Description of utility-programs for scattering problem in the 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 element are dropped. Number of records 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**

Omega file recording

$$\begin{array}{r} \\ \\ \\ \\ np \\ \\ \\ io \\ \\ \mathrm{ntarg} \end{array} \left(\begin{array}{cccccccc} \to &&&&&&& \\ \to & \to &&&&&& \\ \to & \to & \to &&&&& \\ \to & \to & \to & \to &&&& \\ \to & \to & \to & \to & \to &&& \\ \to & \to &&&&&& \\ \to & \to &&&&&& \\ \to & \to &&&&&& \\ &&&&&&& \\ &&&&&&& \end{array}\right)$$

$$ni$$

T - matrix file recording

$$\begin{array}{r} \\ \\ \\ \\ kopen \\ \\ \\ nopen \\ \\ nch \end{array} \left(\begin{array}{cccccccc} \to &&&&&&& \\ \to & \to &&&&&& \\ \to & \to & \to &&&&& \\ \to & \to & \to & \to &&&& \\ \to & \to & \to & \to & \to &&& \\ \to & \to &&&&&& \\ \to & \to &&&&&& \\ \to & \to &&&&&& \\ &&&&&&& \\ &&&&&&& \end{array}\right)$$

$$nj$$

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} cm^2
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 A_λ 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

<table>
<tr><td>Description:</td><td>provides differential and angle-integrated (ordinary and momentum transfer) cross sections for given transition</td></tr>
<tr><td>Input files:</td><td>zarm.tma (or zarm.tmb, tmat.done), target</td></tr>
<tr><td>Output files:</td><td>dcs_ii_ff, mt_ii_ff</td></tr>
<tr><td>Call as:</td><td>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=…</td></tr>
</table>

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, $\delta_{i,}$ are defined by eigenvalues of K-matrix:

$$\delta_i = \tan(K_{ii}^{diag})$$

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 E_r, and the resonance width is related to the inverse of the eigenphase-sum derivative at E_r by the following expression

$$\Gamma = 2/(d\delta/dE)_{E=E_r}$$

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.

SEC_TOP_CBE

Description:	top-up procedure for the cross sections (CBE method for dipole transitions and geometric series for others)
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_CBE [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
method [1]	if =2, the CBE approximation is used for dipole transitions

Coulomb-Bethe approximation and partial-wave summation

In the Coulomb-Bethe (CBE) approximation, the incident electron is supposed not to penetrate the target at all. Therefore, one can neglect the short-range interaction and replace the whole electrostatic-interaction (radial part) by its asymptotic form $r_1^{\lambda}/r_2^{\lambda+1}$ which is correct at $r_2 > r_1$. Then

$$K_{if,\lambda}^{CBe} = \int_0^{\infty} F_{l_i}(r_2) r_2^{\lambda-1} F_{l_f}(r_2) dr_2 \int_0^{\infty} P_{l_0}(r_1) r_1^{\lambda+1} P_{l_1}(r_1) dr_1 \qquad (1)$$

Let us restrict our consideration to the case of the dipole transitions with $\lambda = 1$. In this case the integration over the angular variables for the collision strength can be carried out analytically (Burke and Seaton 1986), and for the partial collision strength we obtain the following expression

$$\Omega(l_i, l_f) = \Omega(\Gamma_i S_i L_i l_i, \Gamma_f S_f L_f l_f) = \frac{16}{3} S(\Gamma_i S_i L_i, \Gamma_f S_f L_f) l_> \left| \int F_{k_i l_i} r^{-2} F_{k_f l_f} dr \right|^2 \qquad (2)$$

where

$$S(\Gamma_i S_i L_i, \Gamma_f S_f L_f) = \sum_{M_s, M_L, \mu} \left| \left(\Gamma_i S_i L_i M_{S_i} M_{L_i} \middle| \sum_{n=1}^{N} C_{1\mu}(\hat{r}_n) r_n \middle| \Gamma_f S_f L_f M_{S_f} M_{L_f} \right) \right|^2 \qquad (3)$$

is the usual definition of the line strength for radiative transitions.

