

DQΦpac Version 2.0

Chad E. Hoyer, Laura Gagliardi, and Donald G. Truhlar

University of Minnesota

User's Manual

Date of Last Modification: July 23, 2017

Contents

1	Introduction	1
2	Recommended Citations	3
3	Version History	4
3.1	Version 2.0	4
4	Installation Details	5
4.1	Version 1.0 and 2.0	5
5	Basic Instructions	6
6	Electronic Structure Packages	7
6.1	Q-Chem	7
6.2	Molpro	7
6.3	Gen	8
7	GEN Keywords	9
7.1	AENE	9
7.2	EXDX	9
7.3	TRDX	9
7.4	EXDY	10
7.5	TRDY	10
7.6	EXDZ	10
7.7	TRDZ	11
7.8	EXXX	11
7.9	TRXX	11
7.10	EXYY	12
7.11	TRYX	12

7.12	EXZZ	12
7.13	TRZZ	13
7.14	EXEP	13
7.15	TREP	13
7.16	Comment on EXEP and TREP keywords	14
8	Keywords	15
8.1	SOFT	15
8.2	NSTATE	15
8.3	METH	15
8.4	STATES	16
8.5	GAMMA	16
8.6	NQOR	17
8.7	ALPHA	17
8.8	QORI	17
8.9	NEOR	18
8.10	BETA	18
8.11	CHARGE	19
8.12	COORD	19
8.13	QNUC	19
8.14	NUCL	20
8.15	ETHRESH	20
8.16	WALPH	20
8.17	ZSIG	21
8.18	FPRINT	21
9	Sample Input and Output	24

1 Introduction

The DQΦpac program is a program for computing diabatic states and the diabatic potential energy matrix. Specifically, the program uses the N -electron dipole, quadrupole, and/or electrostatic potential to compute an adiabatic-to-diabatic transformation matrix by way of the dipole-quadrupole-electrostatic-potential (DQΦ) method. [1, 2]

The function that is maximized to produce diabatic states is

$$\begin{aligned}
 f_{\text{DQ}\Phi} = & \sum_A \left(\gamma \left| \left\langle \phi_A \left| \sum_p^{N_{\text{el}}+N_{\text{nuc}}} q_p |\mathbf{R}_p - \mathbf{R}_0| \right| \phi_A \right\rangle \right|^2 \right) \\
 & + \sum_A \left(\sum_j^{N_Q} \alpha_j \left| \left\langle \phi_A \left| \sum_p^{N_{\text{el}}+N_{\text{nuc}}} q_p |\mathbf{R}_p - \mathbf{R}_j|^2 \right| \phi_A \right\rangle \right|^2 \right) \\
 & + \sum_A \left(\sum_k^{N_\Phi} \beta_k \left| \left\langle \phi_A \left| \sum_p^{N_{\text{el}}} q_p |\mathbf{R}_p - \mathbf{R}_k|^{-1} \right| \phi_A \right\rangle \right|^2 \right)
 \end{aligned} \tag{1}$$

where ϕ_A is a diabatic state, γ is a parameter that weights the dipole term, p is an index over particles (a particle is an electron or nucleus), N_{el} is the number of electrons, N_{nuc} is the number of nuclei, q_p is the charge of particle p , α_j is a parameter that weights the quadrupole term for origin j , j is a choice of origin for the quadrupole integrals, N_Q is the number of origins used for the quadrupole integrals, β_k is a parameter that weights the electrostatic potential term for origin k , k is a choice of origin for the electrostatic potential integrals, and N_Φ is the number of origins used for the electrostatic potential integrals. Note that when α_j and β_j are set to 0 the DQΦ method reduces to Boys localized diabatization. [3]

In addition we developed a weighted version of DQΦ diabatization that is analogous to the weighted versions for the diabatization of Werner and Meyer [4] and Boys [5] and uses the same weighting function as in Ref. [5]. In weighted DQΦ the off-diagonal elements of each adiabatic property are multiplied by a weighting function

$$W_{IJ} \langle \Psi_I | \hat{O} | \Psi_J \rangle \tag{2}$$

where W is a weighting function, \hat{O} is a property operator, and Ψ is an adiabatic wavefunction. Then, $f_{\text{DQ}\Phi}$ is maximized as normal using the weighted adiabatic properties.

One choice of weighting function for weighted DQ Φ is the one used in Ref. [5]

$$W_{IJ} = \frac{1}{4} \operatorname{erfc} \left(\frac{E_I - E_{\text{thresh}}}{\alpha_{\text{Weight}}} \right) \operatorname{erfc} \left(\frac{E_J - E_{\text{thresh}}}{\alpha_{\text{Weight}}} \right) \quad (3)$$

where E_I is the energy of adiabatic state I , E_{thresh} is an energy threshold parameter, and α_{Weight} is a parameter (a recommended value is 0.005 a.u.). See the the WALPH and ETHRESH keywords in Section 8.

Another choice of weighting function for weighted DQ Φ was developed by Zoltan Varga. The function is only dependent upon the splitting of the adiabatic energies.

$$W_{IJ} = \left(\operatorname{erfc} \left(\frac{|E_I - E_J|}{\sigma_Z} \right) \right)^{\frac{1}{2}} \quad (4)$$

where E_I is the energy of adiabatic state I , E_J is the energy of adiabatic state J , and σ_Z is a parameter that affects the difference in energy at which the weights will become smaller.

