11. The program BSR_PHOT (version 3)

The BSR_PHOT program runs the photoionization calculations for one partial wave and one initial states, based on the **d.nnn** and **h.nnn** files prepared by the programs BSR_DMAT and BSR_HD, respectively. The present implementation of the *R*-matrix method for photoionization calculations differs from the previous one in that the initial state may be calculated independently from the scattering solutions, with any amount of non-orthogonality between initial and final one-electron radial functions. We assume that the initial state is well confined in the inner region. The BSR_PHOT program was not designed for massive calculations, which would include many initial states and many final partial-wave channels. It was designed for careful investigation of the highly correlated cases, such as photoionization of neutral atoms or photodetachment of negative ions. It is also have tools for more detailed investigation of the near-threshold resonance structure.

The main bottleneck of the program is the using obsolete ASYPCK program (Crees 1980) to determine the solutions in the asymptotic region.

11.1. Related theory

In *R*-matrix theory, the photoionization cross-sections can be defined through the dipole matrix between the initial state Φ_0 and the *R*-matrix basis states Ψ_k, provided that all radial orbitals of the initial state are well confined to the inner region. The total photoionization cross section for a given photon energy ω is

$$\sigma(\omega) = (\tfrac{4}{3}\pi^2 a_0^2 \alpha) \left(\frac{\omega C}{2L_0 + 1} \right) \sum_j |(\Psi_j^- \,||\, \boldsymbol{D} \,||\, \Phi_0)|^2 \tag{11.1}$$

where $\boldsymbol{D}$ is a dipole operator. It could be either in the length or the velocity form, with $C = 1$ in the length form, and $C = 4/\omega^2$ in the velocity form, and the photon energy ω being in Ry. The index j runs over different R-matrix solutions, and the other quantities have their usual meaning. The solutions Ψ_j^- in (11.1) correspond to asymptotic conditions with a plane wave in the direction of the ejected electron momentum $\boldsymbol{k}$ and ingoing waves in all open channels. The corresponding radial functions $\boldsymbol{F}^-$ are related to the $\boldsymbol{F}(r)$ with the *K*-matrix asymptotic form (2.27) via

$$\boldsymbol{F}^- = -i\boldsymbol{F}(1 - i\boldsymbol{K})^{-1}. \tag{11.2}$$

Expanding Ψ_j^- in terms of the *R*-matrix states as in equation (2.13) and using the expressions (2.23), we find that

$$(\Psi_j^- \parallel D \parallel \Phi_0) = \frac{1}{a}\sum_k \frac{(\Psi_k \parallel D \parallel \Phi_0)}{E_k - E_0 - \omega} \boldsymbol{w}_k^T \boldsymbol{R}^{-1} \boldsymbol{F}_j^-(a) \qquad (11.3)$$

where $(\Psi_k \| \mathrm{D} \| \Phi_0)$ are reduced matrix elements between the initial state and the R-matrix basis functions.

In order to use the expression (2.31), we need the values of the solutions $F_i(a)$ at the R-matrix boundary. They can be obtained by matching the general asymptotic solutions of equation (2.25) to the solutions in the internal region at $r = a$. This can be done with relation (2.19), which in matrix form reads

$$\boldsymbol{F} = a\boldsymbol{RF}' - b\boldsymbol{RF} \qquad (r \le a) \qquad (11.4)$$

Note that to obtain dipole matrix elements (11.3) we really need the $\boldsymbol{R}^{-1}\boldsymbol{F}$ quantity, which can be defined from (11.4):

$$\mathbf{R}^{-1}\mathbf{F} = a\mathbf{RF}' - b\mathbf{RF} \qquad (r = a) \qquad (11.5)$$

