**Description of utility-programs for scattering problem in the (D)BSR complex**

(folder **UTILS/SCT\_LS** )

The BSR\_HD program creates the standard H.DAT file, which can be used for the scattering calculations either with SRGF or FARM programs. These programs output results for scattering strengths (Ω), T-matrix and K-matrix elements in files with some specific formats. The BSR complex uses its own format to record these data, namely, **zarm.om**, **zarm.tma** and **zarm.kma** files, respectively. These files are supposed to accumulate the data from several runs of STFG or FARM programs (or other programs if any) using the utility programs **add\_stgf** or **add\_farm**.

The **zarm.om** is just a list of records of collision strengths for given energy. Each record has format:

1. ek, ns

2. ((OM(i,j),j=1,i),i=1,io) z = 0 (neutral case, electron-atom scattering)

2. ((OM(i,j),j=1,i-1),i=2,io) z > 0 (electron-ion scattering case)

ek - electron energy in Rydbergs (*k*2-value)

ns - number of elements recorded, io\*(io+1)/2 or io\*(io-1)/2 for ions.

io - number of open target states.

Because omega matrix is symmetric, it is recorded in half, row by row. For ion case, the diagonal elements are dropped. Number of energies are not recorded, so the user should first scam the file to determine the number of energies recorded.

The **zarm.kma** (**zarm.tma**) is a list of records of K-matrix (T-matrix) for given energy. Each record has format:

1. ek, nopen, ntr, ilsp

2. ((KMAT(i,j),i=1,j),j=1,nopen) or

2. ((TMATr(i,j),TMATi(i,j),i=1,j),j=1,nopen)

The matrixes are symmetric, they are recorded in half, row by row. T-matrix contains real and imaginary parts. Number of records are not recorded, so the user should first scam the file to determine the number of energies recorded.

ek - electron energy in Rydbergs (*k*2-value)

nopen - number of open scattering channels.

ntr - number of elements recorded, nopen\*(nopen+1)/2.

ilsp - index of partial wave according to the target file

The **zarm.om\_par** is a list of records of collision strengths for given energy and given partial wave. Each record has format:

1. ek,ilsp,ns,io

2. ((OM(i,j),j=1,i),i=1,io) z = 0 (neutral case, electron-atom scattering)

2. ((OM(i,j),j=1,i-1),i=2,io) z > 0 (electron-ion scattering case)

Because omega matrix is symmetric, it is recorded in half, row by row. For ion case, the diagonal element are dropped. Number of records are not recorded, so the user should first scam the file to determine the number of energies recorded.

ek - electron energy in Rydbergs (*k*2-value)

ns - number of elements recorded, io\*(io+1)/2 or io\*(io-1)/2 for ions.

io - number of open target states.

ilsp - index of partial wave

We will call the above recording format as **mode 'a'**. For the large-scale calculations the corresponding matrixes can be very large and take too much space (especially, for electron-ion calculations, where we need many energy points to recover the near-threshold resonance structure).

In this case, we use '**mode b**', where we introduce two new parameters:

np - number of physical states

ni - number of "ionizing" states

In this mode, we record only the matrix elements which describe the all transitions between physical states, np, and all matrix elements, related to the "ionizing" states, for which we are supposed to calculate the ionization cross sections (as sum of excitation cross section to the continuum pseudo-states). The scheme of recording is illustrated on the following figures. Arrows indicate as the matrix elements are placed in the output record. For the electron-ion scattering the omega diagonal elements are not recoded. The schemes are given for the case when number of open target states is greater then number of physical states. For T-matrix recording is used following notation:

nch - number of channels for the given partial wave.

kopen - number of open channels for the physical states

nopen - number of all open channel

nj - number of open channels for the ionizing states

K-matrix, if needed is recorded in full (as in mode 'a').

