Article:

Shaw, R.A. and Hill, J.G. orcid.org/0000-0002-6457-5837 (2023) BasisOpt: a Python package for quantum chemistry basis set optimization. The Journal of Chemical Physics, 159 (4). 044802. ISSN 0021-9606

https://doi.org/10.1063/5.0157878

The Journal of Chemical Physics

AIP Publishing

RESEARCH ARTICLE | JULY 25 2023

BasisOpt: A Python package for quantum chemistry basis set optimization

Robert A. Shaw ; J. Grant Hill

J. Chem. Phys. 159, 044802 (2023)
https://doi.org/10.1063/5.0157878

The Journal of Chemical Physics

ARTICLE

pubs.aip.org/aip/jcp

BasisOpt: A Python package for quantum chemistry basis set optimization

Cite as: J. Chem. Phys. 159, 044802 (2023); doi: 10.1063/5.0157878
Submitted: 12 May 2023 • Accepted: 3 July 2023 •
Published Online: 25 July 2023

Robert A. Shaw and **J. Grant Hill**[a)]

AFFILIATIONS

Department of Chemistry, University of Sheffield, Sheffield S3 7HF, United Kingdom

[a)]Author to whom correspondence should be addressed: grant.hill@sheffield.ac.uk

ABSTRACT

The accuracy and efficiency of molecular quantum chemical calculations depend critically on the basis set used. However, the development of novel basis sets is hindered because much of the literature relies on the use of opaque processes and tools that are not publicly available. We present here BasisOpt, a tool for the automated optimization of basis sets with an easy-to-use framework. It features an open and accessible workflow for basis set optimization that can be easily adapted to almost any quantum chemistry program, a standardized approach to testing basis sets, and visualization of both the optimized basis sets and the optimization process. We provide examples of usage in realistic basis set optimization scenarios where: (i) a density fitting basis set is optimized for He, Ne, and Ar; (ii) the exponents of the def2-SVP basis are re-optimized for a set of molecules rather than atoms; and (iii) a large, almost saturated basis of sp primitives is automatically reduced to (10s5p) while achieving the lowest energy for such a basis set composition.

I. INTRODUCTION

Quantum chemistry is a vital part of modern chemistry research, with applications ranging from characterizing unknown products all the way through to designing new molecules. However, it is still limited in the size of systems that are tractable. There is typically a trade-off between accuracy and efficiency due to the computational expense of the various steps in typical algorithms, for example, in calculating molecular integrals or in the diagonalization of an effective Hamiltonian. There are two primary aspects of a quantum chemical calculation that contribute to both the cost and accuracy: the choice of method and the basis set used to represent the system.

A great deal of research has focused on the first of these. In pursuit of accuracy, there have been many attempts at simplifying high-cost wavefunction theory, for example, through the use of local orbitals[1,2] and density fitting[3–8] approximations. In this way, it is now possible to routinely run coupled-cluster level calculations on systems of up to a few hundred atoms,[9,10] although the cost is still far from negligible, especially for properties requiring derivatives. Similarly, attempts have been made to make low-cost methods more accurate, for example by mixing wavefunction theory and density functional theory (DFT).[11] On the other end of the spectrum, several groups have sacrificed some accuracy in order to develop very low cost methods; particularly popular are the semi-empirical DFT extended tight-binding methods.[12]

At their root, however, all of these methods rely on a basis set used to expand the unknown wavefunction (or electronic density) in terms of simple, known functions. A full, authoritative discussion of basis sets in quantum chemistry, along with their design, development, and optimization, would necessitate a lengthy monograph. Herein, we instead focus on providing the level of detail required to provide sufficient context and refer the interested reader to additional sources at appropriate points. For molecular applications, the basis set expands every molecular orbital ψ as a linear combination of atomic orbitals (LCAOs) that are typically termed basis functions, χ:

$$\psi_i = \sum_{p}^{N_b} c_{pi}\chi_p,$$

where c_{pi} are the molecular orbital coefficients, and N_b is the total number of basis functions in the set. The number of basis functions of each angular symmetry (s, p, d, f, etc.) defines the "composition" of the basis set, with many sets further classified into so-called "qualities." Several review articles discuss basis set classifications in detail and should be consulted for further details,[13–17] but as a brief definition, a basis that contains only the functions matching the minimal set of atomic orbitals required for the electronic ground state of the atoms within a molecular system is known as a minimal basis set. Doubling the number of functions such that each atomic orbital is described by two basis functions leads to a double zeta (DZ) quality basis, tripling the functions leads to a triple zeta (TZ) quality basis, and so-on. The majority of basis sets also contain "polarization functions" of higher angular symmetry than required for ground state electronic configurations.

There are multiple types of LCAO basis set that may be used, and the most common of these, including Slater-type orbitals, Gaussian-type orbitals, and numerical atomic orbitals, have recently been reviewed.[18] The same review also discusses the advantages and disadvantages of LCAO approaches. The usual choice for molecular applications, for reasons of efficiency when calculating integrals,[19] is the Gaussian-type orbital,

$$\chi = N Y_{l,m} r^{2n-2-l} e^{-\alpha r^2},$$

where N ensures normalization, n, l, and m are the principal, azimuthal (orbital angular momentum), and magnetic quantum numbers, respectively, $Y_{l,m}$ are spherical harmonic functions, and α is commonly known as the basis set exponent. While there have been various investigations into, for example, midbond functions,[20–22] or "floating" s-type orbitals,[23] the usual procedure is to have a basis set associated with each atom type and duplicate this set on each atomic center.

Many, though not all, Gaussian basis sets are used in a contracted form, whereby the "primitive" Gaussian functions, χ^{prim}, are combined into a smaller set by taking linear combinations of the primitives

$$\chi^{\text{con}} = \sum_{j}^{k} a_j \chi_j^{\text{prim}},$$

where χ^{con} is known as a contracted function, and a_j are fixed contraction coefficients. While the use of a contracted Gaussian basis reduces the computational cost, relative to the same set of uncontracted primitives, it also introduces a contraction error.[24] For a contracted basis set, the basis set quality (DZ, TZ, etc.) outlined above usually refers to the number of contracted functions rather than the number of primitives. In terms of defining which primitive functions enter a given contraction, one of two schemes is commonly used: general contraction or segmented contraction. In a general contraction, all of the primitives for a given atom type (and with the same angular symmetry) appear in all of the contracted functions, although the contraction coefficients will differ across all contracted functions.[25] By contrast, segmented contraction schemes first sub-divide the primitives of a given angular symmetry into smaller sets (segments) of functions, and it is these segmented sets that are then contracted.[26,27] Practically speaking, many contemporary Gaussian basis sets use what is sometimes termed a "partial" contraction scheme, whereby the basis has a combination of a number of uncontracted primitives along with some (general or segmented) contracted functions. Again sticking with practicalities, the integral evaluation routines in a given quantum chemical package may favor one particular contraction scheme; hence, some work has been carried out on procedures for converting generally contracted basis sets into segmented contracted ones without loss of accuracy.[28]

An assumption is typically made that, by increasing the number of functions in the basis set, we approach a limit—the "complete basis set" (CBS) limit—whereby the wavefunction has been represented exactly in the relevant Hilbert space to within numerical limits. Whether this assumption is valid is so far unproven for most methods,[29–32] although there is empirical evidence to support it.[33–35] Unfortunately, Gaussian basis sets designed in this way approach completeness very slowly. In particular, correlation energy tends to require many high-angular momentum polarization functions,[36] greatly increasing the cost of the method. There are two principal reasons for this: Gaussian basis sets are incapable of accurately representing the wavefunction in the neighborhood of a nucleus (the Coulomb hole); the exponents and positions of the basis functions are fixed and only adaptable via linear combination. The former problem can be addressed by either using different function types with the correct behavior at the nucleus (such as Slater-type orbitals),[37] or by adding geminals to the wavefunction that are asymptotically linear in the electron separation (so-called F12 methods).[38] The necessity to fix the positions and exponents, however, comes primarily from the need for calculations to be computationally facile. In an attempt to alleviate this, the exponents for a given atom are carefully optimized on a method-by-method basis according to criteria that depend on the desired use.

