List of the ionization utility-programs:

BSR_SE1

BSR_SE2

BSR_SEn

DIF_SEC_He

TDCS

DDCS

SDCS

SDCS_OMT

BOUND _ OVL

SEC _ IONB

For more information, see also **BSR_ionization.pdf** notes

The BSR_SE1 utility-program

The BSR_SE1 program calculates the continuum-pseudostates overlaps $\left\langle \Psi_{LS}^{L_f S_f, k_2(-)} \middle| \Phi^p(LS) \right\rangle$, needed for determination of the ionization amplitudes in the pseudostate approach (see BSR_ionization description). $\Psi_{LS}^{L_f S_f, k_2(-)}$ is an ion-ejected-electron eigenstate of total orbital angular momentum L and total spin S with ingoing-wave boundary conditions for well-defined orbital spin angular momenta L_f and S_f of the ion; $\Phi^p(LS)$ is the pseudostate function. Note that in the present approach both the scattering and pseudostates functions obtained in the same B-spline basis and using the same close-coupling expansions but with different boundary conditions. The calculation of the continuum functions are very similar to the algorithm implemented in the BSR_PHOT program for photoionization calculations, and also use the ASYPCK program (Crees 1980) to determine the solutions in the asymptotic region.

1. Related formulas

In *R*-matrix approach, the matrix elements between some bound state and given scattering solution can be obtained from the matrix elements between this state and R-matrix basis functions $\Psi^k(LS)$ as

$$\left\langle \Psi_{LS}^{L_f S_f, k_2(-)} \middle| \Phi^p(LS) \right\rangle = \sum_k A_{kE} \left\langle \Psi^k(LS) \middle| \Phi^p(LS) \right\rangle \tag{1}$$

where

$$A_{Ek,j} = \frac{1}{2a} (E_k - E)^{-1} \mathbf{w}_{k,j}^T \mathbf{R}^{-1} \mathbf{F}_j^-(a) \,. \tag{2}$$

Index *j* define the solutions which correspond to asymptotic conditions with a plane wave in the direction of the ejected electron momentum *k* in channel *j* and ingoing waves in all open channels, R stands for R-matrix, w is the vector of surface amplitudes, and $\mathbf{F}_j^-(a)$ is the value of scattering radial function at the boundary $r=a$. This function we find using the asymptotic solutions from program ASYPCK (more details see in BSR_PHOT description).

2. Structure and data flow

The block diagram of the program BSR_SE1, along with the data flow, is shown in Figure 1. All input/output files have prescribed internal names (see section 4 below). First, in subroutine read_data, the program reads the input parameters from the file bsr_se1.inp (section 3). The only mandatory parameters is the energy of ejected electron. Then program also reads the description of scattering channels for the ejected electron from file target and the description of pseudostates from file target_ps.

The subroutine rovl_out calculates preliminary quantities, the R-matrix overlap vectors a = S b, where S is the overlap matrix (placed in the bsr_mat.nnn), b are the R-matrix solutions (placed in the rsol.nnn). These quantities are recorded in file rovl.nnn, to be possibly used in calculations for other energies. Then the subroutine sub_sct carries out the scattering calculations for the given electron energy, giving as result the R-matrix, K-matrix and boundary values of scattering radial functions at boundary. Then in loop over pseudostates, the program first calculate the vector of overlaps of this particular pseudostate with all R-matrix solutions (i.e., the quantities $\left\langle \Psi^k(LS) \middle| \Phi^p(LS) \right\rangle$ in Eq.(1)) . Computationally, we need just matrix multiplication c a, where c is the pseudostate B-spline expansion, a is the matrix Sb described above. Finally, the subroutine rm_ovlf calculates the coefficients $A_{Ek,j}$, Eq.(2), and the matrix elements $\left\langle \Psi_{LS}^{L_f S_f, k_2(-)} \middle| \Phi^p(LS) \right\rangle$ for each pseudostate and each open scattering channel. These data are recorded in file projection1.

3. Input parameters

The input parameters are provided in the file bsr_se.inp or/and from command line. All input data have the format '*name=value*', and should be placed at the beginning of the line; otherwise they will be ignored. The default values of the data are indicated in the brackets. All default values, along with the unit numbers and the default file names are placed in the module mod_bsr_se. Some of the input parameters deal with the asymptotic package ASYPCK. The parameters without default values are mandatory (only electron energy).

EL	electron energy in eV.
debug [0]	defines the level of debug printing in the file bsr_se.out.
	parameters for ASYPCK program:

ac [0.001]	accuracy for the asymptotic solutions.
delta [0.1]	in ASYPCK, the solutions are first obtained at R1=R-DR and R2=R+DR,. and then by interpolation - in R.
mfgi [300]	an asymptotic expansion of the solution will be sought at R + DR*MFG.
iauto [2]	if > 0, then ASYPCK may automatically increase parameter MFG.

