Manual for MQDT solver

Rui Jin

September 12, 2024

1 Theory background

The energy eigen wave function for a atomic excited complex should be a superposition of the eigenchannel wavefunctions,

$$\Psi^{J\pi} = \sum_{\alpha}^{N_p} A_\alpha \psi_\alpha^{J\pi}(E), \tag{1}$$

where N_p is the number of physical channels.

The asymptotic boundary condition (for bound and continuum) leads to a set of secular equations,

$$\begin{cases} \sum_\alpha U_{i\alpha} \sin\pi(\nu_i + \mu_\alpha) A_\alpha^\rho = 0 & \text{for } i \in \text{closed channels}, \\ \sum_\alpha U_{i\alpha} \sin\pi(-\tau^\rho + \mu_\alpha) A_\alpha^\rho = 0 & \text{for } i \in \text{open channels}. \end{cases} \tag{2}$$

The existence of non-trivial solution A_α^ρ in eqns. (2) requires the determinant of coefficients matrix to vanish, which gives rise to the multichannel quantum defect theory (MQDT) eqns. (3a):

$$F(x_i, U_{i\alpha}, \mu_\alpha) = \det[U_{i\alpha} \sin\pi(x_i + \mu_\alpha)] = 0, \tag{3a}$$

where,

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

and the effective principle quantum number are defined by,

$$E = I_i - q^2/\nu_i^2. \tag{3b}$$

We can employ the graphic solution method to solve the MQDT equation in autoionization region naturally. However, it's beneficial to also treat the bound energy region in the similar fashion(i.e., Lu-Fano plot) by the fact that the phase shift can be smoothly connected with the quantum defect number by the analytical continuation property of S matrix.

There are situations where we need to treat several channels, that corresponds to more than one thresholds, as openned ones (denote as $\{I_o\}$). However, due to the limitation of 2D illustration of the Lu-Fano type plot, only one energy relation condition line can be presented to find the physical discrete levels. In such situations, we need to project these channels onto one of these thresholds (e.g. an arbitrary one $I_y \in \{I_o\}$), via the following additional constrains,

$$-\tau_i = \begin{cases} -\tau_i & i = y, \\ -\tau_y + \Delta_{i,y} & i \neq y, \end{cases} \tag{4a}$$

where,

$$\Delta_{i,y} = \nu_i - \nu_y = \left[\frac{1}{\nu_x^2} - \frac{I_i - I_x}{q^2}\right]^{-1/2} - \left[\frac{1}{\nu_x^2} - \frac{I_y - I_x}{q^2}\right]^{-1/2}. \tag{4b}$$

As we can see constrains (4) varies smoothly and fulfills the Rydberg energy relation at the physical discrete levels.

The mixing coefficients can be easily shown to be proportional to the cofactor $C_{i\alpha}$ of $F(x_i, U_{i\alpha}, \mu_\alpha)$ by the analogy of the secular eqns (2) and the definition of determinant.

$$A_\alpha = C_{i\alpha}/(\sum_\alpha C_{i\alpha}^2)^{1/2}, \quad \forall i \leq N_p \tag{5}$$

2 Program structure

```
constants      # global constants and parameters
type_def       # user defined complex data structure for easier data communication
numerical      # frequently used numerical algorithms
darray         # dynamical array for easier data storage
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 ...)
gengrid        # generating X and Y grids.
SOLVMQDT       # solving the MQDT equations
```

3 Numerical algorithms

(i) X-grid

- uniform
- divide X-axis into several segments based on the eigenchannel quantum defect $\mu_\alpha \pm \delta_\alpha/2$.
- δ_α is the resonance width estimated by the imaginary part of projected K_{cc} matrix. (J. M. Lecomte, J. Phys. B: At. Mol. Phys. 20 (1987) 3645-3662.)
- (Optional) manually set x-segments and increase grid density.
- (Not useful) alternatively, divide X-axis into several segments based on effective eigenchannel quantum defect $\tilde{\mu}_\alpha \pm \delta_\alpha/2$. Where the effective eigenchannel quantum defect $\tilde{\mu}_\alpha$ is obtained by diagonalizing the real part of projected K_{cc} matrix.(J. M. Lecomte, J. Phys. B: At. Mol. Phys. 20 (1987) 3645-3662.). But it seems working poorly in the test.

(ii) Y-grid

- For the first x step $\nu_x^{(0)}$, initiate the Y-grid segments with $\mu_\alpha(\alpha \in$ open channels). Use rather dense y grid to search.
- Second x step $\nu_x^{(1)}$: update Y-grid segments with the results first step. Only a few grid per segment is needed from now on.
- $n > 2$ x step: update Y-grid segments with the extrapolation of the result of $n-1'th$ and $n-2'th$ x step.
- Be careful with sudden jump due to modulo 1.
- (Optional)manually set y-segments and increase grid density.

(iii) equation solver

- Newton-Ralphson.
- Bisect search.
- Hybrid of Newton and bisect updating formula.
- A relaxation coefficient can be used to slow down the oscillation.
- For autoionization(continuum) region, Greene's generalized eigenvalue technique is now implemented too improve efficiency. Caveate could be that now I am using Lapack routine, so only double precision is possible, which only allows for not so high dimension problems (where the efficiency of the original root finding version is tolerable).

4 Program input

Required input files:

```
miuang.out      # S matrix prepared by smooth-toolkits
mqdt.in         # MQDT solver control file
dalfa.in        # dipole matrix elements (optional)
```

The major control file *mqdt.in*. The input file is case-insensitive. 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.

Below is a sample input file.

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.