CIS and TDHF - Homework

Albert Makhmudov

19/03/2025

Data Availability

The source code of this project as well as the example input files are available at the corresponding Github page. There one could also find the example output files and the detailed instructions on how to run and compile the code. The example output files are the outputs of the CIS and TDHF calculations. The source code of this project is written in `Fortran90`.

Code Overview

Input

In order to run the code, one should provide the atom or molecule name and the respective basis set. Currently, Be, He, Ne, and H_2O are supported as well as the `cc-pvdz` and `cc-pvtz` basis sets. The example on how to execute the program can be found in `README.md` file.

AO to MO transformation

The AO to MO transformation is performed using the following equation:

$$(pq|rs) = \sum_{\mu\nu\lambda\sigma} c_{\mu p} c_{\nu q} c_{\lambda r} c_{\sigma s} (\mu\nu|\lambda\sigma) \tag{1}$$

where $c_{\mu p}$ and the others are the coefficients of the MOs in the AO basis and $(\mu\nu|\lambda\sigma)$ are the integrals in the AO basis. The transformation is performed in the `AO_to_MO` subroutine. Below one could find a code snippet. Only part of the code is visualised.

```fortran
! Initialisation
tmp1 = 0.0d0
tmp2 = 0.0d0

! First quarter-transformation
do p = 1, nBas
    do mu = 1, nBas
    tmp1(p,:,:,:) = tmp1(p,:,:,:) + c(mu,p)*ERI_AO(mu,:,:,:)
    end do
end do

! Second quarter-transformation
do p = 1, nBas
    do q = 1, nBas
    do nu = 1, nBas
```

```
        tmp2(p,q,:,:) = tmp2(p,q,:,:) + c(nu,q)*tmp1(p,nu,:,:)
    end do
    end do
end do

tmp1 = 0.0d0

! Third quarter-transformation
do p = 1, nBas
    do q = 1, nBas
    do r = 1, nBas
        do la = 1, nBas
        tmp1(p,q,r,:) = tmp1(p,q,r,:) + c(la,r)*tmp2(p,q,la,:)
        end do
    end do
    end do
end do

tmp2 = 0.0d0

! Fourth quarter-transformation
do p = 1, nBas
    do q = 1, nBas
    do r = 1, nBas
        do s = 1, nBas
        do si = 1, nBas
            tmp2(p,q,r,s) = tmp2(p,q,r,s) + c(si,s)*tmp1(p,q,r,si)
        end do
        end do
    end do
    end do
end do

ERI_MO = tmp2
```

Listing 1: Subroutine for the `AO_to_MO` transformation. Only part of the full code is mentioned.

The transformation is done in an efficient way as discussed in the [1] to overcome the $N(O^8)$ scaling. The transformation is done in four steps. This way the scaling is reduced to $N(O^5)$.

Configuration Interaction with Singles (CIS)

The CIS matrix is constructed using the following equation:

$$\mathbf{A} \cdot \mathbf{X}_m = \omega_m \mathbf{X}_m \tag{2}$$

where the elements of the matrix $\mathbf{A}$ are given by

$$A_{ia,jb} = (\epsilon_a - \epsilon_i)\delta_{ij}\delta_{ab} + 2(ia|bj) - (ij|ba) \tag{3}$$

The excitation energies are obtained by diagonalising the matrix $\mathbf{A}$. The code snippet for the construction of the CIS matrix is provided below.

```
! Construct the CIS matrix
A_matrix = 0.0d0
do ia = 1, nStates
    i = (ia - 1) / nV + 1
    a = mod(ia - 1, nV) + nO + 1

    do jb = 1, nStates
```

```
        j = (jb - 1) / nV + 1
        b = mod(jb - 1, nV) + nO + 1

        ! Diagonal term
        if (i == j .and. a == b) then
            A_matrix(ia,jb) = e(a) - e(i)
        end if

        ! Exchange terms
        A_matrix(ia,jb) = A_matrix(ia,jb) + 2.0d0 * ERI_MO(i,b,a,j) &
        - ERI_MO(i,b,j,a)
    end do
end do
```

Listing 2: Subroutine for the CIS matrix construction. Only part of the full code is mentioned.

Time-Dependent Hartree-Fock (TDHF)

To tackle the TDHF problem, the following **A** and **B** matrices are constructed first:

$$A_{ia,jb} = (\epsilon_a - \epsilon_i)\delta_{ij}\delta_{ab} + 2(ia|bj) - (ij|ba) \tag{4}$$

$$B_{ia,jb} = 2(ia|bj) - (ib|ja) \tag{5}$$

The **A** matrix is the same as in the CIS case.

