

Computational Physics

The `qspect` Python package: A physics toolbox for laser spectroscopyP. Müller^{a,*}, W. Nörtershäuser^{a,b}^a Institut für Kernphysik, Technische Universität Darmstadt, 64289, Darmstadt, Germany^b Helmholtz Research Academy Hesse for FAIR, Campus Darmstadt, 64289, Darmstadt, Germany

ARTICLE INFO

The review of this paper was arranged by Prof. W. Jong

Dataset link: <https://pypi.org/project/qspect/>,
<https://github.com/patmlr/qspect>

Keywords:

Laser spectroscopy
 Python
 Data analysis
 Multidimensional linear regression
 Maximum likelihood fitting
 Master equation

ABSTRACT

The analysis of experimental results with Python often requires writing many code scripts which all need access to the same set of functions. In a common field of research, this set will be nearly the same for many users. The `qspect` Python package was developed to provide functions for physical formulas, simulations and data analysis routines widely used in laser spectroscopy and related fields. Most functions are compatible with `numpy` arrays, enabling fast calculations with large samples of data. A multidimensional linear regression algorithm enables a King plot analyses over multiple atomic transitions. A modular framework for constructing lineshape models can be used to fit large sets of spectroscopy data. A simulation module within the package provides user-friendly methods to simulate the coherent time-evolution of atoms in electromagnetic fields without the need to explicitly derive a Hamiltonian.

1. Introduction

Laser spectroscopy is in general concerned with the manipulation of the inner and outer degrees of freedom of atoms or ions using laser light. The subject of the interaction reaches from individual ions or atoms captured in a trap, over hot thermal ensembles in a neutral or charged plasma up to beams at high and even relativistic speed. The goal of the laser interaction can also be divers, e.g., transferring atoms into a specific (excited) state, generating fluorescence photons for detection, change the ions motional degree by momentum transfer (laser cooling), or to produce nuclear or atomic polarization or alignment. The `qspect` package presented here is intended to provide a fundamental physics and data processing framework for experiments based on laser-atom interactions, where atoms are understood as electrons bound to a moving unresolved nucleus. While parts of `qspect` are made for general data processing and physics calculations, being based in laser spectroscopy experiments, the generic experiment benefiting the most from `qspect` produces data by reading out the response of atoms after manipulating them with lasers. Ultimately, these are experiments whose means are the

determination of high-precision transition frequencies for a wide variety of possible goals, which range from collinear laser spectroscopy (CLS) to probe atomic or nuclear structure theory [1–6], over searches for new physics [7,8], to the development of atomic clocks [9–11] or quantum computing with atom or ion traps [12–14]. During the planning or the analysis phase, most of these experiments require simulations of the time evolution of atoms in laser fields, nonlinear fits to spectroscopy data, error propagation or simple calculations of laser frequencies, power, polarization as well as energies, velocities, Doppler, hyperfine-structure (hfs), or Zeeman shifts *etc.*

The goal of the `qspect` package is to make calculations of such observables and simulations easily accessible in any analysis script without the need to copy source code into every new project. While this is already useful for the most basic functions, it is even more practical for more extensive calculations, as long as they can still be defined in sufficiently general terms. In this sense, the package has been widely used and tested in simulating and analyzing laser spectroscopy results in collinear laser spectroscopy at COALA [15], COLLAPS/ISOLDE [16], and spectroscopy at storage rings [17], but can be easily used for laser spectroscopy on

* Corresponding author.

E-mail address: pmueller@physics.ucla.edu (P. Müller).¹ Present address: Department of Physics & Astronomy, University of California Los Angeles, Los Angeles, California 90095, USA.

thermal beams or in ion or atom traps as well.

`qspect` expands on the concept of assisting the planning and analysis of experiments by also providing general mathematical, statistical and optimization methods, building on the foundation of the `numpy` and `scipy` packages. Additionally, object-oriented frameworks to create fit models in a modular way (`qspect.models`) and to time-evolve coherent laser-atom interactions (`qspect.simulate`) are implemented. These two frameworks and the derivation of a maximum-likelihood fit of a straight line to n -dimensional uncertain data points constitute the main body of this article. In the following section, a brief introduction into atomic hyperfine structure and fluorescence spectra is given, as these constitute the core of the implemented theory of light-matter interactions. In Sec. 3, a technical summary and an overview over the `qspect` package is given, including a short tutorial on how to get started.

2. Theory

In atomic physics, the required theory and calculations are often well understood and developed such that the user of a physics code library can quickly relate a name of a method to the theory and assess the underlying physical assumptions and approximations. Hence, for introductions to the general physical concepts and the standard approaches to laser spectroscopy and the underlying atomic physics of the `qspect` package, standard textbook resources can be consulted [18–20]. However, in high-precision experiments, also small effects such as hyperfine-induced mixing [21] or quantum interference in the photon scattering rate [22,23] can significantly influence an experiment. Since these effects go beyond the standard frameworks, but are also within the scope of `qspect`, they shall be briefly introduced in the following paragraphs. In the field of laser spectroscopy, usually transition frequencies between fine-structure states are addressed, e.g., for determining isotope shifts, or are specified in literature, while hyperfine-structure splittings, emerging in isotopes with nonzero nuclear spin, often need to be calculated. The hyperfine structure Hamiltonian, describing the higher-order multipole terms of the electromagnetic interaction between the electrons and the nucleus beyond the Coulomb interaction, can be written as [18,24]

$$H_{\text{HFS}} = \sum_{k \geq 1} \mathbf{M}^{(k)} \cdot \mathbf{T}^{(k)} \equiv \sum_{k \geq 1} \sum_{\lambda = -k}^k (-1)^\lambda \mathbf{M}_{-\lambda}^{(k)} \cdot \mathbf{T}_\lambda^{(k)}, \quad (1)$$

where $\mathbf{M}^{(k)}$ and $\mathbf{T}^{(k)}$ are irreducible tensor operators of rank and multipole order k , acting on the nucleus and the electrons, respectively. In the second expression, the scalar product is written out in the spherical basis. Using the Wigner-Eckhart theorem, this can be written as

$$\begin{aligned} h\nu_{(\gamma J)(\gamma' J')}^F &:= \langle \alpha I, \gamma J; F m | H_{\text{HFS}} | \alpha' I', \gamma' J'; F' m' \rangle \\ &= \sum_{k \geq 1} \delta_{FF'} \delta_{mm'} (-1)^{I' + J + F} \left\{ \begin{matrix} I & J & F \\ J' & I' & k \end{matrix} \right\} \\ &\quad \times \langle \alpha I | \mathbf{M}^{(k)} | \alpha' I' \rangle \langle \gamma J | \mathbf{T}^{(k)} | \gamma' J' \rangle, \end{aligned} \quad (2)$$

$$\langle II | \mathbf{M}_0^{(k)} | II \rangle =: \left(\begin{matrix} I & k & I \\ I & 0 & -I \end{matrix} \right) \langle II | \mathbf{M}^{(k)} | II \rangle \quad (3)$$

$$\langle JJ | \mathbf{T}_0^{(k)} | JJ \rangle =: \left(\begin{matrix} J & k & J \\ J & 0 & -J \end{matrix} \right) \langle JJ | \mathbf{T}^{(k)} | JJ \rangle \quad (4)$$

where h is the Planck constant, $\delta_{FF'}$, $\delta_{mm'}$ are Kronecker-deltas, $(:::)$ and $\{:::\}$ are the Wigner-3j and -6j symbols and reduced matrix elements are denoted by $\langle \cdot | \cdot | \cdot \rangle$. The quantum numbers I , I' represent the nuclear spin, J , J' the total electron angular momentum, F the combined total angular momentum and α , γ encapsulate additional quantum numbers such as particle spins. If the off-diagonal elements, linking different I and J , vanish, this expression can be simplified to the textbook hyperfine structure formula [18,24]

$$\nu_{(\gamma J)(\gamma' J')}^F = A \frac{K}{2} + B \frac{\frac{3}{4}K(K+1) - I(I+1)J(J+1)}{2I(2I-1)J(2J-1)} \quad (5)$$

$$+ C \frac{\left[\frac{5}{4}K^3 + 5K^2 - 5I(I+1)J(J+1) + K(I(I+1) + J(J+1) - 3I(I+1)J(J+1) + 3) \right]}{I(I-1)(2I-1)J(J-1)(2J-1)}$$

$$K := F(F+1) - I(I+1) - J(J+1) \quad (6)$$

$$A_k := \langle II | \mathbf{M}_0^{(k)} | II \rangle \langle JJ | \mathbf{T}_0^{(k)} | JJ \rangle \quad (7)$$

$$hA := \frac{A_1}{IJ}, \quad hB := 4A_2, \quad hC := A_3, \quad (8)$$

with the hyperfine structure constants A , B and C in frequency units for the magnetic dipole, electric quadrupole and magnetic octupole term, respectively. Off-diagonal elements with respect to I are often neglected due to the large energy gaps between nuclear states, but will be of interest in very rare cases like ^{229}Th , which has a particularly low-lying isomeric state that leads to hyperfine-induced nuclear-level mixing with dramatic consequences for the lifetime of the nuclear state [25]. Off-diagonal terms in J are much more common and can give rise to hyperfine-induced fine-structure mixing, as for example in ^3He [26] or $^{11}\text{B}^{3+}$ [27]. If this is the case, the full Hamiltonian has to be diagonalized block-wise for each F to extract individual fine-structure frequencies ν_J , i.e., the matrices

$$N_{(\gamma J)(\gamma' J')}^F = \nu_J \delta_{(\gamma J)(\gamma' J')} + \nu_{(\gamma J)(\gamma' J')}^F \quad (9)$$

need to be diagonalized. The `qspect` package enables fitting to spectra with mixed J through numerical diagonalization. However, this requires additional input from theory or experiment. The nuclear moments, as well as the off-diagonal matrix elements $\langle \gamma J | \mathbf{T}^{(k)} | \gamma' J' \rangle$ need to be provided. Nuclear magnetic dipole moments ($k=1$) can be found in [28], electronic magnetic dipole matrix elements for He-like systems are listed, e.g., in [21]. The implementation in `qspect` is currently limited to $k=1$, but the extension is straightforward and may follow in a future update.

