinkevicius[2,3] | Michael F. Herbst[4] | Xin Li[2] | Maximilian Scheurer[1] | Manuel Brand[2] | Adrian L. Dempwolff[1] | Iulia E. Brumboiu[2,5] | Thomas Fransson[1,6] | Andreas Dreuw[1] | Patrick Norman[2]

[1]Interdisciplinary Center for Scientific Computing, Ruprecht-Karls University, Heidelberg, Germany

[2]Department of Theoretical Chemistry and Biology, School of Engineering Sciences in Chemistry, Biotechnology and Health, KTH Royal Institute of Technology, Stockholm, Sweden

[3]Department of Physics, Faculty of Mathematics and Natural Sciences, Kaunas University of Technology, Kaunas, Lithuania

[4]CERMICS, École des Ponts ParisTech, 6 & 8 avenue Blaise Pascal, 77455 Marne-la-Vallée, France and Inria Paris, Paris Cedex, France

[5]Department of Chemistry, Korea Advanced Institute for Science and Technology, Daejeon, South Korea

[6]Fysikum, Stockholm University, Albanova, Stockholm, Sweden

Correspondence
Andreas Dreuw, Interdisciplinary Center for Scientific Computing, Ruprecht-Karls University, Im Neuenheimer Feld 205, 69120 Heidelberg, Germany.
Email: dreuw@uni-heidelberg.de

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Number: GSC220; Horizon 2020 Framework Programme, Grant/Award Number: 765739; Knut och Alice Wallenbergs Stiftelse, Grant/Award Number: KAW-2013.0020; Vetenskapsrådet, Grant/Award Numbers: 2017-06419, 2017-00356, 2018-4343; European Research Council, Grant/Award Number: 810367

Edited by: Anna Krylov, Associate Editor

Abstract

The Gator program has been developed for computational spectroscopy and calculations of molecular properties using real and complex propagators at the correlated level of wave function theory. Currently, the focus lies on methods based on the algebraic diagrammatic construction (ADC) scheme up to the third order of perturbation theory. An auxiliary Fock matrix-driven implementation of the second-order ADC method for excitation energies has been realized with an underlying hybrid MPI/OpenMP parallelization scheme suitable for execution in high-performance computing cluster environments. With a modular and object-oriented program structure written in a Python/C++ layered fashion, Gator additionally enables time-efficient prototyping of novel scientific approaches, as well as interactive notebook-driven training of students in quantum chemistry.

This article is categorized under:

- Computer and Information Science > Computer Algorithms and Programming
- Electronic Structure Theory > Ab Initio Electronic Structure Methods
- Software > Quantum Chemistry

KEYWORDS

computational spectroscopy, electronic structure theory, propagator theory, response theory

1 | INTRODUCTION

Quantum chemical calculations have become a corner stone for fundamental and applied research in molecular sciences. Simulations of spectroscopies and predictions of molecular properties are made available through various formulations of response theory and allow for the revelation of molecular structure–function relationships that are indispensable for experimental spectrum characterization and rational material design.[1, 2] However, any attempt to compute molecular spectra and/or properties remains a compromise between accuracy and applicability—particularly so for large molecular systems embedded in complex environments.[3] While reliable quantum chemical methods exist to address the electronic ground state, electronic structure theory as a whole and efficient computer programs in particular are lagging behind the current needs in the field of computational spectroscopy.[4] A wide range of semiempirical and density functional-based methods exist, which are applicable to a wide range of large molecular systems,[5, 6] but they are not reliable and predictive without extensive benchmarking and/or comparisons with experiments. This turns out to be an increasingly difficult way forward with growing system complexity as suitable experiments are often missing and highly accurate methods are computationally unaffordable.[7] Therefore, the development of wave function-based *ab initio* methods and the provision of efficient computer program implementations are in high demand, not least because such methods are systematically improvable and usually possess predictive power. In contrast to semiempirical and density functional-based approximations, the physical effects contained in the ab initio Hamiltonian are a priori well defined, and from the outset, it is clear which properties and spectra can be described in a physically correct manner.

The focus of the Gator program lies on single-reference methods, currently using the algebraic diagrammatic construction (ADC) scheme for the polarization propagator up to the third order in the fluctuation potential[8–11] as implemented in the ADC connect (adcc) toolkit.[12] It had been inspired by the adcman module contained in Q-Chem[13, 14] but evolved into an independent, open-source hybrid Python/C++ module. Based on the capabilities of adcc, the Gator program enables the calculation of molecular properties using linear real and complex response functions. In addition, Møller–Plesset (MP) perturbation theory and ADC(2) have been implemented with efficient exploitation of high-performance computing (HPC) resources. This implementation is accomplished using a distributed auxiliary Fock matrix-driven tensor contraction scheme with hybrid MPI/OpenMP parallelization. Overall, the Gator program is designed to provide a flexible platform for (i) high-performance ab initio quantum chemistry, (ii) scientific extensions (methods and algorithms) with relative ease and quick turnover time, and (iii) educational training with a high degree of student interactivity. The available features of Gator are described in more detail in the corresponding subsection.

With partly different scientific objectives but an otherwise largely shared philosophy and intentions, similar program development efforts have recently been undertaken, for example: (i) the Psi4NumPy[15, 16] Python programming environment for facilitating the use of the self-consistent field (SCF) and post-Hartree–Fock (post-HF) kernels from the Psi4 program through NumPy[17]; (ii) the PySCF program platform[18, 19] for Python-based SCF and post-HF electronic structure theory calculations of finite and periodic systems; (iii) the molsturm[20] quantum chemistry framework, which provides a contraction-based and basis function-independent SCF, integrating readily with existing integral libraries and post-HF codes; and (iv) the Dalton Project[21] providing a platform written in Python for the interoperability of quantum chemical software projects. There exist several other efforts to provide massively parallel quantum chemistry programs, which do not necessarily exploit a Python-based modular program setup. For an overview, see Refs. 22, 23.