The files in mode 'a' cam be converted in mode 'b' using the utilities **tma\_tmb**, **oma\_omb**, **oma\_omb\_par**

 

**ADD\_STGF**

|  |  |
| --- | --- |
| Description: | accumulates results after STGF (PSTGF) runs |
| Input files: | OMEGA, KMAT.DAT, TMAT.DAT, target |
| Output files: | zarm.om, zarm.kma, zarm.tma |
| Call as: | add\_stgf [klsp=..]  klsp - index of partial wave, if calculations for one partial wave only,  in this case output is recorded in zarm.om\_par file |

**ADD\_FARM**

|  |  |
| --- | --- |
| Description: | accumulates results after FARM runs |
| Input files: | farm.om, farm.kma, farm.tma, farm.pha, target |
| Output files: | zarm.om, zarm.kma, zarm.tma, zarm.pha |
| Call as: | add\_farm |

**SEC\_OM**

|  |  |
| --- | --- |
| Description: | provides cross sections in table form |
| Input files: | zarm.om + target |
| Output files: | tr\_ii\_ff.dat |
| Call as: | sec\_om ii1 ii2 ff1 ff2 [i16] |

Arguments:

ii1,ii2 - range for initial index ii

ff1,ff2 - range for final index ff

i16 - control the output:

0 - sigma in a\_o^2 (default)

1 - sigma in 10-16 cm^2

2 - sigma in 10-18 cm^2

-1 - only omega

Electron energies in tables are given relative to the ground state.

**SEC\_DCS\_JK**

|  |  |
| --- | --- |
| Description: | provides differential and angle-integrated (ordinary and momentum transfer) cross sections for given transition |
| Input files: | zarm.tma (or zarm.tmb, tmat.done), target |
| Output files: | dcs\_ii\_ff, mt\_ii\_ff |
| Call as: | sec\_dif\_JK itr1=ii itr2=ff i16=-1|0|. ifano=0|1 ek1=... ek2=... or ekk=… Glow=... Ghigh=... Gstep=... dcs=0|1 tdone=0|1 JJ\_extend=… |

All arguments are optional.

|  |  |
| --- | --- |
| itr1 [1] | index of initial state (default - 1) |
| itr2 [1] | index of final state (default - 1) |
| i16 [16] | i16 - controls the output units:  = 0 - sigma in a.u.  > 0 - sigma in 10-i16 cm2 |
| ifano [0] | = 0 - Condon-Shortly phase convention, default  = 1 - Fano phase convention |
| ek1 [0] | if > 0, restriction on minimum electrovn energy (in Ry) |
| ek2 [0] | if > 0, restriction on maximum electron energy (in Ry) |
| ekk [0] | if > 0, exact electron energy (ek1=ek2=ekk) (output in tmat.done\_inp) |
| Glow [0] | lowest scattering angle |
| Ghigh [180] | highest scattering angle |
| Gstep [1] | step fort scattering angle |
| dcs [1] | if = 0, skip the calculations of differential cross sections |
| tdone [0] | if =1, redirect input from zarm.tma to tmat.done file |
| JJ\_top [0] | if > 0, extrapolate T-matrix elements to JJ\_extend value  (input tmat.done\_inp -> output tmat.done\_out) |

The utility **SEC\_DIF\_JK** first checks zarm.tma (zarm.tmb) file and create tmat.done file with T-matrix elements, specific for the given transition. The tmat.done file has the same format as in program MJK (Grum-Grzhimailo 2003). If ekk parameter is not equal 0, the program additionally analyzes the T-matrix elements for the given energy and prepares them for extrapolation to higher J-values. To do it, the program first divided all matrix elements on subsets with the same changes of involved l- and j-values. The values in subsets are supposed to reduce as in geometric series. The corresponding coefficients are found as ratio of two highest values in the series, T(n)/T(n-1). This information is recorded in the tmat.done\_inp file and the program stops. The user may check the extrapolation coefficients (in the end of the tmat.done\_inp file) and rerun the program with jj\_top parameter. The program with extrapolate the T-matric coefficients with 2J values up two jj\_top. The resulting T-matrix elements are recorded in tmat.done\_out file and program stops. The user may check extrapolated data and copy this file to tmat.done. Then, in order to get differential cross sections, he can use SEC\_DIF\_JK with tdone=1 option, or any other program, which employ the tmat.done input. Note that for high J-values of J (> 50), the SEC\_DIF\_JJ program may take too much time due to big number of Aλ coefficients (see below). In this case, it is advised to use SEC\_DIF\_JK\_ampl program (described below), which is much faster.

