A computational toolbox for quantum and atomic optics

To cite this article: Sze M Tan 1999 *J. Opt. B: Quantum Semiclass. Opt.* **1** 424

J. Opt. B: Quantum Semiclass. Opt. **1** (1999) 424–432. Printed in the UK PII: S1464-4266(99)00873-3

A computational toolbox for quantum and atomic optics

Sze M Tan[†]

Physics Department, The University of Auckland, Private Bag 92019, Auckland, New Zealand

Received 12 January 1999

Abstract. A collection of routines is described which largely automates the process of generating the quantum mechanical equations of motion for problems involving systems with relatively few degrees of freedom. Their use allows the user to adopt a high-level approach to writing simulation programs which concentrates on the physics of the problem, rather than on the details of the solution. Examples are taken from the fields of quantum and atomic optics, but the toolbox is also useful for problems involving quantum information and in teaching quantum mechanics. The toolbox has been implemented using the Matlab programming language, but the ideas may be applied to any other object-oriented language.

Keywords: Open quantum systems, quantum computing, numerical simulation, object-orientated programming

1. Introduction

In many areas of physics, the solution of problems involves the processes of formulating a mathematical model of the system, writing down the equations which are appropriate for the system, and solving the resulting equations. Except in the simplest situations, numerical methods are required to generate solutions. In classical mechanics, for example, a system is described in terms of a set of (generalized) coordinates which evolve in time in accordance with Hamilton's canonical equations. Once the Hamiltonian of the system is known, it is usually straightforward to use a numerical differential equation solver in order to obtain the dynamics of the system. In quantum mechanics, however, the mathematical objects used to represent the states of physical systems and observable quantities are more complicated, making the process of writing computer programs to solve for and visualize the dynamics more involved. For the purposes of teaching quantum mechanics, a variety of pre-written packages are available (e.g., in the book by Brandt and Dahmen [1], the CUPS project [2], etc), which solve specific problems and allow the effects of varying the parameters in these problems to be investigated. Although valuable for giving insight into the consequences of the equations of quantum mechanics, this approach does not help the user to write programs which solve new problems.

Part of the difficulty in writing computer programs for quantum mechanics problems arises from a mismatch between the primitive data objects which exist in most computer languages (numbers and arrays) and the objects used in quantum mechanics (which include state vectors, density matrices and operators which may be functions of space and/or time). The equations of motion for rather simple systems are often a large system of coupled differential equations which has to be formulated carefully and transcribed into a program. With the availability of fast computers and programming languages which allow new classes of objects to be defined and manipulated, it is possible for the computer to help in the formulation of the differential equations from a higher-level description of the system, such as its Hamiltonian or Liouvillian, thus allowing the user to concentrate on the system at this level. Recently, Shack and Brun [3] have written a library of routines in C++ for solving master equations using quantum trajectory methods in which much of the mechanics of formulating the equations of motion are carried out automatically via the introduction of state and operator classes. Here, we discuss how the 'objects' of quantum mechanics may be mapped onto sparse matrices in such a way that allows the equations of motion for even quite complicated systems to be formulated rather simply. For problems which are small enough that the density matrix can be manipulated directly, steady-states, operator correlations, spectra and the like may be computed. For larger problems where numerical integration is unavoidable, it is possible to perform this integration for the components of the density matrix directly or to use quantum trajectory methods. An implementation of the toolbox has been written using the widely used programming language Matlab [4], which provides the underlying support for matrix manipulation and for user-defined classes of objects. Since Matlab is an interpreted language with many powerful graphics and visualization tools, the results of calculations may be examined interactively within the programming environment. For efficiency, however, after the equations of motion are formulated within Matlab, lengthy numerical integrations may be performed using external C programs

[†] E-mail address: `s.tan@auckland.ac.nz`

1464-4266/99/040424+09$30.00

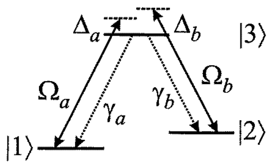


Figure 1. The level scheme for a three-level Λ atom driven by two classical coherent fields with detunings Δ_a and Δ_b. The Rabi frequencies are Ω_a and Ω_b and the spontaneous emission rates are γ_a and γ_b.

which may (in principle) be run on a remote machine, and the results read back into Matlab.

Due to limitations of space, only a few of the toolbox routines are described and many details of the implementation have been omitted. The reader is encouraged to contact the author for a preliminary copy of the toolbox and for more comprehensive documentation which are currently being developed. This paper is in the form of an introduction by way of a series of examples rather than a formal description of the data structures and internal algorithms.

2. The three-level atom in classical light fields

We first consider a typical problem with a small state space to illustrate how the process of deriving the equations of motion may be mechanized. Suppose that a three-level atom in a lambda configuration is illuminated by two classical light fields as shown in figure 1. In order to include the effects of spontaneous emission, a master equation for the density matrix is required. Since the state space $\mathcal{H}$ is three-dimensional, the density matrix has nine complex entries. The resulting differential equations for the density matrix are the optical Bloch equations (see, for example [5]). By Hermiticity, the equations of motion may be reduced to a system of nine real linear differential equations.

As is usual, a ket $|\psi\rangle$ is represented with respect to a basis $|e_k\rangle$ by a column vector $[|\psi\rangle]$ whose kth component is $\langle e_k|\psi\rangle$. A bra is represented by a row vector which is the conjugate transpose of the matrix of its associated ket. An operator $\hat{A}$ is represented by a matrix $[A]$ such that $[A]_{kl} = \langle e_k|\hat{A}|e_l\rangle$. For the three-level atom, we may define base states as $[|e_1\rangle] \equiv$ `e1` $=$ `[1; 0; 0]`, $[|e_2\rangle] \equiv$ `e2` $=$ `[0; 1; 0]`, and $[|e_3\rangle] \equiv$ `e3` $=$ `[0; 0; 1]`. Note that in Matlab, matrices are enclosed in square brackets: rows are separated by semicolons, while elements within a row are separated by commas or spaces. In an interaction picture, the Hamiltonian for the coherent part of the interaction between the atom and the light fields may be written as (taking $\hbar = 1$, for convenience):

$$\hat{H} = -\Delta_a|1\rangle\langle 1| - \Delta_b|2\rangle\langle 2| + \mathrm{i}\Omega_a(|1\rangle\langle 3| + |3\rangle\langle 1|) \\ +\mathrm{i}\Omega_b(|2\rangle\langle 3| + |3\rangle\langle 2|),$$

where Δ_a and Δ_b are the detunings and Ω_a and Ω_b are proportional to the amplitudes of the classical light fields. With the above definitions, the matrix $[H]$ corresponding to the Hamiltonian may be computed as

```
H=-Da*e1*e1'-Db*e2*e2'+...
  i*Oma*(e1*e3'+e3*e1')+...
  i*Omb*(e2*e3'+e3*e2');
```

where in Matlab, the apostrophe denotes the conjugate transpose and the ellipsis indicates line continuation.