The Coulomb function has the saddle point at

$$r_0 = \{ [z^2 + k^2 l(l+1)]^{1/2} - z \} / k^2 \qquad (4)$$

and at $r \ll x_0$ the Coulomb function is very small, i.e. the incident electron almost does not penetrate this region. Therefore CBE is correct only for l_i and l_f for which r_0 is much larger than the mean radii r_i and r_f of the target states. From these considerations one can obtain the following condition of CBE validity

$$l > \left(k^2\overline{r}^2 + 2z\overline{r} + 1/4\right)^{1/2} - 1/2 \qquad (5)$$

This condition is fulfilled only for very large l_i and l_f . However, since in the case of optically allowed transitions the sum over the partial cross sections converges very slowly, the use of the CBE approximation is very convenient, especially, for the evaluation of high-L contributions at high energies. Moreover, in this approach not only the analytical expressions for the evaluation of the partial waves can be obtained, but the analytical summation of all these contributions can be carried out as well. Let us introduce the following notation

$$I\left(k_i l_i, k_f l_f; \lambda\right) = \int_0^\infty F_{kl_i}(r) r^{-\lambda-1} F_{k_f l_f}(r) dr \qquad (6)$$

Burgess et al (1970) have obtained analytical formulae for $I\left(k_i, k_j, \lambda\right)$ in the case of the dipole transitions. Their programing implementation ig given in set of routines incorporated in the STGF program (see fdip_stgf.f in BSR/UTILS/SCT_LS folder).

For the evaluation of the higher partial contributions it is necessary to evaluate the sums of $\Omega\left(l_i, l_f\right)$. Burgess (1974), based on the recurrent relations between Coulomb integrals (6), obtained the following relations:

$$\begin{aligned} S_\lambda &= \sum_{l=\lambda}^{\infty} l(I^2(k_i\lambda, k_f\lambda-1;1) + I^2(k_i\lambda, k_f\lambda-1;1) \ \propto \sum_{l=\lambda+1}^{\infty} (\Omega(l,l-1) + \Omega(l-1,l) \\ &= [(1+\lambda^2 k_i^2) I^2(k_i\lambda, k_f\lambda-1;1) - (1+\lambda^2 k_f^2) I^2(k_i\lambda-1, k_f\lambda)] / [\lambda^2(k_i^2 - k_f^2)] \end{aligned} \qquad (7)$$

$$\begin{aligned} S_{\lambda+1} &= \sum_{l=\lambda}^{\infty} l(I^2(k_i\lambda, k_f\lambda-1;1) + I^2(k_i\lambda, k_f\lambda-1;1) \propto \sum_{l=\lambda+1}^{\infty} (\Omega(l,l-1) + \Omega(l-1,l) = \\ &= [(1+\lambda^2 k_f^2) I^2(k_i\lambda, k_f\lambda-1;1) - (1+\lambda^2 k_i^2) I^2(k_i\lambda-1, k_f\lambda)] / [\lambda^2(k_i^2 - k_f^2)] \end{aligned} \qquad (8)$$

Note that here the linear momentum *k* is replaced to the reduced one, $\kappa \rightarrow k/z$, where *z* is the ion charges. It is not clear how to use these relations for neutral case. In this case we can use the other relation:

$$\sum_{l=\lambda}^{\infty} \sum_{l'=l\pm1} l_> I^2(\kappa_i l, \kappa_f l';1) = [I^2(\kappa_i\lambda, \kappa_f\lambda-1;1) - I^2(\kappa_i\lambda-1, \kappa_f\lambda)] \frac{z^2 + k_i^2\lambda^2}{[\lambda^2(k_i^2 - k_f^2)]} \qquad (9)$$

where *k* is the normal linear momentum. In terms of partial collision strengths, Buress & Sheorey (1974)

$$\Omega = \sum_{l=0}^{\lambda} \sum_{l'=l\pm1} \Omega(l,l') \ + \ [\Omega(\lambda,\lambda+1) - \Omega(\lambda+1,\lambda)] \frac{z^2/(\lambda+1) + k_i^2}{(k_i^2 - k_f^2)} \qquad (10)$$