Weighting is useful for cases where the highest-energy diabatic state is unimportant in some regions (e.g., very high in energy). The inclusions of such a state in the diabaticization may deteriorate the quality of the other diabatic states in a region of unimportance. In such a case one can set a smaller weight on the state, such that the properties of that diabatic state are not as important in the maximization of the DQ Φ function.

2 Recommended Citations

- Citation for DQΦpac software: Reference [6]
- Citation for DQ method: References [1, 2]
- Citation for DQΦ method: Reference [2]

3 Version History

3.1 Version 2.0

The following additions were added in DQΦpac, version 2.0.

- Weighted versions of the Boys localized diabatization, DQ, and DQΦ methods
- The weighting function used in Ref. [5]
- The weighting function described in Equation 4
- Optional printing of “.dat” files
- Option for the user to manually create a “qc.out”
- Printing of weighted adiabatic properties and of diabatic properties

4 Installation Details

This section lists installation details with which DQΦpac has been tested.

4.1 Version 1.0 and 2.0

The DQΦ code has been tested successfully on the following platforms:

Computer	OS	Compiler
Itasca Linux Cluster	CentOS Linux	GCC Compiler 4.4.7

5 Basic Instructions

To use $DQ\Phi$ pac, run the executable in a directory that also contains two other files “qc.dqv” and “qc.out” (the files must be named as such). The file “qc.dqv” is the $DQ\Phi$ pac input and contains keywords covered in Section 8. The file “qc.out” contains the output from the electronic structure calculation. It is recommended that the “qc.out” file only contain properties that will be used (e.g., if two origins for the electrostatic potential are needed, do not have three origins in the “qc.out” file).

6 Electronic Structure Packages

This section discusses the electronic structure softwares that are interfaced with DQ Φ pac. Please see the SOFT keyword in Section 8.

6.1 Q-Chem

The DQ Φ pac software works with REM module (it has been tested using random phase approximation (RPA), Kohn-Sham time-dependent density functional theory (TDDFT), configuration interaction with singles (CIS), and Kohn-Sham TDDFT with the Tamm-Dancoff approximation (TDA)) of Q-Chem 4.2. Do not include the output of more than one method per “qc.out” to ensure that the correct values are read. The quadrupole moment computed by Q-Chem includes only the electronic contribution, but in Equation 1 we see that the DQ Φ method also includes the nuclear contribution. In addition, Q-Chem calculates the electronic quadrupole moment with the incorrect sign. To add the nuclear contribution to the quadrupole integrals with DQ Φ pac, one must use the COORD, NUCL, and QNUC keywords covered in Section 8. The coordinates used with the NUCL keyword of DQ Φ pac should be the ones listed under “Standard Nuclear Orientation” in the Q-Chem output. Q-Chem translates the origin to the center of nuclear charge, which is why one needs to use the Standard Nuclear Orientation geometries. No action is required by the user to fix the sign of the quadrupole moment computed with Q-Chem; DQ Φ pac fixes the sign automatically.

6.2 Molpro

The DQ Φ pac software works only with wave functions that can be used in the TRANS module of Molpro Version 2012.1. The operator that is used in the TRANS module for the quadrupole is the ‘second moment’ in the Molpro manual. The TRANS module cannot be used to compute the electrostatic potential since it includes the nuclear contribution; however, in Equation 1 we see that the DQ Φ method does not include the nuclear contribution in the electrostatic potential. To compute electrostatic potential integrals in Molpro, one

can use the density and transition density from TRANS to then compute electrostatic potential integrals in MATROP. One must compute all transition elements of the electrostatic potential. To compute multiple origins for the electrostatic potential, one has to compute them in the same Molpro calculation, which includes calls to MATROP for each choice of origin.

6.3 Gen

The DQΦpac software is compatible with any electronic structure software that is capable of computing properties of interest. One simply needs to enter the data into “qc.out” using the keywords that are covered in Section 7.

7 GEN Keywords

This section covers keywords for use in “qc.out”. See the SOFT keyword in Section 8 and Section 6.3. This section is only needed when using the GEN option for the SOFT keyword. All examples below are for a 3-state case. If “END” is in the example, it is required for that keyword.

7.1 AENE

AENE This keyword specifies adiabatic energies.

Input Example

```
AENE
-2.71114
-2.12187
-1.98790
END
```

7.2 EXDX

EXDX This keyword specifies expectation values of the x component of the dipole.

Input Example

```
EXDX
0.23814
-4.12187
-3.28190
END
```

7.3 TRDX

TRDX This keyword specifies transition values of the x component of the dipole. The entries are in order of increasing value in going left to right along a row of the matrix as shown symbolically for the example case below: X_{12} , X_{13} , then X_{23} .

Input Example

```
TRDX
0.33214
-2.92187
-1.21190
END
```

7.4 EXDY

EXDY This keyword specifies expectation values of the y component of the dipole.

Input Example

```
EXDY
0.23814
-4.12187
-3.28190
END
```

7.5 TRDY

TRDY This keyword specifies transition values of the y component of the dipole. The entries are in order of increasing value in going left to right along a row of the matrix as shown symbolically for the example case below: Y_{12} , Y_{13} , then Y_{23} .

Input Example

```
TRDY
0.33214
-2.92187
-1.21190
END
```

7.6 EXDZ

EXDZ This keyword specifies expectation values of the z component of the dipole.

