Propagation Methods

Andreas Markmann

22nd October 2003

1 Introduction

Propagation of wavepackets in time means application of the operator

$$\widehat{U} = \widehat{T} e^{-\frac{i}{\hbar}\int_{t_s}^{t_e} \widehat{H}(t)\mathrm{d}t}$$

to a given wavefunction $\psi(t = t_a = 0)$ to obtain the state of the system at a later time. Trying to simulate the time evolution of a wavepacket, several problems arise:

- representation of the wavefunction on the computer,
- evaluation of the operator $\widehat{H}$,
- approximation of the exponential of an operator.

This article is meant to introduce and classify a few typical techniques which solve these problems and enable the reader to choose or derive the technique which is best suited to his or her problem.

2 Evaluation of $\widehat{H}\psi$ [1]

2.1 Local and semilocal representation of Laplace–Δ

The finite differencing scheme (FD) considers the derivative only by a local approximation such that in one dimension

$$\frac{d^2\psi(x_i)}{dx^2} = \frac{\psi(x_{i+1}) - 2\psi(x_i) + \psi(x_{i-1})}{(\Delta x)^2}.$$

Higher order differencing schemes involving more interpolation points x_j are called semilocal. All of these approximations, however, are not very well–suited for application with quantum–mechanical problems as the consideration of wavefunctions is intrinsically global.

2.2 Global representation of Δ and Fast Fourier Transform

A really global consideration can only be achieved by either involving all available interpolations points x_j in the semilocal approach or, which is the same, calculate the derivative by multiplying the wave vectors in the Fourier transformed representation of the wavefunction. This means that the momentum operator is local in momentum space:

$$\begin{aligned}
\psi(k) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi(x) e^{-ikx} dx = FT[\psi(x)], \\
\psi(x) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi(k) e^{ikx} dk = FT^{-1}[\psi(k)], \\
\frac{d\psi(x)}{dx} &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi(k)(ik) e^{ikx} dk = FT^{-1}[(ik\psi(k)], \\
\frac{d^2\psi(x)}{dx^2} &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi(k)(-k^2) e^{ikx} dk = FT^{-1}[(-k^2)\psi(k)].
\end{aligned}$$

The Fourier representation yields the exact energy spectrum

$$T_{FT}(k) = \frac{\hbar^2 k^2}{2m},$$

while the finite difference approach gives

$$T_{FD}(k) = \frac{\hbar^2}{2m} \left[\frac{2\sin(k\Delta x/2)}{\Delta x} \right]$$

which converges to the exact spectrum for very small Δx. So for an application of the finite difference scheme one has to choose a finer mesh for the representation of the wavefunction.

2.3 Discrete Variable Representation (DVR) [1], [5]

A wavefunction ψ is represented as a linear combination of given interpolation functions ϕ_n by

$$\psi(x) = \sum_{n=0}^{N-1} a_n \phi_n(x) \tag{1}$$

such that for given interpolation points $x_i = x_0, \ldots, x_N$

$$\psi(x_i) = \sum_{n=0}^{N-1} a_n \phi_n(x_i).$$

If the ϕ_n satisfy the orthonormality relation in the arguments

$$\sum_{n=0}^{N-1} \phi_n^*(x_i)\phi_n(x_j) = \delta_{ij} \tag{2}$$

then this interpolation expression, i.e. the matrix $\phi_n(x_j)$ can be directly inverted to give

$$a_n = \sum_{j=0}^{N-1} \psi(x_j)\phi_n^*(x_j) \tag{3}$$

for then

$$\psi(x_i) = \sum_{n=0}^{N-1}\sum_{j=0}^{N-1} \psi(x_j)\phi_n^*(x_j)\phi_n(x_i) = \sum_{j=0}^{N-1} \psi(x_j)\delta_{ij}.$$

If the ϕ_n furthermore satisfy the orthonormality relation in their order

$$\langle\phi_n|\phi_m\rangle = \int \phi_n^*(x)\phi_m(x)dx = \delta_{mn} \tag{4}$$

or, more generally,

$$\int w(x)\phi_n^*(x)\phi_m(x)dx = \delta_{mn}$$

then, choosing the sampling points x_j as the zeros of the N–th polynomial ϕ_N, it follows from Gaussian integration theory that their discrete representations also fulfill the orthonormality relation

$$\sum_{j=0}^{N-1} w_j\phi_n^*(x_j)\phi_m(x_j) = \delta_{mn} \tag{5}$$

with constant point weights w_j because their degrees are all smaller than N (see appendix 4.1). The coefficients become

$$a_n = \sum_{j=0}^{N-1} w_j\psi(x_j)\phi_n^*(x_j).$$

In this case, the scalar product formula simplifies to

$$\langle\psi|\chi\rangle = \sum_n \frac{1}{w_n} a_n^* b_n = \sum_{j=0}^{N-1} \frac{1}{w_j}\psi^*(x_j)\chi(x_j)$$

with b_j the coefficients of the representation of χ.

