Computer Programs in Physics

mqdtfit: A collection of Python functions for empirical multichannel quantum defect calculations ☆,☆☆

R.M. Potvliege

Physics Department, Durham University, South Road, Durham DH1 3LE, UK

ARTICLE INFO

ABSTRACT

The Python functions distributed with this article can be used for calculating the parameters of multichannel quantum defect theory models describing excited bound states of complex atoms. These parameters are obtained by fitting a model to experimental data provided by the user. The two main formulations of the theory are supported, namely the one in which the parameters of the model are a set of eigen channel quantum defects and a transformation matrix, and the one where these parameters are the elements of a reactance matrix. The distribution includes programs for calculating theoretical energy levels, calculating mixing coefficients and channel fractions and producing Lu-Fano plots.

Program summary
Program Title: mqdtfit
CPC Library link to program files: https://doi.org/10.17632/nzgygzd96c.1
Developer's repository link: https://github.com/durham-qlm/mqdtfit
Licensing provisions: BSD 3-clause.
Programming language: Python 3.
Nature of problem: Multichannel quantum defect theory aims at giving a unified description of the excited states of multielectron systems whereby the properties of entire series can be predicted from models involving a relatively small number of parameters. These parameters are obtained by fitting theory to experimental data, in the empirical version of the theory. The present programs specifically concern the application of this theory to the bound states of complex atomic systems, such as divalent atoms, for which spectroscopic series are often perturbed by isolated states of a different symmetry, making a multichannel description necessary.
Solution method: The functions forming this library are grouped and linked to each other into a single Python module. They can be used to calculate the parameters of a given multichannel quantum defect theory model by fitting the model to experimental data provided by the user. These parameters are either a set of eigen channel quantum defects and a transformation matrix, in the eigenchannel parametrization of the theory, or the elements of an orthogonal matrix, in the reactance matrix parametrization of the theory. Given these parameters, this software can also be used to calculate theoretical energy levels, calculate coefficients describing the mixing between the channels considered and produce Lu-Fano plots.

1. Introduction

Multichannel quantum defect theory (MQDT) aims at giving a unified description of the excited states of multielectron systems composed of a single outer electron and a more compact electronic core. This theory makes it possible to understand the properties of entire series of states from models involving a relatively small number of parameters. These parameters are obtained by fitting theory to experiment in the empirical approach we are concerned with, but they may also be calculated from first principles [1].

MQDT has a long history, starting in the early days of Quantum Mechanics [2]. This early work has been considerably extended to mul-

tichannel cases, initially by Seaton in the 1960s [2,3], and later on by a number of other authors. Fano, in particular, pointed out the relevance of the "eigenchannel" formulation of the theory in a seminal study of the spectrum of molecular hydrogen published in 1970 [4]. This study and a contemporary analysis of atomic spectra by Lu and Fano [5] were followed shortly afterwards by further pioneering work, notably by Lu [6], and Lee and Lu [7]. Apart from more recent extensions, such as the inclusion of various observables other than bound state energies [8], the general approach to obtaining empirical parameters of MQDT models of complex atoms has changed little over the years — see, e.g., Refs. [9–12] for recent examples. Work in that area has often been based on the eigenchannel formulation of MQDT rather on Seaton's original formulation in terms of a K (or reactance) matrix. However, the latter is also in use [10,11,13–17]. The calculation yields theoretical energies and mixing coefficients (the Z_i coefficients defined below), which in turn can be used for calculating spontaneous lifetimes and dispersion coefficients — e.g., [10,11,14–16,18].

To the author's knowledge, only one open source, documented, publicly available computer program for empirical MQDT calculations has been published so far, despite the wide relevance of this approach [19]. While general in scope, this program does not cover the K-matrix formulation of the theory. Moreover, the language in which it is written, Fortran 77, may make it impractical to present-day users. The present publication aims at filling this gap.

Specifically, its purpose is to offer easy-to-use functions for the construction of empirical multichannel quantum defect models of complex atoms. In its present state of development, the code is restricted to models in which all channels are closed, i.e., to bound states. It provides the possibility of fitting the model parameters to a set of experimental data, whether these parameters are expressed in terms of eigen channel quantum defects and a transformation matrix or in terms of a reactance matrix. The module also includes functions calculating energy levels and mixing coefficients and functions plotting channel fractions or Lu-Fano plots, given a set of model parameters. The programming is kept straightforward so as to facilitate customisation and future extensions. Revised versions of the code and new related material will be made available from the URL https://github.com/durham-qlm/mqdtfit (a GitHub repository of the Quantum, Light and Matter research group of Durham University).

The principles of empirical MQDT and the applications of this approach have been reviewed in a number of authoritative articles [1,2,8,20,21], however, without any being sufficiently comprehensive and succinct for serving as a convenient reference for the programs described in this document. These details are reviewed in Section 2, for completeness and clarity. Practical advice about fitting MQDT models to experimental spectra can be found, e.g., in Refs. [6,7,19,22]. The numerical methods used in this implementation of MQDT theory and other specific details are outlined in Section 3. Further information about the use of this module can be found in Section 4. A more detailed user manual is joined with the distribution, as are example programs and their output. The latter are briefly described in Section 5.