Input Example

```
EXDZ
0.73214
-2.52187
-40.28190
END
```

7.7 TRDZ

TRDZ This keyword specifies transition values of the z component of the dipole. The entries are in order of increasing value in going left to right along a row of the matrix as shown symbolically for the example case below: Z_{12} , Z_{13} , then Z_{23} .

Input Example

```
TRDZ
0.63224
-2.52287
-1.41290
END
```

7.8 EXXX

EXXX This keyword specifies expectation values of the x^2 component of the quadrupole.

Input Example

```
EXXX
2.73214
-5.52187
-80.28190
END
```

7.9 TRXX

TRXX This keyword specifies transition values of the x^2 component of the dipole. The entries are in order of increasing value in going left to right along a row of the matrix as shown symbolically for the example case below: X_{12}^2 , X_{13}^2 , then X_{23}^2 .

Input Example

```
TRXX
0.33224
-4.12287
-2.91290
END
```

7.10 EXYY

EXYY This keyword specifies expectation values of the y^2 component of the quadrupole.

Input Example

```
EXYY
2.73214
-5.52187
-80.28190
END
```

7.11 TRY Y

TRY Y This keyword specifies transition values of the y^2 component of the dipole. The entries are in order of increasing value in going left to right along a row of the matrix as shown symbolically for the example case below: Y_{12}^2 , Y_{13}^2 , then Y_{23}^2 .

Input Example

```
TRY Y
0.33224
-4.12287
-2.91290
END
```

7.12 EXZZ

EXZZ This keyword specifies expectation values of the z^2 component of the quadrupole.

Input Example

```
EXZZ
4.73214
-15.52187
-94.28190
END
```

7.13 TRZZ

TRZZ This keyword specifies transition values of the z^2 component of the dipole. The entries are in order of increasing value in going left to right along a row of the matrix as shown symbolically for the example case below: Z_{12}^2 , Z_{13}^2 , then Z_{23}^2 .

Input Example

```
TRZZ
1.93224
-7.82287
-3.41290
END
```

7.14 EXEP

EXEP This keyword specifies expectation values of the electrostatic potential for a given origin. To read in the electrostatic potential for more than one origin, see Section 7.16.

Input Example

```
EXEP
-1.73214
-2.52187
-6.28190
END
```

7.15 TREP

TREP This keyword specifies transition values of the electrostatic potential. The entries are in order of increasing value in going left to right along a row of the matrix as shown symbolically for the example case below: R_{12}^{-1} , R_{13}^{-1} , then R_{23}^{-1} . For more information on how to read in the electrostatic potential for more than one origin, see Section 7.16.

Input Example

```
TREP
0.43224
-2.42287
-1.71290
END
```


7.16 Comment on EXEP and TREP keywords

To read in the electrostatic potential for multiple origins, one needs multiple calls of the EXEP and TREP keywords. The ordering of the keywords matters, follow the ordering in the example below.

Input Example For 2 Origins

```
EXEP  
-1.73214  
-2.52187  
-6.28190  
END
```

```
TREP  
0.43224  
-2.42287  
-1.71290  
END
```

```
EXEP  
-1.69714  
-2.13187  
-5.73611  
END
```

```
TREP  
0.33224  
-1.94857  
-1.54893  
END
```

8 Keywords

Some input examples contain “END” and some do not. If “END” is in the example, it is required for that keyword.

8.1 SOFT

SOFT The keyword SOFT is *mandatory*. It specifies which electronic structure package is used.

Acceptable Values

1. MOLPRO - The “qc.out” file was generated using Molpro
2. QCHEM - The “qc.out” file was generated using Q-Chem
3. GEN - The “qc.out” file was generated by the user

Input Example

```
SOFT
QCHEM
```

8.2 NSTATE

NSTATE The keyword NSTATE is *mandatory*. It specifies how many electronic states are of interest. This defines the size of the $N \times N$ property matrices and output $N \times N$ diabatic potential energy matrix.

Input Example

```
NSTATE
4
```

8.3 METH

METH The keyword METH is *highly recommended*. It specifies which diabatisation method is used. The default is to use Boys localized diabatisation.

Acceptable Values

1. DIP - Uses Boys localized diabatisation to produce diabatic states (i.e., only the first term in Equation 1 in Section 1 is used).
2. DQUAD - Uses DQ diabatisation to produce diabatic states (i.e., only the first and second term in Equation 1 in Section 1 are used).
3. DQV - Uses $DQ\Phi$ diabatisation to produce diabatic states (i.e., all three terms in Equation 1 in Section 1 are used). This is also the keyword for the $D\Phi$ method.

Input Example

```
METH
DQV
```

8.4 STATES

STATES The keyword STATES is *optional*. It specifies which electronic states are of interest. This is important when one wishes exclude certain states in the diabatisation (e.g., the ground state). By default, $DQ\Phi_{\text{pac}}$ will assume that the user is interested in states 1 to NSTATE. The value for NSTATE should correspond to the number of entries for STATES (in the example below NSTATE would be 3).

Input Example

```
STATES
2
3
4
END
```

8.5 GAMMA

GAMMA The keyword GAMMA is *optional*. It specifies the value of the γ parameter in Equation 1 in Section 1. The value entered must be a real number. The default is 1.0.

Input Example

```
GAMMA  
2.0
```

8.6 NQOR

NQOR The keyword NQOR is *optional*. It specifies the number of origins used for the quadrupole integrals, which is the value of N_Q in Equation 1 in Section 1. The value entered must be an integer. The default is 1.

