eChem: A notebook exploration of quantum chemistry

Thomas Fransson,[1] Mickael G. Delcey,[1] Iulia Emilia Brumboiu,[2] Manuel Hodecker,[1] Xin Li,[1] Zilvinas Rinkevicius,[1] Andreas Dreuw,[3] Young Min Rhee,[4] and Patrick Norman[1, a)]

[1)] *Division of Theoretical Chemistry and Biology, KTH Royal Institute of Technology, 106 91 Stockholm, Sweden*

[2)] *Faculty of Physics, Astronomy and Informatics, Nicolaus Copernicus University, 87-100 Toruń, Poland*

[3)] *Interdisciplinary Center for Scientific Computing, Heidelberg University, Im Neuenheimer Feld 205, 69120 Heidelberg, Germany*

[4)] *Department of Chemistry, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, Korea*

The eChem project features an e-book published as a web page (https://bit.ly/e-chem), collecting a repository of Jupyter notebooks developed for the dual purpose of explaining and exploring the underlying theory behind computational chemistry in a highly interactive manner as well as providing a tutorial-based presentation of the complex workflows needed to simulate embedded molecular systems of real biochemical and/or technical interest. For students ranging from beginners to advanced users, the eChem book is well suited for self-directed learning, and workshops led by experienced instructors for targeting student bodies with specific needs and interests can readily be formed from its components. The members of the eChem team are engaged in both education and research and as a mirroring activity, we develop the open-source software upon which this e-book is predominantly based. The overreaching vision and goal of our work is to provide a science- and education-enabling software platform for quantum molecular modeling on contemporary and future high-performance computing systems, with the resulting development and workflows now being documented in the eChem book.

I. INTRODUCTION

Theoretical chemistry has made tremendous progress in the past few decades and is today an indispensable tool in all fields of molecular science, exemplified by biochemistry and nanotechnology, where it can be employed to reveal the microscopic origins of functionality and interactions as well as to tune performance and guide synthesis. As such, computational chemistry has matured to a highly active field, capable of modeling a plethora of chemical processes. However, due to the complexity of the problems being addressed, as well as the multitude of available software packages, this may seem like a daunting task, both for new users and also experienced users facing a need to adopt computational methodologies outside their own toolbox. At the other end, the link between the equation and the computer implementation found in existing software can be vague at best. As such, more user-friendly software tools, interactive education, and instructive tutorials and workflows are important. For the educational aspects, we would argue that to reach deep levels of understanding, it is beneficial to blend in components of exploration by students themselves and, being computational scientists, we advocate the use of an interactive framework such as Jupyter Notebook.[1] With these modern, flexible software packages and frameworks, we are more capable than ever of providing a blended learning platform for interactive, easily adaptable, computational workflows and deep explorations into the underlying theory. In this way, students can create their own algorithms and solutions for the physical processes at hand, reaching an understanding in line with the quip:

> *What I cannot create, I do not understand.*
> —Richard Feynman

In this article we present the eChem project, which seeks to do precisely that. This project serves a dual task, both bringing down the sometimes opaque theory of quantum chemistry to a more concrete level, including explicit computations alongside derivation of important theories in computational chemistry, as well as by including more advanced tutorials showing various aspects of state-of-the-art modeling of chemical processes, including explicit instructions in how to consider theses processes. As such, the project is aimed for a large width of readership, ranging from undergraduate students just entering their first course in theoretical chemistry, to experienced researchers who need to freshen up aspects of theory and modeling, or who are moving into a field of study they are not yet that familiar with.

The framework we have chosen to use is Jupyter Notebook,[1] which can be used to build web pages through the Jupyter Book project.[2] Jupyter is an increasingly popular interactive web tool[3] which can combine code, data collection, text, calculations, and more. We use the Python version of it, which has as predecessor the IPython project,[4] which Nature placed on a list of ten codes that transformed science,[5] due to the ease of conducting, communicating, and teaching science. Jupyter has indeed been used in a number of influential studies, including contributing to the discovery of gravitational waves in 2016,[6] and the first imaging of a black hole in 2019.[7] It has been noted that Jupyter has seen a massive

[a)] Electronic mail: panor@kth.se

growth, with the number of public notebooks increasing from 200,000 in 2015, to over 10,000,000 in 2021.[3,5] The software can easily be installed on any operating system, and is run through a web interface, with a kernel running either on a local computer or through a cluster- or cloud-based service. A notebook then consists of any number of cells with text or executable code, which can be run in any order. This gives a large flexibility for exploration, explanation, and execution, although care needs to be taken as there is always a risk that the non-linear use of blocks can make the code messy.[3,8,9]

The educational benefits of interactive notebooks have been realized in many fields, with examples in, *e.g.*, bioscience and informatics,[10] and radiology physics.[11] In chemistry, specifically, notebooks have been developed on the topics of scientific computing,[12,13] analytical chemistry,[14] stochastic simulations of processes,[15] labs in physical chemistry,[16] machine learning,[17,18] molecular docking,[19] or to explain basic concepts like the hydrogen molecule,[20] the particle in a box,[21] reciprocal space,[22] and more. It has been noted that deeper insights are gained after using interactive notebooks,[21] and students have been seen to quickly adopt Jupyter notebooks also for other courses.[12] Notebooks can be implemented in a problem-based learning protocol,[23] and it is particularly interesting to be able to use research-grade computational chemistry software.[23] In addition, projects such as nbgrader simplify the use of notebooks for exercises by facilitating assignments.[24]

The eChem project uses a variety of quantum chemical software packages for providing the routines necessary to carry out the calculations. The primary packages are those developed by us, including VeloxChem,[25] MultiPsi, and Gator.[26] Some additional quantum chemical software routines from adcc[27] and PySCF[28,29] are also used, together with semiempirical methods from xTB.[30,31] Such semiempirical approaches are increasingly capable of providing structures and sampling of conformational spaces close to high-level quantum mechanical calculations, while requiring limited parameter fitting.[31–33] Routines from geomeTRIC,[34] and OpenMM[35] are used for structure relaxation and, respectively, for semi-classical or classical dynamics. For analysis and visualization, routines in NumPy,[36] matplotlib,[37] py3Dmol,[38,39] and k3d are used. These software suites are chosen for their performance and ease of use, as well as having a Python layer from which calculations can be run, enabling easy manipulation, exploration, and development. Furthermore, we favor free and open-source software packages, which enable wider adoption, improved user friendliness, flexibility, and a democratization of science.[40,41] The move to more interactive software modules and improving interoperability between software packages has been realized by many groups,[28,29,40,42–46] including program packages and projects with education in mind.[47–50] Projects such as Psi4NumPy also include a number of tutorials for explaining the underlying theory,[47] and thus have some additional overlap with the eChem project. Other frameworks include the Open Chemistry project,[41] which focuses on the interactive production, sharing, and visualization of quantum chemical data on the web, using Jupyter as a platform.

In the following, we describe the philosophy of the eChem project, considering its dual purpose of education and exploration. We then outline the current contents of the e-book, briefly describing what is contained in each part. We stress that we are here concerned with today's snapshot version of the book, but that it is a living document that is expected to change over time. For example, the chapters on molecular dynamics and polarizable embedding are knowingly incomplete and tutorials on photochemistry and other important fields are being developed. The use of this e-book for education is then discussed, including self-directed and workshop situations, as well as our experiences in teaching using eChem. In closing, we provide outlooks focused on the future expansion of the e-book, and general conclusions.

