



HAL
open science

QCMATH: Mathematica modules for electronic structure calculations

Enzo Monino, Antoine Marie, Pierre-François Loos

► **To cite this version:**

Enzo Monino, Antoine Marie, Pierre-François Loos. QCMATH: Mathematica modules for electronic structure calculations. 2023. hal-04191016

HAL Id: hal-04191016

<https://hal.science/hal-04191016v1>

Preprint submitted on 30 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

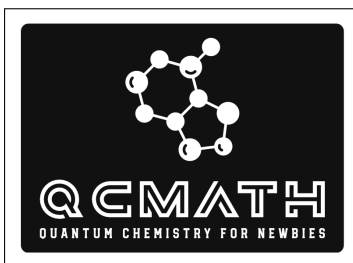
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

QCMATH: Mathematica modules for electronic structure calculations

Enzo Monino,^{1, a)} Antoine Marie,^{1, b)} and Pierre-François Loos^{1, c)}

Laboratoire de Chimie et Physique Quantiques (UMR 5626), Université de Toulouse, CNRS, UPS, France

We introduce QCMATH, a user-friendly quantum chemistry software tailored for electronic structure calculations, implemented using the Wolfram Mathematica language and available at <https://github.com/LCPQ/qcmath>. This software, designed with accessibility in mind, takes advantage of the symbolic capabilities intrinsic to Mathematica. Its primary goal is to provide a supportive environment for newcomers to the field of quantum chemistry, enabling them to easily conceptualize, develop, and test their own ideas. The functionalities of QCMATH encompass a broad spectrum of methods, catering to both ground- and excited-state calculations. We provide a comprehensive overview of these capabilities, complemented by essential theoretical insights. To facilitate ease of use, we offer an exhaustive blueprint of the software's architecture. Furthermore, we provide users with comprehensive guides, addressing both the operational aspects and the more intricate programming facets of QCMATH.



I. INTRODUCTION

Quantum chemistry methods are highly compatible with computer utilization, owing to the matrix formulation of quantum mechanics, which leverages the power of linear algebra packages like **BLAS** and **LAPACK**. As a result, a wide array of quantum chemistry software is currently available, encompassing both free and commercial options. These software packages cater to specific methods or offer a diverse range of methodologies, utilizing various types of basis functions, among other features. A considerable number of quantum chemistry codes exist, covering a comprehensive range of methods. For a comprehensive list of these codes, refer to [Wikipedia's page](#) on quantum chemistry and solid-state physics software.

Regrettably, despite the efficient design of many of these software packages, they can be challenging to comprehend as they often employ low-level programming languages. Moreover, these programs are not primarily intended for educational purposes or for facilitating understanding. This is precisely where **QCMATH** comes into play. QCMATH aims to assist newcomers in the field of quantum chemistry by providing a user-friendly platform for developing ideas and codes. It is worth noting that certain software packages utilize higher-level programming languages, which can enhance code comprehension. As for QCMATH, it is a compilation of **MATHEMATICA**

modules specifically designed for conducting electronic structure calculations.

Before delving into the specifics of QCMATH, let us provide an overview of the **MATHEMATICA** environment. **MATHEMATICA** is a comprehensive software system developed by Wolfram Research, initially conceptualized by Stephen Wolfram. It boasts a wide range of built-in libraries that can be utilized for diverse purposes. One of its key strengths lies in its ability to perform computer algebra operations, such as derivatives, integrals, and expression simplifications. Furthermore, **MATHEMATICA** enables the numerical evaluation of these expressions. Another notable feature is its advanced plotting capabilities, supporting intricate visualizations of functions in one, two, and three dimensions. Numerous books offer extensive examples of **MATHEMATICA**'s applications across various domains. With its versatility, **MATHEMATICA** has become a powerful tool employed in numerous scientific fields, including education, research, and industry.

MATHEMATICA comprises two main components: the kernel and the front end. The kernel interprets expressions and generates result expressions, which can then be displayed using the front end. The original front end takes the form of a notebook interface, facilitating the creation and editing of notebook documents that can contain code, plaintext, images, and graphics. QCMATH, specifically, relies on these notebook documents. It is important to note that QCMATH is **not** primarily designed for computational efficiency but rather focuses on providing a user-friendly environment.

^{a)} Electronic mail: emonino@irsamc.ups-tlse.fr

^{b)} Electronic mail: amarie@irsamc.ups-tlse.fr

^{c)} Electronic mail: loos@irsamc.ups-tlse.fr

II. INSTALLATION GUIDE

The QCMATH software can be downloaded on GITHUB as a git repository

```
git clone https://github.com/LCPQ/qcmath.git
```

Then, one must define the variable `QCMATH_ROOT` and install `PySCF` using `pip`

```
pip install pyscf
```

PySCF is used for the computation of one- and two-electron integrals. Here is the list of the requirements to use QCMATH:

- Linux or Mac OS
- Wolfram Mathematica \geq 12.1
- PySCF
- Python \geq 3.6.0
- Numpy \geq 1.13

Note that the version of Python and Numpy is fixed by PySCF.

