# ANTOINE MANUAL

This manual is organized as follows :

I) Generalities.
II) Files.
III) Interactions.
IV) Generalities about the data.
V) List of options.
VI) Specific data following the options.

## I. Generalities

The code is written in FORTRAN NORM 77 .

The main program (aaa9.f) defines some parameters . The parameter TOT defines the length of the array F. In F(*) are written all the most important arrays needed in the calculation. Notice that even if F is defined integer, It can contain double precision quantities .

The most imortant limit of this code comes from the size of this vector. It must be able, at least, to contain 2 Lanczos vectors + some working length (see option 2) . During the calculation the code tests often if the size is sufficient (if not, it stops with a clear message). But the tests are not always possible... so sometimes it can break ... Before to suspect an error in the code, increase this parameter.

The main program fixes also the output file "WRITE (IOUT,*) ...." and some quantities ( LNN,LW2, ...) . The code checks them and send a message if it is necessary to increase them .

Some dimensions are however fixed in some commons. This produces some limitations :

1. j maximum of a shell 15/2
2. number maximum of shells (p or n) 21
3. number maximum of nljm states (p or n) 112
4. maximum of jumps in each space (see defintion later) 10

These limitations can be changed (this needs however some knowledge of the code)

Some dimensions are arbitrary fixed. During the execution of the job, they are (in principle !) tested and a clear message appears if they are too small.

## II. Files

All the precalculations, which are done separetely in the two spaces are stored in a directory called "mat".

The Lanczos vectors are stored in a directory called "vec". This allows to restart a Lanczos calculation (see later).

The Hamiltonian and the vectors files are defined with a numero which is read in the data and corresponds to a Fortan unit .

ln -sf $HOME/fp/Ti46 fort.50
ln -sf $HOME/hamil/kb3 fort.90

If a file is not used take numero 0. For example if you want only energies without eigenvectors, take 0 for this file.

Fortran Units 1 and 2 are reserved for the files in "mat" and "vec" .

At the end of an option, files are in principle rewinded. It is possible to avoid that (for example you read a vector in an option and you want to write an other one behind in the next option) For that take a negative value for the numero of the file (absolute value will be taken in the calculation)

## III Interactions

```
USD INT.                    A)
```

```
1 3 203 205 1001                B)
  1.64658  -3.94780  -3.16354  C)
   1    8    8    0.300000    0.000000   D)
0 1 203 203 203 203 0 3                   E)
  0.00000  -1.41510   0.00000  -2.88420  F)
 -2.18450   0.00000  -0.06650   0.00000
0 1 205 203 203 203 1 3
  0.56470   0.00000   2.03370
  0.00000  -0.61490   0.00000
.................................
0 1 1001 205 1001 203 2 2
  2.06640
 -1.94100
.........................
0 1 1001 1001 1001 1001 0 1
  0.00000  -3.26280
 -2.12460   0.00000
```

A) TEXT (character*20)

This is the name of the interaction. This will be written with eigenvectors of the diagonalization. (after some months it can be useful to remember what interaction has been used !)

B) QMAX, SH, ( NLJ( k ), k= 1, SH)
QMAX=1 individual energy identical for a proton or neutron shell
QMAX=2 individual energy different for a proton or neutron shell
SH= number of shells in the valence space
NLJ= definition of the shell nlj with NLJ= 1000*n + 100*l+2*j

n begins at 0 . Examples 307 = 0f7/2 1103 = 1p3/2 . 511 = 0h11/2 .

This notation for the shells will be the same in all the data.

C) Loop on q=1,QMAX
(ZEPSI( k,q), k= 1, SH)

ZEPSI=individual energies

D) IDENS, (CORE(k),k=1,2), ZMASS

density dependance IDENS=0 no dependance with the mass ; =1 dependance fot only TBME ; =2 dependance for TBME and individual energies.
CORE = number of particles (space 1 and 2 ) in the core.
ZMASS : dependance with mass (see subroutine HDENS in hamil.f)

E) TMIN, TMAX, (NLJ( k ), k= 1, 4), JMIN, JMAX

F) loop t= TMIN, TMAX
(VMAT( j, t), j= JMIN, JMAX)

t=isospin. For standart Hamiltonian take TMIN=0 TMAX=1
If V(pp) ≠ V(nn) take TMAX=2 V(pn,T=1)=V(nn)=V(t=1) and V(pp)=V(t=2)
If V(pp) ≠ V(nn) ≠ V(pn,T=1) take TMAX=3 V(pn,T=1)=V(t=1); V(nn)=V(t=2) and V(pp)=V(t=3)
(it is always possible to have TMIN ≠ 0).