II. OUR PHILOSOPHY

The "e" in eChem can refer to *electronic* as well as *educational*. The goal of this project and our software development is to be both science- and education-enabling, providing a comprehensive and living repository of theory, workflows, and tutorials. The *living* descriptor means that the eChem web page is subject to constant changes, being updated with new and improved discussions of underlying theory, as well as workflows and illustrations of our most recent research efforts. Together with the ability of providing numerical examples and tutorials, this approach has major advantages over a more static printed book, an idea which was originally considered but subsequently discarded. This means that the book is never finished, and portions of it are in a rather preliminary state—this article thus focuses on the parts that are sufficiently polished at the time of writing.

The term science-enabling refers in this context to workflows and recommendations for a wide range of methodologies and applied studies, providing a very transparent overview of how we model a range of chemical processes. For this, we require software capable of considering dense 3D systems of large size (with stable and reliable solvers), excited states of diffuse character, time-efficient prototyping of novel methods, providing transparent exposure of data and data-structures to enable in-depth analysis, interaction with other software packages, and programs suitable for both laptops and top high performance computing (HPC) resources. The exploratory and flexible nature of a Jupyter framework here benefits novices and veterans in the field alike, enabling the use of novel computational workflows, interactive computations and analysis, and rapid development cycles. For instance, during the writing of the eChem book the authors appreciated how easily unusual workflows and examples could be implemented and visualized, resulting in many examples illustrated on the web page.

In the context of education-enabling, the notion of deep learning refers to taking each student's understanding of the subject matter to another (deeper) level. Our experience tells us that the process of implementing methods to solve fundamental equations is supremely efficient as a means to achieve that type of deeper learning and understanding, but only a small number of students are granted this opportunity as many core modules of scientific software were written a long time ago and have often been made obscure by code optimization. We believe that a core issue is that humans are not good at learning in abstract terms. We can get very far with a lecture or a textbook, but we will never build as much intuition about how a clock works as by simply breaking one apart and rebuilding it from scratch. This is exactly the aim of the eChem book, allowing a hands-on approach to computational chemistry. Together, we dismantle the black box that a computational chemistry code often seems to be, go through all the cogs and gears, and build back together some of the main computational methods of modern computational chemistry. Discussions of theory are thus interspersed with numerical examples, bringing the material beyond a more opaque discussion and into the realm of deep learning. This, coupled with the exploratory nature of notebooks and the ample opportunities for visualizing complex theory, benefits the interest in and deep understanding of complex subjects.

Finally, an important part of our philosophy is that research and education should, as far as possible, be freely and easily available as well as user-friendly. As such, we favor software packages that are free and open-source, easy to install with a common package managing tool, and offering interoperability at the Python level. We have chosen Jupyter Notebook as our vehicle for interactive computations. We note that, historically, such a choice has suffered from lower performance than running more specialized codes written in, for instance, Fortran through a batch queuing system, but developments made to combine Python modules with C++ routines for compute-intensive parts as well as frameworks such as JupyterHub to access powerful computational resources is constantly diminishing this gap. In this new landscape, we believe that the decrease in cognitive cost reached by adapting computational protocols to the sequentially operating (single-core) brain[51] rather than multi-tasking computers often outweighs the increase in wall time.

III. THE ECHEM BOOK

The title page of the eChem book is shown in Fig. 1. The book is composed of six parts with underlying chapters and sections, intertwining theory with numerical examples and analyses. The first part titled *Tutorials and workflows* focuses on practical calculation and analysis of spectra and properties, and the parts following this are more directed toward a discussion of the underlying theory. Notebook markdown cells with text and equations presenting the theory are intertwined with code cells implementing the said expressions. These code cells, which also perform insightful calculations with associated figure illustrations, are key to the deeper student learning, not least since they are made available for download and further curiosity-driven exploration. A simple example devoted to the concept of electron correlation is illustrated in Fig. 2.

The compilation of the many separate notebooks into one coherent and interlinked eChem book is based on the Jupyter Book project.[2] The entire stack of software needed to create the book can be installed as follows

```
conda env create -f echem.yml
```

where the `echem.yml` file, provided on the web page, conveniently defines all dependencies. Conda is an open-source package and environment management system that first needs to be installed and it runs under any of the three main operating systems: Windows, MacOS, and Linux. The Conda package manager gives access to the Anaconda repository of scientific software, featuring environment encapsulation.

A. Tutorials and workflows

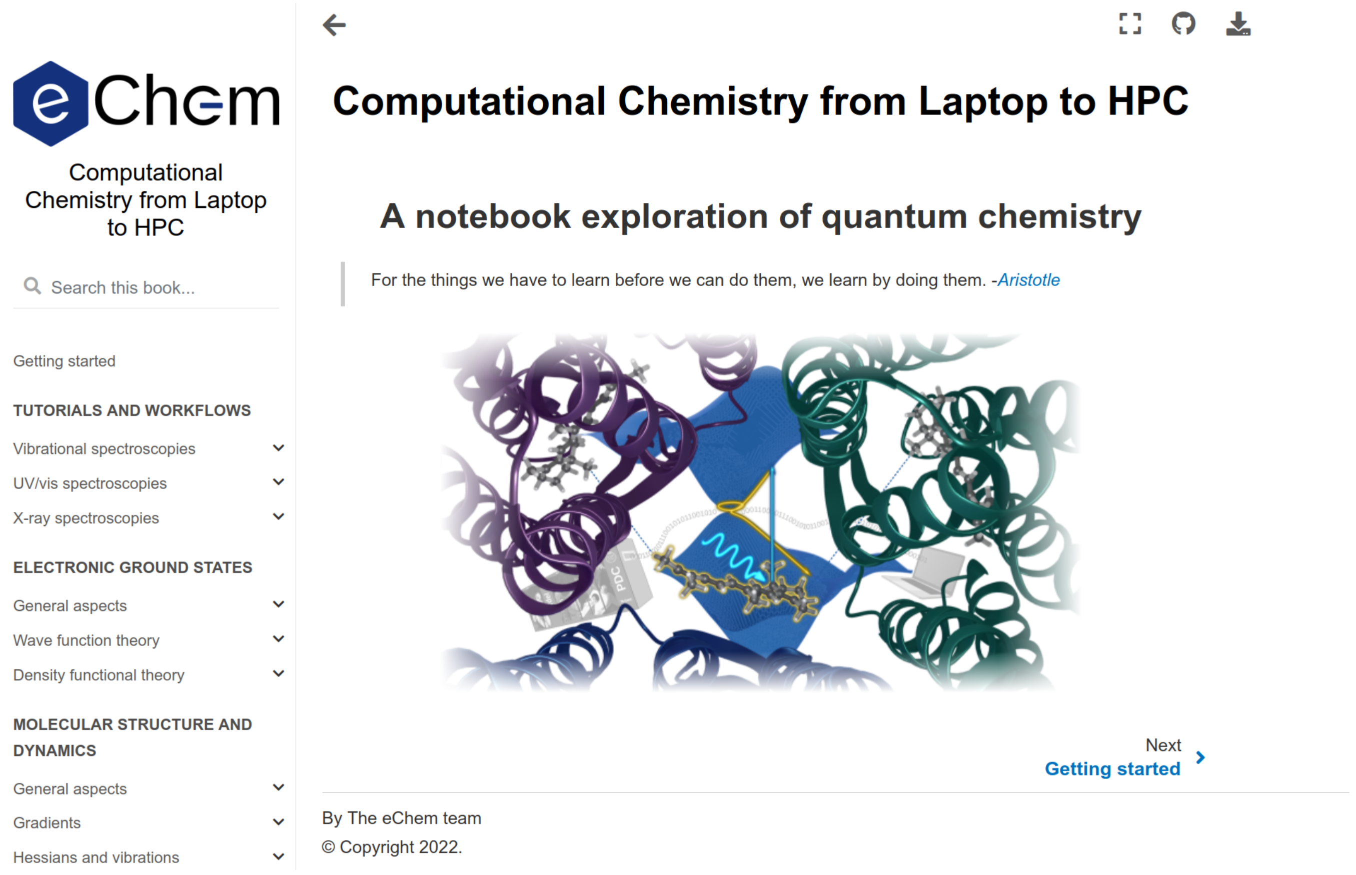
This first part of the eChem book demonstrates how one or several software programs (or modules) are used to perform commonly conducted modeling in theoretical chemistry and how the notebook approach to simulations naturally promotes software interoperability. A choice was made here to deviate from a conventional textbook structure where theory would typically be presented before applications. The underlying reason for doing so is well captured by an ancient quote that is also found on the title page of the eChem book:

> *For the things we have to learn before we can do them, we learn by doing them.* —Aristotle

Moreover, this choice was made with the more advanced user in mind that has already grasped the underlying principles and methods of theoretical chemistry and is in need of learning about methodologies and practicalities. We feel that turning the order of presentation of the material upside down in this way can be done in the format of an e-book without compromising readability owing to the opportunity to introduce ample cross-linking between pages. The tutorial topic selection reflects the present research activities of the current members of the eChem team, and will be expanded in the future.

1. Vibrational spectroscopies

This chapter shows how IR and Raman spectra are computed in practice using VeloxChem. The sections go



Computational Chemistry from Laptop to HPC

Search this book...

Getting started

TUTORIALS AND WORKFLOWS

Vibrational spectroscopies

UV/vis spectroscopies

X-ray spectroscopies

ELECTRONIC GROUND STATES

General aspects

Wave function theory

Density functional theory

MOLECULAR STRUCTURE AND DYNAMICS

General aspects

Gradients

Hessians and vibrations

Computational Chemistry from Laptop to HPC

A notebook exploration of quantum chemistry

> For the things we have to learn before we can do them, we learn by doing them. -*Aristotle*

Next
Getting started

By The eChem team
© Copyright 2022.

FIG. 1. Title page of the eChem book.

through all the practical steps, from geometry optimization to computing the Hessian and performing the vibrational analysis. From an educational perspective, it offers students the possibility to explore the origins of IR and Raman peaks by computing the molecular dipole moment and polarizability along different vibrational modes and by comparing similar molecules that differ, for example, by only one atom substitution.

2. UV/vis spectroscopies

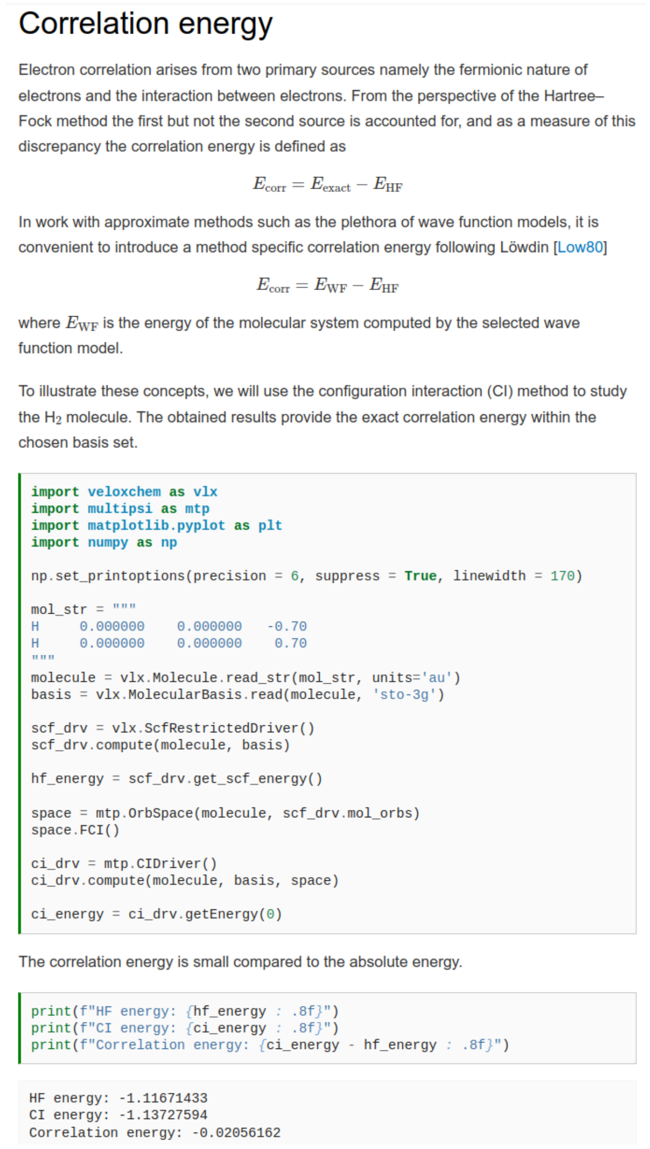
In this chapter the computation of UV/vis spectra are considered, primarily using time-dependent density functional theory (TDDFT), the algebraic diagrammatic construction (ADC) approach, and multi-reference methods. Different methods of resolving spectral features, including the resolution of individual eigenstates or calculating the damped linear response function, are considered, discussing the pros and cons of each approach. Exercises considers the reliable calculation and analysis of the excited states of an arbitrary molecule, as well as ways of lowering computational costs.

3. X-ray spectroscopies

The modeling of X-ray spectra is discussed primarily in the context of ADC and TDDFT, considering X-ray photoelectron, X-ray absorption, and X-ray emission spectroscopy (XPS, XAS, and XES, respectively), as well as resonant inelastic X-ray scattering (RIXS).[52] Various challenges relating to modeling these processes, such as strong relaxation and relativistic effects, are considered, and technical aspects of carrying out the calculations (optimizing core-hole states, using core-valence separation for considering core-excitations, and more) are discussed. More non-standard calculations are also included, which illustrate the flexibility of the frameworks in use. The chapter closes with some general recommendations of which levels of theory are suitable for modeling these processes—this is obviously biased by the experiences of the authors—and exercises for illustrating chemical shifts, the calculation and assignment of spectra, and more.

B. Electronic ground states

This second part of the eChem book gives an introduction to standard electronic structure methods used to



Correlation energy

Electron correlation arises from two primary sources namely the fermionic nature of electrons and the interaction between electrons. From the perspective of the Hartree–Fock method the first but not the second source is accounted for, and as a measure of this discrepancy the correlation energy is defined as

$$E_{\mathrm{corr}} = E_{\mathrm{exact}} - E_{\mathrm{HF}}$$

In work with approximate methods such as the plethora of wave function models, it is convenient to introduce a method specific correlation energy following Löwdin [Low80]

$$E_{\mathrm{corr}} = E_{\mathrm{WF}} - E_{\mathrm{HF}}$$

where E_{WF} is the energy of the molecular system computed by the selected wave function model.