III. QUICK START

Before running any QCMATH calculation, we need to define the working directory as

```
1 SetDirectory[NotebookDirectory[]];
2 path=Directory[];
3 py="your_path_to_python"
4 NotebookEvaluate[path<>"/src/Main/Main.nb"]
```

To streamline the execution of other notebooks and prevent the need for directory changes, the first line of code sets the working directory as the directory containing the notebook. This ensures a seamless evaluation process. Following that, the second line establishes the variable `path` as the current directory, which corresponds to the working directory. The third line designates the path to your Python installation, allowing for appropriate configuration. Lastly, the main notebook is evaluated, with the inclusion of the `path` variable to locate the correct directory. This approach ensures smooth execution and seamless integration of the required files.

Once this first step is done, one can run a QCMATH calculation as follows

```
1 qcmath[molecule_name,basis_set,methods]
```

To invoke the QCMATH module, use the keyword `qcmath` followed by three arguments. These arguments are either strings or a list of strings. The first argument is the name of the molecule to be studied, represented as a string. For example, in the case of the H_2 molecule, it would be specified as `"H2"`. The second argument corresponds to the basis set and is also provided as a string. For instance, in the example of the 6-31G basis set, it would be specified as `"6-31g"`. In summary, taking the example of the H_2 molecule in the 6-31G basis set using the restricted Hartree-Fock method, the QCMATH module call would resemble the following code:

```
1 qcmath["H2","6-31g",{ "RHF" }]
```

The molecular geometry is provided through a `.xyz` file located in the `mol` directory, while the basis set file is stored in the `basis` directory. It is also possible to directly enter the geometry of the molecule as an input. Here is an example for the H_2 molecule:

```
1 qcmath[{{"H",0.0,0.0,0.0},{ "H",0.0,0.0,1.0}}, "6-31g",{ "RHF" }]
```

First, one must specify the atom as a string and, then, the xyz coordinates.

Additional options can be specified, such as the charge and spin multiplicity of the molecule. If these options are not explicitly stated, the default values are zero for the charge (neutral) and singlet state for the spin multiplicity. Furthermore, options related to different methods can also be specified, but we will discuss them in the upcoming section. It is worth noting that most of the presented methods offer both spin and spatial orbital implementations. You can choose between them using the keyword `"spinorbital"`, with the default value being `False` (indicating spatialorbital as the default choice). For a comprehensive list of all available options, including charge, spin multiplicity, and method-related choices, please refer to the `Main/default_options.nb` notebook. This notebook presents the options in the form of a dictionary, providing a convenient reference for configuring and customizing the calculations.

IV. USER GUIDE

The QCMATH software is currently undergoing active development. The features discussed below are currently available, and they represent the initial roadmap for the software's future. This User Guide provides a comprehensive introduction to the underlying theoretical concepts and showcases the functionalities offered by these methods.¹⁻⁵ It serves as a valuable resource to gain insights into the theoretical background and explore the capabilities that will be incorporated into QCMATH as it continues to evolve.

A. Ground-state calculations

Hartree-Fock theory

In the context of the Hartree-Fock (HF) approximation, the electronic wave function is expressed as a Slater determinant comprising N one-electron orbitals.¹ Within the restricted HF (RHF) formalism, the Roothaan-Hall equations come into play, given by $\mathbf{F} \cdot \mathbf{C} = \mathbf{S} \cdot \mathbf{C} \cdot \boldsymbol{\varepsilon}$. Here, \mathbf{F} represents the Fock matrix, \mathbf{C} denotes the matrix of orbital coefficients, \mathbf{S} stands for the matrix representing atomic orbital overlaps, and $\boldsymbol{\varepsilon}$ is a diagonal matrix containing the orbital energies. Since the Fock matrix relies on the orbital coefficients \mathbf{C} , which are obtained from the Fock matrix itself, these equations necessitate a self-consistent solution. To facilitate this process, various options can be specified to customize the calculations and achieve desired outcomes:

- The initial guess of the Fock matrix that needs to be diagonalized to give the orbital coefficients. This initial guess is described by the keyword `"guess_type"`
 - `"guess_type"="core"` (default) corresponds to the core Hamiltonian defined as $\mathbf{H}^c = \mathbf{T} + \mathbf{V}$, where \mathbf{T} is the kinetic energy matrix and \mathbf{V} is the external potential.
 - `"guess_type"="huckel"` corresponds to the Hückel Hamiltonian.
 - `"guess_type"="random"` corresponds to random orbital coefficients.
- Options to control the convergence of HF calculations
 - `"maxSCF"`: maximum number of iterations, by default `"maxSCF"=100`.
 - `"threshHF"`: convergence threshold on the commutator $\mathbf{F} \cdot \mathbf{P} \cdot \mathbf{S} - \mathbf{S} \cdot \mathbf{P} \cdot \mathbf{F}$ where \mathbf{P} is the density matrix, by default `"threshHF"=10-7`.
 - `"DIIS"`: rely on the Direct Inversion in the Iterative Subspace (DIIS) where the Fock matrix is extrapolated at each iteration using the ones of the previous iterations, by default `"DIIS"=True`.
 - `"n_DIIS"`: size of the DIIS space, by default `"n_DIIS"=5`.
 - `"level_shift"`: a level shift increases the gap between the occupied and virtual orbitals, it can help to converge the SCF process for systems with a small HOMO-LUMO gap, by default `"level_shift"=0`.

- Orthogonalization matrix with the keyword `"ortho_type"`.
 - `"ortho_type"="lowdin"` (default): Löwdin orthogonalization.
 - `"ortho_type"="canonical"`: Canonical orthogonalization.