**SEC\_DCS\_JK\_AMPL**

|  |  |
| --- | --- |
| Description: | provides differential and angle-integrated (ordinary and momentum transfer) cross sections for given transition |
| Input files: | zarm.tma (or zarm.tmb, tmat.done), target |
| Output files: | dcs\_ii\_ff, mt\_ii\_ff |
| Call as: | sec\_dif\_JK\_ampl itr1=ii itr2=jj i16=-1|0|. ifano=0|1 ek1=... ek2=... or ekk=… Glow=... Ghigh=... Gstep=... dcs=0|1 tdone=0|1 JJ\_extend=… |

The utility SEC\_DIF\_JK\_AMPL used the direct calculations of scattering amplitude instead of analytical approach implemented in SEC\_DIF\_JK. It has the same input argument and the same file structure. The SEC\_DIF\_JK\_AMPL turned out to be much faster then SEC\_DIF\_JK, especially for big T-matrix sets with high maximum J-values (50 and more).

**KMA\_PHASE**

|  |  |
| --- | --- |
| Description: | provides SUM-of-EIGENPHASES for selected pratial waves |
| Input files: | zarm.kma + target |
| Output files: | phahes.nnn\_mmm or phases.nnn **(phases are given in π units)** |
| Call as: | kma\_phase [ilsp1=.. ilsp2=.. E\_min=.. E\_max=.. ] |

Arguments:

ilsp1,ilsp2 - range for partial wave included into output (nnn=ilsp1 and mmm=ilsp2)

E\_min, E\_max - range for output electron energies (in Ry)

If ilsp1=ilsp2, phases.nnn contains additional output: derivatives and widths in meV (see related formulas)

Related formulas:

The eigenpahses, δi, are defined by eigenvalues of K-matrix:



In the region of the resonance, the SUM of eigenphases, δ, shows step-like rise on π value (approximately, depending on the non-resonant background). Derivatives shows the Lorentz form with maximum at resonance energy *Er*, and the resonance width is related to the inverse of the eigenphase-sum derivative at *Er* by the following expression



The Lorentz forms can be distorted close to excitation thresholds and in case of overlap resonances.

**SEC\_TOP**

|  |  |
| --- | --- |
| Description: | top-up procedure for the cross sections (geometric series is used for spin-allowed transitions) |
| Input files: | zarm.omb\_par + target |
| Output files: | zarm.omb\_top, zarm.omb (without top-up), sec\_top\_omb.log, sec\_top\_coef\_fail |
| Call as: | sec\_top [par=.. top=.. jtr1=.. jtr2=.. ek1=... ek2=... tail=.. x=..] |

All arguments are optional.

|  |  |
| --- | --- |
| par [zarm.omb.par] | file with input partial collision strengths |
| top [zarm.omb\_top] | file with output top-up collision strengths |
| ek1 [0] | if > 0, restriction on minimum electrovn energy (in Ry) |
| ek2 [0] | if > 0, restriction on maximum electron energy (in Ry) |
| tail [0.001] | tolerance for top-up procedure |
| x [0.2] | tolerance for geometric series coefficient (x-1), if smaller -> extrapolation |
| jtr1, jtr2 [0,0] | debug parameters: if ≠ 0, considered only one transition jtr1-> jtr2, with more information |

If there is several zarm.omb\_par files, the user should run sec\_top in raw for all cases with increasing energies.

The geometric series coefficient is defined as x = om(*L*max-1)/om(*L*max) and correction is om(*L*max)/(x-1),

It is checked with energy estimated x = (ek – E(target initial) ) / (ek – E(target final) .

If x < x\_input , the extrapolation procedure is used.