To illustrate these concepts, we will use the configuration interaction (CI) method to study the H_2 molecule. The obtained results provide the exact correlation energy within the chosen basis set.

```
import veloxchem as vlx
import multipsi as mtp
import matplotlib.pyplot as plt
import numpy as np

np.set_printoptions(precision = 6, suppress = True, linewidth = 170)

mol_str = """
H     0.000000    0.000000   -0.70
H     0.000000    0.000000    0.70
"""
molecule = vlx.Molecule.read_str(mol_str, units='au')
basis = vlx.MolecularBasis.read(molecule, 'sto-3g')

scf_drv = vlx.ScfRestrictedDriver()
scf_drv.compute(molecule, basis)

hf_energy = scf_drv.get_scf_energy()

space = mtp.OrbSpace(molecule, scf_drv.mol_orbs)
space.FCI()

ci_drv = mtp.CIDriver()
ci_drv.compute(molecule, basis, space)

ci_energy = ci_drv.getEnergy(0)
```

The correlation energy is small compared to the absolute energy.

```
print(f"HF energy: {hf_energy : .8f}")
print(f"CI energy: {ci_energy : .8f}")
print(f"Correlation energy: {ci_energy - hf_energy : .8f}")
```

```
HF energy: -1.11671433
CI energy: -1.13727594
Correlation energy: -0.02056162
```

FIG. 2. A self-contained eChem notebook showing the integration of explanatory text with numerical examples, thus illustrating the easy accessibility of concepts in theoretical chemistry.

describe electronic ground states. Methods presently included are Hartree–Fock (HF), density functional theory (DFT), configuration interaction (CI), Møller–Plesset (MP) perturbation theory, and multi-configuration (MC) state approaches such as the complete active space self-consistent field (CASSCF) method. This selection will be expanded as to encompass also other commonly used quantum chemistry methods.

1. General aspects

In this chapter, we introduce fundamental concepts of molecular quantum mechanics and set the notation for other parts of the book. We present one- and many-electron wave functions in line with the Pauli principle, and systematically construct the Hilbert spaces in which the wave functions reside with the use of Kronecker products. The introduction of operators acting on the elements of these vector spaces follows suit with distinctly separated actions for the orbital- and spin-degrees of freedom. Owing to its importance, the molecular Hamiltonian is given a detailed treatment, with the electronic part introduced on the basis of the adiabatic Born–Oppenheimer approximation. The interpretation of quantum mechanics is probabilistic and the one- and two-electron densities therefore take center stage in quantum chemistry, leading to a discussion about electron correlation. Here, the eChem book shows its strength by being able to provide the student with visual aids to understand this difficult topic of ubiquitous importance, see Fig. 3.

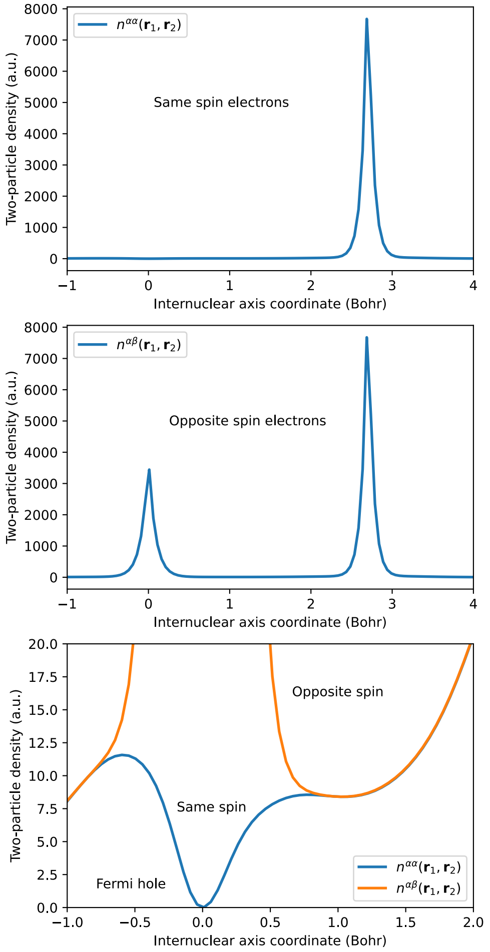


FIG. 3. An eChem illustration of two-particle densities along the internuclear axis of carbon monoxide, with electrons 1 and 2 positioned at the carbon nucleus and along the internuclear axis, respectively. The same (top) and opposite (middle) spin densities are depicted, clearly showing a Fermi hole at the carbon nucleus for the same spin density (bottom).

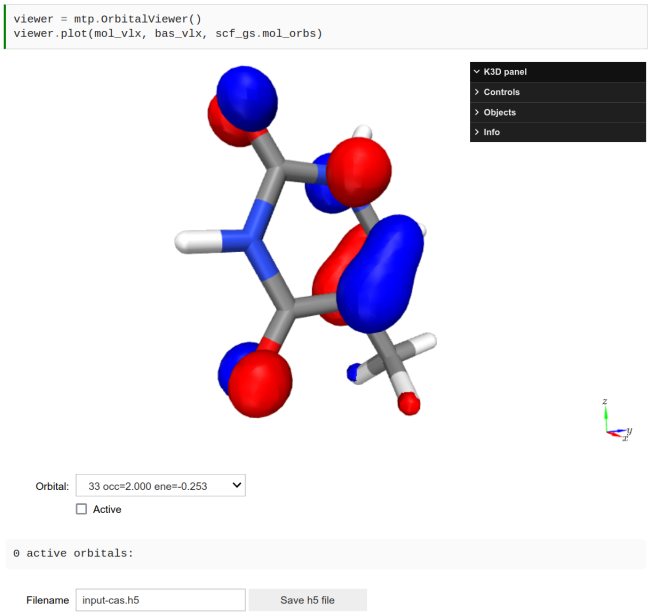
2. Wave function theory

The foundation of wave function approaches in quantum chemistry is the HF method. The eChem book combines a conventional presentation of the underlying self-consistent field (SCF) theory with an associated intertwined notebook implementation based on the data exposure of integrals in the NumPy layer of the VeloxChem program. The poor SCF convergence in the original Roothaan–Hall scheme is illustrated and improved on with an implementation of the direct inversion in iterative subspace (DIIS) acceleration scheme. The pedagogic value of directly experiencing the effects of a numerical scheme in this manner can hardly be overestimated.

The presentation of the CI method introduces the important concept of orbital spaces to control and limit the number of determinants in linearly expanded post-HF wave functions. Class methods in the MultiPsi program are employed to define orbital spaces, view the resulting determinants for a given spin multiplicity, and construct the associated CI Hamiltonian. The section continues with the considerations of *direct* and *truncated* CI techniques that in practical work are imperative to introduce and the issue and lack of size consistency in truncated CI is discussed and illustrated.

In contrast, the presentation made in the next section of MP perturbation theory ends with an illustration that this electron correlation method indeed preserves size consistency.