If the spaces for protons and neutrons are different,we can have matrix elements defined in p-n formalism.In that case we will have

TMIN=1 for V(pp) and V(nn)
TMIN=-1 for V(pn) (loop on t=iabs(TMIN),TMAX)

And in these cases we read only (X( j ), j= JMIN, JMAX)

important note:

The order of shells has no importance. The code checks that all the individual energies are defined (if not, it stops with a message), but does not check if some matrix elements are missing , they will be put at a value 0.0 without a message . However the total number of readings E) is printed .

## GENERALITIES about DATA

We give here the fundamendal readings which occur often in the different options. Examples for each option with explanation about their specific data are given later.

All the data (except character type data) are read with a free format. All the Fortran variables beginning with a letter :

A-U are integers

V-Z are double precision.

First reading is the same for ALL the options.

```
      READ (5,*,END=99) CAS, KTEXT, IPRI
      IF(KTEXT.NE.0) READ (5,100,END=99) TEXT
100   FORMAT(A20)
```

CAS= numero of the option (see list).
If CAS=0 or has not been defined, it stops.

TEXT is a variable of character-type which will be written with what is created. It is the "name" of the vectors or the Hamiltonians.

IPRI=0 standard, IPRI=1 more intermediate printings .

After this reading, except the options concerning manipulations of files or interactions ( CAS 6#6 30) which have specific data, all the other have the following reading :

```
      IF(KF.GT.0)
     *READ(5,*) (FIL( k ), k= 1,KF),(NLEC( k ), k= 1,KF), NCAL
```

KF, defined with a data in the subroutine achoi.f , is the number of files of vectors that you can have to manipulate during the option. It can be

1. KF=0 we calculate dimensions ...
2. KF=1 we read OR write vectors (calculation of transitions ...).
3. KF=2 we write AND read vectors (usual Lanczos calculations ...) .

FIL( k ) are the numero of the file (can be =0 if not used)

If KF=2 :

FIL( **1** ) is always the file on which we **write** vectors.

FIL( **2** ) is always the file on which we **read** vectors.

Remember that at the end of the option, the files are rewinded, except if the numero of the file has been defined negative.

NLEC( k ) (initialisation of the files) is the number of vectors read at the beginning of the task.

If NLEC( k ) is larger than the number of stored states, a message is given, a backspace is done and the program continues.

NCAL is the number of calculations that we want to do. For options in which we do something with vectors (transitions, change Tz, ...), if NCAL=0, all the vectors on the file will be "treated".

The 3 next read lines define the valence space

```
      DO 1 I=1,2
      READ(5,*) A(I),SH,(NLJ(K,I),K=1,SHH),(CLAS(K,I),K=1,SH),SAUT(i)
1     CONTINUE
      READ(5,*) MPRO,PARI,JUMP
```

The 2 first lines define each subspace, I=1 and 2 for protons and neutrons (or opposite). Especially for large matrices, **take the first space the larger**, large in the sense of number of Slater determinants (combinatorial factor of the number of particles and the number of valence states).

the third one defines the total space.

If we need 2 different valence spaces (ex: change Tz) we must read successively the 2 valence spaces .

1) the FINAL basis

2) the INITIAl basis

A=number of particles.
SH=number of shells.
NLJ= 1000*n+100*l+2j
MPRO= 2*Jz PARI= global parity. 0 for positive, 1 for negative.

Notice that

all the quantities which can be half-integer are multiplied by 2.

**Truncations.**

At each shell is associated a number CLAS(m,i) which is used to truncate the space in configurations.
We must have always CLAS(m,i) ≤ CLAS(m+1,i).