For the majority of molecular quantum chemistry calculations, a pre-existing Gaussian basis set will be selected from an internal library within an electronic structure package or perhaps downloaded from an online repository.[39–41] However, in some cases, it may be necessary or desirable to design, develop, and optimize new basis sets. Assuming an LCAO basis set comprising Gaussian-type orbitals, this would consist of optimizing the exponents of a number of primitive functions, followed by (optionally) determining how to contract the functions and the values of the contraction coefficients. An introduction to mathematical optimization is beyond the scope of the present article, and the interested reader is referred to standard texts, such as Ref. 42. For basis sets, this encompasses selecting the best possible exponents as variables to minimize some "objective function" (also known as a loss function) using an optimization method, for example, the simplex method or a quasi-Newton method such as BFGS.[42] Most commonly, exponents are "atom-optimized," where they are determined for some electronic state(s) of an individual atom, even if they are to be later re-used in calculations on molecules. Similarly, the majority of basis sets are "energy-optimized," meaning the objective function is a (variational) energy or the difference between the energy from the current set of exponents and some reference value, such as the CBS limit. There are also situations where having the objective function related to other properties, such as density or some "completeness profile," may be more useful.[43,44]

In terms of the values of the exponents, these may be "freely optimized," where the exponents are free to take on any value and are not constrained relative to the value of other exponents. While this works well for relatively small numbers of exponents, the optimization becomes significantly more difficult as the number of basis functions increases. This is not just that the problem has grown in size, but it also becomes ill-conditioned and is likely to lead to a so-called "variational collapse,"[45] with two exponents with the same angular symmetry converging to the same value. An alternative is to use a "geometric sequence" to define the exponents based on a smaller number of parameters to be optimized. A straight-forward example is an "even-tempered" basis set,[46] where the exponents of a given angular symmetry are defined by

$$\alpha_n = k^{n-1}\alpha_0,$$

where k and α_0 are the two variables to be optimized, and the positive integer n defines the nth exponent in the sequence. Other sequences that also eliminate the ill-conditioning problem, yet introduce more flexibility than an even-tempered basis, include well-tempered basis sets,[47] and the use of Legendre polynomials.[48] In addition to the number of basis functions to be optimized, an initial "starting guess" at the exponent values, or other parameters in the case of geometric sequences, will need to be provided. A potential problem at this stage is one familiar throughout mathematical optimization, namely convergence to a local, rather than global, minimum. The selection of the appropriate minimum in terms of the exponents is usually guided by extensive benchmarking/test calculations of candidate basis sets. These candidates are often generated by re-running the optimization procedure from several different starting guesses.

Assuming that a contracted Gaussian basis set is desired, and that primitive exponents have been optimized, a design choice on the type of contraction scheme may be made. Focusing momentarily on general contraction, one of the simplest methods for determining the contraction coefficients is that adopted by the atomic natural orbital (ANO) family of basis sets.[49] Here the contraction coefficients are the coefficients of the natural orbitals from an atomic configuration interaction calculation; hence, they may be "read" directly from the output of an electronic structure program with no further optimization required. Further refinements of this technique have averaged the density matrix over multiple atomic states, ions, and atoms in external electric fields,[50] but the core process of computing a set of coefficients and re-using them as the contraction coefficients remains the same. One method of generating partial general contractions would be to take this fully contracted basis and then sequentially uncontract primitives until the contraction error reaches some desired threshold. Alternatively, the number and pattern of exponents to uncontract may also be defined by the overall design of the basis set or be the result of benchmarking calculations.

As noted by Jensen,[24] the "definition of segmented basis sets is significantly more ambiguous than a general contraction." The usual method is for the primitive exponents and the contraction coefficients to be optimized simultaneously, resulting in a highly nonlinear optimization problem.[24] Determining the total number of primitives to include in the basis set and how they are to be subdivided into segments is also a non-trivial, combinatorial problem. As with the optimization of primitive exponents, the final choice of segmented contraction pattern and local minimum is most often the result of extensive test calculations. Given these difficulties in defining how to construct a segmented basis set, it is perhaps not surprising that procedures exist for constructing a segmented scheme from a previously developed general contraction basis set. The Davidson purification method can be used to produce a partial segmentation of some of the functions in a general contracted basis,[51] while the above-mentioned P-orthogonalization method of Jensen identifies a full pathway from general contraction to segmented contraction.[28]

Although the above-mentioned outline of basis set optimization is mostly focused on Gaussian-type orbital basis sets, electronic structure methods using density-fitting approximations require (possibly several) auxiliary basis sets matched to the orbital basis. A number of procedures exist for automatically generating auxiliary basis sets,[52–55] but the more traditional approach is to explicitly optimize the exponents of auxiliary sets.[7,56] The latter typically produces slightly more compact sets in terms of auxiliary basis set composition. For an F12 calculation, three different auxiliary sets are required, including one specifically for use in the resolution-of-the-identity approximation that simplifies multielectron integrals into products of no more than two-electron integrals.[57,58] The "OptRI" family of auxiliary basis sets is often used for this purpose.[59]

The process of developing LCAO basis sets for molecular applications has so far largely been done by a small number of groups and is not particularly accessible or obvious. Some electronic structure packages have tools available for basis set development, such as the general-purpose optimizer in Molpro,[60] or the basis set optimization program that interfaces with GAMESS,[61,62] but these are typically limited in terms of features and require substantial technical knowledge of the specific package. In order to increase the applicability and efficiency of quantum chemistry, particularly as new methods are developed at an increasing rate, it would be useful to have a software-agnostic, easy-to-use framework for basis set optimization. In this work, we present a Python package, BasisOpt, developed to tackle this problem.

II. AIMS AND METHODOLOGY OVERVIEW

BasisOpt is an open-source library written in modern Python, designed to bring together computational chemistry programs with state-of-the-art optimization methods. We have three primary aims with this package:

1. to provide an open, accessible, and flexible workflow for basis set optimization;
2. to allow for the development of novel basis set types and optimization methods, and to formalize existing approaches in a robust manner;
3. to standardize the testing and reporting of basis sets across different platforms.

The typical methodology workflow we envision is shown in Fig. 1. The user starts with a particular choice of basis set type, an atom or molecule (or collection of either) for which the basis set is to be optimized, and a calculation method. The optimization proceeds according to a given strategy (see Sec. III C), checking at each stage that it is proceeding correctly. Finally, the optimized basis set is tested across a varied selection of criteria, and the results are stored and visualized. The workflow may then proceed in a loop, or a different strategy can be tried if the desired accuracy is not achieved. In order to achieve the first two of our aims, it is important that each stage of this process is both easy to use with minimal code, and easily customizable, so that new approaches can be developed. In Sec. III, we describe each of the parts of the workflow in more detail. In the

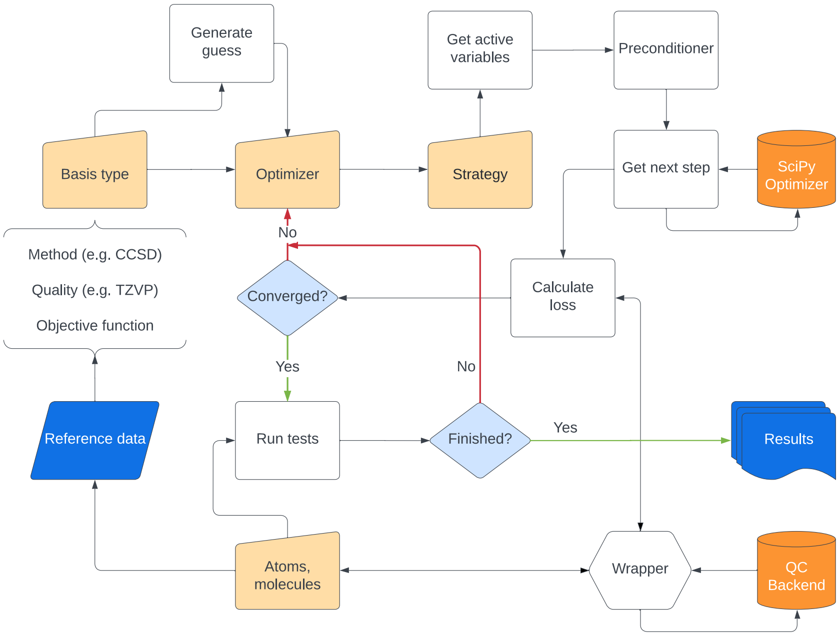


FIG. 1. The high-level workflow used to optimize basis sets with BasisOpt. The user chooses a basis type and optimization strategy for a set of atoms and/or molecules. BasisOpt then uses SciPy and an external quantum chemistry backend to iteratively optimize the basis sets, outputting the results in a standardized format.

final two sections, we will look at the possible limitations and future developments and give proof-of-concept examples for how to use BasisOpt. The software and all the examples given here can be found at https://github.com/robashaw/basisopt, along with documentation that includes a quick start guide, tutorials, and detailed information on the application programming interface (API).

