Crash course on numerical methods for the Schrödinger equation, version 0.1

Simen Kvaal

January 21, 2009

1 Introduction

In these notes, I will try to assemble some general facts concerning numerical integration of the (non-relativistic) Schrödinger equation. I will assume that the reader knows basic quantum mechanics, linear algebra, and so on, but I will try to be as specific as possible. I will assume some knowledge of MATLAB and give some examples where appropriate.

It is natural to split the treatment into spatial and temporal discretization, respectively. Usually, it is possible to "factor" the definition of a complete numerical scheme in this way: First, we define a spatial discretization, giving a "semidiscrete" scheme which in fact is equivalent to a coupled set of prdinary differential equations. Second, we use some appropriate numerical scheme for this set of ODEs.

I will denote operators and abstract Hilbert space vectors with ordinary letters and kets, respectively, i.e., H, U, $|\psi\rangle$ and so on, and use boldface for concrete finite dimensional matrices and vectors in $\mathbb{C}^N$, i.e., $\mathbf{H}$, $\mathbf{U}$, $\mathbf{y}$ and so on. Note that in the finite-dimensional case, every Hilbert space $\mathscr{H}$ *is* essentially $\mathbb{C}^N$ for $N = \dim(\mathscr{H})$, and matrices *are* the operators and vectors the kets.

2 Concerning the Hamiltonian

We begin by recalling some basic facts from abstract quantum mechanics. We consider a completely general Hamiltonian $H = H^\dagger$, which we first assume is *independent of time*. This operator acts in a given Hilbert space $\mathscr{H}$, which in general is infinite-dimensional, such as $L^2(\mathbb{R}^3)$ for a single particle in three spatial dimensions.

The time-dependent Schrödinger equation gives the dynamics of a state $|\psi\rangle$ as function of time, i.e., it determines $|\psi(t)\rangle$ for each t. The time-dependent Schrödinger equation is written

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

with $|\psi(0)\rangle$ a given initial condition. The equation is also valid in the more general case of a time-dependent Hamiltonian $H(t)$. For very general time-independent Hamiltonians it is a fact that such dynamics can be found for all initial states $|\psi(0)\rangle$, and in that case the solution $|\psi(t)\rangle$ can be written

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle \equiv \exp(-itH/\hbar)|\psi(0)\rangle, \tag{2}$$

where $U(t)$ is a unitary operator with

$$U(t)^{-1} = U(t)^\dagger = U(-t). \tag{3}$$

Notice that the initial time $t = 0$ is completely arbitrary, so that we have

$$U(t+t') = U(t)U(t'). \tag{4}$$

Usually, the exponential $\exp(A)$ of an operator A can be taken to be defined in terms of the Taylor series for $\exp(z)$, viz,

$$\exp(A) \equiv \sum_{n=0}^{\infty} \frac{1}{n!} A^n = 1 + A + \frac{1}{2}A^2 + \cdots \tag{5}$$

In this text I shall take this viewpoint, even though it is only completely rigorous in the case of finite-dimensional Hilbert spaces. (In the end, we shall only be concerned with the dynamics of semidiscrete versions, which *are* finite dimensional, e.g., since we have a finite number of grid points or degrees of freedom.)

Again, in the case of a time-independent system (i.e., time-independent Hamiltonian), it is easy to see that *if* the eigenvalue problem for H can be solved, we have in fact a situation where the dynamics is trivial to compute. Assume that we can find a complete set of orthonormal eigenvectors $|\psi_k\rangle \in \mathscr{H}$ and corresponding eigenvalues E_k, i.e.,

$$H|\psi_k\rangle = E_k|\psi_k\rangle, \quad \langle\psi_k, \psi_\ell\rangle = \delta_{k,\ell} \tag{6}$$

such that the identity operator can be written

$$1 = \sum_{k=0}^{\infty} |\psi_k\rangle\langle\psi_k|. \tag{7}$$

This last relation is the same as saying that *any* wave function can be expanded in the $|\psi_k\rangle$, viz,