Substitution of the coefficients (3) in the representation (1) results in

$$\begin{aligned}\psi(x) &= \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} w_j\phi_i^*(x_j)\psi(x_j)\phi_i(x) \\ &= \sum_{j=0}^{N-1} \psi_j\xi_j(x)\end{aligned}$$

with orthonormal coordinate eigenfunctions $\xi_{j(x)}$ and the sampling values ψ_j:

$$\begin{aligned} \xi_j(x) &= \sqrt{w_j} \sum_{i=0}^{N-1} \phi_i^*(x_j)\phi_i(x), \\ \psi_j &= \sqrt{w_j}\psi(x_j). \end{aligned}$$

Then the derivative operators can be written as

$$\begin{aligned} \frac{d^n\psi(x)}{dx^n} &= \sum_{j=0}^{N-1} \psi_j \frac{d^n\xi_j(x)}{dx^n}, \\ \frac{d^n\xi_j(x)}{dx^n} &= \sqrt{w_j} \sum_{i=0}^{N-1} \phi_i^*(x_j) \frac{d^n\phi_i(x)}{dx^n}. \end{aligned}$$

2.4 Comparison

In both the DVR and the discrete representation a discrete Hilbert space is employed. In the Fourier representation a uniform grid spacing is used to obtain a finite Fourier series. The quadrature weight is the grid spacing (trapezoidal rule). This quadrature is of comparable accuracy to the Gaussian quadrature employed in the DVR representation.

The Fourier method scales as $N \log N$ as opposed to an N^2 scaling of the DVR method. The DVR, however, provides the opportunity of choosing the basis functions ϕ_j according to the physical system so their number can be kept rather small (e.g. LCAO).

3 Propagation methods [2]

The problem is to calculate the propagation operator

$$\widehat{U}(t_e, t_s) = \widehat{T} e^{-\frac{i}{\hbar}\int_{t_s}^{t_e} \widehat{H}(t')dt'}$$

with the time order operator $\widehat{T}$ which solves the Schroedinger equation. If $\widehat{H}$ is time dependent, one can extrapololate by dividing the considered time period into smaller sections over which $\widehat{H}$ is assumed to be constant. Then the propagation operator assumes the form

$$\widehat{U}(t_e, t_s) = e^{-\frac{i}{\hbar}\widehat{H}_s(t_e - t_s)}$$

with different $\widehat{H}_s$ dependent on t_s. This extrapolation makes use of the relation $\widehat{U}(t_1 + t_2, t_0) = \widehat{U}(t_2, t_1)\widehat{U}(t_1, t_0)$.

A function f applied to an operator $\widehat{O}$ can be spectrally decomposed to give

$$f\left(\widehat{O}\right) = \sum_n f(\lambda_n)\widehat{P}_n,$$

where λ_n are the eigenvalues and $\widehat{P}_n = |u_n\rangle \langle u_n|$ are the projection operators to the eigenspaces with

$$\widehat{P}_n \widehat{P}_m = \delta_{mn} \widehat{P}_n.$$

If the eigenvalues are known, these projection operators can be formed by successively applying operators of the form $\widehat{Q}_i = \left(\widehat{O} - \lambda_i \widehat{I}\right)$, where $\widehat{I}$ denotes unity:

$$\widehat{P}_n = \frac{1}{\aleph} \widehat{Q}_N \cdots \widehat{Q}_{n+1} \widehat{Q}_{n-1} \cdots \widehat{Q}_1$$

with the normalization factor

$$\aleph = (\lambda_n - \lambda_N) \cdots (\lambda_n - \lambda_{n+1}) (\lambda_n - \lambda_{n-1}) \cdots (\lambda_n - \lambda_1).$$

However, solving the eigenvalue problem is computationally prohibitively expensive $\left(O\left(N^2\right)\right)$ and the number of eigenvalues may be infinite, so approximate methods have to be found to confine the problem to finding a small subset of the set of eigenvalues. Applying the above formulae directly leads to the Lagrangian interpolation of the operator (as the operators $\widehat{Q}_i$ are evaluated recursively) which has numerical disadvantages because adding another sampling point results in the necessity of calculating a completely new interpolation polynomial.

3.1 Newtonian Interpolation

It is therefore better to approximate the function f polynomially by the Newtonian term

$$f(z) = a_0 + a_1(z - x_0) + a_2(z - x_1)(z - x_0) + a_3(z - x_2)(z - x_1)(z - x_0) + \cdots$$

with the coefficients a_i being the divided differences

$$\begin{aligned}
a_0 = f[x_0] &= f(x_0), \\
a_1 = f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0}, \\
a_k = f[x_0, x_1, \ldots, x_k] &= \frac{f(x_k) - a_0 - \sum_{i=1}^{k-1} a_l (x_k - x_0) \cdots (x_k - x_{l-1})}{(x_k - x_0) \cdots (x_k - x_{k-1})} \\
&= \frac{f[x_0, x_1, \ldots, x_{k-1}] - f[x_1, x_2, \ldots, x_k]}{x_0 - x_k}.
\end{aligned}$$

which are approximations of the i–th derivative of f at x_0.

