Journal of Physics B:
Atomic, Molecular and
Optical Physics

TUTORIAL • OPEN ACCESS

Simple Python tools for modelling few-level atom-light interactions

To cite this article: Lucy Downes 2023 *J. Phys. B: At. Mol. Opt. Phys.* **56** 223001

OPEN ACCESS
IOP Publishing

Journal of Physics B: Atomic, Molecular and Optical Physics

J. Phys. B: At. Mol. Opt. Phys. **56** (2023) 223001 (17pp)

https://doi.org/10.1088/1361-6455/acee3a

Tutorial

Simple Python tools for modelling few-level atom-light interactions

Lucy Downes

Department of Physics, Durham University, South Road, Durham DH1 3LE, United Kingdom

E-mail: lucy.downes@durham.ac.uk

Received 20 April 2023, revised 31 July 2023
Accepted for publication 8 August 2023
Published 25 October 2023

Abstract

Understanding the interactions between atoms and light is at the heart of atomic physics. Being able to 'experiment' with various system parameters, produce plots of the results and interpret these is very useful, especially for those new to the field. This tutorial aims to provide an introduction to the equations governing near-resonant atom-light interactions and present examples of setting up and solving these equations in Python. Emphasis is placed on clarity and understanding by showing code snippets alongside relevant equations, and as such it is suitable for those without an excellent working knowledge of Python or the underlying physics. Hopefully the methods presented here can form the foundations on which more complex models and simulations can be built. All functions presented here and example codes can be found on GitHub.

Keywords: atom-light interactions, optical Bloch equations, Python, thermal vapor

(Some figures may appear in colour only in the online journal)

1. Introduction

The interactions of hydrogen-like atoms with narrowband near-resonant fields form the foundations of much of modern atomic physics. Being able to describe and model these interactions is therefore of great importance from both a theoretical and experimental perspective. The simplest way to describe atom-light interactions is to consider a classical picture, such as Einstein's rate equations. However classical descriptions do not give rise to many phenomena of interest in atom-light systems such as electromagnetically induced transparency (EIT) [1]. In order to see how these effects emerge, a semi-classical approach must be used that takes into account aspects of the quantum mechanical nature of the system. The equations that this approach gives rise to cannot always be solved analytically, so numerical methods need to be applied. Even when analytic solutions can be found they often require making assumptions about the system so care needs to be taken to ensure these are valid. One way of exploring a system is by building a simple computational model and changing the system parameters to understand their effects. Having tools with which to build these simulations is an invaluable resource to both students and established scientists alike.

Python is a high-level programming language that has proved to be a useful tool in scientific computing due to its ease of use, vast scope and free availability. Specialized open-source atomic physics packages are commonplace, allowing users to perform complex calculations such as model the electric susceptibility of an atomic gas [2, 3] or integrate wavefunctions and calculate interaction potentials between highly-excited Rydberg atoms [4–6]. There are already a multitude of different Python packages designed to deal with solving problems involving interactions between atoms and light, for example QuTiP [7], mbsolve [8], PyLCP [9] and MaxwellBloch [10]. Many focus on time-dependent Hamiltonians and solutions and as such are highly complex. Often they require installation of a package which provides 'black box' solutions-functions are called 'blind' without knowledge of the underlying equations or methods. This means it is difficult to understand how the solutions are reached and what assumptions are being made, and it is even harder to troubleshoot when things go wrong.

Here we present a simple way of exploring the steady state behavior of few-level atom-light systems using basic Python functions, without the need to install multiple packages beyond the standard Python libraries. `NumPy` (numerical Python [11]), `SciPy` (scientific Python [12]) and `SymPy` (symbolic Python [13]) are all that is required, alongside `Matplotib` for producing plots [14]. We will outline how all of the Python functions are created from the equations, how they are manipulated and then how the steady-state solution can be found. We will also present examples of using the defined functions to create plots and interpret the underlying physics for both zero and finite temperature systems.

1.1. Paper structure

We will first review the semi-classical approach to near-resonant atom-light interactions using the density matrix to describe an atom-light system. We will derive the optical Bloch equations (OBEs) and look at how measurable quantities can be extracted to compare to experiments. In the next section we will look at two possible solution methods and the regimes in which they are valid. These methods are presented alongside relevant code snippets to give some insight into how the equations are translated into Python functions. Given lists of the system parameters, we will use `Sympy` to derive the OBEs from an interaction Hamiltonian and decay/dephasing operator, then expresses these as matrix equations and uses a SVD to solve them. We then look at some examples of using the code to generate plots of quantities of common interest in experiment, and use these to understand some of the phenomena observed in atom-light systems. We finish by demonstrating how the functions could be used to account for effects such as Doppler broadening in systems with a finite temperature.

1.2. Notes on code and examples

In this tutorial, selected code snippets are presented to illustrate how the equations outlined are coded and solved. All functions outlined here are available in a module on GitHub (https://github.com/LucyDownes/OBE_Python_Tools) alongside Jupyter notebooks showing how the functions are created and some examples of their use. All codes used to generate the plots in this work are also available there. For most plots, parameters are in units of 2π MHz. The factor of 2π is omitted from plot axis labels for clarity. In examples, parameters were chosen such that a particular effect is clearly visible in the plot. Hence they are purely illustrative and not intended to represent any particular physical experiment. All of the code presented here was written in Python 3.8 but could very easily be made backwards compatible with Python 2.7. The code is in no way optimized for speed, it is intended to be as transparent as possible to aid insight. Variable names have been chosen to align with the notation used in this paper. Parameters (and hence variables) referring to a field coupling two levels i and j with have the subscript ij, whereas parameters that are relevant to a single atomic level n will have the single subscript n. Variable names with upper/lower-case letters denote upper/lower-case Greek symbols respectively, for example `Gamma` denotes Γ while `gamma` denotes γ.

2. Theory

2.1. Semi-classical atom-light interactions

In order to model how a single atom interacts with electromagnetic (EM) fields, we take a semi-classical approach in which the radiation field is considered as classical but the atom is considered quantum mechanically [15, 16].

The time evolution of the atomic wavefunction $|\psi\rangle$ is described by the time-dependent Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = H|\psi\rangle. \tag{1}$$

The total Hamiltonian H of the system is made up of two parts

$$H = H_0 + H_I(t), \tag{2}$$

where H_0 is the bare atomic Hamiltonian in the absence of any radiation and H_I describes the interaction with a time-dependent EM field. To begin with we will ignore the effects of dephasing processes and focus only on the coherent dynamics of the system.

First let us consider a two level atom with non-degenerate levels $|1\rangle$ and $|2\rangle$ having energies of $E_1 = \hbar\omega_1$ and $E_2 = \hbar\omega_2$, as shown on the left hand side of figure 1. These are known as the bare states of the system, and form a complete basis set. Any state of the atom $|\psi\rangle$ can then be written as a superposition of these bare states, which for our two-level atom can be written

$$|\psi\rangle = c_1|1\rangle + c_2|2\rangle \tag{3}$$

where the coefficients c_i represent the probability amplitude of being in state $|i\rangle$.

In the basis of bare states we can write the atomic Hamiltonian as

$$H_0 = \begin{pmatrix} \hbar\omega_1 & 0 \\ 0 & \hbar\omega_2 \end{pmatrix}. \tag{4}$$

The eigenvalues of this Hamiltonian give the energies of the two levels. By defining the transition frequency

$\omega_0 = \omega_2 - \omega_1$, we can shift the zero point energy and rewrite the Hamiltonian as

$$H_0 = \begin{pmatrix} 0 & 0 \\ 0 & \hbar\omega_0 \end{pmatrix}. \tag{5}$$

The energy difference between the states remains the same ($\hbar\omega_0$), we have just defined the energy of the lowest state to be zero.

The levels are coupled by a laser field with frequency ω. Since we are considering near-resonant excitation, we have that $\omega \approx \omega_0$, but we can quantify the difference between the laser frequency and the resonant frequency of the transition as $\Delta = \omega - \omega_0$. This is known as the detuning.

The interaction Hamiltonian H_I takes the form $-\mathbf{d}\cdot\mathbf{E}$ where $\mathbf{E}$ is the electric field and $\mathbf{d} = -e\mathbf{D}$ is the electric dipole operator.

Treating the EM field classically means it can be modelled as a simple oscillating field, meaning it can be written as $\mathbf{E} = E_0\hat{\boldsymbol{e}}\cos(\omega t)$ where E_0 is the amplitude of the field, $\hat{\boldsymbol{e}}$ is a polarization vector and ω is the frequency of the field. Here we have neglected the spatial dependence of the field, an approximation known as the dipole approximation. This approximation is valid since the scale of the atomic wavefunction is much smaller than the wavelength of the applied fields, so the field can be considered uniform across the extent of the atom.

The field interacts with the atom via the dipole operator. This can be written as

$$\hat{\mathbf{D}} = \sum_{i,j} \mathbf{D}_{ij} |i\rangle\langle j| \tag{6}$$

where $|i\rangle$ and $|j\rangle$ are basis states of the bare atomic Hamiltonian. The term $\mathbf{D}_{ij} = \langle i|D|j\rangle$ is known as the matrix element [15]. Now we can write the interaction Hamiltonian as

$$H_I = eE_0\cos(\omega t)\sum_{i,j}\mathbf{D}_{ij}\cdot\hat{\mathbf{e}}|i\rangle\langle j|. \tag{7}$$

It is convenient at this point to define a new quantity

$$\Omega_{ij} = \frac{eE_0}{\hbar}\langle i|\mathbf{D}\cdot\hat{\mathbf{e}}|j\rangle \tag{8}$$

which is known as the Rabi frequency. This defines the strength of the coupling between the levels i and j due to the field, and has units of frequency. It depends on both the intensity E_0 of the field and the polarization through $\hat{\mathbf{e}}$. From the definition of the Rabi frequency we can see that $\Omega_{ii} = 0$, and $\Omega_{ji} = \Omega_{ij}^*$.

Our interaction Hamiltonian now becomes

$$H_I = \frac{\hbar\Omega}{2}\left(e^{i\omega t} + e^{-i\omega t}\right)\left[|1\rangle\langle 2| + |2\rangle\langle 1|\right], \tag{9}$$

where we have dropped the subscript on the Rabi frequency since there is only one driving field. The terms in the round brackets describe the field, and can be thought of as describing the emission and absorption of a photon respectively. The operators in the square brackets describe the atom, and are related to a transition $|2\rangle \to |1\rangle$ and $|1\rangle \to |2\rangle$ (lowering and raising operators). The process whereby the atom makes the transition $|1\rangle \to |2\rangle$ after absorbing a photon, or makes the transition $|2\rangle \to |1\rangle$ by emitting a photon are more important than the other processes, so we will keep only the two terms that describe these [15]. This is an approximation known as the 'rotating wave approximation', and means the interaction Hamiltonian now has the form

$$H_I = \frac{\hbar\Omega}{2}\left(e^{i\omega t}|1\rangle\langle 2| + e^{-i\omega t}|2\rangle\langle 1|\right). \tag{10}$$

Combining this with the atomic Hamiltonian H_0, we get the total Hamiltonian

$$H = \hbar\omega_0|2\rangle\langle 2| + \frac{\hbar\Omega}{2}\left(e^{i\omega t}|1\rangle\langle 2| + e^{-i\omega t}|2\rangle\langle 1|\right), \tag{11}$$

which could also be expressed in matrix form as

$$H = \frac{\hbar}{2}\begin{pmatrix} 0 & \Omega e^{i\omega t} \\ \Omega e^{-i\omega t} & 2\omega_0 \end{pmatrix}. \tag{12}$$

In order to get rid of the explicit time dependence in the Hamiltonian, we can perform a frame transformation into a rotating frame. In this simple two-level case this can be done[1] by introducing new variables $\tilde{c}_1(t)$ and $\tilde{c}_2(t)$ and writing the state vector as

$$|\psi\rangle = \tilde{c}_1|1\rangle + \tilde{c}_2 e^{-i\omega t}|2\rangle. \tag{13}$$

By inserting this expression for the state vector and the total Hamiltonian into the time-dependent Schrödinger equation, it is possible to find equations for the time dependence of the coefficients $\tilde{c}_1(t)$ and $\tilde{c}_2(t)$. This can also be written in the form of a time-dependent Schrödinger equation with the new time-independent Hamiltonian

$$\tilde{H} = \frac{\hbar}{2}\begin{pmatrix} 0 & \Omega \\ \Omega & -2\Delta \end{pmatrix}, \tag{14}$$

where $\Delta = \omega - \omega_0$ is the detuning we defined earlier. This is the most commonly used form of the Hamiltonian, and the form we will use from here onwards.