For each configuration, defined by n(k,i) number of particles in each shell k, we calculate :

$t(i)= \sum_k n(k,i)*clas(k,i)$

We define :

tmin(i)=min(t(i)) (all the particles put in the first shells)

and keep only configurations with :

$0 \le t(i) \le SAUT(i)+tmin(i)$ in each subspace and

$0 \le t(1)+t(2) \le JUMP$ in the global space.

It means that the code takes always SAUI(i)=min(SAUT(i),JUMP).

examples in a specific subspace :

```
   6 4 307 1103 305 1101     0 0 0 0    0       a)
   6 4 307 1103 305 1101     0 1 1 1   10       b)
   6 4 307 1103 305 1101     0 1 1 1    2       c)
  10 4 307 1103 305 1101     0 1 1 1    2       d)
```

a) no restriction
b) no restriction, possibility to analyze the wave function in term of t.
c) maximum 2 particles in the space (1103,305,1101)
d) tmin=2 maximum 4 particles in the space (1103,305,1101)

Exception : if we take JUMP < 0 , we take a fixed truncation :

t(1)+t(2) = IABS(JUMP)

## V OPTIONS.

**01. Dimensions of the space.**
**02. Number of terms in the matrix.**
**03. Creation of a pivot state.**
**04. Usual Lanczos calculation (Random Pivot).**
**05. Usual Lanczos calculation (Pivot with minimal energy).**
**06. Usual Lanczos with projected pivots.**
**07. Idem with projection on $J^2$ of the Lanczos vectors.**
**11. Calculation of shells occupations.**
**12. E(L) and M(L) transitions. Parity unchanged.**
**13. Idem with intial vectors on many files**
**14. E(L) or M(L) transitions. Parity changed.**
**21. Sum Rule of a transition Operator. Parity unchanged.**
**22. Sum Rule of a transition Operator. Parity changed.**
**23. Change Tz value.**
**24. Gamow-Teller.**
**26. Spectroscopic factors (one particle).**
**31. Change truncation.**
**32. Change the basis.**
**33. Change Jz.**
**34. Projection on $J^2$.**
**35. Projection on Isospin.**
**41. List all the vectors of files.**
**42. Copy Vectors.**
**43. Overlap between all the vectors on a file.**
**44. Overlap between vectors of 2 files.**
**45. Overlap between vectors (one by one) of many files.**
**46. Change precision of vectors.**
**47. Spectrum.**
**51. Information about interactions.**
**52. Modification of an interaction (monopole,pairing).**
**53. Creation of a H file from standard TBME.**
**54. Addition of interactions.**

## VI SPECIFIC DATA FOLLOWING THE OPTIONS .

### 01. Dimension of a valence space

```
     1 0 0
6  4 307 1103 305 1101   0 1 1 1     8
10 4 307 1103 305 1101   0 1 1 1     8
  0 0 2
```

Calculation of the dimensions in the space and that for all the Jz values. It gives also the dimension in the J space and the effective dimension of the matrix for M=0 when time reversal symetry is taken in account . It allows to see until what value of J it is better to take Jz=0 instead of Jz=J.

### 02. Number of non-zero of the matrix

```
     2 0 0
10 4 307 1103 305 1101      0 1 1 1     8
19 5 307 1103 305 1101 409  0 8 8 8 9   8
1 1 6
```

Exactly the same data as for the calculation of dimensions.

The number of non-zero in the matrix gives an order of magnitude for the computing time. This option gives also the number of equivalent integer numbers to store on disk.

In this example we work in a valence space defined with a $^{48}$Ca core but with 8 neutrons blocked in the $f_{7/2}$shell.

### 03. Creation of a pivot state.

```
     3 0 0
 50 0 2              a)
6 4 307 1103 305 1101      0 1 1 1     8
4 4 307 1103 305 1101      0 1 1 1     8
0 0 2
 0 4                b)
12457  1            c)
```

a) the pivot states will be written on file 50 and we calculate 2 states .

b) (JVAL(k),k=1,NKEEP)

c) KPVT, ISO
The created pivots are random . KPVT is an integer which initiates the random numbers.
ISO : defines Tz
IS0=1 pivot states are random but built with T=Tz.
ISO=0 no isospin projection. if MPRO $\neq$ 0 RVT=0

Supposing A(1) ≥ A(2), States of the basis for which all the occupied states in space 2 are occupied in space 1 have T=Tz. These states allow to get a pivot state with good isospin . (It is not possible to get it for a nucleus N=Z=odd with Jz=0) .