2. Multichannel quantum defect theory of bound systems

Multichannel quantum defect theory concerns systems in which several different configurations of the core need to be considered, hence different configurations of the (core - outer electron) system. Each of these configurations is referred to as a channel. We are primarily interested by the dissociation channels, which are characterised by the angular momentum of the outer electron and by the state the core would be in if this electron was infinitely far away from it (dissociation channels are also called scattering channels). We will assume that we need to consider N_{ch} such channels and denote by $\mathfrak{I}_i$ the ionisation limit of channel i ($i = 1, \ldots, N_{ch}$): in a state of energy $\mathfrak{E}$, channel i is open or closed according to whether $\mathfrak{E} > \mathfrak{I}_i$ or $\mathfrak{E} < \mathfrak{I}_i$. We reserve

the symbols E and I_i for the corresponding energies expressed as wave numbers:

$$E \equiv \mathfrak{E}/hc, \qquad I_i \equiv \mathfrak{I}_i/hc, \tag{1}$$

where h is Planck's constant.

The programs described here specifically address the case where all dissociation channels are closed. The theory is based on two ansätze, namely

$$\Psi(X, r) = \mathcal{A} \sum_i C_i \chi_i(X) \phi_i(r) \tag{2}$$

and

$$\Psi(X, r) = \mathcal{A} \sum_\alpha A_\alpha \left[\mathcal{F}_\alpha(X, r) \cos(\pi \mu_\alpha) - \mathcal{G}_\alpha(X, r) \sin(\pi \mu_\alpha) \right], \tag{3}$$

where r denotes the radial coordinate of the outer electron, X represents all the other relevant coordinates and $\mathcal{A}$ is the antisymmetrization operator. These two equations both express the wave function of an energy eigenstate of interest as an antisymmetrized linear combination of product wave functions; however, they do so in different ways: Eq. (2) expresses the wave function directly in terms of the individual dissociation channels, whereas Eq. (3) expresses it in terms of configurations better suited to describing the inner region where all the electrons are close to each other and to the nucleus. This other form of the wave function appears naturally in calculations based on the diagonalisation of the (core + outer electron) Hamiltonian in this region. The α-configurations are usually referred to as the eigenchannels and the constants μ_α as the corresponding eigen quantum defects.

In Eq. (2), the C_i's are constant coefficients describing the admixture of the different dissociation channels in the state Ψ, and, as implied by the notation, the ϕ_i's are functions depending only on the radial coordinate of the outer electron while the χ_i's are functions of the angular coordinates of the outer electron, of the coordinates of the other electrons and of the spin of all the electrons. Making the approximation that the outer electron moves in a Coulomb field when outside the core leads to the following equation, in the case of a neutral atom:

$$\left[-\frac{\hbar^2}{2\mu} \left(\frac{d^2}{dr^2} - \frac{l_i(l_i + 1)}{r^2} \right) - \frac{e^2}{4\pi \epsilon_0 r} \right] u_i(r) = \epsilon_i u_i(r), \tag{4}$$

where $u_i(r) = r\phi_i(r)$, μ is the reduced mass, e is the charge of the electron and ϵ_i is an energy measured relative to $\mathfrak{I}_i$. For a state of energy $\mathfrak{E}$,

$$\epsilon_i = \mathfrak{E} - \mathfrak{I}_i = (E - I_i)\,hc. \tag{5}$$

The general solution of Eq. (4) can be expressed in terms of a linear combination of regular and irregular Coulomb functions,[1] $f(\epsilon, l; r)$ and $g(\epsilon, l; r)$. We can thus write

$$u_i(r) = f(\epsilon_i, l_i; r) \cos \pi \nu_i + g(\epsilon_i, l_i; r) \sin \pi \nu_i, \tag{6}$$

where ν_i is a constant.

In Eq. (3), on the other hand, the A_α's are constant coefficients whereas the $\mathcal{F}_\alpha$'s and $\mathcal{G}_\alpha$'s are functions of the coordinates and spin of all the electrons: the $\mathcal{F}_\alpha$'s are assumed to reduce to linear combinations of the functions $\chi_i(X) f(\epsilon_i, l_i; r)/r$ and the $\mathcal{G}_\alpha$'s to linear combinations of the functions $\chi_i(X) g(\epsilon_i, l_i; r)/r$ when the outer electron is far from the core. Specifically, we assume that, for sufficiently large values of r,

---

[1] We take these functions to be the f and g functions defined, e.g., in Refs. [4], [23] and [24], not the f and g functions defined by Seaton [2,3]. In the definition adopted here, these functions are identical to the functions denoted $\mathfrak{f}$ and $\mathfrak{g}$ by Lee and Lu [7] and by Dehmer and Fano [25]. They are almost identical to Seaton's functions s and c [2,3], which are normalised differently: $f \equiv \sqrt{2}s$ and $g \equiv -\sqrt{2}c$.

$$\mathcal{F}_\alpha(X,r) = \sum_i U_{i\alpha}\, \chi_i(X)\, f(\epsilon_i, l_i; r)/r, \tag{7}$$

$$\mathcal{G}_\alpha(X,r) = \sum_i U_{i\alpha}\, \chi_i(X),\, g(\epsilon_i, l_i; r)/r, \tag{8}$$

where the coefficients $U_{i\alpha}$ are real and do not depend on electron coordinates.

The values of ν_i, C_i, μ_α and A_α for which these two ansätze yield physically admissible bound state wave functions are restricted by three requirements, namely (1) that $\phi_i(r)$ goes to 0 for $r \to \infty$; (2) that Eqs. (2) and (3) apply not only to the bound states of interest but also to the states with energy above the lowest ionisation threshold, for which some or all the dissociation channels are open; and (3) that

$$\sum_i C_i \chi_i(X)\phi_i(r) \equiv \sum_\alpha A_\alpha \left[\mathcal{F}_\alpha(X,r)\cos(\pi\mu_\alpha) - \mathcal{G}_\alpha(X,r)\sin(\pi\mu_\alpha)\right] \tag{9}$$

when r is sufficiently large.