This combined atom-light Hamiltonian allows us to explore properties of the coupled atom-light system. The effective energy levels of the atom-light system $E_\pm$, sometimes called eigenenergies, are found as the eigenvalues of this Hamiltonian and are given by

$$E_\pm = -\frac{\hbar\Delta_{12}}{2} \pm \frac{\hbar}{2}\sqrt{{\Delta_{12}}^2 + \Omega^2}. \tag{15}$$

[1] More generally, this frame transformation is performed by the application of a unitary operator of the form $\hat{U} = e^{i\omega t|2\rangle\langle 2|}$. The transformed state vector is then $|\tilde{\psi}\rangle = \hat{U}|\psi\rangle$, and the Schrödinger equation becomes

$$i\hbar\frac{\partial}{\partial t}|\tilde{\psi}\rangle = \tilde{H}|\tilde{\psi}\rangle$$

where the transformed Hamiltonian $\tilde{H} = \hat{U}H\hat{U}^\dagger + i\hbar\dot{\hat{U}}\hat{U}^\dagger$.

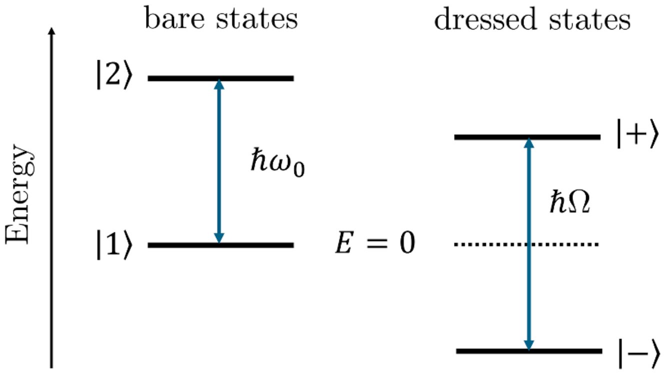

Figure 1. In the absence of any field coupling the atomic states, the bare states $|1\rangle$ and $|2\rangle$ are the eigenstates of the system, separated in energy by $\hbar\omega_0$. In the coupled atom-light system, the bare states are no longer eigenstates, instead we need to consider the dressed states $|\pm\rangle$, which have eigenenergies separated by $\hbar\Omega$ for a resonant field ($\Delta = 0$).

The eigenvectors of H are called 'eigenstates'. In the absence of the field (i.e. for $\Omega = 0$ and $\omega = 0$) the eigenstates are the bare states of the system ($|1\rangle$ and $|2\rangle$) with eigenenergies of 0 and $\hbar\omega_0$ respectively, but when considering the coupled atom-field system the bare states are no longer eigenstates. For a near resonant field (i.e. for $\Delta \approx 0$) the eigenstates are given by

$$|\pm\rangle = \frac{1}{\sqrt{2}}\left(|1\rangle \pm |2\rangle\right). \tag{16}$$

These new states are known as dressed states, and are linear superpositions of the bare states. From the eigenenergies in equation (15) it can be seen that for increasing Ω the levels $E_\pm$ will be shifted further from each other; an effect known as the AC Stark effect or light shift [17]. This is illustrated on the right hand side of figure 1.

2.2. Optical Bloch Equations

If we were to solve the Schrödinger equation for the coherent dynamics of the state vector $|\psi\rangle$ as defined earlier, we would find that the solutions are oscillatory. This oscillation of population between the two states are known as Rabi oscillations, and occur at a frequency defined by the Rabi frequency. This behavior can be seen in the upper panel of figure 2. However in real atomic systems there are incoherent processes such as decay between the levels which need to be taken into account which is not possible by considering the state vector alone. To this end we introduce the density operator defined as $\hat{\rho} = |\psi\rangle\langle\psi|$ and consider its evolution instead of that of the state vector.

In the basis of bare states, the density operator can be written in matrix form. Using the definition of the state vector from equation (3) we have

$$\hat{\rho} = \begin{pmatrix} |c_1|^2 & c_1 c_2^* \\ c_2 c_1^* & |c_2|^2 \end{pmatrix}. \tag{17}$$

The diagonal elements of the density matrix describe the populations of the bare states $|1\rangle$ and $|2\rangle$, while the off-diagonal elements describe the coherence between states. It is worth noting that using the density matrix allows us to include more information than the state vector alone. For example it is possible to describe mixed states, states that are a statistical mixture of quantum states instead of a linear superposition, which cannot be described by a single state vector. For a pure state (one that can be described by a linear superposition of bare states) the trace of the square of the density matrix (Tr[ρ^2]) is equal to 1, which is not the case for mixed states.

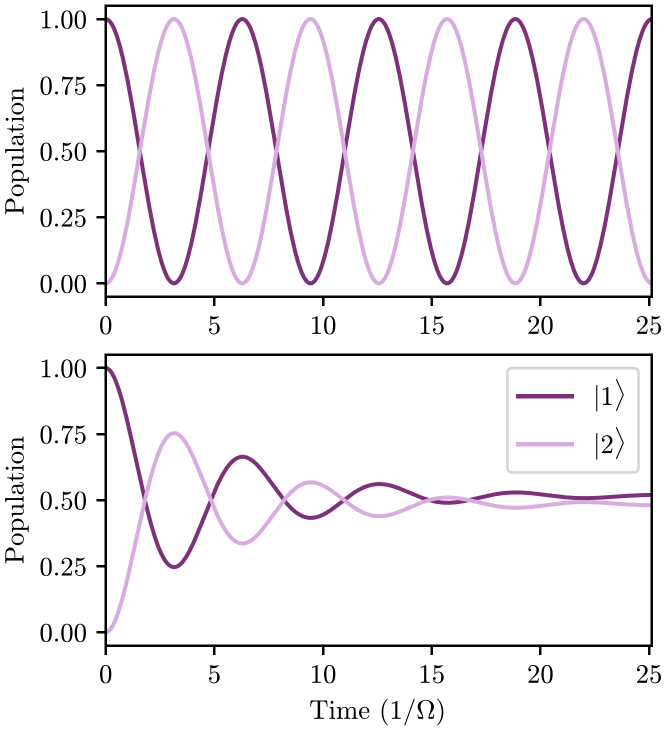

Figure 2. *Upper:* evolution of the populations of both the ground (dark purple) and excited (light purple) state in a driven two-level system without decay. The populations oscillate with characteristic frequency Ω, so-called Rabi oscillations. *Lower:* the same two-level system but including spontaneous decay from the excited state into the ground state at a rate $\Gamma_2 = \Omega/4$. The system quickly reaches the steady state at which point the population of both states is constant. The ratio of population in each state is related to the ratio between the decay rate and the Rabi frequency, as shown in figure 3.

The coherent time evolution of the density matrix is given by the Liouville von–Neumann equation [15]

$$i\hbar\frac{\partial\hat{\rho}}{\partial t} = \left[\hat{H}, \hat{\rho}\right], \tag{18}$$

in which $[\hat{H}, \hat{\rho}]$ represents the commutator of the total Hamiltonian $\hat{H}$ and the density operator $\hat{\rho}$. This equation is analogous to the time-dependent Schrödinger equation for the state vector. In order to include dissipative effects we use the Master equation

$$\frac{\partial\hat{\rho}}{\partial t} = -\frac{i}{\hbar}\left[\hat{H}, \hat{\rho}\right] + \hat{\mathcal{L}}(\hat{\rho}), \tag{19}$$

where again $\hat{H}$ is the Hamiltonian of the atom-light system and the Lindblad operator $\hat{\mathcal{L}}$ describes the decay/dephasing in the system.

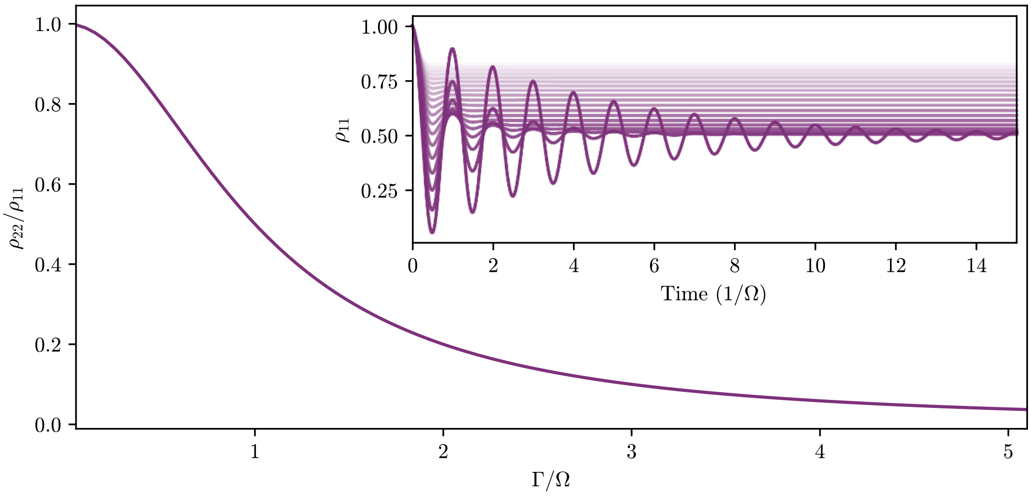

Figure 3. The ratio of excited to ground state population in the steady state for varying decay rates. For low values of Γ_2, the populations of the two states are equal in the steady state, whereas for increasing values of Γ_2, less and less population is transferred to the excited state. *Inset:* examples of the time evolution of the ground state population for values of Γ_2 between $\Omega/10$ and 2Ω (dark to light lines). Oscillations are more strongly damped for larger values of Γ_2.