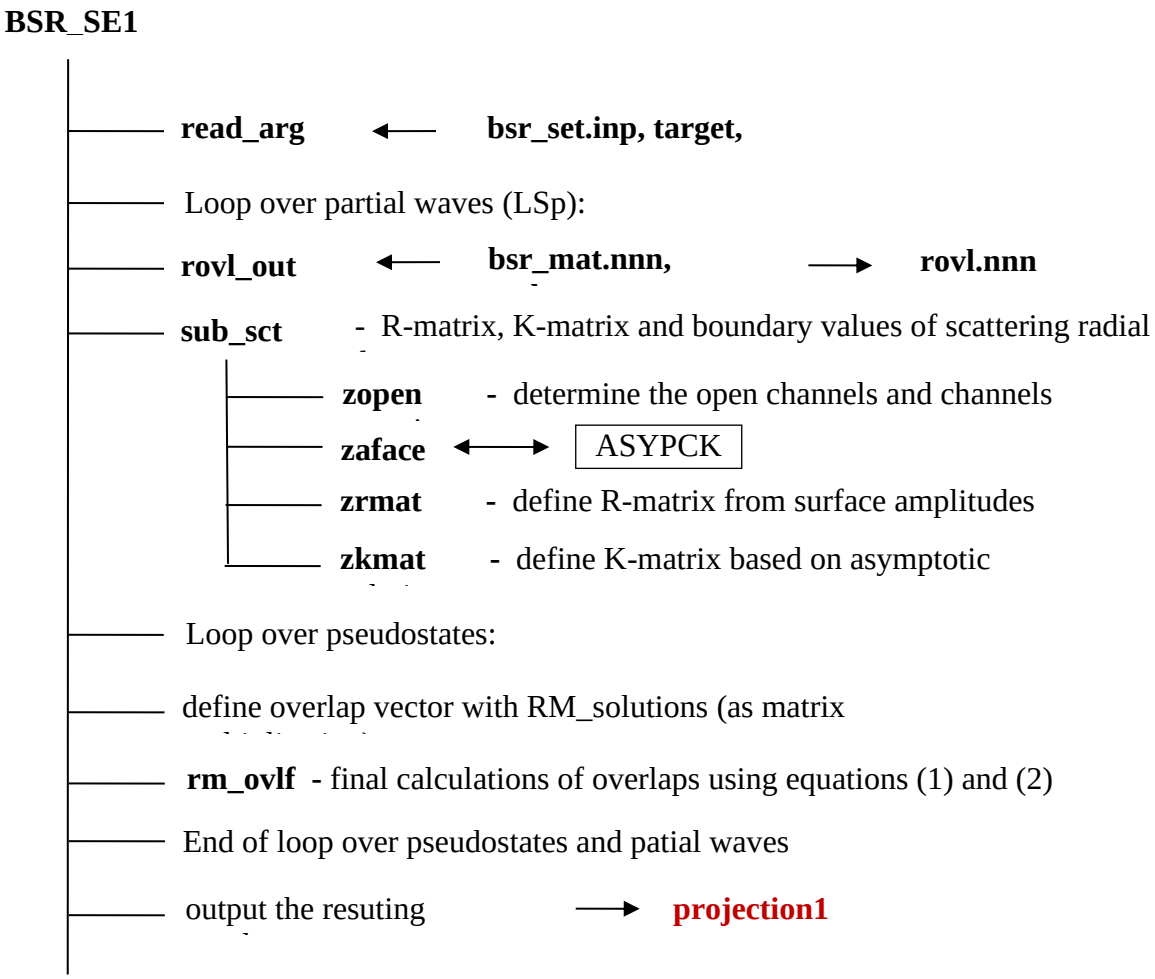

Fig. 11.1. Block diagram for the program BSR_SE1 and data flow (see text).

11.4 Data files

bsr_se1.inp	File type: formatted sequential input (optional). Created by the user. Read by the routine read_data. Description: input parameter for given run.
h.nnn	File type: unformatted sequential input. Created by the BSR_HD program. Read by routine read_Hdat. Description: diagonalized Hamiltonian matrix data for the inner region.
rsol.nnn	File type: unformatted sequential input. Created by BSR_HD program. Read by the routine rovl_out. Description: R-matrix solutions in B-spline basis for scattering partial wave nnn.
bsr_mat.nnn	File type: unformatted sequential input. Written by the program BSR_MAT. Read by the routine rovl_out. Description: used to read overlap matrix S for given partial waves nnn.
ubound.nnn	File type: unformatted sequential input. Written by the program BSR_HD. Read by the routine bsr_se1. Description: B-spline expansions for pseudostates.
rovl.nnn	File type: formatted sequential input/output. Created by the program BSR_SE1. Description: the R-matrix overlap vectors a = S b, where S is the overlap matrix b are the R-matrix solutions. This file created to avoid recalculations for other electron energies.
target	File type: formatted sequential input. Read by routines r_target and r_channels. Description: contains description of target states and scattering channels for BSR calculations of the ejected electron.
target_ps	File type: formatted sequential input. Read by routines r_target. Description: contains description of target states for the BSR calculations of the incident electron. Here used to get information about pseudostates.
bsr_se1.log	File type: formatted sequential output. Created by the program BSR_SE1. Read by the user. Description: running information.
projection1	File type: formatted sequential output. Created by the program BSR_SE1. Description: overlaps of pseudostates with continuum wave function for given electron energy.