2 | PROGRAM STRUCTURE AND MODULARITY

2.1 | Algorithmic considerations

The performance-determining tasks in post-HF spectroscopy calculations are the atomic-to-molecular orbital integral transformations followed by the tensor operations defined in the iterative solution of the underlying response equations. Conventional implementations suited for shared-memory computers typically store the full tensor of transformed antisymmetrized two-electron integrals in memory. While this approach works well for calculations of molecular systems up to about 700 one-particle basis functions on standard computers, it quickly becomes unfeasible for larger systems due to a memory bottleneck. As a complement to a conventional integral transformation, we have also implemented an MPI/OpenMP parallel auxiliary Fock matrix-driven transformation routine. The basic idea of this distributed integral transformation is to express quantities on the molecular orbital (MO) basis (such as integrals and tensor contractions) in terms of auxiliary Fock-like matrices. This has previously been exploited in an atomic orbital-based formulation of the complete active space SCF method on graphical processing units.[24]

For electron-repulsion integrals in the MO basis, we use the simple relation

$$\langle ij|kl\rangle = \sum_{\gamma,\delta}^{N} C_{\gamma k} F^{ij}_{\gamma\delta} C_{\delta l} - C_{\gamma l} F^{ij}_{\gamma\delta} C_{\delta k}, \tag{1}$$

where

$$F^{ij}_{\gamma\delta} = \sum_{\alpha,\beta}^{N} D^{ij}_{\alpha\beta}(\alpha\gamma|\delta\beta); \quad D^{ij}_{\alpha\beta} = C_{\alpha i} C_{\beta j}. \tag{2}$$

Here, i, j, k, l and α, β, γ, δ refer to MO and atomic orbital (AO) indices, respectively, and N is the number of basis functions. The basic advantage is that it allows these quantities to be constructed incrementally, and by using a split MPI communicator, the work of constructing this set of auxiliary Fock matrices can be divided into the available cluster nodes. Any MO integral resulting from Equation (1) remains afterward on the node of its evaluation, being ready, for example, for the distributed calculation of the MP2 energy or ADC matrix–vector products.

For the ADC part, Gator relies on contraction-based iterative algorithms for solving response and eigenvalue equations as has been described previously for adcc.[12] In such schemes, the ADC matrix **M** is never explicitly constructed, but instead, the secular and response equations are solved by calculating matrix–vector products with sets of trial vectors $\{b_i\}$:

$$\sigma_i = \mathbf{M} b_i. \tag{3}$$

Formally, the ADC(2) vectors consist of a single-excitation part and a much larger double-excitation part, which can be folded into the space of single excitations.[25] We implemented this algorithm within the HPC-QC module of Gator as vectors in the single excitation space can be communicated between nodes at a negligible cost. The expressions for the ADC(2) matrix–vector product in Equation (3) have thus also been derived and implemented with the use of the auxiliary Fock matrix-driven MO transformation.

For smaller systems, the computational cost of the auxiliary Fock matrix-driven approach is not beneficial, and hence, both variants of integral transformation and calculations of σ-vectors are made available in the Gator program by VeloxChem.[26] With AO integral screening based on a combination of the Cauchy–Schwarz inequality and the density, the calculation of auxiliary Fock matrices scales somewhat better than N^3. As there are a total of N^2 pair indices (i, j), this results in a total scaling of less than N^5 for the auxiliary Fock matrix-driven integral transformation in Equation (1).

2.2 | Modular structure of Gator

The Gator program consists of three major modules as illustrated in Figure 1: (i) *HPC-QC* for MP2 and ADC(2) calculations in HPC environments, exploiting the Fock matrix-driven integral transformation outlined above; (ii) *Respondo* for the evaluation of real and complex linear response functions; and (iii) *ADC connect* (adcc)[12] for excited-state calculations based on the ADC scheme up to the third order. The Hartree–Fock reference state and the required one- and two-electron integrals are provided by the VeloxChem[26] program.

The modules of Gator are developed in a two-layered fashion. The upper layer is written in Python and includes the management of the hardware resources, input/output handling, result processing, and high-level parts of some algorithms. Numerical array operations are carried out using NumPy[17] and SciPy.[27] Parts of the lower program layer are written in C++, providing functionality for the evaluation of tensor contractions that, in adcc, are performed by libtensor.[28] The two layers communicate by means of the pybind11 library.[29]

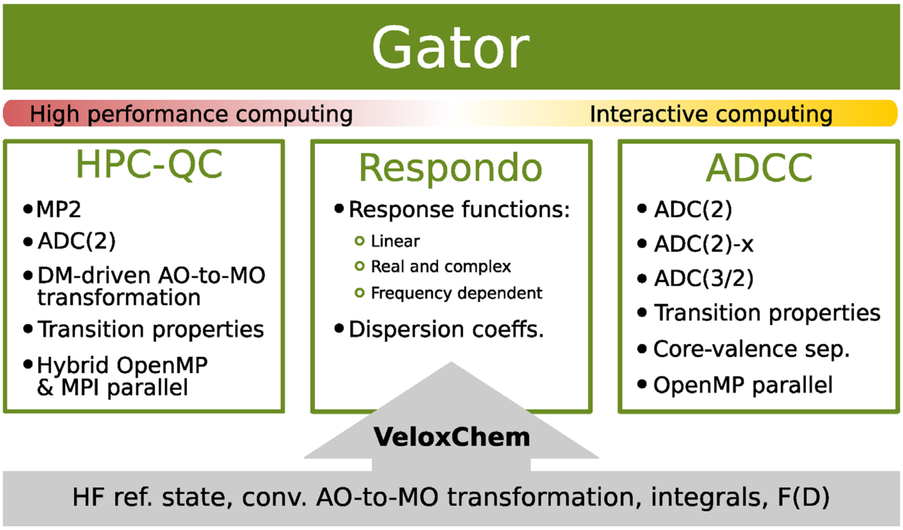


FIGURE 1 Gator program capabilities and module overview

2.3 | Implementation and efficiency of HPC-QC

In addition to the standard implementations of MP2 and ADC(2) available within the adcc module, we have implemented new and highly parallel versions of MP2 and ADC(2) that exploit the auxiliary Fock matrix-driven MO transformation within the HPC-QC module. VeloxChem is responsible for the construction of the auxiliary Fock matrices, including the underlying evaluation of two-electron integrals on the AO basis using a hybrid MPI/OpenMP parallelized module for efficient execution on HPC clusters. The module accepts, depending on memory limits, an arbitrary number of density matrices in a batch and handles those in parallel. With a *single* evaluation of the set of two-electron integrals, the corresponding *complete batch* of auxiliary Fock matrices is constructed. Such a multiauxiliary Fock matrix construction leads to additional costs for integral distribution. As a larger batch size comes at the cost of reduced OpenMP parallel efficiency in the integral evaluation, it may become necessary to balance the two forms of parallelism in the integral transformation depending on the size of the system and the adopted basis set.

In the same way, HPC-QC is *hybrid parallelized*. With regard to MO integral storage, a split MPI communicator is used across cluster nodes, and any given integral naturally resides in the memory of the node executing Equation (1) to enable usage of the aggregated memory of the available cluster nodes, as well as to minimize the need for communication. For example, the storage of an integral block $\langle oo||vv \rangle$ can be distributed over the occupied–occupied (oo) or the virtual–virtual (vv) pair indices. The distributed storage of MO integrals allows for a parallel implementation of ADC (2), where the tensor operations take place on the individual compute nodes that include a list of distributed pair indices and the corresponding auxiliary Fock matrices on the MO basis. The construction of matrix–vector products (σ-vectors) in ADC(2) is evaluated by looping over the distributed pair indices and contracting the corresponding auxiliary Fock matrices on the MO basis with the trial vectors of single excitations. Storage of the large vectors of double excitations is avoided by packing them into the space of single excitations.[25] As such, parallel ADC(2) calculations can be carried out efficiently on cluster nodes with only moderate amounts of memory.