To obtain the TDHF excitation energies, one should construct and diagonalise the **C** matrix depicted in the following equation:

$$\mathbf{C} = (\mathbf{A} - \mathbf{B})^{1/2} \cdot (\mathbf{A} + \mathbf{B}) \cdot (\mathbf{A} - \mathbf{B})^{1/2} \tag{6}$$

The code snippet for the aforementioned procedures is provided below.

```
! Construct A and B matrices
do ia = 1, nStates
    i = (ia - 1) / nV + 1
    a = mod(ia - 1, nV) + nO + 1

    do jb = 1, nStates
        j = (jb - 1) / nV + 1
        b = mod(jb - 1, nV) + nO + 1

        ! A matrix
        ! Diagonal term
        if (i == j .and. a == b) then
            A_matrix(ia,jb) = e(a) - e(i)
        end if
        ! Exchange terms
        A_matrix(ia,jb) = A_matrix(ia,jb) + 2.0d0 * ERI_MO(i,b,a,j) &
        - ERI_MO(i,b,j,a)

        ! B matrix
        ! Exchange terms
        B_matrix(ia,jb) = B_matrix(ia,jb) + 2.0d0 * ERI_MO(i,j,a,b) &
        - ERI_MO(i,j,b,a)
    end do
end do

```

```
! Compute (A - B) and (A + B)
AminB = A_matrix - B_matrix
AplusB = A_matrix + B_matrix
AminB_temp = AminB

! Compute (A - B)^(-1/2) using matrix diagonalisation
call diagonalize_matrix(nStates, AminB_temp, omega)
do i = 1, nStates
    sqrt_AminB_temp(i,i) = sqrt(omega(i))  ! Square root of eigenvalue
end do
! (A-B)^(-1/2)
AminB = matmul(AminB_temp, matmul(sqrt_AminB_temp, transpose(AminB_temp)))

! Compute C matrix
C_matrix = matmul(AminB, matmul(AplusB, AminB))

! Diagonalize C to get excitation energies
! Reuse omega array
omega = 0.0d0
call diagonalize_matrix(nStates, C_matrix, omega)
omega = sqrt(omega)  ! Excitation energies
```

Listing 3: Subroutine to calculate the TDHF excitation energies. Only part of the full code is mentioned.

Results and Discussion

The reported experimental excitation energies for He, Ne, and Be atoms are 24.587, 21.564, and 9.322 eV, respectively [2]. The calculated excitation energies using the cc-pvdz and cc-pvtz basis sets with the CIS and TDHF methods are presented in Table 1.

The results for He and Ne are overestimated while for Be they're underestimated in comparison with the experimental values. The CIS method provides slightly better results compared to the TDHF method in case of the Be atom. On the other hand, the TDHF method shows slightly better results in case of the He atom. The same applies to Ne. This could be due to the lack of correlation effects in the calculations. The correlation effects could be included by using more sophisticated methods such as the flavours of coupled cluster theory.

Taking into account the obtained results, one could conclude that both methods could be used to get the solely quantitative picture of the excitation energies. However, the results should be taken with a pinch of salt as the correlation effects are not included in the calculations.

	CIS		**TDHF**	
	cc-pvdz	cc-pvtz	cc-pvdz	cc-pvtz
He	51.947	31.807	51.577	31.608
Ne	49.010	33.210	48.871	33.174
Be	5.295	5.143	4.993	4.867

Table 1: Excitation energies (in eV) for He, Ne, and Be atoms using the cc-pvdz and cc-pvtz basis sets obtained with the implemented CIS and TDHF methods.

Generative AI Usage

In this work, ChatGPT large language model was used to check grammar and spelling of the main body of text, whilst Perplexity was utilised for the sake of literature search.

Acknowledgments

This project is based on data and instructions provided by Pina Romaniello available at the aforementioned Github page.

References

[1] Joshua Goings. Efficient Two-Electron Integral Transformations in Python, or, Adventures in Scaling, 19-03-2025. URL https://joshuagoings.com/2013/05/14/efficient-two-electron-integral-transformations-in-python-or-adventures-in-scaling/.

[2] A. Kramida, Yu. Ralchenko, J. Reader, and and NIST ASD Team. NIST Atomic Spectra Database (ver. 5.12), [Online], 2025, March 19. URL https://physics.nist.gov/asd.