In the Newtonian interpolation, to add another interpolation point it is only

necessary to calculate the bottom line of the following scheme:

$$\begin{array}{llllllll}
f(x_0) & \ddots & & & & & & \\
f(x_1) & \cdots & f[x_0,x_1] & & & & & \\
\vdots & & \vdots & & \ddots & & & \\
f(x_{N-1}) & \cdots & f[x_{N-2},x_{N-1}] & \cdots & f[x_0,\ldots,x_{N-1}] & \ddots & & \\
f(x_N) & \cdots & f[x_{N-1},x_N] & \cdots & f[x_1,\ldots,x_N] & \cdots & f[x_0,\ldots,x_N]. &
\end{array}$$

In practice, this process is continued until self–consistency. Numerical problems with a small distance between the sampling points can be overcome by permuting these sampling points. A polynomial expansion is also possible for the consideration of non–hermitian operators thats eigenvalues do not lie on a real interval but inside a disk in the complex plane [2].

3.2 Error Analysis

The error of the truncated Newton series is

$$\varepsilon = \left\| f\left[x_0,\ldots,x_{N-1},\widehat{O}\right] \prod_{j=0}^{N-1} \widehat{Q}_j \right\|.$$

The divided difference term in this error cannot be evaluated by repeated application of the operator $\widehat{O}$ as is desired for computational implementation. So the term to minimise is the product term which only depends on the interpolation points x_i. If the number of these points is equal to the number of eigenstates of the operator, this minimization is equivalent to the diagonalization of the operator so the error will converge to zero as this number rises.

There are two approaches to the minimization:

1. **Uniform Approach:** The error is minimised as an operator – with respect to the whole Hilbert space of states.
2. **Non–Uniform Approach:** The norm of the product term applied to a certain wave function is minimised (which is usually the initial state).

3.3 Overview

There is a variety of different propagation schemes for quantum mechanical systems, with different properties. See table 1 for an overview of the methods considered in this article.

3.4 The Second Order Differencing scheme (SOD) [3]

The most natural ansatz to approximating the propagation operator

$$\widehat{U} = e^{-\frac{i}{\hbar}\widehat{H}\Delta t} = 1 - \frac{i}{\hbar}\widehat{H}\Delta t + \cdots$$

Table 1: overview of methods.

method	**remarks**
Second–order differencing (SOD)	simple; interaction between potentials; theoretical considerations
Split operator (SPO)	simple, stable; but no space–momentum mixed terms in $\widehat{H}$
Chebyshev	global, long time method; exact to computer accuracy
Lanczos	subspace depends on initial state

would be the straightforward expansion into a Taylor series. This has, however, proven to be numerically unstable because of its asymmetry with respect to time inversion. Therefore the method has to be symmetrised by considering one forward and one backward step to first order in the Taylor expansion:

$$\begin{aligned} \psi(t+\Delta t) - \psi(t-\Delta t) &= \left(e^{-\frac{i}{\hbar}\widehat{H}\Delta t} - e^{\frac{i}{\hbar}\widehat{H}\Delta t}\right)\psi(t), \\ \rightarrow \psi(t+\Delta t) &\approx \psi(t-\Delta t) - 2\frac{i}{\hbar}\Delta t\widehat{H}\psi(t). \end{aligned} \tag{6}$$

There are two ways of obtaining the second wavefunction $\psi(\Delta t)$ from $\psi(0)$:

1. (SOD) Propagate by a first order scheme for half a time–step and from there propagate with SOD for another half time–step.

2. (SODS) Propagate with SOD half a step forward to get $\psi\left(\frac{\Delta t}{2}\right)$ and half a step backward to get $\psi\left(-\frac{\Delta t}{2}\right)$. The final result is the arithmetic mean value of $\psi\left(t+\frac{\Delta t}{2}\right)$ and $\psi\left(t-\frac{\Delta t}{2}\right)$. This method is more symmetric.

3. $$\begin{array}{ccc} \psi(0) & \rightarrow (1st\,order) \rightarrow & \psi\left(\frac{\Delta t}{2}\right), \\ \psi(0),\, \psi\left(\frac{\Delta t}{2}\right) & \rightarrow (SOD) \rightarrow & \psi(0),\, \psi\left(-\frac{\Delta t}{2}\right), \\ \psi(0),\, \psi\left(-\frac{\Delta t}{2}\right) & \rightarrow (SOD) \rightarrow & \psi(0),\, \psi\left(\frac{\Delta t}{2}\right). \end{array}$$
   this initialization stops there because the lower two steps become cyclic because of the time–reversability of the SOD method.

