Q&A Lecture 2

We now have a computable way to describe the time evolution of a certain state from time t=0 to time t=t. But would it be possible to have a reverse time evolution of the wavefunction at time t=t to come back at time t=0, and can we define a time-reversal operator?

Yes, the “backward” propagation is generated by the Hermitian conjugate time propagator…in practice, this means change t in -t in the expressions and use the same operations in the step.

You mentioned that the modulus square of the quantum propagator is an interesting quantity because it gives me the probability to find a certain state at time t, starting from time t=0. Is that correct?

Starting from a certain state at t=0. other than that, the statement is correct.

The division of the propagation into small time-steps suggests that taking the limit of an infinite number of infinitely small time-steps leads to the exact propagation of the wavefunction. Is this limit well defined ?

The limit is well-defined in the sense that the series leads to convergent results for the propagation. There is a nuance here that we’ll see later in the class when we discuss path integrals and the functional expression of the propagator.

We often see order 2 Trotter decomposition in computational quantum physics/chemistry algorithms. Why not taking higher orders and when does it become penalizing to take higher orders decompositions?

The operators appearing in higher-order expansions are more complex (if not impossible) to apply efficiently. For example, the commutator term that we neglected would require to compute the Hessian of the potential.

How might a time step be chosen for SOFT, and are there ways of validating the choice of time step without doing more expensive calculations?

If information of typical timescales of motion (e.g. vibrations) for the system are available, they can guide the choice of the time step. However, it is ALWAYS necessary to test the choice numerically by, for example, verifying that the energy of the system remains constant (not an expensive operation). This may require repeating calculations for different time steps, but I don’t know of a (clean) way to get around it.

In classical integration schemes there's some more advanced algorithms to make the error converge faster such as Verlet or Runge-Kutta methods (we reach convergence of order dt^4 if I remember well), is there something similar in quantum simulations and if not is there something intrinsic to the problem that is stopping this?

The method that we used to derive the integration algorithm (i.e., considering the operator that generates the dynamics and using it to create the updates) is very general and it can be used also in classical mechanics. This approach guarantees several important properties of the underlying dynamics, both in the classical and quantum cases. One example is the unitarity that we discussed in class, another one is the time reversal symmetry. These properties are not preserved, in classical, by predictor-corrector schemes like Gear or Runge-Kuta. So, even if they enable larger time steps without loss of stability (i.e. the simulation does not explode) that does not necessarily mean that they are more precise (in the sense to respect the properties of the dynamics).

It is possible to obtain propagators of higher order in time also in quantum, but this comes at the cost of more expensive operations (e.g. the need to compute the curvature of the potential) that are usually not adopted.

Is there a way to simulate quantum dynamics without path integral formulation?

Although the Trotter breakup is also at the basis of the path integral formulation, that's not what we did in class. So, the answer to your question is yes: an example is the SOFT algorithm that we discussed last Wednesday. Path integrals are however important in our program and we shall discuss them later.

Why are we restricting ourselves to time independent Hamiltonians, it seems like the path integral formalism is appropriate for both without much added complexity.

Time-dependent propagations - for the case of non-commuting Hamiltonians at different times - are much more difficult to manipulate in any formalism. A key problem is the need to impose time ordering, which is not easy to efficiently translate into an algorithmic procedure. Also, for most of the applications in the course (including response to external time-dependent perturbations) the time independent situation is sufficient.

A function of an operator can be expressed as a series of the operator. Are there any limitations to such expression? For example, what kind of operator can be expressed like this? Under what conditions is this series continuous and convergent?

The series inherits the convergence properties of the function that it represents. These conditions typically define a radius of convergence that is assessed based on an appropriate metric (e.g. the norm of the operators)

Up until now, we have discussed time-independent Hamiltonians. Will the class cover time-dependent Hamiltonians?

We'll discuss linear response theory which is a way to deal with the effects of a time-dependent perturbation on the system. However, there will be no in-depth discussion of explicitly time-dependent Hamiltonians.

How much more complex are these issues?

Very: time-ordering is needed for the formal solution and not easy to implement.

Can the time-dependent potential energy be separated from the position-dependent potential energy?

It depends on the form of the interactions and on the commutation relationships between the two parts.

In practice, what is the usual order of magnitude of the time-step that is taken for a medium sized chemical system (50 atoms organic molecule for example), such that the approximation described by Trotter's Theorem remains valid?

The magnitude of the time step is not determined by the size of the system but by the characteristics of the interactions. Typical choices for quantum dynamics are in the range of fractions of femtoseconds. Note also that the system size that you cite as medium (50 atoms) is in fact at the very high end of the spectrum for exact methods for the solution of the time dependent Schrödinger equation. As mentioned in class, the cost of these methods is exponential in the number of degrees of freedom. Note also, we shall say more about this in future classes, that the Trotter approximation is also key to deriving evolution algorithms for CLASSICAL time evolution.

All of this makes sense but can there be a possibility that we aren't able to find a small enough time step for our simulation? Like no matter what dt we choose, our simulation is not stable, for example, the energy, when it should be conserved, isn't.
OR will there always exist a small enough dt, such that we are able to get "good" results from our simulations or there are some systems for which this split operator method never works.

The small enough time step always exists…in principle! Whether we'll be able to use it depends on (1) finite numerical precision of the machine (imagine that the time step needed is so small that it produces changes in the variables that are smaller than the available numerical precision…); (2) available computational resources: the smaller the time steps, the larger the number of time steps (and therefore the computational time) needed to cover a given time span. For the type of molecular systems that we focus on, it is usually possible to find a good time step…but not to go to very long times. In any case, the computational cost of the SOFT algorithm is exponential with the number of degrees of freedom, so this algorithm is not really used. We introduced to give a first example of the work that we shall do and to highlight the cost to set the stage for approximate methods.