The first of these requirements implies that the ν_i's are necessarily related to the ϵ_i's by the equation [2]

$$\epsilon_i = -\frac{\mu e^4}{(4\pi\epsilon_0)^2\hbar^2}\frac{1}{2\nu_i^2}, \quad i = 1,\dots,N_{\text{ch}}. \tag{10}$$

(By analogy with the case of atomic hydrogen, the ν_i's are often referred to as effective principal quantum numbers.) In terms of energies expressed as wave numbers,

$$E - I_i = -\tilde{R}/\nu_i^2, \quad i = 1,\dots,N_{\text{ch}}, \tag{11}$$

where $\tilde{R}$ is the mass-corrected Rydberg constant for the species considered.

The second requirement implies, as an indirect consequence of the unitarity of the scattering matrix, that the square matrix $[U_{i\alpha}]$ of elements $U_{i\alpha}$ is orthogonal — see, e.g., Refs. [2], [6] or [21].

The last requirement has several consequences, as is explained in Appendix A. One finds, in particular, that the coefficients C_i and A_α satisfy the following systems of equations:

$$\sum_i U_{i\alpha} \sin[\pi(\nu_i + \mu_\alpha)]C_i = 0, \quad \alpha = 1,\dots,N_{\text{ch}}, \tag{12}$$

$$\sum_\alpha U_{i\alpha} \sin[\pi(\nu_i + \mu_\alpha)]A_\alpha = 0, \quad i = 1,\dots,N_{\text{ch}}. \tag{13}$$

Moreover,

$$A_\alpha = \sum_i U_{i\alpha} \cos[\pi(\nu_i + \mu_\alpha)]C_i, \quad \alpha = 1,\dots,N_{\text{ch}} \tag{14}$$

and

$$C_i = \sum_\alpha U_{i\alpha} \cos[\pi(\nu_i + \mu_\alpha)]A_\alpha, \quad i = 1,\dots,N_{\text{ch}}. \tag{15}$$

It is necessary, for Eqs. (12) and (13) to have non-trivial solutions, that

$$\det |U_{i\alpha} \sin[\pi(\nu_i + \mu_\alpha)]| = 0. \tag{16}$$

Given a $[U_{i\alpha}]$ matrix and a set of eigen quantum defects μ_α, this last equation determines the ν_i's, thus the bound state energies.

Alternatively (see Appendix A), the coefficients C_i can also be obtained by solving a related system of linear equations, namely

$$\sum_j \left[K_{ij} + \delta_{ij}\tan(\pi\nu_i)\right]c_j = 0, \quad i = 1,\dots,N_{\text{ch}}, \tag{17}$$

where the K_{ij} coefficients are defined by the matrix equation

$$[K_{ij}] = [U_{i\alpha}][\delta_{\alpha\alpha'}\tan(\pi\mu_\alpha)][U_{j\alpha'}]^\dagger \tag{18}$$

and the c_j's are related to the C_j's by the equation

$$C_j = c_j / \cos(\pi\nu_j). \tag{19}$$

The system (17) admits a non-trivial solution under the condition that

$$\det |K_{ij} + \delta_{ij}\tan(\pi\nu_i)| = 0. \tag{20}$$

This last equation replaces the condition (16) determining the possible values of the ν_i's, in this alternative formulation. The bound states energies are then parametrized by the matrix $[K_{ij}]$ rather than by the matrix $[U_{i\alpha}]$ and the eigen quantum defects μ_α. The two formulations are equivalent as long as Eq. (18) holds. By analogy with similar equations appearing in scattering theory, the matrix $[K_{ij}]$ is often called the reactance (or reaction) matrix. As can be seen from Eq. (18), this matrix is symmetric ($K_{ij} = K_{ji}$), its eigenvalues are $\tan(\pi\mu_\alpha)$, $\alpha = 1,\dots,N_{\text{ch}}$, and its eigenvectors are the corresponding columns of the $[U_{i\alpha}]$ matrix.

The ionization thresholds I_i, the $[K_{ij}]$ matrix, the $[U_{i\alpha}]$ matrix and the eigen quantum defects can, in principle, be calculated from first principles. However, in the empirical approach we follow here, the I_i's are derived directly from measurements and the other parameters are obtained by fitting the theoretical energies to spectroscopic data, assuming that these quantities vary slowly with E. Either the K_{ij}'s or the $U_{i\alpha}$'s and μ_α's are directly obtained by fitting, giving two alternative formulations of empirical MQDT.

It can be noted that the functions $u_i(r)$ of Eq. (6) are proportional to exponentially decreasing Whittaker functions when Eq. (10) is fulfilled. Namely, for such values of ν_i [2],

$$u_i(r) = (-1)^{l_i+1}\nu_i^{3/2}\, D(\nu_i, l_i)\, W_{\nu_i, l_i+1/2}(2r/\nu_i) \tag{21}$$

with $D(\nu_i, l_i) = [\nu_i^2\Gamma(\nu_i + l_i + 1)\Gamma(\nu_i - l_i)]^{-1/2}$. Following Seaton [2,3], Eq. (2) is usually recast in terms of these functions, as

$$\Psi = \mathcal{A}\sum_i Z_i \chi_i\, D(\nu_i, l_i)\, W_{\nu_i, l_i+1/2}(2r/\nu_i)/r, \tag{22}$$

where

$$Z_i = (-1)^{l_i+1}\nu_i^{3/2}\, C_i = (-1)^{l_i+1}\nu_i^{3/2}\sum_\alpha U_{i\alpha}\cos[\pi(\nu_i + \mu_\alpha)]A_\alpha. \tag{23}$$