11.5 Structure of the projection1 *output file*

The projection1 contain the overlaps between all pseudostates and scattering channels for the given electron energy.

1. `'EL =',EL  - electron energy in eV.`
2. 'nsp =', nsp `- number of pseudostates`

`For each psedostate:`

3. is, nopen `- index of the pseudostate and the number of open channels`
4. `ddr(1:nopen)` – real part of the overlaps
5. `ddi(1:nopen)` – imaginary part of the overlaps
6. `wt(1:ntarget)` – ion target weights

The BSR_SE2 utility-program

The BSR_SE2 program is an extension of the BSR_SE1 program to the equal-energy case where energy of scattering electron is close to the energy of ejected electron. In this case we also need the exchange overlaps $\left\langle \Psi_{LS}^{L_f S_f, k_1(-)} \middle| \Phi^p(LS) \right\rangle$, required for determination of the exchange ionization amplitudes. The structure and data flow in BSR_SE2 is closed to the BSR_SE1, however, here we need more input data to find correctly all related energy.

EK	energy of incident electron in Ry (k_0^2 value)
EL	electron energy of the ejected electron in eV. **EL**=0 means equal-energy case.
itarg	index of ion target states after ionization (**itrag**>1 means ionization plus excitation)

Structure of the **projection2** *output file*

The **projection2** contain the overlaps between all pseudostates and scattering channels for two electron energies, corresponding ejected and scattered electrons.

```
1. 'EK =',EK, E0   - incident electron energy in Ry and eV.
2. 'nsp  =', nsp   - number of pseudostates
3. 'ion  =', itarg - index of target state
Ejected electron data:
4. 'EL2 = ', EL2   - ejected electron energy
    For each psedostate:
5.  is, nopen    - index of the pseudostate and the number of open
                   channels
6.  ddr(1:nopen) - real part of the overlaps
7.  ddi(1:nopen) - imaginary part of the overlaps
Ejected electron data:
4. 'EL1 = ', EL1   - ejected electron energy
    For each psedostate:
5.  is, nopen    - index of the pseudostate and the number of open
                   channels
6.  ddr(1:nopen) - real part of the overlaps
7.  ddi(1:nopen) - imaginary part of the overlaps
```

The BSR_SEn utility-program

The BSR_SEn program is the most extension version for the overlap calculations. The whole interval of possible ejected-electron. Whereas the BSR_SE1 and BSR_SE2 is enough for calculations of TDCS for one specific energy, the BSR_SEn is used for SDCS and total ionization(+excitation) cross sections. The whole interval of possible ejected-electron energies is divided on several points and the calculations are carried out for each energy and each possible final ionic state. The input parameters are then included the energy of the incident electron (**EK**) and the number of ejected-electron energy points (**nq**).

Structure of the **projections** *output file*

The **projections** contain the overlaps between all pseudostates and scattering channels for 2***nq** electron energies. For each ejected-electron energy there is corresponding point for the scattered electron. It allows calculate both direct and exchange amplitude.

```
1. 'EK =',EK,' Ry  ',E0,' eV'   - incident electron energy
2. 'nsp   =',nsp          - number of pseudostates
3. 'nopen =',nopen        - maximum number of open ionic states
4. 'nq    =',nq           - number of energy points
Loop over final target states, it=1,nopen:
5. 'targ =',it           - final ion state
Loop over ejected(scattered)-electron energies, iq=1,2*nq:
6. iq,qq,EL    ->  iq,qq,EL  - energy index and energies in Ry and eV
Loop over pseudostates, is=1,nps:
7. is, nop(is)   - index of the pseudostate and the number of open
                   channels in given case
8.  ddr(1:nop(is) - real part of the overlaps
9.  ddi(1:nop(is) - imaginary part of the overlaps
10. wt(1:ntarg)   - weights for all ionic target states
End all loops
```

DIF_SEC_HE

Description:	Differential cross sections and scattering amplitudes (LS coupling) for scattering on neutral atoms with closed subshells (as He). Considered all transitions from the ground state for one input energy.
Input files:	zarm.tmb, target
Output files:	dif_sec_He
Call as:	dif_sec_He ek=... [res=... tm=... g0=... hg=... ng=... diff=...]
Parameters	ek - electron energy (k^2, Ry) res - file-name for results [dif_sec_He] tm - file-name for T-matrix elements [zarm.tmb] g0 - initial point in the degree grid [0] hg - step in degree grid [1] ng - number of point in degree grid [181] diff - if =1, differential cross sections are recorded [0]