When we include the effects of spontaneous emission, the master equation in the interaction picture is

$$\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}t} = -\mathrm{i}[\hat{H}, \hat{\rho}] \\ + \sum_{k\in\{a,b\}} \left(\hat{C}_k\hat{\rho}\hat{C}_k^\dagger - \frac{1}{2}\hat{C}_k^\dagger\hat{C}_k\hat{\rho} - \frac{1}{2}\hat{\rho}\hat{C}_k^\dagger\hat{C}_k \right),$$

where the 'collapse operators' $\hat{C}_a$ and $\hat{C}_b$ are

$$\hat{C}_a = \sqrt{\gamma_a}|1\rangle\langle 3|, \qquad \hat{C}_b = \sqrt{\gamma_b}|2\rangle\langle 3|.$$

The right-hand side of the master equation is linear in the operator $\hat{\rho}$, and hence also in the components of the matrix $[\rho]$. The system of differential equations may be placed in the canonical matrix-vector form $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t)$ provided that the components of $[\rho]$ are regarded as a vector rather than as a matrix. We shall denote the column vector formed by taking the elements of $[\rho]$ columnwise by $\widetilde{[\rho]}$, and refer to the process of converting $[\rho]$ to $\widetilde{[\rho]}$ as 'flattening' the matrix. In Matlab, `rho(:)` denotes the flattened form of `rho`. Thus, for example, for a two-dimensional state space,

$$\widetilde{[\rho]} = \widetilde{\begin{pmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{pmatrix}} = \begin{pmatrix} \rho_{11} \\ \rho_{21} \\ \rho_{12} \\ \rho_{22} \end{pmatrix}.$$

On the right-hand side of the master equation, the density matrix $[\rho]$ is both pre-multiplied and post-multiplied by other matrices. The effect of pre-multiplying the matrix $[\rho]$ by a matrix $[a]$ can be written as an operation acting on the space of flattened matrices. For a two-dimensional Hilbert space,

$$[a][\rho] = \begin{pmatrix} a_{11}\rho_{11} + a_{12}\rho_{21} & a_{11}\rho_{12} + a_{12}\rho_{22} \\ a_{21}\rho_{11} + a_{22}\rho_{21} & a_{21}\rho_{12} + a_{22}\rho_{22} \end{pmatrix}.$$

Flattening $[a][\rho]$, we find that the result may be expressed as a 4×4 matrix acting on $\widetilde{[\rho]}$

$$\widetilde{[a][\rho]} = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 0 & a_{11} & a_{12} \\ 0 & 0 & a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} \rho_{11} \\ \rho_{21} \\ \rho_{12} \\ \rho_{22} \end{pmatrix} \\ = \texttt{spre}([a])\widetilde{[\rho]}.$$

Thus for an N-dimensional state space $\mathcal{H}$, corresponding to each $N \times N$ matrix $[a]$, there is an $N^2 \times N^2$ sparse matrix, here denoted by $\texttt{spre}([a])$, which when applied to $\widetilde{[\rho]}$ has the same effect as pre-multiplying $[\rho]$ by $[a]$. The extension to larger Hilbert spaces is straightforward, and it is similarly possible to find a matrix $\texttt{spost}([a])$ such that

$$\widetilde{[\rho][a]} = \texttt{spost}([a])\widetilde{[\rho]}.$$

In the toolbox, both `spre` and `spost` are provided as functions for converting operators into 'superoperators' acting on the space of flattened density matrices. For the three-level atom problem, the following lines of code thus generate the 9×9 Liouvillian matrix (which defines the right-hand side of the master equation) from the matrices defined above:

```
Ca=sqrt(ga)*e1*e3'; CadCa=Ca'*Ca;
Cb=sqrt(gb)*e2*e3'; CbdCb=Cb'*Cb;
L=-i*(spre(H)-spost(H))+...
  spre(Ca)*spost(Ca')-0.5*spre(CadCa)-...
  spost(CadCa)+...
  spre(Cb)*spost(Cb')-0.5*spre(CbdCb)-...
  spost(CbdCb);
```

These may be written down by inspection of the mathematical expression of the master equation, and their extension to more complicated atomic level schemes is immediate.

3. Composite quantum systems

We next consider a composite quantum system consisting of a two-level atom with transition frequency ω_0 in a coherently driven high-Q optical cavity which supports a single-mode light field of frequency ω_c with annihilation operator $\hat{a}$. The cavity field decay rate is κ and the spontaneous emission rate is γ. The frequency of the driving laser is ω_L. The state space is the tensor product of the state spaces for the cavity light field and for the atom, so that the dimensionality of the composite system is the product of the dimensions of the component spaces. The matrices for the composite system may be formed by taking the Kronecker product of the matrices in the component spaces.

In the toolbox, there are routines for returning the (sparse) matrix representations of the annihilation operator in a truncated (N-dimensional) Fock space (denoted by `destroy(N)`) and the angular momentum matrices (such as `jmat('x',1)` for the J_x operator of a spin 1 particle and `sigmay` for the Pauli $[\sigma_y]$ matrix, etc). The function `tensor` is used to apply the Kronecker product to its arguments so that, for example,

```
a=tensor(destroy(N),identity(2));
sm=tensor(identity(N),sigmam);
```

form the matrices $[a]$ and $[\sigma_-]$ corresponding to operators $\hat{a}$ and $\hat{\sigma}_-$ (or more precisely $\hat{a}\otimes\mathbf{1}$ and $\mathbf{1}\otimes\hat{\sigma}_-$, where $\mathbf{1}$ denotes the identity) in the composite space. More arguments may be supplied to the function `tensor` to cope with systems with more than two components.