The C_i's and A_α's, and thus the Z_i's, are defined only to within an overall constant factor by Eqs. (12), (13) and (17). The value of this factor is constrained by the normalisation of the wave function, $\Psi(X,r)$: its value must be such that the Z_i's satisfy the condition [2]

$$\sum_{i,j} Z_i M_{ij} Z_j = 1, \tag{24}$$

where[2]

$$M_{ij} = \delta_{ij} + \frac{1}{2}\frac{\mu e^4}{(4\pi\epsilon_0)^2\hbar^2} q_i \left(\frac{dK_{ij}}{d\mathcal{E}}\right)q_j$$

$$= \delta_{ij} + \tilde{R}\, q_i \left(\frac{dK_{ij}}{dE}\right)q_j \tag{25}$$

with

$$q_i = (-1)^{l_i}\left(\frac{2}{\pi\nu_i^3}\right)^{1/2}\cos(\pi\nu_i). \tag{26}$$

The mixing coefficients Z_i and the channel fractions $|Z_i|^2$ quantify the importance of each dissociation channel in the states of interest. The energy dependence of K is often neglected in the normalisation condition, in which case the channel fractions sum to unity:

$$\sum_i |Z_i|^2 = 1. \tag{27}$$

It is often preferable to use different angular momentum coupling schemes for the dissociation channels than for the eigenchannels, as the

---

[2] Note that the energy $\mathcal{E}$ appearing in Eq. (6.50) of Ref. [2] is the "Z-scaled energy", $\mathcal{E} = 2\mathcal{E}/Z^2$ where $Z = 1$ for a neutral atom.

latter refer to a region where the outer electron interacts more strongly with the core than in the former. Transforming the mixing coefficients from the coupling scheme used for the dissociation channels to one better adapted to the eigenchannels can be done by way of an orthogonal transformation matrix, $[U_{i\bar\alpha}]$, where the index $\bar\alpha$ ($\bar\alpha = 1, \dots, N_{\mathrm{ch}}$) labels the channels in the transformed coupling scheme:

$$Z_{\bar\alpha} = \sum_i U_{i\bar\alpha} Z_i. \tag{28}$$

In the case of complex atoms, for example, LS-coupling is often better suited for describing the eigenchannels, since all the electrons interact strongly in the inner region, whereas jj-coupling is often better suited for describing the dissociation channels, since the outer electron is nearly free when far from the core. The recoupling matrix can be written in terms of Wigner 9-j symbols in this case [7]:

$$U_{i\bar\alpha} = \sqrt{(2j_{ci}+1)(2j_{ei}+1)(2L_{\bar\alpha}+1)(2S_{\bar\alpha}+1)} \begin{Bmatrix} l_{ci} & s_{ci} & j_{ci} \\ l_{ei} & s_{ei} & j_{ei} \\ L_{\bar\alpha} & S_{\bar\alpha} & J_{\bar\alpha} \end{Bmatrix}, \tag{29}$$

where l_{ci}, s_{ci} and j_{ci} are the quantum numbers of the core in channel i, l_{ei}, s_{ei} and j_{ei} the quantum numbers of the outer electron in channel i, and $L_{\bar\alpha}$, $S_{\bar\alpha}$ and $J_{\bar\alpha}$ the quantum numbers of the LS-coupled system.

The matrix $[U_{i\alpha}]$ is usually written in the form of a product of the transformation matrix $[U_{i\bar\alpha}]$ and another orthogonal matrix, $[V_{\bar\alpha\alpha}]$, the latter being itself factorised into a product of N_{rot} rotation matrices $[R_{\bar\alpha\alpha}(m)]$:

$$[U_{i\alpha}] = [U_{i\bar\alpha}][V_{\bar\alpha\alpha}] = [U_{i\bar\alpha}][R_{\bar\alpha\alpha}(1)][R_{\bar\alpha\alpha}(2)] \cdots [R_{\bar\alpha\alpha}(N_{\mathrm{rot}})]. \tag{30}$$

The rotation matrices $[R_{\bar\alpha\alpha}(m)]$ are defined as follows: each m corresponds to an angle θ_m and to a pair of indices, p_m and q_m, and

$$R_{\bar\alpha\alpha}(m) = \begin{cases} \cos\theta_m & \text{if } \bar\alpha = \alpha = p_m, \\ -\sin\theta_m & \text{if } \bar\alpha = p_m \text{ and } \alpha = q_m, \\ \sin\theta_m & \text{if } \bar\alpha = q_m \text{ and } \alpha = p_m, \\ \cos\theta_m & \text{if } \bar\alpha = \alpha = q_m, \\ 1 & \text{if } \bar\alpha = \alpha \neq p_m, q_m, \\ 0 & \text{for any other values of } \bar\alpha \text{ and } \alpha. \end{cases} \tag{31}$$

Since $[V_{\bar\alpha\alpha}]$ is an $N_{\mathrm{ch}} \times N_{\mathrm{ch}}$ orthogonal matrix, at most $N_{\mathrm{ch}}(N_{\mathrm{ch}}-1)/2$ rotations are necessary for factorizing $[V_{\bar\alpha\alpha}]$ in that way. The angles θ_m are taken to be fitting parameters in empirical MQDT calculations carried out along these lines, together with the eigen quantum defects μ_α. In practice, fewer than $N_{\mathrm{ch}}(N_{\mathrm{ch}}-1)/2$ are usually necessary for achieving a suitable fit. The factorisation therefore has the double advantage of making it possible to reduce the number of fitting parameters and of ensuring that the resulting $[U_{i\alpha}]$ matrix is orthogonal.