$$|\psi\rangle \equiv \sum_{k=0}^{\infty} |\psi_k\rangle\langle\psi_k|\psi\rangle = \sum_{k=0}^{\infty} c_k|\psi_k\rangle, \tag{8}$$

with the coefficients $c_k = \langle\psi_k|\psi\rangle$ being unique. Moreover, we have

$$H \equiv \sum_{k=0}^{\infty} E_k|\psi_k\rangle\langle\psi_k|, \tag{9}$$

i.e., "the Hamiltonian is diagonal in the eigenvector basis". The time-dependent Schrödinger equation for $|\psi_k\rangle$ reads

$$i\hbar\frac{\partial}{\partial t}|\psi_k(t)\rangle = E_k|\psi_k(t)\rangle, \tag{10}$$

so that, trivially,

$$|\psi_k(t)\rangle = e^{-itE_k/\hbar}|\psi_k(0)\rangle. \tag{11}$$

This means that if the system starts out in the eigenstate $|\psi_k\rangle \equiv |\psi_k(0)\rangle$, it will only gain a new phase with time.

3 Splitting of the Hamiltonian

3.1 Different splittings

In these notes, we will encounter several versions of splittings of H, depending on the setting. For example, for the split-step technique for a single particle in an external potential in one dimension, we have

$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x,t) \equiv T + V. \tag{12}$$

In the split-step technique we treat the potential V and the kinetic operator T separately, since they are natural operators in the position domain and the frequency (momentum) domain, respectively.

Another splitting can come from a time-independent part and a time-dependent part. Suppose that the potential $V(x,t)$ contains a time-dependent perturbation, viz,

$$H = T + V + W(t). \tag{13}$$

Then it might be useful split the Hamiltonian between $H_0 = T + V$ and $H_1(t) = W(t)$.

Both of these splittings might be present at once, such as in the Blanes–Moan method. On the other hand, if for example the eigenvalue problem for $H_0 = T + V$ (which is time-independent!) is "solved" (such as for the case of using the output from, e.g., my quantum dot code), then we do not wish to split H_0: It is much easier to compute the matrix of $H_1(t)$ with respect to these eigenvectors. More on this later.

3.2 Split-step technique: Strang splitting

A "natural" splitting of $H = T + V$ can be used to define a good approximation to the propagator $U(t) = \exp[-it(T+V)]$. If T and V were ordinary numbers, then $[T,V] = 0$ and we would have

$$U(t) = \exp[-it(T+V)] = \exp(-itT)\exp(-itV). \tag{14}$$

In general, since $[T,V] \neq 0$ this is not true (why?) so instead we get

$$U(t) = \exp[-it(T+V)] = \exp(-itT)\exp(-itV) + O(t^2). \tag{15}$$

If t is very small, then the error of order $O(t^2)$ can be very small.

A further trick can be made: It is not so difficult yo show using the BCH formula, that

$$U(t) = \exp[-it(T+V)] = \exp(-itT/2)\exp(-itV)\exp(-itT/2) + O(t^3). \tag{16}$$

This particular splitting is often called Strang splitting (after Gilbert Strang, a really cool guy). There are other possible splitting schemes as well, of course, many can be obtained by playing DJ: propagating back and forth with different time steps which add up to t, but with resulting accuracy of very high order!

Physically, we may view the approximate $U(t)$ as being composed of two alternating quantum mechanical processes: First, the potential $V = 0$ for a period $t/2$, referred to as "drift". Then the potential is turned on very strongly for a time t, called a "kick", and then a drift again for a period of $t/2$ completes this approximate propagation.

Now, if $\exp(-itT/2)$ and/or $\exp(-itV)$ can be computed *cheaply*, we have a useful approximation! Notice, that the splitting of H is, in principle, arbitrary. The result (16) is *not* dependent of T being kinetic energy or V being a potential energy. Indeed, if we had the eigenvectors of H available, we may use the trivial splitting "$T = H$" and "$V = 0$", and computing $\exp(-itT/2)$ is indeed cheap! In this case, since $[T,V] = 0$ the example is somewhat contrieved, but you get the idea.

