Homework **LTTC Quantum dynamics** Mauro Gascon Navas
TCCM March 19, 2025

All scripts and files generated while completing this homework can be found in this Github repository.

1 Hermite Polynomials Function

To calculculate the hermite polynomials, a recursive function can be used. The function calls itself to calculate $H_{n-1}(y)$ and $H_{n-2}(y)$ until reaching $H_1(y) = 1$ and $H_0(y) = 2y$, where its values are set and the recursion loop is exited. In the code *propagate.f90*, the functions are written in a module called *functions2use*.
Something worth noting, is that at each step of the recursion, the function is called twice, so the complexity of the function is $O(2^n)$, which is not ideal for large n and could cause a stack overflow. This contrasts to e.g. a factorial function (which is also typically written as a recursive function), where its complexity goes as $O(n)$. In this case this is not noticeable, but if we wanted to use it to find large order H_n, this function could be implemented as a loop.

The function taks two inputs: the order of the polynomial (n) and the point to evaluate (y), and outputs one result (h).

Listing 1: Recursive function to calculate hermite polynomials

```
recursive function hermite(n,y) result(h)
!Recursive function to calculate the hermite polynomials
    implicit none

    integer, intent(in) :: n
    real(8), intent(in) :: y
    real(8) :: h

    if (n<0) then !Check for proper usage
    write(*,*) "Error in hermite: n must be <=0"
    h=0.d0
    endif

    if (n==0) then !If statements to end recursion
    h=1.d0
    elseif (n==1) then
    h=2.d0*y
    else
    h=2.d0*y*hermite(n-1,y)-2.d0*(n-1)*hermite(n-2,y) !Recursion
    endif
end function hermite
```

2 3 4 Wavefunction initialization

The *initpsi* subroutine is modified so that the wavefunction is initialized as a weighted sum of the harmonic eigenfunctions. To make it as general as possible, first the number of eigenfunctions to use (*ncoeff*) is read from the input file *wavefunction*. Then the coefficients for each parent eigenfunction are read and stored as an array (*coeff*) in a sequential manner. In *initpsi*, *coeff* is looped over. Finally, the wavefunction is normalized exploiting the fact that eigenfunctions with nonequal eigenvalues are orthogonal to eachother: $N^2 = \Sigma_i c_i^2$.

The modified subroutine takes as inputs: *npoints* (x axis grid), *dx* (grid spacing), *x0* (center of the wavefunction), *psi0* (wavefunction array), *ncoeff* (number of eigenfunctions), *coeff* (ordered array with the eigenfuction coefficients).

Listing 2: Modified *initpsi* subroutine

```
subroutine initpsi(npoints,dx,x0,psi0,ncoeff,coeff)
!Modified initpsi to initiate wavefunction expanded in the harmonic
    eigenfunctions
    use parameters
    use functions2use
    implicit none

    integer :: i,j,l,npoints,ncoeff
    real(8) :: x,x0,dx,mo,sqmopi,sqmo,A,coeff(ncoeff), N
    complex(8) :: psi0(npoints), res

    mo = mass*angfreq !Constants that will be needed in the loop
    sqmo = sqrt(mo)
    sqmopi = sqmo/sqrt(pi)
    psi0 = (0.d0,0.d0)
    do l = 0, ncoeff-1 !Loop through the user specified eigenfunctions
        if (coeff(l+1)==0) cycle !if Coefficient = 0 skip
        A = sqrt(1.d0/(2.d0**l*dble(gamma(real(l+1))))*sqmopi) !Constant
        do i=-npoints/2+1,npoints/2
            x=dble(i)*dx
            if (i>0) then
            j=i
            else
            j=i+npoints
            endif
            res = coeff(l+1)*A*exp(-mo*(x-x0)**2/2.d0)*hermite(l,(x-x0)*sqmo) !
    coefficient*Eigenfucntion
            psi0(j) = psi0(j) + res
        end do
    enddo

    N = sqrt(sum(coeff**2))!Normalize eigenfunction
    psi0 = psi0 / N
end subroutine initpsi
```

The initial coefficients are read from the *wavefunction* file by modifying the main propgram *propagate*: The added arguments are the number of eigenfunctions to use (*ncoeff*), which is also the highest eigenfunction order, followed by the cefficients, one at each line in growing order. Additionally, to make the outputs lighter, a snapshot frequency is also added; it will only output the positions every *snapshot* steps.