Related formulas

For neural atoms,

$$f_{01}(\alpha_0 L_0 S_0, M_0 M_{S_0}, \mu_0 \to \alpha_1 L_1 S_1, M_1 M_{S_1}, \mu_1, \theta, \varphi) =$$

$$-i\sqrt{\frac{\pi}{k_0 k_1}} \sum_{l_0, l_1, L, S, \Pi} i^{(l_0 - l_1)} \sqrt{(2l_0+1)} C^{L_0 l_0 L}_{M_0 0 M_L} C^{L_1 l_1 L}_{M_1 m_1 M_L} C^{S_0 \frac{1}{2} S}_{M_{S_0} \mu_0 M_S} C^{S_1 \frac{1}{2} S}_{M_{S_1} \mu_1 M_S} T^{LS\pi}_{l_0 l_1}(\alpha_0 L_0 S_0 \to \alpha_1 L_1 S_1) Y_{l_1 m_1}(\theta, \varphi)$$

with

$M_L = M_0 + m_0 = M_0$ (incident beam along z-axes, m_0=0)

$m_1 = M_L - M_1 = M_0 - M_1$

$M_S = M_{S_0} + \mu_0 = M_{S_1} + \mu_1$ (μ_0 - defines scattering electron polarization)

Spherical harmonic:

$$Y_l^m(\theta, \varphi) = (-1)^m \sqrt{\frac{(2l+1)}{4\pi} \frac{(l-m)!}{(l+m)!}} e^{im\varphi} P_l^m(\cos\theta), \qquad (m > 0)$$

$$Y_l^{-m}(\theta, \varphi) = (-1)^m (Y_l^m)^* = \sqrt{\frac{(2l+1)}{4\pi} \frac{(l-m)!}{(l+m)!}} e^{-im\varphi} P_l^m(\cos\theta), \qquad (m > 0)$$

Another representation:

$$Y_l^m(\theta, \varphi) = \Theta_{lm}(\theta) \Phi_m(\varphi) \qquad \Phi_m(\varphi) = \sqrt{\frac{1}{2\pi}} e^{im\varphi}$$

$$\Theta_{lm}(\theta) = (-1)^m \sqrt{\frac{(2l+1)}{2}\frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta) = (-1)^m \overline{P}_l^m(\cos\theta), \qquad (m>0)$$

$$\Theta_{l-m}(\theta) = (-1)^m \Theta_{lm}(\theta) = \overline{P}_l^m(\cos\theta)$$

where we use normalized associated Legendre polynomials:

$$\tilde{P}_{lm}(\theta) = (-1)^{m(m>0)} \sqrt{\frac{(2l+1)}{2}\frac{(l-|m|)!}{(l+|m|)!}} P_l^m(\cos\theta) \qquad \text{(subroutine ALEGFM)}$$

Usually, we use representation

$$f(\alpha_1 L_1, M_1, S_1, M_{S_1}, \mu_1; \alpha_0 L_0, M_0, S_0, M_{S_0}, \mu_0) = \sum_{SM_S} C^{S_0 \frac{1}{2} S}_{M_{S_0} \mu_0 M_S} C^{S_1 \frac{1}{2} S}_{M_{S_1} \mu_1 M_S} f^S(M_1, M_0; \theta, \varphi)$$

where we extract the part which does not depend on the spin projections:

$$f^S{}_{01}(\alpha_0 L_0 S_0, M_0 \to \alpha_1 L_1 S_1, M_1; \theta, \varphi) =$$

$$-i\sqrt{\frac{\pi}{k_0 k_1}} \sum_{l_0, l_1, L, \Pi} i^{(l_0 - l_1)} \sqrt{(2l_0+1)} C^{L_0 l_0 L}_{M_0 0 M_L} C^{L_1 l_1 L}_{M_1 m_1 M_L} T^{LS\pi}_{l_0 l_1}(\alpha_0 L_0 S_0 \to \alpha_1 L_1 S_1) Y_{l_1 m_1}(\theta, \varphi)$$

If we sum and average over spin projections, for any product f(S,L)*f*(S',L') we obtain:

$\frac{(2S+1)}{2(2S_0+1)} \delta(S,S') f(L) * f^*(L')$ so we may working only with $f^S(M_1, M_0; \theta)$

and in final results we put $|f|^2 = \sum_S \frac{(2S+1)}{2(2S_0+1)} |f_S|^2$ - good for unpolarized electrons.

In general, scattering amplitude $f(\ldots)$ is a function of the 8 parameters (6 m's and (θ, φ)). Switching to f^S reduces it to 4 parameters.