Choosing of λ: $\lambda = \min(J_{\max}^{partial} - J_1^{target} - 1/2, J_{\max}^{partial} - J_2^{target} - 1/2) - 1$

Therefore, the sum over momenta up to infinity can be reconstructed on the basis of last two calculated partial collision strength. However, due to numerical instability for high *l*-values, it should be used with precautions. We use formula (10) with numerical partial collision strengths (after stgf or farm calculation) in the first sum, and the CBE values in the second term. It requires additionally determine the one-electron $\Omega(l,l')$ values based on the partial wave $\Omega(l,l',\pi LS)$ values after R-matrix calculations, however, it can be applied in different coupling schemes (LS. JK, JJ).

The generalization of method to the case of the quadrupole transitions is discussed in Chidichimo and Haig (1989) and Chidichimo (1988).

f_values

Description:	calculation of f-values for transition between target states based on the asymptotic coefficients in the H.DAT file
Input files:	**H.DAT** or **h.nnn, target**
Output files:	**f_values** or **f_values.nnn**
Call as:	f_values [h=.. klsp=..]

h [H.DAT]	name of input file (H.DAT format)
klsp [1]	partial wave index to choose h.nnn file for the results

s_values

Description:	calculation of f-values for transition between target states based on the asymptotic coefficients in the H.DAT file
Input files:	**H.DAT** or **h.nnn, target**
Output files:	**s_values** or **s_values.nnn**
Call as:	f_values [h=.. klsp1=.. klsp2=.. L1=.. L2=..]

h [H.DAT]	alternative name for H.DAT file
klsp1[0]	minimum index of partial wave to be considered
klsp2[0]	maximum index of partial wave to be considered
L1[-1]	minimum total orbital moment to be considered
L2[-1]	maximum total orbital moment to be considered

Asymptotic coefficients decomposition

LS coupling

The long-range potential coefficients coupling two channels are

$$ACF(i,j,k) = 2a_{ij}^{k} = 2 < \overline{\Phi}_i(x_1 \ldots x_N, \hat{\boldsymbol{r}}_{N+1}\sigma_{N+1}) \,|\, \sum_{n=1}^{N} r_n^k P_k(\cos \hat{\boldsymbol{r}}_N \bullet \hat{\boldsymbol{r}}_{N+1}) \,|\, \overline{\Phi}_j(x_1 \ldots x_N, \hat{\boldsymbol{r}}_{N+1}\sigma_{N+1} > \qquad (1)$$

In tensor notation

$$a_{ij}^{k} =< \alpha_i L_i S_i l_i s; LM_L SM_S \,|\, \boldsymbol{M}^k \bullet \boldsymbol{C}^k \,|\, \alpha_j L_j S_j l_j s; LM_L SM_S > \qquad (2)$$

where

$$M_q^k = \left(\frac{4\pi}{2k+1} \right)^{1/2} \sum_{n=1}^{N} r_n^k Y_q^k (\hat{\boldsymbol{r}}_n) \tag{3}$$

and

$$C_q^k = \left(\frac{4\pi}{2k+1} \right)^{1/2} Y_q^k (\hat{\boldsymbol{r}}_{N+1}) \ . \tag{4}$$

To evaluate expression (2), we may use the general expression (see, e.g., Cowan 1981, Eq.11.47) for matrix elements of a scalar product when angular momenta j_1, j_2 correspond to different subsystems

$$\begin{aligned} &< j_1 j_2 jm \,|\, P^{(k)}(1) \bullet Q^{(k)}(2) \,|\, j'_1\, j'_2\, j'm'> \\ &\quad = \delta_{j,j'} \delta_{m,m'} (-1)^{j'_1+j_2+j} \begin{Bmatrix} j_1 & j_2 & j \\ j'_2 & j'_1 & k \end{Bmatrix} < j_1 \,\|\, P^{(k)} \,\|\, j'_1 >< j_2 \,\|\, Q^{(k)} \,\|\, j'_2 > \end{aligned} \tag{5}$$