Once these operators are defined, it is straightforward to compute the matrix representation of the Liouvillian $\mathcal{L}$ in the master equation. For this problem in an interaction picture,

$$\begin{aligned}\hat{H} &= (\omega_0-\omega_L)\hat{\sigma}_+\hat{\sigma}_- + (\omega_c-\omega_L)\hat{a}^\dagger\hat{a}\\ &\quad + \mathrm{i}g(\hat{a}^\dagger\hat{\sigma}_- - \hat{\sigma}_+\hat{a}) + \mathcal{E}(\hat{a}^\dagger+\hat{a}),\\ \mathcal{L}\rho &= -\mathrm{i}[\hat{H},\hat{\rho}] + \sum_{k=1}^{2}(\hat{C}_k\hat{\rho}\hat{C}_k^\dagger - \tfrac{1}{2}\hat{C}_k^\dagger\hat{C}_k\hat{\rho} - \tfrac{1}{2}\hat{\rho}\hat{C}_k^\dagger\hat{C}_k),\\ \hat{C}_1 &= \sqrt{2\kappa}\hat{a}, \qquad \text{and} \qquad \hat{C}_2 = \sqrt{\gamma}\hat{\sigma}_-.\end{aligned}$$

The program listing on the left of figure 2 shows how the matrices corresponding to $\hat{H}$ and to $\mathcal{L}$ are formed. The function `steady` computes the steady-state density matrix by finding a vector in the null space of the superoperator $\mathcal{L}$, and interprets the result as a density matrix by normalizing it to have unit trace.

The expectation value of an operator is a linear function of the density matrix and so for each operator $\hat{s}$, there must exist a row vector $\widehat{[s]}$ with the property that

$$\langle\hat{s}\rangle = \mathrm{trace}(\hat{s}\hat{\rho}) = \widehat{[s]}\widetilde{[\rho]}.$$

For example, in a two-dimensional Hilbert space,

$$\begin{aligned}\mathrm{trace}(\hat{s}\hat{\rho}) &= s_{11}\rho_{11} + s_{12}\rho_{21} + s_{21}\rho_{12} + s_{22}\rho_{22}\\ &= (\,s_{11} \quad s_{12} \quad s_{21} \quad s_{22}\,)\,\widetilde{[\rho]},\end{aligned}$$

and so

$$\widehat{[s]} = \left(\widetilde{[s]^{\mathrm{t}}}\right)^{\mathrm{t}}.$$

In the toolbox, the function `expect` computes expectation values of operators.

On the right-hand side of figure 2, graphs of the intensity of the output light from cavity $\langle\hat{C}_1^\dagger\hat{C}_1\rangle$ and the intensity of spontaneous emission $\langle\hat{C}_2^\dagger\hat{C}_2\rangle$ are shown as functions of the detuning $(\omega_L-\omega_0)$ of the driving light field from the common frequency of the cavity and the atomic transition $(\omega_c=\omega_0)$. As expected, the presence of the atom introduces a minimum in the output light from the cavity precisely at resonance and the probability of atomic excitation and the spontaneous emission are maximum at this frequency.

4. Exponential series

If the Liouvillian $\mathcal{L}$ is time-independent, the solution of the master equation $\mathrm{d}\hat{\rho}/\mathrm{d}t = \mathcal{L}\hat{\rho}$ for a given initial condition $\hat{\rho}(0)=\hat{\rho}_0$ may be formally written as

$$\widetilde{[\rho]} = \mathrm{e}^{[\mathcal{L}]t}\widetilde{[\rho_0]}.$$

If the matrix representing $[\mathcal{L}]$ is small enough, and if the eigenvalues $s_1,\ldots,s_{N^2}$ of $[\mathcal{L}]$ are non-degenerate, it is convenient to write the solution as

$$\widetilde{[\rho]}(t) = \sum_{j=1}^{N^2}\boldsymbol{A}^{(j)}\exp s_j t,$$

where each $\boldsymbol{A}^{(j)}$ is an N^2 element column vector. In order to find the components of $\boldsymbol{A}^{(j)}$, we diagonalize $[\mathcal{L}]$ writing it as

$$[\mathcal{L}]_{ik} = \sum_j v_{ij}s_j u_{jk},$$

where $\boldsymbol{V} = (v_{ij})$ is the matrix of eigenvectors of $[\mathcal{L}]$ and $\boldsymbol{U} = (u_{jk}) = \boldsymbol{V}^{-1}$, so that

$$\begin{aligned}\widetilde{[\rho]}_i(t) &= \sum_k(\mathrm{e}^{[\mathcal{L}]t})_{ik}\widetilde{[\rho_0]}_k\\ &= \sum_j\left(v_{ij}\sum_k u_{jk}\widetilde{[\rho_0]}_k\right)\exp(s_j t)\\ &= \sum_j v_{ij}r_j\exp(s_j t) = \sum_j a_i^{(j)}\exp(s_j t),\end{aligned}$$

where r_j are the components of the vector $\boldsymbol{r} = \boldsymbol{U}\widetilde{[\rho_0]} = \boldsymbol{V}^{-1}\widetilde{[\rho_0]}$. This shows that the vectors $\boldsymbol{A}^{(j)} = \{a_i^{(j)}\}$ can be computed from the matrix of eigenvectors $\boldsymbol{V}$ of $[\mathcal{L}]$ and from $\boldsymbol{r}$.

```
ida = identity(N); idatom = identity(2);
% Define cavity field and atomic operators
a  = tensor(destroy(N),idatom);
sm = tensor(ida,sigmam);
% Hamiltonian
H = (w0-wl)*sm'*sm + (wc-wl)*a'*a + ...
     i*g*(a'*sm - sm'*a) + E*(a'+a);
% Collapse operators
C1    = sqrt(2*kappa)*a;
C2    = sqrt(gamma)*sm;
C1dC1 = C1'*C1;
C2dC2 = C2'*C2;
% Calculate the Liouvillian
LH = -i * (spre(H) - spost(H));
L1 = spre(C1)*spost(C1')-0.5*spre(C1dC1)-0.5*spost(C1dC1);
L2 = spre(C2)*spost(C2')-0.5*spre(C2dC2)-0.5*spost(C2dC2);
L  = LH+L1+L2;
% Find steady state
rhoss = steady(L);
% Calculate expectation values
count1  = expect(C1dC1,rhoss);
count2  = expect(C2dC2,rhoss);
infield = expect(a,rhoss);
```

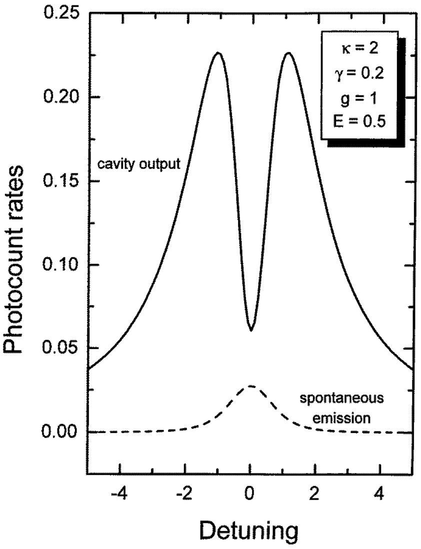


Figure 2. The left-hand side is a program fragment which sets up the Liouvillian for a two-level atom in a coherently driven high-Q cavity and which calls a routine to find the steady state. Graphs on the right-hand side show the steady-state cavity output intensity $\langle \hat{C}_1^\dagger \hat{C}_1 \rangle$ (solid curve) and the spontaneous emission intensity $\langle \hat{C}_2^\dagger \hat{C}_2 \rangle$ (dashed curve) as functions of the detuning of the laser driving field from the common frequency of the cavity light field and the atomic transition.