Input Example

```
NQOR  
3
```

8.7 ALPHA

ALPHA The keyword ALPHA is *optional*. It specifies the α parameter. As a reminder α weights the importance of the quadrupole integrals for each choice of origin, which is the α_j in Equation 1 in Section 1. The values entered must be real numbers. Each line corresponds to each choice of origin, j . If ALPHA is not specified, each value of α_j is assumed to be 0.0. The example below is for a case when NQOR is 3.

Input Example

```
ALPHA  
1.0  
0.5  
10.0  
END
```

8.8 QORI

QORI The keyword QORI is *optional* and only useful when using more than one quadrupole origin. It specifies the geometry of the origin for the quadrupole integrals. These geometries correspond to the \mathbf{R}_j in Equation 1 in Section 1. The values entered must be real numbers and are given in Cartesians (x,y,z). Each line corresponds to a choice of origin, j .

Only one set of quadrupole integrals at a given origin is needed from “qc.out”. The quadrupole integrals at all other origins are computed relative to the integrals read from “qc.out” (the first entry of QORI should correspond to the origin of the quadrupole integrals from “qc.out”). Additionally, the dipole is used to compute new quadrupole integrals. Please make sure the units of the dipole and origin are consistent with the units used for the quadrupole integral read from “qc.out”. By default, DQΦpac will assume that all origins are (0.0,0.0,0.0). The following example corresponds to an NQOR of 3.

Input Example

```
QORI
0.0 0.0 0.0
2.0 2.0 2.0
-2.0 0.5 100.0
END
```

8.9 NEOR

NEOR The keyword NEOR is *optional*. It specifies the number of origins used for the electrostatic potential integrals, which is the value of N_Φ in Equation 1 in Section 1. The value entered must be an integer. The default is 1.

Input Example

```
NEOR
3
```

8.10 BETA

BETA The keyword BETA is *optional*. It specifies the β parameter. As a reminder β weights the importance of the electrostatic potential integrals for each choice of origin, which is the β_k in Equation 1 in Section 1. The values entered must be real numbers. Each line corresponds to each choice of origin, k . If BETA is not specified, each value of β_k is assumed to be 0.0. The example below is for a case when NEOR is 3.

Input Example

```
BETA
10.0
1.0
0.5
END
```

8.11 CHARGE

CHARGE The keyword CHARGE is *optional*. It specifies the charge of the molecule. This is only used when computing quadrupole integrals at more than one origin (see ALPHA, NQOR, and QORI). The value must be an integer. The default is 0 (neutral molecule).

Input Example

```
CHARGE
1
```

8.12 COORD

COORD The keyword COORD is *mandatory* when using quadrupole integrals that are computed with Q-Chem (see Section 6.1). It specifies the number of nuclei. It is used in conjunction with the QNUC and NUCL keywords to add on the nuclear contribution to the quadrupole integrals of Q-Chem.

Input Example

```
COORD
3
```

8.13 QNUC

QNUC The keyword QNUC is *mandatory* when using quadrupole integrals that are computed with Q-Chem (see Section 6.1). The QNUC keyword needs to be specified, along with the COORD and NUCL keywords, to add the nuclear contribution to the quadrupole integrals of Q-Chem.

Input Example

```
QNUC
```

8.14 NUCL

NUCL The keyword NUCL is *mandatory* when using quadrupole integrals that are computed with Q-Chem (see Section 6.1). It specifies the charge and coordinates of the nuclei, which are needed to add the nuclear contribution to the quadrupole integrals of Q-Chem. The QNUC and COORD keywords also need to be specified. All values for the charge and coordinates must be real numbers, and the geometries must be in Angstroms. Below is an example for the H₂O molecule and corresponds to the case when COORD is 3. The order of each line is nuclear charge, x, y, then z.

Input Example

```
NUCL  
1.0 2.22 3.71 0.22  
1.0 1.29 10.71 7.22  
8.0 101.29 0.71 11.22  
END
```

8.15 ETHRESH

ETHRESH The keyword ETHRESH is *mandatory* for weighted DQ Φ with the weighting function in Equation 3 in Section 1. It specifies the value of the E_{thresh} parameter (in a.u.) in Equation 3 in Section 1. The value entered must be a real number. There is no default at present. The ETHRESH keyword is mutually exclusive with the ZSIG keyword.

Input Example

```
ETHRESH  
-184.203
```

8.16 WALPH

WALPH The keyword WALPH is *mandatory* for weighted $DQ\Phi$ with the weighting function in Equation 3 in Section 1. It specifies the value of the α_{Weight} parameter in Equation 3 in Section 1. The value entered must be a real number. There is no default at present, but the 0.005 a.u. value is recommended. The WALPH keyword is mutually exclusive with the ZSIG keyword.

Input Example

```
WALPH  
0.005
```

8.17 ZSIG

ZSIG The keyword ZSIG is *mandatory* for weighted $DQ\Phi$ with the weighting function in Equation 4 of Section 1. It specifies the value of the σ_Z parameter (in kcal/mol) in Equation 4 in Section 1. The value entered must be a real number. There is no default at present. The ZSIG keyword is mutually exclusive with the ETHRESH and WALPH keywords.