Could you suggest a reference for understanding the equivalence between the density matrix evolution and the Schrodinger equation?

There is a chapter that I like a lot on this on the book by Cohen Tannoudji on quantum mechanics.

It was shown that different expansions can give errors of different orders in δt.
How do we pick a particular expansion, since we are limited to finite computational resources?

Best compromise between complexity of the calculation and accuracy. The two Trotter break ups that you have seen have the same numerical complexity in that they only require the kinetic energy and potential (and not the derivatives of the potential). In this case, no doubt: go with the symmetrical splitting — I used the other one in class because it was faster for the first introduction of the splitting.

In particular, is there any relation between the polynomial order of the potential and the error which can be exploited to decide the decomposition?

The error in Trotter as we have seen it is proportional to the commutator between P^2 e V(X). It can be shown that this commutator (expressed in the coordinate basis) is related to the second derivative of the potential with respect to the coordinates. There are other splittings that introduce commutators of commutators in the leading order of the error and these can in turn be related to higher order derivatives of the potential. If the potential is polynomial, these derivatives will be zero for sufficiently high number of "commutators". However, the most frequent case is that potentials depend on inverse powers of the coordinates so the argument is not super useful.

During the course we mentioned the quantum propagator (which is approximated with the Trotter theorem), what is his physical meaning?

The abstract quantum propagator generates the time evolution of any abstract state. In a given basis (e.g. the matrix element in the x basis indicated as K(x_o,x_t,t) in class), when squared, gives the probability to observe the state in the initial basis element (x_o) at the initial time of the propagation and to observe the state in the final basis element (x_t) after a time t evolution.

During the lecture, professor used the $e^{(-i \hat{H} t / \hbar)} \approx e^{(-i \hat{p}^2 t / (2 m \hbar))} \times e^{(-i V(\hat{x}) t / \hbar)}$ *for* $\langle xt|e^{(-i \hat{H} t / \hbar)}|x0\rangle$*. Then later in the exercise lesson, Edrick explained SOFT, in there , we separate the evolution operator differently, so how should I decide to use which one?*

The two splits differ in the order of the error but require operations of the same numerical cost. In this case, always choose the form with the smaller error, i.e. the one presented by Edrick. I chose the other one in class because (as it requires one less splitting) it was easier to discuss and manipulate to explain the core idea of the approximation.

Why do we choose to stop the approximations after the linear term, as higher order terms would allow for a more precise approximation?

Terms of higher order contain operators (typically derivatives of the potential) that are either costly to evaluate or to compute.

Can you explain the role/meaning of the quantum propagator K? Is K relevant in the derivations we made?

K is the coordinate representation of the quantum propagator. It's square give the probability to find the system at x_0 at time zero and at x_t at time t. K is a good starting point to derive the SOFT algorithm.

Are there any situations where the Trotter approximation fails in simulating quantum time evolution?

For the type of Hamiltonians that we are considering (sum of independent functions of positions and momenta operators) and time-independent, Trotter is general. It can happen that the tilmestep needed for the approximation too hold is too small to be numerically viable but this is a practical, not formal problem.

Trotter theorem and suzuki-trotter factorization are different? If so, how is the factorization obtained?

They are not (different names for factorisations of different order, but same general idea and procedure for derivations).

We can check our error and we can always take a smaller step size to reduce the error due to our first-order approximation. However, we want to make our timestep as big as possible, so that we can simulate longer time spans. How do we get a good initial guess for this maximal timestep and how do we keep an optimal step size?

If we have information on typical timescales of motion (e.g. vibrational frequencies) we can use them to guide our initial guess for the time-step. However, the time-step must always be checked as appropriate by testing it. A key criterion is energy conservation that must be observed (within a preset numerical threshold, say 10^-5). If we don't have info on physical timescales in the system, we proceed by trial and refinement (going up if we start with a tilmestep small enough to conserve the energy better than required, going down in the opposite case). Typically, the timestep is set once at the beginning of the calculations.

I'm not sure about the relation between the "basis sets" used in DFT (e.g. 6-31G) and the basis functions we discuss in quantum mechanics lectures. Is it discussed later in the course?

The concept of basis set is very general. It is any set of vectors or functions that can be used to represent fully other vectors. Many basis exist and are chosen for a specific problem because they are more adapted to it. In class, we used basis to represent the state of a quantum system. As you say, the notion is used also in DFT and we shall say something (quickly) in future lectures.

Is the split-operator method viable for multiparticle systems with interactions? It seems to me that in this case the interactions might lead to error worse than O(Δt2) in practice.

Yes, the method is viable - with same time discretisation error - also for multi-particle systems. The derivation proceeds exactly as we did in class, with a basis now given by a collection of all coordinates or momenta of the system (tensor product basis).

If the expression of the wave function is abstract, how does taking its projection onto the position space and transforming it to an integrable form make it possible for the computer to manipulate it? Why can't a computer manipulate an abstract expression, and what changes when we convert it into a function?

Computers manipulate on-off like signals (bit) that can be used to map numbers (not abstract things). The wave function is not abstract (see above) but a function of x. The variable x (discretised) is mapped into a set of numbers (its values on a grid). The function evaluated on a grid point (and eventually on all of them) is a (a set of) numbers. These can be manipulated by the computer.