We shall refer to the representation of $[\widetilde{\rho}]$ as a sum of products of complex exponential functions of time with matrix-valued 'amplitudes' as an exponential series. The collection of 'rates' $\{s_j\}$ and 'amplitudes' $\{\boldsymbol{A}^{(j)}\}$ together define the exponential series. From the above, we see that the solution of a linear initial value problem with constant coefficients can sometimes be expressed as an exponential series. In the toolbox, a function `ode2es` performs this transformation.

Having obtained the solution of the system of equations as an exponential series, it is possible to evaluate this series at any value of t. This is often preferable to carrying out a direct numerical integration of the differential equations, which can be computationally intensive when t is large.

In figure 3, the Liouvillian calculated previously is used together with an initial condition indicating that the cavity field is in the vacuum state (which is the first basis element of an N-dimensional truncated Fock space, denoted by `basis(N,1)`) and that the atom is in the ground state (which is the second basis element of the two-dimensional state space of the atom, denoted by `basis(2,2)`). The solution for the density matrix `rhoES` is an exponential series, and the toolbox function `expect` is used to compute the expectation values of various operators. Note that in this example, since the second argument to `expect` is an exponential series, the resulting expectation values are also an exponential series, with scalar-valued amplitudes. The function `esval` evaluates the resulting exponential series at the collection of times specified in `tlist`. On the right-hand side of figure 3, the time-dependent expectation values $\langle \hat{C}_1^\dagger \hat{C}_1 \rangle$ and $\langle \hat{C}_2^\dagger \hat{C}_2 \rangle$ are shown when the driving field is on resonance. Note that the steady-state values coincide with those shown in figure 2 for zero detuning.

Exponential series are also useful for computations based on the quantum regression theorem (see e.g., [6]). If we work temporarily in the joint space of the system and the environment so that the evolution is specified by a total (time-independent) Hamiltonian $\hat{H}_{\text{tot}}$, the total density matrix evolves according to

$$\hat{\rho}_{\text{tot}}(\tau) = \exp(-\mathrm{i}\hat{H}_{\text{tot}}\tau)\hat{\rho}_{\text{tot}}(0)\exp(\mathrm{i}\hat{H}_{\text{tot}}\tau).$$

The time evolution of the expectation value of an operator such as $\hat{a}^\dagger$ is then

$$\langle \hat{a}^\dagger(\tau) \rangle = \text{trace}\{\hat{a}^\dagger \exp(-\mathrm{i}\hat{H}_{\text{tot}}\tau)\hat{\rho}_{\text{tot}}(0)\exp(\mathrm{i}\hat{H}_{\text{tot}}\tau)\}$$

where the operator $\hat{a}^\dagger(\tau)$ on the left-hand side is in the Heisenberg picture while those on the right are in the Schrödinger picture. If we now wish to find the two-time correlation function of the Heisenberg operator $\hat{a}(t)$, this may be written in the Schrödinger picture by using the identity

$$\begin{aligned}\langle \hat{a}^\dagger(t+\tau)\hat{a}(t) \rangle &= \text{trace}(\hat{a}^\dagger(t+\tau)\hat{a}(t)\hat{\rho}_{\text{tot}}(0)) \\ &= \text{trace}\{\hat{a}^\dagger \exp(-\mathrm{i}\hat{H}_{\text{tot}}\tau)\hat{a}\hat{\rho}_{\text{tot}}(t)\exp(\mathrm{i}\hat{H}_{\text{tot}}\tau)\}.\end{aligned}$$

Formally, the right-hand side is identical to that involved in the calculation of $\langle \hat{a}^\dagger(\tau) \rangle$, except that the initial condition $\hat{\rho}_{\text{tot}}(0)$ is replaced by $\hat{a}\hat{\rho}_{\text{tot}}(t)$. If the operators whose correlation is required are in the space of the system alone, the integration of the differential equation for time evolution may be carried out in the system space alone by utilizing the Liouvillian in place of the total Hamiltonian. The technique of converting an initial value problem into an exponential series discussed above is thus directly applicable. From the correlation function, it is simple to calculate quantities such as spectra. If the covariance function is written as a sum of exponentials

$$\langle \hat{a}^\dagger(t+\tau)\hat{a}(t) \rangle - \langle \hat{a}^\dagger(t+\tau) \rangle \langle \hat{a}(t) \rangle = \sum_k \phi_k \exp(-s_k|\tau|),$$

then the spectrum is given by

$$S(\omega) = \text{Re}\sum_k \frac{2\phi_k}{\mathrm{i}\omega - s_k}.$$

```
% Calculation of L as before
L = LH+L1+L2;
% Initially: vacuum field, atom in ground state
psi0 = tensor(basis(N,1),basis(2,2));
rho0 = psi0 * psi0';
% Calculate solution as an exponential series
rhoES = ode2es(L,rho0);
% Calculate expectation values
count1 = esval(expect(C1dC1,rhoES),tlist);
count2 = esval(expect(C2dC2,rhoES),tlist);
infield = esval(expect(a,rhoES),tlist);
```

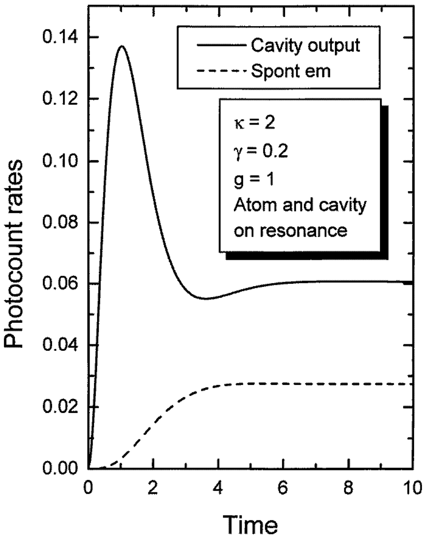


Figure 3. The left-hand side shows how a time-dependent solution may be obtained from the Liouvillian and the initial condition by converting the system of differential equations into an exponential series. Graphs on the right-hand side show the result of evaluating the exponential series to give the time variation of the cavity output intensity $\langle \hat{C}_1^\dagger \hat{C}_1 \rangle$ (solid curve) and the intensity of spontaneous emission $\langle \hat{C}_2^\dagger \hat{C}_2 \rangle$ (dashed curve).