Input Example

```
ZSIG  
80.0
```

8.18 FPRINT

FPRINT The keyword FPRINT is *optional*. If entered, $DQ\Phi_{\text{pac}}$ will print out .dat files to the working directory. The .dat files contain data listed in a list format. Many files are printed, and the number of files is dependent on the specific calculation. Files

containing weighted properties will only be printed for a weighted calculation and will have a “W” in front of the filename (e.g., “WAdiaX-diag.dat”). Diabatic properties begin with “Dia” instead of “Adia” (e.g., “DiaX-diag.dat”). Below is a list of some .dat files and a brief description:

1. “AdiaE.dat” - adiabatic energies
2. “DiaE-diag.dat” - diabatic energies
3. “DiaE-off.dat” - diabatic couplings
4. “AdiaX-diag.dat” - adiabatic x component of the dipole (diagonal part)
5. “AdiaX-off.dat” - adiabatic x component of the dipole (off-diagonal part)
6. “AdiaY-diag.dat” - adiabatic y component of the dipole (diagonal part)
7. “AdiaY-off.dat” - adiabatic y component of the dipole (off-diagonal part)
8. “AdiaZ-diag.dat” - adiabatic z component of the dipole (diagonal part)
9. “AdiaZ-off.dat” - adiabatic z component of the dipole (off-diagonal part)
10. “AdiaR-diag.dat” - adiabatic magnitude of the dipole (diagonal part)
11. “AdiaR-off.dat” - adiabatic magnitude of the dipole (off-diagonal part)
12. “AdiaXX-diag.dat” - adiabatic x^2 component of the quadrupole (diagonal part)
13. “AdiaXX-off.dat” - adiabatic x^2 component of the quadrupole (off-diagonal part)
14. “AdiaYY-diag.dat” - adiabatic y^2 component of the quadrupole (diagonal part)
15. “AdiaYY-off.dat” - adiabatic y^2 component of the quadrupole (off-diagonal part)
16. “AdiaZZ-diag.dat” - adiabatic z^2 component of the quadrupole (diagonal part)
17. “AdiaZZ-off.dat” - adiabatic z^2 component of the quadrupole (off-diagonal part)
18. “AdiaRRN-diag.dat” - adiabatic quadrupole for origin N (diagonal part)
19. “AdiaRRN-off.dat” - adiabatic quadrupole for origin N (off-diagonal part)

20. “AdiaEPN-diag.dat” - adiabatic electrostatic potential for origin N (diagonal part)
21. “AdiaEPN-off.dat” - adiabatic electrostatic potential for origin N (off-diagonal part)
22. “DiaP-diag.dat” - diabatic property matrix (diagonal part)

An element of the property matrix is defined as

$$P_{12} = \sqrt{\gamma |\boldsymbol{\mu}_{12}|^2 + \sum_j \alpha_j |Q_{12}^{(j)}|^2 + \sum_k \beta_k |\Phi_{12}^{(k)}|^2} \quad (5)$$

where $\boldsymbol{\mu}$ is the dipole, Q is the quadrupole, and Φ is electrostatic potential. Further definitions can be found in Section 1, specifically in the text under Equation 1. The property matrix is useful for labeling the diabatic states correctly.

23. “DiaP-off.dat” - diabatic property matrix (off-diagonal part)

Input Example

FPRINT

9 Sample Input and Output

The following is sample input ("qc.dqv"). This input corresponds to a $DQ\Phi$ calculation that uses quantities computed from Molpro. The calculation includes 3 states, 1 origin for the quadrupole, an α value of 1.0, two choices of origin for the electrostatic potential, and a β value of 1.0 for each electrostatic potential integral.

```
SOFT
MOLPRO
```

```
METH
DQV
```

```
NSTATE
3
```

```
NQOR
1
```

```
ALPHA
1.0
END
```

```
NEOR
2
```

```
BETA
1.0
1.0
END
```

The following is from the Molpro output ("qc.out") and contains the quantities that will be used by $DQ\Phi$ pac.

```
***,
memory, 200,m
basis={
!
! HYDROGEN      (5s) -&gt; [3s]
! HYDROGEN      (5s) -&gt; [3s]
s, H , 74.6900000, 11.2300000, 2.5460000, 0.7130000, 0.2249000
c, 1.3, 0.0253740, 0.1896840, 0.8529330
```

```

c, 4.4, 1
c, 5.5, 1
}
bohr
geometry={
H1  0.000  0.000  0.010
H2  0.000  0.000  1.490
H3  0.000  0.000  9.99
H4  0.000  0.000  11.51
}

{hf;
wf 4,1,0;
CLOSED 0}
{multi, MAXIT=200;
CLOSED 0;
config;
WF 4,1,0;
state, 3;
natorb, 2140.2
print, ref1
natorb,ci,print}

{ci;noexc;State,3;save,6000.2}
{ci;TRANS, 6000.2, 6000.2,DM}
{ci;TRANS, 6000.2, 6000.2,SM,,0,0,0}
{ci;TRANS, 6000.2, 6000.2,pot,,0,0,0.75}
{ci;TRANS, 6000.2, 6000.2;
dm,4300.2}

{matrop;
load,trdm,dens,4300.2,transition,stateb=1.1,statek=1.1;
load,pot,oper,pot(0),,0,0,0.75
trace,ep,trdm,pot,1,0
load,trdm,dens,4300.2,transition,stateb=1.1,statek=2.1;
load,pot,oper,pot(0),,0,0,0.75
trace,ep,trdm,pot,1,0
load,trdm,dens,4300.2,transition,stateb=1.1,statek=3.1;
load,pot,oper,pot(0),,0,0,0.75
trace,ep,trdm,pot,1,0
load,trdm,dens,4300.2,transition,stateb=2.1,statek=2.1;
load,pot,oper,pot(0),,0,0,0.75
trace,ep,trdm,pot,1,0
load,trdm,dens,4300.2,transition,stateb=2.1,statek=3.1;
load,pot,oper,pot(0),,0,0,0.75