An extension of CI theory known as multiconfigurational SCF (MCSCF), or CASSCF, is presented next, including an insight into how the method can be implemented. This section features an orbital viewer that serves as a convenient tool when constructing active spaces that should be well balanced, see Fig. 4. This issue can be particularly troublesome upon the breaking of chemical bonds when static correlation is prevalent. The dissociation of the O–H bond in water is used as an illustrative example in this section.



```
viewer = mtp.OrbitalViewer()
viewer.plot(mol_vlx, bas_vlx, scf_gs.mol_orbs)
```

Orbital: 33 occ=2.000 ene=-0.253

☐ Active

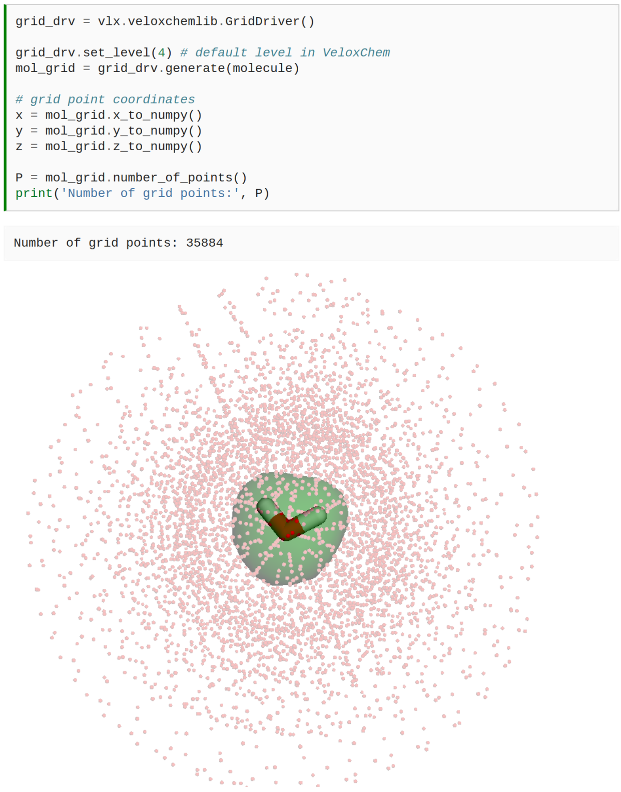
```
0 active orbitals:
```

Filename input-cas.h5

FIG. 4. The OrbitalViewer of MultiPsi, here illustrating the highest-occupied molecular orbital (HOMO) of thymine. The viewer also includes a selection tool to aid the choice of a balanced active space.

3. Density functional theory

The DFT chapter provides a conventional presentation of the fundamental Hohenberg–Kohn theorems, the Kohn–Sham formulation of DFT, and Jacob's ladder of exchange–correlation (xc) functionals. More importantly, the data exposure of grid points and weights in the NumPy layer of the VeloxChem program enables, in a separate section, an exploration of the xc kernel integration that typically is treated in a black-box manner in teaching as well as research, see Fig. 5. The chapter continues with a modification of the SCF program from the HF section and with the resulting DFT program, the important and sometimes critical artifact of the self-interactions error in the Kohn–Sham operator is analyzed and an insightful orbital decomposition is given.



```
grid_drv = vlx.veloxchemlib.GridDriver()

grid_drv.set_level(4) # default level in VeloxChem
mol_grid = grid_drv.generate(molecule)

# grid point coordinates
x = mol_grid.x_to_numpy()
y = mol_grid.y_to_numpy()
z = mol_grid.z_to_numpy()

P = mol_grid.number_of_points()
print('Number of grid points:', P)
```

```
Number of grid points: 35884
```

FIG. 5. Excerpts from the eChem book section illustrating the exposure in the NumPy layer of data structures—here the grid point distribution for numerical integration in DFT.

C. Molecular structure and dynamics

This third part of the eChem book discusses and illustrates the concepts required to determine relaxed molecular structures and other special points on the potential energy surface (PES),[53,54] as well as to perform ground and excited state dynamics simulations. The topics include molecular gradients and Hessians, optimization methods, together with approaches relevant to molecular dynamics such as PES interpolation techniques and force field parametrization.

1. General aspects

We open the part by discussing concepts which are important to all types of explorations of molecular potential energy surfaces. Here we address the different types of coordinates that can be used to describe the molecular structure, from Cartesian to different internal coordinates,[53] as well as methods to transform from one type to another. Additionally, a section on how force fields are parameterized to describe bonded or non-bonded interactions is also included.

2. Gradients

We here provide an in-depth dive into the way that analytical energy and property gradients are derived. The chapter starts with the general Lagrange formalism which can be used in any type of analytical gradient derivation, followed by examples of energy gradients in the ground state (HF, DFT, MP2), and excited states (CIS, TDDFT). Additionally, the way that property gradients, such as the dipole moment gradient and the polarizability gradient, can be determined analytically is also discussed.

3. Hessians and vibrations

This chapter discusses in detail the equations required to determine analytical molecular Hessians at the SCF level and to perform vibrational analyses. Together with the dipole moment and polarizability gradients which relate to the IR and Raman spectral intensities, respectively, this chapter constitutes the theoretical basis for the tutorials on IR and Raman spectroscopies addressed in the *Tutorials and workflows* part of eChem.

4. Structure optimization

Here, we address aspects relevant to finding local minima on the global PES. The chapter starts by discussing and comparing different optimization methods, from the simple gradient-descent algorithm to Newton–Raphson and quasi-Newton methods. It includes examples of practical geometry optimizations using VeloxChem and allows students to implement their own optimization drivers and compare their performances.

5. Transition-state theory

The second class of special points on the PES that we discuss are transition states, which are important in explaining the mechanisms of chemical and photo-chemical reactions. The concept is illustrated by a practical example where students are led through all the steps required to optimize the transition state for a ground state chemical reaction and follow the minimum energy reaction path. Further examples of photo-chemical reactions will be included in the future. As the current notebook contains all the necessary ingredients for students to undertake this exploration independently, this future inclusion may be set up as an exercise.

6. Potential energy surface interpolation techniques

This chapter discusses how potential energy surfaces of ground or excited states can be constructed by interpolation using a set of data points calculated quantum-mechanically (QM).[55] After considering the theoretical aspects of the method, an example of how it performs in the case of the ground state of water is set up as an exercise. The students must first transform the gradient and Hessian from Cartesian to internal coordinates, then perform the interpolation between a set of QM data points that they choose themselves. The interactive Jupyter notebook allows them to explore how the number of QM data points affects the accuracy of the interpolated PES, as well as what happens when PES regions outside the region covered by the QM set are calculated (*i.e.*, performing *extrapolation* instead of interpolation). In the future, we plan to include a simple molecular dynamics driver to allow also dynamics simulations on the interpolated PES with an example of a ground state chemical reaction described using PES interpolation and excited-state dynamics on a multi-state interpolated PES.

7. Molecular dynamics

We here illustrate the parameterization of a force field, which is an important yet often complicated step for setting up a successful molecular dynamics simulation. The rest is primarily a (brief) theory discussion, with more (including numerical examples) to come.