In Figure 2, we show results for guanine oligomers. We investigated the scaling of the MO integral calculation compared to the construction of σ-vectors with respect to the number of contracted basis functions (gray curve in Figure 2). The MO integral calculation involves all those necessary at the level of ADC(2) theory. For guanine oligomers with the def2-SVP basis set, the MO integral calculation scales as $N^{4.9}$ in line with the discussion above. The evaluation of MO integrals therefore uses the majority of the computational time (core-hours) in the parallel implementation of ADC (2) as the construction of σ-vectors shows a lower scaling of $N^{3.8}$.

2.4 | Capabilities

Calculations using the ADC(2),[8] ADC(2)-x,[9] and ADC(3)[10] methods can be performed using the adcc module for shared-memory architectures. For spectrum calculations in the soft X-ray region, the core valence separation (CVS) is

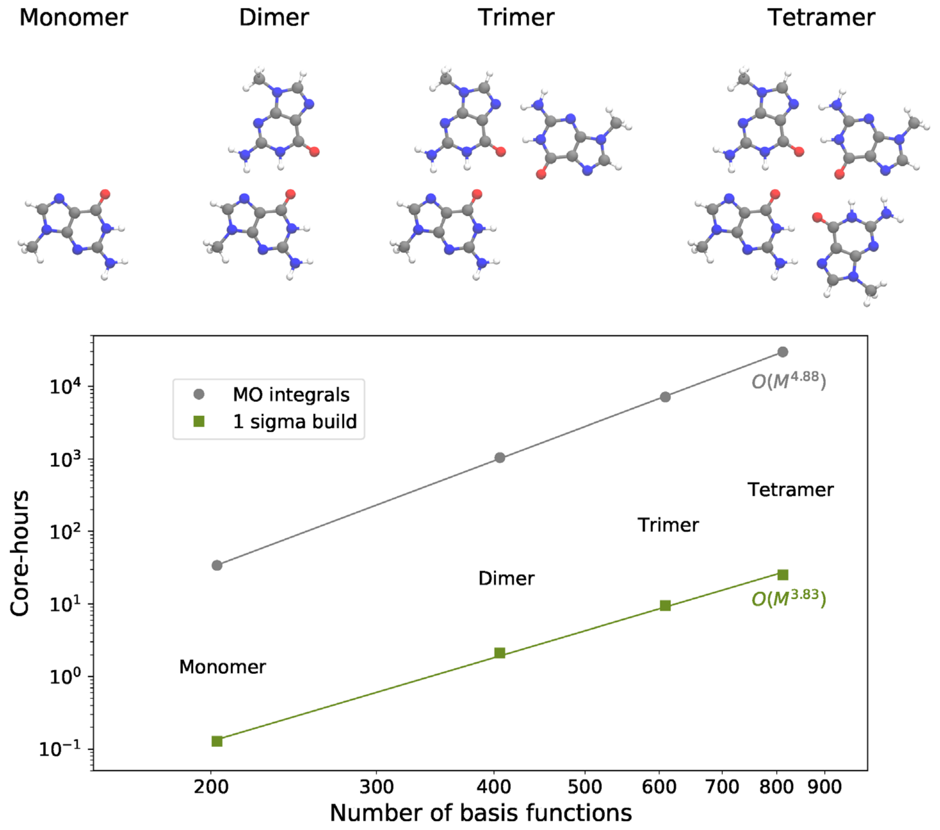


FIGURE 2 Scaling of the construction of molecular orbital (MO) integrals and ADC(2) σ-vectors with respect to the number of contracted basis functions. Results are obtained for guanine oligomers using the def2-SVP basis set (203 contracted basis functions per guanine). Calculations were performed on standard cluster nodes (Intel Xeon Haswell, 32 cores, 64 GB)

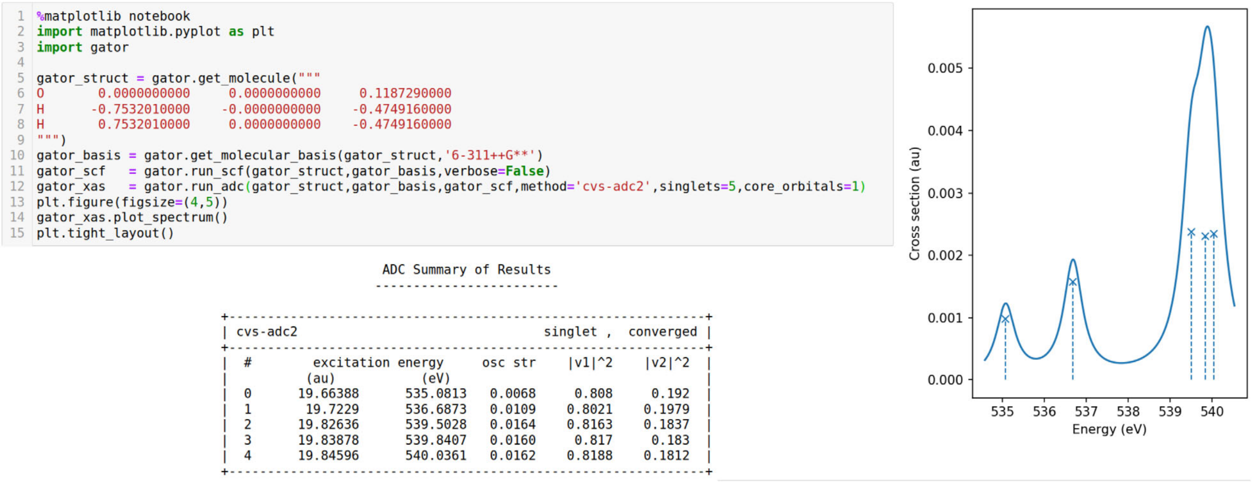
available.[30–32] All excited state and transition properties implemented in adcc are also available in the Gator program. Absorption cross sections and C_6 dispersion coefficients can be obtained through the complex polarization propagator (CPP) approach,[33–38] implemented in the Respondo library. Calculations with explicit solvents are currently available using the polarizable embedding (PE) model[39–41] through the CPPE library[42] as interfaced with VeloxChem.

MP2 energies and ADC(2) excitation energies can additionally be obtained by employing the HPC-QC module using the distributed aggregated memory on HPC clusters.