He-simplification:

S_0=0; S=1/2; $\frac{(2S+1)}{2(2S_0+1)} = 1$, so only one f_S is needed.

L_0=0, $M_L = M_0 = 0$, m_1=−M_1

$$f^{He}(\alpha_1 L_1 M_1, S_1; \theta, \phi) = -i\sqrt{\frac{1}{2k_0 k_1}} \sum_{l_0, l_1} i^{l_0 - l_1} (2l_0+1)^{1/2} (L_1, -m_1; l_1 m_1 | L, 0) \delta(L, l_0) e^{i m_1 \phi} \tilde{P}_{l_1 m_1}(\theta) T^{LSP}_{\alpha_0 L_0 l_0; \alpha_1 L_1 l_1}$$

Program dif_sec_He outputs the following reduced scattering amplitudes:

(for the excitation to each $\alpha_1 L_1 S_1$ excited state from the ^{1}S ground state)

$$f^{\alpha_1 L_1 S_1}(M_1 = -m_1; \theta) = \sum_{l_0, l_1} i^{l_0 - l_1} (2l_0+1)^{1/2} (L_1, -m_1; l_1 m_1 | L, 0) \delta(L, l_0) \tilde{P}_{l_1 m_1}(\theta) T^{LSP}_{\alpha_0 L_0 l_0; \alpha_1 L_1 l_1}$$

where we do not include common factor: $-i\sqrt{\frac{1}{2k_0k_1}}e^{im_1\phi}$

To check the amplitudes, angle-integrated cross section is estimated by direct integration:

$$\sigma(\theta,\phi)=\frac{k_1}{k_0}|\ f\ |^2\ d\Omega \qquad \sigma(\theta,\phi)=\frac{1}{2k_0^2}\sum_{M_1}|\ f^{\alpha_1 L_1 S_1}(M_1;\theta)|^2\ d\Omega$$

T D C S

Description:	triple differential cross section for ionization of He or other closed-shell atoms, using the projection method
Input files:	**target_ion, projection2, difsec_He**
Output files:	**tdcs.out**
Call as:	**tdcs** [key-word parameters]

Key-word parameters:

ig [10]	polar scattered-electron angle (degrees, -180 < ig < 180)
fi [0]	azimuthal scattered-electron angle (0 - 2π)
itarg [1]	index of the ionic-target state
mode [2]	mode for adding the direct and exchange amplitudes: 0 – incoherent 1 – coherent 2 – spin-dependent (see BSR_ionozation.pdf, Eq.(15,16) .
geom [1]	geometry of the ionization process (see below, section 7)

Input-output parameters:

ion [target_ion]	target file for e-A+ scattering
dcs [difsec_He]	file with pseudostates amplitudes
p [projection2]	file with pseudostate-continuum overlaps
out [tdcs.out]	output file for final TDCS
iout [1]	output format for the given geometry
i16 [0]	define units for output TDCS
idte [1]	step for θ_2 in case of output of full 3D TDCS [**geom**=10]
idfi [1]	step for φ_2 in case of output of full 3D TDCS [**geom**=10]
idel [90]	= θ_2 -θ_1, when considered fixed (θ_2 -θ_1) geometry [**geom**=6]

Debug parameters:

exch [1]	if =0, exchange is excluded; if=2, direct process is excluded.

Related formulas

Take some main expressions from BSR_ionization.pdf notes. Working formulae are

$$f^{S_1}(\mathbf{k}_1,\mathbf{k}_2,M_f)=\mathrm{e}^{-\mathrm{i}m_1\varphi_1}\sum_{l_2m_2} i^{l_2}\,\mathrm{e}^{-\mathrm{i}(\sigma_{l_2}-\sigma_0)}\,\mathrm{e}^{-\mathrm{i}m_2\varphi_2}\,\tilde{P}_{l_2m_2}(\theta_2)\sum_{L_1,M_1} C^{L_f l_2 L_1}_{M_f m_2 M_1}\sum_{p(S_1)} f_p^{He}(M_1;\theta_1)d_p^f$$
$$g^{S_1}(\mathbf{k}_1,\mathbf{k}_2,M_f)=f^{S_1}(\mathbf{k}_2,\mathbf{k}_1,M_f) \qquad (1)$$

$$\frac{d\sigma^f}{d\Omega_1 d\Omega_2 dE}=\frac{1}{2\pi}\frac{1}{k_0^2}\sum_{M_f}(|F^0(\mathbf{k}_1,\mathbf{k}_2)|^2+|F^1(\mathbf{k}_1,\mathbf{k}_2)|^2) \qquad (2)$$

$$F^0(\boldsymbol{k}_1,\boldsymbol{k}_2)=f^0(\boldsymbol{k}_1,\boldsymbol{k}_2)-\frac{1}{2}g^0(\boldsymbol{k}_2,\boldsymbol{k}_1)-\frac{\sqrt{3}}{2}g^1(\boldsymbol{k}_2,\boldsymbol{k}_1) \qquad (3a)$$