In the outer region we have n_o independent physical solutions, where n_o is the number of open channels, which is defined by all the target states accessible at a given excitation energy. To relate the $n \times n$ dimensional R-matrix to the $n_\mathrm{o} \times n_\mathrm{o}$ K-matrix defined in Eq. (2.27), we introduce $n + n_\mathrm{o}$ linearly independent solutions $s_{ij}(r)$ and $c_{ij}(r)$ of Eq. (2.25) satisfying the boundary conditions

$$\left.\begin{array}{l} s_{ij}(r) \\ c_{ij}(r) \\ c_{ij}(r) \end{array}\right\} \underset{r\to\infty}{\sim} \left\{\begin{array}{lll} \sin\theta_i\delta_{ij} & i = 1, n & j = 1, n_o \\ \cos\theta_i\delta_{ij} & i = 1, n & j = 1, n_o \\ \exp(-\phi_i)\delta i,_{j-n_o} & i = 1, n & j = n_o + 1, n \end{array}\right. \qquad (11.6)$$

where θ_i and ϕ_i define the asymptotic phases in the open and closed channels, respectively:

$$\begin{aligned} \theta_i &= k_i r - \tfrac{1}{2} l_i \pi - \frac{z}{k_i}\ln 2k_i r + \arg\Gamma(l_i + 1 + i\frac{z}{k_i}) \\ \phi_i &= |k_i| r - \frac{z}{|k_i|}\ln(2|k_i| r) \end{aligned} \qquad (11.7)$$

Now we can rewrite the asymptotic form of scattering wavefunction in a general form

$$\boldsymbol{F} = \boldsymbol{s} + \boldsymbol{cK} \qquad (r \ge a)\,. \qquad (11.8)$$

Substituting this into Eq. (2.22) and solving for $\boldsymbol{K}$, we get

$$\boldsymbol{K} = \boldsymbol{B}^{-1}\boldsymbol{A} \qquad (11.9)$$

where

$$\boldsymbol{A} = -\boldsymbol{s} + a\boldsymbol{R}(\boldsymbol{s}' - \frac{b}{a}\boldsymbol{s}), \qquad \boldsymbol{B} = +\boldsymbol{c} - a\boldsymbol{R}(\boldsymbol{c}' - \frac{b}{a}\boldsymbol{c})\,. \qquad (11.10)$$

This completes the evaluation of the reactance matrix $\boldsymbol{K}$ and the value $\boldsymbol{F}(a)$, provided that the asymptotic solutions $s_{ij}(r)$ and $c_{ij}(r)$ are known. These solutions in the given version are obtained with the program ASYPCK .

The initial state Φ_0 in the present program can be obtained either in an independent MCHF calculation, or in the framework of *B*-spline bound-states calculations discussed in the section (2.xx). In general, it requires the evaluation of dipole matrix elements between states with non-orthogonal orbitals. Details of the calculation of dipole matrix elements in this case are presented in sections 9 (MULT) and 10 (BSR_DMAT).

11.2. Structure and data flow

The block diagram of the program BSR_PHOT, along with the data flow, is shown in Fig.11.1. All input/output files have prescribed internal names (see section 11.4 below). First, the program reads the input parameters from the file **bsr_phot.inp.** The input electron energies are given in Ry and defined in blocks with specific initial energy, energy, step and final energy. The user can define up to ten blocks for a given run that allows to calculate the cross sections with different 'intensity', i.e., some energy regions can be investigated in more detailed, with small energy increments. All output files are opened in the position mode APPEND, and hence all sequential results are added to the output without deleting results from previous calculations. This allows the user to accumulate data from different runs.

The radiative data for a given case are defined in the file **d.nnn**, together with the terms for the initial and final states, as well as the energy of the initial state. Based on the term of the final state, the program looks for the corresponding scattering and *R*-matrix data in the file **h.nnn**, which may contain data only for one partial waves. Finally, the program reads the internal solution weights from the file **w.nnn**. This optional input is guided by the input parameter **nwt** (see section 10.3). These data make it possible to define the channel composition of the final total wavefunction for a given energy, that may be used for detecting and classification of possible resonance structure. The program also check if exists the **bound.nnn** file, and then read the energies of all possible bound solutions in the inner-region with zero boundary conditions.

At the next stage the program executes fully independent calculations of the photoionization cross section for each input energy. The algorithm is given in section 11.2. In order to determine the photoionization cross section, we need the set of linearly independent solutions (11.8) in the external region. The determination of these solutions is beyond the scope of the present package, which deals with the internal region solutions in the general case. For the treatment of the external region, we currently choose the program ASYPCK (Cress 1980,1981). This is a general program, which can determine the solution for both neutral and ionized atoms. It works well over a wide range of energies, including near-threshold regions. ASYPCK is called as a subroutine, that is convenient for its using by another program.

