Time dependent Shrödinger equation

5^{th} week assignment
December 7, 2021

Campesan Giulia

Università degli Studi di Padova

1-D quantum harmonic oscillator: ground-state time evolution
Given the ground state $|\psi_0\rangle = |n=0\rangle$ of the mono-dimensional quantum harmonic oscillator, we want to compute its time evolution $|\psi(t)\rangle$ when subject to the **time-dependent Hamiltonian**

$$\hat{H}(t) = T + V(t) = \frac{\hat{p}}{2m} + \frac{1}{2}m\omega^2\left(\hat{q} - \hat{q}_0(t)\right)^2$$

in which $q_0(t) = \frac{t}{T}$, with T being the characteristic time of the potential
Knowing that the solution of the time-dependent Shrödinger equation is obtained by applying to $|\psi_0\rangle$ the **time-evolution operator** $\hat{U}(t)$, the required task ends up at solving numerically

$$\psi(x, t_0 + \Delta_t) = \exp(-i\Delta t\hat{H})\psi(x, t_0)$$

- time and space discretization
  $t_j = t_0 + j * \Delta t, j = 0, N_t \quad x_i = L_1 + i * \Delta x, i = 0, N_x$
- **Baker-Campbell-Haussdorf** approximation

$$exp(-i\hat{H}\Delta t) = exp(-i\hat{V}\frac{\Delta t}{2})exp(-i\hat{T}\Delta t)exp(-i\hat{V}\frac{\Delta t}{2}) + \mathcal{O}(\Delta t^3)$$

- **split-operator** method

$$\psi_0(x_i, t_{j+1}) \simeq e^{-iV(x_i,t_j)\frac{\Delta t}{2}}\mathcal{F}^{-1}e^{-iT(p_i)\Delta t}\mathcal{F}e^{-iV(x_i,t_j)\frac{\Delta t}{2}}\psi_0(x_i, t_j)$$

Code

- **Parameters setting**
  - Default values for the system parameters: $N_x = 2000, [L_1, L_2] = [-10, 10], N_t = 2000, [t_0, t_f] = [0, 7], T = 1, m = 1, \omega = 1$
  - User-defined parameters passed as command-line arguments
- exploiting the **dfftw_plan_dft_1d** and **dfftw_execute_dft** subroutines contained in the **fftw3** library

```
function FFT(psi) result(fft_psi)

    double complex :: psi(:), fft_psi(size(psi, 1))

    integer ::  N, FFTW_FORWARD, FFTW_MEASURE
    integer(8) :: plan

    parameter (FFTW_FORWARD=-1)
    parameter (FFTW_MEASURE=1)

     N = size(psi, 1)

    call dfftw_plan_dft_1d(plan,N,psi,fft_psi,FFTW_FORWARD,FFTW_MEASURE)
    call dfftw_execute_dft(plan, psi, fft_psi)

    fft_psi = fft_psi/sqrt(1d0 * N)

    call dfftw_destroy_plan(plan)
```

Figure: FFT function

- **Normalization factor**: $\frac{1}{\sqrt{N}}$

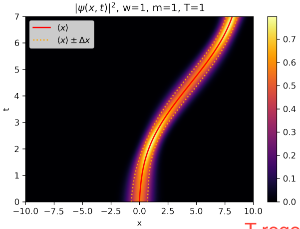
Code

- the **FFT** and **IFFT** functions are called inside the **time_evolution** function

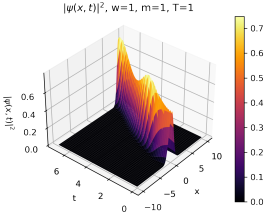
```
function time_evolution(psi, interval, omega, mass, time, TT, dt) result(psi_evolved)

    double complex :: psi(:), psi_evolved(size(psi, 1))
    real(8) ::  interval(2), omega, mass, time, TT, dt, V(size(psi, 1)), T(size(psi, 1))
    integer(8) :: N, ii

    N = size(psi, 1)-1

    V = H_V(N, interval, omega, mass, time, TT)
    T = H_T(N, interval, mass)

    do ii=1, N+1
        psi_evolved(ii)  =  EXP ( COMPLEX (0.0d0, - 1d0 / 2 * V(ii) * dt) )  * psi(ii)
    enddo

    psi_evolved = FFT(psi_evolved)

    do ii =1, N+1
        psi_evolved(ii) =  EXP ( COMPLEX (0.0d0, - 1d0 * T(ii) * dt) )  * psi_evolved(ii)
    enddo

    psi_evolved = IFFT(psi_evolved)

    do ii=1, N+1
        psi_evolved(ii)  =  EXP ( COMPLEX (0.0d0, - 1d0 / 2 * V(ii) * dt) )  * psi_evolved(ii)
    enddo

end function time_evolution
```

Figure: Time evolution function

Results

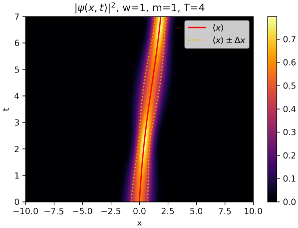


(a) $|\psi(x,t)|^2$



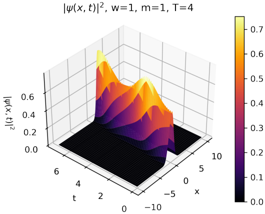
(b) $|\psi(x,t)|^2$, 3D plot

T regola se la trasformazione è
adiabatica o meno:
più T fa evolvere piano il potenziale
più facilmente la particella
lo seguirà

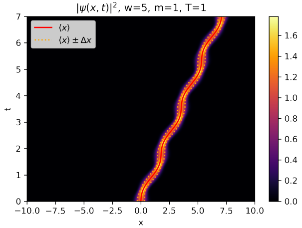


(a) $|\psi(x,t)|^2$

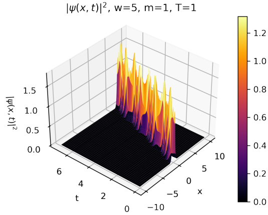
v=1/T



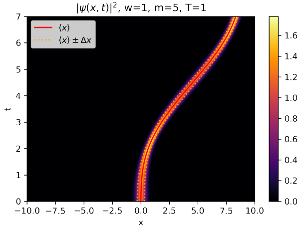
(b) $|\psi(x,t)|^2$, 3D plot



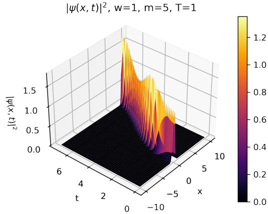
(a) $|\psi(x,t)|^2$



(b) $|\psi(x,t)|^2$, 3D plot



(a) $|\psi(x,t)|^2$



(b) $|\psi(x,t)|^2$, 3D plot