$$F^1(\boldsymbol{k}_1,\boldsymbol{k}_2)=f^1(\boldsymbol{k}_1,\boldsymbol{k}_2)+\frac{1}{2}g^1(\boldsymbol{k}_2,\boldsymbol{k}_1)-\frac{\sqrt{3}}{2}g^0(\boldsymbol{k}_2,\boldsymbol{k}_1) \qquad (3b)$$

where pseudostates amplitudes $f_p^{He}(M_1;\theta)$ are provided by the utility **difsec_He**, and overlap factors d_p^f are provided by the utilities **BSR_SE2**.

Structure and data flow

1. Read all arguments described above. Read the description of ionic scattering channels from file **target_ion**.

2. The **tdcs** utility then reads (from file **dif_sec_He**) the incident electron E_o and the amplitudes $f_p^{He}(M_1;\theta)$ in four-dimensional array

$$f(1:2,-L_1^{\max}:L_1^{\max},1:n_{ps},1:n_g),$$

where first index stands for real and imaginary parts, second index corresponds to M_1, third index runs pseudostates, and forth index define the angular grid θ_1.

3. Then we read the direct overlap factors d_p^f for each pseudostate as real and imaginary arrays

$$\mathrm{dr}(1{:}n_{\mathrm{ch}},\,1{:}\,n_{\mathrm{ps}}) \quad \text{and} \quad \mathrm{di}(1{:}\,n_{\mathrm{ch}},\,1{:}\,n_{\mathrm{ps}})$$

for each final ionic channel $e_2 + A^+_f$ and for each pseudostate. It is done for given ejected-electron energy E_2. (Why we take complex conjugate: di → -di? Does it matter?)

3. Then we combine the arrays f and dr, di and make preliminary summation over pseudostates with the same total term $\alpha_1 L_1 S_1$. In result we get array

$$fi(1:2,-L_1^{\max}:L_1^{\max},1:n_{ch},1:n_{\alpha_1 L_1 S},1:n_g)$$

and the total ionization amplitude looks as

$$f^{S_1}(\mathbf{k}_1,\mathbf{k}_2,A_f^+)=\mathrm{e}^{-\mathrm{i}m_1\varphi_1}\sum_{l_2m_2} i^{l_2}\,\mathrm{e}^{-\mathrm{i}\sigma_{l_2}}\,\mathrm{e}^{-\mathrm{i}m_2\varphi_2}\,\tilde{P}_{l_2m_2}(\theta_2)\sum_{L_1,M_1} C^{L_f l_2 L_1}_{M_f m_2 M_1}\delta(M_1,-m_1)\,f_i\;(M_1;channel_f,\alpha_1 L_1 S_1,\theta_1) \quad (4)$$

4. Then we read the overlap overlap factors d_p^f and prepare array f_j which will be used to define the exchange ionization amplitude $g^{S_1}(\mathbf{k}_1,\mathbf{k}_2,A_f^+)$

5. Then we call subroutines **TDCS1** and **TDCS2** to get final expressions for the amplitudes $f^{S_1}(\mathbf{k}_1,\mathbf{k}_2,A_f^+)$ and $g^{S_1}(\mathbf{k}_1,\mathbf{k}_2,A_f^+)$. In programs they are defined by arrays

$$FMS1(1:2,-L_f^{\max}:L_f^{\max},1:(2S_1+1),1:n_{target})$$

$$FMS2(1:2,-L_f^{\max}:L_f^{\max},1:(2S_1+1),1:n_{target})$$

which obtained by summation expression (20) over l_2,m_2,L_1,M_1 for given final state $\alpha_1 L_1 S$, M_f and S_1. These arrays are defined for given values of θ_1,θ_2, and φ_2, whereas φ_1=0. The formula programed in program is a complex conjugate to the Eq.(4):

$$f^{S_1}(\theta_1,\theta_2,\varphi_2,A_f^+,M_f)=\sum_{L_1,M_1}\sum_{l_2m_2}(-i)^{l_2}\,\mathrm{e}^{\mathrm{i}\sigma_{l_2}}\,\mathrm{e}^{\mathrm{i}m_2\varphi_2}\,\tilde{P}_{l_2m_2}(\theta_2)C^{L_f l_2 L_1}_{M_f m_2 M_1} f_i\;(M_1;channel_f,\alpha_1 L_1 S_1,\theta_1) \quad (5a)$$

$$g^{S_1}(\theta_1,\theta_2,\varphi_2,A_f^+,M_f)=\sum_{L_1,M_1}\sum_{l_2m_2}(-i)^{l_2}\,\mathrm{e}^{\mathrm{i}\sigma_{l_2}}\,\mathrm{e}^{\mathrm{i}m_1\varphi_2}\,\tilde{P}_{l_2m_2}(\theta_1)C^{L_f l_2 L_1}_{M_f m_2 M_1} f_j\;(M_1;channel_f,\alpha_1 L_1 S_1,\theta_2) \quad (5b)$$