**04. Usual Lanczos calculation (Random Pivot).**

```
     4 0 0
 50 0 2                                a)
6 4 307 1103 305 1101       0 1 1 1     8
4 4 307 1103 305 1101       0 1 1 1     8
0 0 2
  0   4              b)
  1   2                b)
  30 0.0005 0          c)
```

a) the resulting states from the calculation will be written on file 50
2 states are projected.

b) (JVAL(k),k=1,NKEEP)
We will calculate and write on file 50 one J=0 state and 2 J=2 states.

b) (NCON(k),k=1,NCAL) number of converged states that we need for each matrix.

c) NLOOP,ZFIT,ORTH

NLOOP= number maximum of Lanczos iterations

ZFIT : test of convergence.
each 3 iterations the tridiagonal Lanczos matrix is diagonalized. If all the states that we want to converge have an increase of their energy smaller than ZFIT, the Lanczos procedure is stopped.

ORTH usual calculation take 0 (see later)

**05. Idem but Pivot with minimal energy**

**06. Usual Lanczos with projected pivots.**

```
     6 0 0
 50 51 0 0 2                              a)
6 4 307 1103 305 1101       0 1 1 1     8
4 4 307 1103 305 1101       0 1 1 1     8
0 0 2
  1   2
  1 30 0.0005 0 0       b)
```

a) the initial vectors are read from file 51 and the resulting vectors are stored on file 50

b) KPVT, NLOOP, ZFIT, ORTH, NLANZ

KPVT : determination of the pivot

KPVT > 0 pivot : linear combination of the KPVT vectors on the initial file.
KPVT = 0 pivot : linear combination of the NCON(k) vectors on the initial file.

In the starting calculation we have one vector for each J value , we must take KPVT=1 . If we change the truncation, the convergence is better in taking a linear combination of the NCON (k) states precedently calculated we will take KPVT=0

NLOOP= number maximum of Lanczos iterations

ZFIT : test of convergence.
each 3 iterations the tridiagonal Lanczos matrix is diagonalized. If all the states that we want to converge have an increase of their energy smaller than ZFIT, the Lanczos procedure is stopped.

ORTH , NLANZ usual calculation take 0 (see later)

**Special possibilities with Lanczos.**

STRENGTH FUNCTIONS

```
  6   0  0
 50 51 0 0 1
6 4 307 1103 305 1101       0 1 1 1     8
4 4 307 1103 305 1101       0 1 1 1     8
0 0 2
    90 0
    2
  1  100 -0.0005  0  0         a)
    20  1                      b)
```

a) In a strength function we calculate Lanczos vectors until the end without convergence criteria . For that take ZFIT < 0.0

b) STEP, NKEEP
In standart Lanczos we have STEP=3 . If you do many iterations take a larger value . The NKEEP states, higher in enrgy than the NCON first states, and having a maximum overlap with th pivot state are kept in FIL(1). At the end we kept always NCON+NKEEP vectors.

RESTART A LANCZOS CALCULATION.

```
  6   0  0
 50 51  0 1 2                 b)
 ..............
```

```
  1   30  0.0005  0 100       a)
```

a) For different reasons you can want to restart a Lanczos calculation . Give a value at NLANZ . The code checks the number NSTOR of vectors that he has and restart Lanczos with NLANZ = min ( NLANZ, NSTOR)

b) the pivot vector is not read. Think to shift it . In the next calculations NLANZ=0

ORTHOGONALIZATION TO THE PRECEDENT CALCULATED STATES

```
  5   0  0
 ..............
  1   30  0.0005  1   0       a)
```

a) ORTH=1 :
we orthogonalize each new Lanczos vector to all the vectors stored on FIL(1) . It can be an alternative at option 6. It can be used to calculate a new state of a given spin , or converge succesively 2 states instead of the 2 both.