In the toolbox, this transformation is carried out using the function `esspec`. Figure 4 shows the code required to carry out these computations and the resulting spectrum for two amplitudes of coherent driving for the two-level atom in the cavity.

5. ‘Quantum objects’ and the toolbox

In the above example, we see that it is convenient to consider an exponential series as a single object, even though it is actually composed of a collection of rates and amplitudes (each of which may be a scalar, vector or matrix). Compatible exponential series may be meaningfully combined by operations such as addition and multiplication, and it is useful to be able to pass an exponential series as an argument to a function such as `expect`. By defining a new class of objects to represent exponential series, and ‘overloading’ the arithmetic operators so that operations on these series can be carried out using natural notation, it becomes possible to manipulate them conveniently within programs.

Although we have so far described states, operators and superoperators as if they were vectors and matrices, in the toolbox, they are actually instances of a class `qo` which stands for a ‘quantum object’. The advantage of encapsulating the data in this way is that information about the dimensions of the Hilbert space over which the object is defined may be included. Without such auxiliary information, a 9 × 9 matrix might represent an operator over a nine-dimensional Hilbert space, a superoperator over a three-dimensional Hilbert space or various other possibilities. By keeping track of the sizes of the component spaces, it becomes possible to distinguish operators from superoperators, to check if operators are being applied to states with compatible dimensions and to write routines which calculate partial traces of density matrices over subsystems.

It also turns out to be useful to consider arrays of related quantum objects and to denote them using a single name. For example we can group together the angular momentum operators for a spin 2 particle, $\hat{J}_x$, $\hat{J}_y$ and $\hat{J}_z$ into a single quantum object array by using the notation:

```
J={jmat('x',2),jmat('y',2),jmat('z',2)};
```

After this definition, each array member (`J{1}`, `J{2}` and `J{3}`) is an operator represented by a 5 × 5 matrix. Another use of quantum object arrays is to store quantum objects which depend on time or space. For example, if we use `esval` to evaluate a operator-valued exponential series at a list of times, the result is an array of operators, the ith array member being the value of the operator at the ith time in the list.

Although the rules for combining quantum objects cannot be described in detail here, they are designed so that operations such as applying an operator to a vector can be represented ‘naturally’ using the multiplication sign. For example, if both of the operands are in fact arrays, operations take place between corresponding array members. Exponential series are actually defined as a subclass of the class of quantum object arrays, and are constructed so that the members of the array are the amplitudes of the series. They inherit the methods which apply to quantum object arrays, except when these are overridden to reflect the somewhat different rules which apply for combining exponential series. As will become clearer, much of the notational convenience of the toolbox arises from the use of object-oriented ideas such as inheritance, operator overloading and function polymorphism. During normal use of the toolbox, it is not usually necessary to be aware of the details of these low-level issues.

6. Semi-classical laser light force on atoms

As further examples of the way in which the toolbox helps the user to concentrate on the essential physics of a problem

```
% Calculation of L as before
L = LH+L1+L2;
% Find steady state density matrix and field
rhoss = steady(L);
ass = expect(a,rhoss);
% Initial condition for regression theorem
arho = a*rhoss;
% Solve differential equation
solES = ode2es(L,arho);
% Find trace(a' * solution)
corrES = expect(a',solES);
% Calculate the covariance
covES = corrES - ass'*ass;
% Find power spectrum
wlist = linspace(-10,10,200);
Ps = esspec(covES,wlist);
```

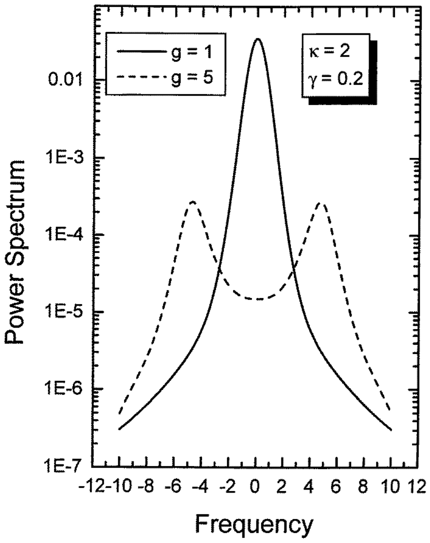


Figure 4. The left-hand side shows how two-time correlations and spectra may be found using the quantum regression theorem by modifying the initial condition of the master equation. Graphs on the right-hand side show the spectrum of the cavity output light for two values of the coherent driving.

and of the utility of constructs such as exponential series, we consider the problem of finding the force on an atom moving in a light field [7]. Such calculations are important for laser cooling, where the light force provides viscous damping to decelerate rapidly moving atoms. For two-level atoms, cooling to the Doppler limit is possible, while for multi-level atoms in light fields with polarization gradients, sub-Doppler cooling can occur [8]. It is usual in laser cooling calculations to suppose that the atoms are massive so that their speeds remain approximately constant as they traverse a single wavelength of the light field. The force on the atom may be averaged over a wavelength as the atom is dragged at constant speed v through the light field. The atomic motion is then treated classically as being that of a point particle experiencing a force given by the cycle-averaged force as a function of the atomic velocity. If the light is assumed to be a classical, coherent field, the state space is that of the atom alone.

Physically, the main difference between this problem and that of a stationary atom in a classical light field is that the atom experiences a time-varying field as it moves within the light beams. Since the velocity is assumed to be constant, the sinusoidal spatial variation translates into a periodic sinusoidal temporal variation. It is still necessary to solve the master equation $\dot\rho = \mathcal{L}\rho$, but the Liouvillian is now a sinusoidal function of time. The force on the atom is the expectation value of an operator, but this operator is also varying across a cycle of the light field. Since a periodic variation may be represented by Fourier series, all of these operators may be written as exponential series where the rates are integer multiples of the base rate $\mathrm{i}2\pi/T$ where T is the cycle time.