The dissipative processes can be split into two separate parts; one part describing the spontaneous decay within the atom ($\hat{L}_{\text{decay}}$) and another describing dephasing due to the finite linewidths of the driving fields ($\hat{L}_{\text{dephasing}}$). The total operator will then be the sum of these parts

$$\hat{\mathcal{L}} = \hat{L}_{\text{decay}} + \hat{L}_{\text{dephasing}}. \tag{20}$$

First we focus on the operator $\hat{L}_{\text{decay}}$ describing spontaneous decay between levels. The decay rate of population out of each level is given by Γ_i. For our two-level atom we assume that since the state $|1\rangle$ is the ground state, there is no decay out of this level. This amounts to saying $\Gamma_1 = 0$. We also assume that the system is closed; population remains within the two-level system and is not lost to an external reservoir. This means that any population that decays from $|2\rangle$ must end up back in $|1\rangle$. The amount of population transfer will depend on the population of the excited state. Hence the diagonal elements of the decay operator describing the change in the population of the ground and excited state will be given by $\Gamma_2\rho_{22}$ and $-\Gamma_2\rho_{22}$ respectively. Since decay can only ever cause a loss of coherence, the off-diagonal elements of the decay operator are always negative, and decay at the average rate of decay for the two levels. This means that for our two-level system the decay operator can be written in matrix form as

$$\hat{L}_{\text{decay}} = \begin{pmatrix} \Gamma_2\rho_{22} & -\frac{\Gamma_2}{2}\rho_{12} \\ -\frac{\Gamma_2}{2}\rho_{21} & -\Gamma_2\rho_{22} \end{pmatrix}. \tag{21}$$

The action of decay is to damp the oscillations, meaning that eventually the system reaches an equilibrium, a so-called steady state. This is illustrated in the right-hand panel of figure 2. In continuous-wave experiments, this steady state is reached in very short timescales compared to the measurement time, so the steady state solution is often of most interest.

Secondly we establish the form of the dephasing operator $\hat{L}_{\text{dephasing}}$ which accounts for the fact that the driving field can never be truly single-frequency, but will instead have a finite width. The dephasing due to the finite laser linewidths only affects the coherences (off-diagonal density matrix elements). Since we have only two levels, we only need to include effects due to a single field. In this case the dephasing operator takes the form

$$\hat{L}_{\text{dephasing}} = \begin{pmatrix} 0 & -\gamma_{12}\rho_{12} \\ -\gamma_{21}\rho_{21} & 0 \end{pmatrix} \tag{22}$$

where $\gamma_{12} = \gamma_{21}$ is the linewidth of the field coupling the levels $|1\rangle$ and $|2\rangle$. Now we have expressions for the Hamiltonian and decay operator, we can substitute them into equation (19) to get an expression for the time evolution of the density matrix. In the steady-state, the solution is not time-dependent. This steady-state solution can be found by setting $\partial\hat{\rho}/\partial t = 0$ in equation (19) and solving for $\hat{\rho}$.

2.3. Measurable quantities

Once we have the values of the density matrix elements in the steady state, we would like to be able to extract useful measurable quantities to compare to experiment. The response of the medium to an applied field is characterized by the electric susceptibility χ. This macroscopic response of the system can be related to the individual atomic response as

$$\chi = \frac{-2ND_{ji}^2}{\epsilon_0\hbar\Omega_{ij}}\rho_{ji} \tag{23}$$

where N is the number density (atoms per unit volume in the ensemble), $D_{ji} = \langle j|D|i\rangle$ is the dipole matrix element defined earlier, Ω_{ij} is the Rabi frequency of the field coupling the ith and jth levels and ρ_{ji} is the element of the density matrix describing the coherence between the ith and jth levels.

The real part of the complex susceptibility is proportional to the dispersion of the medium, while the imaginary part is proportional to the absorption coefficient α. Often the experimental quantity of interest is the amount of absorption of a

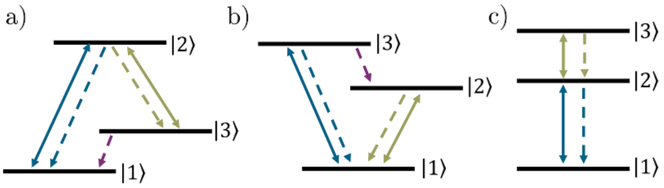

Figure 4. Different possible configurations of three atomic levels and two driving fields. (a) Lambda (Λ) (b) Vee (V) (c) Ladder (Ξ) configurations. The solid lines show laser fields coupling two states, the dashed lines show possible decay pathways.

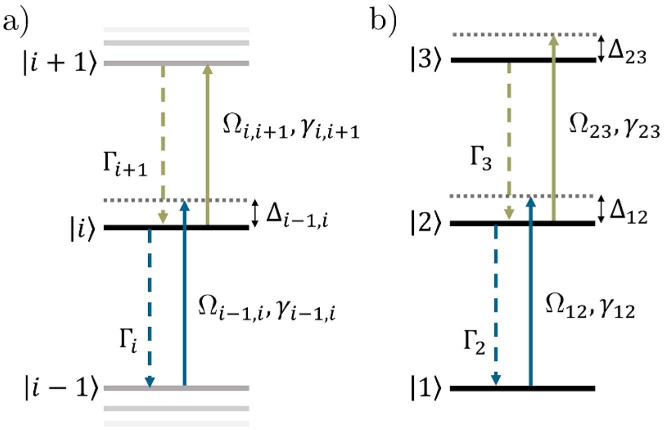

Figure 5. Level diagrams for ladder systems. (a) Any level $|i\rangle$ within the ladder will have a field $(\Omega_{i,i+1}, \gamma_{i,i+1}, \Delta_{i,i+1})$ coupling it to the level $|i+1\rangle$ above. There will also be a field $(\Omega_{i-1,i}, \gamma_{i-1,i}, \Delta_{i-1,i})$ coupling to the level $|i-1\rangle$ below. There will also be decay Γ_i to the $i-1^{\text{th}}$ level below. (b) A specific example of a three-level ladder scheme. There is no decay from the lowest energy level ($\Gamma_1 = 0\,\text{MHz}$).

laser after passing through the medium, most commonly the absorption of the probe laser which couples the lowest two levels. From the above relations we can see that the probe absorption $\alpha_{12} \propto -\Im[\rho_{21}]$, so often we will be interested in the steady state value of the density matrix element ρ_{21}.

2.4. Three-level systems

When considering the simple case of a two-level system we did not have to pay attention to the configuration of the states, however as we add more states and fields coupling these states we need to think about how they are connected. For many-level systems there are multiple possible arrangements of energy levels, driving fields and decay pathways within an atom. For 3-level systems, three of the most commonly considered configurations are the Lambda (Λ), Vee (V) and ladder (Ξ) systems, shown in figure 4 from left to right. Here we will deal only with ladder systems, in which each level is only coupled to the levels directly above and below it in energy. This level configuration is common in experiments which excite atoms to high-lying Rydberg states through multi-photon excitation [18–21]. Obviously this simplified description means that hyperfine structure is not accounted for, but this is often not resolved in higher-lying states and so can be neglected.

We can then extend our simple two-level model from the previous section to include an extra state $|3\rangle$ and driving field Ω_{23} that couples the intermediate and highest energy states. A diagram of this simple three-level ladder scheme and the relevant parameters can be seen in figure 5. For this three-level system we apply the same methods and approximations as before and can write the total Hamiltonian in the interaction picture as

$$H = \frac{\hbar}{2}\begin{pmatrix} 0 & \Omega_{12} & 0 \\ \Omega_{12} & -2\Delta_{12} & \Omega_{23} \\ 0 & \Omega_{23} & -2(\Delta_{12}+\Delta_{23}) \end{pmatrix}, \qquad (24)$$

where $\Omega_{12,23}$ are the Rabi frequencies of the laser fields and $\Delta_{12,23}$ are the corresponding detunings.

In the ladder configuration we assume that each level decays to the level directly below it at a rate Γ_n, and that there is no decay out of the lowest energy level. In this case the diagonal elements of the decay operator $\hat{L}_{\text{decay}}$ will be given by $L_{ii} = \Gamma_{i+1}\rho_{i+1,i+1} - \Gamma_i\rho_{ii}$ while the off-diagonal elements will be given by $L_{ij,i\neq j} = -\frac{\Gamma_i+\Gamma_j}{2}\rho_{ij}$ [15]. This means that for our example 3-level system we have

$$\hat{L}_{\text{decay}} = \begin{pmatrix} \Gamma_2\rho_{22} & -\frac{\Gamma_2}{2}\rho_{12} & -\frac{\Gamma_3}{2}\rho_{13} \\ -\frac{\Gamma_2}{2}\rho_{21} & \Gamma_3\rho_{33} - \Gamma_2\rho_{22} & -\frac{\Gamma_2+\Gamma_3}{2}\rho_{23} \\ -\frac{\Gamma_3}{2}\rho_{31} & -\frac{\Gamma_3+\Gamma_2}{2}\rho_{32} & -\Gamma_3\rho_{33} \end{pmatrix}, \qquad (25)$$

where we have already made the assumption that there is no decay out of the lowest level ($\Gamma_1 = 0$). The elements of the dephasing matrix $\hat{L}_{\text{dephasing}}$ can be expressed as

$$L_{ij,i\neq j} = -\left(\gamma_{i,i+1} + \cdots + \gamma_{j-1,j}\right)\rho_{ij}, \qquad (26)$$

where $\gamma_{n,n+1}$ is the linewidth of the field coupling the nth and $(n+1)$th levels [22].

2.5. Beyond 3 levels

In Rydberg experiments it is often necessary to use three or more fields to reach the desired Rydberg state [18–21]. If another field is introduced, for example to couple to adjacent Rydberg states when performing electrometry experiments [23–25], then this again adds to the complexity of the system. These multi-level systems can often be thought of as ladder systems. Note that the codes could all be adapted to the other system configurations shown in figure 4, but the form of the Hamiltonian and decay/dephasing operators would need to be redefined.

3. Solution methods

Even for relatively small numbers of atomic levels ($n > 2$) the system of equations for the elements of the density matrix can only be solved numerically unless some assumptions are made, the most common of which is the assumption that the probe field (the field coupling the two lowest energy states) is weak. In this case a perturbative approach can be taken and analytic solutions found (see section 3.2), but this assumption is not always valid. Hence it is useful to develop an approach that does not rely on making further assumptions about the system. In this section we will outline a numerical method that does not require the weak-probe assumption, and an analytic solution that is only valid in the weak-probe regime.

3.1. Matrix solution

Any linear system of equations can be expressed as a matrix of coefficients multiplied by a vector of variables. Our aim is to find a way to write the OBEs in this way, such that we can find the steady state solution using linear algebra.

3.1.1. Setting up the matrices. First we set up a function to create the density matrix $\hat{\rho}$ itself. Given a number of levels, this function returns a matrix populated with Sympy symbolic objects which will allow us to set up the equations and manipulate the terms. For an n-level system the density matrix will have n^2 components.

```
def create_rho_matrix(levels = 3):
    rho_matrix = numpy.empty((levels, levels),
    dtype = 'object')
    for i in range(levels):
        for j in range(levels):
            globals()[
                'rho_'+str(i+1)+str(j+1)] = \
            sympy.Symbol(
                'rho_'+str(i+1)+str(j+1))
            rho_matrix[i,j] = globals()[
                'rho_'+str(i+1)+str(j+1)]
    return numpy.matrix(rho_matrix)
```

Next we set up the Hamiltonian of the interaction between the atom and the light, in the form shown in equations (14) and (24).

```
def Hamiltonian(Omegas, Deltas):
    levels = len(Omegas)+1
    H = numpy.zeros((levels,levels))
    for i in range(levels):
        for j in range(levels):
            if numpy.logical_and(i==j, i!=0):
                H[i,j] = -2*(numpy.sum(
                    Deltas[:i]))
            elif numpy.abs(i-j) == 1:
                H[i,j] = Omegas[numpy.min([i,j])]
    return numpy.matrix(H/2)
```

This function returns a matrix of floats. Note that we have omitted the factor of $\hbar$ when we create this Hamiltonian. This is because it will cancel with the factor of $\hbar$ in the Master equation later on. We could include it explicitly, but because it is often many orders of magnitude smaller than the values of Ω and Δ it makes the overall values within the Hamiltonian very small and we risk errors due to floating point precision.