4 Semi-discrete formulations

In this section, we shall see how different discretizations in *space* leads to a finite-dimensional matrix formulation of the original Schrödinger equation. In fact, such a discretization leads to the consideration of a *different quantum system with only finitely many dimensions*. This new system is an approximation to the complete system, but is in itself a case of the completely general quantum system! Think about this, it is very important. For example, the

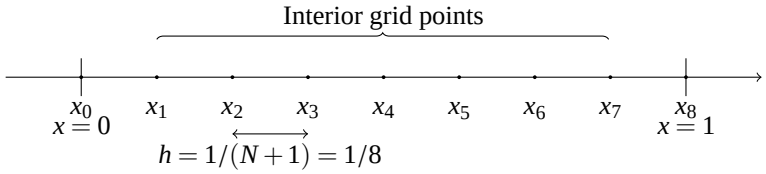

Figure 1: Illustration of a simple finite difference grid with $N = 7$ interior grid points, dividing the interval $[0,1]$ into $N+1 = 8$ equal sub-intervals of length $h = 1/(N+1)$

propagator $U(t)$ is well-defined for this system, along with the eigenvalue problem for H and so on, except that it's *all in terms of finite matrices*!

The spatial discretization (such as FFT, finite difference, ...) always gives a semi-discrete Schrödinger equation on the form:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) = \mathbf{H}\mathbf{y}(t), \quad \mathbf{y}(t) \in \mathbb{C}^N, \tag{17}$$

where $\mathbf{H} \in \mathbb{C}^{N\times N}$ is an $N \times N$ Hermitian matrix. Notice that the system is still continuous in time t. Our Hilbert space is now $\mathscr{H} = \mathbb{C}^N$.

If the original Hamiltonian $H = T + V$, then $\mathbf{H} = \mathbf{T} + \mathbf{V}$, i.e, splittings carry over to the semi-discrete formulation. However, *different* discretizations leads to *different* $\mathbf{T}$ and/or $\mathbf{V}$.

Let's see this through some examples.

4.1 Finite difference approximation

Let's consider a single particle of mass $m = 1$ living on the interval $[0,1]$, that is to say there are infinitely high walls outside $[0,1]$. An additional possibly time-dependent potential $V(x,t)$ is added. The particle has no spin, so the exact infinite-dimensional Hilbert space is $L^2[0,1]$. The Hamiltonian is

$$H = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + V(x,t). \tag{18}$$

The time-dependent Schrödinger equation is (the abstract equation (1) spelled out)

$$i\psi_t(x,t) = -\frac{1}{2}\psi_{xx}(x,t) + V(x,t)\psi(x,t), \tag{19}$$

$$\psi(x,0) = f(x) \in L^2[0,1] \qquad \text{(initial condition)} \tag{20}$$

$$\psi(0,t) = \psi(1,t) = 0, \qquad \text{(boundary condition)} \tag{21}$$

where the last equation are the boundary conditions imposed by the hard walls.

The discretization consists of picking N equally spaced points inside the interval $[0,1]$, i.e.,

$$x_j \equiv \frac{j}{N+1} \equiv jh, \quad 1 \le j \le N, \tag{22}$$

and at time t approximate $\psi(x_j) \approx y_j$, such that a time-dependent vector $\mathbf{y}(t) \in \mathbb{C}^N$ is defined, see Figure 1.

We approximate the differential operator $\partial^2/\partial x^2$ by a central finite difference, viz,

$$\psi_{xx}(x_j) = \frac{1}{h^2}[\psi(x_{j-1}) - 2\psi(x_j) + \psi(x_{j+1})] + O(h^2), \tag{23}$$

so that for the j'th grid point, the action of the discrete Hamiltonian on $\mathbf{c}$ is defined to be

$$(\mathbf{Hy})_j \equiv [(\mathbf{T}+\mathbf{V})\mathbf{y}]_j = -\frac{1}{2h^2}(y_{j-1}-2y_j+y_{j+1})+V(x_j)y_j, \quad 1 \le j \le N. \tag{24}$$