III. SOFTWARE ARCHITECTURE

The package is written in Python 3 and can be installed via the PyPI package manager. It provides an interface with the basis set exchange,[39] which can be used to fetch standard reference basis sets, and provides methods for converting basis sets between formats. Thus, any basis set developed with BasisOpt can automatically be used in essentially any quantum chemistry package. In version 1.0.0, we provide wrappers for Psi4[63] and ORCA,[64] although, as will be discussed in Sec. III A, it is fairly simple to add wrappers for other programs. This allows for a software-agnostic approach to basis set development, as any given wrapper can be used identically, so long as the correct methods are exposed to the interface.

For routine use, there is a high-level API, with each part being adaptable and encapsulated. The components of the optimization are contained within four base objects: a `Molecule`, `Basis`, `Strategy`, and `Result`. The latter three will be described in more detail shortly. The `Molecule` object contains all the basic information for the system (or systems) of interest; typically, this includes the coordinates, charge, multiplicity, and method to use. Any calculation can be run by passing a `Molecule` to the API, along with a `Basis` and any optional settings. For example, a traditional atomic basis set optimization where the exponents are freely optimized could be achieved as presented in Listing 1.

LISTING 1. Python code to optimize a triple-zeta quality basis for the neon atom at the Hartree–Fock level, using Psi4[63] for calculations and the default optimization `Strategy`. The results are then saved as a Python pickle file.

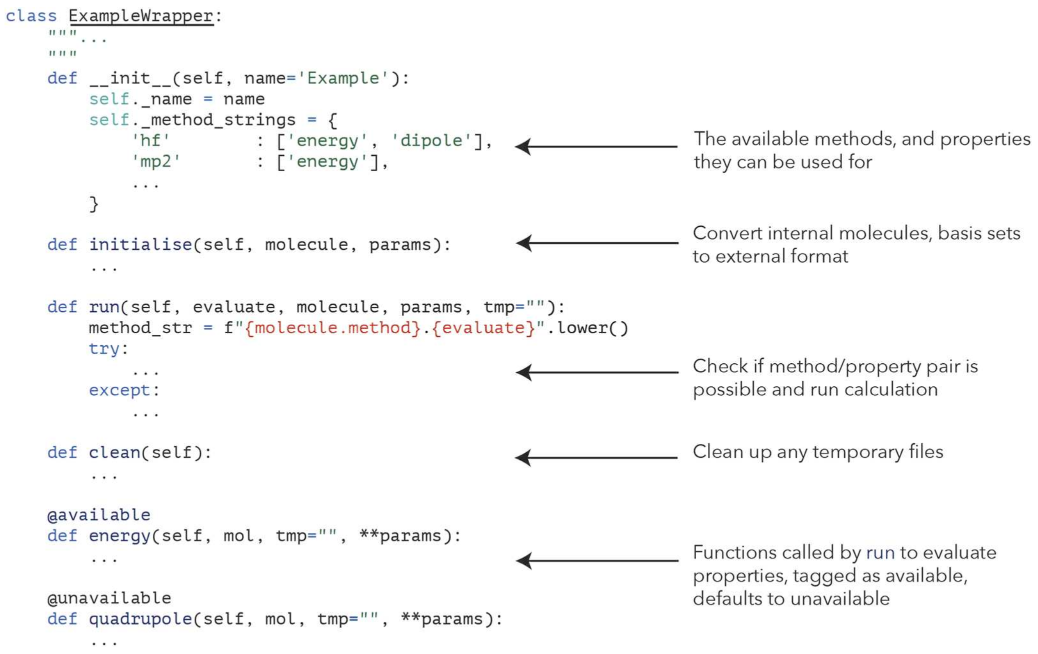
```
import basisopt as bo
from basisopt.basis import AtomicBasis

bo.set_backend('psi4')
ne = AtomicBasis('Ne')
ne.setup(method='scf', quality='tz')
ne.optimize(algorithm='Nelder-Mead')
ne.save('neon-tz.pkl')
```

A. Backends

To achieve the most general applicability of BasisOpt, it is important that the same procedures can be applied irrespective of the particular quantum chemical method and, beyond that, the choice of implementation of that method. For example, there has been an explosion of new local correlation methods available only in specific codes, yet the nature of the basis set and the optimization process should be the same. To this end, BasisOpt uses a skeleton `Wrapper` class, from which interfaces to specific programs can be derived. Any given method can be exposed to the library in a standard fashion, and use of the high-level API requires no knowledge of how the program is operating.

A diagram showing the structure of a wrapper is given in Fig. 2. Currently, there are built-in wrappers for Psi4 and ORCA,



```
class ExampleWrapper:
    """...
    """
    def __init__(self, name='Example'):
        self._name = name
        self._method_strings = {
            'hf'        : ['energy', 'dipole'],
            'mp2'       : ['energy'],
            ...
        }

    def initialise(self, molecule, params):
        ...

    def run(self, evaluate, molecule, params, tmp=""):
        method_str = f"{molecule.method}.{evaluate}".lower()
        try:
            ...
        except:
            ...

    def clean(self):
        ...

    @available
    def energy(self, mol, tmp="", **params):
        ...

    @unavailable
    def quadrupole(self, mol, tmp="", **params):
        ...
```

The available methods, and properties they can be used for

Convert internal molecules, basis sets to external format

Check if method/property pair is possible and run calculation

Clean up any temporary files

Functions called by run to evaluate properties, tagged as available, defaults to unavailable

FIG. 2. An outline of how wrappers for external quantum chemistry programs are laid out.

which act as examples for how further wrappers could be written. The principal idea is that there is a list of possible methods, and these are marked as "available" to the API when implemented in a particular wrapper. The calculation is then called via the `run` command with only a molecule and the name of the electronic structure method—e.g., "ccsd," for coupled cluster with single and double excitations. The result is then returned in a native Python data type.

A key principle of this design is that all electronic structure methods and properties are by default unavailable and will result in an error if called until implemented and marked available in the code. This makes it clear to both developers and users what is possible with a given backend. All of the input file generation and output file parsing is encapsulated by the wrapper, for example, through external libraries such as cclib.[65,66] Moreover, with this approach, it is simple to have calculations sent to a batch queuing system, rather than performed directly on the local computer, and to run multiple calculations in parallel.

B. Basis set types

The central object in BasisOpt is the `Basis`. This defines a general collection of properties and routines that run and store the results of a basis set optimization. The abstract `Basis` class has the following three core parts: methods to set up and run the optimization; containers for the molecule(s) and result(s) of the optimization; and a collection of tests to be run on the optimized set. In addition, there are basic utility functions, for example, uncontracting a basis, making an even-tempered expansion, and fixing the ratio between exponents. Subclasses of the `Basis` object implement the specific functionality for the type of basis being optimized within this general framework. As the simplest case, we describe in more detail the `AtomicBasis`; summaries of a few other example types are given in Table I.

An `AtomicBasis` is initialized with an element name, charge, and multiplicity. The data for the element, in particular the electronic configuration, are collected from the Mendeleev library.[67] The two steps in every basis optimization are the setup, and the optimization itself. In the setup, we provide a computational method (e.g., CCSD or Hartree–Fock), a basis set "quality," and a `Strategy` for the optimization. The latter will be described in more detail in Sec. III C.

As mentioned in the Introduction, the quality of the basis essentially encodes the composition of the basis set. For an atomic basis, example qualities are n-zeta, where $n = 2, 3, 4,$ and 5. The quality defines the number of basis functions of each type and cannot be "less" than the minimal configuration based on the electronic ground state. For example, for a neutral nitrogen atom, the minimal basis would have a 2s1p composition, while for a gold cation, it would be 5s4p3d1f. A split-valence double-zeta "quality" basis would then add an additional exponent for each valence shell. This would give 3s2p for nitrogen, and Au^+ would be 5s4p4d1f. In the most general sense, a quality is just a function that, given an electron configuration, determines the number of exponents of each angular symmetry to use in the basis. BasisOpt provides functions for automatically generating a quality that corresponds to those used in the correlation consistent or Karlsruhe families of basis sets,[68–70] and also to define an entirely custom composition of exponents.

With the configuration of the basis determined, the next step is to generate a starting guess for the variables to be optimized. A user is free to define this guess, but examples could include an even-tempered expansion, a reference basis from the Basis Set Exchange, or even randomly generated numbers in a given interval. Finally, the

TABLE I. Summaries of the `Basis` types available in BasisOpt version 1.0.0.