- Print additional information about the calculation with the keyword `"verbose"`
 - `"verbose"=False` by default, if `"verbose"=True` then more information about the CPU timing and additional quantities are printed. Note that this option is available for most methods in QCMATH.

Two flavors of Hartree-Fock (HF) are available in qcmath: restricted HF (RHF) and unrestricted HF (UHF). To run a UHF calculation, one simply does

```
1 qcmath["H2", "6-31g", {"UHF"}]
```

Møller-Plesset perturbation theory

The second-order Møller-Plesset (MP2) correlation energy is defined by

$$E_c^{\text{MP2}} = \frac{1}{4} \sum_{ij}^{\text{occ}} \sum_{ab}^{\text{vir}} \frac{|\langle ij||ab \rangle|^2}{\epsilon_i^{\text{HF}} + \epsilon_j^{\text{HF}} - \epsilon_a^{\text{HF}} - \epsilon_b^{\text{HF}}} \quad (1)$$

where $\langle pq||rs \rangle = \langle pq|rs \rangle - \langle pq|sr \rangle$ are antisymmetrized two-electron integrals (in Dirac notations) in the spinorbital basis and the ϵ_p^{HF} 's are the HF orbital energies. From here on, i, j, \dots are occupied spinorbitals, a, b, \dots denote virtual (unoccupied) spinorbitals, and $p, q, r,$ and s indicate arbitrary (orthonormal) spinorbitals. Since MP2 needs HF quantities, first, a HF calculation needs to be done. This is automatically taken into account by QCMATH and an MP2 calculation can be done using

```
1 qcmath["H2", "6-31g", {"RHF", "MP2"}]
```

or

```
1 qcmath["H2", "6-31g", {"MP2"}]
```

Note that in the last case, a RHF is performed by default so if one wants to rely on a UHF reference, one has to run

```
1 qcmath["H2", "6-31g", {"UHF", "MP2"}]
```

Configuration interaction methods

One of the most conceptually simple (albeit expensive) approaches to recovering a large fraction of the

correlation energy is the configuration interaction (CI) method.¹⁻³ The general idea is to expand the wave function as a linear combination of "excited" determinants. These excited determinants are built by promoting electrons from occupied to unoccupied (virtual) orbitals usually based on the HF orbitals, i.e.

$$|\Psi_{\text{CI}}\rangle = c_0 |\Psi_0\rangle + \sum_i \sum_a^{\text{occ virt}} c_i^a |\Psi_i^a\rangle + \sum_{ij} \sum_{ab}^{\text{occ virt}} c_{ij}^{ab} |\Psi_{ij}^{ab}\rangle + \sum_{ijk} \sum_{abc}^{\text{occ virt}} c_{ijk}^{abc} |\Psi_{ijk}^{abc}\rangle + \dots \quad (2)$$

where $|\Psi_i^a\rangle$, $|\Psi_{ij}^{ab}\rangle$ and $|\Psi_{ijk}^{abc}\rangle$ are singly-, doubly- and triply-excited determinants. $|\Psi_{ij}^{ab}\rangle$ corresponds to the excitations of two electrons from the occupied spinorbitals i and j to virtual spinorbitals a and b . It is easy to show that the CI energy

$$E_{\text{CI}} = \frac{\langle \Psi_{\text{CI}} | \hat{H} | \Psi_{\text{CI}} \rangle}{\langle \Psi_{\text{CI}} | \Psi_{\text{CI}} \rangle} \quad (3)$$

is an upper bound to the exact energy of the system.

When all possible excitations are taken into account, the method is called full CI (FCI) and it recovers the entire correlation energy for a given basis set. Albeit elegant, FCI is very expensive due to the exponential increase of the number of excited determinants. For example, when only singles and doubles are taken into account, the method is called CISD. It recovers an important chunk of the correlation. However, it has the disadvantage to be size-inconsistent.⁶ Two CI methods are available in qcmath, the FCI and CISD methods. To run these calculations, method keyword need to be specified:

- **"CISD"**: run a CISD calculation
- **"FCI"**: run a FCI calculation

Coupled-cluster theory

The coupled cluster (CC) family of methods is widely regarded as one of the most successful wave function approaches for describing chemical systems.^{7,8} In particular, low-order truncated CC methods, such as CC with singles, doubles and perturbative triples CCSD(T), properly describe weak correlation, while inclusion of higher-order excitations is required for strongly correlated systems. Therefore, these methods offer a balanced treatment of weak and strong electronic correlation by employing different levels of truncation.

In CC theory, the exponential excitation operator

$$e^{\hat{T}} = \hat{1} + \hat{T} + \frac{\hat{T}^2}{2!} + \frac{\hat{T}^3}{3!} + \dots \quad (4)$$

with

$$\hat{T} = \sum_{n=1}^N \hat{T}_n \quad (5)$$

(where N is the number of electrons) acts on a (normalized) single Slater determinant $|\Psi_0\rangle$ [such as Hartree-Fock (HF)] to generate the exact wave function

$$|\Psi\rangle = e^{\hat{T}} |\Psi_0\rangle \quad (6)$$

The n th excitation operator \hat{T}_n is defined, in second-quantized form, as

$$\hat{T}_n = \frac{1}{(n!)^2} \sum_{ij\dots ab\dots} t_{ij\dots}^{ab\dots} \hat{a}_i^\dagger \hat{a}_b^\dagger \dots \hat{a}_j \hat{a}_i \quad (7)$$

where \hat{a}_i and \hat{a}_a^\dagger are the usual annihilation and creation operators which annihilates an electron in the occupied spinorbital i and creates an electron in the vacant spinorbital a , respectively.