**07. Idem as previous option but each new Lanczos vector is projected on $J^2$.**

**11. Occupation of Shells**

```
 11   0  0
 50 0 0                                          a)
6 4 307 1103 305 1101      0 1 1 1     8
4 4 307 1103 305 1101      0 1 1 1     8
0 0 2
```

a) Remember that if NCAL=0 all the vectors of the file are read (true for all the following options).

Results of the options are :

Occupation of shells.
Print of occupation > 0.01 .This number can be changed (see subroutine TCONF2 in tconf.f) .
Two-Body occupations ( mean value n(i).n(j) and n(i).(n(i)-1)/2) if IPRI > 0

**12. Electromagnetic Transitions (parity unchanged).**

```
 12   0  0
 50 0 0                                      a)
```

```
6 4 307 1103 305 1101      0 1 1 1     8
4 4 307 1103 305 1101      0 1 1 1     8
0 0 6
 2 0 20 20  0.5 1.5                                  b)

12 0 1
60 0 3
 10  4 307 1103 1101 305      0 1 1 1         8
 19  5 307 1103 1101 305 409  0 8 8 8 9       8
 1 1  2
 1 0  20 20   5.586   1.000    -3.826   0.000       b)
```

a) no initialization of the file (NVEC=0) . The code loops on the states calculating the sum rule , rewind the file and overlap with the read states. States not belonging to the defined valence space are skipped.

b) LB,DP,(CORE(k),k=1,2),(ZCOEF(K,PN),K=1,1+MOD(LB+DP,2)),PN=1,2)

LB=lambda
DP=0 no parity change (in this option)
CORE : number of particles in the core for space 1 and 2 . In the present case it is used only to calculate the oscillator length.
ZCOEF=
electric transition : effective charge 1 and 2
magnetic transition : sigma(1), orbital (1), sigma(2) , orbital (2)

**13. Electromagnetic Transitions (parity unchanged). Vectors on many files.**

Same as previous option but due to the size of matrices, vectors are stored on several files.

```
 13   0  0
2 50 51 0 0 2 3                                          a)
6 4 307 1103 305 1101      0 1 1 1     8
4 4 307 1103 305 1101      0 1 1 1     8
0 0 6
 2 0 20 20  0.5 1.5
```

a) the states are stored on the two files 50 and 51.
Each file contains here two and three states to be taken into account.

**14. Electromagnetic Transitions (parity changed).**

Same as previous option but a final basis (first) and an initial basis are read.
Note that this option has not been tested extensively and needs cautious use !

```
 14    0  0
 50 0 0                                                  a)
6 4 307 1103 305 1101       0 1 1 1      8
4 4 307 1103 305 1101       0 1 1 1      8
0 1 6
6 4 307 1103 305 1101       0 1 1 1      8
4 4 307 1103 305 1101       0 1 1 1      8
0 0 6
 2 0 20 20   0.5 1.5                                     b)

14 0 1
60 0 3
 10   4 307 1103 1101 305       0 1 1 1          8
 19   5 307 1103 1101 305 409   0 8 8 8 9        8
 1 0   2
 10   4 307 1103 1101 305       0 1 1 1          8
 19   5 307 1103 1101 305 409   0 8 8 8 9        8
 1 1   2
 1 0   20 20    5.586    1.000     -3.826    0.000        b)
```

a) no initialization of the file (NVEC=0) . The code loops on the states calculating the sum rule , rewind the file and overlap with the read states.
States not belonging to the defined valence space are skipped.

b) LB,DP,(CORE(k),k=1,2),(ZCOEF(K,PN),K=1,1+MOD(LB+DP,2)),PN=1,2)

LB=lambda
DP=0 no parity change (in this option)
CORE : number of particles in the core for space 1 and 2 . In the present case it is used only to calculate the oscillator length.
ZCOEF=
electric transition : effective charge 1 and 2
magnetic transition : sigma(1), orbital (1), sigma(2) , orbital (2)

**21. Sum Rule of an Electromagnetic transition operator (Parity unchanged) .**

```
 21    0  0
 81 50 0 0   0                                    a)
6 4 307 1103 305 1101       0 1 1 1      8
4 4 307 1103 305 1101       0 1 1 1      8
0 0 6
 2 0 20 20   0.5 1.5
```

a) Same data as the calculation of transitions , except that we have a final and initial files. The final states are normalized but not projected.