The method is unitary and conserves norm and energy. It can also be very easily and naturally adjusted to involve interactions between potential surfaces *a* and *b* due to the electromagnetic field *E(t)* and the (approximately constant) magnetic dipole moment μ: [7]

$$\begin{aligned} \psi_a(t+\Delta t) &= \psi_a(t-\Delta t) - 2\frac{i}{\hbar}\Delta t\widehat{H}_a\psi_a(t) - 2\frac{i}{\hbar}\Delta t\mu E(t)\psi_b(t), \\ \psi_b(t+\Delta t) &= \psi_b(t-\Delta t) - 2\frac{i}{\hbar}\Delta t\widehat{H}_b\psi_b(t) - 2\frac{i}{\hbar}\Delta t\mu E(t)\psi_a(t). \end{aligned}$$

3.5 Analysis of the SOD method

Apply formula (6) twice to obtain the top row of the matrix in the propagation equation

$$\begin{pmatrix} \psi^{n+1} \\ \psi^{n} \end{pmatrix} = \begin{pmatrix} 1-4\frac{\Delta t^2}{\hbar^2}\widehat{H}^2 & -2\frac{i}{\hbar}\widehat{H}\Delta t \\ -2\frac{i}{\hbar}\Delta t\widehat{H} & 1 \end{pmatrix} \begin{pmatrix} \psi^{n-1} \\ \psi^{n-2} \end{pmatrix}.$$

The eigenvalues of this propagation matrix are

$$\begin{aligned} \lambda_{1/2} &= 1-2\frac{\Delta t^2}{\hbar^2}\widehat{H} \pm 2\frac{\Delta t}{\hbar}\widehat{H}\sqrt{\frac{\Delta t^2}{\hbar^2}\widehat{H}^2-1} \\ &= 1-2\frac{\Delta t^2}{\hbar^2}\widehat{H} \pm 2\frac{\Delta t}{\hbar}\widehat{H}\left(i-\frac{i}{2}\frac{\Delta t^2}{\hbar^2}\widehat{H}^2-\frac{3}{4!}\frac{\Delta t^4}{\hbar^4}\widehat{H}^4+\cdots\right) \end{aligned} \tag{7}$$

The second expression stems from the Taylor–series of the squareroot at the expansion point -1.

The determinant of the propagation matrix has to be unity to make the map area unitary: $\lambda_1\lambda_2 = 1$. The mapping is stable only if the eigenvalues lie on the complex unit circle for otherwise $\lambda_1 > 1$ and the method diverges. So the radicant has to be negative, hence:

$$\Delta t < \frac{\hbar}{E_{max}}.$$

The eigenvalue of the exact operator is

$$\lambda_{exact} = e^{-2\frac{i}{\hbar}E_m\Delta t} = 1-2i\frac{\Delta t}{\hbar}E_m - 2\frac{\Delta t^2}{\hbar^2}E_m^2 - 4i\frac{\Delta t^3}{3\hbar^3}E_m^3+\cdots$$

Comparing with (7) gives the error per time–step

$$\varepsilon = \frac{(\Delta t E_m)^3}{3\hbar^3}.$$

In practice, a time–step which is safely smaller than the optimal one is chosen, usually $\Delta t = \frac{\Delta t_{opt}}{5}$ because this yields a high accuracy even after a large number K of recursion steps:

$$\varepsilon \approx \frac{K}{375}.$$

Now, $\langle\psi(t)|(6)\rangle$ is

$$\langle\psi(t)|\psi(t+\Delta t)\rangle = \langle\psi(t)|\psi(t-\Delta t)\rangle - 2\frac{i}{\hbar}\Delta t\left\langle\psi(t)|\widehat{H}|\psi(t)\right\rangle \tag{8}$$

and $\langle(6)|\psi(t)\rangle$ is

$$\langle\psi(t+\Delta t)|\psi(t)\rangle = \langle\psi(t-\Delta t)|\psi(t)\rangle + 2\frac{i}{\hbar}\Delta t\left\langle\psi(t)|\widehat{H}|\psi(t)\right\rangle. \tag{9}$$

Adding these two equations, ((8)+(9)), yields

$$Re\,\langle\psi(t+\Delta t)|\psi(t)\rangle = Re\,\langle\psi(t)|\psi(t-\Delta t)\rangle = const$$

which means norm conservation for real overlaps. Similarly, $\left\langle \widehat{H}\psi(t)\right|$can be multiplied with $|(6)\rangle$ to obtain energy conservation.