The input data for the external region consist only of the description of the scattering channels and asymptotic coefficients (2.26). Only four main parameters, which are given in subsection 11.3, guide the execution of ASYPCK. For a more detailed description of the ASYPCK package, the user is referred to its write-up and that of the program IMPACT [36]. The ASYPCK program is called in the present calculation by using the interface subroutine **zaface** (see Fig. 11.1). We did not introduce any modifications to ASYPCK itself, and hence the user should check if the dimension parameters in ASYPCK are big enough for a given run. Over the past years, a set of new more effective algorithms were developed for the treatment of the external region, such as the flexible asymptotic *R*-matrix package FARM [28] or a new version of the program STGF [59]. In principle, ASYPCK can easily be replaced by any other algorithm, provided the latter is organized as a subroutine call.

The photoionization calculations for a given energy consist of the following steps. The routine **zopen** defines the number of open channels and the corresponding channels energies. Then routine **zaface** calls the program ASYPCK to determine the asymptotic radial functions (2.27) at the *R*-matrix radius. The routines **zrmat, zkmat** and **ephase** calculate the *R* and *K* matrices, together with the eigenphase sum at the energy of interest. Then the total dipole matrix elements (2.31), and the partial and total cross sections are calculated by the routine **phot_sec**. Optionally, the routine **ak_coef** is called if one needs the channel composition for total wave function.

The results are saved in a set of output files. The file **photo.nnn** contains the total cross section and the accuracy of the calculations, defined via the average asymmetry of the *K*-matrix. The results in this file are sorted according to energy by the routine **sort_photo**. The file **photo.nnn** is in simple table form and can be used for a quick check of the calculations. The full results including partial cross sections, eigenphase sums, channel weights and full dipole matrix elements (2.31) are recorded in the file **bsr_phot.nnn**. These data may then be used for the calculation of differential cross sections or the asymmetry parameter β (see description of utility-program **photo_tab** in section 13).

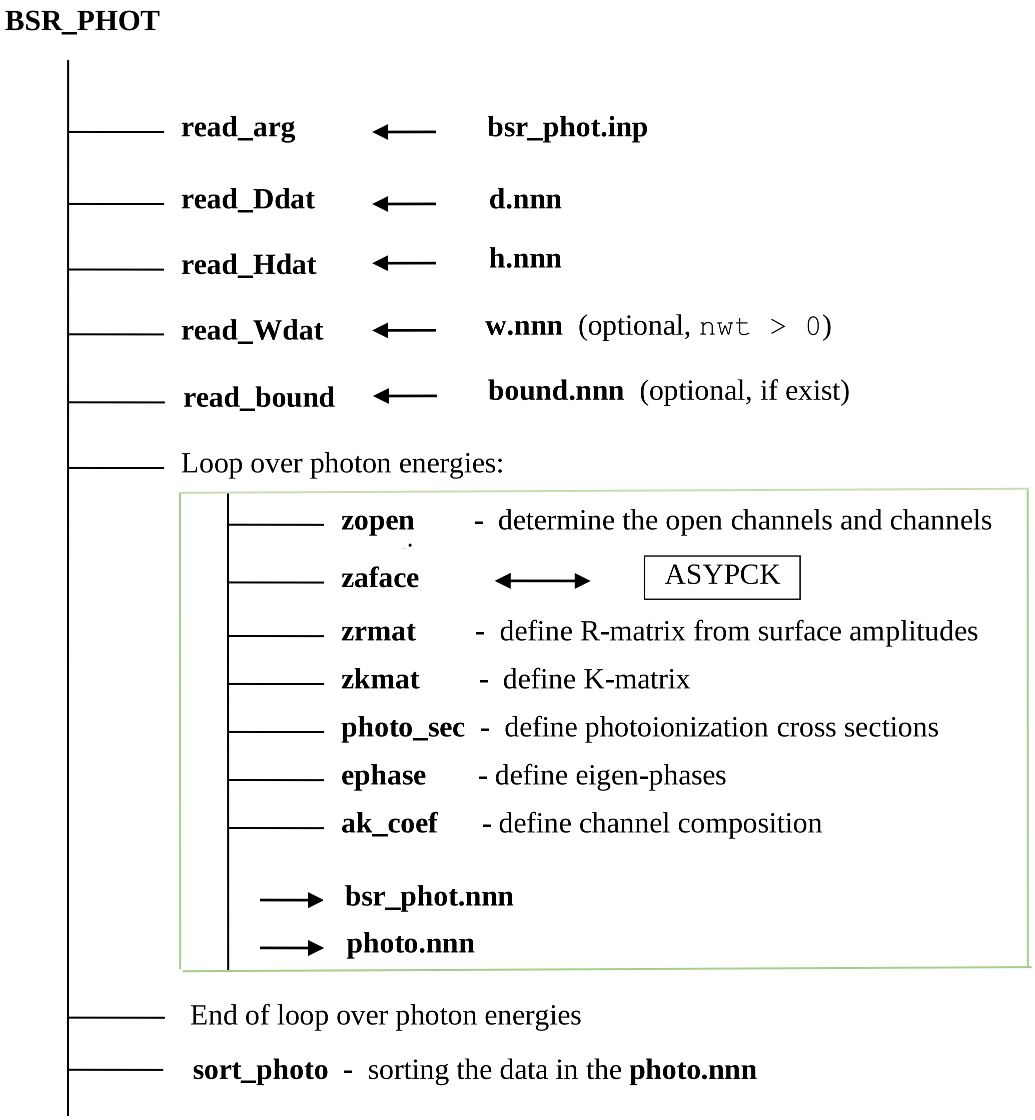
BSR_PHOT
read_arg ← bsr_phot.inp
read_Ddat ← d.nnn
read_Hdat ← h.nnn
read_Wdat ← w.nnn (optional, nwt > 0)
read_bound ← bound.nnn (optional, if exist)
Loop over photon energies:
zopen - determine the open channels and channels
zaface ↔ ASYPCK
zrmat - define R-matrix from surface amplitudes
zkmat - define K-matrix
photo_sec - define photoionization cross sections
ephase - define eigen-phases
ak_coef - define channel composition
→ bsr_phot.nnn
→ photo.nnn
End of loop over photon energies
sort_photo - sorting the data in the photo.nnn