```

```
trace,ep,trdm,pot,1,0
load,trdm,dens,4300.2,transition,stateb=3.1,statek=3.1;
load,pot,oper,pot(0),,0,0,0.75
trace,ep,trdm,pot,1,0}
```

```
{matrop;
load,trdm,dens,4300.2,transition,stateb=1.1,statek=1.1;
load,pot,oper,pot(1),,0,0,0.5
trace,ep,trdm,pot,1,0
load,trdm,dens,4300.2,transition,stateb=2.1,statek=1.1;
load,pot,oper,pot(1),,0,0,0.5
trace,ep,trdm,pot,1,0
load,trdm,dens,4300.2,transition,stateb=2.1,statek=2.1;
load,pot,oper,pot(1),,0,0,0.5
trace,ep,trdm,pot,1,0
load,trdm,dens,4300.2,transition,stateb=1.1,statek=3.1;
load,pot,oper,pot(1),,0,0,0.5
trace,ep,trdm,pot,1,0
load,trdm,dens,4300.2,transition,stateb=2.1,statek=3.1;
load,pot,oper,pot(1),,0,0,0.5
trace,ep,trdm,pot,1,0
load,trdm,dens,4300.2,transition,stateb=3.1,statek=3.1;
load,pot,oper,pot(1),,0,0,0.5
trace,ep,trdm,pot,1,0}
```

...

```
RESULTS FOR STATE 1.1
```

```
=====
```

...

```
!MRCI STATE 1.1 Energy                -2.258327276828
```

...

```
RESULTS FOR STATE 2.1
```

```
=====
```

...

```
!MRCI STATE 2.1 Energy                -1.671360324494
```

...

```
RESULTS FOR STATE 3.1
```

```
=====
```

...

!MRCI STATE 3.1 Energy -1.660918269362

...

1PROGRAM * CI (Multireference internally contracted CI) Authors: H.-J. Werner, P.J.

Transition moment calculation

=====

Ket wavefunction restored from record 6000.2 to file 8

Bra wavefunction restored from record 6000.2 to file 7

!MRCI overlap	<1.1 1.1>	1.000000000000		
!MRCI expc	<1.1 DMZ 1.1>	-0.000007449185 au =	-0.000018932699 Debye	
!MRCI trans	<1.1 DMZ 2.1>	1.669632418742 au =	4.243504362826 Debye	
!MRCI trans	<1.1 DMZ 3.1>	0.752100031138 au =	1.911522397141 Debye	
!MRCI trans	<2.1 DMZ 1.1>	1.669632418742 au =	4.243504362826 Debye	
!MRCI overlap	<2.1 2.1>	1.000000000000		
!MRCI expc	<2.1 DMZ 2.1>	0.002712838007 au =	0.006894894821 Debye	
!MRCI trans	<2.1 DMZ 3.1>	-0.002551226911 au =	-0.006484147292 Debye	
!MRCI trans	<3.1 DMZ 1.1>	0.752100031138 au =	1.911522397141 Debye	
!MRCI trans	<3.1 DMZ 2.1>	-0.002551226911 au =	-0.006484147292 Debye	
!MRCI overlap	<3.1 3.1>	1.000000000000		
!MRCI expc	<3.1 DMZ 3.1>	-0.002865543568 au =	-0.007283008222 Debye	
...				
!MRCI overlap	<1.1 1.1>	1.000000000000		
!MRCI expc	<1.1 DMZ 1.1>	-0.000007449185 au =	-0.000018932699 Debye	
Operator origin	0.00000000	0.00000000	0.00000000	
!MRCI expc	<1.1 XX(0) 1.1>	-2.722124955020		
!MRCI expc	<1.1 YY(0) 1.1>	-2.722124955020		
!MRCI expc	<1.1 ZZ(0) 1.1>	-2.047847308884		