Basis type	Summary	Notes
`AtomicBasis`	A traditional single-atom basis set, which is the base for all other basis types. Can use any optimization method, property, or wrapper	Minimal and even-tempered basis sets are provided as functions
`MolecularBasis`	A collection of one or more `AtomicBasis` objects, optimized against one or more molecules instead of a single atom	Can automatically handle tests as part of the optimization process
`MidbondBasis`	An `AtomicBasis` for ghost atoms placed not on atomic centers. The positions of the basis functions can optionally also be optimized, and both total energy and counterpoise-corrected energies can be used	Counterpoise-correction can only be used on energies, optimizing positions can be expensive
`JKBasis`	Auxiliary basis sets for one or more `AtomicBasis` objects, used in the density fitting of the Coulomb and exchange matrices in a self-consistent calculation. The J and K components can optionally be treated separately	Only one loss type available (see example later), and it only works with the ORCA backend currently
`OptRIBasis`	As for JKBasis, but for the resolution-of-the-identity approximation in F12 methods	Only works with the ORCA backend currently

setup checks that, if necessary, there is a suitable reference, such as a CBS energy, for whatever the objective function will be (which is dependent on the `Strategy`). Once the setup is complete, for an `AtomicBasis`, the optimization is as simple as calling optimize; for other basis set types, additional steps could be added, for example, to perform optimizations across multiple atom types or multiple molecules.

C. Optimization strategies

In essence, once the particular choice of basis is set up, the actual optimization is largely handled by external Python libraries. In particular, we call the SciPy optimize routines, and any algorithm available through SciPy is available in BasisOpt.[71] We additionally provide finite difference derivative calculators that are applied automatically if analytic derivatives are unavailable. In principle, other optimization routines could equally be used in place of those implemented in SciPy.

The outcome of the optimization is highly dependent on the choice of objective function, which in turn depends on the application. For example, density fitting auxiliary sets are sometimes optimized to minimize the error in the density, not the energy.[5,7,72,73] We provide standard preconditioner and regularization functions that can be applied to any optimization, in an effort to improve the robustness of the optimization procedure. Typical examples of regularizers are the L^1- or L^2-norms. A useful, simple preconditioner for free optimization of exponents is one that ensures that all exponents remain positive and are at least a certain ratio apart from each other within each shell. Essentially, any function that takes and returns an array of exponents can be used.

In addition, the different angular momentum shells are typically highly coupled, but optimizing all exponents at once is impractical as the number of variables to be optimized simultaneously leads to high computational costs. At the simplest level, a suitable approach is, therefore, to cyclically optimize the shells separately until they stop changing, although this can also be very expensive. To formalize this, and to allow for novel algorithms, these procedures are encapsulated in a "`Strategy`." As an example, the `Strategy` for optimizing an even-tempered basis to a given accuracy for some arbitrary, user-defined loss/objective function is outlined in Algorithm 1. Other `strategies` currently available include the default, which optimizes each angular momentum shell in turn, and one for taking an existing basis and systematically removing exponents and re-optimizing until a certain maximum allowed error is reached. That is, the basis set composition is adjusted to meet a user-defined energy or other property criterion. We note that unless an even-tempered `Strategy` is used, the current default is to freely optimize the primitive exponents in a basis set. Users of BasisOpt are able to write their own `Strategy` objects to re-create existing

ALGORITHM 1. `Strategy` for optimizing an even-tempered basis set to achieve a specified accuracy.

```
Choose absolute value-compatible property, $P$, to evaluate, desired accuracy $\epsilon$, and max. number of functions per shell, $M$
Give (or calculate) maximum angular momentum shell, $L$, needed
Guess initial prefactor, $k_i$, exponent, $\alpha_i$, and number of functions, $N_i$, for each shell, $i$
Mark each shell as not done
while Not all shells done do
    for $i = 0, \ldots, L$ do
        Expand basis function exponents as $\{k_i^n \alpha_i : n = 0, \ldots, N_i - 1\}$
        Optimize $\alpha_i$ and $k_i$ to minimize absolute value loss
        if $|\Delta P| < \epsilon$ OR $N_i = M$ then
            Mark shell as done
        else
            Set $N_i = N_i + 1$
        end if
    end for
    Store current parameters and overall loss
end while
Store final parameters and loss
```

basis set designs, or to develop their own, and we encourage their contribution to the main open source project repository.

D. Testing and visualization

Each `Basis` object stores a full history of the results of each step of the optimization, and any additional results and tests that might be desired. This allows for visualization and assessment of the optimization process. Moreover, it provides a format in which the exponents, contraction coefficients, energies, etc., can be saved for use in other contexts or to restart optimizations. Currently, these can be stored as Python pickles or in the JSON file format.

We provide a number of tests that can be added to `Basis` objects, which can be run automatically and the results aggregated as described above. There are then additional procedures for statistical analysis and visualization, designed to be compatible with standard scientific Python packages. These tests range from simple property tests—e.g., energies or multipole moments—to completeness metrics. There is also functionality to rank the contributions of individual functions, produce correlation-consistent plots, and perform Dunham analyses on diatomic molecules. These tests can very simply be run across multiple molecules and multiple basis sets, with the results automatically collated and summarized in a way that simplifies the process of basis set validation. We note that the testing of basis sets remains a vitally important component of the development process and helps determine the appropriate minimum in terms of the exponents. Thus, being able to automate the running of these tests across multiple candidate basis sets makes this process easier and aids reproducibility.

IV. POTENTIAL AND EXTENSIONS

While BasisOpt has been designed to be a flexible and easy-to-use tool for automating basis set optimization, it would be impossible to produce a framework that is applicable to all of the innumerable optimization strategies that could be used when developing new basis sets. One particular example is that the BasisOpt framework makes it difficult to re-use results from previous electronic structure calculations in subsequent calculations during the same optimization, as might be useful for particularly expensive methods. There are workarounds for this—such as by saving densities or wavefunctions to a file and pointing the next electronic structure calculation to them—but there is still an associated overhead that would not be incurred if the optimization was able to be performed natively in a given quantum chemistry package. Another potential issue is keeping backend wrappers up to date with the ever-changing code bases in the quantum chemistry community; this is not a unique issue to BasisOpt, however.

It should be noted that a few features are still in development or are currently quite rudimentary. Of particular importance are procedures for the automatic selection of contraction coefficients. This is somewhat harder to distill as a list of instructions, relying on a certain amount of "chemical intuition," insofar as that exists, and the results of significant benchmarking and testing. The `Basis` object is able to store contraction coefficients; hence, they could, in principle, be passed to an appropriate `Strategy` as variables to be optimized. For the traditional atom-optimized basis set approach, a general contraction could be generated from the density matrix coefficients,[49] but an algorithm/`Strategy` for uncontracting primitives to form a partial general contraction would need to be developed. It is also possible to envisage a `Strategy` that uses Jensen's P-orthogonalization method to automate the conversion of a general contraction basis to segmented contraction.[28] To the best of our knowledge, the generalization of Gaussian basis set contraction, whether general or segmented, to the molecular basis set case is an open research question. BasisOpt may help in this endeavor by providing the framework for user-defined contraction `strategies` to be developed and tested using the `MolecularBasis` (see Table I).

Currently, only Gaussian-type orbital exponents can be optimized in BasisOpt, as those are the only type of orbital basis set available in the external quantum chemistry programs that wrappers are provided for. Other types of LCAO basis functions, particularly Slater-type orbitals, could be optimized by developing a wrapper for an appropriate external program. This would not require any changes to the existing `Basis` object within BasisOpt.

We have aimed to design the component parts such that they are easily extensible. In particular, there are essentially no restrictions on the type of basis set or optimization procedure that can be used beyond computational resources. We hope this will allow for the rapid prototyping and development of novel basis sets and perhaps a move away from the restrictive paradigm of atom-centered, atom-optimized, Gaussian-type orbitals. We encourage anyone with interest to contribute new wrappers, basis set types, optimization strategies, and tests to the main repository so that others may benefit.

V. EXAMPLES

The following examples demonstrate how BasisOpt can be used in modern basis set development and testing. The Python code to run these examples can be downloaded from https://github.com/robashaw/basisopt, while the data are provided as the supplementary material.