D. Spectra and properties

This fourth part of the eChem book presents methods to describe excited states and calculate transition properties and spectra. Some methods provide a state-specific reference to excited states whereas others are based on an indirect reference via poles and residues of ground state response functions.[56]

1. General aspects

This chapter discusses fundamental aspects of exact state response theory, and how this can be used to derive molecular properties, and later expanded to approximate state methods. With Numerov's method, the rovibronic states of the molecular Hamiltonian for hydrogen chloride are determined using a module implemented in VeloxChem. The vibrational wave functions are pedagogically illustrated on top of the vibrational levels inside the potential energy curves and the associated rovibronic spectra are calculated and plotted. Next an approximation is made to focus on the electronic states of molecular systems, which is what computational chemistry is most often concerned with. We provide an intuitive time-domain introduction to response theory from which numerical values of frequency-dependent response functions are straightforwardly determined from plots of induced dipole moments. Analytical formulas for these response functions is what quantum chemical software implement for the different standard electronic structure theory methods, some of which are discussed in the subsequent chapters of this part of the book.

2. Time-dependent DFT

As DFT does for ground state electronic structure calculations, its time-dependent formalism, TDDFT, represents most of the excited state calculations nowadays.[57] We first go through the underlying time-dependent Hartree–Fock theory and then expand to TDDFT, considering practical aspects of performing calculations and showcasing its strengths and pitfalls (*e.g.* charge-transfer states).

3. Multi-reference methods

Excited-state properties and especially non-adiabatic dynamics constitute an area where multi-configuration state methods play an important role. In this chapter, we describe the different ways we can target excited states using multi-configurational methods, both using CI response and the related concept of state-averaged MCSCF and response based on MCSCF approaches. Once again, our codes allow us to quickly illustrate simple concepts like the erratic convergence of excitation energies in truncated CI (and thus why CIS is really the only truncation level used) or the undesirable dependence of the excitation energies on the number of states in state-averaged MCSCF.

4. Algebraic diagrammatic construction

This chapter discusses the theoretical basis behind the algebraic-diagrammatic construction scheme for the polarization propagator.[58–60] The chapter starts by illustrating the concept of the polarization propagator and shows how its series expansion gives rise to the ADC(n) hierarchy of methods for excited states. The derivation of relevant equations is performed in the more intuitive intermediate state representation and applied to both valence-excited and core-excited states, the latter within the core-valence separation (CVS) approximation. From an educational point of view, the exercises included in this chapter allow students to construct and diagonalize their own ADC matrix for diatomic molecules in a minimal basis. In this way, they learn how to explore the structure of the matrix, which is otherwise difficult to grasp. Additionally, this allows them to estimate the influence of the CVS approximation on the final core-excitation energies.

5. Exciton coupling model

The exciton coupling model is a low-cost approach to determine response properties of a multi-chromophore systems by splitting it into subsystems.[61] It gives a natural classification of electronic transitions as being of local and/or charge-transfer character. This method is illustrated by the calculations of absorption and dichroism spectra of the ethylene dimer in a twisted configuration by placing the two ethylene molecules in separate subsystems, and a comparison is made with a conventional quantum chemical response calculation made on the full dimer system.

E. Environment

This fifth part of the eChem book describes how localized multipole moments and polarizabilities are defined, calculated, and used for molecular representations in dynamics and spectroscopy simulations.

1. Localized properties

This chapter presents two fundamentally different approaches to define localized properties, namely the Merz–Kollman *electrostatic potential* (ESP) and *localized properties* (LoProp) schemes. It describes how modules in the VeloxChem program are available to access these schemes, including the improved retrained version of ESP that defines the so-called RESP charges suitable for the derivation of classical force fields. The LoProp scheme is quite elaborate and typically used in a black-box manner by researchers. The eChem book offers a notebook presentation of this scheme that guides the reader step by step through the process of calculating localized charges and electric-dipole polarizabilities of hydrogen fluoride. An in-depth understanding of these parameters is made available in this interactive e-book and it in turn promotes critical thinking in building models of complex molecular systems.

2. Polarizable embedding

Hybrid quantum mechanics/molecular mechanics (QM/MM) methods are nowadays widely used in simulations of molecular systems in the condensed phase. For calculations of spectra in the polarizable embedding (PE) model, the classical environment of the quantum core region is represented by site-distributed multipole moments and polarizabilities and the aforementioned LoProp method is often used in this context. This chapter presents the basic PE theory and gives an illustrative example of a spectrum calculation of a ligand in aqueous solution.

F. Visualization

This sixth part of the eChem book discusses available techniques for the visualization of molecular structures and orbitals, MD trajectories, reduced particle densities, and transition and excited state densities. It uses modules in VeloxChem, Gator, and MultiPsi, as well as the more generic library modules py3Dmol and k3d.

1. Molecular structure and dynamics

This chapter illustrates how py3Dmol can be used to visualize molecular structures, from small molecules to proteins and using different possible layouts. Additionally, we show how to animate MD trajectories, geometry optimizations, and molecular vibrations.

2. Electronic structure

In this chapter we show how molecular orbitals can be visualized with the OrbitalViewer module available in both VeloxChem and MultiPsi, which provides a lightweight and highly responsive interface for showing molecular orbitals. As support for multi-reference calculations, OrbitalViewer also presents natural occupation numbers and provides an interactive interface for molecular orbital selection and the construction of active spaces, see Fig. 4. This selection can be saved and subsequently used for multi-reference calculations.

Furthermore, the generation of cube-files for visualization of molecular orbitals, natural transition orbitals, and attachment/detachment densities are illustrated, and the theoretical basics are discussed. Radial densities of orbitals are also considered, together with illustrations of reduced one- and two-particle densities—the illustration of the Fermi hole in carbon monoxide shown in Fig. 3 is taken from this section of the e-book.

IV. ECHEM FOR EDUCATION

A. Self-directed learning

Chemistry students in general, including undergraduates and experimentalists, will find an opportunity in the eChem book to learn the basics of theoretical chemistry by performing actual calculations that mirror the theory and concepts under study. We argue that, just as a theoretician benefits from knowledge about the basic experimental methods, the experimentalist benefits from knowledge about the basic computational methods. This promotes a mutual understanding and interdisciplinary work.

Self-directed learning with the eChem book can also be applied in the context of flipped learning, where the focus is on active learning from the part of the student, as supported by instructors. While flipped learning comes with its set of challenges, it has been seen to generally provide additional insight and improved critical thinking skills,[62–64] with the field of chemistry in particular likely to benefit from such an approach.[62]

Additionally, a reoccurring event in a dynamic and active research group is the graduation of developed PhD students followed by the recruitment of new ones. In this situation, the eChem book serves as a means to transfer knowledge. The beginner student may have access to a curriculum of advanced courses, yet there will always be a gap between basic theory and the field-specific knowledge and practical skills that need to be acquired before research activities can be conducted. This training is greatly accelerated with interactive and targeted study materials that are made suitable for self-directed learning.

B. Designing a course module or workshop

There are entire books available discussing the application of computational methods for teaching chemistry and chemical concepts as well as the introduction of programming into the chemistry curriculum.[65,66] It has been noted that access to computational chemistry is essential for undergraduate chemistry programs, a viewpoint that is strongly supported by us. However, the integration of computational chemistry in undergraduate teaching is hampered by a lack of user-friendly and easily accessible software/hardware environments and instructors with adequate training.[48,67] The eChem project is designed to break these barriers.