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

11.3. Input parameters

The input parameters are provided in the file **bsr_phot.inp**. 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_phot**. Some of the input parameters deal with the asymptotic package ASYPCK. The parameters without default values are mandatory.

klsp index of partial wave (**nnn**)

elow
estep
ehigh Define the range of input electron energies (in Ry) as the initial energy (**elow**), the energy increment (**estep**), and the final energy (**ehigh**). This can be given for up to ten different input energy intervals.

awt [100] atomic number, used to convert Ry to eV.

ibug [0] defines the level of debug printing in the file **bsr_phot.out**.

e_exp [0.0] if ≠ 0, define the shift ionization (detachment) energy.

nwt [0] if > 0, then the weights of each channel in the total solutions are also calculated, based on the data in the file **w.dat.**

parameters for ASYPCK program:

AC [0.001] accuracy for the asymptotic solutions.

DR [0.1] in ASYPCK, the solutions are first obtained at R1=R-DR and R2=R+DR,. and then by interpolation - in R.

mfg [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.

11.4 Data files

bsr_phot.inp
File type: formatted sequential input.
Created by the user.
Read by the routine **read_arg**.
Description: input parameter for given run.

d.nnn
File type: unformatted sequential input.
Created by BSR_DMAT program.
Read by the routine **read_Ddat**.
Description: radiative-dipole transition vector for given initial state and final scattering partial wave **nnn**.

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**.**

w.nnn
File type: unformatted sequential input.
Written by the program BSR_HD**.**
Read by the routine **read_Wdat**.
Description: weights of different channels in the inner-region solutions.

photo.nnn
File type: formatted sequential output.
Created by the program BSR_PHOT**.**
Read by the user for quick estimation of results.
Description: electron and photon energies, total cross-section in the length and velocity forms, eigenphase and accuracy of the calculations, defined as the average asymmetry of the *K*-matrix.

bsr_phot.nnn
File type: formatted sequential output.
Created by program the BSR_PHOT**.**
Description: full information of the photoionization process, including channels cross sections, eigenphases, dipole matrix elements (2.31), channel weights and K-matrixes for each input energy.

bsr_phot.log
File type: formatted sequential output.
Created by the program BSR_PHOT**.**
Read by the user**.**
Description: running information.