Next we set up the matrix form of the decay operator $\hat{\mathcal{L}}$. As we did earlier, we will split this into two separate parts for simplicity. First we focus on $\hat{L}_{\text{decay}}$ describing spontaneous decay. The function takes a list of values for the decay rates of the excited states Γ_i. For an n-level system, this list will have $n-1$ entries as we always assume no decay from the lowest energy level ($\Gamma_1 = 0$). Again the function returns an $n \times n$ matrix populated by multiples of Sympy symbolic objects.

```
def L_decay(Gammas):
    levels = len(Gammas)+1
    rhos = create_rho_matrix(levels = levels)
    Gammas_all = [0] + Gammas
    decay_matrix = numpy.zeros((levels, levels),\
    dtype = 'object')
    for i in range(levels):
        for j in range(levels):
            if i != j:
                decay_matrix[i,j] = -0.5*(
                    Gammas_all[i]+Gammas_all[j])\
                    *rhos[i,j]
            elif i != levels - 1:
                into = Gammas_all[i+1]*\
                rhos[1+i,j+1]
                outof = Gammas_all[i]*rhos[i, j]
                decay_matrix[i, j] = into - outof
            else:
                outof = Gammas_all[i]*rhos[i, j]
                decay_matrix[i,j] = - outof
    return numpy.matrix(decay_matrix)
```

We can create the matrix $\hat{L}_{\text{dephasing}}$ in a similar way, again from a list of values of γ_{ij}.

```
def L_dephasing(gammas):
    levels = len(gammas)+1
    rhos = create_rho_matrix(levels = levels)
    deph_matrix = numpy.zeros((levels, levels),\
    dtype = 'object')
    for i in range(levels):
        for j in range(levels):
            if i != j:
                deph_matrix[i,j] = -(numpy.sum(
                    gammas[numpy.min([i,j]):\
                    numpy.max([i,j])]))*rhos[i,j]
    return numpy.matrix(deph_matrix)
```

Now we have expressions for the Hamiltonian and Lindblad operators, we can put them into the Master equation to get an expression for the time evolution of the density matrix. As we saw in equation (19) we have that

$$\frac{\partial\hat{\rho}}{\partial t} = -\frac{i}{\hbar}(\hat{H}\hat{\rho} - \hat{\rho}\hat{H}) + \hat{L}_{\text{decay}} + \hat{L}_{\text{dephasing}}. \tag{27}$$

[Note that the $\hbar$ omitted earlier from the Hamiltonian would have cancelled with the $\hbar$ here, so when we define the function we omit the factor of $1/\hbar$.]

```
def Master_eqn(H_tot, L):
    levels = H_tot.shape[0]
    dens_mat = create_rho_matrix(levels = levels)
    return -1j*(
        H_tot*dens_mat - dens_mat*H_tot) + L
```

This gives us an expression for the time evolution of each of the components of the density matrix in terms of other components. To make this complex system of equations more convenient to solve we want to write them in the form

$$\frac{\partial \hat{\rho}_{\text{vect}}}{\partial t} = \hat{M}\hat{\rho}_{\text{vect}} \tag{28}$$

where $\hat{\rho}_{\text{vect}}$ is a column vector containing all of the elements of $\hat{\rho}$, and $\hat{M}$ is a matrix of coefficients. First we set up a function to create $\hat{\rho}_{\text{vect}}$, a list of the components of the density matrix $\rho_{i,j}$.

```
def create_rho_list(levels = 3):
    rho_list = []
    for i in range(levels):
        for j in range(levels):
            globals()[
                'rho_'+str(i+1)+str(j+1)] = \
            sympy.Symbol(\
                'rho_'+str(i+1)+str(j+1))
            rho_list.append(globals()[
                'rho_'+str(i+1)+str(j+1)])
    return rho_list
```

We can then create the matrix of coefficients $\hat{M}$ by using the Sympy functions `expand` and `coeff`. The `expand` function collects together terms containing the same symbolic object. The `coeff` function allows us to extract the multiplying coefficient for a specified symbolic object. This works since the only terms we have are linear with respect to the density matrix elements, e.g. we never have terms in ρ_{ij}^2 or higher.

```
def OBE_matrix(Master_matrix):
    levels = Master_matrix.shape[0]
    rho_vector = create_rho_list(levels = levels)
    coeff_matrix = numpy.zeros((levels**2, \
    levels**2), dtype = 'complex')
    count = 0
    for i in range(levels):
        for j in range(levels):
            entry = Master_matrix[i,j]
            expanded = Sympy.expand(entry)
            for n,r in enumerate(rho_vector):
                coeff_matrix[count, n] = complex(
                    expanded.coeff(r))
            count += 1
    return coeff_matrix
```

3.1.2. *Time-dependent solutions.*

We now have an expression for $\partial \hat{\rho}_{\text{vect}}/\partial t$ in terms of the matrix $\hat{M}$ multiplied by $\hat{\rho}_{\text{vect}}$. The value of $\hat{\rho}_{\text{vect}}$ at any time t can be found from

$$\hat{\rho}_{\text{vect}}(t) = \exp\left(\hat{M}t\right)\hat{\rho}_{\text{vect}}(0) \tag{29}$$

where $\hat{\rho}_{\text{vect}}(0)$ is the value of $\hat{\rho}_{\text{vect}}$ at time $t = 0$. Since we already have a way of creating the matrix $\hat{M}$, it is trivial to define functions to find the time-dependent behavior of $\hat{\rho}_{\text{vect}}$.

```
import scipy.linalg as linalg

def time_evolve(operator, t, p_0):
    exponent = operator*t
    #scipy.linalg.expm does matrix exponentiation
    term = linalg.expm(exponent)
    return numpy.matmul(term, p_0)
```

3.1.3. *SVD.*

Often in experiments the measurement timescales are long such that transient oscillatory behavior such as Rabi oscillations are not observed, and the system reaches a so-called steady-state. In the steady state, $\partial \hat{\rho}_{\text{vect}}/\partial t = 0$ so we need to solve the expression

$$\hat{M}\hat{\rho}_{\text{vect}} = 0. \tag{30}$$

This can be solved numerically in many different ways[2], but here we will solve the system by performing a singular value decomposition (SVD) on the matrix $\hat{M}$ [26]. The full details of the theory of the SVD are beyond this discussion, it is suffice to say that the SVD transforms $\hat{M}$ into three matrices such that

$$\hat{M} = U\Sigma V^{\dagger} \tag{31}$$

where $V^{\dagger}$ is the conjugate transpose of V. The matrix Σ is diagonal, the elements corresponding to the singular values of $\hat{M}$. The solution to the system of equations is then the column of V corresponding to the zero singular value. Since we have the same number of equations as we have unknowns we expect only one non-trivial solution. If none of the singular values are zero ($\hat{M}$ is non-singular) then there is no non-trivial solution. Floating point precision means that it is unlikely that any of the values will be identically zero, so we set a limit and assume that values smaller than this are due to floating point errors.

We perform the SVD using the built-in `numpy.linalg.svd` routine which returns the matrices U and $V^{\dagger}$ along with an array of singular values Σ. Before returning the values of the elements of the density matrix we normalize the sum of the populations to be 1. The function returns an array of values for $\hat{\rho}_{\text{vect}}$, the order of which is the same

[2] There are built-in routines in numpy/scipy that solve systems of linear equations, such as `numpy.linalg.lstsq`. These are based on the SVD but it is tricky to coax out the non-trivial solution. They are also only fractionally faster, as shown by the open blue points in figure 6.

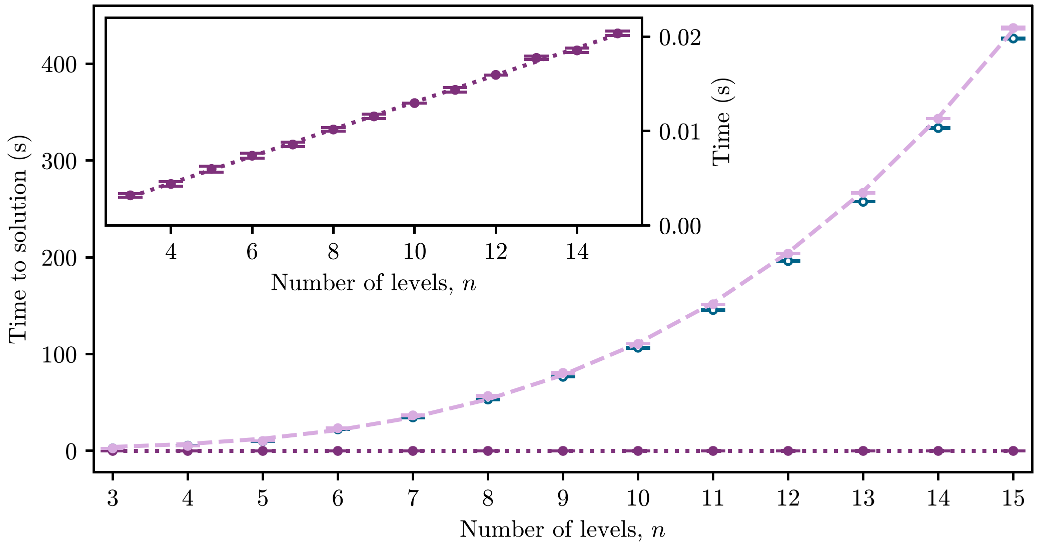

Figure 6. The datapoints show the time taken to find the weak probe coherence for an n-level system using the full matrix (light purple) and iterative (dark purple) methods. The errorbars represent the statistical error on five repeat calculations using the same hardware. The lines show linear (dark dotted) and power law (light dashed) fits to the iterative and matrix data respectively. Blue open circles show the time taken to use `scipy.linalg.lstsq` in place of `numpy.linalg.svd` for the matrix method. The inset highlights the linear scaling of the iterative method.

as the elements of the output of the `create_rho_list` function.

```
def SVD(coeff_matrix):
    levels = int(numpy.sqrt(
        coeff_matrix.shape[0]))
    u,sig,v = numpy.linalg.svd(coeff_matrix)
    abs_sig = numpy.abs(sig)
    minval = numpy.min(abs_sig)
    if minval>1e-12:
        # to account for floating point precision
        print('ERROR - Matrix is non-singular')
        return numpy.zeros((levels**2))
    index = abs_sig.argmin()
    rho = numpy.conjugate(v[index,:])
    # extract populations and normalise
    pops = numpy.zeros((levels))
    for l in range(levels):
        pops[l] = numpy.real(rho[(l*levels)+l])
    t = 1/(numpy.sum(pops))
    rho_norm = rho*t.item()
    #item() gets Python scalar from array
    return rho_norm
```

For ease of calculation we combine all of the previous functions into one function that takes lists of the different parameters (Ω, Δ, Γ and γ) and outputs the steady state solution for the density matrix in the order that `rho_list` is created. Note that it outputs the solution of the entire density matrix, so we need to extract the element of interest (usually ρ_{21}).

```
def steady_state_soln(
    Omegas, Deltas, Gammas, gammas = []):
    L_atom = L_decay(Gammas)
    # include dephasing if gammas values given
    if len(gammas) != 0:
        L_laser = L_dephasing(gammas)
        L_tot = L_atom + L_laser
    else:
        L_tot = L_atom
    # build Hamiltonian
    H = Hamiltonian(Omegas, Deltas)
    # insert H into Master equation
    Master = Master_eqn(H, L_tot)
    # create matrix of coefficients
    rho_coeffs = OBE_matrix(Master)
    # perform SVD to get steady state solution
    soln = SVD(rho_coeffs)
    return soln
```

All we need to do now is pass lists of the system parameters to this function.