Listing 3: Reading the coefficients in the main program

```
open(unit=10,file='wavepacket') ! User specified eigenfunctions/coefficents
    ...
    read(10,*) snapshot              !snapshot frequency
    read(10,*) ncoeff                !Number of coefficients
    allocate(coeff(ncoeff))          !Initialize coeff array
    coeff=0.d0
    do i = 1, ncoeff                 !Read ncoeff coefficients or until EOF
        read(10,*,iostat=iostat) coeff(i)
        if (iostat /= 0) exit
    end do
close(10)
```

Listing 4: Wavefunction input file

```
512       #Number of lattice points
-0.0      #Initial position (in Bohr)
12.755    #Governs the initial width of the wave packet (in Bohr^(-2))
0.01      #Propagation time step (in fs)
4000      #Number of propagation steps
20        #Snapshot frequency (after 20 timesteps a snapshot will be taken)
3         #Number of Coefficients
1         #c0
1         #c1
0         #c2
```

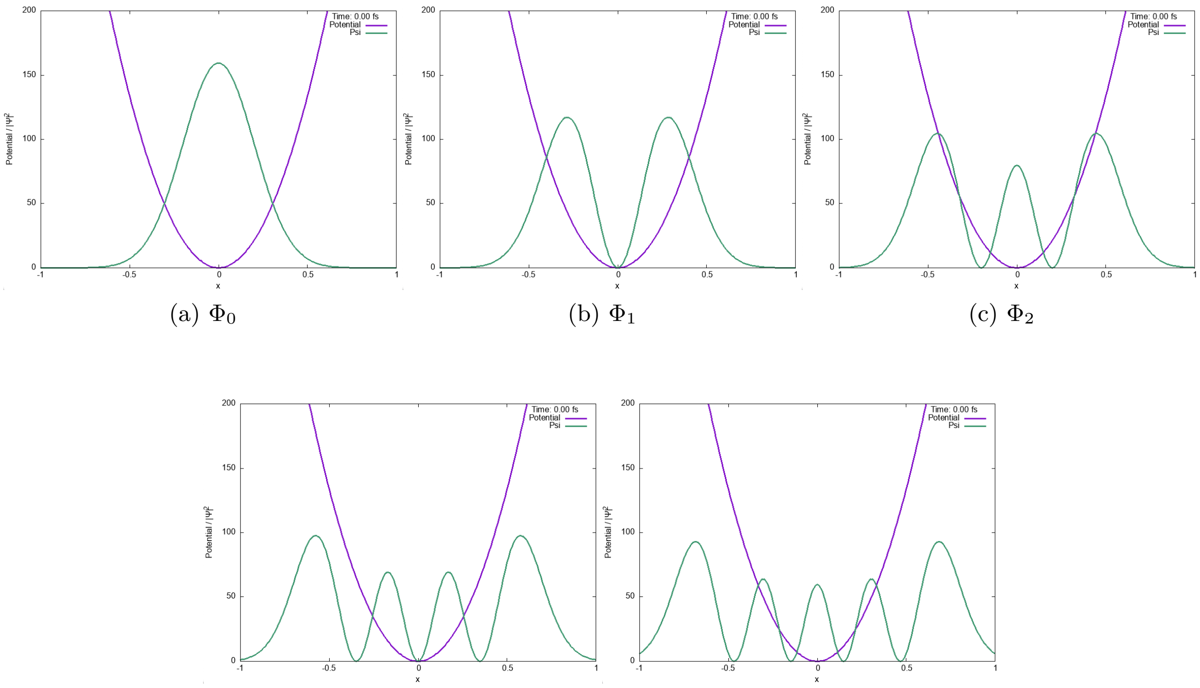
Finally, though not shown, for convenience the plotting subroutines were rewritten to use gnuplot.

5 Pure Eigenfunctions

For pure eigenfunctions, the probability density remains static:

$$\Psi_n(x,t) = e^{-iE_n t} \cdot \Phi_n(x) = e^{-it\omega(n+\frac{1}{2})} \cdot \Phi_n(x)$$

$$|\Psi_n(x,t)|^2 = [e^{itE_n} \cdot \Phi_n(x)][e^{-itE_n} \cdot \Phi_n(x)] = |\Phi_n(x)|^2$$



(a) Φ_0

(b) Φ_1

(c) Φ_2

(d) Φ_3

(e) Φ_4

Figure 1: Stationary $|\Psi_n|^2$ made of a single Eigenfucntion Φ_n

6 Mixed Eigenfunctions

When the wavefunction is in a superposition of eigenstates, the probability density will no longer be static as the frequency of each eigenstate is different, causing interferences:

$$\Psi_n(x,t) = \sum_n c_n \cdot e^{-iE_n t} \cdot \Phi_n(x) = \sum_n c_n \cdot e^{-it\omega(n+\frac{1}{2})} \cdot \Phi_n(x)$$