Here, we must define $y_0 = y_{N+1} = 0$ to incorporate the boundary conditions. Since the action of a matrix $\mathbf{H}$ on $\mathbf{y}$ can be written

$$(\mathbf{Hy})_j = \sum_{k=1}^{N} H_{jk}y_k, \tag{25}$$

it is easy to see that $\mathbf{H}$ is a tridiagonal matrix, possibly dependent on t through V, which is symmetric (i.e., Hermitian). The semi-discrete time-dependent Schrödinger equation is thus

$$i\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) = \mathbf{H}(t)\mathbf{y}(t), \tag{26}$$

$$y_j(0) = f(x_j), \quad 1 \le j \le N. \tag{27}$$

Notice that the boundary conditions are absorbed into the definition of the vector $\mathbf{y}$ and matrix $\mathbf{H}$.

Notice, that this is a coupled set of ordinary differential equations (ODEs). Standard methods in MATLAB such as `ode45` may be used (Runge-Kutta methods), but these are not satisfactory for our needs, since they do not preserve the norm of $\mathbf{y}$ etc.

4.2 Strang splitting for FDM

Notice that the FDM scheme above indeed naturally gave us a splitting of $\mathbf{H} = \mathbf{T}+\mathbf{V}$, and in fact you may be tempted to try this:

$$\exp(-it\mathbf{T}) \approx I - it\mathbf{T} - \frac{1}{2}t^2\mathbf{T} + \cdots, \tag{28}$$

and truncating after a few terms. Then you may compute $\exp(-it\mathbf{V})$ similarly, but since $\mathbf{V}$ is diagonal, you might as well compute the exact exponential. In fact, using the MATLAB function `expm`, you can compute the exponential $\exp(-it\mathbf{T})$ automatically, to machine precision. Try this:

```
% Assume that T and V are given as NxN matrices.
% Assume that t is given, e.g., t = 0.01.
% Let's compute the split-step propagator approximation:
U = expm(-0.5i*t*T)*expm(-1i*t*V)*expm(-0.5i*t*T);
% Now we can use U to propagate any N-vector psi:
psi2 = U*psi; % psi evolved to t!
```

4.3 Pseudospectral method, or split-step FFT

Now, let us discuss "the" split-step technique, using the fast Fourier transform and the Strang splitting propagator (16). This combination is often called the pseudospectral method, or split-step FFT.

The finite difference approximation given above is very simple, but the error in the approximation of the differential operator is of order $O(h^2)$. Using the so-called pseudospectral approximation, we increase this to $O(h^N)$! The spatial approximation is almost the same, but we approximate $\partial^2/\partial x^2$ using a discrete Fourier transform (DFT, not to be confused with density functional theory!), commonly implemented using fast Fourier transform techniques (FFT).

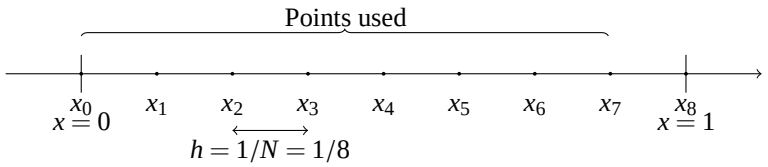

Figure 2: Illustration of a simple pseudospectral grid with $N = 8$ points (excluding the rightmost boundary), dividing the interval $[0,1]$ into $N = 8$ equal sub-intervals of length $h = 1/N$

I will not go into much detail about the DFT and the FFT algorithm – take a look at MATLAB's documentation for FFT or perhaps the section on FFT in "Numerical Recipes in C", available for free on the Internet. You must be aware of certain differences in *convention* when using the FFT, regarding normalization factors, ordering of the transformed vector components, phase shifts, and so on.

If you have difficulties using MATLAB's FFT routine to do sensible calculations, you can have a look at my simple code, `pseudospectral.m`. I strongly recommend that you fully understand this method, though. I confess that there are some points still unclear to me as well, so there might be some small errors or inaccuracies in this exposition. The program *works correctly* though.