3.2. Analytic solutions

Obviously because the matrix method requires solving an eigenvalue problem for each set of parameters it is not fast, however it does allow for the computation of results without any further assumptions about the system parameters. If the system is in the weak-probe regime, defined as the regime where $\Omega_{12}^2/\Gamma_3(\Gamma_2+\Gamma_3) \ll 1$, a number of assumptions can be made which allow analytic expressions for the probe coherence to be found. Since these assumptions involve treating the probe field as a weak perturbation, we keep only terms linear in Ω_{12}. In this case it can be said that no population transfer occurs, so we have that

$$\rho_{ii} = \begin{cases} 1, & i = 1 \\ 0, & i > 1. \end{cases} \tag{32}$$

For a three-level system, the probe coherence in the weak-probe regime can be expressed as

$$\rho_{21} = \frac{-i\Omega_{12}}{2}\left[-i\Delta_{12} + \frac{\Gamma_2}{2} + \gamma_{12} + \frac{\Omega_{23}^2}{4}\left[-i(\Delta_{12}+\Delta_{23}) + \frac{\Gamma_3}{2} + \gamma_{12} + \gamma_{23}\right]^{-1}\right]^{-1}. \tag{33}$$

which can easily be encapsulated into a function.

3.2.1. Analytic solution for an arbitrary number of levels. It is possible to formulate an expression for the analytic solution in the weak-probe regime for a ladder scheme with any number of levels. This has to be done iteratively.

For an system with n levels (hence $n-1$ driving fields) we have an expression for the probe coherence which needs to be evaluated for all fields $m = 1, 2 \ldots n-2, n-1$

$$\rho_{21} = \frac{2iK_1}{\Omega_{12}} \tag{34}$$

where

$$K_m = \left(\frac{\Omega_{m,m+1}}{2}\right)^2 [Z_m + K_{m+1}]^{-1} \tag{35}$$

and

$$Z_m = \sum_{k=1}^{m} (i\Delta_{k,k+1} - \gamma_{k,k+1}) - \frac{\Gamma_{m+1}}{2}. \tag{36}$$

For example, if we have a three-level scheme the expression will be

$$\rho_{21} = \frac{2i}{\Omega_{12}}\left(\frac{\Omega_{12}}{2}\right)^2\left[Z_1 + \left(\frac{\Omega_{23}}{2}\right)^2 [Z_2]^{-1}\right]^{-1} \tag{37}$$

where

$$Z_1 = i\Delta_{12} - \gamma_{12} - \Gamma_2/2$$

and

$$Z_2 = i(\Delta_{12} + \Delta_{23}) - (\gamma_{12} + \gamma_{23}) - \Gamma_3/2,$$

which is identical to equation (33).

This has been encapsulated into the function `fast_n_level` which takes the same sets of parameters as the `steady_state_soln` matrix method and returns the value of ρ_{21}.

```
def term_n(n, Deltas, Gammas, gammas = []):
    # Valid for n>0
    if len(gammas) == 0:
        gammas = numpy.zeros((len(Deltas)))
    return 1j*(numpy.sum(Deltas[:n+1])) - \
    (Gammas[n]/2) - numpy.sum(gammas[:n+1])

def fast_n_level(
    Omegas, Deltas, Gammas, gammas = []):
    n_terms = len(Omegas)
    term_counter = numpy.arange(n_terms)[::-1]
    func = 0
    for n in term_counter:
        prefactor = (Omegas[n]/2)**2
        term = term_n(n, Deltas, Gammas, gammas)
        func = prefactor/(term+func)
    return (2j/Omegas[0])*func
```

3.3. Computation time

The time taken for both the matrix and iterative solutions to find the weak-probe coherence in an n-level system is shown in figure 6. The matrix solution is around three orders of magnitude slower than the iterative solution, and has a power law dependence on the number of levels. The iterative solution scales linearly with the number of levels, as can be seen in the inset of figure 6. Obviously the exact times will vary with the hardware used, but it gives an illustration of the usefulness of the iterative method as a tool to fast computation for an arbitrary number of levels. Note that none of the code presented here has been optimized for speed. The aim is to get an insight into how the equations are constructed and solved in the simplest way possible, so there are likely much faster but possibly more complex ways to solve the equations presented here.

4. Examples

We will now show examples of using both the matrix method and analytic solutions to model the response of a system to applied fields. As noted earlier, often the parameter of interest is the probe coherence ρ_{21} as this is proportional to the absorption of the probe laser through the medium. Since $\rho_{21} = \rho_{12}^*$, we will usually extract ρ_{12} from the vector of the solution as this is always at the same index within the vector ρ_{vect} (`[1]`) whereas the position of ρ_{21} will vary depending on the number of levels. The probe absorption is then $\alpha_{12} \propto \mathrm{Im}[\rho_{12}]$. We will interpret the results by looking at the eigenenergies of the system, found from the total Hamiltonian.

4.1. Three-level system

Below is a simple example of using the matrix method (via the `steady_state_soln` function) to look at the probe absorption as a function of probe detuning (Δ_{12}) in a three-level system. We create an array of values for Δ_{12}, then for each of these values we calculate the steady state solution, extract the value of ρ_{21} and store the imaginary part in an array.

```
Omegas = [0.1,4] # 2pi MHz
Deltas = [0,0] # 2pi MHz
Gammas = [1,0.1] # 2pi MHz

Delta_12 = numpy.linspace(-10,10 ,200) # 2pi MHz
# create empty array to store solution
probe_abs = numpy.zeros((len(Delta_12)))
for i, p in enumerate(Delta_12):
    Deltas[0] = p # update value of Delta_12
    # Pass parameters to the function to find
    # the steady state solution for the
    # density matrix
```

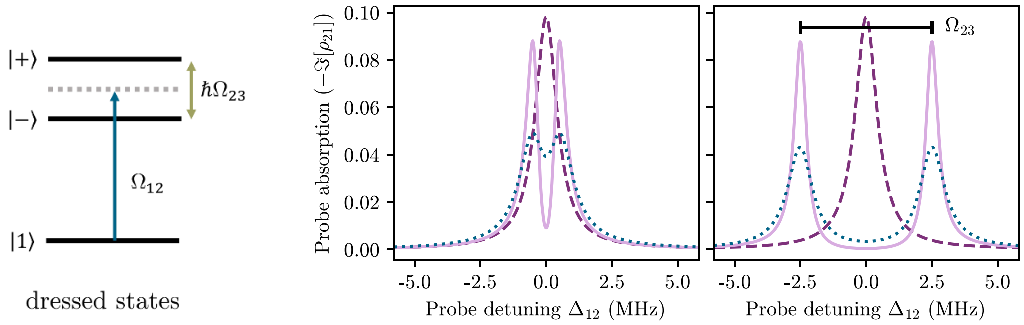

Figure 7. Examples of using the matrix method to find the probe laser absorption as a function of detuning for a three-level system. The response of the medium without the second field ($\Omega_{23} = 0$) is shown by the dark purple dashed line. *Left:* a weak second field ($\Omega_{23} = 1\,\text{MHz}$) opens a narrow transparency on resonance, as seen by the reduction in the probe absorption at $\Delta_{12} = 0$ on the light purple trace. *Right:* increasing the Rabi frequency of the second field widens the transparency until the peaks are completely separate. The spacing between the peak maxima is equal to the Rabi frequency of the second field, here $\Omega_{23} = 5\,\text{MHz}$. In the light purple solutions, dephasing due to laser linewidths has been ignored ($\gamma_{12,23} = 0\,\text{MHz}$). The dark blue dotted line shows the result of including a small amount of dephasing due to laser linewidth ($\gamma_{12,23} = 0.2\,\text{MHz}$). This results in the peaks becoming broader and reduced in size. Other parameters remain the same between panels and are set to $\Omega_{12} = 0.1\,\text{MHz}$, $\Delta_{23} = 0\,\text{MHz}$, $\Gamma_2 = 1\,\text{MHz}$, $\Gamma_3 = 0.1\,\text{MHz}$.

```
    solution = steady_state_soln(
        Omegas, Deltas, Gammas)
    # Store the imaginary part of rho_12 in
    # an array
    probe_abs[i] = numpy.imag(solution[1])
```

The results of running the code for different parameters are shown in figure 7. When there is only one field ($\Omega_{23} = 0\,\text{MHz}$), the system can be thought of as a two-level system. In this case the probe is maximally absorbed on resonance ($\Delta_{12} = 0\,\text{MHz}$) and we see a Lorentzian absorption feature as a function of probe detuning, the width of which is dependent on the ratio between Γ_2 and Ω_{12}.

Setting $\Omega_{12} = 0$ in equation (24) and solving for the eigenvectors yields the dressed states

$$|\pm\rangle = (\Delta_{23} \pm \sqrt{{\Delta_{23}}^2 + {\Omega_{23}}^2})|2\rangle + \Omega_{23}|3\rangle, \qquad (38)$$

with eigenenergies given by

$$E_{\pm} = -\frac{\hbar}{2}\left(2\Delta_{12} + \Delta_{23} \pm \sqrt{\Delta_{23}^2 + \Omega_{23}^2}\right). \qquad (39)$$

When both beams are on resonance ($\Delta_{12,23} = 0$) the expression for the dressed states becomes $|\pm\rangle = 1/\sqrt{2}(|3\rangle \pm |2\rangle)$ and we see that their eigenenergies are separated by $\hbar\Omega_{23}$. This means the probe beam that was resonant with the bare state $|2\rangle$ is no longer resonant with either of the dressed states of the coupled atom-field system, so a transparency occurs. When the probe becomes resonant with one of the dressed states we see absorption, hence in figure 7 we see two absorption peaks separated by Ω_{23}. This is known as EIT or Autler-Townes (AT) splitting, depending on the parameters used. The distinction between Autler–Townes splitting and EIT is not well defined but it is generally accepted that EIT requires the condition of weak fields (i.e. small $\Omega_{12,23}$) [27]. As Ω_{23} increases, Autler–Townes splitting becomes the dominant effect and the transparency window widens. For $\Delta_{23} = 0$ both dressed states contain an equal amount of the bare state $|2\rangle$ so the heights of the features in the absorption spectrum will be equal, as seen in figure 7. Any detuning of the second field ($\Delta_{23} \neq 0$) will result in an unequal mixture of the bare states in the dressed states as seen in equation (38) hence an asymmetric coupling to the ground state. This leads to features of unequal heights in the absorption spectra, as seen in figure 8. The negatively (positively) detuned peaks are due to coupling with the $|-\rangle$ ($|+\rangle$) dressed state. The peaks also become further apart as Δ_{23} is increased, since the eigenenergies are separated by $\Delta_{23} + \sqrt{\Delta_{23}^2 + \Omega_{23}^2}$.

Using the full matrix solution, it is possible to extract all of the parameters of the density matrix. For example if we wanted to look at the populations of the states we just need to extract the relevant terms from the output. For the three-level case, the populations of the states $|1\rangle$, $|2\rangle$ and $|3\rangle$ will be the first, fifth and ninth elements respectively. An example of looking at state population as a function of probe detuning and Rabi frequency is shown in figure 9. When the probe beam is weak ($\Omega_{12}^2 \ll \Gamma_3(\Gamma_2 + \Gamma_3)$) we see very little population transfer; one of the assumptions made when imposing the ‘weak-probe regime’. As Ω_{12} is increased (fainter lines in figure 9), more population is transferred between states.

