ADF54: general excitation promotional rules

Each data set of promotional rules provides a rule specification for all possible ground states of ions of elements. The central ADAS data base has generic samples which can be applied to arbirtrary elements. In practice the rules would be customised for a particular computer system, with customising of one element at a time. The central ADAS data base archives the rules used in the PLT power optimisation project. There is potential redundancy in such archiving, since the rules for one element may well be apposite for all elements lighter than it.

Data mnemonic:
Data root: **/home/adas/adas/adf54/**
Last update: June, 2017
Utilising subroutines: **xxdata_54.py**

Data lines:

Data lines	Format
nmax, nel	31x,1i3,5x,1i2
do i=0 to n_index-1	
do j=0 to nel-1	
index(i*nel+j), config(i*nel+j)	1i5,1a*
enddo	
index[nel*i:(nel*(i+1))-1]	16x,10i5
n_el[nel*i:(nel*(i+1))-1]	16x,10i5
no_v_shl[nel*i:(nel*(i+1))-1]	16x,10i5
max_dn_v1[nel*i:(nel*(i+1))-1]	16x,10i5
min_dn_v1[nel*i:(nel*(i+1))-1]	16x,10i5
max_dl_v1[nel*i:(nel*(i+1))-1]	16x,10i5
min_dl_v1[nel*i:(nel*(i+1))-1]	16x,10i5
max_dn_v2[nel*i:(nel*(i+1))-1]	16x,10i5
min_dn_v2[nel*i:(nel*(i+1))-1]	16x,10i5
max_dl_v2[nel*i:(nel*(i+1))-1]	16x,10i5
min_dl_v2[nel*i:(nel*(i+1))-1]	16x,10i5
prom_cl[nel*i:(nel*(i+1))-1]	16x,10i5
max_n_cl[nel*i:(nel*(i+1))-1]	16x,10i5
min_n_cl[nel*i:(nel*(i+1))-1]	16x,10i5
max_l_cl[nel*i:(nel*(i+1))-1]	16x,10i5
min_l_cl[nel*i:(nel*(i+1))-1]	16x,10i5
max_dn_cl[nel*i:(nel*(i+1))-1]	16x,10i5
min_dn_cl[nel*i:(nel*(i+1))-1]	16x,10i5
max_dl_cl[nel*i:(nel*(i+1))-1]	16x,10i5
min_dl_cl[nel*i:(nel*(i+1))-1]	16x,10i5
fill_n_v1[nel*i:(nel*(i+1))-1]	16x,10i5
fill_par[nel*i:(nel*(i+1))-1]	16x,10i5
for_tr_sel[nel*i:(nel*(i+1))-1]	16x,10i5
last_4f[nel*i:(nel*(i+1))-1]	16x,10i5
grd_cmplx[nel*i:(nel*(i+1))-1]	16x,10i5
repeat	

Variable identification:

Name	Meaning	Comment
nmax	number of ground configs.	
nel	number of ground configs per block	
n_index	number of blocks (= nmax/nel)	
index[]	ground config. index	
config[]	ground config. strings	Cowan form
n_el[]	number of electrons in ion	
no_v_shl[]	number of open valence shells	
max_dn_v1[]	maximum Δn for promotion from first valence shell	
min_dn_v1[]	minimum Δn for promotion from first valence shell	
max_dl_v1[]	maximum Δl for promotion from first valence shell	
min_dl_v1[]	minimum Δl for promotion from first valence shell	
max_dn_v2[]	maximum Δn for promotion from second valence shell	
min_dn_v2[]	minimum Δn for promotion from second valence shell	
max_dl_v2[]	maximum Δl for promotion from second valence shell	
min_dl_v2[]	minimum Δl for promotion from second valence shell	
prom_cl[]	promote from inner closed shells	
max_n_cl[]	maximum n for inner shell promotion	
min_n_cl[]	minimum n for inner shell promotion	
max_l_cl[]	maximum l for inner shell promotion	
min_l_cl[]	minimum l for inner shell promotion	
max_dn_cl[]	maximum Δn for inner shell promotion	
min_dn_cl[]	minimum Δn for inner shell promotion	
max_dl_cl[]	maximum Δl for inner shell promotion	
min_dl_cl[]	minimum Δl for inner shell promotion	
fill_n_v1[]	add all nl configurations	
fill_par[]	add parity	
for_tr_sel[]	Cowan option for radiative transitions	
last_4f[]	shift an electron valence shell to $4f$	
grd_cmplx[]	include configurations of same complex	

