Manual for MQDT post process

Rui Jin

September 12, 2024

1 Theory background

Following the manual for MQDT solver,the discrete levels can be obtained by searching the cross point of

$$F(x_i, U_{i\alpha}, \mu_\alpha) = \det[U_{i\alpha} \sin \pi(x_i + \mu_\alpha)] = 0, \tag{1a}$$

$$\nu_y(\nu_x) = \frac{q}{\left[I_y - I_x + q^2/\nu_x^2\right]^{1/2}} \tag{1b}$$

where,

$$x_i = \begin{cases} \nu_i & \text{for } i \in \text{closed channels}, \\ -\tau & \text{for } i \in \text{open channels}, \end{cases}$$

For the autoionization region, the center of resonances are at local maximum of the energy derivative of phase shift,

$$\frac{d\tau}{dE} = \frac{\nu_x^3}{2q^2} \frac{d\tau}{d\nu_x}. \tag{2}$$

the local maximum can be found by searching for root of the derivative of expression (**??**).

$$\frac{d}{d\nu_x} \left(\frac{\nu_x^3}{2q^2} \frac{d\tau}{d\nu_x} \right) |_{\nu_{res}} = 0. \tag{3}$$

2 Program structure

```
constants     # global constants and parameters
type_def      # user defined complex data structure for easier data communication
numerical     # frequently used numerical algorithms
file_cmdl_io  # file i/o and commandline parser
stdio         # frequently used string operations and block output.
envset        # seting up machine-dependent systematic parameters (EPS, PI, Nan ...)
interpolate   # a set of interpolation functions, employed in root finding algorithms
   .
darray        # dynamical array for easier data storage
search_root   # search levels and resonances and create readable and easy to plot
   files, if provide energy levels, generate a comparison file.
```

3 Numerical algorithms

(i) discrete level finding

- First roughly search for cross
  Scan x axis for ν_x^f where the sign of the difference of $G(\nu_x) = F(x_i, U_{i\alpha}, \mu_\alpha) - \nu_y(\nu_x)$ flips.
- Finer search
  Choose a suitable vicinity of ν_x^f, find the root of the difference function $G(\nu_x)$. Where the actual value of $F(x_i, U_{i\alpha}, \mu_\alpha)$ and $\nu_y(\nu_x)$ are calculated on-the-fly ($F(x_i, U_{i\alpha}, \mu_\alpha)$ is obtained by spline-interpolation).

(ii) resonance finding

- First rought search
  The Local peaks are $\frac{d}{dE}\tau(x_{i-1}) <= \frac{d}{dE}\tau(x_i) <= \frac{d}{dE}\tau(x_{i+1})$. There might be small fluctuations of the derivative due to numerical operations(especially two-point interpolation is used). So we need to filter out some of them.
- Noise filter
  noise filter threshold is automatically determined based on the widest resonance width $\max\{\delta_\alpha\}$. The widths are estimated by the imaginary part of projected K_{cc} matrix. (J. M. Lecomte, J. Phys. B: At. Mol. Phys. 20 (1987) 3645-3662.)
  Or you can manually specify one by adding option [-*filter XXX*] to the command line.
- Finer search
  Employ bisect searching algorithm to search for the derivative of $\frac{d}{dE}\tau$ function in a suitable vicinity. The function and it's derivative are Spline-interpolated on-the-fly.
- Eigenchannel sort

4 Program input

Required input files:

```
miuang.out     # S matrix prepared by smooth-toolkits
[dalfa.in]     # dipole matrix moment, if provided, automatically calculate oscillator
    strength (density)
mqdt.in        # MQDT solver control file
Anorm.out      # Normalized A-coefficients
Ara.out        # Unnormalized A-coefficients
Dn.out         # Normalization factor N for A-coefficients.
[NormDos.out]  # Density of States if discrete calculation
[elev.exp]     # experimental levels(resonances) to compare. (If provided, the program
    will generate comparison files automatically.)
```

The major control file *mqdt.in* is the same as that for mqdt solver. You can add blank lines and # to comment. You can also use {} to wrap the data block, you can put them in one line or break the wrapper into multiple lines.

Table 1: Options in *mqdt.in*

`Z`	target charge
`nchan`	number of physical channels (optional, program will determine it from S-matrix input file)
`nop`	number of open channels (optional, program will determine it from the energy region and the choice of x-axis threshold `IPx`.)
`nip`	number of ionization potentials (IP), (optional, program will determine from the IP array block `IP = {....}`).
`IP_unit`	optional unit for IP, cm-1, ryd, au, a.u., ev, Hartree are accepted (case-insensitive). It will be treated as cm-1 if not specified
`IP`	IP array block `IP = {....}`, the `{}` wrapper accept `,` and space to divide strings.
`IP_seq`	indices to assign IP to each physical channel.
`IPy`	threshold to present effective quantum defect/phase-shift. channels associated to threshold $IP \leq IP_y$ will be treated opened. You must specify it for bound-state calculation. It will be determined internally in continuum calculation.
`IPx`	threshold to present energy. channels associated to threshold $IP \geq IP_x$ will be treated closed. You must specify it for bound-state calculation. It will be determined internally in continuum calculation.
`xrange`	optional ν_x with respect to IP_x. You must either specify `xrange` or `E_continuum`.
`E_continuum`	optional continuum electron energy with respect to the first IP I_1. You can specify unit for the energy range string e.g. `E_continuum = {0.0:0.20_ryd}`. Accepted energy units are IP, cm-1, ryd, au, a.u., ev, Hartree (case-insensitive). The unit is cm-1 by default.
`x_grid_type`	Options: mu-eigen / mu-proj / uniform
`y_grid_type`	Options: adaptive / uniform
`nx_flat`	number of grid per X-grid segment for the flat region.
`nx_spike`	number of grid per X-grid segment for the sharp peaks.
`ny_init_guess`	number of grid per Y-grid segment in the initiative guess algorithm
`ny_adapt`	number of grid per Y-grid segment in the adaptive algorithm
`x_fine`	optional. Manually refine the X-grid. You can specify several sections by `{}` wrapper. For example `x_fine = {2.19:2.21%1000, 2.29:2.31%1000}`, where `%1000` means you want to use 1000 times finer grid for this section.
`y_fine`	optional. Manually refine the Y-grid. Usage the same as `x_fine`.
`eqnsolv_method`	How to solve the MQDT equation. Options are: hyb, bisect, newton.
`relax`	optional. Relaxation coefficient to prevent oscillation. Default 1.0.
`dmu`	optional. Constant correction to μ_α. Specify it with `{}` wrapper: `dmu = {0.0 0.0 0.0 0.0 0.0 0.0}`
`au_peak_search_method`	For postmqdt. Either search the steepest change of each collisional-channel phase shift or the total phase shift. Options: tot / chan.
`k_mat_cofact`	the column index for cofactor when solving the A-coefficients from the secular equations (asymptotic boundary condition). $\forall i \leq nchan$ will work, this is only left for debugging purpose.
`twoJ`	Not used currently. $2j$ of the system, used to calculate Lande-g factor.