3.6 The Split–Operator method (SPO)

Approximate the operator $\widehat{U} = e^{\lambda(\widehat{T}+\widehat{V})}$, $\lambda = -\frac{i\Delta t}{\hbar}$ as

$$\begin{aligned} \widehat{U} &= S_2(\widehat{T},\widehat{V},\lambda) + S'(\lambda^3) + O(\lambda^4), \\ S_2(\widehat{T},\widehat{V},\lambda) &= e^{\frac{\lambda\widehat{T}}{2}} e^{\lambda\widehat{V}} e^{\frac{\lambda\widehat{T}}{2}}. \end{aligned}$$

The error term $S'(\lambda^3) = \frac{1}{24}\left[\widehat{T}+2\widehat{V},\left[\widehat{T},\widehat{V}\right]\right]\lambda^3$ can be easily obtained from Taylor expansion of the exponentials. As the commutator plays a role, the eigenvalues of the kinetic and potential energies have to be bounded to achieve convergence. The SPO method is very easily implemented, in fact I implemented the relaxation method (see 3.7) with SPO first before I used it as a benchmark to test and debug the more efficient but hard to implement Chebyshev method.

The SPO can be generalised to a higher order operator by

$$\widehat{U} = S_2(\widehat{T},\widehat{V},\gamma\lambda)S_2(\widehat{T},\widehat{V},(1-2\gamma)\lambda)S_2(\widehat{T},\widehat{V},\gamma\lambda)+S'\left(\left(2\gamma^3+(1-2\gamma)^3\right)\lambda^3\right)+O(\lambda^4).$$

This operator can be thought of as a split of the propagation step into several shorter time propagations, so the error terms add up. Setting $\gamma = \frac{1}{2-\sqrt[3]{2}}$ makes the S'–term zero so the new operator is of fourth order. The time–dependence of the error of the SPO method points to its main use as a *short time* propagator. Operators of any order can be built but they become computationally very expensive because of the high number of Fourier–transformations needed [9].

3.7 The Chebyshev method

The operator is expressed as an expansion in Chebyshev polynomials

$$f(z') = \frac{b_0}{2} + \sum_{n=1}^{\infty} b_n T_n(z') \tag{10}$$

with the coefficients

$$b_n = \frac{2}{\pi}\int_{-1}^{1} \frac{f(z')T_n(z')}{\sqrt{1-(z')^2}}dz'. \tag{11}$$

This inversion formula is using the orthonormality relation of type (4)

$$\frac{2}{\pi}\int_{-1}^{1} \frac{T_m(z)T_n(z)}{\sqrt{1-z^2}}dz = \begin{cases} \delta_{mn} & , m,n>0 \\ 2 & , m=n=0 \end{cases}$$

of the Chebyshev polynomials – substitute the series expansion (10) of f in (11).

In order to use the Chebyshev polynomials, the range of eigenvalues of the operator $\widehat{O}$ has to be adjusted by the substitution

$$\widehat{O'} = 2\frac{\widehat{O} - \lambda_{min}\widehat{I}}{\lambda_{max} - \lambda_{min}} - \widehat{I}$$

which results in the necessity of multiplying a factor Φ after the calculation to get the result. The images $\phi_n = T_n\left(\widehat{O'}\right)\psi$ are then built up by the recursion relation for Chebyshev polynomials:

$$\begin{aligned} \phi_0 &= \psi, \\ \phi_1 &= \widehat{O'}\psi, \\ \phi_{n+1} &= 2\widehat{O'}\phi_n - \phi_{n-1}. \end{aligned}$$

The summation of the derived vectors $b_n\phi_n$ is continued until the deviation of the coefficients from zero drops below a given accuracy. This makes it possible to achieve any desired accuracy up to computer accuracy.

The Chebyshev method is preferentially used for propagation of time–independent operators for otherwise it has to be run several times for subintervals of time over which it is *assumed to be constant*. The same has to be done if intermediate states in time are sought. Both will reduce the efficiency of the method.

Example 1: Propagation operator. The operator $f\left(\widehat{H'}\right)$ to approximate is $U = e^{-\frac{i}{\hbar}\widehat{H'}t}$ with the maximal and minimal eigenvectors $\lambda_{min} = E_{min}$, $\lambda_{max} = E_{max}$, $\Delta E = E_{max} - E_{min}$.

This means the factor $\Phi = e^{\frac{i}{\hbar}(\Delta E/2 + E_{min})t}$ is a phase factor, $f(z') = e^{-\frac{i}{\hbar}z't}$ and hence

$$b_n = (-i)^n 2J_n(\alpha),\ \alpha = \frac{\Delta E t}{2\hbar}$$

with J_n the Bessel function of first kind of order n (the proof for this expression is to be found in in appendix 4.2).[1]

The coefficients go to zero exponentially as n becomes greater than α so the method is very fast *except* for very short times where this exponentially decaying tail becomes dominant in the calculation time.