!MRCI trans	<1.1 DMZ 2.1>	1.669632418742 au =	4.243504362826 Debye
!MRCI trans	<1.1 XX(0) 2.1>	-0.000000280477	
!MRCI trans	<1.1 YY(0) 2.1>	-0.000000280477	
!MRCI trans	<1.1 ZZ(0) 2.1>	26.853789792058	
!MRCI trans	<1.1 DMZ 3.1>	0.752100031138 au =	1.911522397141 Debye
!MRCI trans	<1.1 XX(0) 3.1>	0.000018816303	
!MRCI trans	<1.1 YY(0) 3.1>	0.000018816303	
!MRCI trans	<1.1 ZZ(0) 3.1>	-7.870402443242	
!MRCI trans	<2.1 DMZ 1.1>	1.669632418742 au =	4.243504362826 Debye
!MRCI trans	<2.1 XX(0) 1.1>	-0.000000280477	
!MRCI trans	<2.1 YY(0) 1.1>	-0.000000280477	
!MRCI trans	<2.1 ZZ(0) 1.1>	26.853789792058	
!MRCI overlap	<2.1 2.1>	1.000000000000	
!MRCI expec	<2.1 DMZ 2.1>	0.002712838007 au =	0.006894894821 Debye
!MRCI expec	<2.1 XX(0) 2.1>	-3.086200758574	
!MRCI expec	<2.1 YY(0) 2.1>	-3.086200758574	
!MRCI expec	<2.1 ZZ(0) 2.1>	-4.387751172844	
!MRCI trans	<2.1 DMZ 3.1>	-0.002551226911 au =	-0.006484147292 Debye
!MRCI trans	<2.1 XX(0) 3.1>	-0.005386800320	
!MRCI trans	<2.1 YY(0) 3.1>	-0.005386800320	
!MRCI trans	<2.1 ZZ(0) 3.1>	-0.021193570018	
!MRCI trans	<3.1 DMZ 1.1>	0.752100031138 au =	1.911522397141 Debye
!MRCI trans	<3.1 XX(0) 1.1>	0.000018816303	
!MRCI trans	<3.1 YY(0) 1.1>	0.000018816303	
!MRCI trans	<3.1 ZZ(0) 1.1>	-7.870402443242	
!MRCI trans	<3.1 DMZ 2.1>	-0.002551226911 au =	-0.006484147292 Debye
!MRCI trans	<3.1 XX(0) 2.1>	-0.005386800320	
!MRCI trans	<3.1 YY(0) 2.1>	-0.005386800320	
!MRCI trans	<3.1 ZZ(0) 2.1>	-0.021193570018	
!MRCI overlap	<3.1 3.1>	1.000000000000	

```

!MRCI expec      <3.1|DMZ|3.1>    -0.002865543568 au =    -0.007283008222 Debye

!MRCI expec      <3.1|XX(0)|3.1>    -3.100247042155
!MRCI expec      <3.1|YY(0)|3.1>    -3.100247042155
!MRCI expec      <3.1|ZZ(0)|3.1>    -4.521311447439

```

...

```

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 1.1/1.1)

```

```

Operator POT(0) loaded to POT

```

```

EP = tr[TRDM*POT] = -2.10504393

```

```

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 1.1/2.1)

```

```

Operator POT(0) loaded to POT

```

```

EP = tr[TRDM*POT] = -0.01230245

```

```

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 1.1/3.1)

```

```

Operator POT(0) loaded to POT

```

```

EP = tr[TRDM*POT] = 0.00447836

```

```

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 2.1/2.1)

```

```

Operator POT(0) loaded to POT

```

```

EP = tr[TRDM*POT] = -2.05509656

```

```

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 2.1/3.1)

```

```

Operator POT(0) loaded to POT

```

```

EP = tr[TRDM*POT] = 0.13965997

```

```

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 3.1/3.1)

```

```

Operator POT(0) loaded to POT

```

```

EP = tr[TRDM*POT] = -1.73168521

```

...

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 1.1/1.1)

Operator POT(1) loaded to POT

EP = tr[TRDM*POT] = -1.98997134

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 2.1/1.1)

Operator POT(1) loaded to POT

EP = tr[TRDM*POT] = -0.11060064

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 2.1/2.1)

Operator POT(1) loaded to POT

EP = tr[TRDM*POT] = -1.95295932

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 1.1/3.1)

Operator POT(1) loaded to POT

EP = tr[TRDM*POT] = -0.26183121

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 2.1/3.1)

Operator POT(1) loaded to POT

EP = tr[TRDM*POT] = 0.10559738

Transition density TRDM read from reco 4300.2 Type=MRCI/TRANSITION (states 3.1/3.1)

Operator POT(1) loaded to POT

EP = tr[TRDM*POT] = -1.70932097

... Then the calculation ends.

The following is the resulting output from DQΦpac.

Beginning DQPhi diabaticization

This is Version 2.0

First, I am reading qc.dqv

Software keyword:MOLPRO

Method keyword:DQV

of states: 3

of origins for quadrupole: 1

of origins for elec. pot: 2

Could not find string 'GAMMA' in qc.dqv
Assuming Gamma=1.0

Alpha value(s) (in (a.u.)**(-2)):
1.000

Could not find string 'STATES' in qc.dqv
Assuming states 1 to NSTATE

Could not find string 'QORI' in qc.dqv
Assuming all origins are zero

Beta value(s) (in (a.u.)**4):
1.000
1.000

Now, I am reading qc.out

So far, the output has just repeated back information regarding the reading of 'qc.dqv'.

Finally, I will perform DQPhi diabaticization

The variables that are read in from 'qc.out' are printed.

ADIABATIC ENERGIES (in a.u.)

V(1, 1)= -2.25833

V(2, 2)= -1.67136

V(3, 3)= -1.66092

Adiabatic X component of dipole (in a.u.)

000.00000	000.00000	000.00000
	000.00000	000.00000
		000.00000

Adiabatic Y component of dipole (in a.u.)

000.00000	000.00000	000.00000
	000.00000	000.00000
		000.00000

Adiabatic Z component of dipole (in a.u.)

-000.00001	001.66963	000.75210
	000.00271	-000.00255
		-000.00287

Adiabatic magnitude of dipole (in a.u.)

000.00001	001.66963	000.75210
	000.00271	000.00255
		000.00287

Adiabatic XX (in (a.u.)**2) for origin: 1

-002.72212	-000.00000	000.00002
	-003.08620	-000.00539
		-003.10025

Adiabatic YY (in (a.u.)**2) for origin: 1

```
-002.72212  -000.00000   000.00002
             -003.08620  -000.00539
                                 -003.10025
```

Adiabatic ZZ (in (a.u.)**2) for origin: 1

```
-002.04785   026.85379  -007.87040
             -004.38775  -000.02119
                                 -004.52131
```

The following is the quadrupole matrix. It is computed from the components taken from “qc.out” (the XX, YY, and ZZ values from above).