Then coefficients (2) are reduced to

$$a_{ij}^k = (-1)^{L_j+l_i+L} < l_i \,\|\, C^{(k)} \,\|\, l_j > \begin{Bmatrix} L_i & l_i & L \\ l_j & L_j & k \end{Bmatrix} < \alpha_i L_i \,\|\, M^{(k)} \,\|\, \alpha_j L_j > \tag{6}$$

This expression can be used for determination radiative matrix elements for transitions between target states from the asymptotic coefficients in *LS* coupling case.

jj coupling

$$\begin{aligned} a_{12}^k &=< \alpha_1 J_1 (l_1 s) j_1; JM_J \,|\, \boldsymbol{M}^k \bullet \boldsymbol{C}^k \,|\, \alpha_2 J_2 (l_2 s) j_2; JM_J > \\ &= (-1)^{J_2+j_1+J} < (l_1 s) j_1 \,\|\, C^{(k)} \,\|\, (l_2 s) j_2 > \begin{Bmatrix} J_1 & j_1 & J \\ j_2 & J_2 & k \end{Bmatrix} < \alpha_1 J_1 \,\|\, M^{(k)} \,\|\, \alpha_2 J_2 > \end{aligned} \tag{7}$$

Here we can use the uncoupling formula when operator operates only within the first subspace (see Cowam 1981, Eq.11.38):

$$< j_1 j_2 j \,\|\, P^{(k)}(1) \,\|\, j'_1\, j'_2\, j'> \ = \delta_{j_2, j'_2} (-1)^{j_1+j_2+j'+k} [j, j']^{1/2} \begin{Bmatrix} j_1 & j_2 & j \\ j' & k & j'_1 \end{Bmatrix} < j_1 \,\|\, P^{(k)} \,\|\, j'_1 > \tag{8}$$

Then

$$a_{12}^k = (-1)^{J_2+j_1+J+l_1+s+j_2+k} [\mathrm{j}_1, \mathrm{j}_2]^{1/2} \begin{Bmatrix} l_1 & s & j_1 \\ j_2 & k & l_2 \end{Bmatrix} < l_1 \,\|\, C^{(k)} \,\|\, l_2 > \begin{Bmatrix} J_1 & j_1 & J \\ j_2 & J_2 & k \end{Bmatrix} < \alpha_1 J_1 \,\|\, M^{(k)} \,\|\, \alpha_2 J_2 > \tag{9}$$

jK coupling

$$a_{12}^k =< \alpha_1(J_1l_1)K_1,s;JM_J \mid \boldsymbol{M}^k \bullet \boldsymbol{C}^k \mid \alpha_2(J_2l_2)K_2,s;JM_J > \tag{10}$$

First we should uncoupled the J_1 and J_2 by transferring to *jj*-coupling (Cowan 1981, Eq.9.25):

$$< (J_1l_1)K_1,s;J \mid J_1,(l_1s)j_1;J >= (-1)^{J_1+l_1+s+J}[K_1,j_1]^{1/2}\begin{Bmatrix} J_1 & l_1 & K_1 \\ s & J & j_1 \end{Bmatrix} \tag{11}$$

Then

$$\begin{aligned} a_{12}^k = \sum_{j_1,j_2} & (-1)^{J_1+l_1+s+J}[K_1,j_1]^{1/2}\begin{Bmatrix} J_1 & l_1 & K_1 \\ s & J & j_1 \end{Bmatrix}(-1)^{J_2+l_2+s+J}[K_2,j_2]^{1/2}\begin{Bmatrix} J_2 & l_2 & K_2 \\ s & J & j_2 \end{Bmatrix} \\ & \times(-1)^{J_2+j_1+J+l_1+s+j_2+k}[\mathrm{j}_1,\mathrm{j}_2]^{1/2}\begin{Bmatrix} l_1 & s & j_1 \\ j_2 & k & l_2 \end{Bmatrix}< l_1 \| C^{(k)} \| l_2 >\begin{Bmatrix} J_1 & j_1 & J \\ j_2 & J_2 & k \end{Bmatrix}< \alpha_1J_1 \| M^{(k)} \| \alpha_2J_2 > \\ = \sum_{j_1,j_2} & (-1)^{J_1+2J_2+3J+j_1+j_2+l_2+3s+k}[j_1,j_2]\begin{Bmatrix} J_1 & l_1 & K_1 \\ s & J & j_1 \end{Bmatrix}\begin{Bmatrix} l_1 & k & l_2 \\ j_2 & s & j_1 \end{Bmatrix}\begin{Bmatrix} J_1 & k & J_2 \\ j_2 & J & j_1 \end{Bmatrix} \\ & \times[K_1,K_2]^{1/2}\begin{Bmatrix} J_2 & l_2 & K_2 \\ s & J & j_2 \end{Bmatrix}< l_1 \| C^{(k)} \| l_2 >< \alpha_1J_1 \| M^{(k)} \| \alpha_2J_2 > \end{aligned} \tag{12}$$