3. Methods

3.1. Energy dependence of the MQDT parameters

The elements of the $[K_{ij}]$ matrix as well as those of the $[V_{\bar\alpha\alpha}]$ matrix and the eigen quantum defects μ_α are, in principle, energy-dependent. However, it is found that this dependence is usually slow and does not need to be described to a high level of accuracy in order to obtain bound states energies in satisfactory agreement with experiment.

Standard practice in calculations based on the eigenchannel version of the theory is to take $[V_{\bar\alpha\alpha}]$ to be constant in energy and assume a linear dependence on E for the eigen quantum defects. Following previous work in that area — see, e.g., Refs. [19] and [26] — two alternative parametrisations are implemented in the present programs: each energy-dependent μ_α is written in terms of two constants, $\mu_\alpha^{(0)}$ and $\mu_\alpha^{(1)}$, and either

$$\mu_\alpha = \mu_\alpha^{(0)} + \mu_\alpha^{(1)} \frac{(I_s - E)}{I_s}, \tag{32}$$

where I_s is the first ionisation threshold, or

$$\mu_\alpha = \mu_\alpha^{(0)} + \mu_\alpha^{(1)} \frac{(I_s - E)}{\tilde R}. \tag{33}$$

Neglecting the dependence on E of the matrix $[V_{\bar\alpha\alpha}]$ makes it possible to reduce Eqs. (24) and (25) to the simpler form [7]

$$\sum_i Z_i^2 + 2\tilde R \sum_\alpha \left(\frac{\mathrm{d}\mu_\alpha}{\mathrm{d}E} \right) A_\alpha^2 = 1. \tag{34}$$

The calculations based on the K-matrix version of the theory follow Ref. [14], in that this matrix is allowed to vary with energy through a linear dependence of its diagonal elements, its off-diagonal elements being taken to be constant. As for the eigenchannel formulation, two alternative parametrisations of this dependence are implemented in this module, one in which the diagonal elements are written as

$$K_{ii} = K_{ii}^{(0)} + K_{ii}^{(1)} \frac{(I_s - E)}{I_s}, \tag{35}$$

and one in which they are written as

$$K_{ii} = K_{ii}^{(0)} + K_{ii}^{(1)} \frac{(I_s - E)}{\tilde R}, \tag{36}$$

the coefficients $K_{ij}^{(0)}$ and $K_{ij}^{(1)}$ being constant. The off-diagonal elements of the K-matrix are assumed to be constant:

$$K_{ij} \equiv K_{ij}^{(0)}, \quad i \neq j. \tag{37}$$

3.2. Calculation of theoretical energies

The energies are determined by the condition that Eqs. (11) and either Eq. (16) or Eq. (20) are satisfied simultaneously, for a given K-matrix or for given eigen quantum defects and rotation angles. One of the channels, channel j, is given a special role in the calculation. Typically, channel j would be chosen amongst those dominating the series of interest; however, this choice is largely immaterial. In view of Eqs. (11), a given value of v_j corresponds to a value of v_i for the i-th channel given by the equation

$$v_i = F_i(j; v_j), \tag{38}$$

where

$$F_i(j; v_j) = \left[\frac{I_i - I_j}{\tilde R} + \frac{1}{v_j^2} \right]^{-1/2}. \tag{39}$$

The bound state energies are found by identifying the values of v_j for which Eq. (16) or Eq. (20) is satisfied when all the other v_i's are calculated as per Eq. (38). The program offers the possibility of finding these values through a direct search of the roots of the relevant determinant or through a more indirect method outlined below.

For efficiency, the direct search is confined to the vicinity of values of v_j corresponding to experimental energies: given a set of experimental energies E_n^{exp}, the program searches for a root of the determinant in each of the intervals

$$v_{j,n}^{\mathrm{exp}} - \delta v_n^{(-)} \leq v_j \leq v_{j,n}^{\mathrm{exp}} + \delta v_n^{(+)},$$

where

$$v_{j,n}^{\mathrm{exp}} = [(I_j - E_n^{\mathrm{exp}})/\tilde R]^{-1/2}. \tag{40}$$

Two different schemes are possible for this: the $\delta v_n^{(+)}$'s and $\delta v_n^{(-)}$'s can be chosen to be multiples of a single user-defined δv or to vary from energy to energy.

For each E_n, in the first scheme, the program explores a sequence of search intervals $\Delta_n(0)$, $\Delta_n(-1)$, $\Delta_n(1)$, $\Delta_n(-2)$, $\Delta_n(2)$, etc., until one is found to contain a root of the determinant. Here

$$\Delta_n(p) = [\nu_{j,n}^{\exp} - \delta\nu + p\,\delta\nu, \ \nu_{j,n}^{\exp} + \delta\nu + p\,\delta\nu], \quad p = 0, \pm 1, \pm 2, \dots \tag{41}$$

$\Delta_n(p)$ is deemed not to bracket a root if the sign of the determinant is the same at $\nu_{j,n}^{\exp} - \delta\nu + p\,\delta\nu$ as at $\nu_{j,n}^{\exp} + \delta\nu + p\,\delta\nu$. Once a bracketing interval is found, a root is narrowed down by the Brent's method based `brentq` function of the `scipy.optimize` library.