**22. Sum Rule of an Electromagnetic transition operator (Parity changed).**

```
 22    0  0
 81 50 0 0   0                                       a)
6 4 307 1103 305 1101       0 1 1 1      8
4 4 307 1103 305 1101       0 1 1 1      8
0 1 6
6 4 307 1103 305 1101       0 1 1 1      8
4 4 307 1103 305 1101       0 1 1 1      8
0 0 6
 2 0 20 20   0.5 1.5
```

a) Same data as the calculation of transitions , except that we have a final and initial files. The final states are normalized but not projected.

**23. Change Tz value.**

```
23 0 0
63 50 0 0 0
 3    4 307 1103 1101 305       0 0 0 0          0       a)
 3    4 307 1103 1101 305       0 0 0 0          0
 0 0   0
 4    4 307 1103 1101 305       0 0 0 0          0       b)
 2    4 307 1103 1101 305       0 0 0 0          0
 0 0   0
```

we define the final basis a) and the initial b)

**24. Gamow-Teller.**

```
24 0 1
0  50 0 0 1
  5   4 307 1103 1101 305       0 1 1 1          8       a)
  1   4 307 1103 1101 305       0 1 1 1          8
 0 0   3
  4   4 307 1103 1101 305       0 1 1 1          8       b)
  2   4 307 1103 1101 305       0 1 1 1          8
 0 0   2

24 0 1
51 50 0 1 1
  3   4 307 1103 1101 305       0 1 1 1          8       a)
  3   4 307 1103 1101 305       0 1 1 1          8
 0 0   3
  4   4 307 1103 1101 305       0 1 1 1          8       b)
  2   4 307 1103 1101 305       0 1 1 1          8
 0 0   2
```

```
  3   4 307 1103 1101 305       0 1 1 1          8       c)
  3   4 307 1103 1101 305       0 1 1 1          8
 0 0   3
 2 2 6                                                    d)
```

we define the final basis a) and the initial b) If the initial state is J=0 , the final state is J=1 and it is written on the file FIL(1) (if FIL(1) ≠ 0)

If (second example) the initial state is not J=0 we must project this "sum rule" state on $J^2$. To do it we have the readings c and d like in the option 4.

Note :
Possiblity to have JUMP final ≠ initial (in the present example to keep Ikeda sum rule)
In the second example we have Jz=0 excluding to have a final state J=2 since J initial=2 . To have it Jz value must be changed .(see next option)

### 26. Spectroscopic Factors (one particle).

```
26 0 0
51   50 0 0 1                                            a)
  5   4 307 1103 1101 305       0 1 1 1          8       b)
  1   4 307 1103 1101 305       0 1 1 1          8
 0 0   2
  4   4 307 1103 1101 305       0 1 1 1          8       c)
  2   4 307 1103 1101 305       0 1 1 1          8
 0 0   2
```

States corresponding to basis b) are previously calculated and stored on file 51,
states corresponding to basis c) are previosly calculated and stored on file 50.

Spectroscopic Factors between states on file 51 and states on file 50 are then calculated.

### 31. Change truncation.

```
 31    0   0
 51 50 0 0 3                                            a)
6 4 307 1103 305 1101       0 1 1 1      8
4 4 307 1103 305 1101       0 1 1 1      8
0 0 6                                                   b)
```

a) FLRES FLINI INI1 INI2 NCAL.
NCAL number of vectors to treat. NCAL=0, take ALL vectors on FILINI.
b) We read the final truncation . Read vectors (on FIL(2)) must have the same type of truncation but smaller or equal.