The Schrödinger equation can be rewritten in the CC framework as

$$\hat{H} e^{\hat{T}} |\Psi\rangle = E e^{\hat{T}} |\Psi\rangle \quad (8)$$

which can be rewritten as

$$\bar{H} |\Psi\rangle = E |\Psi\rangle \quad (9)$$

by defining the effective (non-Hermitian) similarity-transformed Hamiltonian

$$\bar{H} = e^{-\hat{T}} \hat{H} e^{\hat{T}} \quad (10)$$

Despite being non-Hermitian, the similarity transformation described in Eq. (10) guarantees that \bar{H} possesses the same energy spectrum as the original Hermitian operator \hat{H} . Additionally, the exponential structure of the wave operator ensures rigorous size-extensivity. This property contributes to the notable accuracy of the theory while maintaining a relatively low computational cost. The cluster amplitudes $t_{ij\dots}^{ab\dots}$ defined in Eq. (7) are the key quantities that need to be determined.

Truncating \hat{T} to double excitations, i.e., $\hat{T} = \hat{T}_1 + \hat{T}_2$ with

$$\hat{T}_1 = \sum_{ia} t_i^a \hat{a}_a^\dagger \hat{a}_i \quad (11a)$$

$$\hat{T}_2 = \frac{1}{4} \sum_{ijab} t_{ij}^{ab} \hat{a}_a^\dagger \hat{a}_b^\dagger \hat{a}_j \hat{a}_i \quad (11b)$$

defines CC with singles and doubles (CCSD) and one gets the single and double amplitudes, t_i^a and t_{ij}^{ab} , via the amplitude equations

$$\langle \Psi_i^a | \bar{H} | \Psi \rangle = 0 \quad (12a)$$

$$\langle \Psi_{ij}^{ab} | \bar{H} | \Psi \rangle = 0 \quad (12b)$$

The CCSD energy, which is non-variational, is obtained through projection:

$$E_{CC} = \langle \Psi | \bar{H} | \Psi \rangle = \frac{\langle \Psi | e^{-\hat{T}} \hat{H} e^{\hat{T}} | \Psi \rangle}{\langle \Psi | e^{-\hat{T}} e^{\hat{T}} | \Psi \rangle} \quad (13)$$

In contrast, its variational counterpart, denoted as VCC, is given by:

$$E_{VCC} = \frac{\langle \Psi | e^{\hat{T}^\dagger} \hat{H} e^{\hat{T}} | \Psi \rangle}{\langle \Psi | e^{\hat{T}^\dagger} e^{\hat{T}} | \Psi \rangle} \geq E_{FCI} \quad (14)$$

where the Rayleigh-Ritz variational principle is used to determine the energy and amplitudes. The VCC energy provides an upper bound to the exact FCI energy E_{FCI} . Unfortunately, VCC is computationally intractable. Even for truncated CC methods like CCSD, VCC exhibits factorial complexity because the power series expansion of the VCC energy (14) does not naturally terminate before reaching the N -electron limit.

Various CC schemes are available in QCMATH. For ground-state calculations, CC with doubles (CCD),⁹ CC with singles and doubles (CCSD),¹⁰ the distinguishable-cluster doubles (DCD),¹¹ direct-ring CCD (drCCD),[?] ring CCD (rCCD),[?] ladder CCD (lCCD),¹² crossed-ring CCD (crCCD),¹² and pair CCD (pCCD).¹³ For excited-state calculations, the EOM-CCSD method¹⁴ for excited-state calculations is available. To run these calculations, method keywords need to be specified:

- **"CCD"**: run a CCD calculation
- **"DCD"**: run a DCD calculation
- **"drCCD"**: run a drCCD calculation
- **"rCCD"**: run a rCCD calculation
- **"lCCD"**: run a lCCD calculation
- **"crCCD"**: run a crCCD calculation
- **"pCCD"**: run a pCCD calculation
- **"CCSD"**: run a CCSD calculation
- **"EOMCCSD"**: run an EOM-CCSD calculation

Moreover, two options are also available:

- **"max_SCF_CC"**: maximum number of iterations for CC calculations
- **"thresh_CC"**: convergence threshold of the maximum absolute value between the residuals

B. Charged excitations

Methods based on the one-body Green's function offer a means to describe charged excitations, namely, the ionization potentials (IPs) and electron affinities (EAs) of a system.^{4,5,15,16} This particular aspect forms the heart of QCMATH, with a diverse range of methods, approximations, and options available. To ensure clarity and coherence, this section is organized as follows: Firstly, we provide a brief introduction to the general equations that depend on the degree of (partial) self-consistency. These general equations are shared among the three self-energy approximations implemented in QCMATH: the second-order Green's function (GF2), the GW approximation, and the T -matrix approximation. By outlining these common equations, we establish a foundational understanding of the framework. Subsequently, we present the specific expressions corresponding to each of the self-energy approximations. This breakdown allows for a comprehensive exploration of the distinct methodologies incorporated in QCMATH, enabling users to leverage the most suitable approach for their research goals.