Sample:

```
/ADF54    /PROMOTION RULES    / 18    /10     /Optimization project
index    config
    0    1s2  2s2  2p6  3s2  3p6
    1    1s2  2s2  2p6  3s2  3p5
    2    1s2  2s2  2p6  3s2  3p4
    3    1s2  2s2  2p6  3s2  3p3
    4    1s2  2s2  2p6  3s2  3p2
    5    1s2  2s2  2p6  3s2  3p1
    6    1s2  2s2  2p6  3s2
    7    1s2  2s2  2p6  3s1
    8    1s2  2s2  2p6
    9    1s2  2s2  2p5

index             0     1     2     3     4     5     6     7     8     9
n_el             18    17    16    15    14    13    12    11    10     9
no_v_shl          1     1     1     1     1     1     1     1     1     1
max_dn_v1         1     5     5     5     5     0     5     5     5     5
min_dn_v1        -1     0     0     0     0     0     0     0     1     0
max_dl_v1         3     2     5     4     5     0     5     3     4     5
min_dl_v1        -1    -1    -1    -1    -1     0     0     0    -1    -1
max_dn_v2         0     0     0     0     0     0     0     0     0     0
min_dn_v2         0     0     0     0     0     0     0     0     0     0
max_dl_v2         0     0     0     0     0     0     0     0     0     0
min_dl_v2         0     0     0     0     0     0     0     0     0     0
prom_cl           1     1     1     1     1     1     1     1     1     1
max_n_cl          3     3     3     3     3     3     2     2     2     2
min_n_cl          3     3     3     2     3     2     2     2     2     1
max_l_cl          1     0     0     0     0     1     1     1     0     0
min_l_cl          0     0     0     0     0     0     0     0     0     0
max_dn_cl         0     5     5     4     5     3     4     5     5     5
min_dn_cl         0     0     0     0     0     0     0     0     0     0
max_dl_cl         2     3     3     2     3     2     2     2     3     2
min_dl_cl         0     0     0     0     0     0    -1    -1     0     0
fill_n_v1         1     1     1     1     0     0     1     1     0     0
fill_par          0     0     0     0     0     0     0     0     0     0
for_tr_sel        3     3     3     3     3     3     3     3     3     3
last_4f           0     0     0     0     0     0     0     0     0     0
grd_cmplx         0     0     0     0     1     1     0     0     0     0


index    config
   10    1s2  2s2  2p4
   11    1s2  2s2  2p3
   12    1s2  2s2  2p2
   13    1s2  2s2  2p1
   14    1s2  2s2
   15    1s2  2s1
   16    1s2
   17    1s1

index            10    11    12    13    14    15    16    17
n_el              8     7     6     5     4     3     2     1
no_v_shl          1     1     1     1     1     1     1     1
```

```
max_dn_v1         5     5     5     5     5     5     5     5
min_dn_v1         0     0     0     0     0     0     1     1
max_dl_v1         5     5     4     5     5     6     5     5
min_dl_v1        -1    -1    -1    -1     0     0     0     0
max_dn_v2         0     0     0     0     0     0     0     0
min_dn_v2         0     0     0     0     0     0     0     0
max_dl_v2         0     0     0     0     0     0     0     0
min_dl_v2         0     0     0     0     0     0     0     0
prom_cl           1     1     1     1     0     1     0     0
max_n_cl          2     2     2     2     1     1     0     0
min_n_cl          2     2     2     2     1     1     0     0
max_l_cl          0     0     0     0     0     0     0     0
min_l_cl          0     0     0     0     0     0     0     0
max_dn_cl         5     5     5     5     0     4     0     0
min_dn_cl         0     0     0     0     0     0     0     0
max_dl_cl         2     3     2     2     0     4     0     0
min_dl_cl         0     0     0     0     0     0     0     0
fill_n_v1         0     0     0     0     1     1     0     0
fill_par          0     0     0     0     1     1     0     0
for_tr_sel        3     3     3     3     3     3     3     3
last_4f           0     0     0     0     0     0     0     0
grd_cmplx         0     0     0     0     1     0     0     0

C------------------------------------------------------------------------------
C
C An iso-nuclear set of promotion rules for a pre-determined set of
C metastable configurations. Further optimisation is performed to join
C configuration sets derived from promotions of multiple metastables
C and to adhere to a prescribed set of limitations. This final configuration
C set is used to produce the isonuclear set of adf04 data stored in:
C    ...adas/adf04/copssh#18/ssh41_<rule_set_size>_<resolution>#<ion>.dat
C
C The corresponding optimized adf11 PLT coefficients are stored in:
C    ...adas/adf11/plt41/plt41_ar.dat
C and described in:
C
C Henderson et al, PPCF, v59 (2017) 055010
C    DOI: 10.1088/1361-6587/aa6273.
C
C
C CODE     : ADAS808
C PRODUCER : Stuart Henderson
C DATE     : 06/07/17
C
C------------------------------------------------------------------------------
```