The spatial discretization must be modified somewhat, to incorporate *periodic boundary conditions*. The periodic boundary condition is an essential feature: the DFT is only valid for periodic functions. Thus you will get a somewhat artificial behaviour, as the one-dimensional particle suddenly lives on a circle (why?) rather than in an infinite well, and in the two-dimensional case, it would live on a topological torus, like Pacman!

We now create a uniform grid with $N+1$ grid points in total – that amounts to $N-1$ interior points, see Figure 2. However, due to the periodic boundary condition we *are* interested in $\psi(0,t)$, so that when we include the left boundary point x_0, we get a total of N relevant grid points, i.e., N degrees of freedom. Be aware of this difference in convention (at least in my codes and these notes ...)

The expression for the grid points x_j now becomes

$$x_j = hj = \frac{1}{N} j, \quad 0 \leq j \leq N-1. \tag{29}$$

Notice that j starts at 0 instead of 1 as in the FDM. Moreover, it ends at $j = N-1$, totalling N points. We do not use x_{N+1}, due to the identification.

The periodic boundary condition leads to the identification $y_{N+1}(t) = y_0(t)$, so y_{N+1} drops from the equations. In the FDM scheme, this would lead to extra matrix elements (can you see that?) in the "corners" of the matrix $\mathbf{T}$.

The DFT is based on a *change of basis* to the functions $\phi_k(x) = e^{i2\pi kx}$, which forms a basis for the space $L_p^2[0,1]$, i.e., periodic square integrable functions. (I deliberatedly avoid the ket-notation here, as it is not standard in this context.) When we write out an expansion of $\psi(x) \in L_p^2[0,1]$ in this basis, we recognize it as the standard Fourier series over $[0,1]$:

$$\psi(x) \equiv \sum_{k=-\infty}^{\infty} Y_k e^{i2\pi kx}, \tag{30}$$

where

$$Y_k = \int_0^1 e^{-i2\pi kx} \psi(x) \mathrm{d}x. \tag{31}$$

The DFT simply approximates this integral with a *sum*, viz,

$$Y_k \approx \frac{1}{N}\sum_{n=0}^{N-1} e^{-i2\pi k x_n}\psi(x_n), \tag{32}$$

where the prefactor $1/N$ is the spacing between each grid point x_n, i.e., the sum is a Riemann sum approximation to the integral.

We furthermore truncate the Fourier series using $0 \leq k < N$, which might seem unsymmetrical, but it turns out it doesn't matter if we include negative k or not, as long as we take N of them (in sequence)! This is because Y_k turns out to be periodic.

In the DFT, the function $\psi(x)$ is approximated by its values at x_n, i.e., we define

$$y_k \equiv \psi(x_k), \quad 0 \leq k < N. \tag{33}$$

We now have a pair of discrete functions y_k and Y_k, the latter being the Fourier transform of the former. In my opinion, the factor $1/N$ makes the whole think unsymmetric. We can normalize, and we now have the pair of transforms

$$y_n = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{i2\pi kn/N} Y_k, \quad Y_k = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} e^{-i2\pi kn/N} y_n \tag{34}$$

In terms of matrices,

$$\mathbf{y} = \mathbf{Z}^\dagger \mathbf{Y}, \quad \mathbf{Y} = \mathbf{Z}\mathbf{y}, \quad Z_{kn} = \exp(-2\pi ikn/N)/\sqrt{N}. \tag{35}$$

Notice, that since $\psi(x)$ is periodic, we could have "shifted" the grid or function, and essentially gotten the same result for Y_k. The obvious symmetry in y_k and Y_k shows that we could "shift" the "frequency grid" as well, i.e., the allowed values of k, and still gotten the same result. It is indeed easy to see that adding N to k leaves x_n invariant. This is the reason for selecting just positive values of k. Note, however, that for $k > N/2$, the *actual* frequency is $k - N$. This is a matter of convention. In my code, I adhere to this convention, which is the same as MATLAB's.