3 | HOW TO USE GATOR

To illustrate the modular, low-barrier nature of our Python-based environment, we outline the calculation of the X-ray absorption spectrum of water in Figure 3. This calculation was performed with the Gator program in a Jupyter notebook,[43] which is an open, web-based interactive environment of increasing popularity. By dividing the code into multiple blocks, Jupyter enables the calculation of individual code sections, keeping results in local memory. This simplifies rewriting and recomputing code snippets, such that the user does not need to recalculate all quantities after a modification is performed. This approach allows for an interactive path to programming, analysis, and education. In Gator, we take advantage of these concepts by combining ab initio computational spectroscopy with intuitive analysis and plotting routines through NumPy[17] and Matplotlib.[44] Molecular systems can be investigated in an iterative manner by varying parameters such as basis sets, convergence criteria, numerical solvers, and convolution/broadening functions.

Gator also offers traditional job submission using input files. An example input file is shown in Figure 4.



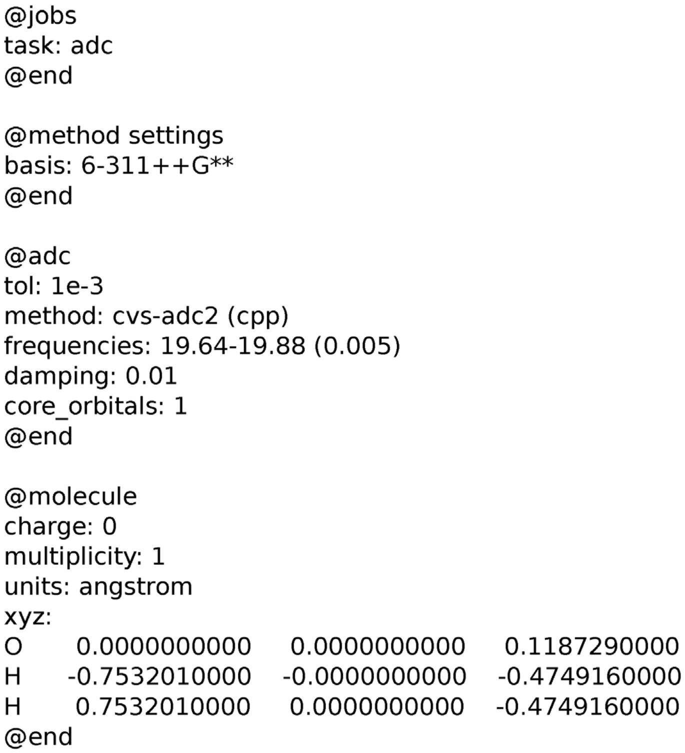
```
%matplotlib notebook
import matplotlib.pyplot as plt
import gator

gator_struct = gator.get_molecule("""
O       0.0000000000     0.0000000000     0.1187290000
H      -0.7532010000    -0.0000000000    -0.4749160000
H       0.7532010000     0.0000000000    -0.4749160000
""")
gator_basis = gator.get_molecular_basis(gator_struct,'6-311++G**')
gator_scf   = gator.run_scf(gator_struct,gator_basis,verbose=False)
gator_xas   = gator.run_adc(gator_struct,gator_basis,gator_scf,method='cvs-adc2',singlets=5,core_orbitals=1)
plt.figure(figsize=(4,5))
gator_xas.plot_spectrum()
plt.tight_layout()
```

```
                    ADC Summary of Results
                   -----------------------

+---------------------------------------------------------------+
| cvs-adc2                                  singlet ,  converged |
+---------------------------------------------------------------+
|  #       excitation energy     osc str    |v1|^2    |v2|^2   |
|         (au)           (eV)                                    |
|  0     19.66388      535.0813   0.0068     0.808     0.192   |
|  1      19.7229      536.6873   0.0109    0.8021    0.1979   |
|  2     19.82636      539.5028   0.0164    0.8163    0.1837   |
|  3     19.83878      539.8407   0.0160     0.817     0.183   |
|  4     19.84596      540.0361   0.0162    0.8188    0.1812   |
+---------------------------------------------------------------+
```

FIGURE 3 Gator notebook mode: The notebook input for the calculation of the X-ray absorption spectrum of water (upper left) and program output (lower left), focusing only on information for the requested five excited states. The resulting spectrum plot is shown on the right



```
@jobs
task: adc
@end

@method settings
basis: 6-311++G**
@end

@adc
tol: 1e-3
method: cvs-adc2 (cpp)
frequencies: 19.64-19.88 (0.005)
damping: 0.01
core_orbitals: 1
@end

@molecule
charge: 0
multiplicity: 1
units: angstrom
xyz:
O      0.0000000000     0.0000000000     0.1187290000
H     -0.7532010000    -0.0000000000    -0.4749160000
H      0.7532010000     0.0000000000    -0.4749160000
@end
```

FIGURE 4 Gator program input file for performing the core valence separation-complex polarization propagator-algebraic diagrammatic construction (CVS-CPP-ADC(2)) spectrum calculation shown in Figure 3

4 | NUMERICAL EXAMPLES

Within the electric dipole approximation, the absorption cross section of a randomly oriented molecular system is obtained from the isotropic average of the complex polarizability[33, 35, 37]:

$$\sigma(\omega) = \frac{\omega}{c\epsilon_0}\mathrm{Im}[\bar{\alpha}(\omega)], \tag{4}$$

allowing for the calculation of an absorption spectrum at arbitrary energy intervals by solving the associated linear response equations over a range of frequencies. To illustrate the capabilities of the Gator program, we computed the UV/vis and X-ray absorption cross sections, the static polarizability, and the polarizability at imaginary frequencies, yielding C_6 dispersion coefficients[38] for furan and thiophene as shown in Figure 5. A damping term $\gamma = 0.24$ eV was used for the calculation of absorption cross sections. For comparison, the excitation energies and intensities obtained from explicit calculations of 25 eigenstates are convoluted using a Lorentzian function with the same broadening parameter. All property calculations were performed using the 6-311++G**[45] basis set. The structures of the molecules were optimized at the B3LYP[46]/cc-pVTZ[47] level of theory using the Q-Chem 5.1 program.[14]

With imaginary frequency arguments, we obtain a smooth monotonous polarizability as illustrated in the left panels of Figure 5, resulting in an isotropic static polarizability ($\bar{\alpha}_0$) and a C_6 coefficient of 47.07 and 906.3 a.u. for furan and 62.34 and 1438 a.u. for thiophene. Experimental static polarizabilities are reported to be 48.59 and 65.18 a.u. for furan and thiophene, respectively—see Ref. 51 and references therein.