### 32. Change The Basis .

```
32 0 1                                                  a)
63 50 0 0 1
  4   4 307 1103 1101 305       0 1 1 1          8
  2   4 307 1103 1101 305       0 1 1 1          8
 0 0 -2
  4   4 307 1103 1101 305       0 0 0 0          8
  2   4 307 1103 1101 305       0 0 0 0          8
 0 0   0

32 0 1                                                  b)
63 50 0 0 1
  4   4 307 1103 1101 305       0 0 0 0          8
  4   4 307 1103 1101 305       0 0 0 0          8
 0 0   0
  4   4 307 1103 1101 305       0 0 1 1          8
  4   4 307 1103 1101 305       0 0 1 1          8
 0 0   0
```

the truncation can be totally different . We can increase the space, reduce it or mix the 2 things . a) In this example we extract from a calculation without truncation, the component with a fixed truncation. The only condition is that the shells must be identical.
b) If you want to add shells in the valence space you can "play" with the truncations . In this example we write vectors from the restricted (f7/2,p3/2) space in the full space.

### 33. Change Jz .

```
33 0 1
53 51 0 4 1                                              a)
  4   4 307 1103 1101 305       0 1 2 2          8
  2   4 307 1103 1101 305       0 1 2 2          8
 8 0   4
```

The change of Jz is now for any unit difference (with the initial value).
Only the final basis (for the new Jz) is needed.

a) Think to initialize correctly the initial file. ( Changes only states with J ≥ Jz final).

**34. Projection on J .**

```
34 0 1
53 51 0 0 1
  4  4 307 1103 1101 305       0 1 1 1          8
  2  4 307 1103 1101 305       0 1 1 1          8
 8 0  4
 2 0 4                                      a)
```

a) NCAL, (JVAL(K),K=1,NCAL)

Here we project and keep J=0,2 values

**35. Projection on $T^2$ .**

```
35 0 1
53 51 0 0 1
  4  4 307 1103 1101 305       0 1 1 1          8
  2  4 307 1103 1101 305       0 1 1 1          8
 8 0  4
 2 0 1                                      a)
```

a) NCAL, (TVAL(K),K=1,NCAL)

Here we project and keep T-Tz=0,1 values

**41. List all the vectors of files**

```
   41 0 0
      3  38 40 -42                                a)
```

a) NFIL, (FIL( k ), k= 1, NFIL)

We read and list all the vectors of files. It can be used to go at the end of a file.

**42. Copy vectors from a file to an another one.**

```
   42 0 0
    -10  38    0 1    2                          a)
   42 0 0
     10  40    0 1    0                          a)
```

a) (FIL( k ), k= 1, 2), (NLEC( k ), k= 1, 2), NCOP

NCOP : number of files copied from FIL(2) to FIL(1)
Remember that, like in the first example, a negative sign for a file means that it is not rewinded at the end of the option. If NCOP=0 all the states are copied.

**43. Overlap between all the read vectors of a file**

```
   43 0 0
  40  0  0
```

Overlap between all the vectors of the file. Calculation is done only between vectors of same dimension and same time reversal symmetry but possible different J value.

**44. Overlap between vectors of 2 files**

```
   44 0 0
     41  40  0 0    4 5
```

a) (FIL( k ), k= 1, 2), (NLEC( k ), k= 1, 2), (NVEC( k ), k= 1, 2)

Overlap calculated only between vectors of different files. Same restriction precedent option.

**45. Overlap between vectors (one by one) of many files**

```
   45 0 0
 4 10 11 12 13   0 0 0 0     10                  a)
```

a) NFIL, (FIL(k), k=1,NFIL), (NLEC(k), k=1,NFIL), NVEC

NVEC times we read one state on each file and take the overlaps. Example of use : comparison of yrast bands got with different interactions.

### 46. Change precision of the vectors

```
 46 0 0
 50 51 0 0 10                                   a)
```

a) FLRES,FLLEC,INIRES,INILEC,NVEC

NVEC=0: All the states read on all the initial file are taken in account.
The final states are then to be used in a simple precision calculation.
The simple precision options are simply obtained from standard ones by adding 100 to the number of the option (e.g. 104 for diagonalization)

### 47. Spectrum

```
 47 0 0
   1 12 0 100                                   a)
```

a) NFIL, (FIL(k), k=1,NFIL), (NLEC(k), k=1,NFIL), (NVEC(k), k=1,NFIL)

All the states read on all the files are taken in account to calculate the spectrum.