In section 3.2 we said that if the weak-probe condition was satisfied ($\Omega_{12}^2 \ll \Gamma_3(\Gamma_2 + \Gamma_3)$) it was possible to find analytic solutions for the probe coherence, but that these solutions were not valid outside of the weak-probe regime. We compare the results of the full matrix solution and analytic solution in figure 10. In the first case upper panel of figure 10), the weak-probe approximation is valid (since $\Omega_{12}^2/\Gamma_3(\Gamma_2 + \Gamma_3) = 1/21 \ll 1$) and so the two methods give equal results. However the analytic solution is many orders of magnitude faster. If the weak-probe condition is not satisfied, the difference in results becomes apparent. This is seen in the lower panel of figure 10. The weak-probe condition is no longer

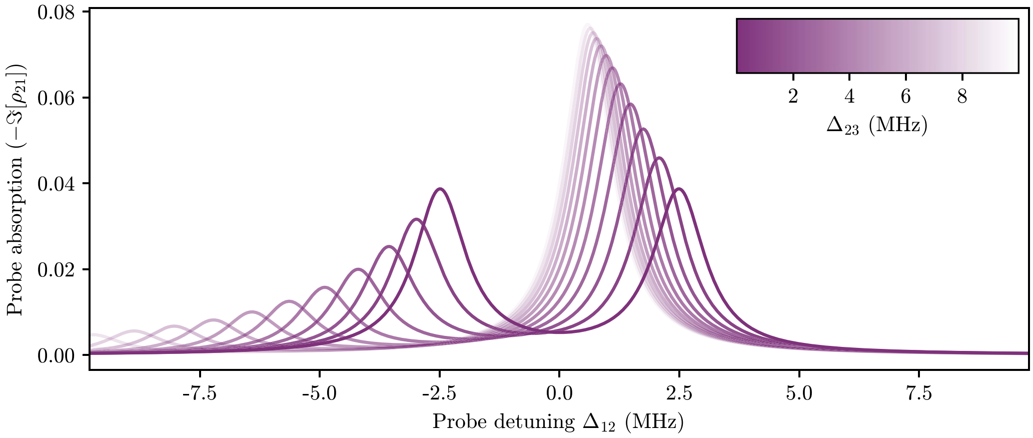

Figure 8. Effect of changing the detuning of the second field ($\Delta_{23} \neq 0$). When $\Delta_{23} = 0$ (darkest line), the two peaks are equal in height since the dressed states contain equal amounts of the bare state $|2\rangle$. They are also symmetrically spaced around $\Delta_{12} = 0$. For $\Delta_{23} > 0$ (lighter lines) the peaks become unequal in height since the dressed states contain different amounts of $|2\rangle$. Parameters used are $\Omega_{12} = 0.1\,\text{MHz}$, $\Omega_{23} = 5\,\text{MHz}$, $\Gamma_{2,3} = 1\,\text{MHz}$, $\gamma_{12,23} = 0.1\,\text{MHz}$.

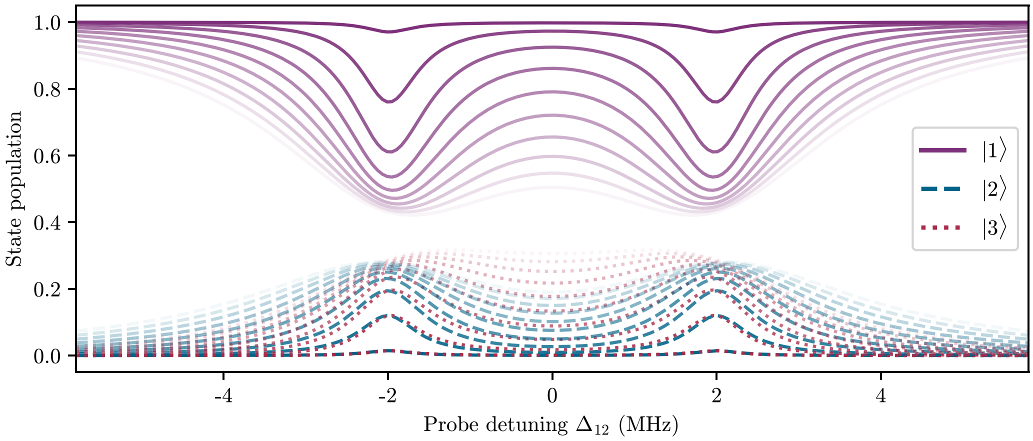

Figure 9. Populations of levels in a three-level system as a function of probe detuning for different values of probe Rabi frequency. The darkest lines are for $\Omega_{12} = 0.1\,\text{MHz}$, increasing to $\Omega_{12} = 2\,\text{MHz}$ with increasing transparency. When the probe beam is weak we see very little population transfer; one of the assumptions made when imposing the 'weak-probe regime'. As Ω_{12} is increased, more population is transferred between states. The other parameters used here are $\Omega_{23} = 4\,\text{MHz}$, $\Delta_{23} = 0\,\text{MHz}$, $\Gamma_2 = 0.5\,\text{MHz}$, $\Gamma_3 = 0.2\,\text{MHz}$ $\gamma_{12,23} = 0.1\,\text{MHz}$.

satisfied (since $\Omega_{12}^2/\Gamma_3(\Gamma_2+\Gamma_3) = 2.3 > 1$) and so the results of the analytic solution no longer match the full matrix solution.

4.2. More levels

Now we can look at a four-level scheme with three laser fields ($\Omega_{12}, \Omega_{23}, \Omega_{34}$). For the case where $\Omega_{34} = 0\,\text{MHz}$ we recover the spectrum for the three-level case as there is no coupling to the 4th level, as shown by the light purple solid line in figure 11.

As we increase the value of Ω_{34} a central feature appears and grows while the other features reduce in size, as shown by the dashed and dotted lines.

Similarly, for four levels we can compare the results of the full matrix solution to the analytic one, as shown in figure 12. If we go far outside the weak probe case, then the need for the full matrix solution becomes clear.

5. Thermal vapors

Up until this point we have not considered the effects of temperature on the atomic ensemble. These effects arise due to the motion of the atoms and can include collisional effects and modifications due to interactions with a laser beam of finite size [1]. In this tutorial we will only consider the first-order Doppler effects due to the velocity of the atoms within the ensemble, and we will neglect any collisional or time-of-flight effects. We will also assume that all excitation fields are colinear such that we only have to consider the effects of motion in one dimension. Fields propagating in the same direction are referred to as co-propagating, whereas fields propagating in opposite directions are counter-propagating. For an atom moving in the same direction as the propagation of a laser field, the frequency of the light will be red-shifted (lower frequency), whereas for an atom moving in the opposite direction the laser frequency would appear blue-shifted. To account for

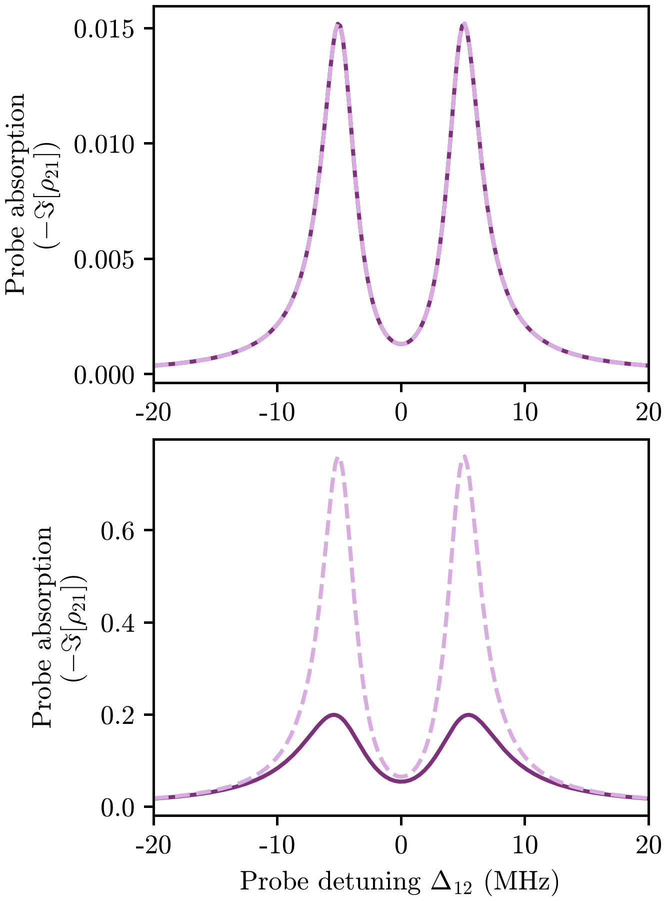

Figure 10. Comparison of results for the matrix (solid dark purple) and analytic weak-probe (dashed light purple) solutions for a three-level system. *Upper*: $\Omega_{12} = 0.1\,\text{MHz}$ so the weak-probe condition is satisfied and the methods give identical solutions. *Lower*: $\Omega_{12} = 5\,\text{MHz}$ so the weak-probe condition is not satisfied and the methods give very different solutions. Other parameters remain the same and are set at $\Omega_{23} = 10\,\text{MHz}$, $\Delta_{23} = 0\,\text{MHz}$, $\Gamma_2 = 5\,\text{MHz}$, $\Gamma_3 = 1\,\text{MHz}$, $\gamma_{12,23} = 0.1\,\text{MHz}$.

this we can modify the detuning terms in the Hamiltonian as $\Delta_{\text{eff}} = \Delta - \vec{k}\cdot\vec{v}$ where $\vec{k}$ is the wavevector of the laser light and $\vec{v}$ is the atomic velocity. Hence for an atom moving in the propagation direction we have $\Delta_{\text{eff}} = \Delta - |k|v$ and for an atom moving opposite to the direction of propagation we get $\Delta_{\text{eff}} = \Delta + |k|v$. This effect is illustrated in figure 13.

5.1. Velocity distribution

After calculating the response of atoms moving with different velocities, their relative abundances need to be accounted for. The atoms will have a distribution of velocities as given by the Maxwell-Boltzmann distribution

$$f_v(v_i) = \sqrt{\frac{m}{2\pi k_B T}} \exp\left(-\frac{m v_i^2}{2 k_B T}\right), \tag{40}$$

where m is the mass of an atom, k_B is the Boltzmann constant and T is the temperature of the vapour in Kelvin. We can either define this function explicitly, or recognize that this is a Gaussian distribution centered on 0 with width $\sqrt{\frac{k_B T}{m}}$ and use a built-in function such as `scipy.stats.norm`.

To compute the Doppler averaged response of the vapour we need to take the response of each velocity class and multiply its contribution to the probe absorption by the probability of an atom having that velocity (given by the Maxwell–Boltzmann distribution we defined earlier). Then, for each value of Δ_{12} we integrate (sum) over all the weighted velocity class contributions.

5.1.1. Choice of velocity range.

Obviously more velocity classes (a wider range of velocities or finer steps) means more computing time, but one needs to be careful not to lose information. Ideally the simulation would include a wide enough velocity range such that 99% of the atoms were accounted for. However for a vapour at room temperature, this would mean including velocities up to $\pm 500\,\text{m}\,\text{s}^{-1}$. To get high resolution, this would require a lot of discrete velocity classes. It is possible to truncate without loss of information, but this must be done with intuition about the system. Alternatively, one can choose non-linear velocity steps to increase resolution around points of interest.