For the calculation of valence excitation spectra, Gator offers the possibility to either solve explicitly for eigenstates or to scan over relevant frequency intervals using damped response theory. Using a Lorentzian broadening for the transitions in the first approach will result in a spectrum that is practically identical to that obtained in the second approach, as illustrated in the middle and right panels of Figure 5. The linear absorption cross section obtained from (CVS-)CPP-ADC and the broadened eigenvalue results agree perfectly for lower energies. However, for higher energies, differences between the two curves occur because the resolved 25 eigenstates are not sufficient to obtain the full spectrum window.

The carbon *K*-edge spectra of furan and thiophene obtained with CVS-ADC(2) are given together with the experimental gas-phase spectra as insets in the right panels of Figure 5. Here, we include the first use of CVS-CPP-ADC, which has been developed and implemented in Gator. The use of the CVS approximation in addition to CPP here serves multiple purposes: (i) reduced matrix dimensions and a resulting decrease in computational effort; (ii) removal of spurious valence–continuum features that may contaminate the resolved spectrum window; and (iii) reducing difficulties in converging pure CPP-ADC calculations in the energy region of core excitations. Convergence issues have also been observed for CPP calculations of core excitations in the context of coupled cluster theory, and we note that the use of the CVS scheme introduces only a minor scalar shift in absolute energies. The overall discrepancy with respect to experiment amounts to a blue-shift of ~0.5 eV.[52]

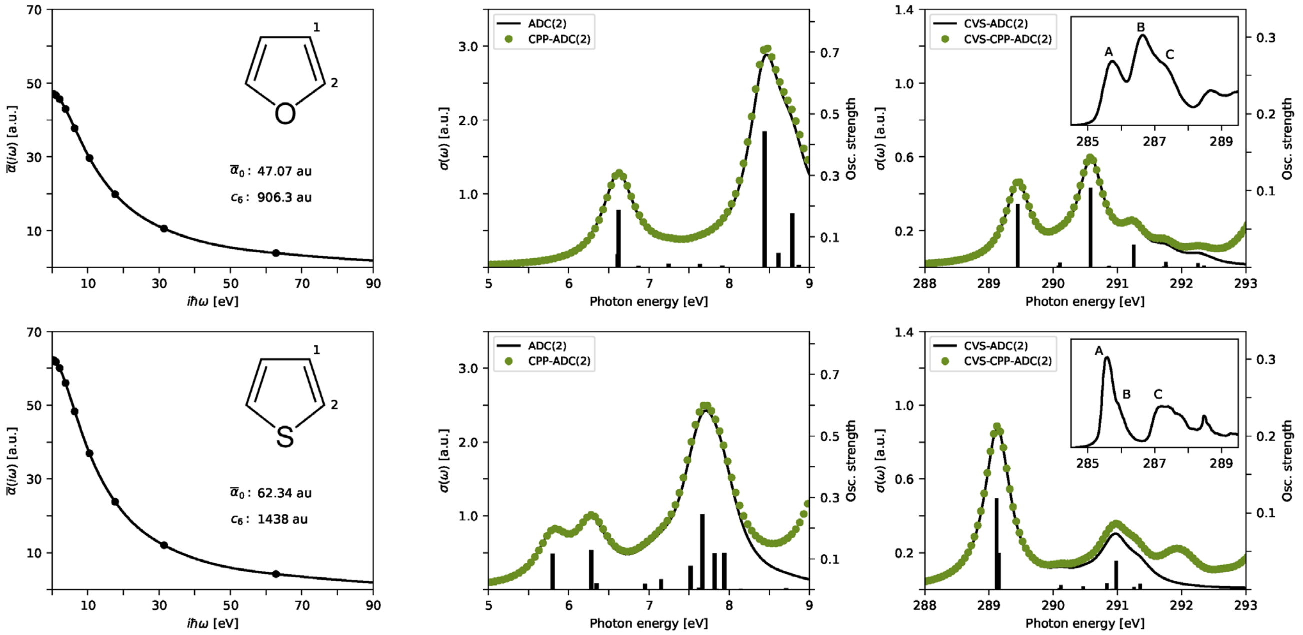


FIGURE 5 Properties of furan (top) and thiophene (bottom): (left) isotropic polarizability at imaginary frequencies, $\bar{\alpha}(i\omega)$, along with the resulting C_6 dispersion coefficient; (mid) UV absorption cross section, $\sigma(\omega)$, and oscillator strengths; (right) X-ray absorption cross section, $\sigma(\omega)$, and oscillator strengths at the near carbon *K*-edge together with insets showing the experimental spectra[48, 49] digitized with WebPlotDigitizer[50]

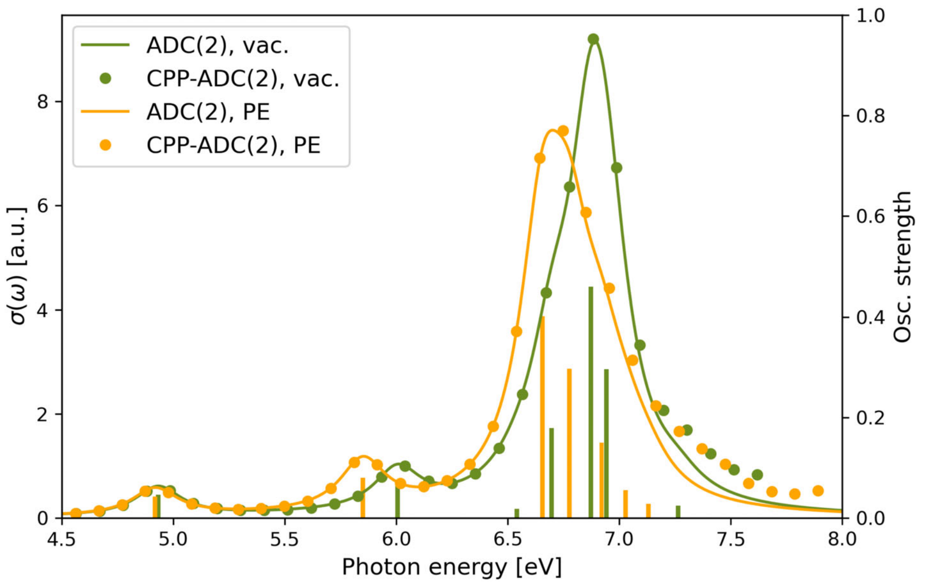


FIGURE 6 UV/vis spectra of noradrenaline in vacuum and solution. Continuous lines show the convoluted stick spectra, obtained from a Lorentzian function with $\gamma = 0.124$ eV. Dots depict the one-photon absorption cross section $\sigma(\omega)$ computed with complex polarization propagator (CPP)

To demonstrate the capabilities of Gator in computing the absorption spectra of solvated molecules at the ADC (2) level of theory, the spectrum of noradrenaline has been computed in the gas phase and in the aqueous solution using the standard eigenvalue solver, as well as the CPP approach. The vacuum structure of noradrenaline was optimized at the CAM-B3LYP/def2-svp[53, 54] level of theory as implemented in ORCA 4,[55] which served as input for the subsequent calculation of the seven energetically lowest singlet excited states at the ADC(2)/cc-pVDZ level of theory. Then, CPP-ADC(2) calculations were carried out on the frequency range obtained via the regular ADC(2) approach with a spacing of 0.1 eV, yielding the one-photon absorption cross section for each frequency; see Figure 6.