A. Auxiliary basis sets

As mentioned earlier, one of the biggest modern developments in accelerating quantum chemistry calculations are density fitting approximations. These require one or more auxiliary basis sets, which typically must be optimized for each orbital basis set, which itself is different for each method. In particular, the fitting sets used for the Fock matrix integral fitting have different requirements from those used for fitting the transformed, molecular integrals in correlated methods.[7] There have been a number of heuristic approaches suggested for automatically generating auxiliary basis sets,[52,54,55] but these all necessarily produce overly large sets, so as to avoid large fitting errors. As even a well-optimized fitting set can be two or three times the size of the orbital set, there are many instances where such unoptimized fitting sets may become a computational bottleneck. However, the process for optimizing these sets is fairly opaque; therefore, as our first example, we demonstrate in Listing 2 the optimization of JK (Coulomb and exchange) fitting sets for the first three noble gases, for the cc-pVnZ orbital basis sets, n = T, Q, and 5.[68,74] For these calculations, we used the ORCA (version 5.0.3) backend, an even-tempered starting guess, free optimization of the exponents, and the default preconditioner, which simply enforces that the exponents remain positive. While the time taken to optimize any type of basis set is dependent on a large number of factors, including the electronic structure method used, the number and type of variables to be optimized, the quality of the starting guess, and so on, it may be helpful for those new to the field to provide an indication of how long the example basis set development tasks take when automated with BasisOpt. Optimizing the JK-fitting auxiliary sets for helium, neon, and argon using the full script of Listing 2 took ~3 h 10 min (wall time) to run to completion on a single core of an Apple M2 Pro. This comprised roughly 22 min to optimize the three (T, Q, and 5) auxiliary sets for He, 55 min for Ne, and 1 h 53 min for Ar.

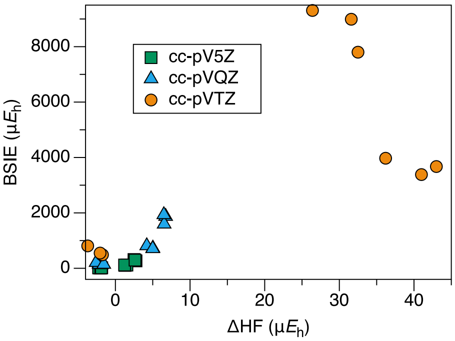


FIG. 3. The basis set incompleteness error (BSIE) vs the difference in energy between the density fitted and conventional Hartree–Fock calculations (ΔHF), both per atom. Results are across all test molecules, using the optimized JKFit auxiliary basis sets for cc-pVnZ orbital sets (n = T, Q, and 5) for the first three noble gas atoms.

LISTING 2. Sample of the Python script used to optimize JK-fitting auxiliary basis sets for helium, neon, and argon. The full script is provided as the supplementary material.

```python
from basisopt.basis.jkfit import jkfit_collection
atoms = ['He', 'Ne', 'Ar']
for atom in atoms:
        pairs = [..., (vqz, [10, 7, 5, 3, 2, 1, 0]), ...]
jk_bases = jkfit_collection(atom, guess, basis_pairs=pairs, charge=1)
for jk in jk_bases:
        bo.write_json(..., jk)
```

In the JK-fitting procedure, we usually optimize for the monohydride of the atom, and the objective function (or loss) is the root-mean-square deviation between the full, unfitted JK matrix and the fitted JK matrix. The `Strategy` we use is to optimize for the largest orbital basis (in this case, cc-pV5Z), then start optimizations for the smaller bases by systematically removing the least important functions and re-optimizing. We then calculated the fitting error for a small set of molecules (XH^+, X_2, XCO, and XCF_4, where X = He, Ne, or Ar) and compared these to approximated CBS-limit results (details in the supplementary material). The results of these tests are given in Fig. 3, and demonstrate that the fitting errors remain more than an order of magnitude smaller than the basis set incompleteness errors, as is typically required.[75]

B. Molecule-optimized basis sets

The vast majority of Gaussian basis sets for molecular applications are optimized for the energy or properties of various electronic states of atoms and then applied to molecular systems with the exponents fixed at those atom-optimized values. Some notable exceptions include the polarization consistent basis sets of Jensen and co-workers, where the exponents of the polarization functions are determined from molecular calculations,[76,77] the MOLOPT sets of VandeVondele and Hutter,[78] and the basis set included as part of the ωB97X-3c method that was optimized for a combination of atoms, ions, and hydrides.[79] Some progress has also been made in using automatic differentiation approaches to re-optimize existing basis sets for molecules.[80–82] In this second example of the usage of BasisOpt, shown as a code snippet in Listing 3 with the full script provided in the supplementary material, we will demonstrate how the package can be used to easily re-optimize the exponents of the def2-SVP basis set for the molecules H_2O, CH_4, CH_3OH, CH_2O, and O_2.[70] The geometries of these molecules were optimized at the B3LYP/cc-pVTZ level in the Molpro package.[60,83]

In this example, the def2-SVP basis set is de-contracted, and the default `Strategy` is used to construct the objective function plus apply a preconditioner to ensure all exponents are positive. For each molecule, the energy difference between the current basis and the reference (def2-QZVP in this case) is evaluated, with the objective function being the sum of the energy differences over all molecules. Within the default `Strategy` for a `MolecularBasis,` there are a number of loops, including a loop over the different atom types, looping over the angular symmetry shells for each of those atoms, and iterating through the list of molecules. In this example, we use native Python to add an additional outer loop that iterates over the rounds of atomic optimizations, which continues until the change in objective function is less than a user-defined threshold (in this case, 1 μE_h). The incremental change in energy for each iteration of this additional loop is shown in Fig. 4. The decrease in energy observed for the first iteration is due to the relaxation of the primitive exponents from the starting atomic values to molecular values and, unsurprisingly, is the largest energy change within this loop. The subsequent changes in energy are then due to the relaxation of the coupling between the different atoms and the angular symmetry shells in the basis set composition, and it is clear from the figure that this additional relaxation results in a significant decrease in energy. The final part of Listing 3 writes the optimized basis set to a file (in Molpro format).

The popular ωB97X-D exchange-correlation functional is used for all electronic structure calculations,[84] with the Psi4 program (version 1.7a1) using LibXC (version 5.1.5)[85] as the backend. Density fitting is turned off, and the example listing also demonstrates the use of the logging functionality in BasisOpt/Python, which can be used to monitor and record the progress of the optimization when used with a batch submission system. Not shown in the sample listing (see the supplementary material for the full script) is that the DFT integration grid is set to 175 radial points with 974 spherical points, and grid pruning is disabled. The script took ~9 h and 20 min to run on a single core of an Apple M2 Pro.

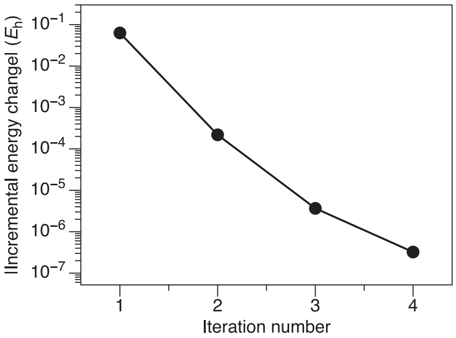


FIG. 4. Incremental change in the absolute value of the sum of the total molecular electronic energies for water, methane, methanol, formaldehyde, and oxygen as a function of a loop over the rounds of atomic basis set exponent optimizations. This outer loop probes the relaxation of the coupling between the atoms and the angular symmetry shells, with the loop terminated when the incremental change is lower than a threshold of 1 μE_h.

LISTING 3. Sample of the Python script used to optimize a basis set for five molecules. The full script is provided as the supplementary material.

```python
import basisopt as bo
from basisopt.basis.molecular import MolecularBasis
from basisopt.opt.strategies import Strategy
from basisopt.util import bo_logger
bo.set_backend('psi4')
mb = MolecularBasis(name="double")
list_of_mols = ['water', 'methane', 'methanol', 'formaldehyde', 'oxygen']
params = {
    'functional': "wb97x-d",
    'scf_type': "pk"
}
strategy = Strategy()
strategy.guess_params = {'name': 'def2-svp'}
mb.setup(method='dft', strategy=strategy, params=params, reference='def2-qzvp')
mb.optimize()
while e_diff > 1.0e-6:
    bo_logger.info("Starting consistency iteration %d", counter+1)
    mb.optimize()
    bo_logger.info("Objective function difference from previous iteration: %f\n", e_diff)
filename = "opt_basis.txt"
bo_logger.info("Writing optimized basis to %s", filename)
f.write(bo.bse_wrapper.internal_basis_converter(mb.get_basis(), fmt='molpro'))
```

The improvement in molecular energy from reoptimizing the basis set is shown in Fig. 5, which plots the basis set error, relative to def2-QZVP, for the molecules used in the basis set optimization and for acetaldehyde, which was not used in the optimization. The improvement is relatively consistent across all molecules considered, with a mean average reduction in error of 0.93 mE_h per electron. For context, the average error for the decontracted def2-SVP basis is 4.98 mE_h per electron; hence, the error is reduced by almost 20% in this reoptimization. We note that, despite this improvement, the basis set generated by Listing 3 is provided simply as an example of how BasisOpt could be used to generate a basis set optimized formolecular calculations, and the resulting exponents are not necessarily a production-ready basis set.