Fluorescence spectra are generated from detected photons resonantly scattered from an atom. Resonance peaks appear where the frequency of the incident photon matches that of an atomic transition. If excited quantum states are energetically close, a fluorescence spectrum cannot be simply described as a sum of resonance peaks anymore, but the superposition of the excited states has to be considered. This is particularly relevant for unresolved hyperfine structure spectra [3], but can also become relevant in high-precision measurements where peaks are hundreds of linewidths separated [23]. This “quantum interference” (QI) effect depends on the solid angle of detection of the emitted fluorescence light and leads in general to asymmetric lineshapes, which leads to shifts in the determined transition frequencies if simply fitted with a sum-of-peaks model. In `qspect`, the QI scattering rate derived in [22] is implemented as a fit model of the form

$$\begin{aligned} \frac{d\Gamma}{d\Omega}(\mathbf{k}_{\text{sc}}) &:= \frac{1}{4\pi} \frac{I}{I_0} \left(\frac{\Gamma}{2} \right)^3 \left[\sum_{F'} \frac{f(\mathbf{q}, \mathbf{k}_{\text{sc}}, F, F')}{\Delta_{FF'}^2 + (\Gamma/2)^2} \right. \\ &\quad \left. + \sum_{\substack{F', F'' \\ F' \neq F''}} \frac{g(\mathbf{q}, \mathbf{k}_{\text{sc}}, F, F', F'')}{(\Delta_{FF'} + \frac{i}{2}\Gamma)(\Delta_{FF''} - \frac{i}{2}\Gamma)} \right] \end{aligned} \quad (10)$$

$$f(\mathbf{q}, \mathbf{k}_{\text{sc}}, F, F') = A_F^{F'} + B_F^{F'} p(\mathbf{q}, \mathbf{k}_{\text{sc}}) \quad (11)$$

$$g(\mathbf{q}, \mathbf{k}_{\text{sc}}, F, F', F'') = C_F^{F'F''} p(\mathbf{q}, \mathbf{k}_{\text{sc}}) \quad (12)$$

$$I_0 = \frac{\hbar \Gamma \omega^3}{12\pi c^2}, \quad (13)$$

where $p(\mathbf{k}_{\text{sc}}, \mathbf{q})$ can be used as a fit parameter, takes values in $[-0.5, 1]$ and depends on the polarization of incident and the directions of detectable scattered photons. I_0 is the saturation intensity, $\hbar$ the reduced Planck constant, c the vacuum speed of light, $\Delta_{FF'} := \omega_{F'} - \omega_F$, $\omega := 2\pi\nu$ and Γ is the natural linewidth of the electronic transition. The coefficients $A_F^{F'}$, $B_F^{F'}$ and $C_F^{F'F''}$ only depend on angular momentum

algebra and are derived in [22]. For simulations, as part of `qspec.simulate`, the scattering rate was extended to address individual magnetic substates, allowing the use of a magnetic field. If the perturbative scattering rate is not sufficient, the `qspec.simulate` module can be used to coherently evolve an atom in a laser field using a master-equation approach. The resulting density matrix can be used to calculate the expectation value of a detection operator that gives the full non-perturbative solution and converges to the perturbative result for low laser powers and closed transitions in the equilibrium state. This is described in more detail in Sec. 6.

3. Structure

The `qspec` package provides functions for physical calculations and simulations such as the aforementioned examples in a user-friendly and efficient way. Whenever useful, functions are compatible with the `ndarray` class of `numpy` [29]. This enables fast calculations with large samples of data. In addition to `numpy`, a small number of established scientific Python packages are required with `scipy` [30], `sympy` [31] and `matplotlib` [32]. Parts of `qspec` are written in C++ and are currently only compiled for Windows systems in a Dynamic Link Library (DLL). A detailed overview of `qspec` with installation instructions, tutorials, the API documentation and links to the Python Package Index (PyPI), the GitHub repository and this publication can be found on the official website [33]. Supplemental details are also given in [34]. The content of the `qspec` package is distributed over seven largely independent modules/categories. These are named `analyze`, `algebra`, `models`, `physics`, `simulate`, `stats`, and `tools`. However, all but the `models` and the `simulate` modules can be accessed by simply importing `qspec`. This is demonstrated in the following short code example, often required in collinear laser spectroscopy. In this technique, a fast beam of ions or atoms is superimposed with a copropagating or counterpropagating laser beam [35] and a resonance curve is usually recorded by Doppler-tuning, *i.e.*, changing the ion velocity with a small potential change ΔU . The laboratory laser frequency required to excite the 24516.65 cm^{-1} transition in $^{88}\text{Sr}^+$ ions, accelerated to 20 keV, with a counterpropagating laser beam is calculated in the following example together with the change in the rest-frame frequency of the ion for a potential variation of 1 V. Note that, all of the scalar values in this example could be replaced by `numpy` arrays.

```

1 import qspec as qs
2 # This imports the analyze, algebra,
3 # physics, stats, and tools modules
4
5 q = 1 # (e), Ion charge state
6 m = 87.905612253 - q * qs.me_u # (u)
7 # Mass of  $^{88}\text{Sr}^+$ 
8
9 U = 20000 # (V), Acceleration voltage
10
11 # Resonance frequency from NIST database
12 f0 = qs.inv_cm_to_freq(24516.65) # (MHz)
13 # >>> 734990676.5 MHz
14
15 # Relativistic velocity of  $^{88}\text{Sr}^+$ 
16 v = qs.v_el(U, q, m) # (m/s)
17 # >>> 209533.6 m/s
18
19 # The anti-collinear lab. frequency
20 f_laser = qs.doppler(
21     f0, v, qs.pi, return_frame='lab') # (MHz)
22 # >>> 734477149.8 MHz
23
24 # The differential Doppler shift
25 df_atom = qs.doppler_el_d1(
26     f_laser, qs.pi, U, q, m) # (MHz / V)
27 # >>> +12.84 MHz / V

```

Functions are generally named according to their returned observable and required input parameters. Similar to this simple example, all functions and classes can be accessed, including the n -dimensional linear fit algorithm described in Sec. 4. Elaborate introductions to the two separate frameworks `qspec.models` and `qspec.simulate` are given in Sec. 5 and 6, respectively. The following paragraphs give a short overview over the seven modules/categories.

3.1. Algebra

The `algebra` module is a collection of functions related to angular momentum coupling such as Wigner- j symbols, reduced dipole matrix elements and relative transition strengths for electronic transitions. This is the only module that uses the `sympy` package to optionally return algebraic expressions of coupling strengths.

3.2. Analyze

The `analyze` module contains general purpose routines for (non)linear data fitting as well as a convenience class for multi-dimensional King plots [36]. Wrappers for the `scipy.optimize.curve_fit` and the `scipy.odr.odr` routine, which use least square optimization and orthogonal distance regression, respectively [30], extend their functionality and standardize their syntax. For fitting of a straight line, as needed in King-plots, the York algorithm described in [37] and the Monte-Carlo (MC) method described in the supplementary material of [38] are implemented. In both cases, also a generalized version is implemented for fitting a straight line to data points in multiple dimensions, whose uncertainties are determined by independent multivariate normal distributions. The multidimensional fit algorithms are described in Sec. 4.

3.3. Models

The `models` module contains classes and helper functions to create lineshape models of fluorescence spectra or more general fit models for nonlinear curve-fitting. This module is similar to the existing `satlas2` Python package [39,40], which is widely used by laser spectroscopy groups at KU Leuven, CERN/ISOLDE and JYVL. In `models`, additional features such as numerical convolution with commonly used probability density functions, quantum interference (QI) effects [22] or hyperfine-induced mixing [21] are available. The `models` module is described in detail in Sec. 5.

3.4. Physics

The `physics` module provides physical relations, unit conversions and observables surrounding (collinear) laser spectroscopy such as particle energy and velocity, Doppler, photon recoil, hyperfine structure and Zeeman shifts, refractive indexes, etc.

3.5. Simulate

The `simulate` module offers an intuitive framework for simulations of laser-atom interactions and is comparable to other packages that simulate the time evolution of quantum systems such as `qutip` [41,42] or `PyLCP` [43]. Currently, solvers for the Lindblad master, rate and Schrödinger equation as well as a MC solver for the Lindblad master equation are included. The differential photon scattering rate, which also includes quantum interference effects, can be calculated from the density matrix or the state vector. Additionally, a perturbative scattering rate can be calculated, as described in [22]. The `qspec.simulate` module is described in more detail in Sec. 6.

3.6. Stats

The `stats` module contains routines for the statistical analysis of data. Besides basic statistical measures such as the (weighted) average

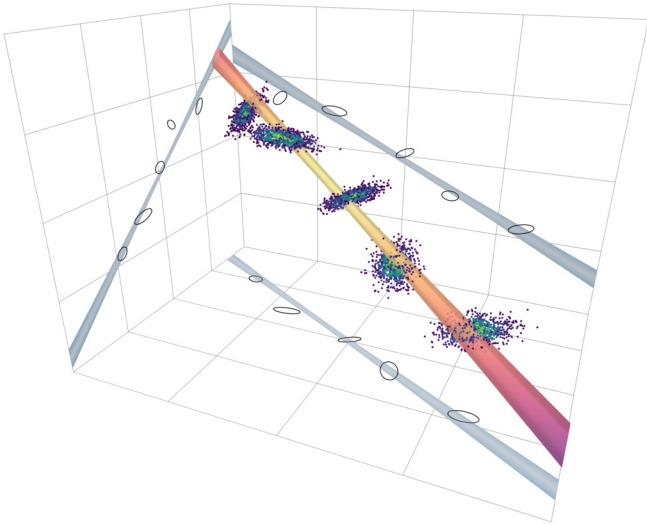


Fig. 1. Fit of a straight line to 5 data points in 3 dimensions. The scatter plots indicate the uncertainty regions of the data points. In the Monte-Carlo (MC) method, similar random samples are generated to construct a distribution of sample straight lines. The 1σ uncertainty region of the straight line fitted with the analytical algorithm is shown as a yellow-purple colormap. The projections of the 1σ uncertainty regions of the data points and the fitted straight line are shown on the three orthogonal planes. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

or median, the module also provides a MC error propagator. In combination with the nonlinear curve fitters of the *analyze* module, this can be used to calculate uncertainty bands for the fitted function while taking the full covariance matrix into account.

3.7. Tools

The *tools* module provides methods for data processing and conversion as well as general mathematical functions. Some examples are the conversion between lists and dictionaries, merging intervals, rounding to n significant decimal places, transforming vectors or printing colored text to the console *etc.*

4. Multidimensional linear regression

The *qspec* package provides an analytical and a MC method for fitting a straight line to data points in $k \in \mathbb{N}$ dimensions. For any set of data points $(\vec{X}_i)_{i \in I}$, with the index set I , the algorithms described here assume that the real value of the random vector $\vec{X}_i$ is distributed according to an independent multivariate normal distribution $\mathcal{N}(\vec{\mu}_i, \Sigma_i)$, with the mean vector $\vec{\mu}_i \in \mathbb{R}^k$ and the positive-semidefinite covariance matrix $\Sigma_i \in \mathbb{R}^{k \times k}$. Here, we distinguish between an uncertain data point $\vec{X}_i$, which may be the result of a measurement, and the actual or real but unknown value $\vec{x}_i$. Fig. 1 depicts the optimization problem in 3 dimensions with the resulting straight line fit.

The MC approach generates sample vectors $\vec{x}_i$ according to the distributions $\mathcal{N}(\vec{\mu}_i, \Sigma_i)$ under the condition that they are aligned on a straight line. Accordingly, the result of the MC method are samples of straight lines following the probability density function imposed by the uncertain data points and the linear relation. The algorithm is described in detail in the supplemental material of [38] for the 3-dimensional case. However, the extension of this algorithm to k dimensions is straightforward and is, thus, not described here.

The analytical approach is a maximum likelihood fit given the above described assumptions. Its result corresponds to the most probable sample straight line generated by the MC method. Note that in general this straight line is not given by the mean values of the slopes and y-intercepts of the samples. The algorithm described by York et al. [37]

yields the analytical solution for 2 dimensions. Here, the extension to k dimensions is not obvious. Hence, in the following, the solution developed for *qspec* is described.