The use of notebooks for courses and workshops represents a form of blended learning which promotes active learning and has been noted to fare well when compared to traditional learning activities.[68] In active learning, the student is engaged in active, meaningful exercises via technological tools that provide cognitive support.[69] It can be noted that assessment practices are about as important as presentation practises,[70] and in this context nbgrader[24] is of particular relevance. This is a tool which facilitates creating and grading assignments (coding or written form) in a Jupyter notebook environment, including auto-grading functionalities as well.

A module in an undergraduate course based on the eChem book would typically take the form of a series of computer exercises dispersed among more conventional lectures covering the topics addressed in the parts titled *Electronic ground states* and *Molecular structure and dynamics*. A workshop is likely to cover the more advanced topics covered in *Spectra and properties* and *Environment* and combine a presentation of the underlying theoretical foundations with hands-on exercises either taken directly or inspired from the part titled *Tutorials and workflows*.

C. Our experiences

During May 2022, we held a first workshop based on the eChem book. The workshop was carried out over three days, and we had 13 student participants with very different backgrounds and skills. While some were well versed in software development, others had no prior experience in coding. The participants turned out to be very receptive to the strategy of combining lectures with explicit numerical examples, and in the future we plan to organize the workshop in the format of a five-day event to allow for it to be even more interactive. The topics covered were very varied, including tutorials designed to explore the intricacies of different electronic structure methods, as well as tutorials focused on the application of ready implemented routines to probe particular chemistry problems.

For the outcome of a workshop like this to be successful, it is essential that participants do not lose time and interest due to software installation issues. As trivial as this may sound, it becomes a challenge when everyone brings his or her own laptop to the event, which could be running any one of the three main operating systems that in turn have been customized according to personal preferences. For this occasion, we smoothly guided the participants before the event to perform a Conda installation of a predefined and tested Conda environment. However, to make it feasible to study more realistic chemistry problems, we are considering adding the possibility of including access to HPC resources, with which the participants can consider much larger systems. This can be done by using, *e.g.*, JupyterHub. That would be a *complement* to the installation on laptops, as this has benefits in terms of ease of use, and the participants can continue using the packages also after the event.

The varying levels of expertise of the participants showed that some preparation work of the participants would sometimes be suitable. This can be carried out by referring to other Jupyter e-books, focusing on, *e.g.*, the basics of Jupyter, or how to create good illustrations (see, *e.g.*, Ref. 13). This will be suggested before future events.

V. SUMMARY AND OUTLOOK

We have here presented the eChem educational project, featuring an online e-book that enables and stimulates interactive self-directed learning but can also be used to form course modules and workshops. Numerical examples with enlightening figure plots carried out in notebook code cells are intertwined with presentations of basic theory in markdown cells. These self-contained notebooks are easy to download and execute on the reader's personal computer, which promotes further student exploration driven by author-designed exercises and/or stimulated curiosity.

Different parts of the eChem book are suited for students not only at different stages in their educational training but also with different focus in their own work. Undergraduate and beginner graduate students are likely to benefit mostly from the chapters devoted to basic theory and concepts, whereas more senior graduate students and researchers in computational chemistry are likely to benefit mostly from the workflows and tutorials describing methodologies and software interoperability to tackle complex molecular systems of real technical and/or biochemical interest. In addition, the eChem notebooks demonstrate the exposure of C++ data structures in the Python layer which in turn enables the prototyping of novel scientific methods. This will benefit graduate students and researchers with a focus in their work on theory development and program implementation.

As noted earlier, the eChem book is a living document, designed to be constantly improved and expanded such that its repository of notebooks exemplifies and explains the state-of-the-art methodologies in theoretical chemistry. It will never be complete in coverage and will always be biased by the research interests of its authors. The most imminent additions to be made include tutorials on catalysis and reactivity, optical activities, and nonlinear field interactions.

ACKNOWLEDGEMENTS

Acknowledgements are directed toward financial support from the Swedish e-Science Research Centre (SeRC), the Swedish Research Council (Grant No. 2018-4343), the Air Force Office of Scientific Research (Grant FA8655-20-1-7010), and Samsung Electronics (IO211126-09175-01) as well as computing resources provided by the Swedish National Infrastructure for Computing (SNIC).

AUTHOR DECLARATIONS

Conflicts of interest

The authors have no conflicts of interest to disclose.

Licensing

[1] T. Kluyver, B. Ragan-Kelley, , B. Pérez, F. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, and S. Corlay *et al.*, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, edited by F. Loizides and B. Schmidt (2016) pp. 87 – 90.

[2] Executable Books Community. Jupyter Book (v0.10). Zenodo. `https://doi.org/10.5281/zenodo.4539666`.

[3] J. M. Perkel, Nature **563**, 145 (2018).

[4] F. Perez and B. E. Granger, Comp. Sci. Eng. **9**, 21 (2007).

[5] J. M. Perkel, Nature **589**, 344 (2021).

[6] B. P. Abbott *et al.*, Phys. Rev. Lett. **116**, 061102 (2016).

[7] The Event Horizon Telescope Collaboration *et al.*, Astrophys. J. Lett. **875**, L1 (2019).

[8] A. Rule, A. Tabard, and J. D. Hollan (2018).

[9] A. Rule, A. Birmingham, C. Zuniga, I. Altintas, S.-C. Huang, R. Knight, N. Moshiri, M. H. Nguyen, S. B. Rosenthal, F. Perez, and P. W. Rose, PLoS Comput. Biol. **15**, e1007007 (2019).

[10] A. Davies, F. Hooley, P. Causey-Freeman, I. Eleftheriou, and G. Moulton, PLOS Comp. Biol. **16**, e1008326 (2020).

[11] M. L. Richardson and B. Amini, Acad. Radiol. **25**, 801 (2018).

[12] C. J. Weiss, J. Chem. Educ. **94**, 592 (2017).

[13] C. J. Weiss, J. Chem. Educ. **98**, 489 (2021).

[14] E. J. Menke, J. Chem. Educ. **97**, 3899 (2020).

[15] C. J. Weiss, J. Chem. Educ. **94**, 1904 (2017).

[16] M. van Staveren, J. Chem. Educ. (2022).

[17] E. S. Thrall, J. Lee, S. E. ans Schrier, and Y. Zhao, J. Chem. Educ. **98**, 3269 (2021).

[18] D. Lafuente, B. Cohen, G. Fiorini, A. A. García, M. Bringas, E. Morzan, and D. Onna, J. Chem. Educ. **98**, 2892 (2021).

[19] B. A. Clent, Y. Wang, H. C. Britton, F. Otto, C. J. Swain, M. H. Todd, J. D. Wilden, and A. B. Tabor, J. Chem. Educ. **98**, 2899 (2021).

[20] V. W. D. Cruzeiro, A. Roitberg, and N. C. Polfer, J. Chem. Educ. **93**, 1578 (2016).

[21] V. W. D. Cruzeiro, X. Gao, and V. D. Kleiman, J. Chem. Educ. **96**, 1663 (2019).

[22] M. N. Srnec, S. Upadhyay, and J. D. Madura, J. Chem. Educ. **93**, 2106 (2016).

[23] J. Rodríguez-Becerra, L. Caceres-Jensen, T. Diaz, S. Druker, V. B. Padilla, J. Pernaa, and M. Aksela, Chem. Edu. Res. Prac. **21**, 638 (2020).

[24] See `https://nbgrader.readthedocs.io/en/stable/`.