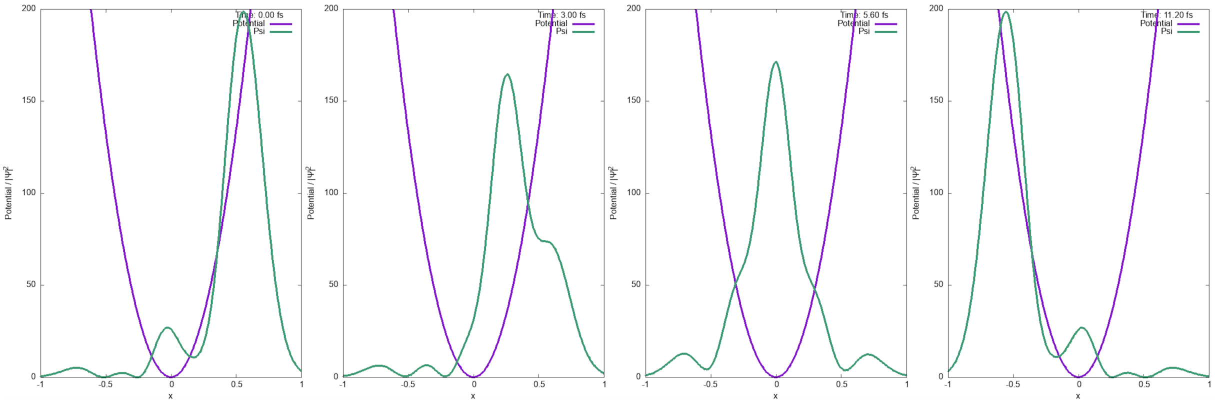
$$\begin{aligned}
|\Psi_n(x,t)|^2 &= \left[\sum_n c_n e^{-iE_n t}\Phi_n(x)\right]\left[\sum_n c_n e^{iE_n t}\Phi_n(x)\right] \\
&= \sum_n |c_n|^2|\Phi_n(x)|^2 + \sum_n \sum_{m>n} c_n c_m \Phi_n(x)\Phi_m(x) \cdot \left(e^{i(E_m-E_n)t} + e^{-i(E_m-E_n)t}\right) \\
&= \sum_n |c_n|^2|\Phi_n(x)|^2 + 2\sum_n \sum_{m>n} c_n c_m \Phi_n(x)\Phi_m(x) \cdot \mathrm{Re}\left(e^{-i(E_m-E_n)t}\right) \\
&= \sum_n |c_n|^2|\Phi_n(x)|^2 + 2\sum_n \sum_{m>n} c_n c_m \Phi_n(x)\Phi_m(x) \cdot \cos(t\omega(m-n))
\end{aligned}$$

The resulting probability density is a function of time where each cross term oscilates with a frequency of $\omega \cdot (m-n)$. The overall period will be determined by the greatest common divisor (GCD) of the $(m-n)$ set, being $T = \frac{2\pi}{GCD_{\{m-n\}} \cdot \omega}$.
In this case, the $\omega_0 = 0.2825$ fs^{-1}, giving a maximum period of $T = 22.23$ fs.

The overall motion of the wavefunction will come from the constructive/destructive interference of its parent eigenfunctions. For instance, all even Φ_n have an antinode at the center, so we can expect a great variation in the probability density there. Bellow the cases for $\Psi = |11111\rangle$, $\Psi = |10101\rangle$ and $\Psi = |01010\rangle$ are explored.

6.1 Case $\Psi = |11111\rangle$

The wavefunction is initialized as a superposition of the first 5 eigenfunctions with equal weight. The probability density is plotted at different times in Figure 2. $|\Psi_n(x,t)|^2$ oscilates with a period of $T = 2\pi/\omega = 22.23$ fs, as $GCD_{\{m-n\}} = 1$. The overall motion would be that of a ”travelling wave”.