Let a straight line in k dimensions be parameterized by

$$f[\vec{a}, \vec{b}] : \mathbb{R} \rightarrow \mathbb{R}^k, \quad t \mapsto \vec{a} + t\vec{b}. \quad (14)$$

Then we want to find the offset vector $\vec{a} \in \mathbb{R}^k$ and the directional vector $\vec{b} \in \mathbb{R}^k$ that define the most likely straight line through the set of uncertain data points $(\vec{X}_i)_{i \in I}$, which generate the independent multivariate normal distributions $\mathcal{N}(\vec{\mu}_i, \Sigma_i)_{i \in I}$. The probability density of $\vec{X}_i$ can be written as

$$p_i(\vec{x}_i) = \frac{1}{\sqrt{(2\pi)^k |\Sigma_i|}} \exp \left[-\frac{1}{2} (\vec{x}_i - \vec{\mu}_i)^T \Sigma_i^{-1} (\vec{x}_i - \vec{\mu}_i) \right] \quad (15)$$

where T indicates transposition and $|\Sigma_i|$ and Σ_i^{-1} are the determinant and the inverse of the covariance matrix, respectively. We can now define the likelihood of observing a sample $\vec{x}_i$ if it is constrained to the straight line $f[\vec{a}, \vec{b}]$

$$\begin{aligned} L_i(\vec{a}, \vec{b} | t) &:= p_i(f[\vec{a}, \vec{b}](t)) \\ &= \frac{1}{\sqrt{(2\pi)^k |\Sigma_i|}} \exp \left[-\frac{1}{2} \left(\vec{a}_i^T \Sigma_i^{-1} \vec{a}_i - \frac{t_i^2}{\sigma_i^2} - \frac{(t - t_i)^2}{\sigma_i^2} \right) \right] \end{aligned} \quad (16)$$

with the definitions

$$\vec{a}_i := \vec{a} - \vec{\mu}_i, \quad \sigma_i := \frac{1}{\sqrt{\vec{b}^T \Sigma_i^{-1} \vec{b}}}, \quad t_i := -\sigma_i^2 \vec{a}_i^T \Sigma_i^{-1} \vec{b}. \quad (17)$$

The most likely straight line can be found by maximizing this likelihood function with respect to $\vec{a}$ and $\vec{b}$. However, for any given straight line $f[\vec{a}, \vec{b}]$, the real value of t that generates $\vec{x}_i$ is unknown. Using

$$\begin{aligned} \max_{\vec{a}, \vec{b}} L_i(\vec{a}, \vec{b} | t) &\leq \max_{\vec{a}, \vec{b}} L_i(\vec{a}, \vec{b} | t = t_i) \\ &= \max_{\vec{a}, \vec{b}} \frac{1}{\sqrt{(2\pi)^k |\Sigma_i|}} \exp \left[-\frac{1}{2} \left(\vec{a}_i^T \Sigma_i^{-1} \vec{a}_i - \frac{t_i^2}{\sigma_i^2} \right) \right], \end{aligned} \quad (18)$$

and recalling that all $(\vec{X}_i)_{i \in I}$ are independent by definition, the likelihood function that needs to be maximized to find the desired straight line through all data points is given by

$$L(\vec{a}, \vec{b}) := \prod_{i \in I} L_i(\vec{a}, \vec{b} | t = t_i). \quad (19)$$

Analogously to standard least-square optimization algorithms, this is most efficiently done by minimizing

$$\begin{aligned} \mathcal{L} &:= -\log(L(\vec{a}, \vec{b})) \\ &= \frac{1}{2} \sum_{i \in I} \left[\log((2\pi)^k |\Sigma_i|) + \vec{a}_i^T \Sigma_i^{-1} \vec{a}_i - \frac{t_i^2}{\sigma_i^2} \right], \end{aligned} \quad (20)$$

where the terms $\log((2\pi)^k |\Sigma_i|)$ can be omitted as they are independent of $\vec{a}$ and $\vec{b}$. In *qspec*, the minimization is performed using the `scipy.optimize.minimize` function with the Newton conjugate gradient (Newton-CG) method [30,44]. For the fastest possible convergence, both the gradient and the Hessian of Eq. (20) are computed analytically, see Appendix A. The covariance matrix of the resulting $\vec{a}$ and $\vec{b}$ are calculated by numerically inverting the Hessian at the solution with `numpy.linalg.inv`. For two dimensions, the algorithm described here reproduces the fit performed with the York algorithm to machine precision. The estimated uncertainties deviate slightly in the third significant digit.

The *qspec* package also contains a class to set up King plots where usually the data vectors, consisting of isotope shifts or nuclear charge radii, are multiplied by mass factors to yield modified data vectors according to

Table 1

List of features implemented in `satlas2` or `qspect.models`. Please note that this feature list is only a momentary capture, as both packages are in active development, but it serves as a useful overview. Here (x, y) is a data point to be fitted, Δy is the uncertainty of the y value, f is the fit model, $\{p\}$ is the set of fit parameters and ρ returns the uncertainty Δy of a data point at runtime.

Feature	<code>satlas2</code>	<code>qspect.models</code>
Modular fit models	✓	✓
Quantum interference	—	✓
Hyperfine mixing	—	✓
Convolved lineshapes	—	✓
Shared parameters	✓	✓
Parameters as functions	✓	✓
Priors	✓	✓
Gaussian statistics	✓	✓
Poisson statistic	✓	—
Δy as a function	$\rho(f(x, \{p\}))$	$\rho(x, y, f(x, \{p\}), \{p\})$
Bayesian inference	✓	—

$$\vec{x}_{\text{mod}}^{AA'} = \mu^{AA'} \vec{x}^{AA'} = \frac{(M^A + m_e)(M^{A'} + m_e)}{M^{A'} - M^A} \vec{x}^{AA'}, \quad (21)$$

where M^A is the mass of the nucleus of the isotope with mass number A and m_e is the mass of the electron. The modified data vectors $\vec{x}_{\text{mod}}$ are then aligned on a straight line. In this case, the covariance matrices Σ_i are calculated from the masses and data vectors $\vec{x}^{AA'}$. This yields notable positive correlations between the vector components if the uncertainties of the masses are comparable or larger than those of the data vectors. Correlations between different data vectors are neglected. Improved or new data is gained from using the fitted straight line to calculate missing vector components j from one known component l

$$x_{\text{mod}}^j = a^j + \frac{b^j}{b^l} (x_{\text{mod}}^l - a^l). \quad (22)$$

The improved unmodified data and their covariance matrix is determined by inverting Eq. (21) and using Gaussian error propagation. A code example of a simple 2d King plot with Ca^+ isotopes is given in Code Listing 1, with data from [6,55,56].

5. Modular fit models

In `qspect.models`, a modular system for constructing fit models, with a focus on lineshape models, is implemented. This module can be compared to the `satlas2` package, which is used for fitting lineshape models to low- and high-statistics data [39,40]. In the backend, `qspect.models` extends the `scipy.optimize.curve_fit` method for fitting [30] while `satlas2` uses the `lmfit` package [45]. Both packages provide a modular system for constructing nonlinear functions, routines for fitting data with low statistics and options to alter the influence of individual parameters. While `qspect.models` implements models and features that are currently not available in `satlas2`, we also acknowledge that `satlas2` provides additional fit options such as Poisson-statistic maximum-likelihood fitting or Bayesian inference that are currently not available in `qspect.models`. A complete up-to-date feature comparison is given in Table 1.

There are several reasons that have lead to the development of `qspect.models`. At the beginning of its development, only the old version of `satlas` was available [39]. Therefore, a large performance boost was expected in `qspect.models`, as evidenced by the now available faster new version `satlas2` [40]. In `qspect.models`, new features and physical models are implemented that were/are not available in `satlas2`, such as consideration of quantum interference effects or convolutions of lineshape models. The dependencies of `qspect.models` are kept at a minimum of only `numpy` and `scipy`. Moreover, `qspect.models` is now used as the backend of an updated version of the `Tilda.PolliFit` framework, which provides a graphical user in-

terface and is already used for the analyses of online data [46–48]. In the following, the structure of `qspect.models` is described and a performance comparison to `satlas2` is given.

The basis for the modular system is the abstract `Model` class, from which all models inherit, see the class diagram in Fig. C.5. Composite models are created by passing a model to the constructor of another model. A basic example could be the creation of a hyperfine-structure model for a transition from a lower state $|J = 1/2\rangle$ to an upper state $|J' = 3/2\rangle$ in a system with nuclear spin $I = 1/2$

$$f(x) = p_0 \sum_{F \rightarrow F'} a(F, F') \text{Voigt}(x - \bar{x}, \Gamma, \sigma) + y_0 \quad (23)$$

$$\bar{x} = x_0 + \frac{A'}{2} [F'(F' + 1) - 9/2] - \frac{A}{2} [F(F + 1) - 3/2],$$

where Γ , σ , A , A' , $a(F, F')$, x_0 , p_0 and y_0 are parameters. This would correspond to the following lines of code.

```
1 from qspect.models import \
2     Voigt, Hyperfine, NPeak, Offset
3
4 I, J_l, J_u = 0.5, 0.5, 1.5 # = I, J, J'
5 hfs = Hyperfine(Voigt(), I, J_l, J_u)
6 f = Offset(NPeak(hfs, n_peaks=1))
```

Here, `Voigt` creates the peak shape model with parameters Γ (Lorentzian FWHM) and σ (Gaussian standard deviation) and `Hyperfine` introduces the sum in Eq. (23), the hyperfine-structure constants A , A' and the peak intensities $a(F, F')$. The `NPeak` model creates a sum over, in this case, one instance of the passed hyperfine-structure model at the position x_0 with intensity p_0 . The `Offset` model creates the offset parameter y_0 .