11.5 Structure of the **bsr_phot.nnn** *files*

The **bsr_phot.nnn** file is the collection of the set of records, one for each photon energies:

1. `ek, eph, nopen, nwt, ikm`

   `ek` – energy of the photoionized electron in Ry (relative to the ion ground state)

   `eph` – photon energy in eV (relative to the initial atomic state)

   `nopen` – number of open channels

   `nwt` – flag for output of channel weights

   `ikm` – flag for output of K-matrix

2. `SLP, SL(1:nopen)` – total and channels cross sections in length form
3. `SVP, SV(1:nopen)` – total and channels cross sections in velocity form
4. `us, ui(1:nopen)` – eigenpahse sum and channels phases
5. `dlr(i),dli(i),dvr(i),dvi(i)` - dipole matrix elements in channel *i*.

   `repeat record 5 for i = 1, nopen`

`if nwt > nopen (weights of close channels, records 6-8):`

6. `CC` – total contribution of closed channels
7. `WTch(1:nwt)` – weights of closed channels, where WTch(1:nopen) - contribution of all closed channels in the solution j=1,nopen; WTch(nopen+1:nwt) - relative contribution of ith closed channel in all solutions.
8. `AK(1:nwt,i)` - weight of all channel in solution *i*

   `repeat record 8 for i = 1, nopen`

`if ikm > 0 (output of K-matrix):`

`9. ((KMAT(i,j),i=1,j),j=1,nopen)`

The channel weights (which are supposed to employ for classification of resonances) and K-matrixes are rarely used and often are skipped from output. That may save big memory on the disks. Drawback of output is that nor initial state, nor partial wave index is recoded. User should keep record by himself. We may added in the file name the label describing the initial state, like as **bsr_phot.001_5s.** The most convenient way tuned out to run one particular case in the separate folder, and then to combine the results if needed. It also allowed to run in parallel a few programs in parallel.

The **photo.nnn** file provides only the total partial cross sections, which allows the user quickly analyze the results and decide if additional energy points are needed to resolve all possible structures. The **bsr_phot.nnn** files provide complete information about the photoionization process, which can be used for generations different parameters and tables, see the description of the **photo_tab** utility below.

11.6. The MPI version

The MPI version BSR_PHOT_MPI has the same logical structure as the serial program, and the parallelization is provided in trivial manner: each energy is calculated on separate processor. The routines which run in parallel on different processors are marked in Fig.11.1 by green square.

Each processor records the results in separate files **photo.nnnn** and **bsr_phot.nnnn** files, where **nnnn** is the processor index. In the end of calculations, the data from these files are collected in the **photo.nnn** and **bsr_phot.nnn** files as in the serial program, and the above intermediate files are deleted. If happened that the program is terminated for some reasons, we may re-run program with parameter **ibug** = -1. Then the program will just collect the results from the existing **photo.nnnn** and **bsr_phot.nnnn** files, which have not-corrupted data.

11.7 The further development

The further development is very desirable by replacing the obsolete ASYPCK program for finding the asymptotic solutions. One of the possibility, it is use of the STGF program.

*11.8 The **photo_tab** utility*

Description:	produces the tables for channel or total cross sections and asymmetry parameters
Input files:	**bsr_phot.nnn, target**
Output files:	indicated by user
Call:	**photo_tab** with interactive response

This utility serves for processing the data recorded during photoionization calculations with program BSR_PHOT. The program BSR_PHOT can be run for different energy intervals and different partial waves separately. The results are accumulated in files **bsr_phot.nnn**. Then final tables for channel or total cross sections and asymmetry parameters may be generated with utility **photo_tab**.

The **photo_tab** utility has a set of different options for output:

1 - total cross sections
2 - channel cross sections
3 - ionic-state cross sections
4 - beta-parameters
5 - dipole matrix elements for given channel
6 - delete points (provide **delete_list**)
7 - check archives
8 - clean archives (provide **energy_list**)
9 - dipole matrix elements for range of channels
10 - eigenphases and its derivatives

The **bsr_phot** utility is working interactively. It first outputs the list of available options and as to choose one. Then the program may ask additional parameters and the file name for output. It includes, e.g., the list of partial waves included into consideration, the range of channels or ionic target states of interest. The most of options are clear and self-explained. The program allows to clean the **bsr_phot.nnn** (which can be very big) from not needed energy points indicated in the **delete_list** file. The special interest is the calculation of anisotropy parameter and eigenphases and its derivatives. The related formula are discussed below.