C. Systematic removal of functions

A common task in the development of correlation consistent basis sets is to take a large, near-saturated basis of Gaussian primitives of the angular symmetry required to describe an atom's ground state and reduce it in size until it reaches the desired criteria in terms of number of functions and HF energy. In this final example, we demonstrate how BasisOpt can be used with the `Reduce Strategy` to automate this process of reducing a large basis set for neon until it reaches the same uncontracted sp configuration as the cc-pVTZ basis. Listing 4 begins by loading the (18s13p) basis set of Partridge,[86] sometimes known as "Partridge Uncontracted 3,"

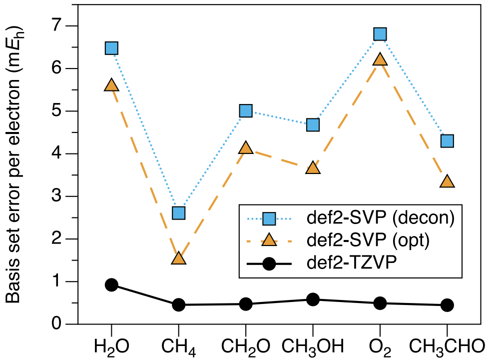


FIG. 5. Basis set error, relative to def2-QZVP, for the def2-TZVP, decontracted def2-SVP [def2-SVP (decon)], and the reoptimized def2-SVP [def2-SVP (opt)] basis sets. All molecular energies were computed using the ωB97X-D exchange-correlation functional. The acetaldehyde molecule was not used in the basis set optimization.

from the Basis Set Exchange, then systematically removing functions and freely re-optimizing exponents until the basis reaches (10s5p). The choice of function to remove is automated by computing the incremental change in energy if each individual function is removed (without re-optimization) and subsequently ranking the functions for removal. Although a specific basis set size is requested in this example, the reduction algorithm can alternatively be instructed to stop when the incremental change in energy from removing a function exceeds a user-provided threshold. Listing 4 also demonstrates some of the visualization functionality of BasisOpt in the form of plotting basis set exponents.

LISTING 4. Sample of the Python script used to reduce the Partridge Uncontracted 3 basis set until the basis set size is (10s5p). The full script is provided as the supplementary material.

```python
from basisopt.basis.atomic import AtomicBasis
from basisopt.bse_wrapper import fetch_basis
from basisopt.opt.reduce import ReduceStrategy
import matplotlib.pyplot as plt
from basisopt.viz.basis import plot_exponents
bo.set_backend('psi4')
ne = AtomicBasis('Ne')
basis_name = 'Partridge Uncontracted 3'
start_basis = fetch_basis(basis_name, ['Ne'])
strategy = ReduceStrategy(start_basis,
                          eval_type='energy', method='scf',
                          target=5e-1, shell_mins=[10, 5])
ne.setup(method='scf', strategy=strategy, params=params,
         reference=(basis_name, None))
res=ne.optimize()
fig, ax = plot_exponents(start_basis, atoms=['Ne'])
```

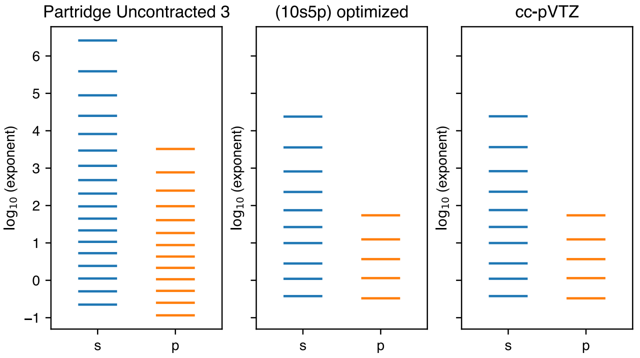


FIG. 6. Comparison of the (10s5p) exponents optimized for neon using Listing 4 with those from the Partridge Uncontracted 3 basis, and the s and p functions from the correlation consistent cc-pVTZ basis.

Using the basis set optimized by BasisOpt and Listing 4 in Psi4 to calculate the HF ground state energy of the neon atom results in a total energy of −128.531 863E_h, which is effectively the same as the −128.531 862E_h from using the (10s5p) extracted from cc-pVTZ. The script took ~1 h and 20 min to run to completion on a single core of an Apple M2 Pro. Figure 6 compares the exponents optimized via Listing 4 with those of the Partridge Uncontracted 3 set that was used as the starting point and also with the s and p functions taken from the cc-pVTZ basis. There are large changes relative to the basis of Partridge; several tighter exponents have been removed, along with some in the valence-space of the basis, and overall the exponents are significantly less densely packed. However, it is immediately obvious that the exponents generated by the reduction `Strategy` in BasisOpt are very close to those from cc-pVTZ, with the exponents spanning a similar range and with the same spacing between adjacent exponents. This result demonstrates the effectiveness of BasisOpt in automating the reduction of large basis sets to a more computationally efficient set of primitives with specified energy and/or composition criteria.

VI. CONCLUSIONS

We have presented a Python package for the automated optimization of quantum chemical basis sets for use in molecular applications. It is primarily focused, at the time of writing, on primitive exponents of Gaussian-type orbitals, additionally providing tools for testing and visualizing the resulting basis sets. The use of general wrappers and basis set `strategies` provides a flexible framework that should allow for new, more efficient basis sets to be developed for a wide range of methods, particularly in packages where there is no built-in general purpose optimizer. However, even for more traditional optimizations, opening up the process to the full power of the Python ecosystem should allow for interesting new and more robust optimization procedures. We have given a high-level description of the constituent parts of BasisOpt and demonstrated a few example use cases. More detailed documentation and examples can be found in the main code repository. The code is fully open source, and contributions and feedback from the wider community are encouraged.

SUPPLEMENTARY MATERIAL

See the supplementary material for data used in examples A and B (CSV) and for Python code listings (PDF). The code listings are also included as examples in the GitHub repo.

ACKNOWLEDGMENTS

The authors thank the UK Engineering and Physical Sciences Research Council (EPSRC) for support through Grant No. EP/T027134/1. The authors also thank Susi Lehtola for his useful discussions in the early stages of this project and for his detailed comments on an initial draft of the manuscript.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Robert A. Shaw: Conceptualization (lead); Data curation (lead); Formal analysis (lead); Investigation (equal); Software (lead); Visualization (supporting); Writing – original draft (lead); Writing – review & editing (supporting). **J. Grant Hill**: Funding acquisition (lead); Investigation (equal); Resources (lead); Software (supporting); Visualization (lead); Writing – original draft (supporting); Writing – review & editing (lead).

DATA AVAILABILITY

The data that support the findings of this study are available within the article and its supplementary material.

REFERENCES

[1] M. Sparta and F. Neese, "Chemical applications carried out by local pair natural orbital based coupled-cluster methods," Chem. Soc. Rev. **43**, 5032–5041 (2014).

[2] Q. Ma and H.-J. Werner, "Explicitly correlated local coupled-cluster methods using pair natural orbitals," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **8**, e1371 (2018).

[3] J. L. Whitten, "Coulombic potential energy integrals and approximations," J. Chem. Phys. **58**, 4496–4501 (1973).

[4] B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, "On some approximations in applications of Xα theory," J. Chem. Phys. **71**, 3396–3402 (1979).

[5] O. Vahtras, J. Almlöf, and M. W. Feyereisen, "Integral approximations for LCAO-SCF calculations," Chem. Phys. Lett. **213**, 514–518 (1993).

[6] M. W. Feyereisen, G. Fitzgerald, and A. Komornicki, "Use of approximate integrals in ab initio theory. An application in MP2 energy calculations," Chem. Phys. Lett. **208**, 359–363 (1993).

[7] F. Weigend, "A fully direct RI-HF algorithm: Implementation, optimised auxiliary basis sets, demonstration of accuracy and efficiency," Phys. Chem. Chem. Phys. **4**, 4285–4291 (2002).

[8] F. Aquilante, T. B. Pedersen, and R. Lindh, "Low-cost evaluation of the exchange Fock matrix from Cholesky and density fitting representations of the electron repulsion integrals," J. Chem. Phys. **126**, 194106 (2007).

[9] D. G. Liakos and F. Neese, "Is it possible to obtain coupled cluster quality energies at near density functional theory cost? Domain-based local pair natural orbital coupled cluster vs modern density functional theory," J. Chem. Theory Comput. **11**, 4054–4063 (2015).