6. Next step – summation of amplitudes for given scattering angles and final ion state f in subroutine **Sum_ion_ampl**(mode). There is three modes:

mode =0 - incoherent adding (debug option)

$$DCS(\theta_1,\theta_2,\varphi_2,A_f^+)=\sum_{S_1,M_f}|\,f^{S_1}(\theta_1,\theta_2,\varphi_2,A_f^+,M_f)|^2+\sum_{S_1,M_f}|\,g^{S_1}(\theta_1,\theta_2,\varphi_2,A_f^+,M_f)|^2$$

mode =1 - coherent adding (debug option)

$$DCS(\theta_1,\theta_2,\varphi_2,A_f^+)=\sum_{S_1,M_f}|\,f^{S_1}(\theta_1,\theta_2,\varphi_2,A_f^+,M_f)+g^{S_1}(\theta_1,\theta_2,\varphi_2,A_f^+,M_f)|^2$$

mode =2 - spin-dependent mode (default main option). See Eq(2) and Eq.(3)

7. The calls to subroutines **TDCS1, TDCS2** and **SUM_ION_AMPL** depends on the geometry under consideration. The program covers several geometries (defined by parameter **geom**, see also Fig.1-4):

geom=1 Standard scattering plane, Fig.1
Calculations done for given fixed θ_1, φ_1=0, φ_2=0 and θ_2=-180-180^o.
Program has two output modes, **out=1** - θ_2=0-360^o **and out=2** - θ_2=-180 - 180^o .

geom=2 Perpendicular scattering plane, Fig.2
Calculations done for given fixed θ_1, φ_1=0, φ_2=90 and θ_2=-180 - 180^o.
Program has two output modes, **out=1** - θ_2=0-360^o **and out=2** - θ_2=-180 - 180^o .

geom=3 Fully-perpendicular scattering plane, Fig.3
Calculations done for given fixed θ_1, θ_2=90^o , φ_2=0-360^o.
Program has two output modes, **out=1** - φ_2=0-360^o **and out=2** - φ_2=-180 - 180^o .

geom=4 The scattering geometry of Dublin Groop with Ψ=90^o, Fig.4
Calculations done for given fixed θ_1=θ_2=90^o , φ_2=ξ_1+ξ_2=0-360^o.
Program has one output modes, **out=1** - φ_2=0-360^o.

geom=5 Symmetric θ_1, θ_2: θ_1=-θ_2, φ_2=0.
Calculations done for θ_2=0-180^o
Program has one output modes, **out=1** - θ_2=0-180^o.

geom=6 Fixed θ_1=- θ_2 - delta, φ_2=0. The difference 'delta' is given by paremeter **idel**
Calculations done for θ_2=-180 - 180^o
Program has one output modes, **out=1** - θ_2=0-360^o.

geom=10 Full 3D mode for fixed θ_1, φ_1=0.
Calculations done for θ_2= 0 - 180^o, φ_2=0-360^o.
Main output for θ_2= 0,180^o,**idte**; φ_2=0,360^o,**idfi**, is given in file **<out>.theta_fi**.
Results in Cartesian coordinate is given in file **<out>.xyz**
Additional output in file **<out>.sec** TDCS in the xz, yz, and xy planes what corresponds **geom** = 1,2,3.

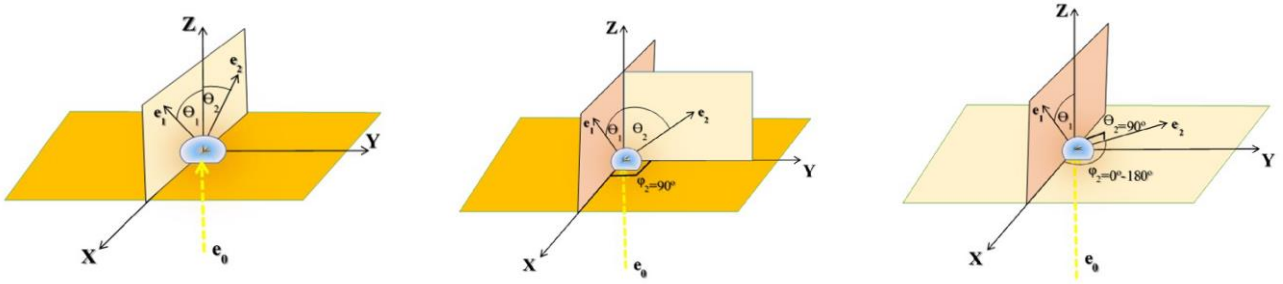

Fig. 1 The x-z Scattering Plane for Heidelberg group (Coplanar). φ_1 = 0^o; φ_2 = 0.