The lineshape model defined in Eq. (23) can be substantially modified. The Voigt profile can be replaced by other peak shapes. The chosen peak shape can also be numerically convolved with a second peak shape of choice. The multipole expansion of the hyperfine structure is implemented up to magnetic octupole terms. Quantum interference (QI) effects can be considered by replacing `Hyperfine` with `HyperfineQI`, see Eq. (10). This creates an extra parameter that encapsulates the detector geometry determining the QI effect. Hyperfine-induced mixing is considered in `HyperfineMixed`. The `NPeak` model can be used to create additional copies of the passed submodel with individual positions and intensities. `Offset` can divide the x -axis into intervals with individual offset functions, which can be any polynomial. All parameters of a model can be free, fixed, functions of other parameters, bounded to an interval or statistically constraint by a reference value with uncertainty, also known as a *prior* [40,49]. Additionally, simultaneous fits of multiple data sets with shared parameters are possible. In a future update, `qspect.models` can be combined with the `qspect.simulate` module, described in Sec. 6, to enable fitting of non-perturbative scattering rates. The computing time for a fit with shared parameters increases polynomially with the number of linked data sets. Therefore, the fit models have to be calculated efficiently. To test the speed of `qspect.models`, the benchmark test described in the documentation of `satlas2` [40] was adapted, which compares `satlas2` to `satlas` for fits to different numbers of data sets using shared parameters. There, a speedup factor of two to three orders of magnitude was found, depending on the number of data sets [40]. The benchmark test is a fit of Voigt profiles to the hyperfine structure spectrum of a $|J = 1/2\rangle \rightarrow |J' = 3/2\rangle$ transition with low Poisson statistics in a system with nuclear spin $7/2$. The hyperfine-structure parameters A , A' and B' as well as the parameters x_0 , Γ and σ were shared across data sets, compare Eq. (23). The magnetic octupole constant C' was fixed and set to zero. The low Poisson statistics were accounted for by a custom function

$$\rho(x, y, f(x, \{p\}), \{p\}) = \sqrt{f(x, \{p\})}, \quad (24)$$

which computes the uncertainty of the data during runtime, see Table 1. Fig. 2 shows a comparison of the computing times of `qspect.models`

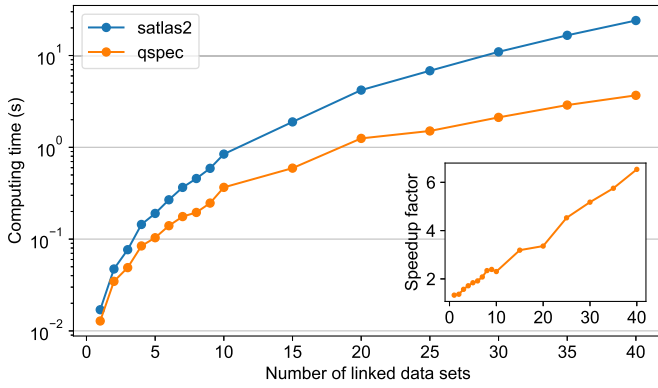


Fig. 2. Speed comparison of `qspec.models` with `satlas2` using an adaption of the benchmark test described in the documentation of `satlas2`. The computing time increases polynomially with the number of linked data sets. The ratio of the computing times of `satlas2` and `qspec.models` grows linearly. The benchmark test was performed with `satlas2` version 0.2.7 and `qspec` version 0.3.5 with an AMD Ryzen™ 7 7800X3D 8-Core Processor on Windows 11.

and `satlas2`. A linearly growing speedup factor of 1.2 to 7 was found when using `qspec.models` for simultaneously fitting 1 to 40 randomly generated data sets, respectively. The fit results agree well and only deviate in the third significant digit of the uncertainties. Please note that these computing speed results are only valid for this specific test configuration, as the speedup factor can depend on many parameters such as the used fit algorithm, the number of fit parameters, the number of linked parameters or the used test machine. Moreover, `satlas2` provides additional statistical measures for the fit results whose computing times we could not test. However, these results show that the basic framework of `qspec.models` meets the required performance standards and additional features, currently not implemented in `qspec.models`, such as Bayesian inference or maximum-likelihood fitting, can now be added to the robust basis. These will call for a separate benchmark test.

6. Interaction of light and matter

The lineshape of fluorescence spectra cannot always be described by an analytical model, especially when atomic states are optically pumped. In such cases it is desirable to simulate the evolution of the atomic state population in the laser field to understand the origin of the experimental lineshape and to investigate systematic frequency shifts when using a simple lineshape model. Moreover, the simulations can help to find the optimal experimental parameters in experiments that rely on optical pumping effects. All experiments that involve light-atom interactions can potentially benefit from such simulations. Some examples are saturation spectroscopy, Ramsey interferometry or quantum computing with atom/ion traps.

Python packages to simulate the interaction of light and matter already exist. However, they are often designed for a certain subspace of atomic systems and applications or require the user to derive the mathematical objects such as Hamiltonians or observable operators. For example, the `qutip` package is a powerful library of solvers for open quantum systems which takes the Hamiltonian of the system as an input [41,42]. For the simulation of laser spectroscopy experiments, where multiple isotopes with different hyperfine structures are of interest, a lot of work is required to set up all Hamiltonians. The `PyLCP` package facilitates this process, provides classes to set up lasers and external static electromagnetic fields and also allows to simulate the motional degrees of freedom, e.g., to simulate laser cooling [43]. While this package has a large range of applications, it still requires the user to construct a Hamiltonian from individual energy contributions and it currently does not support passing `numpy` arrays of velocities or laser detunings to simulate large samples. As a consequence, iterations over the samples and

multi-threading for better performance have to be implemented by the user.

The `qspec.simulate` module provides an intuitive class system, which is comparable to drawing an atomic level scheme, to define laser-atom interactions. A class diagram of `qspec.simulate` is depicted in Fig. C.6. Time-independent Hamiltonians in the interaction picture are generated automatically from the specified atom, lasers and environment. Initial populations of atomic states, initial velocities and laser detunings can be passed as `numpy` arrays. The backend utilizes multi-threading and is written in C++ for the best performance and long-term support. Solvers for the rate equations, the Schrödinger equation and the master equation are available. Additionally, a Monte-Carlo (MC) master equation solver can be used to also simulate the recoil motion of the atom in the laser field. In `qspec.simulate`, an atom is defined as a list of states $|i\rangle := |\gamma, S, L, J, I, F, m_F\rangle$ with eigen energies $h\nu_i$ that are connected through spontaneous decay rates Γ_{ij} . The eigen frequencies ν_i are either specified directly or calculated from Eq. (5) and the center-of-gravity frequency ν_0 of the fine-structure states $|\gamma, S, L, J\rangle$. Lasers are approximated as classical monochromatic plane waves, for which the user specifies the frequency $\tilde{\nu}_k$, intensity I_k and complex polarization vector $\vec{q}^k$. The atom and the lasers are combined into an `Interaction` object which generates the differential equations to be solved. For a complete list of definitions for the mathematical objects used in `qspec.simulate`, see Appendix B.

6.1. Hamiltonian

To simulate the coherent dynamics of the system, a Hamiltonian matrix is created in the basis of the atom, spanned by the user-specified states $|i\rangle$. The starting point is the time-dependent Hamiltonian

$$H = \sum_i \sigma_{ii} \hbar \omega_i + \sum_k \sum_{i,j} \sigma_{ij} \hbar \Omega_{ij}^k \cos(\tilde{\omega}_k t), \quad (25)$$

where $\sigma_{ij} := |i\rangle\langle j|$, $\omega = 2\pi\nu$ are circular frequencies and Ω_{ij}^k are the Rabi frequencies generated by laser k . This Hamiltonian describes the interaction of an atom with multiple classical laser fields. In this form, every laser drives all atomic transitions σ_{ij} . The `qspec.simulate` module identifies the transitions that fulfill the electric-dipole-transition rules and calculates the Rabi frequencies from the parameters of the lasers and the properties of the involved atomic states. The full set of conditions for a transition σ_{ij} to be considered is given by

$$|J_j - J_i| \leq 1, \quad 0 \nleftrightarrow 0 \quad (26)$$

$$|F_j - F_i| \leq 1, \quad 0 \nleftrightarrow 0 \quad (27)$$

$$|m_j - m_i| \leq 1, \quad 0 \nleftrightarrow 0 \quad \text{if } \Delta J, F = 0 \quad (28)$$

$$\Gamma_{ij} > 0 \quad (29)$$

$$I_k > 0 \quad (30)$$

$$(\vec{q}^k)_{m_j - m_i} \neq 0, \quad \nu_j > \nu_i \quad (31)$$

$$| |\nu_j - \nu_i| - \tilde{\nu}_k | \leq \Delta_{\max}, \quad (32)$$

with the Einstein A_{ji} coefficient and a user-defined cutoff frequency $\Delta_{\max}$ to ignore fast oscillations. Since the Hamiltonian still depends on the fast oscillating terms $\cos(\tilde{\omega}_k t)$, it is transformed into an interaction picture and the rotating-wave approximation is applied to frequencies of the order of $2\tilde{\omega}_k$. The transformation is constructed to produce a time-independent Hamiltonian whenever possible to avoid complex exponentials for faster computation. This is possible if and only if two or more lasers do not form loops in the multigraph generated by connecting the atomic states with the lasers. Since the transformation depends on the particular graph, which is only determined at runtime, no closed expression for the transformed Hamiltonian can be specified. The graph search algorithm that is used for the transformation is described in Fig. 3

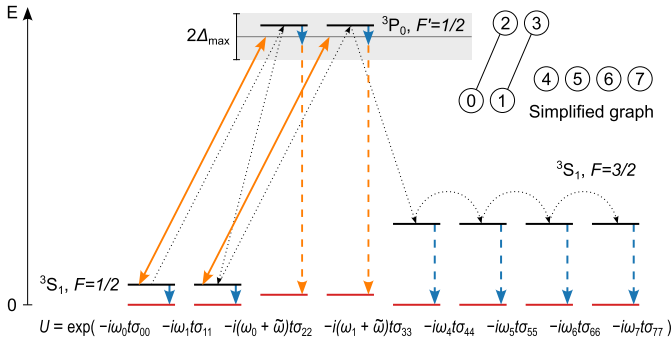


Fig. 3. Sketch of the algorithm used to transform the Hamiltonian into the interaction picture for a laser-driven $^3S_1 \rightarrow ^3P_0$ transition in a nuclear spin- $1/2$ system. The solid black and red lines depict the energies of the atomic states (diagonal elements of the Hamiltonian) before and after the transformation, respectively. The solid orange arrows depict the lasers, which are slightly detuned from resonance. The dashed arrows indicate the transformations applied to each state. The transformations applied to the $^3S_1 (F = 1/2)$ states (small dashed blue arrows) are carried over to the 3P_0 states due to the connection through the laser. Additionally, the 3P_0 states are shifted by the laser frequency $\tilde{\omega}$ (dashed orange arrow). The dotted black arrows show the path taken by the graph search algorithm. Note that this path changes, e.g., if the maximum detuning $\Delta_{\max}$ is large enough to bring the laser in resonance with the $F = 3/2 \rightarrow F' = 1/2$ transition.

based on a laser-driven $^3S_1 \rightarrow ^3P_0$ transition in a system with nuclear spin $1/2$. The transformed time-independent Hamiltonian of this example is given by [50]

$$H' := U^\dagger H U + i\hbar \frac{dU^\dagger}{dt} U \\ = \hbar(\omega_2 - \omega_0 - \tilde{\omega})(\hat{\sigma}_{22} + \hat{\sigma}_{33}) \\ + \hbar \frac{\Omega}{2}(\hat{\sigma}_{02} + \hat{\sigma}_{13}) + \hbar \frac{\Omega^*}{2}(\hat{\sigma}_{20} + \hat{\sigma}_{31}), \quad (33)$$

where the indexes correspond to the states indicated in the simplified graph in Fig. 3, $\tilde{\omega}$ and Ω are the laser and Rabi frequencies, respectively, U is the unitary transformation defined in Fig. 3, and $\omega_0 = \omega_1$ as well as $\omega_2 = \omega_3$ due to the absence of an external magnetic field. When the time-dependence cannot be eliminated completely in the presence of multiple lasers, the off-diagonal elements of the Hamiltonian will contain time-dependent Rabi frequencies, which oscillate with the frequency differences between the lasers. This is automatically detected by the algorithm and the Hamiltonian is adjusted accordingly.