The other scheme determines the $\delta\nu_n^{(+)}$'s and $\delta\nu_n^{(-)}$'s from the distance between the different experimental energies. Assuming that these energies are ordered by increasing values, $\delta\nu_n^{(+)}$ and $\delta\nu_n^{(-)}$ are taken such that

$$\delta\nu_n^{(+)} = \alpha\left(\nu_{j,n+1}^{\exp} - \nu_{j,n}^{\exp}\right), \qquad \delta\nu_n^{(-)} = \alpha\left(\nu_{j,n}^{\exp} - \nu_{j,n-1}^{\exp}\right), \tag{42}$$

where α is a user-defined numerical factor. These intervals do not vary in the course of the calculation, and, as above, the `brentq` function is used to locate the roots of the determinant.

As an alternative on a direct search for the roots of the relevant determinant, the program also implements the method described in Ref. [14], in which the problem is reformulated in terms of a search for the minima of a function. Two channels play a special role in this method, channel j as above, and channel k, which must have a different ionisation limit than channel j but whose choice is otherwise arbitrary. Given a value of ν_j and the corresponding values of ν_i for the channels for which $I_i \neq I_k$, those ν_i's being obtained from Eq. (38), the left-hand side of Eq. (20) is a polynomial of degree N_k in $\tan(\pi\nu_k)$, where N_k is the number of dissociation channels converging to the same ionisation limit as channel k ($N_k = 1$ if channel k is the only channel with that ionisation limit). Calculating the roots of this polynomial yields N_k different ν_j-dependent values of ν_k in the interval $[0, 1)$, i.e., $G_{k;1}(\nu_j)$, $G_{k;2}(\nu_j)$, ..., $G_{k;N_k}(\nu_j)$.[3] One of these values must coincide with $F_k(\nu_k) \bmod 1$ at those values of ν_j corresponding to a bound state energy. Solving Eq. (20) thus amounts to searching for the zeros of the functions $G_{k;s}(\nu_j) - F_k(\nu_j) \bmod 1$, $s = 1, \dots, N_k$. It has been noticed [15] that this problem is best tackled by searching for the minima of the function $\Xi^2(\nu_j)$, where

$$\Xi^2(\nu_j) \equiv \min\left[G_{k;s}(\nu_j) - F_k(\nu_j) \bmod 1\right]^2, \quad s = 1, \dots, N_k. \tag{43}$$

The calculation is programmed accordingly, using the implementation of the Brent's minimisation method offered by the function `minimize_scalar` of the `scipy.optimize` library. The search intervals are as described above, with the difference that their bounds are meant to bracket a minimum of $\Xi^2(\nu_j)$ rather than bracket a root of the relevant determinant. As mentioned in the previous paragraph, calculating the $G_{k;s}(\nu_j)$'s involves representing $\det|K_{ij} + \delta_{ij}\tan(\pi\nu_i)|$ as a polynomial of order N_k in $\tan(\pi\nu_k)$. The $(N_k + 1)$ coefficients of this polynomial are obtained by calculating the determinant at $(N_k + 1)$ different values of ν_k [19] and using the `polyfit` function of the `polynomial.polynomial` module of the `numpy` library to fit a polynomial of order N_k to these values. The function `polyroots` of this module is then used to calculate the roots of that polynomial. The required elements of the K-matrix are obtained from the eigen quantum defects and the rotation angles, as indicated by Eqs. (18) and (30), in calculations based on the eigenchannel formalism.

3.3. Optimization of the MQDT parameters

The MQDT parameters (K-matrix elements, or eigen quantum defects and rotation angles) can be optimized by χ^2-fitting to a set of

experimental energies. These parameters are divided into fitting parameters, whose values are variable, and static parameters, whose values (normally 0) remain constant. Given a set of experimental energies $E_n^{\exp}$ and associated experimental errors $\alpha_n^{\exp}$, the program calculates the corresponding theoretical energies E_n^{th} as described in Section 3.2 and varies the fitting parameters so as to minimise the value of χ^2, where

$$\chi^2 = \sum_n [(E_n^{\exp} - E_n^{\text{th}})/\alpha_n^{\exp}]^2. \tag{44}$$

A Nelder-Mead downhill simplex algorithm is used to this effect, through the function `fmin` of the `scipy.optimize` library. The user must provide starting values of the relevant MQDT parameters.

In calculations based on the eigenchannel formulation of the theory, the $\mu_\alpha^{(0)}$'s and the θ_m's are always assumed to be variable fitting parameters, whereas each of the $\mu_\alpha^{(1)}$'s can be taken to be a static parameter (set to a given invariable values) or a variable fitting parameter, to the choice of the user. In calculations based on the K-matrix formulation of the theory, each of the $K_{ij}^{(0)}$'s and of the $K_{ii}^{(1)}$'s can be defined as being static or variable.

3.4. Lu-Fano plots

The program includes a function producing Lu-Fano plots [5], i.e., here, plots of $\nu_k \bmod 1$ vs. ν_j, where ν_j and ν_k are the same effective principal quantum numbers as in Section 3.2. The experimental data are indicated by markers located at the points of co-ordinates $(\nu_{j,n}^{\exp}, \nu_{k,n}^{\exp})$, where the $\nu_{j,n}^{\exp}$'s are defined by Eq. (40) and the $\nu_{k,n}^{\exp}$'s by the equation

$$\nu_{k,n}^{\exp} = [(I_k - E_n^{\exp})/\tilde{R}]^{-1/2}. \tag{45}$$

Theory curves are also plotted, showing how ν_k varies as a function of ν_j when ν_k is taken equal to $G_{k;s}(\nu_j)$, $s = 1, \dots, N_k$. These N_k curves pass through the data points when the MQDT model describes the experimental energies well.