Fig. 2 The y-z Plane shows the Half-Perpendicular Plane for Heidelberg group. φ_1 = 0^o; φ_2 = 90^o.

Fig. 3 The x-y Plane shows the Full-Perpendicular Plane for Heidelberg group. φ_1 = 0^o; φ_2 = 0^o - 180^o.

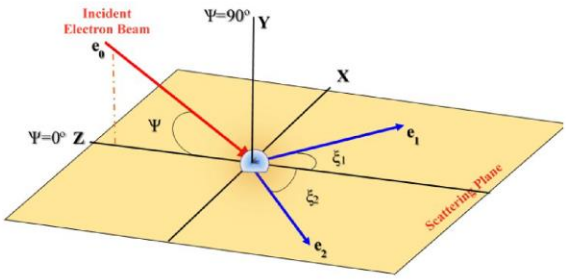

Fig. 4 The scattering geometry of Dublin Groop. Ψ=0º → Coplanar geometry, Ψ=90º → Full-Perpendicular geometry.

DDCS - double-ionization cross sections

SDCS - single-ionization cross sections

These programs repeat the TDCS calculations with further integration other variables.

D D C S

Description:	double differential cross section for ionization of He or other closed-shell atoms, using the projection method
Input files:	**target_ion, projection2, difsec_He**
Output files:	**ddcs.out**
Call as:	**ddcs** [key-word parameters]

Key-word parameters:

itarg [1]	index of the ionic-target state
mode [2]	mode for adding the direct and exchange amplitudes: 0 – incoherent 1 – coherent 2 – spin-dependent (see BSR_ionozation.pdf, Eq.(15,16) .

Input-output parameters:

ion [target_ion]	target file for e-A+ scattering
dcs [difsec_He]	file with pseudostates amplitudes
p [projection2]	file with pseudostate-continuum overlaps
out [ddcs.out]	output file for final DDCS
i16 [0]	define units for output DDCS

Debug parameters:

exch [1]	if =0, exchange is excluded; if=2, direct process is excluded.

S D C S

Description:	single differential cross section for ionization of He or other closed-shell atoms, using the projection method
Input files:	**target_ion, projections, difsec_He**
Output files:	**sdcs_E0_eV** or **sdcs_E0_eV_t**
Call as:	**sdcs** [key-word parameters]

Key-word parameters:

itarg [1]	index of the ionic-target state
mode [2]	mode for adding the direct and exchange amplitudes: 0 – incoherent 1 – coherent 2 – spin-dependent (see BSR_ionozation.pdf, Eq.(15,16) .

Input-output parameters:

ion [target_ion]	target file for e-A+ scattering
dcs [difsec_He]	file with pseudostates amplitudes
p [projection2]	file with pseudostate-continuum overlaps
out [ddcs.out]	output file for final DDCS
i16 [0]	define units for output DDCS

Debug parameters:

exch [1]	if =0, exchange is excluded; if=2, direct process is excluded.

S D C S _ O M T

Description:	single differential cross section for electron-impact ionization based on the excitation collision strengths for pseudo-states and their projections to the real ionic continuum
Input files:	**target_ps, target_ion, zarm.omb, projections**
Output files:	**sdcs_omt_nn** (nn indicates final ionic state)
Call as:	**sdcs_omt [is=… i16=… iion=… eps_ek=… debug=…]**

is [1] - index of initial atomic state

i16 [0] - control the units for output cross sections

iion - possible redifinition of number of bound pseudostates

eps_ek [1.d-6] - tollerance for initial electron energy

ion_state [0] - indicate specific final ion state;
File names can be re-definded through the arguments as; 0 means all states

AF_t=... AF_i=... om=... p=... out=...

B O U N D _ O V L

Description:	decomposition of pseudostates in terms of ionic target states
Input files:	**target, target_ps, w.nnn**
Output files:	**bound_ovl**
Call as:	**bound_ovl** (in the ion-target folder)

Structure of the **bound_ovl** file (formatted):

1. 'ntarg_ion = ', ntarg - number of ionic states

2. 'nps = ', nps - number of pseudostates

Loop over psedostates, i=1,nps:

3. i, LPS(i),SPS(i),PPS(i),EPS(i), wtarg(1:ntarg,i), SUM(wtarg(:,i))

Index, term, energies and decomposition weights for each pseudostate

S E C _ I O N B

Description:	provides ionization+excitation cross sections based on the pseudostate de-composition
Input files:	**target_ps, bound_ovl, zarm.omb_top**
Output files:	**sec_ion_nn** (nn – index of initial state)
Call as:	**sec_ion [is=… i16=… om=…]**
	is [1] - index of initial state
	i16 [0] - control the units for output cross sections
	om [zarm.om.top] – input omega file