6.2. Lindblad master equation

The complete coherent and dissipative population dynamics of the atomic states in the laser fields can be determined by solving the Lindblad master equation. It can be written as [50–52]

$$\frac{\partial \rho}{\partial t} = -\frac{i}{\hbar} [H, \rho] + \sum_{i,j} \Gamma_{ij} \mathcal{D}[\sigma_{ij}] \rho \quad (34)$$

$$\mathcal{D}[\sigma] \rho := \sigma \rho \sigma^\dagger - \frac{1}{2} (\sigma^\dagger \sigma \rho + \rho \sigma^\dagger \sigma), \quad (35)$$

where the coherent dynamics are described by the first term and the coupling to the vacuum by the second term. Note that $\Gamma_{ij} \neq 0 \Rightarrow \Gamma_{ji} = 0$ as spontaneous decay from $|i\rangle$ to $|j\rangle$ only occurs if $\omega_i > \omega_j$. In `qspect.simulate`, the differential Eq. (34) is solved numerically in matrix form using the C++ libraries `Boost.Numeric.Odeint` and `Eigen`. Large parameter spaces or numbers of samples can be explored by specifying frequency detunings, velocities (Doppler shifts) or initial density matrices (state populations) as numpy arrays, which are addressed in parallel by the C++ backend. Hence, a user is neither required to write any loops in Python nor to implement parallel computing. The performance of the master equation solvers in `qspect` and `qutip` for a single

parameter sample are comparable and mainly depend on the chosen integration algorithm, integration steps and error tolerances.

A MC master equation solver is available, following the approach described in [53], which is also implemented in `qutip` [41]. The basic idea is to solve the Schrödinger equation with a modified non-hermitian Hamiltonian that leaks population into the environment according to the dissipation rates Γ_{ij} . Whenever the norm of a simulated state vector $|\psi_s\rangle$, where s numerates the samples, falls below a randomly generated fraction of unity, the state vector “collapses”, destroying all superpositions and resetting the system to the collapsed state. Averaging the resulting state vectors for $n \rightarrow \infty$ samples yields the density matrix described by the master equation

$$\rho_{ij} = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{s=1}^n \langle i | \psi_s \rangle \langle \psi_s | j \rangle. \quad (36)$$

The MC method is particularly useful for large atomic systems since its complexity only increases linearly with the number of atomic states as opposed to the quadratic growth of the master equation. In `qspect.simulate`, the MC master equation additionally can be used to simulate the motional degrees of freedom of the atom. Whenever the atom spontaneously decays, the photon momenta of the lasers leading from the initial to the excited state and the recoil from the spontaneous decay are added to the momentum of the atom. Note that in this method, the momentum of the atom only changes upon spontaneous decays but not during any coherent interaction.

6.3. Rate equations

A simplification of the master equation are the rate equations which omit all coherent dynamics. In laser spectroscopy, the use of rate equations is often sufficient due to how precise the state populations need to be known or due to the width of the velocity distribution of the atomic ensemble, which can average out coherent effects. The rate equations implemented in `qspect.simulate` are given by

$$\frac{\partial \rho_{ii}}{\partial t} = \sum_j \left[\left(\sum_k R_{ij}^k \right) (\rho_{jj} - \rho_{ii}) + \Gamma_{ij} \rho_{jj} - \Gamma_{ji} \rho_{ii} \right], \quad (37)$$

where R_{ij}^k are the stimulated emission/absorption rates for laser k , which depend on the frequency, intensity and polarization of the laser, Γ_{ij} are the spontaneous decay rates from Eq. (34) and $\rho_{ij} = 0$ for $i \neq j$.

6.4. Differential scattering rate

From the density matrices determined through Eq. (34), (36) and (37), a differential scattering rate into the solid angle Ω can be calculated that, in case of the coherent approaches, also considers quantum interference (QI) effects. The differential scattering rate is given by [18]

$$\frac{d\Gamma}{d\Omega}(\vec{k}_{sc}) := \text{Tr} \left(\rho \mathcal{D}(\vec{k}_{sc}) \right) \\ = \frac{1}{4\pi} \sum_{\vec{q} \in \mathcal{P}(\vec{k}_{sc})} \sum_{f \in \mathcal{F}} \sum_{i,j \in \mathcal{I}_f} \sqrt{\Gamma_{fi} \Gamma_{fj}} \frac{(\vec{q} \cdot \vec{d}_{fi})(\vec{d}_{fj}^* \cdot \vec{q}^*)}{|\vec{d}_{fi}| |\vec{d}_{fj}^*|} \rho_{ij}^*, \quad (38)$$

where $\mathcal{D}(\vec{k}_{sc})$ is a detection operator, $\mathcal{P}(\vec{k}_{sc})$ is a basis set of normalized polarization vectors for scattered photons with direction $\vec{k}_{sc}$, $(\mathcal{I}_f)_{f \in \mathcal{F}}$ and $\mathcal{F}$ are sets of excited and final states between which spontaneous decay can occur and $\vec{d}$ are the dipole transition strengths, see Eq. (B.7). Limiting the three sums to detectable spontaneous decay and integrating over the solid angle of detection yields a photon detection rate.

Equation (38) reproduces the perturbative differential scattering rate derived by Brown et al. [22], see Eq. (10), that considers QI effects in closed electronic transitions in the case of small laser intensities, such that $\Omega_{ij} \ll \Gamma_{ij}$, linear polarization and equilibrium population. These conditions are rarely met exactly in the experiment as in most real

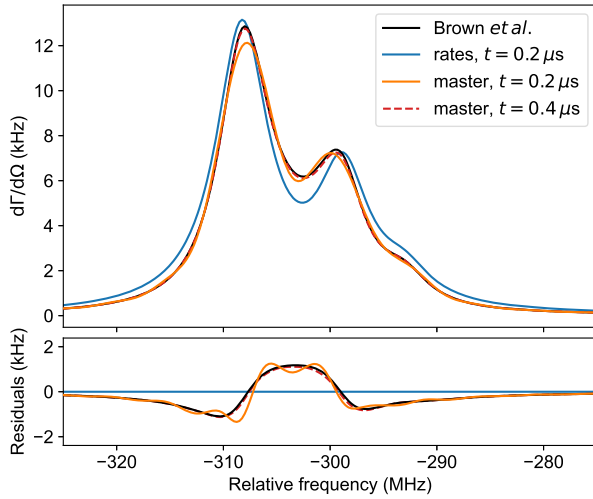


Fig. 4. Simulated differential scattering rate of the $^2S_{1/2}(F=2) \rightarrow ^2P_{3/2}$ transitions in ^7Li in the direction of the linear polarization vector of a spectroscopy laser with an intensity of $1 \mu\text{W}/\text{mm}^2$. The perturbative scattering rate, which includes quantum interference (QI) effects derived by Brown et al. [22] (black) is compared to the solutions of the rate (blue) and master (orange) equations after an integration time of $0.2 \mu\text{s}$. While the rate equations yield a sum of Lorentzians with averaged out coherent and QI effects, the master equation approach shows the intensity modulations of the coherent Rabi oscillations and converges to the perturbative QI approach for later times (red dashed). A small difference comes from optical population transfer into the $F=1$ ground state. This plot reproduces Fig. 2 of [22]. The shown data was generated using Code Listing 2.

atomic systems, especially if they have a hyperfine structure, optical pumping and population transfer will occur already within short interaction times and laser powers on the $\mu\text{W}/\text{mm}^2$ level. A comparison of differential scattering rates of the $^2S_{1/2} \rightarrow ^2P_{3/2}$ transition in ^7Li is shown in Fig. 4. The scattering rates were determined with Eq. (38) using the master and the rate equation approaches as well as with the perturbative scattering rate from [22] for detection at 0° angle relative to the polarization vector of absorbed photons. A code example that produces the data shown in Fig. 4 is given in Code Listing 2, using data from [22,57]. The shown plot reproduces Fig. 2 from [22] and additionally shows the solution of the master equation approach after an integration time of 0.2 and $0.4 \mu\text{s}$, demonstrating the need of the non-perturbative approach if the system equilibrium is not reached yet.

7. Summary and outlook

The Python package `qspect` provides a variety of classes and functions to facilitate the analysis and simulation workflow in laser spectroscopy experiments. In this article, a multidimensional linear regression algorithm, modular fit models and light-matter simulations are introduced. The maximum-likelihood fit of a straight line in multiple dimensions constitutes a generalization of the linear regression algorithm introduced by York et al. [37]. The system of modular fit models provides a basis for combining and linking different fit models, parameters and data sets and can be easily expanded with new models. The simulation module for coherent light-matter interactions offers an intuitive and user-friendly interface without the need to specify a Hamiltonian or other complicated mathematical objects.

The `qspect` package is built to be as general as possible within the given set of physical conditions. An expansion of `qspect` with new features and further generalization for a larger scope of applications is planned. For example, the simulation module can be used to define a new lineshape model for fitting, which can account for the population dynamics of an atom in a laser field. The simulation module itself can be expanded, e.g., by accounting for finite laser linewidths, adding molecular systems or al-

lowing higher-order multipole transitions or time-dependent velocities and laser parameters.

CRediT authorship contribution statement

P. Müller: Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **W. Nörtershäuser:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank K. König, B. Maaß, E. Matthews, K. Mohr, L. Renth and J. Spahn for testing the `qspect` package and their valuable feedback. We acknowledge support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under - Projektnummer 279384907 - SFB 1245, and the German Federal Ministry of Education and Research BMBF under Contract Nos. 05P21RDFN1 and 05P21RDCI1. P. M. acknowledges support from HGS-HIRE.

Appendix A. Gradient and Hessian of the negated log-likelihood function

For the minimization of the negative log-likelihood function given by Eq. (20), its gradient vector and Hessian are calculated analytically. Both are used in the Newton conjugate gradient (Newton-CG) method [30,44] and the Hessian is also used to estimate the covariance matrix of the resulting parameters. We define

$$\tilde{a}_{ij} := (\Sigma_i^{-1} \tilde{a}_i)_j, \quad \tilde{b}_{ij} := \left(\Sigma_i^{-1} \tilde{b} \right)_j. \quad (\text{A.1})$$

Then the components of the gradient vector of Eq. (20) are given by

$$\frac{\partial}{\partial a_j} \mathcal{L} = \sum_{i \in I} \left[\tilde{a}_{ij} - \frac{t_i}{\sigma_i^2} \tilde{b}_{ij} \right] \quad (\text{A.2})$$

$$\frac{\partial}{\partial b_j} \mathcal{L} = \sum_{i \in I} \left[\frac{t_i}{\sigma_i^2} \tilde{a}_{ij} - \frac{t_i^2}{\sigma_i^4} \tilde{b}_{ij} \right], \quad (\text{A.3})$$

and the components of the Hessian are given by

$$\frac{\partial^2}{\partial a_j \partial a_l} \mathcal{L} = \sum_{i \in I} \left[(\Sigma_i^{-1})_{jl} - \frac{\tilde{b}_{ij} \tilde{b}_{il}}{\sigma_i^2} \right] \quad (\text{A.4})$$

$$\begin{aligned} \frac{\partial^2}{\partial b_j \partial b_l} \mathcal{L} = \sum_{i \in I} \left[\frac{t_i^2}{\sigma_i^4} (\Sigma_i^{-1})_{jl} - \frac{1}{\sigma_i^2} \tilde{a}_{ij} \tilde{a}_{il} - 4 \frac{t_i^2}{\sigma_i^6} \tilde{b}_{ij} \tilde{b}_{il} \right. \\ \left. + 2 \frac{t_i}{\sigma_i^4} (\tilde{a}_{ij} \tilde{b}_{il} + \tilde{b}_{ij} \tilde{a}_{il}) \right] \end{aligned} \quad (\text{A.5})$$

$$\frac{\partial^2}{\partial a_j \partial b_l} \mathcal{L} = \sum_{i \in I} \left[-\frac{t_i^2}{\sigma_i^2} (\Sigma_i^{-1})_{jl} + \frac{2 t_i}{\sigma_i^4} \tilde{b}_{ij} \tilde{b}_{il} - \frac{1}{\sigma_i^2} \tilde{a}_{ij} \tilde{b}_{il} \right]. \quad (\text{A.6})$$