Adiabatic RR (in (a.u.)**2) for origin: 1

```
-007.49210   026.85379  -007.87036
             -010.56015  -000.03197
                                 -010.72181
```

The electrostatic potential for each origin is printed below.

Adiabatic EP (in (a.u.)**(-1)) for origin: 1

```
-002.10504  -000.01230   000.00448
             -002.05510   000.13966
                                 -001.73169
```

Adiabatic EP (in (a.u.)**(-1)) for origin: 2

```
-001.98997   000.00000  -000.26183
             -001.95296   000.10560
                                 -001.70932
```

The number of iterations to converge the Jacobi Sweep algorithm is below. It usually converges within 3-7 iterations.

Transformation matrix obtained in 3 iterations

Below is the adiabatic-to-diabatic transformation matrix. It is the **T** matrix in Equation 1 in Ref. [1].

Transformation matrix (rows=diabats,columns=adiabats)

000.72613	000.66031	-000.19166
-000.68756	000.69711	-000.20321
-000.00057	000.27934	000.96019

Below are the diabatic properties. They are computed by transforming the input properties with the adiabatic-to-diabatic transformation matrix.

Diabatic X component of dipole (in a.u.)

000.00000	000.00000	000.00000
	000.00000	000.00000
		000.00000

Diabatic Y component of dipole (in a.u.)

000.00000	000.00000	000.00000
	000.00000	000.00000
		000.00000

Diabatic Z component of dipole (in a.u.)

001.39344	000.07709	000.86204
	-001.38844	-000.81826
		-000.00516

Diabatic magnitude of dipole (in a.u.)

001.39344	000.07709	000.86204
	001.38844	000.81826
		000.00516

Diabatic XX (in (a.u.)**2) for origin: 1

-002.89340	-000.18087	-000.00068
	-002.91314	-000.00043
		-003.10204

Diabatic YY (in (a.u.)**2) for origin: 1

```
-002.89340  -000.18087  -000.00068
              -002.91314  -000.00043
                  -003.10204
```

Diabatic ZZ (in (a.u.)**2) for origin: 1

```
024.78811   000.35792  -000.04025
              -031.22282   000.04069
                  -004.52219
```

Diabatic RR (in (a.u.)**2) for origin: 1

```
019.00131  -000.00382  -000.04161
              -037.04909   000.03983
                  -010.72628
```

Diabatic EP (in (a.u.)**(-1)) for origin: 1

```
-002.09375   000.01190   000.02968
              -002.07862   000.02976
                  -001.71946
```

Diabatic EP (in (a.u.)**(-1)) for origin: 2

```
-001.99348   000.10265  -000.16038
              -001.95893   000.19603
                  -001.69985
```

Below is the diabatic property matrix. It is useful for determining the nature of the diabatic states. See Equation 5 in Section 8.

Diabatic property matrix (in a.u.)

```
019.27043   000.12898   000.87832
              037.18496   000.84288
                  010.99541
```

Below is the diabatic Hamiltonian. It is the \mathbf{U} matrix in Equation 2 in Ref. [1].

Diabatic Hamiltonian - Matrix Form

Warning! The ordering of the states is arbitrary

Use the transformation matrix and property matrix to map to initial
adiabats.

```
-001.98046  000.29345  -000.00168  
           -001.94841  -000.00227  
                   -001.66173
```

This is the message printed when the program runs successfully.

Normal termination

Acknowledgements

We would like to thank Jingjing Zheng for debugging early versions of DQΦpac and Yihan Shao for implementing transition integrals in Q-Chem. We would also like to thank Xuefei Xu, Kelsey Parker, Wei Lin, and Zoltan Varga for testing DQΦpac and providing useful feedback throughout its development.

References

- [1] C. E. Hoyer, X. Xu, D. Ma, L. Gagliardi, and D. G. Truhlar, "Diabatization based on the dipole and quadrupole: The DQ method," *J. Chem. Phys.*, vol. 141, p. 114104, Sep 2014.
- [2] C. E. Hoyer, K. Parker, L. Gagliardi, and D. G. Truhlar, "The DQ and DQ Φ electronic structure diabatization methods: Validation for general applications," *J. Chem. Phys.*, vol. 144, no. 19, p. 194101, 2016.
- [3] J. E. Subotnik, S. Yeganeh, R. J. Cave, and M. A. Ratner, "Constructing diabatic states from adiabatic states: extending generalized Mulliken-Hush to multiple charge centers with Boys localization," *J. Chem. Phys.*, vol. 129, p. 244101, Dec 2008.
- [4] H.-J. Werner and W. Meyer, "MCSCF study of the avoided curve crossing of the two lowest $^1\Sigma^+$ states of LiF," *J. Chem. Phys.*, vol. 74, no. 10, pp. 5802–5807, 1981.
- [5] K. Samanta, J. M. Beames, M. I. Lester, and J. E. Subotnik, "Quantum dynamical investigation of the simplest Criegee intermediate CH₂OO and its O-O photodissociation channels," *J. Chem. Phys.*, vol. 141, p. 134303, Oct 2014.
- [6] C. E. Hoyer, L. Gagliardi, and D. G. Truhlar *DQ Φ pac, Version 2.0*, available for free at <http://comp.chem.umn.edu/dqphipac>.