To model the UV/vis spectrum of noradrenaline in solution, it was placed in a box of size 48 × 44 × 43 Å^3 of water molecules using packmol,[56] and molecular dynamics (MD) simulations were carried out with NAMD 2.12[57] using the CHARMM 36 force field[58] and parameters for noradrenaline obtained with CGenFF.[59] The system was equilibrated for 5 ns at 310 K and a pressure of 1 atm (NPT ensemble). A subsequent hybrid QM/MM minimization was performed with NAMD,[60] where noradrenaline represented the QM region, and the MM region was assigned to the solvent. The QM computations were run at the CAM-B3LYP/def2-SVP level of theory with ORCA 4 using an electrostatic embedding scheme. The final snapshot of the quantum mechanics/molecular mechanics (QM/MM) minimization was extracted and parameterized using PyFramE[61] with a standard potential model. In the spectrum calculation, the solvent effects are modeled self-consistently with the PE model applied for the Hartree–Fock reference state.[39–41] The absorption spectra of noradrenaline in vacuum and solution are shown in Figure 6, where we note that energies in the latter calculation are *not* perturbatively corrected.[41]

5 | GATOR FOR TRAINING AND EDUCATION

The Gator program is a convenient platform for undergraduate training and education in computational and theoretical chemistry with students ranging from the introductory bachelor to the advanced master level. It offers insight into the underlying general theory of quantum chemistry and the adopted electronic structure theory methods with direct and concrete access to the intermediate quantities (tensor elements and wave functions) that are addressed in the workflow of quantum chemical calculations. This is accomplished by interfacing the Gator program with the Jupyter notebook as described above. This allows the student to steer, control, and analyze the course of a desired quantum chemical calculation in a modular manner. Students who are initially not familiar with programming are naturally motivated to overcome barriers involved with entering the field of theory and program development. With basic knowledge of Python programming, it is thus possible for students to design and implement key algorithms such as, for example, the SCF optimization of electron densities or the response theory calculations of excitation energies. Following this philosophy, it is possible to incorporate such practical exercises into existing courses, which has been successfully demonstrated already by the Psi4[16] community for example. The deepened algorithmic understanding gained by the students will be advantageous when entering the research level in this field.

6 | OUTLOOK

The Gator platform will undergo development following several major lines: (i) The HPC-QC module will be extended to include implementations for the calculation of transition and excited-state properties, as well as the inclusion of external one-particle potentials exploiting the intermediate state representation.[62] (ii) The complex linear response function will be implemented in the HPC-QC module. (iii) The capabilities of the Respondo library will be further extended by including higher-order response functions to allow for simulations of advanced linear and nonlinear spectroscopies. (iv) Quantum chemical methods other than the ADC approaches will be included with the primary aim at approximate coupled cluster theory of second order (CC2)[63] and also equation-of-motion and linear-response coupled cluster methods with singles and doubles.[64–66] (v) Modular libraries for environment models other than the PE model are under development. (vi) The engine of the HPC-QC module, that is, the efficient construction of the large number of auxiliary Fock matrices on the AO basis, will be adapted for execution on heterogeneous cluster nodes with varying combinations of central processing units and graphics processing units. These extensions will be realized as developments of the already existing modules (i)–(iii), while (iv) will require the development of a new module of its own.

7 | SOFTWARE AVAILABILITY

Gator is freely available under the GPLv3 license. The source code can be found on GitHub (https://github.com/gator-program/gator). Input files and notebook examples can be found on the GitHub website. Furthermore, the package can be conveniently installed using the conda package manager (https://anaconda.com) by running the command conda install gator -c gator, which will also automatically install all other dependencies.

ACKNOWLEDGMENTS

Financial support is acknowledged from the European Commission in the form of the ITN titled "*Computational Spectroscopy in Natural Sciences and Engineering (COSINE)*" (Grant No. 765739), the European Research Council under the Horizon 2020 program (Grant No. 810367), the Swedish Research Council (Grant Nos. 2018-4343, 2017-00356, and 2017-06419), the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (Grant No. GSC220), and the Swedish e-Science Research Centre, as well as a fellowship from Heidelberg University for P.N. to become a visiting professor at the Interdisciplinary Center for Scientific Computing, and computational resources were provided by the Swedish National Infrastructure for Computing. Fruitful discussions with Evgeny Epifanovsky and Anna I. Krylov are gratefully acknowledged.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

AUTHOR CONTRIBUTIONS

Dirk Rehn: Conceptualization, software, writing-original draft. **Zilvinas Rinkevicius:** Conceptualization, software, writing-original draft. **Michael Herbst:** Conceptualization, investigation, software, writing-original draft. **Xin Li:** Investigation, software, writing-original draft. **Maximilian Scheurer:** Investigation, software, writing-original draft. **Manuel Brand:** Investigation, software, writing-original draft. **Adrian Dempwolff:** Investigation, software. **Iulia E. Brumboiu:** Investigation, software. **Thomas Fransson:** Investigation, software, writing-original draft. **Andreas Dreuw:** Conceptualization, funding acquisition, supervision, writing-original draft, writing-review & editing. **Patrick Norman:** Conceptualization, supervision, writing-original draft.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

RELATEDWIREs ARTICLES

The ORCA program system

Molpro: A general-purpose quantum chemistry program package

VeloxChem: A Python-driven density-functional theory program for spectroscopy simulations in high-performance computing environments

ORCID

Dirk R. Rehn https://orcid.org/0000-0001-9463-3914
Zilvinas Rinkevicius https://orcid.org/0000-0003-2729-0290
Michael F. Herbst https://orcid.org/0000-0003-0378-7921
Xin Li https://orcid.org/0000-0001-6508-8355
Maximilian Scheurer https://orcid.org/0000-0003-0592-3464
Manuel Brand https://orcid.org/0000-0003-3992-043X
Adrian L. Dempwolff https://orcid.org/0000-0003-1106-8716
Iulia E. Brumboiu https://orcid.org/0000-0003-1671-8298
Thomas Fransson https://orcid.org/0000-0002-3770-9780
Andreas Dreuw https://orcid.org/0000-0002-5862-5113
Patrick Norman https://orcid.org/0000-0002-1191-4954