5.2. Example: three-level system

In figure 14 we show an example of calculating the absorption of a weak probe beam in a thermal vapour of three-level atoms. The relative directions of the beams are taken into account with the `direction` variable which can take values of ±1 (same sign = co-propagating beams, opposite sign = counter-propagating beams). The actual value of the sign does not matter (as the velocity distribution will be symmetric about 0), as long as we are consistent with the signs of the wavevectors relative to each other. We also need to specify the wavelengths of the two beams, here chosen to be $\lambda_{12} = 780\,\text{nm}$ and $\lambda_{23} = 480\,\text{nm}$ to match commonly used rubidium two-photon EIT experiments. The beams have been set to be counter-propagating. These calculations can be quite computationally intensive, especially when using the full matrix method. Since the purpose here is to illustrate the effects of atomic motion, we will work in the weak probe regime ($\Omega_{12}^2 \ll \Gamma_3(\Gamma_2 + \Gamma_3)$) and use the iterative solution method (via the `fast_n_level` function) to save time.

```
Omegas = [0.1,10] # 2pi MHz
Deltas = [0,0] # 2pi MHz
gammas = [0.1,0.1] # 2pi MHz
Gammas = [10,1] # 2pi MHz
lamdas = numpy.asarray([780, 480])*1e-9 # m
# unitless direction vectors
directions = numpy.asarray([1,-1])
ks = directions/lamdas # 2pi m -1
```

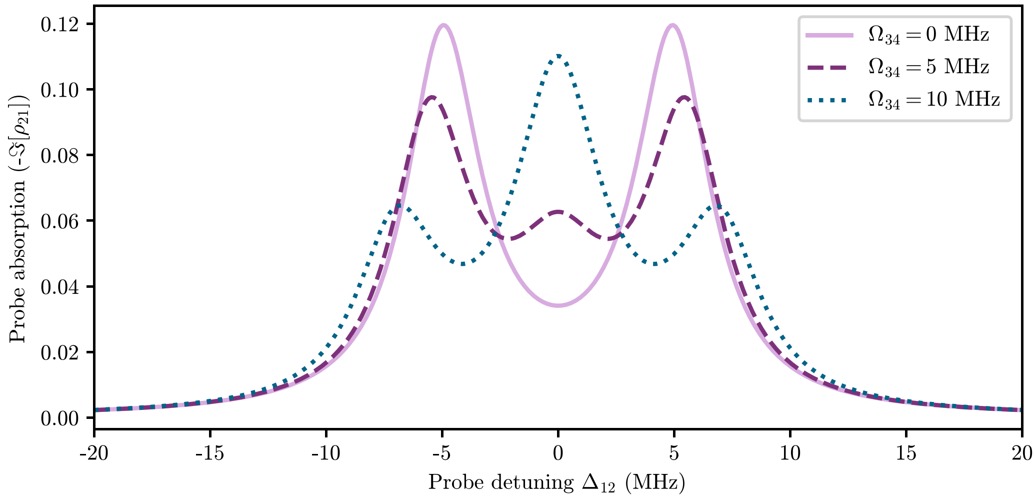

Figure 11. Probe absorption as a function of probe detuning for varying values of Ω_{34} in a four-level ladder system. When $\Omega_{34} = 0\,\text{MHz}$ we recover the two Autler–Townes peaks from the three-level case (solid light purple line). As Ω_{34} is increased, a central feature begins to appear and the outer peaks reduce in size. Simulation parameters used are: $\Omega_{12} = 1\,\text{MHz}$, $\Omega_{23} = 10\,\text{MHz}$, $\Delta_{23,34} = 0\,\text{MHz}$, $\Gamma_{2,3,4} = 2\,\text{MHz}$, $\gamma_{12,23,34} = 0.5\,\text{MHz}$.

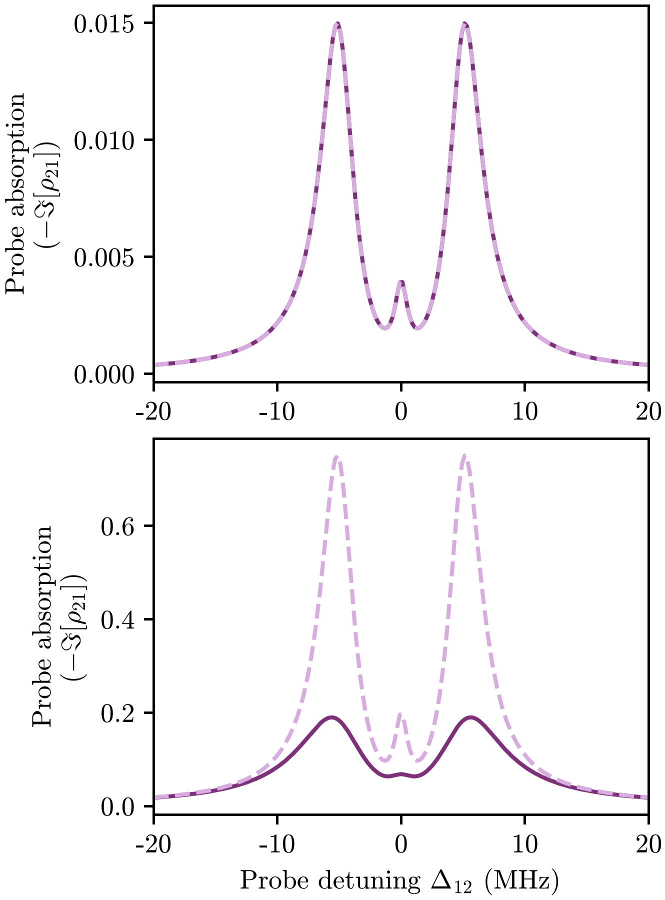

Figure 12. Comparison of results for the matrix (solid dark purple) and analytic weak probe (dashed light purple) solutions for a four-level system. *Upper*: $\Omega_{12} = 0.1\,\text{MHz}$ so the weak probe condition is satisfied and the methods give identical solutions. *Lower*: $\Omega_{12} = 5\,\text{MHz}$ so the weak probe condition is not satisfied and the methods give very different solutions. Other parameters remain the same and are set at $\Omega_{23} = 10\,\text{MHz}$, $\Omega_{34} = 2\,\text{MHz}$, $\Delta_{23,34} = 0\,\text{MHz}$, $\Gamma_2 = 5\,\text{MHz}$, $\Gamma_3 = 1\,\text{MHz}$, $\Gamma_4 = 0.5\,\text{MHz}$, $\gamma_{12,23,34} = 0.1\,\text{MHz}$.

```
Delta_12 = numpy.linspace(-50,50,500) # 2pi MHz
velocities = numpy.linspace(-150, 150, 501) # m/s
probe_abs = numpy.zeros((len(velocities), \
len(Delta_12)))
for i, v in enumerate(velocities):
    # loop over all atomic velocities
    for j, p in enumerate(Delta_12):
        # loop over all values of probe detuning
        # (lab frame)
        Deltas[0] = p
        # create empty array to store modified
        # detunings
        Deltas_eff = numpy.zeros(len(Deltas))
        for n in range(len(Deltas)):
            Deltas_eff[n] = Deltas[n]+(
                ks[n]*v)*1e-6 # in MHz
        # pass parameters to function
        #(including modified detuning)
        solution = fast_n_level(
            Omegas, Deltas_eff, Gammas, gammas)
        probe_abs[i,j] = -numpy.imag(solution)
```

The two dotted lines in the upper plot of figure 14 are the positions at which the dressed states $|+\rangle$ and $|-\rangle$ as defined in equation (39) are resonant with the probe. This will occur when

$$E_{\pm} = -\Delta_{12_{\text{eff}}} - \frac{\Delta_{23_{\text{eff}}}}{2} \pm \frac{1}{2}\sqrt{(\Delta^2_{23_{\text{eff}}} + \Omega^2_{23})} = 0, \qquad (41)$$

where the effective detunings are given by

$$\begin{aligned}\Delta_{12_{\text{eff}}} &= \Delta_{12} + v s_{12} k_{12}\\ \Delta_{23_{\text{eff}}} &= \Delta_{23} + v s_{23} k_{23}.\end{aligned}$$

$k_{12,23} = 1/\lambda_{12,23}$ are the magnitudes of the wavevectors (ignoring the factor of 2π) and $s_{12,23} = \pm 1$ represent the

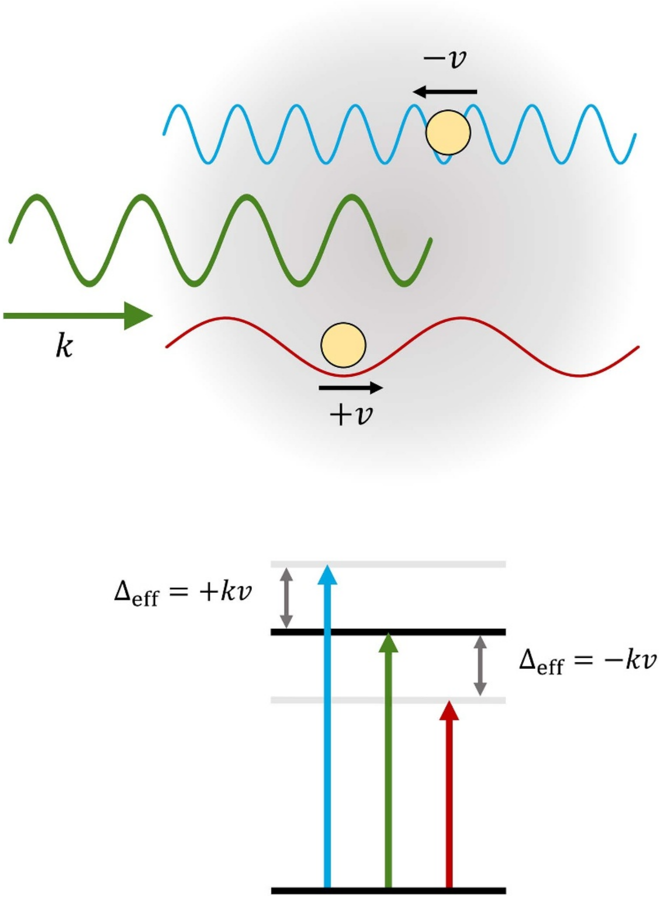

Figure 13. Illustration of how atomic motion affects the frequency 'seen' by the atoms. An atom moving at a velocity v opposite to the direction of propagation of a beam with wavevector $k = 2\pi/\lambda$ will see a higher (blue-shifted) frequency. Similarly an atom moving at a velocity v with the direction of propagation of the beam will see a lower (red-shifted) frequency. This can be included in the Hamiltonian as an effective detuning, as shown in the level diagram.

relative direction of the beams. $E_{\pm}$ are the eigenvalues of the Hamiltonian with $\Omega_{12} = 0$ and detunings replaced by the effective detunings. In the limit of large detuning where $\Delta_{23_{\text{eff}}} \gg \Omega_{23}$, the equations for the lines are

$$v_{+} = \frac{-\Delta_{12}}{s_{12}k_{12}} \tag{42}$$

$$v_{-} = \frac{-(\Delta_{12} + \Delta_{23})}{s_{12}k_{12} + s_{23}k_{23}}. \tag{43}$$

If fewer velocity classes were included, the integrated result would look very different and becomes meaningless. We can however use the knowledge of the equations of the asymptotic lines (found via the eigenvalues) to give an idea of the range of velocities that need to be sampled.

5.3. Example: four-level system

We can use the same method to look at the effects of atomic motion on four-level atoms. As with the previous case, we consider systems that are within the weak probe limit so we can use the faster iterative solution. Since we are only considering the 1D case we can again specify beam direction using the `direction` variable.