The same result is produced by

$$f(z') = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \chi_n(z')$$

with the recursion

$$\begin{aligned} \chi_0 &= \psi, \\ \chi_1 &= -i\widehat{H'}\psi, \\ \chi_{n+1} &= -2i\widehat{H'}\chi_n + \chi_{n-1} \end{aligned}$$

and the coefficients

$$a_n = 2J_n(\alpha).$$

This can be shown by proving inductively that $\chi_n = (-i)^n \phi_n$, as obviously $b_n = (-i)^n a_n$:

$$\begin{aligned} \chi_{n+1} &= -2i\widehat{H'}\chi_n + \chi_{n-1} \\ &= (-i)^{n+1} 2\widehat{H'}\phi_n + (-i)^{n-1}\phi_{n-1} \\ &= (-i)^{n+1}\left(2\widehat{H'}\phi_n - \phi_{n-1}\right) \\ &= (-i)^{n+1}\phi_{n+1}. \end{aligned}$$

Example 2: Relaxation method. The operator $f\left(\widehat{H'}\right)$ to approximate is $\widehat{U} = e^{-\widehat{H'}\tau}$ with the maximal and minimal eigenvectors $\lambda_{min} = E_{min}$, $\lambda_{max} = E_{max}$. This is often referred to as propagation in imaginary time as it results from the above propagation operator by setting

$$t = -i\hbar\tau.$$

(Note that the propagation is actually in *negative* imaginary time.)

If the operator is applied to a mixed state and the result renormalised, the state with the lowest energy in the mixed state is produced and all other states are filtered out at the rate $e^{-(E_1 - E_0)\tau}$, where E_0 and E_1 are the lowest eigenvalues. If a higher state is sought, the ground state has to be first produced and in the second run of the method projected out before every renormalization step. There are operators $e^{-(\widehat{H'}-E)^m\tau}$, m an even number, which produce the eigenstate closest to the energy E directly without the need of producing the lower states first but I shall not go into any details about these [8].

[1] These formulas were occasionally faulty in some of the papers I am citing.

The factor $\Phi = e^{(\Delta E/2+E_{min})\tau}$ for the relaxation operator $\widehat{U}$ is *not* a phase factor anymore, $f(z') = e^{-z'}$ and hence

$$b_n = (-1)^n 2I_n(\alpha),\ \alpha = \frac{\Delta E\tau}{2\hbar}$$

with I_n the *modified* Bessel function of first kind of order n.

This time, the coefficients go to zero exponentially as n becomes greater than $\sqrt{\alpha}$ so the method is very fast *except* for very short times where this exponentially decaying tail becomes dominant in the calculation time.

The same result is produced by

$$f(z') = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \chi_n(z')$$

with the recursion

$$\begin{aligned} \chi_0 &= \psi, \\ \chi_1 &= -\widehat{H'}\psi, \\ \chi_{n+1} &= -2\widehat{H'}\chi_n - \chi_{n-1} \end{aligned}$$

(note the changed sign of χ_{n-1} compared to the second algorithm in Ex. 1) and the coefficients

$$a_n = 2I_n(\alpha).$$

This can be shown by proving inductively that $\chi_n = (-1)^n \phi_n$, as obviously $b_n = (-1)^n a_n$:

$$\begin{aligned} \chi_{n+1} &= -2\widehat{H'}\chi_n - \chi_{n-1} \\ &= (-1)^{n+1} 2\widehat{H'}\phi_n - (-1)^{n-1}\phi_{n-1} \\ &= (-1)^{n+1}\left(2\widehat{H'}\phi_n - \phi_{n-1}\right) \\ &= (-1)^{n+1}\phi_{n+1}. \end{aligned}$$

3.8 The Lanczos recursion scheme

The Lanczos method is a good example for the *non–uniform* approach. The operator a function of which has to be calculated is first represented as a tridiagonal matrix on the finite cyclic subspace spanned by the vectors $\left|\widehat{O}^k\psi_0\right\rangle$, $k = 0,\ldots,n-1$, the Krylov vectors. The vector $|\psi_0\rangle$ is called the initial guess because (together with the dimension n) it uniquely defines this subspace. Usually ψ_0 is taken to be the initial state $\psi(t=0)$.

The build–up of the basis vectors for the finite dimensional representation is initialised by

$$\widehat{O}\psi_0 = \alpha_0\psi_0 \dot{+} \psi_1'$$

where $\dot{+}$ denotes the addition of two linearly independent vectors. The next step is

$$\widehat{O}\psi_1' = \beta_0'\psi_0 \dot{+} \alpha_1'\psi_1' \dot{+} \psi_2''$$

This equation can be divided by $\sqrt{\beta_0'}$, setting

$$\beta_0 = \sqrt{\beta_0'},\ \alpha_1 = \alpha_1',\ \psi_1 = \frac{\psi_1'}{\sqrt{\beta_0'}}, \psi_2' = \frac{\psi_2''}{\sqrt{\beta_0'}}$$

to make the matrix representation in the basis ψ_k symmetric and go on with the algorithm

$$\widehat{O}\psi_k' = \beta_{k-1}'\psi_{k-1} \dot{+} \alpha_k'\psi_k' \dot{+} \psi_{k+1}''$$

until the norm of the vector ψ_k becomes sufficiently small. The matrix representation of $\widehat{O}$ then will be

$$\widehat{O}\begin{pmatrix}\psi_0\\ \vdots\\ \psi_{n-1}\end{pmatrix} \approx \begin{pmatrix} \alpha_0 & \beta_0 & 0 & \cdots & 0\\ \beta_0 & \alpha_1 & \beta_1 & \ddots & \vdots\\ 0 & \beta_1 & \alpha_2 & \ddots & 0\\ \vdots & \ddots & \ddots & \ddots & \beta_{n-2}\\ 0 & \cdots & 0 & \beta_{n-2} & \alpha_{n-1}\end{pmatrix}\begin{pmatrix}\psi_0\\ \vdots\\ \psi_{n-1}\end{pmatrix}.$$