### 51. Information about interactions

```
 51  0 0
  4 90 91 92 93                                 a)
0 1 4 307 1103 305 1101                         b)
```

a) NFIL, (FIL( k ), k= 1, NFIL)
b) TMINI,TMAXI,SH,(NLJ(i),I=1,SH)

TMINI,TMAXI : isospin min et max.

We calculate "norm" and "overlap" of interactions:

$N^2(q)= \sum_{mnpq,jt} (2j+1)(2t+1)V(mnpq,jt,q)*V(mnpq,jt,q)/ \sum_{mnpq,jt} (2j+1)(2t+1)$

Summation is done on all the jt 2-body matrix elements allowed by the Pauli principle.

$O(q1,q2)=N^2(q1,q2)/(N(q1)*N(q2))$

We compare the monopoles.

### 52. Modification of interactions

```
 52 1 0
KB3 with modif
   99 90   1.0 1.0   0.0 0.0                   a)
      0.0  0.0  0.0  0.0                       b)
       307 307  0.2                            c)
       307 305  0.1                            c)
         0  307  0.2                           c)
    0  -1  307  0.                             d)
    0  307 307  0.1                            d)
    1  307 307  0.1                            d)
    0   0  307  0.2                            d)
```

a) (FIL( k ), k=1,2), ZPDIA, ZPNDIA
FIL(1)= new H FIL(2)= old H
ZPDIA =global diagonal pairing added
ZPNDIA=global non-diagonal pairing added

b) modification of individual energies. Loop on QMAX (see chapt. II)
(Z( k ), k=1,SH)

c) (PAIR( i, k ), i=1,2), VPAIR( k )
PAIR( i,k )= nlj , VPAIR : modification of this specific pairing. Terms are read until PAIR(1,k) =0 (in our example we have 2 pairing modifications)

d) ISO( k ), (MONO( i, k ), i=1,2), VMONO( k )

e) Modification of the monopole. Same principle as for the Pairing. ( ISO (k) is the isospin). It is possible to initiate ALL the monopoles at 0 in taking in the first modification (like in our example)
MONO(1,1)=-1

### 53. Change the format of 2-body matrix elemnts

```
  53 1 0
Bonn A for Tin region
  95 1 5  1205 407 1203 2001 511                a)
    0.0 0.2  2.4 2.5 3.0                        b)
    1 50 50  0.333333  0.0                      c)
      1 1 1 1    0 1      0.5                   d)
      .........
      5 5 5 5   11 0      0.2
      0 1 1 1    0 1      0.5
```

c) I1,I2,I3,I4,J,T,XX

Rewriting the 2-body matrix elements in the format used in ANTOINE.

a) FLMAT,QMAX,SH,(NLJ(K),K=1,SH)
FLMAT= file of storage of the interaction.
QMAX=1 no coulomb; QMAX=2 coulomb

SH, (NLJ( k ), k= 1, SH) valence space.

b) looop q=1,QMAX
(ZEPSI( k ), k= 1, SH) individual energies

d) list of TBME
I1, I2, I3, I4, J, T, x for QMAX=1 or T=0 I1, I2, I3, I4, J, T, x(k),k=1,2) for QMAX=2 and T=1

ik=numero of the shells $(a^+(i1)\ a^+(i2))^{jt}\ (a(i3)a(i4))^{jt}$ This loop continues until the end or until we find i1=0 .

Check that i1 in the list is nlj(i1) ...

Some lists are written with T,J or 2J,2T or .... instead of J,T . It is easy to change the reading in the code (see zforma.f).

**54. Addition of Interactions**

```
   54 0 0
2 99 97 98            a)
7 205 203 1001  307 1103 305 1101    b)
1.0 10.0                             c)
```

a) NFIL, FLRES,(FIL(k), k=1,NFIL)

b) SH,(NLJ(K),K=1,SH)

c) (ZCOEFF(K),K=1,NFIL)

A final set of matrix elements is created as a linear combination of the initial interactions with the coefficients ZCOEFF.
Here we create an interaction in file 99 as a combination of interaction 97 (with coeff. 1.0) and 98 (with coeff. 10).
Usefull to add a center of mass hamiltonian for example.
Note that the coefficients apply also on the single particle energies.

---