

# Duda 1

José Antonio QUIÑONERO GRIS

June 20, 2024

Hasta ahora había estado haciendo el programa sobre la 1-RDM  $D_{ij}^{(1)}$ , 2-RDM  $D_{ijkl}^{(2)}$ , integrales monoelectrónicas sobre el hamiltoniano  $h_{ij}$  e integrales bielectrónicas  $\langle ij | kl \rangle$  para un ejemplo (una de las moléculas de H<sub>2</sub>O del scan) para ver simplemente que funcionaba.

Ya lo tengo todo programado y, aunque funciona (obviamente todo compila bien), no obtengo resultados siquiera similares a los que obtengo con DALTON o con Gaussian09.

Aquí desarrollo a detalle todo lo que hago hasta calcular el primer valor de energía.

Nota: añadido en rojo dudas y/o notas en sitios donde creo que puede estar el error o fragmentos en los que tengo dudas.

Mi intención no es que leas todo el documento al detalle, ni mucho menos el código (que incluyo más bien para tenerlo a mano por si tengo que consultarte algo), por eso indico con rojo las dudas. Mi intención más bien es que me puedas corregir si tengo algún malentendido conceptual o con Dalton.

## 1 Cálculos con DALTON

Para los cálculos en DALTON, uso el siguiente archivo .dal para distintos archivos H2O- $n$ .mol en los que voy cambiando la distancia H-O (para ambos H manteniendo el ángulo H-O-H)

---

```

1 **DALTON INPUT
2 .RUN RESPONSE
3 **WAVE FUNCTIONS
4 .HF
5 .MP2
6 .MCSCF
7 .NEPOLD
8 *SCF INPUT
9 .DOUBLY OCCUPIED
10 3 1 1 0
11 *CONFIGURATION INPUT
12 .SYMMETRY
13 1
14 .SPIN MULTIPLICITY
15 1
16 .INACTIVE
17 1 0 0 0
18 .CAS SPACE
19 4 2 2 0
20 .ELECTRONS
21 8
22 **END OF DALTON INPUT

```

---

## 2 Extracción y lectura de $D_{ij}^{(1)}$ , $D_{ijkl}^{(2)}$ , $h_{ij}$ y $\langle ij | kl \rangle$

Las RDMs las extraigo de SIRIFC con la subrutina RD\_SIRIFC.

### 2.0.1 Extracción de $D^{(1)}$

Primero extraigo el array con los elementos de la triangular en AVEC y creo una matriz DVMAT completa:

---

```
1 ! Initialize AVEC & DVMAT
2 !   AVEC: vector with upper triangular elements; AVEC(NNASHX)
3 !   DVMAT: complete matrix; DVMAT(NASHT, NASHT)
4 allocate(avec(nnashx), dvmat(nasht, nasht))
5 avec = 0.d0
6 dvmat = 0.d0
7
8 ! Extract DV from SIRIFC
9 call RD_SIRIFC('DV', FOUND, AVEC)
10
11 ! Complete the full matrix, DVMAT, from the triangular, AVEC
12 k = 0
13 do i = 1, nasht
14   do j = 1, i
15     k = k + 1
16     dvmat(i,j) = avec(k)
17     dvmat(j,i) = avec(k) ! mirror the upper triangular
18   end do
19 end do
```

---

Esta matriz se corresponde al bloque activo-activo,  $D_{a-a}^{(1)}$ , de la 1-RDM completa

$$D^{(1)} = \begin{pmatrix} D_{i-i}^{(1)} & D_{i-a}^{(1)} & D_{i-v}^{(1)} \\ D_{i-a}^{(1)} & D_{a-a}^{(1)} & D_{a-v}^{(1)} \\ D_{i-v}^{(1)} & D_{a-v}^{(1)} & D_{v-v}^{(1)} \end{pmatrix}. \quad (1)$$

Creo la matriz completa teniendo en cuenta que el bloque inactivo-inactivo viene dado por

$$D_{ij}^{(1)} = 2\delta_{ij}, \quad (2)$$

según

---

```
1 ! Initialize D1MO
2 allocate(d1mo(norbt, norbt))
3 d1mo = 0.d0
4
5 ! Fill the inactive orbitals
6 do i = 1, nisht
7   d1mo(i,i) = 2.d0
8 end do
9
10 ! Fill the active orbitals
```

```

11         do i = 1, nasht
12             do j = 1, nasht
13                 d1mo(nisht+i, nisht+j) = dvmat(i,j)
14             end do
15         end do

```

---

Finalmente escribo d1mo a un archivo.