Now let reduce sum over j_1, using the sum rule (Cowan 1981, Eq. 5.33):

$$\sum_x (-1)^{S+x}[x]\begin{Bmatrix} l_1 & j_2 & l_3 \\ l'_3 & l'_2 & x \end{Bmatrix}\begin{Bmatrix} j_2 & j_3 & j_1 \\ l'_1 & l'_3 & x \end{Bmatrix}\begin{Bmatrix} l_1 & j_3 & l_2 \\ l'_1 & l'_2 & x \end{Bmatrix} = \begin{Bmatrix} j_1 & j_2 & j_3 \\ l_1 & l_2 & l_3 \end{Bmatrix}\begin{Bmatrix} l_3 & j_1 & l_2 \\ l'_1 & l'_2 & l'_3 \end{Bmatrix} \tag{13}$$

where S = $j_1 + j_2 + j_3 + l_1 + l_2 + l_3 + l'_1 + l'_2 + l'_3$. Then

$$\begin{aligned} a_{12}^k = \sum_{j_2} & (-1)^{J_1+2J_2+3J+j_2+l_2+3s+k-l_2-l_1-k-J_1-J_2-K_1-j_2-J-s}[j_2]\begin{Bmatrix} l_2 & l_1 & k \\ J_1 & J_2 & K_1 \end{Bmatrix}\begin{Bmatrix} K_1 & l_2 & J_2 \\ j_2 & J & s \end{Bmatrix} \\ & \times[K_1,K_2]^{1/2}\begin{Bmatrix} J_2 & l_2 & K_2 \\ s & J & j_2 \end{Bmatrix}< l_1 \| C^{(k)} \| l_2 >< \alpha_1J_1 \| M^{(k)} \| \alpha_2J_2 > \\ = \sum_{j_2} & (-1)^{J_2+2J+2s-l_1-K_1}[j_2]\begin{Bmatrix} J_2 & l_2 & K_1 \\ s & J & j_2 \end{Bmatrix}\begin{Bmatrix} J_2 & l_2 & K_2 \\ s & J & j_2 \end{Bmatrix} \\ & \times[K_1,K_2]^{1/2}\begin{Bmatrix} l_2 & l_1 & k \\ J_1 & J_2 & K_1 \end{Bmatrix}< l_1 \| C^{(k)} \| l_2 >< \alpha_1J_1 \| M^{(k)} \| \alpha_2J_2 > \end{aligned} \tag{14}$$

Using the orthogonal relation (Cowan 1981, Eq.5.31)

$$\sum_x [x]\begin{Bmatrix} j_1 & j_2 & a \\ l_1 & l_2 & x \end{Bmatrix}\begin{Bmatrix} j_1 & j_2 & b \\ l_1 & l_2 & x \end{Bmatrix} = \frac{\delta(a,b)}{2a+1} \tag{15}$$

finally obtain

$$a_{12}^{k} = \delta(K_1, K_2)(-1)^{J_2+2J+1-l_1-K_1} \begin{Bmatrix} l_2 & l_1 & k \\ J_1 & J_2 & K_1 \end{Bmatrix} < l_1 \| C^{(k)} \| l_2 >< \alpha_1 J_1 \| M^{(k)} \| \alpha_2 J_2 > \tag{16}$$