This tridiagonal matrix can be numerically diagonalised to the matrix D by a transformation matrix Z. The desired propagated wavefunction $\psi(t)$ can then be found by

$$\psi(t) = Z^\dagger e^{-\frac{i}{\hbar}Dt} Z \begin{pmatrix}\langle\psi_0|\psi(0)\rangle\\ \vdots\\ \langle\psi_{n-1}|\psi(0)\rangle\end{pmatrix}$$

i.e. if $\psi_0 = \psi(0)$ only the first column of the resulting matrix is needed.

In the one–dimensional testcase that I have implemented the Lanczos method was the slowest of all mentioned methods. Nevertheless, for a multidimensional system with hundreds of thousands of gridpoints it is still capable of reducing the problem to a much smaller number of basis vectors, so I expect that the efficiency compared to the other methods will be better in these cases. It is

also possible to include some knowledge into the method via the initial guess ψ_0 and thus make it semi–empirical to get a matrix representation of $\widehat{O}$ (or $\widehat{H}$, respectively) of smaller dimension (e.g. LCAO).

4 Appendix

4.1 Gaussian integration

The integral over a finite interval of a function $w(t)f(t)$ for an arbitrary function f with a non–negative weight function $w(t)$ can be approximated by integrating its approximation by Lagrange polynomials:

$$\begin{aligned}
\int_a^b w(t)f(t)dt &\approx \int_a^b w(t)\sum_{j=1}^{n}\prod_{\substack{i=1\\ i\neq j}}^{n}\frac{t-t_i}{t_j-t_i}f(t_j)\mathrm{d}t + \int_a^b w(t)v(t)f[t_1,\ldots,t_n,t]dt \\
&= \sum_{j=1}^{n}\int_a^b w(t)\prod_{\substack{i=1\\ i\neq j}}^{n}\frac{t-t_i}{t_j-t_i}\mathrm{d}t\, f(t_j) + R(f) \\
&= \sum_{j=1}^{n} A_j f(t_j) + R(f) \qquad (12)
\end{aligned}$$

where $v(t) = \prod_{i=1}^{n}(t-t_i)$. Thus the constant coefficients

$$A_j = \int_a^b w(t)\prod_{\substack{i=1\\ i\neq j}}^{n}\frac{t-t_i}{t_j-t_i}\mathrm{d}t$$

are found which are used for the approximation of the integral.

Definition: A number μ is called algebraic degree of accuracy if for all polynomials f with a degree smaller or equal to μ the above integration formula (12) is accurate, i. e. $\forall f \in \mathbf{R}[t] : \deg f \leq \mu \rightarrow R(f) = 0$.

Remark: Formula (12) is exact for polynomials of degree smaller or equal to $n-1$ for then the Lagrangian interpolation is exact. Furthermore consider following example:

$$\begin{aligned}\sum_{j=1}^{n} A_j v(t_j)^2 &= 0, \text{ as } v(t_j) = 0,\\ \int_a^b w(t)v(t)^2 \mathrm{d}t &> 0\,(w \,\text{non} - \text{negative weight}).\end{aligned}$$

Hence $n-1 \le \mu \le 2n-1$.

Theorem: An interpolation formula (12) has the maximal algebraic degree of accuracy of $\mu = 2n-1$ if and only if $\forall p \in \mathbf{R}[t]$, $\deg p \le n-1$:

$$\int_a^b w(t)v(t)p(t)\mathrm{d}t = 0.$$

proof: $(\rightarrow)$ $\mu = 2n-1$.

Then $\int_a^b w(t)v(t)p(t)\mathrm{d}t = \sum_{j=1}^{n} A_j v(t_j)p(t_j) = 0$ because $v(t_j) = 0$. This integral is exact because $\deg(v \cdot p) \le 2n-1$.

$(\leftarrow)$

$$\int_a^b w(t)v(t)p(t)\mathrm{d}t = 0. \tag{13}$$

Let $\deg f \le 2n-1$, $f = q \cdot v + r$, $\deg q \le n-1$, $\deg r \le n-1$. Then

$$\begin{aligned}\sum_{j=1}^{n} A_j f(t_j) &= \sum_{j=1}^{n} A_j r(t_j),\\ \int_a^b w(t)f(t)\mathrm{d}t &= \int_a^b w(t)q(t)v(t)\mathrm{d}t + \int_a^b w(t)r(t)\mathrm{d}t\\ &= \sum_{j=1}^{n} A_j r(t_j),\end{aligned}$$

since (13) and $\mu \ge n-1$.

q.e.d.