[25] Z. Rinkevicius, X. Li, O. Vahtras, K. Ahmadzadeh, M. Brand, M. Ringholm, N. H. List, M. Scheurer, M. Scott, A. Dreuw, and P. Norman, WIREs Comput. Mol. Scii. **10**, e1457 (2019).

[26] D. R. Rehn, Z. Rinkevicius, M. Herbst, M. Li, X. Scheurer, M. Brand, A. L. Dempwolff, I. E. Brumboiu, T. Fransson, A. Dreuw, and P. Norman, WIREs Comput. Mol. Sci. **11**, e1528 (2021).

[27] M. F. Herbst, M. Scheurer, T. Fransson, D. R. Rehn, and A. Dreuw, WIREs Comput. Mol. Sci. **10**, e1462 (2020).

[28] Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K.-L. Chan, WIREs Comput. Mol. Sci. **8**, 1340 (2018).

[29] Q. Sun, X. Zhang, S. Banerjee, P. Bao, M. Barbry, N. S. Blunt, N. A. Bogdanov, G. H. Booth, J. Chen, Z.-H. Cui, and J. J. Eriksen *et al.*, J. Chem. Phys. **153**, 024109 (2020).

[30] S. Grimme, C. Bannwarth, and P. Shushkov, J. Chem. Theory Comput. **13**, 1989 (2017).

[31] C. Bannwarth, E. Caldeweyher, S. Ehlert, A. Hansen, P. Pracht, J. Seibert, S. Spicher, and S. Grimme, WIREs Comput. Mol. Sci. **11**, e1493 (2021).

[32] M. Bursch, H. Neugebauer, and S. Grimme, Angew. Chem. Int. Ed. **58**, 11078 (2019).

[33] P. Pracht, F. Bohle, and S. Grimme, Phys. Chem. Chem. Phys. **22**, 7169 (2020).

[34] L.-P. Wang and C. Song, J. Chem. Phys. **144**, 214108 (2016).

[35] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, and C. D. Stern *et al.*, PLoS Comput. Biol. **13**, e1005659 (2017).

[36] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, and N. J. Smith *et al.*, Nature **585**, 357 (2020).

[37] J. D. Hunter, Comp. Sci. Eng. **9**, 90 (2007).

[38] N. Rego and D. Koes, Bioinform. **31**, 1322 (2015).

[39] K. Seshadri, P. Liu, and D. R. Koes, J. Chem. Educ. **97**, 3872 (2020).

[40] S. Lehtola and A. J. Karttunen, WIREs Comput. Mol. Sci. , e1610 (2022).

[41] M. D. Hanwell, C. Harris, A. Genova, M. Haghighatlari, M. El Khatib, P. Avery, J. Hachmann, and W. A. de Jong, Int. J. Quantum. Chem. **121**, e26472 (2021).

[42] A. H. Larsen, J. J. Mortensen, J. Jakob Blomqvist, I. E. Castelli, R. Christensen, M. Dulak, J. Friis, M. N. Groves, B. Hammer, and C. Hargus *et al.*, J. Phys.: Condens. Matter **29**, 273002 (2017).

[43] D. G. A. Smith, A. T. Lolinco, Z. L. Glick, J. Lee, A. Alenaizan, T. A. Barnes, C. H. Borca, R. Di Remigio, D. L. Dotson, and S. Ehlert *et al.*, J. Chem. Phys. **155**, 204801 (2021).

[44] C. Ehlert and T. Klamroth, J. Comp. Chem **41**, 1781 (2020).

[45] D. G. A. Smith, D. Altarawy, L. A. Burns, M. Welborn, L. N. Naden, L. Ward, S. Ellis, B. P. Pritchard, and T. D. Crawford, WIREs Comput. Mol. Sci. **11**, e1491 (2020).

[46] J. M. H. Olsen, S. Reine, O. Vahtras, E. Kjellgren, P. Reinholdt, K. O. H. Dundas, X. Li, J. Cukras, M. Ringholm, and E. D. Hedegård *et al.*, J. Chem. Phys. **152**, 214115 (2020).

[47] D. G. A. Smith, L. A. Burns, D. A. Sirianni, D. R. Nascimento, A. Kumar, A. W. James, J. B. Schriber, T. Zhang, B. Zhang, and A. S. Abbott *et al.*, J. Chem. Theory Comput. **14**, 3504 (2018).

[48] W. F. Polik and J. R. Schmidt, WIREs Comput. Mol. Sci. **12**, e1554 (2022).

[49] R. C. Fortenberry, A. R. McDonald, T. D. Shepherd, M. Kennedy, and C. D. Sherrill, ACS Symposium Series **1193** (2015).

[50] M. J. Perri, M. Akinmurele, and M. Haynie, ACS Symposium Series **1312** (2019).

[51] P. A. Kirschner and P. De Bruyckere, Teach. Teach. Educ. **67**, 135 (2017).

[52] P. Norman and A. Dreuw, Chem. Rev. **118**, 7208 (2018).

[53] H. B. Schlegel, WIREs: Comput. Mol. Sci. **1**, 790 (2011).

[54] F. Jensen, *Introduction to computational chemistry, 2nd Ed.* (John Wiley & Sons Ltd., 2006).

[55] Y. M. Rhee and J. W. Park, Int. J. Quantum Chem. **116**, 573 (2016).

[56] P. Norman, K. Ruud, and T. Saue, *Principles and Practices of Molecular Properties; Theory, Modeling and Simulations* (John Wiley & Sons, 2018).

[57] A. Dreuw and M. Head-Gordon, Chem. Rev. **105**, 4009 (2005).

[58] A. Dreuw and M. Wormit, WIREs Comput. Mol. Sci. **5**, 82 (2015).

[59] J. Schirmer, Phys. Rev. A **26**, 2395 (1982).

[60] J. Schirmer and A. B. Trofimov, J. Chem. Phys. **120**, 11449 (2004).

[61] X. Li, R. M. Parrish, F. Liu, S. I. L. Kokkila Schumacher, and T. J. Martinez, J. Chem. Theory Comput. **13**, 3493 (2017).

[62] H. Al-Samarraie, A. Shamsuddin, and A. I. Alzagrani, Educ. Tec. Res. Dev. **68**, 1017 (2020).

[63] M. Munzil, M. Pandaleke, and S. Sumari, AIP Conf. Proc. **2215**, 020014 (2020).

[64] M. K. Seery, Chem. Educ. Res. Pract. **16**, 758 (2015).

[65] A. Grushow and M. S. Reeves, ACS Symposium Series **1312** (2019).

[66] A. R. McDonald, ACS Symposium Series **1387** (2021).

[67] R. Kobayashi, T. P. M. Goumans, N. O. Carstensen, T. M. Soini, N. Marzari, I. Timrov, S. Ponce, E. B. Linscott, C. J. Sewell, and G. Pizzi *et al.*, J. Chem. Educ. **98**, 3163 (2021).

[68] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, Proc. Nat. Acad. Sci. **111**, 9410 (2014).

[69] R. F. Schmid, R. M. Bernard, E. Borokhovski, R. M. Tamim, P. C. Abrami, M. A. Surkes, C. A. Wade, and J. Wood, Comp. & Edu. **72**, 271 (2014).

[70] M. Schneider and F. Preckel, Psych. Bul. **143**, 565 (2017).