Notice, that

$$\frac{\partial}{\partial x}\phi_k(x) = 2\pi ik\phi_k(x), \tag{36}$$

i.e., that the continuous basis functions are eigenfunctions for the differential operator. So, transforming $\psi(x)$ to this basis would give us a natural "place" for differentiating! Indeed, if we allowed the *complete* Fourier series ($N = \infty$), we would have

$$\frac{\partial}{\partial x}\psi(x) = 2\pi i Z^{-1} K Z \psi(x),, \tag{37}$$

where K is a diagonal operator: $K\phi_k = k\phi_k$, simple as that, and where Z denotes the transformation of $\psi(x)$ to a Fourier series; a unitary transformation. Notice then, that

$$-\frac{1}{2}\frac{\partial^2}{\partial x^2}\psi(x) = 2\pi^2 Z^{-1} K^2 Z \psi(x),, \tag{38}$$

giving the *exact* kinetic energy operator in the frequency domain.

The truncation leads to the expression

$$\frac{\partial}{\partial x}\psi(x) \mapsto \mathbf{T}\mathbf{y} = 2\pi^2 \mathbf{Z}^{-1}\mathbf{K}\mathbf{Z}\mathbf{y}, \tag{39}$$

where $\mathbf{K}$ is a diagonal matrix. However, due to the convention of shifting $k < 0$ by N, the diagonal elements are $K_{kk} = k - N$ for $k > N/2$, and $K_{kk} = k$ otherwise.

Note that

$$\exp(-it\mathbf{T}) = \mathbf{Z}^{-1}\exp(-2i\pi^2 t\mathbf{K}^2)\mathbf{Z} \tag{40}$$

You *could* compute the matrix $\mathbf{Z}$ yourself easily and then compute the $\mathbf{T} = \mathbf{Z}^{-1}\mathbf{K}\mathbf{Z}$ in (39) using it. However, contrary to the finite-difference approximation, $\mathbf{T}$ is now *dense*, so that this would take $O(N^2)$ operations – it takes a lot of computer time! Using an implementation of FFT would be wiser, since it computes $\mathbf{Y} = \mathbf{Z}\mathbf{y}$ in $O(N\ln N)$ operations. Using MATLAB, the application of, e.g., $\mathbf{T}$ would look someething like

```
Upsi = ifft(D.*fft(psi)); % D = vector related to k^2, but not
                          % entirely simple to show here.
```

Notice that there is no explicit matrix in this expression – it is the `fft` and `ifft` that actually computes *action* of the DFT matrix $\mathbf{Z}$.

On the other hand, in order to *learn* the method and how it works, a brute force approach can be wise.

A final note: Using the FFT, one is constrained to use $N = 2^p$ points due to efficiency. This is due to the fact that the FFT works by subdividing N points into two equal sets, recursively until only single-element sets are left.

4.4 Generalizing to two or more dimensions

The key to generalizing both the FDM and FFT method to two or higher dimensions is to observe that

$$T = T_x + T_y \tag{41}$$

in the two-dimensional case, and similarly in three dimensions. We'll stick to two dimensions for simplicity.

The two-dimensional FDM grid is defined by a *tensor product* of the one-dimensional grid:

$$\mathbf{r}_{jk} = (x_j, y_k) = \frac{1}{N+1}(j,k), \quad 1 \le j,k \le N. \tag{42}$$

This is N^2 points, and we approximate $\psi(x_j, y_k) \approx c_{jk}$. The operator T_x is approximated by a finite difference in the x-direction, and similarly in the y-direction.

Notice that our vector $\mathbf{c}$ has become a matrix, since we need two indices j and k. This is often confusing the first time one meets it, because now $\mathbf{T}$ cannot be a matrix! Rather, it is a tensor, mapping matrices into matrices. However, there is no difficulty in principle to map the indices (j,k) onto a single index α, e.g.,

$$\alpha(j,k) = N(j-1) + k. \tag{43}$$

Clearly, $1 \le \alpha \le N^2$, as it must be. It is not too difficult to show, that our discrete Hamiltonian now can be written as a matrix $\mathbf{H}$ acting on $\mathbf{c}$ when we introduce the new indexing scheme.