REFERENCES

1. Norman P, Ruud K, Saue T. *Principles and practices of molecular properties*. Chichester, UK: John Wiley & Sons, Ltd; 2018.
2. Grunenberg J. *Computational spectroscopy: methods, experiments and applications*. Weinheim: Wiley VCH; 2010.
3. Harbach PHP, Dreuw A. The art of choosing the right quantum chemical excited-state method for large molecular systems. In: Comba P, editor. *Modelling of molecular properties*. Weinheim: Wiley VCH; 2011. p. 29–47.
4. Gonzalez L, Escudero D, Serrano-Andres L. Progress and challenges in the calculation of electronic excited states. *Chem Phys Chem*. 2011;*13*:28–51.
5. Holthausen M, Koch W. *A chemist's guide to density functional theory*. Weinheim: Wiley VCH; 2001.
6. Mardirossiana N, Head-Gordon M. Thirty years of density functional theory in computational chemistry: an overview and extensive assessment of 300 density functionals. *Mol Phys*. 2017;*115*:2315–2372.
7. Dreuw A. Quantum chemical methods for the investigation of photo-initiated processes in biological systems: theory and applications. *Chem Phys Chem*. 2006;*7*:2259–2274.
8. Schirmer J. Beyond the random-phase approximation: a new approximation scheme for the polarization propagator. *Phys Rev A*. 1982; *26*:2395–3416.
9. Trofimov AB, Schirmer J. An efficient polarization propagator approach to valence electron excitation spectra. *J Phys B At Mol Opt Phys*. 1995;*28*:2299–2324.
10. Harbach PH, Wormit M, Dreuw A. The third-order algebraic diagrammatic construction method (ADC(3)) for the polarization propagator for closed-shell molecules: efficient implementation and benchmarking. *J Chem Phys*. 2014;*141*(6):064113.
11. Dreuw A, Wormit M. The algebraic diagrammatic construction scheme for the polarization propagator for the calculation of excited states. *WIREs Comput Mol Sci*. 2015;*5*:82–95.
12. Herbst MF, Scheurer M, Fransson T, Rehn DR, Dreuw A. adcc: a versatile toolkit for rapid development of algebraic-diagrammatic construction methods. *WIREs Comput Mol Sci*. 2020;*10*:e1462.
13. Wormit M, Rehn DR, Harbach PHP, Wenzel J, Krauter CM, Epifanovsky E, et al. Investigating excited electronic states using the algebraic diagrammatic construction (ADC) approach of the polarisation propagator. *Mol Phys*. 2014;*112*:774–784.
14. Shao Y, Gan Z, Epifanovsky E, Gilbert ATB, Wormit M, Kussmann J, et al. Advance in molecular quantum chemistry contained in the Q-Chem 4 program package. *Mol Phys*. 2015;*113*:184–215.
15. Smith DGA, Burns LA, Sirianni DA, Nascimento DR, Kumar A, James AM, et al. Psi4NumPy: an interactive quantum chemistry programming environment for reference implementations and rapid development. *J Chem Theory Comput*. 2018;*14*(7):3504–3511.
16. Smith DGA, Burns LA, Simmonett AC, Parrish RM, Schieber MC, Galvelis R, et al. PSI4 1.4: open-source software for high-throughput quantum chemistry. *J Chem Phys*. 2020;*152*:184108.
17. van der Walt S, Chris Colbert S, Varoquaux G. The NumPy array: a structure for efficient numerical computation. *Comput Sci Eng*. 2011;*13*(2):22–30.
18. Sun Q, Berkelbach TC, Blunt NS, Booth GH, Guo S, Li Z, et al. PySCF: the Python-based simulations of chemistry framework. *WIREs Comput Mol Sci*. 2018;*8*(1):e1340.
19. Sun Q, Zhang X, Banerjee S, Bao P, Barbry M, Blunt NS, et al. Recent developments in the PySCF program package. *J Chem Phys*. 2020; *153*:024109.
20. Herbst MF, Dreuw A, Avery JE. Towards quantum-chemical method development for arbitrary basis functions. *J Chem Phys*. 2018;*149* (8):84106.