For a two-level atom (with ground state $|g\rangle$ and excited state $|e\rangle$) in a standing wave field, the Hamiltonian in the dipole approximation in an interaction picture is

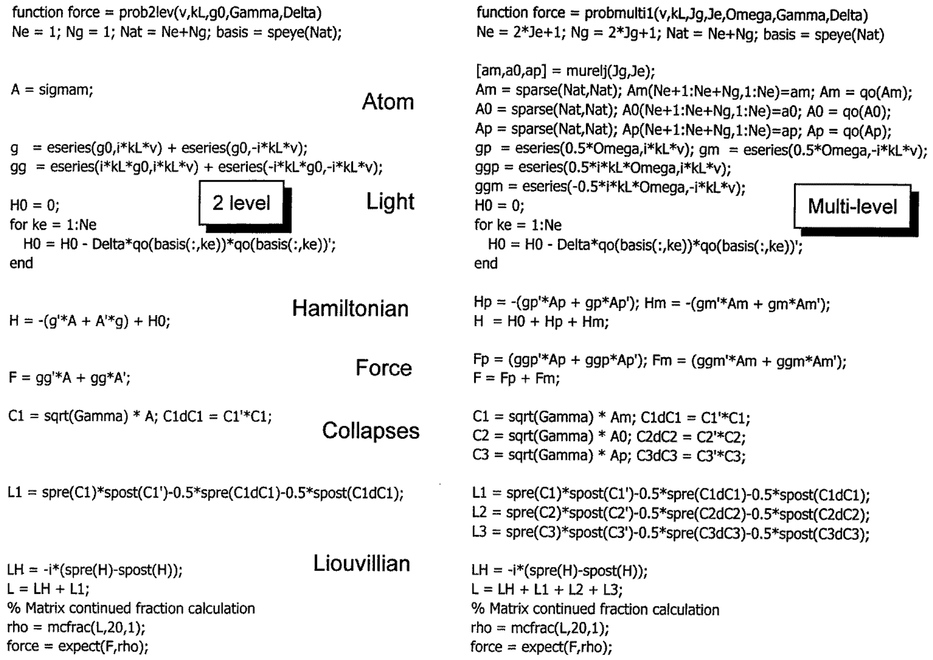
$$\hat{H}(z) = -\Delta|e\rangle\langle e| + \hat{V}(z) \qquad \text{where}$$
$$\hat{V}(z) = -(g^*(z)\hat{\sigma}_- + g(z)\hat{\sigma}_+),$$

Δ is the difference between the laser frequency and the atomic transition frequency, $\hat{\sigma}_- = \hat{\sigma}_+^\dagger = |g\rangle\langle e|$ is the atomic lowering operator and $g(z) = 2g_0\cos k_L z$ indicates the spatial variation of the field amplitude in the standing wave. Setting $z = vt$, we obtain $\hat{H}(t)$, the time-varying Hamiltonian. From the Hamiltonian, the Liouvillian is $\mathcal{L}(t)\hat\rho = -\mathrm{i}[\hat{H}(t), \hat\rho] + \hat{C}^\dagger\hat\rho\hat{C} - \frac{1}{2}\hat{C}^\dagger\hat{C}\hat\rho - \frac{1}{2}\hat\rho\hat{C}^\dagger\hat{C}$, where $\hat{C} = \sqrt{\gamma}\hat{\sigma}_-$ is the collapse operator for spontaneous emission.

The listing on the left-hand side of figure 5 shows how the master equation is set up for this problem. The steps by which we form the Liouvillian superoperator $\mathcal{L}$ from the Hamiltonian and the collapse operators look formally very similar to those in the previous examples. However, because the optical potential in the Hamiltonian is an exponential series in this case (since g is defined to be an exponential series), the Liouvillian is also automatically an exponential series. Solving a master equation with a sinusoidally-varying Liouvillian can be carried out using the method of matrix continued fractions [9], which allows the resulting periodically-varying density matrix $\hat\rho(t)$ to be written as an exponential (actually a Fourier) series. This calculation is carried out by the toolbox function `mcfrac`. The force operator is also an exponential series,

$$\hat{F}(z) = -\boldsymbol{\nabla}\hat{V} = \nabla g^*(z)\hat{\sigma}_- + \nabla g(z)\hat{\sigma}_+.$$

Using polymorphism, the function `expect` is written so as to be able to find the expectation value of an operator which is expressed as an exponential series when the state is also expressed as an exponential series. The constant term of the result is the cycle-averaged force. Figure 6(a) shows the result for the cycle-averaged force on a two-level atom for the parameters used in the paper by Minogin and Serimaa [10]. The laser power is much larger than would be used in a laser cooling experiment, but effects such as Doppleron resonances are visible in the graph, which agrees with that published


Atom
Light
2 level
Multi-level
Hamiltonian
Force
Collapses
Liouvillian

```
function force = prob2lev(v,kL,g0,Gamma,Delta)
Ne = 1; Ng = 1; Nat = Ne+Ng; basis = speye(Nat);

A = sigmam;

g  = eseries(g0,i*kL*v) + eseries(g0,-i*kL*v);
gg = eseries(i*kL*g0,i*kL*v) + eseries(-i*kL*g0,-i*kL*v);

H0 = 0;
for ke = 1:Ne
  H0 = H0 - Delta*qo(basis(:,ke))*qo(basis(:,ke))';
end

H = -(g'*A + A'*g) + H0;

F = gg'*A + gg*A';

C1 = sqrt(Gamma) * A; C1dC1 = C1'*C1;

L1 = spre(C1)*spost(C1')-0.5*spre(C1dC1)-0.5*spost(C1dC1);

LH = -i*(spre(H)-spost(H));
L = LH + L1;
% Matrix continued fraction calculation
rho = mcfrac(L,20,1);
force = expect(F,rho);
```

```
function force = probmulti1(v,kL,Jg,Je,Omega,Gamma,Delta)
Ne = 2*Je+1; Ng = 2*Jg+1; Nat = Ne+Ng; basis = speye(Nat)

[am,a0,ap] = murelj(Jg,Je);
Am = sparse(Nat,Nat); Am(Ne+1:Ne+Ng,1:Ne)=am; Am = qo(Am);
A0 = sparse(Nat,Nat); A0(Ne+1:Ne+Ng,1:Ne)=a0; A0 = qo(A0);
Ap = sparse(Nat,Nat); Ap(Ne+1:Ne+Ng,1:Ne)=ap; Ap = qo(Ap);
gp  = eseries(0.5*Omega,i*kL*v); gm  = eseries(0.5*Omega,-i*kL*v);
ggp = eseries(0.5*i*kL*Omega,i*kL*v);
ggm = eseries(-0.5*i*kL*Omega,-i*kL*v);
H0 = 0;
for ke = 1:Ne
  H0 = H0 - Delta*qo(basis(:,ke))*qo(basis(:,ke))';
end

Hp = -(gp'*Ap + gp*Ap'); Hm = -(gm'*Am + gm*Am');
H  = H0 + Hp + Hm;

Fp = (ggp'*Ap + ggp*Ap'); Fm = (ggm'*Am + ggm*Am');
F = Fp + Fm;

C1 = sqrt(Gamma) * Am; C1dC1 = C1'*C1;
C2 = sqrt(Gamma) * A0; C2dC2 = C2'*C2;
C3 = sqrt(Gamma) * Ap; C3dC3 = C3'*C3;

L1 = spre(C1)*spost(C1')-0.5*spre(C1dC1)-0.5*spost(C1dC1);
L2 = spre(C2)*spost(C2')-0.5*spre(C2dC2)-0.5*spost(C2dC2);
L3 = spre(C3)*spost(C3')-0.5*spre(C3dC3)-0.5*spost(C3dC3);

LH = -i*(spre(H)-spost(H));
L = LH + L1 + L2 + L3;
% Matrix continued fraction calculation
rho = mcfrac(L,20,1);
force = expect(F,rho);
```

