

```

*****
*           *
*   Input   *
*           *
*****

```

Besides the normal GAMESS-UK input directives, to perform an ELMO calculation one needs to specify the "elmo" keyword and to provide additional instructions that are contained in the following three namelists (note that they could also be in part or entirely omitted, as all the variables have default values defined in the code).

&run

- niter (default=100): maximum number of ELMO cycles
- thresh (default=5.0d-07 a.u.): convergence threshold (at present on the maximum component of the energy gradient with respect to the ELMO coefficients)
- lfraginp (default=.false.): flag concerning the format of the localization pattern (see below). If set to .true., pre-localization is turned off (see below).
- lrestart (default=.false.): it has to be set to .true. if performing restart calculation (guess orbitals are read from the file 'elmo' and pre-localization (see below) is turned off).
- ldebug_elmo (default=.false.): turns on debugging prints

&preloc

- npreiter (default=200): maximum number of cycles for the pre-localization algorithm
- prethresh (default=1.0d-10): convergence threshold for the pre-localization algorithm

&conv

- convalg (default='qnewt'): string to chose the ELMO algorithm. It can have the following values:

'scf' (SCF resolution of ELMO eigenvalue equations will be used throughout all the calculation)

'frag' (the same as 'scf', but ELMO eigenvalue equations are solved one fragment at a time)

'conj' (turns on the conjugate gradient method when the maximum component of the gradient drops below 'conjthresh' - see below)

'qnewt' (turns on the quasi-newton method when the maximum component of the gradient drops below 'qnewtthresh' - see below)

- fragthresh (default=5.0d-01): convergence threshold on single fragments, used only when convalg='frag'. Its value is decreased during iterations until global convergence (as given by thresh) is achieved.

- conjthresh (default=5.0d-01): threshold (on MAX grad) for activation of conjugate gradient (when MAX grad > conjthresh the algorithm used is 'scf'). Used when convalg='conj'

- qnewtthresh (default=5.0d-02): threshold (on MAX grad) for activation of Newton methods (when MAX grad > qnewtthresh the algorithm used is 'scf'). Used when convalg='qnewt'

- lsearch (default='brent'): string to chose the algorithm for the one-dimensional search.

Different methods were implemented:

'brent' (Brent method - it uses also the gradient of the energy)

'quad' (quadratic interpolation of points calculated at fixed distances along the direction)

'random' (quadratic interpolation of points calculated at random distances along the direction - generally superior to 'quad')

'cube' (cubic interpolation of points calculated at fixed distances along the direction - it uses also the gradient of the energy)

'brute' (small steps are performed along the line in the direction of decreasing energy until an energy increment is detected. It is not very efficient and is has to be used only when all the other algorithms fail).

'off' (line search turned off => a 'pure' Newton step is performed)

- nanalyt (default=1): number of newton iterations for which the second derivatives of the energy with respect to ELMO coefficients are calculated analytically (with an approximate formula). It is used only when convalg='qnewt'. When iterations exceed nanalyt, the hessian is updated with a numerical formula (as given by 'hessupdt', see below). If nanalyt = 0,

analytic calculation of hessian is turned off and the initial guess for the hessian is an identity matrix.

- lhshift (default=.true.): shifting of hessian eigenvalues turned on
- hessupdt (default = 'bfgs'): string to chose the updating formula for the hessian. It can be 'bfgs' (for the Broyden-Fletcher-Goldfarb-Shanno formula) or 'dfp' (for the Davidon-Fletcher-Powell formula)

After the namelists, the localization pattern has to be given. One can do this on a 'per-ELMO' basis (lfraginp in &run = .false.) or on a 'per-fragment' basis (lfraginp=.true.).

- 'per-ELMO' format

This format could appear a little involved, but it can be understood with a premise. Good guess orbitals for ELMO calculations can be generally obtained by performing a standard RHF calculation and then transforming the MOs according to some 'a posteriori' localization procedure (such as the Boys or Edminston-Ruedenberg criterion). ELMO routines include a subroutine which performs the Pipek-Mezey pre-localization (J. Chem. Phys., 90:4916 1989) and which is invoked prior to the ELMO minimization (unless the variable lrestart in &run is set to .true.). These localized MOs (LMOs) differ from ELMOs as they keep non-null (even if small) components on all the atoms, while ELMOs are forced to use only the AOs centered on predetermined atoms. Thus after pre-localization all the coefficients which are 'forbidden' by the chosen localization pattern are annihilated and kept null during the subsequent ELMO minimization.