[10] D. G. Liakos, M. Sparta, M. K. Kesharwani, J. M. L. Martin, and F. Neese, "Exploring the accuracy limits of local pair natural orbital coupled-cluster theory," J. Chem. Theory Comput. **11**, 1525–1539 (2015).

[11] L. Goerigk and S. Grimme, "Double-hybrid density functionals," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **4**, 576–600 (2014).

[12] C. Bannwarth, E. Caldeweyher, S. Ehlert, A. Hansen, P. Pracht, J. Seibert, S. Spicher, and S. Grimme, "Extended tight-binding quantum chemistry methods," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **11**, e1493 (2021).

[13] E. R. Davidson and D. Feller, "Basis set selection for molecular calculations," Chem. Rev. **86**, 681–696 (1986).

[14] F. Jensen, *Introduction to Computational Chemistry*, 2nd ed. (John Wiley & Sons Ltd., Chichester, 2007), pp. 192–231, Chap. 5.

[15] F. Jensen, "Atomic orbital basis sets," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **3**, 273–295 (2013).

[16] J. G. Hill, "Gaussian basis sets for molecular applications," Int. J. Quantum Chem. **113**, 21–34 (2013).

[17] B. Nagy and F. Jensen, "Basis sets in quantum chemistry," in *Reviews in Computational Chemistry*, edited by A. L. Parrill and K. B. Lipkowitz (John Wiley & Sons, Inc., Hoboken, NJ, 2017), Vol. 30, pp. 93–149.

[18] S. Lehtola, "A review on non-relativistic, fully numerical electronic structure calculations on atoms and diatomic molecules," Int. J. Quantum Chem. **199**, e25968 (2019).

[19] S. F. Boys, "Electronic wave functions—I. A general method of calculation for the stationary states of any molecular system," Proc. R. Soc. A **200**, 542–554 (1950).

[20] F. Tao, "The use of midbond functions for *ab initio* calculations of the asymmetric potentials of He–Ne and He–Ar," J. Chem. Phys. **98**, 3049–3059 (1993).

[21] R. A. Shaw and J. G. Hill, "Midbond basis functions for weakly bound complexes," Mol. Phys. **116**, 1460–1470 (2018).

[22] R. Matveeva, M. Falck Erichsen, H. Koch, and I. Høyvik, "The effect of midbond functions on interaction energies computed using MP2 and CCSD(T)," J. Comput. Chem. **43**, 121–131 (2022).

[23] T. Helgaker and J. Almlöf, "Molecular wave functions and properties calculated using floating Gaussian orbitals," J. Chem. Phys. **89**, 4889–4902 (1988).

[24] F. Jensen, "Contracted basis sets for density functional calculations: Segmented versus general contraction," J. Chem. Phys. **122**, 074111 (2005).

[25] R. C. Raffenetti, "General contraction of Gaussian atomic orbitals: Core, valence, polarization, and diffuse basis sets; molecular integral evaluation," J. Chem. Phys. **58**, 4452–4458 (1973).

[26] E. Clementi and D. R. Davis, "Barrier to internal rotation in ethane," J. Chem. Phys. **45**, 2593–2599 (1966).

[27] T. H. Dunning, Jr., "Gaussian basis functions for use in molecular calculations. I. Contraction of (9s5p) atomic basis sets for the first-row atoms," J. Chem. Phys. **53**, 2823–2833 (1970).

[28] F. Jensen, "Unifying general and segmented contracted basis sets. Segmented polarization consistent basis sets," J. Chem. Theory Comput. **10**, 1074–1085 (2014).

[29] W. Klopper and W. Kutzelnigg, "Gaussian basis sets and the nuclear cusp problem," J. Mol. Struct.: THEOCHEM **135**, 339–356 (1986).

[30] W. Kutzelnigg, "Theory of the expansion of wave functions in a Gaussian basis," Int. J. Quantum Chem. **51**, 447–463 (1994).

[31] W. Kutzelnigg, "Expansion of a wave function in a Gaussian basis. I. Local versus global approximation," Int. J. Quantum Chem. **113**, 203–217 (2013).

[32] R. A. Shaw, "The completeness properties of Gaussian-type orbitals in quantum chemistry," Int. J. Quantum Chem. **120**, e26264 (2020).

[33] K. A. Peterson, "Chapter 11: Gaussian basis sets exhibiting systematic convergence to the complete basis set limit," Annu. Rep. Comput. Chem. **3**, 195–206 (2007).

[34] D. Feller, "Benchmarks of improved complete basis set extrapolation schemes designed for standard CCSD(T) atomization energies," J. Chem. Phys. **138**, 074103 (2013).

[35] F. Pansini, A. Neto, and A. Varandas, "On the performance of various hierarchized bases in extrapolating the correlation energy to the complete basis set limit," Chem. Phys. Lett. **641**, 90–96 (2015).

[36] W. Kutzelnigg and J. D. Morgan, "Rates of convergence of the partial-wave expansions of atomic correlation energies," J. Chem. Phys. **96**, 4484–4508 (1992).

[37] D. P. Chong, E. Van Lenthe, S. Van Gisbergen, and E. J. Baerends, "Even-tempered slater-type orbitals revisited: From hydrogen to krypton," J. Comput. Chem. **25**, 1030–1036 (2004).

[38] L. Kong, F. A. Bischoff, and E. F. Valeev, "Explicitly correlated R12/F12 methods for electronic structure," Chem. Rev. **112**, 75–107 (2012).

[39] B. P. Pritchard, D. Altarawy, B. Didier, T. D. Gibson, and T. L. Windus, "New basis set exchange: An open, up-to-date resource for the molecular sciences community," J. Chem. Inf. Model. **59**, 4814–4820 (2019).

[40] The Molecular Sciences Software Institute, Basis set exchange, https://www.basissetexchange.org (accessed 22 June 2023).

[41] J. G. Hill, ccRepo: A correlation consistent basis sets repository, http://www.grant-hill.group.shef.ac.uk/ccrepo/ (accessed 22 June 2023).

[42] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. (John Wiley & Sons, Ltd., Chichester, West Sussex, 2000).

[43] P. Manninen and J. Vaara, "Systematic Gaussian basis-set limit using completeness-optimized primitive sets. A case for magnetic properties," J. Comput. Chem. **27**, 434–445 (2006).

[44] D. Rappoport and F. Furche, "Property-optimized Gaussian basis sets for molecular response calculations," J. Chem. Phys. **133**, 134105 (2010).

[45] F. Jensen, "The basis set convergence of the Hartree–Fock energy for H_3^+, Li_2 and N_2," Theor. Chem. Acc. **104**, 484–490 (2000).

[46] K. Ruedenberg, R. C. Raffenetti, and R. D. Bardo, *Energy, Structure and Reactivity. Proceedings of the 1972 Boulder Summer Research Conference on Theoretical Chemistry*, edited by D. W. Smith and W. B. McRae (John Wiley & Sons, Inc., New York, 1973), pp. 164–169.

[47] S. Huzinaga and M. Klobukowski, "Well-tempered GTF basis sets for the atoms K through Xe," Chem. Phys. Lett. **120**, 509–512 (1985).

[48] G. A. Petersson, S. Zhong, J. A. Montgomery, Jr., and M. J. Frisch, "On the optimization of Gaussian basis sets," J. Chem. Phys. **118**, 1101–1109 (2003).

[49] J. Almlöf and P. R. Taylor, "General contraction of Gaussian basis sets. I. Atomic natural orbitals for first- and second-row atoms," J. Chem. Phys. **86**, 4070–4077 (1987).

[50] P.-O. Widmark, P.-Å. Malmqvist, and B. O. Roos, "Density matrix averaged atomic natural orbital (ANO) basis sets for correlated molecular wave functions," Theor. Chim. Acta **77**, 291–306 (1990).

[51] E. R. Davidson, "Comment on 'Comment on Dunning's correlation-consistent basis sets,'" Chem. Phys. Lett. **260**, 514–518 (1996).

[52] F. Aquilante, R. Lindh, and T. B. Pedersen, "Unbiased auxiliary basis sets for accurate two-electron integral approximations," J. Chem. Phys. **127**, 114107 (2007).

[53] R. Yang, A. P. Rendell, and M. J. Frisch, "Automatically generated Coulomb fitting basis sets: Design and accuracy for systems containing H to Kr," J. Chem. Phys. **127**, 074102 (2007).

[54] G. L. Stoychev, A. A. Auer, and F. Neese, "Automatic generation of auxiliary basis sets," J. Chem. Theory Comput. **13**, 554–562 (2017).

[55] S. Lehtola, "Straightforward and accurate automatic auxiliary basis set generation for molecular calculations with atomic orbital basis sets," J. Chem. Theory Comput. **17**, 6886–6900 (2021).