Appendix B. Definitions for `qspect.simulate`

The following definitions are used to calculate light-matter interactions in `qspect.simulate` [22,54]

$$\begin{aligned} v_i &= v_{J_i} + v_{(J_i J_i)(J_i J_i)}^{F_i} - m_{F_i} g_{F_i} \frac{\mu_B}{h} B \\ g_F &= g_J \frac{F(F+1) + J(J+1) - I(I+1)}{2F(F+1)} \end{aligned} \quad (\text{B.1})$$

$$+ g_I \frac{\mu_N}{\mu_B} \frac{F(F+1) - J(J+1) + I(I+1)}{2F(F+1)}$$

$$g_J = -\frac{J(J+1) + L(L+1) - S(S+1)}{2J(J+1)}$$

$$+ g_s \frac{J(J+1) - L(L+1) + S(S+1)}{2J(J+1)},$$

$$a_{ij} = (-1)^{I+J_j+F_i+1} \sqrt{2F_i+1} \sqrt{2J_j+1}$$

$$\times \langle F_i m_i; 1 m_j - m_i | F_j m_j \rangle \begin{Bmatrix} J_j & J_i & 1 \\ F_i & F_j & I \end{Bmatrix}$$

$$\tilde{\Gamma}_{ij} = \begin{cases} A_{ji} & \nu_i < \nu_j \\ A_{ij} & \text{else} \end{cases}$$

$$\Gamma_{ij} = a_{ij}^2 A_{ji}$$

$$d_{ij} = \sqrt{\frac{3\epsilon_0 c^3 \hbar \tilde{\Gamma}_{ij}}{8\pi^2 |\nu_j - \nu_i|^3}} a_{ij}$$

$$(B.2) \quad M_{ij}^k = \begin{cases} 1 & \text{Eq. (26) - (32) fulfilled} \\ 0 & \text{else} \end{cases} \quad (B.8)$$

$$(B.3) \quad \Omega_{ij}^k = \sqrt{\frac{2I_k}{\epsilon_0 c}} \frac{d_{ij}(\vec{q}^k)_{m_j-m_i}}{\hbar} M_{ij}^k \quad (B.9)$$

$$(B.4) \quad R_{ij}^k = \frac{|\Omega_{ij}^k|^2 \tilde{\Gamma}_{ji}}{(2\pi)^2 (\tilde{\nu}_k - |\nu_j - \nu_i|)^2 + \frac{1}{4} \tilde{\Gamma}_{ji}^2}, \quad (B.10)$$

where i and j enumerate individual quantum states, ν_i are the eigen frequencies, m_i are the z-projection quantum numbers of the total angular momentum quantum numbers F_i , B is the magnetic flux density, g are the Landé factors, μ_N and μ_B are the nuclear and Bohr magneton, respectively, $\langle \dots | \dots \rangle$ are the Clebsch-Gordan coefficients, $\{ : : : \}$ are the Wigner-6j symbols, $\vec{q}^k$ is a normalized complex laser polarization vector of laser k , A_{ji} are Einstein coefficients, I_k are the laser intensities, ϵ_0 is the vacuum permittivity and c is the vacuum speed of light.

Appendix C. Class diagrams of `qspec.models` and `qspec.simulate`

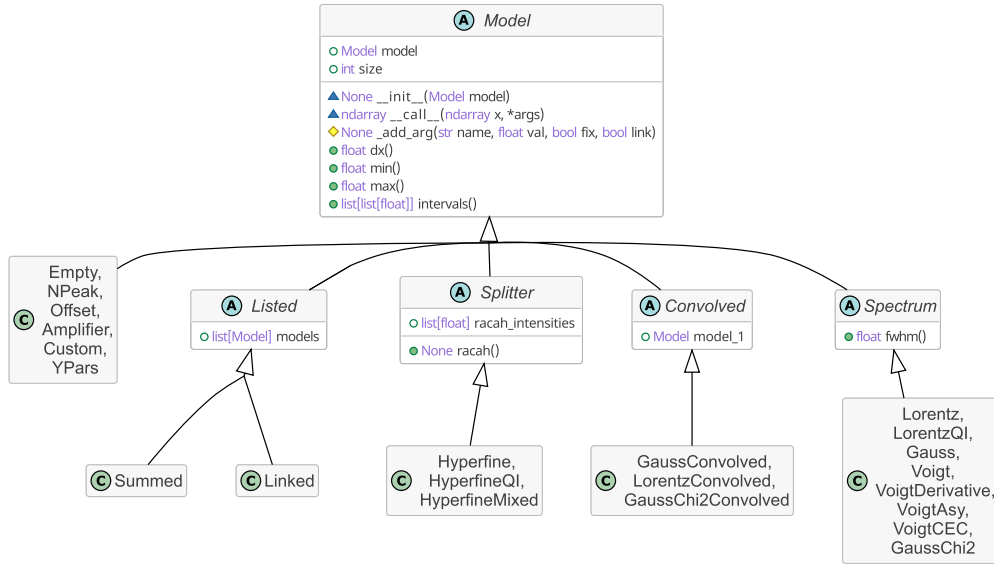


Fig. C.5. Class inheritance diagram of `qspec.models`. All models inherit from the `Model` class. The abstract `Listed` class enables the combination of models to link parameters or sum models. The `Splitter` class is an abstract class for Hyperfine structure models. `Convolved` allows numerical convolution of two models and `Spectrum` is a generic lineshape/peak model.

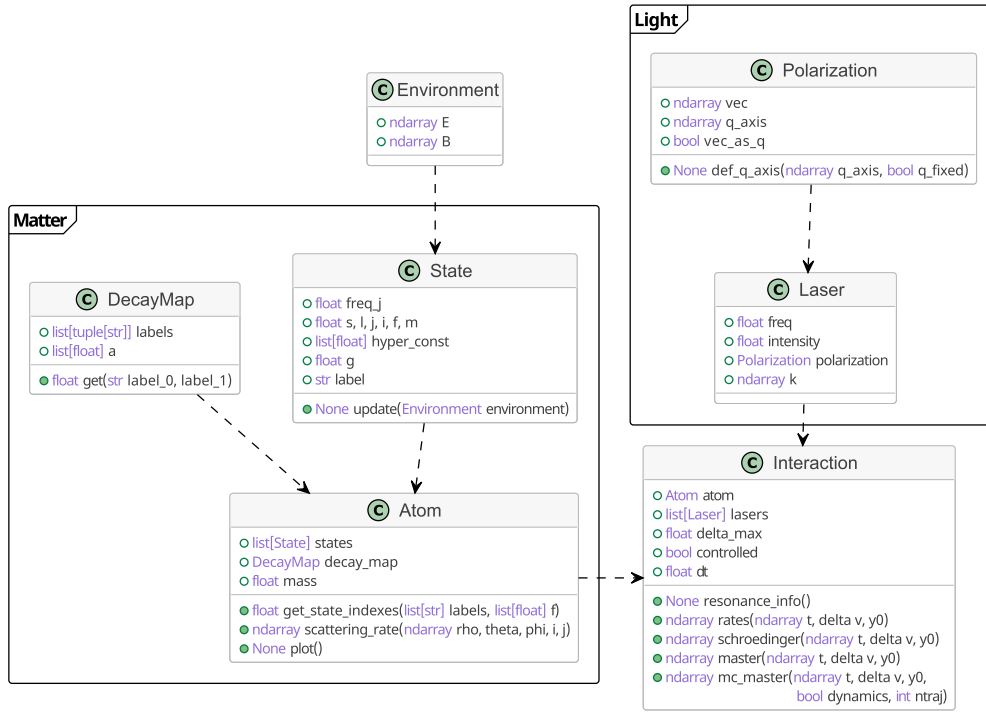


Fig. C.6. Class diagram of `qspec.simulate`. A `State` represents a single quantum mechanical state. A `DecayMap` object links multiple states with the same user-defined labels through spontaneous decay. A list of all states and the `DecayMap` object form an `Atom`. A `Laser` consists of a frequency, an intensity, a `Polarization` and a directional vector. The `Atom` and a list of lasers is combined in an `Interaction`. Rate, Schrödinger, master and a Monte-Carlo (MC) master equation solver can be accessed directly from the `Interaction`. An `Environment` can be defined to alter the quantum state energies, but is not fully implemented yet and currently only supports linear Zeeman shifts. Note that the list of shown member functions is not complete for better clarity.

Appendix D. Code examples of `qspec`

Code Listing 1: Example code to perform a 2d King-plot analysis with Ca^+ isotopes. In this example two transitions, named D1 and D2 line, are plotted against each other for the stable Ca^+ isotopes [6]. Transition frequencies are specified as absolute values (line 14-21) so that `qspec` can calculate isotope shifts at runtime, which facilitates reassignments of reference isotopes (line 28-29). The fitted parameters of the straight line are used to determine the isotope shifts of the D1 line in $^{50,52}\text{Ca}^+$ by inserting the isotope shifts of the D2 line (line 42-43) [55]. Finally the results are printed to the console output (line 45-47). Masses are taken from the Atomic Mass Evaluation (AME) 2020 [56]. Note that this example can be extended to an n -dimensional fit simply by adding more columns/observables to `x_abs`. If absolute values are unknown, the isotope shifts have to be passed to the `king.fit` routine in the same format as `x_abs` and need to concur with the specified lists of (reference) isotopes.

```

1 import qspec as qs
2
3 # The mass numbers of the Ca isotopes.
4 a = [40, 42, 43, 44, 46, 48, 50, 52]
5
6 # The masses of the isotopes (u, AME 2020).
7 m = [(39.962590850, 22e-9), (41.958617780, 159e-9), # 40Ca, 42Ca
8      (42.958766381, 244e-9), (43.955481489, 348e-9), # 43Ca, 44Ca
9      (45.953687726, 2398e-9), (47.952522654, 18e-9), # 46Ca, 48Ca
10     (49.957499215, 1.7e-6), (51.963213646, 720e-9)] # 50Ca, 52Ca
11
12 # Use absolute values given in the shape (#isotopes, #observables, 2).
13 # Frequencies for the (D1, D2) lines (MHz).
14 x_abs = [(755222765.66, 0.10), (761905012.53, 0.11)], # 40Ca
15         [(755223191.15, 0.10), (761905438.57, 0.10)], # 42Ca
16         [(755223443.57, 0.30), (761905691.89, 0.17)], # 43Ca
17         [(755223614.66, 0.10), (761905862.62, 0.09)], # 44Ca
18         [(755224063.27, 0.33), (761906311.60, 0.57)], # 46Ca
19         [(755224471.12, 0.10), (761906720.11, 0.11)], # 48Ca
20         [(0., 0.), (0., 0.)], # 50Ca
21         [(0., 0.), (0., 0.)] # 52Ca
22
23 # Construct a King object. Optionally specify 'x_abs' here
24 # to omit isotope shifts when fitting. 20 electron masses are subtracted
25 # to perform the King plot analysis with the nuclear masses.
26 king = qs.King(a=a, m=m, x_abs=x_abs, subtract_electrons=20)

```

```

27
28 a_fit = [42, 43, 44, 46, 48] # Choose the isotopes to fit.
29 a_ref = [40, 48, 42, 40, 44] # Choose individual reference isotopes.
30
31 # Do a simple 2d King plot. The 'mode' keyword is only used for the axis labels.
32 popt, pcov = king.fit(a_fit, a_ref, mode='shifts')
33 # >>> f(x) = (177.3 u MHz) + 1.00068 * x
34
35 a_unknown = [50, 52] # Specify the unknown isotopes
36 a_unknown_ref = [40, 40] # and their references.
37
38 # Specify the isotope shifts of the D2 line.
39 y = [(1969.2, 5.6), (2219.2, 7.0)]
40
41 # Calculate the isotope shifts of the D1 line and their covariances.
42 x, cov, cov_stat = king.get_unmodified(
43     a_unknown, a_unknown_ref, y, axis=1, show=True, mode='shifts')
44
45 for iso, c in zip(a_unknown, cov):
46     qs.printh(f'\n{iso}Ca+:') # Print colored headline.
47     qs.print_cov(c) # Print color-coded covariance matrix.