In the upper plot of figure 15 we can see three asymptotic 'branches', whose positions can again be found by finding the eigenvalues of the Hamiltonian in the case where $\Omega \ll \Delta_{\text{eff}}$, and equating them to zero. Rewriting the expressions in terms of v we have

$$v_1 = \frac{-\Delta_{12}}{s_{12}k_{12}} \tag{44}$$

$$v_2 = \frac{-\Delta_{12} - \Delta_{23}}{s_{12}k_{12} + s_{23}k_{23}} \tag{45}$$

$$v_3 = \frac{-\Delta_{12} - \Delta_{23} - \Delta_{34}}{s_{12}k_{12} + s_{23}k_{23} + s_{34}k_{34}}, \tag{46}$$

where again $k_{12,23,34} = 1/\lambda_{12,23,34}$ are the magnitudes of the wavevectors and $s_{12,23,34} = \pm 1$ represent the relative directions of the beams. Knowing the equations of these asymptotic lines can be useful when deciding how many velocity classes need to be taken into account. In the above example, there is not much extra information gained by including velocity classes $>100\,\text{m}\,\text{s}^{-1}$ or $<-100\,\text{m}\,\text{s}^{-1}$.

6. Conclusion

In this tutorial we have outlined methods of solving the OBEs and demonstrated how they can be implemented simply in Python. We began by looking at how the OBEs are derived from the semi-classical treatment of atom-light interactions, the role of the density matrix and how to extract measurable quantities from the values. We presented two solution methods in two different regimes; an analytic solution when the system is in the weak-probe regime and the matrix method which requires fewer simplifying assumptions. We then demonstrated the use of these solution methods to examine example spectra for few-level systems in both cold and thermal ensembles. All functions presented here are available on GitHub https://github.com/LucyDownes/OBE_Python_Tools, along with example Jupyter notebooks. The codes used to create all plots in this paper are also available there.

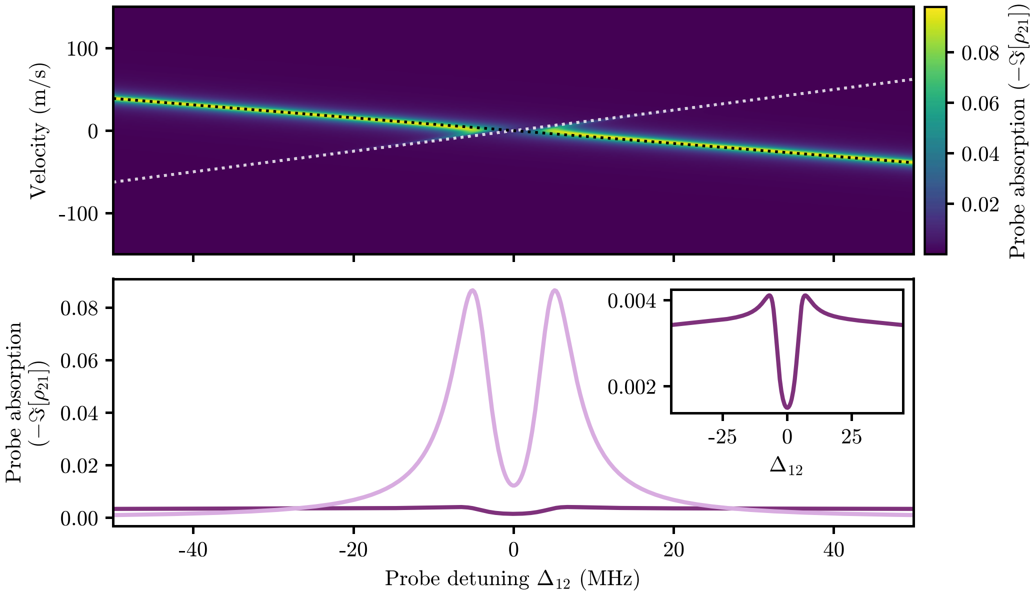

Figure 14. *Top:* probe absorption as a function of probe detuning and atomic velocity in a 3-level system with counterpropagating beams. The dashed lines represent the eigenvalues of the Hamiltonian in the case of no interaction. *Bottom:* the Doppler-averaged probe absorption (dark purple) found by vertically integrating the response multiplied by the Maxwell–Boltzmann distribution. The light purple shows the response for the zero velocity class for comparison. The shape of the Doppler-averaged response (see inset) is very different to the response for stationary atoms. The parameters are $\Omega_{12} = 0.1\,$MHz, $\Omega_{23} = 10\,$MHz, $\Delta_{23} = 0\,$MHz, $\Gamma_2 = 10\,$MHz, $\Gamma_3 = 1\,$MHz, $\gamma_{12,23} = 0.1\,$MHz, $\lambda_{12} = 780\,$nm, $\lambda_{23} = 480\,$nm, $s_{12} = +1$, $s_{23} = -1$.

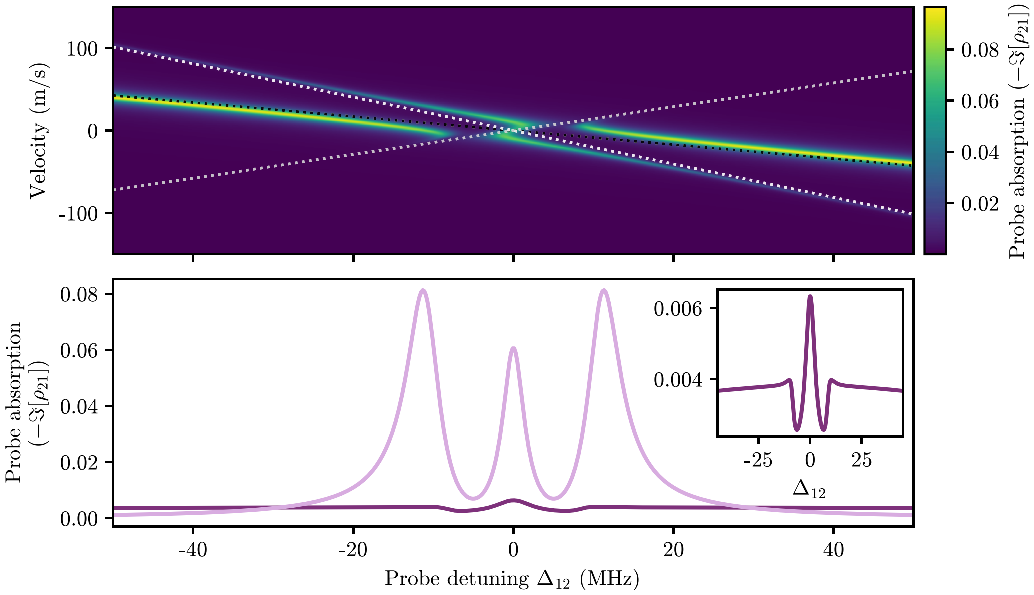

Figure 15. *Top:* probe absorption as a function of probe detuning and atom velocity in a four-level system where the probe beam counter-propagates with the other two. The dashed lines represent the eigenvalues of the Hamiltonian in the case of no interaction. *Bottom:* the Doppler-averaged probe absorption (dark purple) found by vertically integrating the response multiplied by the Maxwell–Boltzmann distribution. The light purple shows the response for the zero velocity class for comparison. The shape of the Doppler-averaged response is very different to the response for stationary atoms (see inset).The parameters are $\Omega_{12} = 1\,$MHz, $\Omega_{23} = 20\,$MHz, $\Omega_{34} = 10\,$MHz, $\Delta_{23,34} = 0\,$MHz, $\Gamma_2 = 10\,$MHz, $\Gamma_{3,4} = 1\,$MHz, $\gamma_{12,23,34} = 0.1\,$MHz, $\lambda_{12} = 852\,$nm, $\lambda_{23} = 1470\,$nm, $\lambda_{34} = 843\,$nm, $s_{12} = +1$, $s_{23,34} = -1$.

Data availability statement

No new data were created or analysed in this study.

Acknowledgments

The author would like to thank Ifan Hughes for constructive comments on an early draft of the manuscript, and Ollie Farley for code checking and GitHub help.

ORCID iD

Lucy Downes https://orcid.org/0000-0002-5999-7437

References

[1] Finkelstein R, Bali S, Firstenberg O and Novikova I 2023 *New J. Phys.* **25** 035001
[2] Zentile M A, Keaveney J, Weller L, Whiting D J, Adams C S and Hughes I G 2015 *Comput. Phys. Commun.* **189** 162–74
[3] Keaveney J, Adams C S and Hughes I G 2018 *Comput. Phys. Commun.* **224** 311–24
[4] Weber S, Tresp C, Menke H, Urvoy A, Firstenberg O, Büchler H P and Hofferberth S 2017 *J. Phys. B: At. Mol. Opt. Phys.* **50** 133001
[5] Šibalić N, Pritchard J, Adams C and Weatherill K 2017 *Comput. Phys. Commun.* **220** 319–31
[6] Robertson E, Šibalić N, Potvliege R and Jones M 2021 *Comput. Phys. Commun.* **261** 107814
[7] Johansson J, Nation P and Nori F 2013 *Comput. Phys. Commun.* **184** 1234–40
[8] Riesch M and Jirauschek C 2021 *Comput. Phys. Commun.* **268** 108097
[9] Eckel S, Barker D S, Norrgard E B and Scherschligt J 2022 *Comput. Phys. Commun.* **270** 108166
[10] Ogden T 2020 (available at: https://github.com/tpogden/maxwellbloch/)
[11] Harris C R *et al* 2020 *Nature* **585** 357–62
[12] Virtanen P *et al* (SciPy 1.0 Contributors) 2020 *Nat. Methods* **17** 261–72
[13] Meurer A *et al* 2017 *PeerJ Comput. Sci.* **3** e103
[14] Hunter J D 2007 *Comput. Sci. Eng.* **9** 90–95
[15] Cohen-Tannoudji C, Grynberg G and Dupont-Roc J 1992 *Atom-Photon Interactions: Basic Processes and Applications* (Wiley)
[16] Loudon R 2000 *The Quantum Theory of Light* (OUP Oxford)
[17] Foot C 2005 *Atomic Physics* (*Oxford Master Series in Physics*) (Oxford University Press)
[18] Wade C G, Šibalić N, de Melo N R, Kondo J M, Weatherill K J and Adams C 2017 *Nat. Photon.* **72** 40–43
[19] Carr C, Tanasittikosol M, Sargsyan A, Sarkisyan D, Adams C S and Weatherill K J 2012 *Opt. Lett.* **37** 3858–60
[20] Downes L A, MacKellar A R, Whiting D J, Bourgenot C, Adams C S and Weatherill K J 2020 *Phys. Rev.* X **10** 011027
[21] Chen S, Reed D J, MacKellar A R, Downes L A, Almuhawish N F, Jamieson M J, Adams C S and Weatherill K J 2022 *Optica* **9** 485–91
[22] Gea-Banacloche J, Li Y-Q, Jin S-Z and Xiao M 1995 *Phys. Rev.* A **51** 576–84
[23] Sedlacek J A, Schwettmann A, Kübler H, Löw R, Pfau T and Shaffer J P 2012 *Nat. Phys.* **8** 819–24
[24] Holloway C L, Gordon J A, Schwarzkopf A, Anderson D A, Miller S A, Thaicharoen N and Raithel G 2014 *Appl. Phys. Lett.* **104** 244102
[25] Holloway C L, Gordon J A, Jefferts S, Schwarzkopf A, Anderson D A, Miller S A, Thaicharoen N and Raithel G 2014 *IEEE Trans. Antennas Propag.* **62** 6169–82
[26] Riley K F, Hobson M P and Bence S J 2002 *Mathematical Methods for Physics and Engineering: A Comprehensive Guide* 2nd edn (Cambridge University Press)
[27] Anisimov P M, Dowling J P and Sanders B C 2011 *Phys. Rev. Lett.* **107** 163604