## 2.0.2 Extracción de $D^{(2)}$

De manera análoga, extraigo el array AVEC y primero creo la matriz empaquetada AMAT

---

```

1         ! Initialize AVEC, AMAT & PVMAT
2         !     AVEC: vector with matrix elements; AVEC(NNASHX*NNASHX)
3         !     AMAT: 2D matrix from AVEC; AMAT(NNASHX,NNASHX)
4         !     PVMAT: 4D AMAT unpacked matrix; PVMAT(NASHT,NASHT,NASHT,NAHST)
5         allocate(avec(nnashx*nnashx), amat(nnashx,nnashx),
6         &         pvmat(nasht,nasht,nasht,nasht))
7         avec = 0.d0
8         amat = 0.d0
9         pvmat = 0.d0
10
11         ! Call RD_SIRIFC
12         call RD_SIRIFC('PV', FOUND, AVEC)
13
14         ! Create AMAT from AVEC
15         ijk1 = 0
16         do kl=1, nnashx
17             do ij=1, nnashx
18                 ijk1 = ijk1 + 1
19                 amat(ij,kl) = avec(ijk1)
20             end do
21         end do

```

---

que desempaqueto en pvmat según

---

```

1         ! Unpack the PV matrix
2         kl = 0
3         do k=1, nasht
4             do l=1, k
5                 kl = kl + 1
6                 ij = 0
7                 do i=1, nasht
8                     do j=1, i
9                         ij = ij + 1
10                        pvmat(i,j,k,l) = amat(ij,kl)
11                        pvmat(j,i,k,l) = amat(ij,kl)
12                        pvmat(i,j,l,k) = amat(ij,kl)
13                        pvmat(j,i,l,k) = amat(ij,kl)
14                    end do
15                end do

```

```

16         end do
17     end do

```

---

Construyo la matriz completa para todo el CAS y completo el bloque inactivo-inactivo como

$$D_{ijkl}^{(2)} = 2\delta_{ij}\delta_{kl} - \delta_{il}\delta_{jk}, \quad (3)$$

---

```

1      ! Initialize D2M0
2      allocate(d2mo(norbt, norbt, norbt, norbt))
3      d2mo = 0.d0
4
5      ! For the inactive space
6      do i=1, nisht
7          do j=1, i
8              ! Occupied orbitals:
9              d2mo(i,i,j,j) = 2.d0    ! Case for i=j, k=l
10             ! Interchange of electrons:
11             ! (-1 to avoid re-countering the same pair)
12             d2mo(i,j,j,i) = -1.d0   ! Case for i=l, j=k
13         end do
14     end do

```

---

y el **bloque activo-activo** a partir de la matriz que había sacado anteriormete como

---

```

1      ! Fill the active orbitals
2      do i=1, nasht
3          do j=1, i
4              do k=1, nasht
5                  do l=1, k
6                      kk = k + nisht
7                      ll = l + nisht
8                      d2mo(i,j,kk,ll) = pvmat(i,j,k,l)
9                      d2mo(j,i,kk,ll) = pvmat(j,i,k,l)
10                     d2mo(i,j,ll,kk) = pvmat(i,j,l,k)
11                     d2mo(j,i,ll,kk) = pvmat(j,i,l,k)
12                 end do
13             end do
14         end do
15     end do

```

---

Por último, los bloques cruzados inactivo-activo como se indica en el artículo que te mandé (Maradzike, 2017) en las ecuaciones 38 y 39:

$$D_{ijtu}^{(2)} = D_{tuij}^{(2)} = D_{tu}^{(1)} \delta_{ij}, \quad (4)$$

$$D_{iutj}^{(2)} = D_{tjiu}^{(2)} = -\frac{1}{2} D_{tu}^{(1)} \delta_{ij}, \quad (5)$$