```

Code Listing 2: Example code to produce the data shown in Fig. 4. Lists of the magnetic substates of the ground (s) and the excited state (p) are created (line 10-13), connected in a DecayMap (line 15) and used to define a ^7Li atom (line 16). A laser is defined with an intensity of $1 \mu\text{W}/\text{mm}^2$ and π -polarized light (line 18-20). An Interaction object combines the atom and a list of lasers (line 22). The rate (line 29) and master (line 32, 35) equation solvers are called directly from the Interaction object for an array of frequency detunings δ . The scattering rate for the different detunings defined in Eq. (38) is calculated directly from the array of density matrices ρ or state populations n (line 30, 33, 36). An extended version of the perturbative scattering rate derived by Brown et al. [22] is calculated using a ScatteringRate object (line 38-39). Transition frequencies and hyperfine structure constants are taken or calculated from the data in [22]. The Einstein coefficient is taken from [57].

```

1 import numpy as np
2 import qspect.simulate as sim
3
4 f_sp = 446810183.163 # Transition frequency (MHz)
5 a_sp = 36.891 # Einstein coefficient (rad MHz)
6
7 s_hyper = [401.75825] # HFS constants (MHz)
8 p_hyper = [-3.055038, -0.29670]
9
10 s = sim.construct_electronic_state(
11     0., s=0.5, l=0, j=0.5, i=1.5, hyper_const=s_hyper, label='s')
12 p = sim.construct_electronic_state(
13     f_sp, s=0.5, l=1, j=1.5, i=1.5, hyper_const=p_hyper, label='p')
14
15 decay = sim.DecayMap(labels=[('s', 'p')], a=[a_sp])
16 li7 = sim.Atom(s + p, decay)
17
18 intensity = 1. # uW /mm**2
19 polarization = sim.Polarization([0, 1, 0]) # Linear polarization
20 laser = sim.Laser(f_sp, intensity, polarization)
21
22 inter = sim.Interaction(li7, [laser])
23 inter.controlled = True # Error controlled integrator
24
25 t = 0.2 # Integration time (us)
26 delta = np.linspace(-325, -275, 201) # Frequency detunings (MHz)
27 theta, phi = 0., 0. # Angles from z-axis in x- and y-direction (rad)
28
29 n = inter.rates(t, delta) # Rate equations, 0.2 us
30 y_rates = li7.scattering_rate(n, theta, phi, as_density_matrix=False)
31
32 rho = inter.master(t, delta) # Master equation, 0.2 us
33 y_master = li7.scattering_rate(rho, theta, phi)
34
35 rho = inter.master(0.4, delta) # Master equation, 0.4 us
36 y4_master = li7.scattering_rate(rho, theta, phi)
37
38 sr = sim.ScatteringRate(li7, laser=laser)
39 y_brown = sr.generate_y(delta, theta, phi)[:, 0, 0] # Brown et al.

```

Data availability

The algorithms described within this article are implemented in the open-source Python package *qspect*, available per request or online at <https://pypi.org/project/qspect/> and <https://github.com/patmlr/qspect>. Data shown in the article is either cited or generated using the *qspect* package.