4. General information about the module

4.1. Installation and general comments

The programs forming this library implement the theory and methods outlined in Section 2 of this document in a generally straightforward way. These programs are entirely written in Python 3 and are grouped in a single file, forming a single module. This module can be made available to other programs by an `import` statement. Specifically, the functions provided by this module can be used (1) to calculate theoretical energies, mixing coefficients and channel fractions given the details of an MQDT model and a set of experimental energies; (2) to optimize the parameters of an MQDT model; (3) to evaluate the χ^2 statistics characterizing the goodness of fit of an MQDT model to a set of experimental energies; and (4) to draw Lu-Fano plots.

The module currently contains 40 functions, amongst which 14 are intended to interface with a driving program provided by the user. A technical description of the input parameters of each of these functions and of the results they return can be found at the start of the respective code. A brief description of each of the user-facing functions is given in a user manual provided with the distribution. Examples of driving programs are also provided and are briefly described in Section 5.

4.2. Using the module

Any use of the module must start by a call to either one of two initialisation functions, `initialize_eigenchannel` or `initialize_Kmatrix`, respectively for calculations in the eigenchannel formulation of the theory and for calculations formulated in terms of a K-matrix. Calculations in one formulation cannot be mixed with calculations in the other. Various details of the MQDT model are passed

---

[3] Due to the periodicity of the tangent function, the system is also satisfied for $\nu_k = G_{k;s}(\nu_j) \pm 1$, $G_{k;s}(\nu_j) \pm 2$, etc. However, these other solutions are not relevant for our purposes.

to the module through this initialisation call, as well as a list of experimental energies and experimental errors, indices identifying the j and k channels, and variables controlling either how the theoretical energies are calculated or how the energy dependence of the MQDT parameters is taken into account.

Initialisation must be followed by a call to `search_intervals`, to set the mesh used in calculations of energies. A list of MQDT parameters must also be passed to the program, i.e., either the $\mu_i^{(0)}$'s, $\mu_i^{(1)}$'s and θ_m's for eigenchannel calculations, which are passed through `mqdtparams_eigenchannel`, or the $K_{ij}^{(0)}$'s and $K_{ii}^{(1)}$'s for K-matrix calculations, which are passed through `mqdtparams_Kmatrx`.

The module provides are a number of possibilities after this initialisation stage:

- call the function `optimizeparams`, which optimises the parameters of the model;
- call the function `print_energies`, which calculates and writes out theoretical energies;
- call the function `list_Zcoeffsandchannelfractions`, which calculates and writes out the mixing coefficients and channel fractions;
- call the function `print_chi2`, which calculates and writes out the χ^2 and reduced χ^2 values characterising the goodness of fit of the model;
- call the function `LuFano_plot`, which draws a Lu-Fano plot;
- call the function `plot_channelfractions`, which calculates and plots channel fractions.

For convenience, the module also contains the following non-computational user-facing functions:

- `reset_calculation_method`, which can be used for changing the calculation method after the initialization stage;
- `print_channelparams`, which writes out the channel parameters passed to the module;
- `print_mqdtparams`, which writes out the MQDT parameters obtained by or passed to the program.

More information about these functions can be found in the user manual joined with this distribution.

The following details should be noted:

1. The experimental energies, experimental errors, ionisation limits and calculated energies are generally passed to or from the module in terms of the corresponding wave numbers expressed in cm^{-1}.
2. The channels are labelled in numerical order, starting at 1 as it is customary in the field. Python arrays containing the corresponding information must be dimensioned accordingly. For example, the array `Ilim` containing the ionisation limits of a 2-channel model needs to be created as containing at least three elements, e.g., through the statement `Ilim = numpy.empty(3)`, so that `Ilim[1]` can be set to I_1 and `Ilim[2]` to I_2 (`Ilim[0]` is not used).
3. The MQDT parameters are passed between many of the functions of this module as a single 1D array rather than as whole matrices or separate arrays of eigen quantum defects and rotation angles. The way in which these parameters are arranged in this 1D array is not important for routine use of this library.

5. Examples

Two example programs are included in this distribution, namely one illustrating the use of the module for calculating energies and producing Lu-Fano plots in the eigenchannel formulation of MQDT, and one illustrating its use for optimizing the parameters of a K-matrix based model and calculating the resulting energies, mixing coefficients and channel fractions.

The first program (file `ytterbium.py`) concerns the 6snd ^{1}D$_2$ and 6snd ^{3}D$_2$ series of ^{174}Yb. It uses the same model and the same experimental energies and experimental errors as in Tables V and VIII of Ref. [9] and reproduces the theoretical energies and Lu-Fano plot of Table V and Fig. 4(a) of that reference. The data are read from the file `ytterbiumD2data.dat`, also included in this distribution. A copy of the output can be found in the files `ytterbium_output.txt` and Lu-Fanoplot.pdf.

The other program (file `strontium.py`) concerns the 5snp ^{1}P$_1$ series of ^{88}Sr [17]. A copy of the output is also joined with the distribution, namely the files `strontium_output.txt` and `channelfractions.pdf`.

CRediT authorship contribution statement

Declaration of competing interest

Data availability

No data was used for the research described in the article.

Acknowledgements

Appendix A. About the C_i and A_α coefficients

The relations between the C_i and A_α coefficients stated in Section 2 are derived in this appendix. The discussion follows Ref. [13].