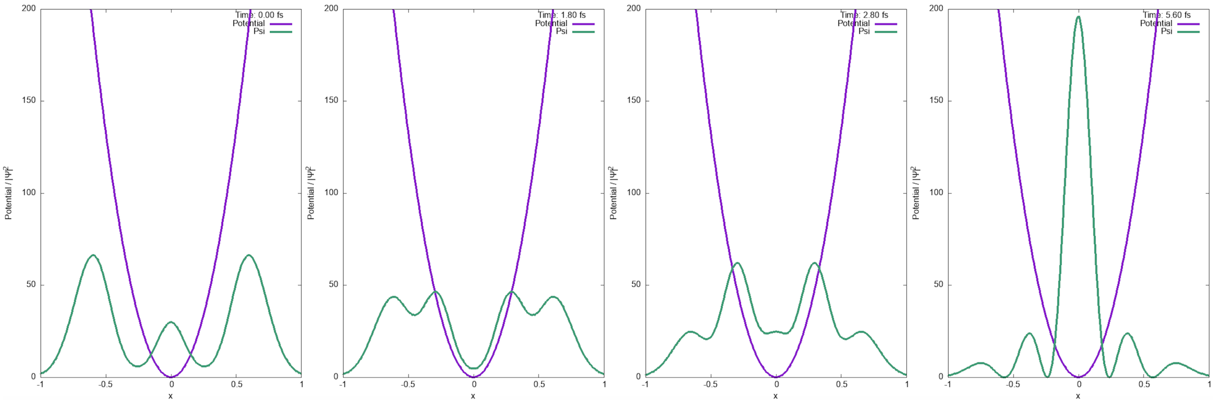


(a) $t = 0.0$ fs $(0T)$ (b) $t = 3.0$ fs $(0.13T)$ (c) $t = 5.6$ fs $(0.25T)$ (d) $t = 11.2$ fs $(0.5T)$

Figure 2: Time evolution for $\Psi = |11111\rangle$. Its period is $T = 2\pi/\omega = 22.23$ fs

6.2 Case $\Psi = |10101\rangle$

The wavefunction is initialized as a superposition of the first 3 even eigenfunctions with equal weight. The probability density is plotted at different times in Figure 3. $|\Psi_n(x,t)|^2$ oscilates with a period of $T = \pi/\omega = 11.12$ ($GCD_{\{m-n\}} = 2$). Its movement is that of a standing wave, symmetric around its center but without nodes. Because all the components are even functions, all contribute with an antinode in the center, and we can observe the greatest interference there.

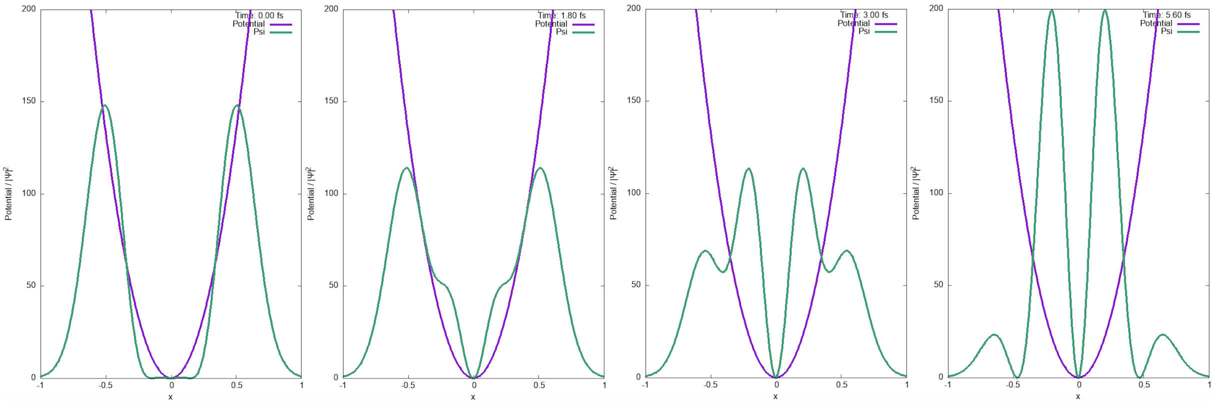


(a) $t = 0.0$ fs ($0T$) (b) $t = 1.8$ fs ($0.16T$) (c) $t = 2.8$ fs ($0.25T$) (d) $t = 5.60$ fs ($0.5T$)

Figure 3: Time evolution for $\Psi = |10101\rangle$. Its period is $T = \pi/\omega = 11.12$ fs

6.3 Case $\Psi = |01010\rangle$

The wavefunction is initialized as a superposition of the first 2 odd eigenfunctions with equal weight. The probability density is plotted at different times in Figure 4. $|\Psi_n(x,t)|^2$ oscilates with a period of $T = \pi/\omega = 11.12$ fs ($GCD_{\{m-n\}} = 2$). Its movement is that of a standing wave, symmetric around its center with a node in the middle. In contrast to the previous case, this node appears as the parent functions also contain it.



(a) $t = 0.0$ fs ($0T$) (b) $t = 1.8$ fs ($0.16T$) (c) $t = 3.0$ fs ($0.25T$) (d) $t = 5.60$ fs ($0.5T$)

Figure 4: Time evolution for $\Psi = |01010\rangle$. Its period is $T = \pi/\omega = 11.12$ fs