21. Olsen JMH, Reine S, Vahtras O, Kjellgren E, Reinholdt P, Hjorth Dundas KO, et al. Dalton project: a Python platform for molecular- and electronic-structure simulations of complex systems. *J Chem Phys*. 2020 jun;*152*(21):214115. https://doi.org/10.1063/1.5144298.
22. Gordon MS, Windus TL. Editorial: modern architectures and their impact on electronic structure theory. *Chem Rev*. 2020;*120*: 9015–9020.
23. Calvin JA, Peng C, Rishi V, Kumar A, Valeev EF. Many-body quantum chemistry on massively parallel computers. *Chem Rev*. 2020;*121*: 1203–1231.
24. Hohenstein EG, Luehr N, Ufimtsec IS, Martinez TJ. An atomic orbital-based formulation of the complete active space self-consistent field method on graphical processing units. *J Chem Phys*. 2015;*142*:224103.
25. Hättig C, Köhn A. Transition moments and excited-state first-order properties in the coupled-cluster model CC2 using the resolution-of-the-identity approximation. *J Chem Phys*. 2002;*117*:6939–6951.
26. Rinkevicius Z, Li X, Vahtras O, Ahmadzadeh K, Brand M, Ringholm M, et al. VeloxChem: a Python-driven density-functional theory program for spectroscopy simulations in high-performance computing environments. *WIREs Comput Mol Sci*. 2020;*10*(5):e1457.
27. Jones E, Oliphant T, Peterson P. SciPy: open source scientific tools for Python. 2001. http://www.scipy.org/. Accessed 14 February 2020.
28. Epifanovsky E, Wormit M, Kus T, Landau A, Zuev D, Khistyaev K, et al. New implementation of high-level correlated methods using a general block-tensor library for high-performance electronic structure calculations. *J Comput Chem*. 2013;*34*:2293–2309.
29. Jakob W, Rhinelander J, Moldovan D. pybind11 – seamless operability between C++11 and Python. 2017. https://github.com/pybind/pybind11. Accessed 14 February 2020.
30. Cederbaum LS, Domcke W, Schirmer J. Many-body theory of core holes. *Phys Rev A*. 1980;*22*:206–222.
31. Wenzel J, Wormit M, Dreuw A. Calculating core-level excitations and X-ray absorption spectra of medium-sized closed-shell molecules with the algebraic-diagrammatic construction scheme for the polarization propagator. *J Comput Chem*. 2014;*35*:1900–1915.
32. Wenzel J, Wormit M, Dreuw A. Calculating X-ray absorption spectra of open-shell molecules with the unrestricted algebraic-diagrammatic construction scheme for the polarization propagator. *J Chem Theory Comput*. 2014;*10*:4583–4598.
33. Norman P, Bishop DM, Jensen HJA, Oddershede J. Near-resonant absorption in the time-dependent self-consistent field and multi-configurational self-consistent field approximations. *J Chem Phys*. 2001;*115*:10323–10334.
34. Norman P, Jiemchooroj A, Sernelius BE. Polarization propagator calculations of the polarizability tensor at imaginary frequencies and long-range interactions for the noble gases and n-alkanes. *J Chem Phys*. 2003;*118*(20):9167–9174.
35. Norman P, Bishop DM, Jensen HJA, Oddershede J. Nonlinear response theory with relaxation: the first-order hyperpolarizability. *J Chem Phys*. 2005;*123*:194103.
36. Jiemchooroj A, Norman P, Sernelius BE. Complex polarization propagator method for calculation of dispersion coefficients of extended π-conjugated systems: the C_6 coefficients of polyacenes and C_{60}. *J Chem Phys*. 2005;*123*(12):124312.
37. Norman P. A perspective on nonresonant and resonant electronic response theory for time-dependent molecular properties. *Phys Chem Chem Phys*. 2011;*13*:20519–20535.
38. Fransson T, Rehn DR, Dreuw A, Norman P. Static polarizabilities and C_6 dispersion coefficients using the algebraic-diagrammatic construction scheme for the complex polarization propagator. *J Chem Phys*. 2017;*146*:094301.
39. Olsen JM, Aidas K, Kongsted J. Excited states in solution through polarizable embedding. *J Chem Theory Comput*. 2010;*6*(12):3721–3734.
40. Olsen JMH, Kongsted J. Molecular Properties through Polarizable Embedding, Advances in Quantum Chemistry, 2011;*61*:107–143.
41. Scheurer M, Herbst M, Reinholdt P, Olsen JM, Dreuw A, Kongsted J. Polarizable embedding combined with the algebraic diagrammatic construction: tackling excited states in biomolecular systems. *J Chem Theory Comput*. 2018;*14*(9):4870–4883.
42. Scheurer M, Reinholdt P, Kjellgren ER, Haugaard Olsen JM, Dreuw A, Kongsted J. CPPE: an open-source C++ and Python library for polarizable embedding. *J Chem Theory Comput*. 2019;*15*(11):6154–6163.
43. Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, et al.. Jupyter notebooks—a publishing format for reproducible computational workflows. 2016. https://jupyter.org/. Accessed 14 February 2020.
44. Hunter JD. Matplotlib: a 2D graphics environment. *Comput Sci Eng*. 2007;*9*(3):90–95.
45. Krishnan R, Binkley JS, Seeger R, Pople JA. Self-consistent molecular orbital methods. XX. A basis set for correlated wave functions. *J Chem Phys*. 1980;*72*:650–654.
46. Becke AD. Density-functional thermochemistry. III. The role of exact exchange. *J Chem Phys*. 1993;*98*:5648–5652.
47. Dunning TH. Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen. *J Chem Phys*. 1989;*90*:1007–1023.
48. Duflot D, Flament JP, Giuliani A, Heinesch J, Hubin-Franskin MJ. Core shell excitation of furan at the O1s and C1s edges: an experimental and ab initio study. *J Chem Phys*. 2003;*119*:8946–8955.
49. Grazioli C, Baseggio O, Stener M, Fronzoni G, de Simone M, Coreno M, et al. Study of the electronic structure of short chain oligothiophenes. *J Chem Phys*. 2017;*146*:054303.
50. Rohatgi A. WebPlotDigitizer. 2019. https://automeris.io/WebPlotDigitizer
51. Kamada K, Ueda M, Nagao H, Tawa K, Sugino T, Schmizu Y, et al. Molecular design for organic nonlinear optics: Polarizability and hyperpolarizabilities of furan homologues investigated by ab initio molecular orbital method. *J Phys Chem A*. 2000;*104*:4723–4734.
52. Herbst MF, Fransson T. Quantifying the error of the core-valence separation approximation. *J Chem Phys*. 2020;*153*:054114.
53. Yanai T, Tew DP, Handy NC. A new hybrid exchange–correlation functional using the coulomb-attenuating method (CAM-B3LYP). *Chem Phys Lett*. 2004;*393*(1–3):51–57.

54. Weigend F, Ahlrichs R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: design and assessment of accuracy. *Phys Chem Chem Phys*. 2005;*7*:3297–3305.
55. Neese F. Software update: the ORCA program system, version 4.0. *WIREs Comput Mol Sci*. 2018;*8*(1):e1327.
56. Martínez L, Andrade R, Birgin EG, Martínez JM. PACKMOL: a package for building initial configurations for molecular dynamics simulations. *J Comput Chem*. 2009;*30*(13):2157–2164.
57. Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, et al. Scalable molecular dynamics with NAMD. *J Comput Chem*. 2005;*26*(16):1781–1802.
58. Huang J, MacKerell AD Jr. CHARMM36 all-atom additive protein force field: validation based on comparison to NMR data. *J Comput Chem*. 2013;*34*(25):2135–2145.
59. Vanommeslaeghe K, Hatcher E, Acharya C, Kundu S, Zhong S, Shim J, et al. CHARMM general force field: a force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *J Comput Chem*. 2010;*31*(4):671–690.
60. Melo MCR, Bernardi RC, Rudack T, Scheurer M, Riplinger C, Phillips JC, et al. NAMD goes quantum: an integrative suite for hybrid simulations. *Nat Methods*. 2018;*15*(5):351–354.
61. Olsen JMH. PyFraME: Python tools for fragment-based multiscale embedding (version 0.1.0). 2018.
62. Schirmer J, Trofimov AB. Intermediate state representation approach to physical properties of electronically excited molecules. *J Chem Phys*. 2004;*120*:11449–11464.
63. Christiansen O, Koch H, Jørgensen P. The second-order approximate coupled cluster singles and doubles model CC2. *Chem Phys Lett*. 1995;*243*:409–418.
64. Stanton JF, Bartlett RJ. Equation of motion coupled-cluster method: a systematic biorthogonal approach to molecular excitation energies, transition probabilities, and excited state properties. *J Chem Phys*. 1993;*98*:7029–7039.
65. Koch H, Kobayashi R, Sanchez de Merás A, Jørgensen P. Calculation of size-intensive transition moments from the coupled cluster singles and doubles linear response function. *J Chem Phys*. 1994;*100*:4393–4400.
66. Kaliman IA, Krylov AI. New algorithm for tensor contractions on multi-core CPUs, GPUs, and accelerators enables CCSD and EOM-CCSD calculations with over 1000 basis functions on a single compute node. *J Comput Chem*. 2017;*38*:842–853.

How to cite this article: Rehn DR, Rinkevicius Z, Herbst MF, et al. Gator: A Python-driven program for spectroscopy simulations using correlated wave functions. *WIREs Comput Mol Sci*. 2021;11:e1528. https://doi.org/10.1002/wcms.1528