Figure 5. Program fragments for calculating the laser light force on an atom moved at constant velocity within the light field. Listing on the left is for a two-level atom in a standing wave light field while listing on the right is for a $J_g \leftrightarrow J_e$ transition in a multi-level atom moving in counter-propagating σ_+ and σ_- polarized light beams. Corresponding lines of code are aligned to show the formal similarity in the formulation of the problems. Although many of the operators are periodically time-varying, they can be manipulated simply via the use of exponential series objects.

in [10], but which are not seen in a low-order perturbative calculation of the force.

We next show consider how the problem changes if a multi-level atom with transition $J_g \leftrightarrow J_e$ is used with counter-propagating beams of σ_+ and σ_- light, a configuration discussed by Dalibard and Cohen-Tannoudji [8]. Due to the more complicated atomic level structure, we expect the expression for the optical potential $\hat{V}(z)$ to change. Instead of a single lowering operator σ_- for the two-level atom which couples to the single polarization of light, there are now three operators $\hat{A}_{+1}$, $\hat{A}_0$ and $\hat{A}_{+1}$ which are couple with light having σ_+, π and σ_- polarizations respectively. Each of these operators are linear combinations of the appropriate ground and excited states of the atom. In fact,

$$\hat{A}_\sigma = \sum_{m_e, m_g} |J_g, m_g\rangle\langle J_g, m_g; 1, \sigma|J_e, m_e\rangle\langle J_e, m_e|,$$

where $\langle J_g, m_g; 1, \sigma|J_e, m_e\rangle$ are the Clebsch–Gordon coefficients for the dipole transition $|e\rangle$ to $|g\rangle$ with polarizations $\sigma = 0, \pm 1$. When written as a matrix, using the atomic levels as the basis, the formal similarity between σ_- for the two-level atom and $\hat{A}_\sigma$ for the multi-level atom becomes apparent. We have that

$$\hat{\sigma}_- = \begin{array}{c} \begin{array}{cc} \mathrm{e} & \mathrm{g} \end{array} \\ \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \end{array} \begin{array}{c} \mathrm{e} \\ \mathrm{g} \end{array} \qquad \text{and} \qquad \hat{A}_\sigma = \begin{array}{c} \begin{array}{cc} \mathrm{e} & \mathrm{g} \end{array} \\ \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{a}_\sigma & \mathbf{0} \end{pmatrix} \end{array} \begin{array}{c} \mathrm{e} \\ \mathrm{g} \end{array},$$

where e and g denote the excited and ground states respectively. For the two-level atom, there is only one excited and one ground state, whereas for the multi-level atom, there are $2J_e + 1$ excited states and $2J_g + 1$ ground states. The $(2J_g + 1) \times (2J_e + 1)$ matrix $\boldsymbol{a}_\sigma$ contains the Clebsch–Gordon coefficients for the appropriate polarizations. In the toolbox, the function `murelj` calculates these coefficients and returns the matrices $\boldsymbol{a}_0$, $\boldsymbol{a}_{\pm 1}$.

The interaction potential is given by

$$\hat{V}(z) = -\sum_{\sigma=0,\pm 1} \{g_\sigma^*(z)\hat{A}_\sigma + g_\sigma(z)\hat{A}_\sigma^\dagger\},$$

where $g_\sigma(z)$ specifies the light field for polarization σ. For example, a σ_+ polarized beam propagating to the right and a σ_- beam propagating to the left is indicated by $g_{+1}(z) = \frac{1}{2}\Omega \exp(\mathrm{i}k_L z)$, $g_{-1}(z) = \frac{1}{2}\Omega \exp(-\mathrm{i}k_L z)$ and $g_0(z) = 0$. In the listing on the right-hand side of figure 5, the code for the multi-level situation is shown for this configuration of light beams. The listings for the two-level case and the multi-level case are aligned to emphasize the structural similarities. The only differences appear in the calculation of $\hat{V}(z)$ and in the fact that there are more channels for spontaneous emission available in the multi-level case. The results are shown in figure 6(*b*), in which the parameters have been chosen to coincide with those in the article by Dalibard and Cohen-Tannoudji [8] for a $J_g = 1 \leftrightarrow J_e = 2$ transition. The large slope in the force-velocity graph for small velocities shows the increased viscosity introduced by the polarization-gradient cooling which allows the Doppler limit to be broken. From the above discussion, it should be clear that only straightforward modifications are needed to solve for other configurations of light fields (such as linear ⊥ linear polarizations) or other atomic level schemes.