[56] F. Weigend, A. Köhn, and C. Hättig, "Efficient use of the correlation consistent basis sets in resolution of the identity MP2 calculations," J. Chem. Phys. **116**, 3175–3183 (2002).

[57] W. Klopper and C. C. M. Samson, "Explicitly correlated second-order Møller–Plesset methods with auxiliary basis sets," J. Chem. Phys. **116**, 6397–6410 (2002).

[58] E. F. Valeev, "Improving on the resolution of the identity in linear R12 ab initio theories," Chem. Phys. Lett. **395**, 190–195 (2004).

[59] K. E. Yousaf and K. A. Peterson, "Optimized auxiliary basis sets for explicitly correlated methods," J. Chem. Phys. **129**, 184108 (2008).

[60] H.-J. Werner, P. J. Knowles, F. R. Manby, J. A. Black, K. Doll, A. Heßelmann, D. Kats, A. Köhn, T. Korona, D. A. Kreplin, Q. Ma, T. F. Miller III, A. Mitrushchenkov, K. A. Peterson, I. Polyak, G. Rauhut, and M. Sibaev, "The Molpro quantum chemistry package," J. Chem. Phys. **152**, 144107 (2020).

[61] G. Schoendorff, "Development of an open source tool for basis set optimizations and application to the development of a quadruple-ζ segmented basis set for actinium," in *Rare Earth Elements and Actinides: Progress in Computational Science Applications*, *ACS Symposium Series* (American Chemical Society, 2021), Vol. 1388, pp. 329–346, Chap. 15.

[62] G. M. J. Barca, C. Bertoni, L. Carrington, D. Datta, N. De Silva, J. E. Deustua, D. G. Fedorov, J. R. Gour, A. O. Gunina, E. Guidez, T. Harville, S. Irle, J. Ivanic, K. Kowalski, S. S. Leang, H. Li, W. Li, J. J. Lutz, I. Magoulas, J. Mato, V. Mironov, H. Nakata, B. Q. Pham, P. Piecuch, D. Poole, S. R. Pruitt, A. P. Rendell, L. B. Roskop, K. Ruedenberg, T. Sattasathuchana, M. W. Schmidt, J. Shen, L. Slipchenko, M. Sosonkina, V. Sundriyal, A. Tiwari, J. L. Galvez Vallejo, B. Westheimer, M. Włoch, P. Xu, F. Zahariev, and M. S. Gordon, "Recent developments in the general atomic and molecular electronic structure system," J. Chem. Phys. **152**, 154102 (2020).

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

[64] F. Neese, F. Wennmohs, U. Becker, and C. Riplinger, "The ORCA quantum chemistry program package," J. Chem. Phys. **152**, 224108 (2020).

[65] N. M. O'Boyle, A. L. Tenderholt, and K. M. Langer, "cclib: a library for package-independent computational chemistry algorithms," J. Comput. Chem. **29**, 839–845 (2008).

[66] E. Berquist, G. Hutchison, K. M. Langer, N. M. O'Boyle, A. L. Tenderholt, and S. Upadhyay (2021). "cclib," Zenodo, Version 1.7, Dataset https://doi.org/10.5281/ZENODO.4420433.

[67] Ł. Mentel (2014). "Mendeleev—A Python resource for properties of chemical elements, ions and isotopes," Zenodo, Version 0.12.1, Dataset https://doi.org/10.5281/ZENODO.5233824.

[68] T. H. Dunning, Jr., "Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen," J. Chem. Phys. **90**, 1007–1023 (1989).

[69] A. Schäfer, H. Horn, and R. Ahlrichs, "Fully optimized contracted Gaussian basis sets for atoms Li to Kr," J. Chem. Phys. **97**, 2571–2577 (1992).

[70] F. Weigend and R. Ahlrichs, "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. **7**, 3297–3305 (2005).

[71] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, A. Vijaykumar, A. P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C. N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D. A. Nicholson, D. R. Hagen, D. V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G. A. Price, G.-L. Ingold, G. E. Allen, G. R. Lee, H. Audren, I. Probst, J. P. Dietrich, J. Silterra, J. T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulick, J. L. Schönberger, J. V. de Miranda Cardoso, J. Reimer, J. Harrington, J. L. C. Rodríguez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmerer, M. Bolingbroke, M. Tartre, M. Pak, N. J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P. A. Brodtkorb, P. Lee, R. T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T. J. Pingel, T. P. Robitaille, T. Spura, T. R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y. O. Halchenko, and Y. Vázquez-Baeza, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," Nat. Methods **17**, 261–272 (2020).

[72] K. Eichkorn, O. Treutler, H. Öhm, M. Häser, and R. Ahlrichs, "Auxiliary basis sets to approximate Coulomb potentials," Chem. Phys. Lett. **240**, 283–290 (1995).

[73] K. Eichkorn, F. Weigend, O. Treutler, and R. Ahlrichs, "Auxiliary basis sets for main row atoms and transition metals and their use to approximate Coulomb potentials," Theor. Chem. Acta. **97**, 119–124 (1997).

[74] D. E. Woon and T. H. Dunning, Jr., "Gaussian basis sets for use in correlated molecular calculations. III. The atoms aluminum through argon," J. Chem. Phys. **98**, 1358–1371 (1993).

[75] F. Weigend, "Hartree–Fock exchange fitting basis sets for H to Rn," J. Comput. Chem. **29**, 167–175 (2008).

[76] F. Jensen, "Polarization consistent basis sets: Principles," J. Chem. Phys. **115**, 9113–9125 (2001).

[77] F. Jensen, "Polarization consistent basis sets. II. Estimating the Kohn–Sham basis set limit," J. Chem. Phys. **116**, 7372–7379 (2002).

[78] J. VandeVondele and J. Hutter, "Gaussian basis sets for accurate calculations on molecular systems in gas and condensed phases," J. Chem. Phys. **127**, 114105 (2007).

[79] M. Müller, A. Hansen, and S. Grimme, "ωB97X-3c: A composite range-separated hybrid DFT method with a molecule-optimized polarized valence double-ζ basis set," J. Chem. Phys. **158**, 014103 (2023).

[80] M. Tachikawa, K. Taneda, and K. Mori, "Simultaneous optimization of GTF exponents and their centres with fully variational treatment of Hartree–Fock molecular orbital calculation," Int. J. Quantum Chem. **75**, 497–510 (1999).

[81] T. Tamayo-Mendoza, C. Kreisbeck, R. Lindh, and A. Aspuru-Guzik, "Automatic differentiation in quantum chemistry with applications to fully variational Hartree–Fock," ACS Cent. Sci. **4**, 559–566 (2018).

[82] M. F. Kasim, S. Lehtola, and S. M. Vinko, "DQC: A Python program package for differentiable quantum chemistry," J. Chem. Phys. **156**, 084801 (2022).

[83] H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, M. Schütz, P. Celani, W. Györffy, D. Kats, T. Korona, R. Lindh, A. Mitrushenkov, G. Rauhut, K. R. Shamasundar, T. B. Adler, R. D. Amos, S. J. Bennie, A. Bernhardsson, A. Berning, D. L. Cooper, M. J. O. Deegan, A. J. Dobbyn, F. Eckert, E. Goll, C. Hampel, A. Hesselmann, G. Hetzer, T. Hrenar, G. Jansen, C. Köppl, S. J. R. Lee, Y. Liu, A. W. Lloyd, Q. Ma, R. A. Mata, A. J. May, S. J. McNicholas, W. Meyer, T. F. Miller III, M. E. Mura, A. Nicklass, D. P. O'Neill, P. Palmieri, D. Peng, T. Petrenko, K. Pflüger, R. Pitzer, M. Reiher, T. Shiozaki, H. Stoll, A. J. Stone, R. Tarroni, T. Thorsteinsson, M. Wang, and M. Welborn, MOLPRO, version 2021.2.1, a package of *ab initio* programs, 2021, see http://www.molpro.net.

[84] J.-D. Chai and M. Head-Gordon, "Long-range corrected hybrid density functionals with damped atom-atom dispersion corrections," Phys. Chem. Chem. Phys. **10**, 6615–6620 (2008).

[85] S. Lehtola, C. Steigemann, M. J. T. Oliveira, and M. A. L. Marques, "Recent developments in LIBXC—A comprehensive library of functionals for density functional theory," SoftwareX **7**, 1–5 (2018).

[86] H. Partridge, "Near Hartree–Fock quality GTO basis sets for the first- and third-row atoms," J. Chem. Phys. **90**, 1043–1047 (1989).