para  $i, j$  orbitales inactivos y  $t, u$  activos. Lo **programo** como

---

```

1      ! Fill the (inactive-active) crossed blocks

```

```

2      do i=1, norbt
3          do it=1, norbt
4              do iu=1, norbt
5
6  C              ! if (i,j \in inactive):
7                  if ( i.le.nisht .and. j.le.nisht
8  C              ! or (i,j \in virtual):
9                  & .or. i.gt.nisht+nasht .and. j.gt.nisht+nasht
10 C              ! and (t,u \in active), i.e. (t,u not \in inactive):
11                  & .and. it.gt.nisht .and. iu.gt.nisht
12 C              ! and (t,u not \in virtual):
13                  & .and. it.le.nisht+nasht .and. iu.le.nisht+nasht ) then
14 C
15                  ! D^(2)_{ijtu} = D^(2)_{tuij} = D^(1)_{tu} d_{ij}
16                  d2mo(i,i,it,iu) = dvmat(it,iu)
17                  d2mo(it,iu,i,i) = dvmat(it,iu)
18 C
19                  ! D^(2)_{iutj} = D^(2)_{tjiu} = -1/2 D^(1)_{tu} d_{ij}
20                  d2mo(i,iu,it,i) = -1.d0/2.d0 * dvmat(it,iu)
21                  d2mo(it,i,i,iu) = -1.d0/2.d0 * dvmat(it,iu)
22 C
23                  end if
24              end do
25          end do
26      end do

```

---

Escribo la matriz completa a un archivo como

---

```

1 write(...) k, 1, i, j, d2mo(k,1,i,j)

```

---

### 2.0.3 Extracción de $\mathbf{h}$ y $\langle ij | kl \rangle$

Básicamente, uso las rutinas que me proporcionaste de Neptunus con alguna modificación. En estos días de desquicio he estado probando todo lo que se me ha ocurrido. Entre otras, he sacado el archivo `molecule.fmt`, del que leo las matrices. Obtengo los mismos resultados con *mis matrices* que con éstas últimas, así que entiendo que aquí no hay error.

### 2.0.4 Lectura de $\mathbf{h}$ y $\mathbf{D}^{(1)}$

Como las escribo por índices, leo los índices y su elemento. Además son simétricas así que no puede haber fallo en liarme con intercambiar los índices de la escritura a la lectura.

### 2.0.5 Extracción de $\mathbf{D}^{(2)}$ y $\langle ij | kl \rangle$

Leo la  $\mathbf{D}^{(2)}$ , siguiendo como está escrita, según

---

```

1      ! Read the triangular part of Amat
2      iind = -1
3      jind = -1
4      kind = -1

```

```

5      lind = -1
6      do
7          read(nlu,*,iostat=ios) kind, lind, iind, jind, xint
8          if (ios /= 0) stop
9          if (kind.eq.0 .or. lind.eq.0 .or. iind.eq.0 .or. jind.eq.0) exit
10
11         Amat(kind,lind,iind,jind) = xint
12     end do
13
14     ! Simetrize the 4D tensor
15     if (is_triangular .eqv. .true.) then
16         do k=1, n
17             do l=k, n
18                 do i=1, n
19                     do j=i, n
20                         Amat(j,i,l,k) = Amat(i,j,k,l)
21                         Amat(i,j,l,k) = Amat(i,j,k,l)
22                         Amat(j,i,k,l) = Amat(i,j,k,l)
23                     end do
24                 end do
25             end do
26         end do
27     end if

```

---

Creo que aquí no hay fallo aunque, un punto importante, es que **no cumple la condición de normalización**

$$\text{tr} [\mathbf{D}^{(2)}] = \sum_{ik} D_{ikk}^{(2)} = \frac{N(N-1)}{2} = \binom{N}{2}. \quad (6)$$

La  $\mathbf{D}^{(1)}$  **sí** que cumple la condición de normalización  $\text{tr} [\mathbf{D}^{(1)}] = N$ .

Como te comenté por correo, las integrales bielectrónicas las escribo como

---

```

1 write(...) k, l, i, j, h2mo(k,l,i,j)

```

---

Me comentaste que están en notación química, es decir,  $(kl|ij)$ . Como en el documento uso notación Dirac porque es la que usan los autores de los artículos/libros que he estado siguiendo, **la leo teniendo en cuenta que  $(kl|ij) = \langle ki|lj \rangle$** , en una rutina exactamente igual a la de  $\mathbf{D}^{(2)}$ , pero con la siguiente modificación

---

```

1      ! xint = (kl|ij) -> <ki|lj>
2      Amat(kind,iind,lind,jind) = xint

```

---

### 3 Extracción de los números de ocupación $\{n_p\}$ , $\mathbf{ON}(:)$

Primero chequeo si  $\mathbf{D}^{(1)}$  ya es diagonal y, en caso contrario, la diagonalizo con una subrutina que tengo escrita de Jacobi, y haría una transformación unitaria de las matrices de integrales con los vectores propios. Como la matriz que sale de Dalton ya es diagonal, no desarrollo más esa condición.