Using Mulliken population analysis, one can see that in general LMOs are mostly localized on one, two or three atoms, so that each of them can be 'labelled' with at most three indices (we can call them 'pre-localization indices'). Here are the pre-localization indices for the fluoroformaldehyde molecule (see below), as they are writtten in the output file of the ELMO calculation:

```
MO:   1 Sum =    1.99951      4
1.9995

MO:   2 Sum =    1.95417      4
1.9542

MO:   3 Sum =    2.07343      1  2  !
1.2689      0.8046              !   H2
                                   !   \
MO:   4 Sum =    1.84036      3   !   C1=O3
1.8404              !   /
                                   !   F4
MO:   5 Sum =    2.09158      3   !

2.0916

MO:   6 Sum =    2.00968      4  1
1.5319      0.4777

MO:   7 Sum =    2.05962      4

2.0596

MO:   8 Sum =    1.92676      4

1.9268

MO:   9 Sum =    2.02102      3  1
1.1994      0.8217

MO:  10 Sum =    2.00357      1
```

2.0036

```
MO: 11 Sum = 2.00958 3 1
1.3433 0.6663
```

```
MO: 12 Sum = 1.99922 3
1.9992
```

Then for example the first LMO is localized on atom F4, while the third extends mostly on atoms C1 and H2.

As often pre-localization indices reflect the simple Lewis structure formula of the molecule, one can guess what they will be prior to the pre-localization. In our example one can see that there are four LMOs localized on fluorine (the core and the three lone pairs), three localized on oxygen (the core and the two lone pairs), only one localized on carbon (its core pair), two localized on the C=O double bond, one on the C-F bond and one on the C-H bond. Pre-localization indices are needed to write the input localization pattern, so when uncertainty exists (for example when handling aromatic systems) one should run the ELMO calculation a first time with a guess for the indices and check that they match those printed at the end of the pre-localization procedure (if a mismatch is found the program stops, so one has to correct the input and re-run the calculation).

Now we can write the localization pattern by giving for each ELMO the three pre-localization indices, the number of 'allowed' atomic centers and their indices, together with an optimization flag. The localization pattern adopted for our example is the following:

```
1 0 0 1 1 .t. ! carbon core
3 0 0 1 3 .t. ! oxygen core/lone pair
3 0 0 1 3 .t. ! "
3 0 0 1 3 .t. ! "
4 0 0 1 4 .t. ! fluorine core/lone pair
4 0 0 1 4 .t. ! "
4 0 0 1 4 .t. ! "
4 0 0 1 4 .t. ! "
1 2 0 2 1 2 .t. ! C-H bond
1 3 0 2 1 3 .t. ! C=O double bond
1 3 0 2 1 3 .t. ! "
1 4 0 2 1 4 .t. ! C-F bond
```

Thus first ELMO will use as guess the LMO pre-localized on carbon atom (as indicated by the first three indices '1 0 0') and at the end of the calculation it will be 'extremely' localized on the same atom ('1 1'). Then three ELMOs follow which are strictly localized on oxygen and which will use the pre-localized oxygen core and lone pairs as guess and so on...

It is furthermore to be noted that in the input localization pattern ELMOs can be given in whichever order you want, which is then independent from the order given in the pre-localization pattern.

This example seems particularly trivial as pre-localization indices (apart from zeroes, of course) coincides with the indices of the allowed centers, i.e. the chosen localization pattern coincides with the Lewis structure formula. But one could want e.g. to partially delocalize the oxygen and fluorine lone pair on the adjacent carbon atom using the following pattern:

```
1 0 0 1 1 .t.
3 0 0 1 3 .t.
3 0 0 2 3 1 .t. ! <- changed line
3 0 0 2 3 1 .t. ! <- changed line
4 0 0 1 4 .t.
4 0 0 2 4 1 .t. ! <- changed line
```

```

4  0  0      2      4  1      .t.      ! <- changed line
4  0  0      2      4  1      .t.      ! <- changed line
1  2  0      2      1  2      .t.
1  3  0      2      1  3      .t.
1  3  0      2      1  3      .t.
1  4  0      2      1  4      .t.

```

Then for example the third ELMO uses a LMO pre-localized on O3, but at the end of the calculation it will extend also on C1.

When the allowed centers for lone pairs are different from those for core pairs, it is good for convergence purposes to distinguish between pre-localized core and lone pairs, so to give the right guess to each ELMO. In fact both core and lone pairs pre-localized on a given atom generally have small components on other atoms, but for core pairs they are smaller than for lone pairs. Then one can use the following pattern in place of the previous one:

```

1  0  0      1      1      .t.
3 -1  0      1      3      .t.      ! <- changed line
3  0  0      2      3  1      .t.
3  0  0      2      3  1      .t.
4 -1  0      1      4      .t.      ! <- changed line
4  0  0      2      4  1      .t.
4  0  0      2      4  1      .t.
4  0  0      2      4  1      .t.
1  2  0      2      1  2      .t.
1  3  0      2      1  3      .t.
1  3  0      2      1  3      .t.
1  4  0      2      1  4      .t.