Three levels of (partial) self-consistency are available in QCMATH:

- the one-shot scheme where quasiparticles and satellites are obtained by solving, for each orbital p , the frequency-dependent quasiparticle equation

$$\omega = \epsilon_p^{\text{HF}} + \Sigma_{pp}^c(\omega) \quad (15)$$

where the diagonal approximation is used. Because we are, most of the time, interested in the quasiparticle solution we can use the linearized quasiparticle equation

$$\epsilon_p^{\text{QP}} = \epsilon_p^{\text{HF}} + Z_p \Sigma_{pp}^c(\epsilon_p^{\text{HF}}) \quad (16)$$

where the renormalization factor Z_p is defined as

$$Z_p = \left[1 - \frac{\partial \Sigma_{pp}(\omega)}{\partial \omega} \Big|_{\omega=\epsilon_p^{\text{HF}}} \right]^{-1} \quad (17)$$

- the eigenvalue scheme where we iterate on the quasiparticle solutions of Eq (16) that are used to build the self-energy Σ_{pp}^c (and Z_p)
- the quasiparticle scheme where an effective Fock matrix built from a frequency-independent Hermitian self-energy as¹⁷

$$\tilde{F}_{pq} = F_{pq} + \tilde{\Sigma}_{pq} \quad (18)$$

where

$$\tilde{\Sigma}_{pq} = \frac{1}{2} [\Sigma_{pq}^c(\epsilon_p^{\text{HF}}) + \Sigma_{qp}^c(\epsilon_p^{\text{HF}})] \quad (19)$$

Note that the whole self-energy is computed for this last scheme.

The non-linear Eq (15) can be exactly transformed in a linear eigenvalue problem by use of the unfolding process¹⁸⁻²⁰. For each orbital p , this yields a linear eigenvalue problem of the form

$$\mathbf{H}_p \cdot \mathbf{c}_\nu = \epsilon_\nu^{\text{QP}} \mathbf{c}_\nu \quad (20)$$

where ν runs overall solutions, quasiparticles, and satellites and with²¹

$$\mathbf{H}_p = \begin{pmatrix} \epsilon_p^{\text{HF}} & \mathbf{V}_p^{2\text{h1p}} & \mathbf{V}_p^{2\text{p1h}} \\ (\mathbf{V}_p^{2\text{h1p}})^\top & \mathbf{C}_p^{2\text{h1p}} & \mathbf{0} \\ (\mathbf{V}_p^{2\text{p1h}})^\top & \mathbf{0} & \mathbf{C}_p^{2\text{p1h}} \end{pmatrix} \quad (21)$$

Note that the different blocks will depend on the approximated self-energy. Now that the general equations have been set, we can turn to the self-energy approximations. Three different approximations are available in QCMATH: the second-order Green's function (GF2), the GW approximation, and the T -matrix approximation. For each approximation the three partially self-consistent schemes and the unfolding process are available. Note also that, regularization parameters are available in QCMATH.

Second-order Green's function approximation

The GF2 correlation self-energy is closely related to MP2 and is given by the following expression

$$\begin{aligned} \Sigma_{pq}^{\text{GF2}}(\omega) &= \frac{1}{2} \sum_{ija} \frac{\langle pa||ij \rangle \langle qa||ij \rangle}{\omega + \epsilon_a^{\text{HF}} - \epsilon_i^{\text{HF}} - \epsilon_j^{\text{HF}}} \\ &+ \frac{1}{2} \sum_{iab} \frac{\langle pi||ab \rangle \langle qi||ab \rangle}{\omega + \epsilon_i^{\text{HF}} - \epsilon_a^{\text{HF}} - \epsilon_b^{\text{HF}}} \end{aligned} \quad (22)$$

Keywords need to be specified for the different schemes:

- **"GOF2"** : run a one-shot calculation
- **"evGF2"** : run an eigenvalue calculation
- **"qsGF2"** : run a quasiparticle calculation
- **"upfGOF2"** : run an unfolded calculation

Example of a one-shot calculation

```
1 qcmath["H2", "6-31g", {"GOF2"}]
```

Note that here, an RHF calculation is done by default.

GW approximation

The GW correlation self-energy is given by

$$\Sigma_{pq}^{\text{GW}}(\omega) = \sum_{im} \frac{M_{pi,m}^{\text{ph}} M_{qi,m}^{\text{ph}}}{\omega - \epsilon_i^{\text{HF}} + \Omega_m^{\text{ph}}} + \sum_{am} \frac{M_{pa,m}^{\text{ph}} M_{qa,m}^{\text{ph}}}{\omega - \epsilon_a^{\text{HF}} - \Omega_m^{\text{ph}}} \quad (23)$$

where the screened two-electron integrals are given by

$$M_{pq,m}^{\text{ph}} = \sum_{ia} \langle pi|qa \rangle (\mathbf{X}^{\text{ph}} + \mathbf{Y}^{\text{ph}})_{ia,m} \quad (24)$$

with \mathbf{X}^{ph} and \mathbf{Y}^{ph} are the eigenvectors and excitations energies Ω_m^{ph} are the eigenvalues of the ph-dRPA problem that is discussed in Section IV C. Keywords for the method argument need to be specified for the different schemes:

- **"GOW0"** : run a one-shot calculation
- **"evGW"** : run an eigenvalue calculation
- **"qsGW"** : run a quasiparticle calculation
- **"upfGOW0"** : run an unfolded calculation

Example of an eigenvalue calculation

```
1 qcmath["H2", "6-31g", {"evGW"}]
```

Note that here, an RHF calculation is done by default.