Esto me lleva a una duda, **como las RDMs extraídas de Dalton ya son diagonales, ¿entonces**

las matrices de las integrales,  $h_{ij}, \langle ij | kl \rangle$  están ya en la base de orbitales (moleculares) naturales? ¿Básicamente, las matrices de integrales ya están en base de los vectores propios de  $\mathbf{D}^{(1)}$ ?

Entiendo que esto es importante ya que, al fin y al cabo, todas las aproximaciones a  $\mathbf{D}^{(2)}$  son en función a los números de ocupación naturales (valores propios de  $\mathbf{D}^{(1)}$ ), de manera que al calcular  $E_{ee}$  como sumas del estilo  $D_{pqrs}^{(2)} \langle pq | rs \rangle$ , la integral bielectrónica está expresada en base de orbitales naturales (vectores propios de  $\mathbf{D}^{(1)}$ ).

Por tanto, tal cual tengo  $\mathbf{D}^{(1)}$  de Dalton que ya es diagonal, ¿ $n_1 = D_{11}^{(1)}$  se corresponde a la ocupación del orbital cuya integral sobre  $\hat{h}$  es  $h_{11}$ ? O dicho de otra manera, ¿la indexación se corresponde entre todas las matrices?

## 4 Cálculo de la energía

Para no extender más el documento, no entro en las distintas aproximaciones, ya que vale (al menos en un principio) con comprobar que la energía a partir de  $\mathbf{D}^{(1)}$  y  $\mathbf{D}^{(2)}$  es equivalente a la que pueda obtener con Dalton o Gaussian.

La energía la calculo como

$$E = E_{oe} + E_{ee}, \quad (7)$$

donde

$$E_{oe} = \text{tr} [\mathbf{hD}^{(1)}] = \sum_{pq} h_{pq} D_{pq}^{(1)}, \quad (8)$$

y

$$E_{ee} = \frac{1}{2} \sum_{pqrs} D_{pqrs}^{(2)} \langle rs | pq \rangle. \quad (9)$$

Para no entrar tampoco en la conversión de MOs a spin-orbitales, hago el cálculo con los valores que tengo directamente de Dalton, es decir

- $\mathbf{D}^{(1)}$  en base MO con  $0 \leq D_{ii}^{(1)} \leq 2$
- $\mathbf{D}^{(2)}$  en base MO
- $h_{pq}$  en base MO
- $\langle rs | pq \rangle$  (aunque escrito en notación Dirac) en base MO

### 4.1 Ejemplo para H2O-0.mol

Sea el archivo H2O-0.mol

---

```
1 BASIS
2 6-31G**
3 Water using the 6-31G** basis
4 Using automatic symmetry detection.
5 Atomtypes=2
6 Charge=8.0 Atoms=1
7 0 0.000000000 0.37558843 -0.14534628
8 Charge=1.0 Atoms=2
9 H_a 0.000000000 0.78383672 -0.43402142
```

correspondiente a una distancia H–O de  $d = 0.5 \text{ \AA}$ , obtengo

- $E_{\text{oe}} = -124.3105 \text{ Hartrees}$
- $E_{\text{ee}} = 0.4417 \text{ Hartrees}$
- $E = E_{\text{oe}} + E_{\text{ee}} = -123.8688 \text{ Hartrees}$

lo cual no tiene ni medio sentido teniendo en cuenta las energías que obtengo del cálculo en Dalton

- $E_{\text{Dalton}} = -66.16992 \text{ Hartrees}$

que obtengo de la última línea que encuentra con MCSCF energy.

## 4.2 Scan para el H<sub>2</sub>O

En la siguiente tabla incluyo las energías que obtengo para todas las estructuras que, en un principio, tenía creadas con un incremento de  $\Delta d = 0.5 \text{ \AA}$

La cosa es que, no sólo salen energías absurdas, como que la menor energía sea para el caso en el que los átomos están más próximos, si no que tampoco siguen la *tendencia* de las energías obtenidas con Dalton (la forma de la curva), ni hay un mínimo.

**Table 1**

$d \text{ (\AA)}$	$E \text{ (Hartree)}$	$E_{\text{Dalton}} \text{ (Hartree)}$
0.5	-123.8688	-66.1699
1.0	-115.7894	-74.8826
1.5	-111.1165	-76.0579
2.0	-107.7657	-76.1280
2.5	-105.0204	-76.0252
3.0	-102.7526	-75.9260
3.5	-101.2976	-75.8616
4.0	-100.7092	-75.8282