We start with Eqs. (7), (8) and (9). Identifying the terms in $f(\epsilon_i, l_i, r)$ and those in $g(\epsilon_i, l_i, r)$ in this last equation yields, respectively,

$$C_i \cos(\pi \nu_i) = \sum_\alpha U_{i\alpha} \cos(\pi \mu_\alpha) A_\alpha \tag{A.1}$$

and

$$C_i \sin(\pi \nu_i) = -\sum_\alpha U_{i\alpha} \sin(\pi \mu_\alpha) A_\alpha. \tag{A.2}$$

Multiplying Eq. (A.2) by $-i$, adding the result to Eq. (A.1) and rearranging gives

$$C_i = \sum_\alpha U_{i\alpha} \exp[i\pi(\nu_i + \mu_\alpha)] A_\alpha. \tag{A.3}$$

Using the orthogonality of the matrix $[U_{i\alpha}]$, one can also deduce, from this last equation, that

$$A_\alpha = \sum_i U_{i\alpha} \exp[-i\pi(\nu_i + \mu_\alpha)] C_i. \tag{A.4}$$

We choose the wave function $\Psi(X, r)$ to be real, which is not restrictive and implies that the coefficients A_α and C_i have a zero imaginary part. As the $U_{i\alpha}$'s are also real, Eqs. (15) and (13) then follow from taking

the real and imaginary parts of Eq. (A.3), and Eqs. (14) and (12) follow from taking the real and imaginary parts of Eq. (A.4).

Eq. (17) follows from re-writing Eq. (12) in the form

$$\sum_j U_{j\alpha} \cos(\pi\mu_\alpha)[\tan(\pi\nu_j) + \tan(\pi\mu_\alpha)]\cos(\pi\nu_j)C_j = 0, \quad \alpha = 1, \dots, N_{\text{ch}},$$

$$(A.5)$$

dividing each equation by $\cos(\pi\mu_\alpha)$, multiplying it on the right by $U_{i\alpha}$, and summing over α. Doing so indeed yields

$$\sum_{j,\alpha} U_{j\alpha} \tan(\pi\nu_j) + \tan(\pi\mu_\alpha)]U_{i\alpha}c_j = 0 \qquad (A.6)$$

with $c_j = \cos(\pi\nu_j)C_j$. Eq. (17) follows since

$$\sum_\alpha U_{j\alpha}\tan(\pi\mu_\alpha)U_{i\alpha} = \sum_\alpha U_{i\alpha}\tan(\pi\mu_\alpha)U_{j\alpha} = K_{ij} \qquad (A.7)$$

by definition of the $[K_{ij}]$ matrix, and

$$\sum_\alpha U_{j\alpha}U_{i\alpha} = \delta_{ij} \qquad (A.8)$$

owing to the orthogonality of $[U_{i\alpha}]$.

References

[1] M. Aymar, C.H. Greene, E. Luc-Koenig, Rev. Mod. Phys. 68 (1996) 1015.
[2] M.J. Seaton, Rep. Prog. Phys. 46 (1983) 167.
[3] M.J. Seaton, Proc. Phys. Soc. 88 (1966) 801.
[4] U. Fano, Phys. Rev. A 2 (1970) 353.
[5] K.T. Lu, U. Fano, Phys. Rev. A 2 (1970) 81.
[6] K.T. Lu, Phys. Rev. A 4 (1971) 579.
[7] C.M. Lee, K.T. Lu, Phys. Rev. A 8 (1973) 1241.
[8] M. Aymar, Phys. Rep. 110 (1984) 163.
[9] H. Lehec, A. Zuliani, W. Maineult, E. Luc-Koenig, P. Pillet, P. Cheinet, F. Niyaz, T.F. Gallagher, Phys. Rev. A 98 (2018) 062506.
[10] F. Robicheaux, D.W. Booth, M. Saffman, Phys. Rev. A 97 (2018) 022508.
[11] F. Robicheaux, J. Phys. B, At. Mol. Opt. Phys. 52 (2019) 244001.
[12] Y. Zhu, R. Jin, Y. Wu, J. Wang, Chin. Phys. B 31 (2022) 043103.
[13] W.E. Cooke, C.L. Cromer, Phys. Rev. A 32 (1985) 2725.
[14] C.L. Vaillant, M.P.A. Jones, R.M. Potvliege, J. Phys. B, At. Mol. Opt. Phys. 47 (2014) 155001.
[15] C.L. Vaillant, Long-Range Interactions in One- and Two-Electron Rydberg Atoms, Ph.D. thesis, University of Durham, 2014, etheses.dur.ac.uk/10594.
[16] C.L. Vaillant, R.M. Potvliege, M.P.A. Jones, Phys. Rev. A 92 (2015) 042705.
[17] C.L. Vaillant, M.P.A. Jones, R.M. Potvliege, 2024, submitted for publication, arXiv: 2403.08742v1.
[18] M. Aymar, R.-J. Champeau, C. Delsart, J.-C. Keller, J. Phys. B, At. Mol. Phys. 14 (1981) 4489.
[19] O. Robaux, M. Aymar, Comput. Phys. Commun. 25 (1982) 223.
[20] U. Fano, J. Opt. Soc. Am. 65 (1975) 979.
[21] C.H. Greene, Quantum defect theory, in: G.W.F. Drake (Ed.), Springer Handbook of Atomic, Molecular and Optical Physics, Springer, Cham, 2023, pp. 751–759.
[22] J.A. Armstrong, P. Escherick, J.J. Wynne, Phys. Rev. A 15 (1977) 80.
[23] C. Greene, U. Fano, G. Strinati, Phys. Rev. A 19 (1979) 1485.
[24] T.F. Gallagher, Rydberg Atoms, Cambridge University Press, 1994.
[25] J.L. Dehmer, U. Fano, Phys. Rev. A 2 (1970) 304.
[26] P. Esherick, Phys. Rev. A 15 (1977) 1920.