T-matrix approximation

The T-matrix correlation self-energy is given by

$$\Sigma_{pq}^{\text{GT}}(\omega) = \sum_{in} \frac{M_{pi,n}^{\text{pp}} M_{qi,n}^{\text{pp}}}{\omega + \epsilon_i^{\text{HF}} - \Omega_n^{\text{pp}}} + \sum_{an} \frac{M_{pa,n}^{\text{hh}} M_{qa,n}^{\text{hh}}}{\omega + \epsilon_a^{\text{HF}} - \Omega_n^{\text{hh}}} \quad (25)$$

where the pp and hh versions of the screened two-electron integrals read

$$M_{pq,n}^{\text{pp}} = \sum_{c<d} \langle pq||cd \rangle X_{cd,n}^{\text{pp}} + \sum_{k<l} \langle pq||kl \rangle Y_{kl,n}^{\text{pp}} \quad (26a)$$

$$M_{pq,n}^{\text{hh}} = \sum_{c<d} \langle pq||cd \rangle X_{cd,n}^{\text{hh}} + \sum_{k<l} \langle pq||kl \rangle Y_{kl,n}^{\text{hh}} \quad (26b)$$

The components $X_{cd,n}^{\text{pp/hh}}$ and $Y_{kl,n}^{\text{pp/hh}}$ and excitation energies $\Omega_n^{\text{pp/hh}}$ are the double addition/removal eigenvector components and eigenvalues, respectively, of the pp-RPA eigenvalue problem discussed in Section IV C. Keywords for the method argument need to be specified for the different schemes:

- **"GOTO"** : run a one-shot calculation
- **"evGT"** : run an eigenvalue calculation
- **"qsGT"** : run a quasiparticle calculation
- **"upfGOTO"** : run an unfolded calculation

Example of a quasiparticle calculation

```
1 qcmath["H2", "6-31g", {"qsGT"}]
```

Note that here, an RHF calculation is done by default.

C. Neutral excitations

Within QCMATH, the computation of excitation energies utilizes methods formulated as a Casida-like equation.²² This equation is an eigenvalue equation that serves as a fundamental component in linear response theory. It plays a pivotal role in various approaches, including time-dependent density functional theory (TD-DFT),²³ the random phase approximation (RPA), and the Bethe-Salpeter equation (BSE).⁴ In this section, we begin by exploring the RPA method and distinguishing between different variations within this framework. By examining these different flavors of the RPA method, we gain insights into their unique characteristics and applicability. Subsequently, we delve into the discussion of the BSE method. This method represents another important approach for computing excitation energies, with its distinct theoretical foundations and computational considerations. By exploring the BSE method, users can gain a comprehensive understanding of its principles and its role within QCMATH. Note that when the spatial orbital implementation of a method is available, then we can use the "singlet" and/or "triplet" keywords to compute only singlet and/or triplet states.

Particle-hole random-phase approximation

The traditional RPA can be found under different names like RPax or ph-RPA.²⁴ We choose to call it ph-RPA to make the difference with the particle-particle RPA (pp-RPA). The ph-RPA problem takes the form of the following Casida-like equation

$$\begin{pmatrix} \mathbf{A}^{\text{ph}} & \mathbf{B}^{\text{ph}} \\ -\mathbf{B}^{\text{ph}} & -\mathbf{A}^{\text{ph}} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{X}^{\text{ph}} & \mathbf{Y}^{\text{ph}} \\ \mathbf{Y}^{\text{ph}} & \mathbf{X}^{\text{ph}} \end{pmatrix} = \begin{pmatrix} \mathbf{X}^{\text{ph}} & \mathbf{Y}^{\text{ph}} \\ \mathbf{Y}^{\text{ph}} & \mathbf{X}^{\text{ph}} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{\Omega}^{\text{ph}} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{\Omega}^{\text{ph}} \end{pmatrix} \quad (27)$$

where Ω_m is the diagonal matrix of the excitation energies, \mathbf{X}^{ph} and \mathbf{Y}^{ph} matrices are the transition coefficients, and the matrix elements are defined as

$$A_{ia,jb}^{\text{ph}} = (\epsilon_a^{\text{HF}} - \epsilon_i^{\text{HF}})\delta_{ij}\delta_{ab} + \langle ib||aj \rangle \quad (28)$$

$$B_{ia,jb}^{\text{ph}} = \langle ij||ab \rangle \quad (29)$$

Now, from these equations, different approximations arise:

- if we only take the direct term for the antisymmetrized two-electron integrals we end up with the direct ph-RPA (ph-dRPA), this is the one used in the *GW* approximation
- if we use the Tamm–Dancoff approximation (TDA) that sets $\mathbf{B}^{\text{ph}} = \mathbf{0}$, we end up with the ph-TDA approach

Note that TDA can be used with the ph-RPA flavor and gives ph-dTDA. Ground state correlation energy can be computed with

$$E_c^{\text{ph-RPA}} = \frac{1}{2} \left(\sum_m \Omega_m^{\text{ph}} - \text{Tr}(\mathbf{A}^{\text{ph}}) \right) \quad (30)$$

Keywords for the method argument need to be specified for the different approaches and options:

- "RPax" : run a ph-RPA calculation
- "RPA" : run a ph-dRPA calculation

The option "TDA" can be set to `True`, by default `"TDA"=False`.