References

- [1] R. Neugart, Laser spectroscopy on mass-separated radioactive beams, *Nucl. Instrum. Methods Phys. Res. B* 152 (1981) 165–175, [https://doi.org/10.1016/0029-554X\(81\)90902-2](https://doi.org/10.1016/0029-554X(81)90902-2).
- [2] E.W. Otten, Nuclear radii and moments of unstable isotopes, in: D.A. Bromley (Ed.), *Treatise on Heavy Ion Science*, Springer, Boston, 1989, pp. 517–638.
- [3] K. Blaum, J. Dilling, W. Nörtershäuser, Precision atomic physics techniques for nuclear physics with radioactive beams, *Phys. Scr. T* 152 (2013) 014017, <https://doi.org/10.1088/0031-8949/2013/T152/014017>.
- [4] P. Campbell, I.D. Moore, M.R. Pearson, Laser spectroscopy for nuclear structure physics, *Prog. Part. Nucl. Phys.* 86 (2016) 127–180, <https://doi.org/10.1016/j.pnpnp.2015.09.003>.
- [5] J. Ullmann, Z. Andelkovic, C. Brandau, A. Dax, W. Geithner, C. Geppert, C. Gorges, M. Hammen, V. Hannen, S. Kaufmann, K. König, Y.A. Litvinov, M. Lochmann, B. Maaß, J. Meisner, T. Murböck, R. Sánchez, M. Schmidt, S. Schmidt, M. Steck, T. Stöhler, R.C. Thompson, C. Trageser, J. Vollbrecht, C. Weinheimer, W. Nörtershäuser, High precision hyperfine measurements in Bismuth challenge bound-state strong-field QED, *Nat. Commun.* 8 (2017) 15484, <https://doi.org/10.1038/ncomms15484>.
- [6] P. Müller, K. König, P. Imgram, J. Krämer, W. Nörtershäuser, Collinear laser spectroscopy of Ca^{+} : solving the field-shift puzzle of the $4s^2S_{1/2} \rightarrow 4p^2P_{1/2,3/2}$ transitions, *Phys. Rev. Res.* 2 (4) (2020) 043351, <https://doi.org/10.1103/PhysRevResearch.2.043351>.
- [7] J.J. Hudson, D.M. Kara, I.J. Smallman, B.E. Sauer, M.R. Tarbutt, E.A. Hinds, Improved measurement of the shape of the electron, *Nature* 473 (7348) (2011) 493–496, <https://doi.org/10.1038/nature10104>.
- [8] M.S. Safronova, D. Budker, D. DeMille, D.F.J. Kimball, A. Derevianko, C.W. Clark, Search for new physics with atoms and molecules, *Rev. Mod. Phys.* 90 (2) (2018) 025008, <https://doi.org/10.1103/RevModPhys.90.025008>.
- [9] B.J. Bloom, T.L. Nicholson, J.R. Williams, S.L. Campbell, M. Bishof, X. Zhang, W. Zhang, S.L. Bromley, J. Ye, An optical lattice clock with accuracy and stability at the 10^{-18} level, *Nature* 506 (7486) (2014) 71–75, <https://doi.org/10.1038/nature12941>.
- [10] A.D. Ludlow, M.M. Boyd, J. Ye, E. Peik, P.O. Schmidt, Optical atomic clocks, *Rev. Mod. Phys.* 87 (2) (2015) 637–701, <https://doi.org/10.1103/RevModPhys.87.637>.
- [11] S.L. Campbell, R.B. Hutson, G.E. Marti, A. Goban, N. Darkwah Oppong, R.L. McNally, L. Sonderhouse, J.M. Robinson, W. Zhang, B.J. Bloom, J. Ye, A Fermi-degenerate three-dimensional optical lattice clock, *Science* 358 (6359) (2017) 90–94, <https://doi.org/10.1126/science.aam5538>.
- [12] J.I. Cirac, P. Zoller, Quantum computations with cold trapped ions, *Phys. Rev. Lett.* 74 (20) (1995) 4091–4094, <https://doi.org/10.1103/PhysRevLett.74.4091>.
- [13] H. Häffner, C. Roos, R. Blatt, Quantum computing with trapped ions, *Phys. Rep.* 469 (4) (2008) 155–203, <https://doi.org/10.1016/j.physrep.2008.09.003>.
- [14] N. Friis, O. Marty, C. Maier, C. Hempel, M. Holzäpfel, P. Jurcevic, M.B. Plenio, M. Huber, C. Roos, R. Blatt, B. Lanyon, Observation of entangled states of a fully controlled 20-qubit system, *Phys. Rev. X* 8 (2) (2018) 021012, <https://doi.org/10.1103/PhysRevX.8.021012>.
- [15] K. König, J. Krämer, C. Geppert, P. Imgram, B. Maaß, T. Ratajczyk, W. Nörtershäuser, A new collinear apparatus for laser spectroscopy and applied science (COALA), *Rev. Sci. Instrum.* 91 (8) (2020) 081301, <https://doi.org/10.1063/1.50010903>.
- [16] R. Neugart, J. Billowes, M.L. Bissell, K. Blaum, B. Cheal, K.T. Flanagan, G. Neyens, W. Nörtershäuser, D.T. Yordanov, Collinear laser spectroscopy at ISOLDE: new methods and highlights, *J. Phys. G, Nucl. Part. Phys.* 44 (6) (2017) 064002, <https://doi.org/10.1088/1361-6471/aa6642>.
- [17] K. Mohr, R. Sánchez, W. Nörtershäuser, Z. Andelkovic, V. Hannen, E.O. Hanu, F. Herfurth, R. Heß, M. Horst, P. Imgram, K. König, C. Krantz, M. Lestinsky, Y.A. Litvinov, E. Menz, P. Müller, J. Palmes, S. Rausch, T. Ratajczyk, L. Renth, J. Rossbach, R.S. Sidhu, F. Sommer, J. Spahn, N. Stallkamp, K. Ueberholz, G. Vorobjev, C. Weinheimer, D. Zisis, A setup to study atomic state population dynamics and optical polarization at CRYRING@ESR, *arXiv:2408.12368*, 2024.
- [18] G.W.F. Drake, *Springer Handbook of Atomic, Molecular, and Optical Physics*, Springer, Cham, 2023.
- [19] R. Loudon, *The Quantum Theory of Light*, Oxford University Press, Oxford, 2000.
- [20] W. Demtröder, *Laser Spectroscopy*, Springer Berlin Heidelberg, 2008.
- [21] W.R. Johnson, K.T. Cheng, D.R. Plante, Hyperfine structure of 2^3P levels of heliumlike ions, *Phys. Rev. A* 55 (4) (1997) 2728–2742, <https://doi.org/10.1103/PhysRevA.55.2728>.
- [22] R.C. Brown, S. Wu, J.V. Porto, C.J. Sansonetti, C.E. Simien, S.M. Brewer, J.N. Tan, J.D. Gillaspay, Quantum interference and light polarization effects in unresolvable atomic lines: application to a precise measurement of the $6^7\text{Li } D_2$ lines, *Phys. Rev. A* 87 (3) (2013) 032504, <https://doi.org/10.1103/PhysRevA.87.032504>.
- [23] A. Beyer, L. Maisenbacher, A. Matveev, R. Pohl, K. Khabarova, A. Grinin, T. Lamour, D.C. Yost, T.W. Hänsch, N. Kolachevsky, T. Udem, The Rydberg constant and proton size from atomic hydrogen, *Science* 358 (6359) (2017) 79–85, <https://doi.org/10.1126/science.aah6677>.
- [24] C. Schwartz, Theory of hyperfine structure, *Phys. Rev.* 97 (2) (1955) 380–395, <https://doi.org/10.1103/PhysRev.97.380>.
- [25] V.M. Shabaev, D.A. Glazov, A.M. Ryzhkov, C. Brandau, G. Plunien, W. Quint, A.M. Volchkova, D.V. Zinenko, Ground-state g factor of highly charged ^{229}Th ions: an access to the M1 transition probability between the isomeric and ground nuclear states, *Phys. Rev. Lett.* 128 (4) (2022) 043001, <https://doi.org/10.1103/PhysRevLett.128.043001>.
- [26] P.F. Liao, R.R. Freeman, R. Panock, L.M. Humphrey, Hyperfine-induced singlet-triplet mixing in ^3He , *Opt. Commun.* 34 (2) (1980) 195–198, [https://doi.org/10.1016/0030-4018\(80\)90013-9](https://doi.org/10.1016/0030-4018(80)90013-9).
- [27] K. Mohr, A. Buß, Z. Andelkovic, V. Hannen, M. Horst, P. Imgram, K. König, B. Maaß, W. Nörtershäuser, S. Rausch, R. Sánchez, C. Weinheimer, Collinear laser spectroscopy of helium-like $^{11}\text{B}^{3+}$, *Atoms* 11 (1) (2023) 11, <https://doi.org/10.3390/atoms1101011>.
- [28] N.J. Stone, Table of recommended nuclear magnetic dipole moments, <https://inis.iaea.org/search/51052833>, 2019.
- [29] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. Del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, *Nature* 585 (7825) (2020) 357–362, <https://doi.org/10.1038/s41586-020-2649-2>.
- [30] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, I. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nat. Methods* 17 (3) (2020) 261–272, <https://doi.org/10.1038/s41592-019-0686-2>.
- [31] A. Meurer, C.P. Smith, M. Paprocki, O. Čertík, S.B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J.K. Moore, S. Singh, T. Rathnayake, S. Vig, B.E. Granger, R.P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M.J. Curry, A.R. Terrel, S. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimman, A. Scopatz, SymPy: symbolic computing in Python, *PeerJ Comput. Sci.* 3 (2017) e103, <https://doi.org/10.7717/peerj-cs.103>.
- [32] J.D. Hunter, Matplotlib: a 2D graphics environment, *Comput. Sci. Eng.* 9 (3) (2007) 90–95, <https://doi.org/10.1109/MCSE.2007.55>.
- [33] P. Müller, *qspect*, <https://patmlr.github.io/qspect/>, 2024.
- [34] P. Müller, Laserspectroscopic determination of the nuclear charge radius of ^{13}C , Doctoral thesis (PhD thesis), Technische Universität Darmstadt, 2024, <https://doi.org/10.26083/tuprints-00026746>.
- [35] S.L. Kaufman, High-resolution laser spectroscopy in fast beams, *Opt. Commun.* 17 (3) (1976) 309–312, [https://doi.org/10.1016/0030-4018\(76\)90267-4](https://doi.org/10.1016/0030-4018(76)90267-4).
- [36] W.H. King, *Isotope Shifts in Atomic Spectra*, Springer, New York, 1984.
- [37] D. York, N.M. Evensen, M.L. Martínez, J. de Basabe Delgado, Unified equations for the slope, intercept, and standard errors of the best straight line, *Am. J. Phys.* 72 (3) (2004) 367–375, <https://doi.org/10.1119/1.1632486>.
- [38] F. Gebert, Y. Wan, F. Wolf, C.N. Angstmann, J.C. Berengut, P.O. Schmidt, Precision isotope shift measurements in calcium ions using quantum logic detection schemes, *Phys. Rev. Lett.* 115 (5) (2015) 053003, <https://doi.org/10.1103/PhysRevLett.115.053003>.
- [39] W. Gins, R.P. de Groote, M.L. Bissell, C. Granados Buitrago, R. Ferrer, K.M. Lynch, G. Neyens, S. Sels, Analysis of counting data: Development of the SATLAS Python package, *Comput. Phys. Commun.* 222 (2018) 286–294, <https://doi.org/10.1016/j.cpc.2017.09.012>.
- [40] W. Gins, B. van den Borne, R.P. de Groote, G. Neyens, SATLAS2: an update to the package for analysis of counting data, *Comput. Phys. Commun.* 297 (2024) 109053, <https://doi.org/10.1016/j.cpc.2023.109053>.
- [41] J.R. Johansson, P.D. Nation, F. Nori, QuTiP: an open-source Python framework for the dynamics of open quantum systems, *Comput. Phys. Commun.* 183 (8) (2012) 1760–1772, <https://doi.org/10.1016/j.cpc.2012.02.021>.
- [42] J.R. Johansson, P.D. Nation, F. Nori, QuTiP 2: a Python framework for the dynamics of open quantum systems, *Comput. Phys. Commun.* 184 (4) (2013) 1234–1240, <https://doi.org/10.1016/j.cpc.2012.11.019>.
- [43] S. Eckel, D.S. Barker, E.B. Norrgard, J. Scherschligt, PyLCP: a Python package for computing laser cooling physics, *Comput. Phys. Commun.* 270 (2020) 108166, <https://doi.org/10.1016/j.cpc.2021.108166>.
- [44] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer, New York, 2006.
- [45] Matt Newville, Renee Otten, Andrew Nelson, Till Stensitzki, Antonino Ingargiola, Dan Allan, Austin Fox, Faustina Carter, Michał, Ray Osborn, Dima Pustakhod, Sebastian Weigand, Ineuhau, Andrey Aristov, Glenn Mark, mgunyo, Christoph Deil, Allan L.R. Hansen, Gustavo Pasquevich, Leon Foks, Nicholas Zobrist, Oliver Frost, Stuermer, Jean-Christophe Jaskula, Shane Caldwell, Pieter Eendebak, Matteo Pompili, Jens Hedegaard Nielsen, Arun Persaud, mfit/lmfit-py: 1.3.2, <https://doi.org/10.5281/zenodo.12785036>, 2024.

- [46] S. Kaufmann, Laser spectroscopy of nickel isotopes with a new data acquisition system at ISOLDE, Doctoral thesis (PhD thesis), Technische Universität Darmstadt, 2019, <https://tuprints.ulb.tu-darmstadt.de/id/eprint/9286>.
- [47] S. Malbrunot-Ettenauer, S. Kaufmann, S. Bacca, C. Barbieri, J. Billowes, M.L. Bissell, K. Blaum, B. Cheal, T. Duguet, R.F.G. Ruiz, W. Gins, C. Gorges, G. Hagen, H. Heylen, J.D. Holt, G.R. Jansen, A. Kanellakopoulos, M. Kortelainen, T. Miyagi, P. Navrátil, W. Nazarewicz, R. Neugart, G. Neyens, W. Nörtershäuser, S.J. Novario, T. Papenbrock, T. Ratajczyk, P.-G. Reinhard, L.V. Rodríguez, R. Sánchez, S. Sailer, A. Schwenk, J. Simonis, V. Somà, S.R. Stroberg, L. Wehner, C. Wraith, L. Xie, Z.Y. Xu, X.F. Yang, D.T. Yordanov, Nuclear charge radii of the nickel isotopes $^{58-68,70}\text{Ni}$, *Phys. Rev. Lett.* 128 (2) (2022) 022502, <https://doi.org/10.1103/PhysRevLett.128.022502>.
- [48] P. Müller, S. Kaufmann, T. Miyagi, J. Billowes, M.L. Bissell, K. Blaum, B. Cheal, R.F. Garcia Ruiz, W. Gins, C. Gorges, H. Heylen, A. Kanellakopoulos, S. Malbrunot-Ettenauer, R. Neugart, G. Neyens, W. Nörtershäuser, T. Ratajczyk, L.V. Rodríguez, R. Sánchez, S. Sailer, A. Schwenk, L. Wehner, C. Wraith, L. Xie, Z.Y. Xu, X.F. Yang, D.T. Yordanov, Electromagnetic moments of the odd-mass nickel isotopes $^{59-67}\text{Ni}$, *Phys. Lett. B* 854 (2024) 138737, <https://doi.org/10.1016/j.physletb.2024.138737>.
- [49] E. Jaynes, Prior probabilities, *IEEE Trans. Syst. Sci. Cybern.* 4 (3) (1968) 227–241, <https://doi.org/10.1109/TSSC.1968.300117>.
- [50] D. Manzano, A short introduction to the Lindblad master equation, *AIP Adv.* 10 (2) (2020) 025106, <https://doi.org/10.1063/1.5115323>.
- [51] G. Lindblad, On the generators of quantum dynamical semigroups, *Commun. Math. Phys.* 48 (2) (1976) 119–130, <https://doi.org/10.1007/BF01608499>.
- [52] H.J. Carmichael, *Statistical Methods in Quantum Optics 1*, Springer Berlin Heidelberg, 1999.
- [53] R. Dum, P. Zoller, H. Ritsch, Monte Carlo simulation of the atomic master equation for spontaneous emission, *Phys. Rev. A* 45 (7) (1992) 4879–4887, <https://doi.org/10.1103/PhysRevA.45.4879>.
- [54] R.C. Hilborn, Einstein coefficients, cross sections, f values, dipole moments, and all that, *Am. J. Phys.* 50 (11) (1982) 982–986, <https://doi.org/10.1119/1.12937>.
- [55] R.F. Garcia Ruiz, M.L. Bissell, K. Blaum, A. Ekström, N. Frömmgen, G. Hagen, M. Hammen, K. Hebeler, J.D. Holt, G.R. Jansen, M. Kowalska, K. Kreim, W. Nazarewicz, R. Neugart, G. Neyens, W. Nörtershäuser, T. Papenbrock, J. Papuga, A. Schwenk, J. Simonis, K.A. Wendt, D.T. Yordanov, Unexpectedly large charge radii of neutron-rich calcium isotopes, *Nat. Phys.* 12 (6) (2016) 594–598, <https://doi.org/10.1038/nphys3645>.
- [56] M. Wang, W.J. Huang, F.G. Kondev, G. Audi, S. Naimi, The AME 2020 atomic mass evaluation (II). Tables, graphs and references*, *Chin. Phys. C* 45 (3) (2021) 030003, <https://doi.org/10.1088/1674-1137/abddaf>.
- [57] A. Kramida, Y. Ralchenko, J. Reader, NIST ASD Team, NIST atomic spectra database (ver. 5.10), <https://doi.org/10.18434/T4W30F>, 2022.