For the pseudo-spectral method, things are similar: Assume that $\mathbf{c}$, now as a matrix, is organized such that the rows correspond to "slices" in the x-direction. To compute the action of the DFT version of T_x, we must apply the DFT to all rows in succession, then compute the action of the diagonal operator before transforming back. The process is similar for T_y.

In MATLAB, this can rather easily be achieved by using the `fft` function, which can work on matrices as well as vectors. For example, `fft(psi,N,1)` will compute the FFT along the x-direction (rows), while `fft(psi,N,2)` will compute along the y-direction (columns). In fact, MATLAB also handles three-dimensional arrays similarly.

In the mentioned code example that I have created, this method is used.

4.5 Arbitrary orthonormal basis

One may also use a completely arbitrary orthonormal basis set $|\phi_k\rangle$, $k = 1, \cdots, \dim(\mathscr{H})$ to define a semi-discrete formulation. In order for this to be meaningful, we need some idea of the "correctness" of this basis, i.e., that it will capture "most" of the exact wave function when we truncate it after say N basis functions.

Let us first elaborate on the formal consequences of having a truncated basis.

The truncated basis set defines an N-dimensional subspace $\mathscr{K} \subset \mathscr{H}$. This subspace can be viewed as $\mathbb{C}^N$ by virtue of the basis functions: for any $|\psi\rangle \in \mathscr{K}$, which is an *abstract vector*, the expansion in the basis vectors define a coefficient vector $\mathbf{c} \in \mathbb{C}^N$, viz,

$$|\psi\rangle = \sum_{k=1}^{N} c_k |\phi_k\rangle, \quad c_k \equiv \langle\psi|\phi_k\rangle. \tag{44}$$

Conversely, any $\mathbf{c} \in \mathbb{C}^N$ defines a $|\psi\rangle \in \mathscr{K}$ by the same correspondence.

It follows, that *any* linear operator on $\mathscr{K}$ can be viewed as a matrix operating on the coefficients. The matrix elements are given by $A_{jk} = \langle\phi_j|A|\phi_k\rangle$, so that

$$A|\psi\rangle \mapsto \mathbf{A}\mathbf{c}. \tag{45}$$

Notice, that if $A = A^\dagger$, then $\mathbf{A} = \mathbf{A}^\dagger$ as well, so that the discrete version of the Hamiltonian is still a Hamiltonian! The time-dependent Schrödinger equation may be written

$$i\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{c}(t) = \mathbf{H}(t)\mathbf{c}(t). \tag{46}$$

The usual machinery of linear algebra may now be used on the finite-dimensional space $\mathscr{K}$ and the discrete Hamiltonian $\mathbf{H}$. In particular, we may diagonalize the Hamiltonian, viz,

$$\mathbf{H} = \mathbf{U}\mathbf{E}\mathbf{U}^\dagger, \tag{47}$$

where $\mathbf{U}$ is unitary and E is diagonal. The *columns* of $\mathbf{U}$ are the eigenvectors $\mathbf{u}_k$ of $\mathbf{H}$, and $E_{kk} \equiv E_k$ is the corresponding eigenvalue.

Let us compute the propagator $\mathbf{U}(t)$ of such a time-independent Hamiltonian:

$$\begin{aligned}\exp(-it\mathbf{H}) &= \sum_{n=0}^{\infty}(-it)^n\mathbf{H}^n = \sum_{n=0}^{\infty}(-it)^n(\mathbf{U}\mathbf{H}\mathbf{U}^\dagger)^n = \mathbf{U}\left(\sum_{n=0}^{\infty}(-it)^n\mathbf{E}^n\right)\mathbf{U}^\dagger \\ &= \mathbf{U}\exp(-it\mathbf{E})\mathbf{U}^\dagger,\end{aligned} \tag{48}$$

where $\exp(-it\mathbf{E})$ is seen to be diagonal, with diagonal elements $\exp(-itE_k)$. This trick, using the eigenvector decomposition of $\mathbf{H}$ to compute an infinite series, is quite universal and useful.