Particle-particle random-phase approximation

The particle-particle RPA (pp-RPA) problem considers the excitation energies of the $(N+2)$ - and $(N-2)$ -electron systems.²⁴ It is also defined by a slightly different eigenvalue problem than ph-RPA:

$$\begin{pmatrix} \mathbf{C}^{\text{pp}} & \mathbf{B}^{\text{pp/hh}} \\ -(\mathbf{B}^{\text{pp/hh}})^\dagger & -\mathbf{D}^{\text{hh}} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{X}^{\text{pp}} & \mathbf{Y}^{\text{hh}} \\ \mathbf{Y}^{\text{pp}} & \mathbf{X}^{\text{hh}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Omega}^{\text{pp}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega}^{\text{hh}} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{X}^{\text{pp}} & \mathbf{Y}^{\text{hh}} \\ \mathbf{Y}^{\text{pp}} & \mathbf{X}^{\text{hh}} \end{pmatrix} \quad (31)$$

where $\boldsymbol{\Omega}^{\text{pp/hh}}$ are the diagonal matrices of the double addition/removal excitation energies, labeled by n , and the matrix elements are defined as

$$C_{ab,cd}^{\text{pp}} = (\epsilon_a^{\text{HF}} + \epsilon_b^{\text{HF}})\delta_{ac}\delta_{bd} + \langle ab||cd \rangle \quad (32a)$$

$$B_{ab,ij}^{\text{pp/hh}} = \langle ab||ij \rangle \quad (32b)$$

$$D_{ij,kl}^{\text{hh}} = -(\epsilon_i^{\text{HF}} + \epsilon_j^{\text{HF}})\delta_{ik}\delta_{jl} + \langle ij||kl \rangle \quad (32c)$$

The $\mathbf{X}^{\text{pp/hh}}$ and $\mathbf{Y}^{\text{pp/hh}}$ are the double addition/removal transition coefficients matrices. In the same way we did for the ph-RPA, we can obtain the correlation energy at the pp-RPA level using^{12,25}

$$E_c^{\text{pp-RPA}} = \frac{1}{2} \left(\sum_n \Omega_n^{\text{pp}} - \sum_n \Omega_n^{\text{hh}} - \text{Tr}\mathbf{C}^{\text{pp}} - \text{Tr}\mathbf{D}^{\text{hh}} \right) \quad (33)$$

The keyword to use the pp-RPA is `pp-RPA`. Note that TDA is also available with the option `"TDA"=True`.

Bethe-Salpeter equation

The Bethe-Salpeter equation (BSE) is related to the two-body Green's function.²⁶ The central quantity is the so-called BSE kernel defined as the functional derivative

of the self-energy with respect to G . As exposed in Section IV B, there are several approximations of the self-energy and each one leads to a different BSE approximation. The common central equation is the following eigenvalue equation

$$\begin{pmatrix} \mathbf{A}^{\text{BSE}} & \mathbf{B}^{\text{BSE}} \\ -\mathbf{B}^{\text{BSE}} & -\mathbf{A}^{\text{BSE}} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{X}_m^{\text{BSE}} \\ \mathbf{Y}_m^{\text{BSE}} \end{pmatrix} = \Omega_m^{\text{BSE}} \begin{pmatrix} \mathbf{X}_m^{\text{BSE}} \\ \mathbf{Y}_m^{\text{BSE}} \end{pmatrix} \quad (34)$$

where the BSE matrix elements depend on the choice of the BSE kernel. To run a BSE calculation we have first to specify the approximation for the self-energy with the method argument and the keyword for this option is `"BSE=True"`. Note that in general a BSE calculation is done in the static approximation, which is the equivalent of the adiabatic approximation in TD-DFT. It is possible to take into account dynamical effects using first-order perturbation theory²⁷ using the option `"dBSE=True"`. This dynamical correction is applicable for all the different BSE kernels available in qcmath. Note that this dynamical correction is only available in TDA with the option `"dTDA"`.

V. PROGRAMMER GUIDE

As mentioned in the first section, one of the primary objectives of QCMATH is to enable newcomers in quantum chemistry to explore and advance their ideas through coding. Therefore, it is crucial to allow them to incorporate their methods into QCMATH. To facilitate this process, we have developed a notebook example called `module_example.nb` to guide users step-by-step. The following outlines the different stages involved in adding a new method to QCMATH:

1. The new method needs to be implemented in its notebook.
2. add your method in the `src/utils/list_method.nb` and specify the dependencies. For example, if a new post-HF method is proposed, then one has to add `"RHF"` or `"UHF"` as a dependency.
3. add default options in `src/Main/default_options.nb` if needed.
4. add a call to your method in `src/Main/Main.nb` as

```

1 NameNewMethod="NameNewMethod"
2 If[ToDoModules[NameNewMethod]["Do"] == True,
3   NotebookEvaluate[path<>"/src/"<>
   NameNewMethod<>".nb"];

1 PrintTemporary[Style[NameNewMethod<>
  calculation...", Bold, Orange]];
2 {time, outputsNewMethod} = Timing[NewMethod[
  arguments, options]];

```

```

1 If[verbose == True,
2   Print["CPU time for "<>NameNewMethod<>"
  calculation=", time]];
3 ];

```

For each new method notebook, it is essential to organize the code into potentially three modules. The first module is responsible for reading the input and options, followed by invoking either the spin or spatial orbitals module, and ultimately returning the corresponding output. The remaining two modules are dedicated to implementing the new method in spin and spatial orbitals, respectively. It is important to note that if your method is exclusively implemented in spatial orbitals, your notebook will consist of only two parts. Further details regarding this structure can be found in the `module_example.nb` notebook.