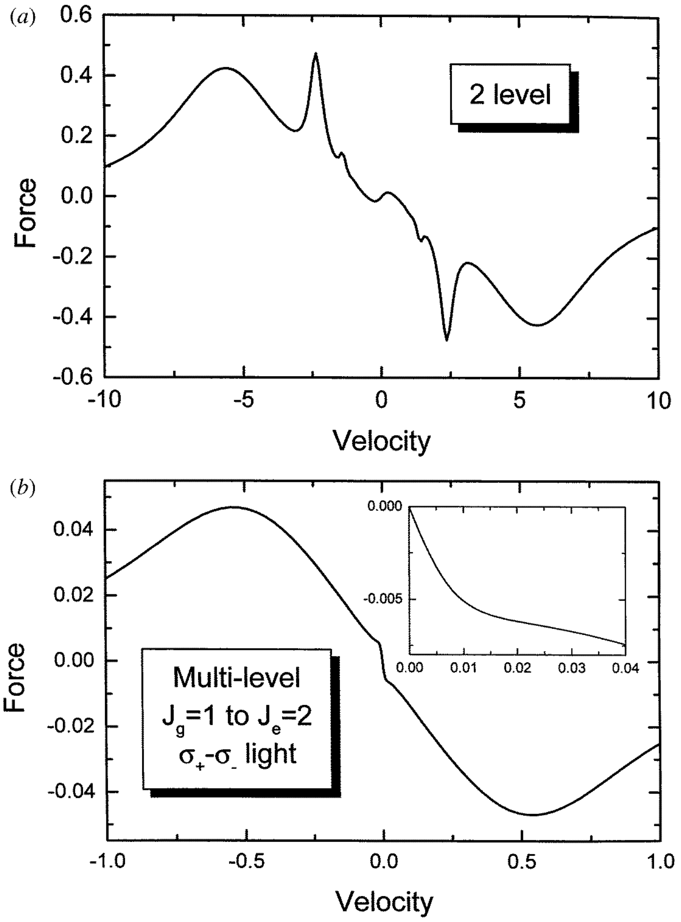


Figure 6. (*a*) Shows the variation of cycle-averaged force with atomic velocity for a two-level atom in a standing wave when $k_L = 1$, $g_0 = \sqrt{25/8}$, $\gamma = 1$ and $\Delta = -5$. (*b*) Shows the variation of cycle-averaged force with atomic velocity for a multi-level atom with a $J_g = 1 \leftrightarrow J_e = 2$ transition in counter-propagating σ_+ and σ_- polarized light beams. The parameters are $k_L = 1$, $\Omega = 0.25$, $\gamma = 1$ and $\Delta = -0.5$. Inset is an expanded portion of the curve, showing increased viscosity at low velocities. Velocity is measured in units of γ/k_L and force is in units of $\hbar k_L \gamma$.

7. Problems requiring numerical integration

For many practical problems, the size of the state spaces involved are so large that there is no alternative to direct numerical integration of the equations of motion. This can also be the case if there are time-dependencies in the problem which cannot be treated by special-purpose algorithms such as matrix continued fractions. The toolbox can still be usefully employed for such problems as it allows a representation of the equations of motion to be set up straightforwardly with respect to some basis in terms of sparse matrices. An outline of the process is given below, but the details of the specific functions required to use the facilities in practice have been omitted.

Depending on the size of the problem and whether it is coupled to an external bath, it may be useful to integrate the Schrödinger equation directly, to integrate the master equation for the density matrix, or to use a quantum trajectory method [11, 12] for evaluating averages over a mixed state via stochastic wavefunction simulation. In the first two cases, the general form of the problem may be expressed in matrix-vector notation as

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{A}(t)\boldsymbol{x},$$

where $\boldsymbol{x}$ might represent a state vector or a flattened density matrix. In general, $\boldsymbol{A}(t)$ is a large, sparse matrix which can depend on time in an arbitrary way. For most problems of interest the time variation is due to detunings, the profiles of light beams etc, for which $\boldsymbol{A}(t)$ may be written as a series of simple scalar functions of t multiplied by ‘amplitudes’ which are sparse matrices, i.e.,

$$\boldsymbol{A}(t) = \boldsymbol{A}_1 f_1(t) + \boldsymbol{A}_2 f_2(t) + \cdots + \boldsymbol{A}_m f_m(t).$$

This is called a ‘function series’ by analogy with the exponential series defined previously. Typically, the functions $f_i(t)$ are simple, and may involve complex exponentials, Gaussian pulses, modulated pulses and the like.

Within the toolbox, large problems are handled by using Matlab to formulate the problem and to produce a data file which may be read into an external program (written in C) which carries out the actual numerical integration. For deterministic Schrödinger and master equations, a problem is specified by the function series on the right-hand side, the initial condition, and the list of times at which the solution is required. After the integration program has performed the integration and produced the results, the data are read back into Matlab for subsequent processing and display.

Routines are also provided which allow quantum trajectory simulations to be carried out. In this case, if the master equation in Linblad form is

$$\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}t} = -\mathrm{i}[\hat{H}, \hat{\rho}] + \sum_k \left(\hat{C}_k \hat{\rho} \hat{C}_k^\dagger - \frac{1}{2} \hat{C}_k^\dagger \hat{C}_k \hat{\rho} - \frac{1}{2} \hat{\rho} \hat{C}_k^\dagger \hat{C}_k \right),$$

the problem is characterized in terms of the effective Hamiltonian

$$\hat{H}_{\mathrm{eff}} = \hat{H} - \frac{\mathrm{i}}{2} \sum_k \hat{C}_k^\dagger \hat{C}_k,$$

and the list of collapse operators $\{\hat{C}_k\}$. Each of these operators can be a function series, if necessary. A data file containing these operators, together with the initial conditions is passed to a generic program written in C which carries out as many trajectories of the quantum Monte Carlo algorithm as desired. It is possible to record all of the stochastic wavefunctions, or alternatively to specify a list of operators whose expectation values are required (averaged over the trajectories). A record of the times at which the collapses occur (the ‘classical record’) is also obtained during the simulation.

Besides carrying out trajectory simulations involving quantum jumps, the toolbox also provides routines for quantum state diffusion calculations, which are useful for simulating the effects of homodyne and heterodyne measurements made on the output light from a system.

8. Conclusions

The toolbox of routines described here has proved useful for writing programs for simulating a variety of quantum systems as well as for teaching the quantum mechanics of open systems. In this paper, the close correspondence between the mathematical formulation of the physical problem and the lines of code required to simulate the problem has been emphasized. The structures of the objects in the toolbox are such that they are also useful for simulating simple quantum computing and information processing systems where dynamics conditioned on observations are important. Besides its role in simulation, the toolbox also provides facilities for visualizing quantum states using quasi-probability distributions such as Wigner and Husimi Q functions.

Acknowledgments

I thank H Jeff Kimble for his hospitality during a period of sabbatical leave at the California Institute of Technology at which this work was started. Financial support was provided by the University of Auckland Research Council and the Marsden Fund of the Royal Society of New Zealand.

References

[1] Brandt S and Dahmen H 1989 *Quantum Mechanics on The Personal Computer* (Berlin: Springer)
[2] Hiller J R, Johnston I D and Styer D F 1995 *Quantum Mechanics Simulations: The Consortium for Upper-Level Physics Software* (New York: Wiley)
[3] Shack R and Brun T 1997 *Comp. Phys. Commun.* **102** 210
[4] Webpage URL: `http://www.mathworks.com` (The Mathworks, MA)
[5] Meystre P and Sargent M III 1990 *Elements of Quantum Optics* (Berlin: Springer)
[6] Swain S 1981 *J. Phys. A: Math. Gen.* **14** 2577
[7] Gordon J and Ashkin A 1980 *Phys. Rev.* A **21** 1606
[8] Dalibard J and Cohen-Tannoudji C 1989 *J. Opt. Soc. Am.* B **6** 2023.
[9] Risken H 1989 *The Fokker–Planck Equation* (Berlin: Springer)
[10] Minogin V and Serimaa T 1979 *Opt. Commun.* **30** 373
[11] Gardiner C, Parkins A and Zoller P 1992 *Phys. Rev.* A **46** 4363
[12] Carmichael H 1993 *An Open Systems Approach to Quantum Optics* (Berlin: Springer)