Now, suppose that we *change basis* in $\mathscr{K}$, using the eigenvectors $|\psi_k\rangle \mapsto \mathbf{u}_k$ as basis. Any arbitrary $|\psi\rangle \in \mathscr{K}$ (or $\mathbf{c} \in \mathbb{C}^N$) can be written in terms of $|\psi_k\rangle$ (or $\mathbf{u}_k$), viz,

$$|\psi\rangle = \sum_{k=1}^{N} c_k|\phi_k\rangle = \sum_{k=1}^{N} d_k|\psi_k\rangle. \tag{49}$$

Now, we must have

$$\mathbf{d} = \mathbf{U}^\dagger\mathbf{c}, \tag{50}$$

i.e., $\mathbf{U}^\dagger$ transforms from the original basis ($|\phi_k\rangle$) *to the new eigenvector basis* ($|\psi_k\rangle$). Now, multiplying Eqn. (46) on the left with $\mathbf{U}^\dagger$, we obtain that the the time-dependent Schrödinger equation is diagonal in the energy eigenvector basis, viz,

$$i\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{d}(t) = \mathbf{E}\mathbf{d}(t). \tag{51}$$

So, we simply get

$$d_k(t) = \exp(-itE_k)d_k(0). \tag{52}$$

Looking back, I see that I have written most of this earlier, but that's okay.

4.6 A time-dependent perturbation and a quantum dot system

Let us return to the question of the "correctness" of the basis. Suppose that $H = H_0 + H_1$ where H_1 *may* be a time-dependent perturbation; it may be independent of time as well. Suppose that it was the eigenvalue problem of H_0 that we solved, perhaps even exactly as in Section 2. This would be the case for quantum dots, where H_0 is a (many-body) harmonic oscillator, and where H_1 is the different interactions between the particles (which are independent of time). Using the exact eigenvectors of H_0 as basis, we obtain

$$\mathbf{H} = \mathbf{H}_0 + \mathbf{H}_1, \quad \mathbf{H}_0 = \mathrm{diag}(\epsilon_1, \cdots \epsilon_N), \tag{53}$$

where ϵ_n is the eigenvalue corresponding to the n'th eigenfunction ϕ_n of H_0. In reality, these are indexed by many different numbers in a complicated way, much like the mapping of the index pairs in the 2D FDM grid onto a single number, but well ... more complicated. What matters is that we have chosen *finitely many* to construct our finite dimensional space $\mathscr{K}$.

The matrix $\mathbf{H}_1$ on the other hand, is defined by

$$(H_1)_{n,m} = \langle \phi_n | H_1 | \phi_m \rangle, \tag{54}$$

and is *not* diagonal. It is often in a sense "small", corresponding to the "hope" that the basis is good. For the quantum dot case, I discuss this at (great) length in my thesis.

Let us do another iteration on this idea: Suppose that we have computed the eigenfunctions of H, and not only H_0. For the quantum dot, that's what my code does. Suppose we select N of these, and suppose that we impose a time-dependent perturbation $W(t)$ on H, i.e., the actual H is given by

$$H_2 = H_0 + H_1 + W(t) = H + W(t). \tag{55}$$

Now, H is diagonal with respect to the eigenvectors produced by the code, but $W(t)$ is not – it will have a certain matrix $\mathbf{W}(t)$ with respect to this basis.

Suppose that the energies $\mathbf{E}$ are available (arranged in a diagonal matrix), as well as $\mathbf{W}(t)$. Now one may use (for example) the Blanes–Moan method, or a Runge-Kutta method, or whatever, to solve the time-dependent system according to the ODE

$$i\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{t}(t) = [\mathbf{E} + \mathbf{W}(t)]\mathbf{c}(t), \tag{56}$$

where I stress that $\mathbf{E}$ is diagonal, and the only nontrivial part of the evolution is that arising from $\mathbf{W}(t)$. I also stress, that in the BM method you would *not* perform a splitting with respect to T and V. In fact, those operators would not be available yo you directly!