VI. CONCLUSION AND PERSPECTIVES

Here, we have presented the current capabilities of QCMATH, a set of MATHEMATICA modules to perform electronic structure calculations. We hope to implement new methods in the near future, especially at the equation-of-motion CC level where our group has recently designed an automatic equation generator based on MATHEMATICA, named `EOMCCGEN`.²⁸ Moreover, various schemes based on many-body Green's functions are being currently implemented.²⁹⁻³¹

ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 863481).

- ¹A. Szabo and N. S. Ostlund, *Modern quantum chemistry* (McGraw-Hill, New York, 1989).
- ²F. Jensen, *Introduction to Computational Chemistry*, 3rd ed. (Wiley, New York, 2017).
- ³T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic-Structure Theory* (John Wiley & Sons, Inc., 2013).
- ⁴R. M. Martin, L. Reining, and D. M. Ceperley, *Interacting Electrons: Theory and Computational Approaches* (Cambridge University Press, 2016).
- ⁵A. L. Fetter and J. D. Waleck, *Quantum Theory of Many Particle Systems* (McGraw Hill, San Francisco, 1971).
- ⁶A method is said to be size-consistent if the correlation energy of two non-interaction systems is equal to twice the correlation energy of the isolated system. Size-extensivity means that the correlation energy grows linearly with the system size.
- ⁷I. Shavitt and R. J. Bartlett, *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*, Cambridge Molecular Science (Cambridge University Press, Cambridge, 2009).
- ⁸T. D. Crawford and H. F. Schaefer, in *Reviews in Computational Chemistry* (John Wiley & Sons, Ltd, 2000) pp. 33-136.
- ⁹J. A. Pople, R. Krishnan, H. B. Schlegel, and J. S. Binkley, *Int. J. Quantum Chem.* **14**, 545 (1978).
- ¹⁰J. F. Stanton, J. Gauss, J. D. Watts, and R. J. Bartlett, *J. Chem. Phys.* **94**, 4334 (1991).
- ¹¹D. Kats and F. R. Manby, *J. Chem. Phys.* **139**, 021102 (2013).

- ¹²G. E. Scuseria, T. M. Henderson, and I. W. Bulik, *J. Chem. Phys.* **139**, 104113 (2013).
- ¹³T. M. Henderson, I. W. Bulik, T. Stein, and G. E. Scuseria, *J. Chem. Phys.* **141**, 244104 (2014).
- ¹⁴J. F. Stanton and R. J. Bartlett, *J. Chem. Phys.* **98**, 7029 (1993).
- ¹⁵G. Csanak, H. Taylor, and R. Yaris, in *Advances in atomic and molecular physics*, Vol. 7 (Elsevier, 1971) pp. 287–361.
- ¹⁶R. D. Mattuck, *A guide to Feynman diagrams in the many-body problem*, 2nd ed., Dover books on physics and chemistry (Dover Publications, New York, 1992).
- ¹⁷E. Monino and P.-F. Loos, *J. Chem. Theory Comput.* **17**, 2852 (2021).
- ¹⁸O. J. Backhouse, M. Nusspickel, and G. H. Booth, *J. Chem. Theory Comput.* **16**, 1090 (2020).
- ¹⁹S. J. Bintrim and T. C. Berkelbach, *J. Chem. Phys.* **154**, 041101 (2021).
- ²⁰E. Monino and P.-F. Loos, *J. Chem. Phys.* **156**, 231101 (2022).
- ²¹J. Tölle and G. Kin-Lic Chan, *J. Chem. Phys.* **158**, 124123 (2023).
- ²²M. E. Casida, *J. Chem. Phys.* **122**, 054111 (2005).
- ²³C. Ullrich, *Time-Dependent Density-Functional Theory: Concepts and Applications*, Oxford Graduate Texts (Oxford University Press, New York, 2012).
- ²⁴P. Ring and P. Schuck, *The Nuclear Many-Body Problem* (Springer, 2004).
- ²⁵D. Peng, S. N. Steinmann, H. van Aggelen, and W. Yang, *J. Chem. Phys.* **139**, 104112 (2013).
- ²⁶G. Strinati, *Riv. Nuovo Cimento* **11**, 1 (1988).
- ²⁷P.-F. Loos and X. Blase, *J. Chem. Phys.* **153**, 114120 (2020).
- ²⁸R. Quintero-Monsebaiz and P.-F. Loos, “Equation generator for equation-of-motion coupled cluster assisted by computer algebra system,” (2023), [arXiv:2306.01573](https://arxiv.org/abs/2306.01573) [physics.chem-ph].
- ²⁹R. Quintero-Monsebaiz, E. Monino, A. Marie, and P.-F. Loos, *J. Chem. Phys.* **157**, 231102 (2022).
- ³⁰E. Monino and P.-F. Loos, *J. Chem. Phys.* **159**, 034105 (2023).
- ³¹A. Marie and P.-F. Loos, *J. Chem. Theory Comput.* **19**, 3943 (2023).