Remark A: From this the discrete orthogonality relation (5) from subsection 2.3 can be derived as follows: Let t_j be the zeros of the n–th orthogonal

polynomial P_n. Then $v = const \cdot P_n$ and hence it follows that $\mu = 2n-1$. This implies that formula (12) is exact for all integrals of the form

$$\int_a^b w(t)P_i(t)P_j(t)\mathrm{d}t = \delta_{ij},\ i,j < n.$$

Remark B:

1. $w(t) = \frac{1}{\sqrt{1-t^2}}$ for the Chebyshev polynomials.
2. For $w(t) \equiv 1$ formula (12) is the Gaussian integration formula.
3. Then $\sum_{j=1}^{n} A_j = \int_a^b \mathrm{d}t = b-a$, i.e. the A_j are constant *weights* (see 2.4).

4.2 Proof of the Chebyshev method (propagation)

$$b_n = (-i)^n 2J_n(\alpha),\ \alpha = \frac{\Delta E t}{2\hbar}.$$

proof: Most integral tables, e.g. [10], know the integrals

$$\begin{aligned}\int_0^1 \frac{T_{2k+1}(x)\sin(\alpha x)}{\sqrt{1-x^2}}dx &= (-1)^k\frac{\pi}{2}J_{2k+1}(\alpha) = (-i)^{2k}\frac{\pi}{2}J_{2k+1}(\alpha),\\ \int_0^1 \frac{T_{2k}(x)\cos(\alpha x)}{\sqrt{1-x^2}}dx &= (-1)^k\frac{\pi}{2}J_{2k}(\alpha) = (-i)^{2k}\frac{\pi}{2}J_{2k}(\alpha).\end{aligned}$$

Now, $\sin\alpha x$ is an odd function, so its representation as a series of Chebyshev polynomials would only contain odd order Chebyshev polynomials, i.e. $\sin\alpha x$ is orthogonal to all even order Chebyshev polynomials. Analogous to that $\cos\alpha x$ is orthogonal to all odd order Chebyshev polynomials:

$$\begin{aligned}\int_0^1 \frac{T_{2k}(x)\sin(\alpha x)}{\sqrt{1-x^2}}dx &= 0,\\ \int_0^1 \frac{T_{2k+1}(x)\cos(\alpha x)}{\sqrt{1-x^2}}dx &= 0.\end{aligned}$$

Furthermore the product of two odd functions is even, so for both integrals $\int_{-1}^{1} . dx = 2 \cdot \int_{0}^{1} . dx$. Hence

$$\begin{aligned}
\frac{\pi}{2} b_n &= \int_{-1}^{1} \frac{e^{-i\alpha x} T_n(x)}{\sqrt{1-x^2}} dx \\
&= \int_{-1}^{1} \frac{T_n(x)\cos(\alpha x)}{\sqrt{1-x^2}} dx - i \cdot \int_{-1}^{1} \frac{T_n(x)\sin(\alpha x)}{\sqrt{1-x^2}} dx \\
&= \begin{cases} (-i)^{2k}\pi J_{2k}(\alpha) - i \cdot 0, & n = 2k, \\ 0 - i \cdot (-i)^{2k}\pi J_{2k+1}(\alpha) = (-i)^{2k+1}\pi J_2(\alpha), & n = 2k+1 \end{cases} \\
&= (-i)^n \pi J_n(\alpha).
\end{aligned}$$

q.e.d.

References

[1] N. Balakrishnan, C. Kalyanaraman, N. Sathyamurthy: Phys. Rep. 280 (1997) 79.

[2] R. Kosloff; Annu. Rev. Phys. Chem., 45 (1994) 145.

[3] C. Leforestier et al.; J. Comp. Phys. 94 (1991) 59.

[4] R. Kosloff; J. Phys. Chem. 92 (1988) 2087.

[5] R. Kosloff; Quantum Molecular Dynamics on Grids *in* R. E. Wyatt, J. Z. H. Zhang; Dynamics of Molecules and Chemical Reactions, p. 185.

[6] C. Schwartz; J. Math. Phys. 26[3] (March 1985) 411.

[7] D. J. Tannor, R. Kosloff, S. A. Rice; J. Chem. Phys. 85[10] (15 Nov 1986) 5805.

[8] A. D. Hammerich, J. G. Muga, R. Kosloff; Israel J. Chem. 29 (1989) 461.

[9] A. D. Bandrauk, H. Shen; J. Chem Phys. 99[2] (15 July 1993) 1185.

[10] I. S. Gradshteyn, I. M. Ryzhik; Table of Integrals, Series, and Products, 5th Ed., 7.355 (p.842).