```

where '-1' is used to label pre-localized core orbitals (which will be then used as guess for core ELMOs).

Optimization flags can be used to freeze ELMOs, which are then kept unchanged during the minimization of the other orbitals.

At last, the order of ELMOs in the input localization pattern is not important except for one aspect: if consecutive ELMOs are found with the same number and indices of allowed atomic centers, they are considered to be part of the same molecular fragment (and thus they will be obtained as eigenvalues of the same modified Fock operator), so that one has to check that optimization flags for ELMOs belonging to the same fragment are the same. If one wants for example to freeze core pairs while optimizing lone pairs, but using the same partial basis set for them, it is possible to use a pattern of this type:

```

1  0  0      1      1      .f.
3 -1  0      1      3      .f.
4 -1  0      1      4      .f.
3  0  0      1      3      .t.
3  0  0      1      3      .t.
4  0  0      1      4      .t.
4  0  0      1      4      .t.
4  0  0      1      4      .t.
1  2  0      2      1  2      .t.
1  3  0      2      1  3      .t.
1  3  0      2      1  3      .t.
1  4  0      2      1  4      .t.

```

where the core and lone pairs localized on atom 3 and 4 are made non-consecutive.

- 'per-fragment' format

This format can be used only for restart calculation, for which pre-localization is disabled. Then initial MOs can be taken from previous ELMO calculations or by pre-localizing the RHF MOs with other programs.

Localization patterns could be faster to write down in 'per-fragment' format if one has much fewer fragments than ELMOs. Furthermore it is possible to define fragments which use only some AOs of a given atomic center. Nevertheless, as one has to give AO intervals, the format is dependent on the basis set.

Data required are

nf (number of fragments)

(optflag(i), i = 1, nf)

1 line per fragment containing the number of AOs and the number of occupied ELMOs of the fragment

1 (or more) line(s) per fragment containing the AO indices of the fragment (as intervals)

Then the localization pattern which in the 'per-ELMO' format is written as

```
1 0 0      1      1      .t.
3 0 0      1      3      .t.
3 0 0      1      3      .t.
3 0 0      1      3      .t.
4 0 0      1      4      .t.
4 0 0      1      4      .t.
4 0 0      1      4      .t.
4 0 0      1      4      .t.
1 2 0      2      1 2    .t.
1 3 0      2      1 3    .t.
1 3 0      2      1 3    .t.
1 4 0      2      1 4    .t.
```

in the 'per-fragment' format (for a 6-31G basis set) becomes

```
6                                ! number of fragments
.t. .t. .t. .t. .t. .t.       ! optimization flags (one for each fragment)
9      1                        ! number of AOs and number of ELMOs for fragment 1
9      3
9      4
11     1
18     2
18     1
```

```
1      9      ! AO indices (as interval) for fragment 1
```

```
12     20
```

```
21     29
```

```
1      11
```

```
1      9      !\
```

```
12     20     !/ AO indices (as intervals) for fragment 5
```

```
1      9
```

```
21     29
```

```
*****
*
*   Output   *
*
*****
```

As a result of the ELMO calculations we obtain:

* the output file: "name_file.out"

It is identical to a normal GAMESS-UK output until the call to the ELMO routines. Then a summary of the convergence options and of the localization pattern is reported. The pre-localization pattern is written at the end of the pre-localization procedure. Some data concerning the fragment composition and the memory requirements follow. ELMO energy per cycle is reported together with the maximum component of the gradient and its RMSD. At last, final (non-orthonormalized) ELMOs are written, but it must be noted that they are different from the ELMOs afterwards reported in the GAMESS-UK format, which have components also on the 'forbidden' AOs because of the subsequent orthonormalization. In fact, the final eigenvectors are the occupied orthonormalized ELMOs (virtual orbitals are zeroed in the printing), while EIGENVALUES ARE NOT UPDATED (since ELMOs belonging to different fragments are calculated as eigenvectors of different operators, ELMO eigenvalues have not the same meaning as HF ones).

* the following files that can be used as input in other programs:

- elmo: ELMOs (updated at each iteration). If lrestart = .true. starting orbitals are read from this file.
- elmo_orto: final orthonormalized ELMOs
- elmo_ov and elmo_ovz: the same as 'elmo', but they contain also virtual ELMOs

NOTE: it is important to stress again that the eigenvalues printed with the final eigenvectors ARE NOT UPDATED AND THEY DO NOT HAVE ANY PHYSICAL MEANING.