Anisotropy parameters

An anisotropy parameter β defines the angular distribution of photoelectrons. For example, for linearly polarized incident radiation the angular distribution of photoelectrons is given by

$$\frac{d\sigma}{d\hat{k}} = \frac{\sigma}{4\pi}[1+\beta P_2(\cos\theta)] \ , \qquad (11.11)$$

where *θ* is the angle of the ejected electron relative to the axis of polarization, while for unpolarized radiation it takes the form

$$\frac{d\sigma}{d\hat{k}} = \frac{\sigma}{4\pi}[1 - \frac{\beta}{2} P_2(\cos\theta)] \ , \qquad (11.12)$$

where *θ* is the angle of the ejected electron relative to the incident radiation beam. The computation of parameters depends on the coupling scheme and define by following formulas (presemted by A. Grum-Grimailo and E. Grizlova, private communication)

LS-case:

$$\beta = \frac{\sqrt{30}(-1)^{L_f+L_0}}{\sum_j |(\Psi_j^- \| D \| \Phi_0)|^2} \sum_{jj'} (-1)^{L-L'} i^{l-l'} e^{-i\sigma_l + i\sigma_{l'}} [(2l+1)(2l'+1)(2L+1)(2L'+1)]^{1/2}$$
$$\times \begin{pmatrix} l & l' & 2 \\ 0 & 0 & 0 \end{pmatrix} \begin{Bmatrix} L & L' & 2 \\ l' & l & L_f \end{Bmatrix} \begin{Bmatrix} L & L' & 2 \\ 1 & 1 & L_0 \end{Bmatrix} (\Phi_0 || D || \Psi_{j'}^-)(\Phi_0 || D || \Psi_j^-) \qquad (11.13)$$

JK-case:

$$\beta = \frac{\sqrt{30}(-1)^{J_f+J_0-\frac{1}{2}}}{\sum_j |(\Psi_j^- \| D \| \Phi_0)|^2} \sum_{jj'} i^{l-l'} e^{-i\sigma_l + i\sigma_{l'}} [(2l+1)(2l'+1)(2K+1)(2K'+1)(2J+1)(2J'+1)]^{1/2}$$
$$\times \begin{pmatrix} l & l' & 2 \\ 0 & 0 & 0 \end{pmatrix} \begin{Bmatrix} K & K' & 2 \\ l' & l & J_f \end{Bmatrix} \begin{Bmatrix} K & K' & 2 \\ J' & J & \frac{1}{2} \end{Bmatrix} \begin{Bmatrix} J & J' & 2 \\ 1 & 1 & J_0 \end{Bmatrix} (\Phi_0 || D || \Psi_{j'}^-)(\Phi_0 || D || \Psi_j^-) \qquad (11.14)$$

JJ-case:

$$\beta = \frac{\sqrt{30}(-1)^{J_f-J_0-\frac{1}{2}}}{\sum_j |(\Psi_j^- \| D \| \Phi_0)|^2} \sum_{jj'} (-1)^{J-J'} i^{l-l'} e^{-i\sigma_l + i\sigma_{l'}} [(2l+1)(2l'+1)(2j+1)(2j'+1)(2J+1)(2J'+1)]^{1/2}$$
$$\times \begin{pmatrix} l & l' & 2 \\ 0 & 0 & 0 \end{pmatrix} \begin{Bmatrix} j & j' & 2 \\ l' & l & J_f \end{Bmatrix} \begin{Bmatrix} j & j' & 2 \\ J' & J & \frac{1}{2} \end{Bmatrix} \begin{Bmatrix} J & J' & 2 \\ 1 & 1 & J_0 \end{Bmatrix} (\Phi_0 || D || \Psi_{j'}^-)(\Phi_0 || D || \Psi_j^-) \qquad (11.15)$$

Note the order of initial and final states in the *D*-matrix elements. The change of these orders may introduce of additional phase factors.

Eigenphases

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.