Quantum information and computing: Report 7

Nicola Lonigro
University of Padua
(Dated: November 23, 2020)

The time independent Shrodinger's equation is solved using the corresponding eigenvalue equation for an harmonic oscillator. The ground state is then evolved through the Split operator Method.

I. THEORY

The quantum harmonic oscillator with a potential well translating with time is described by the Hamiltonian

$$H = \frac{p^2}{2m} + \frac{1}{2}\omega^2(x - \frac{t}{T})^2$$

where T is a parameter and it is the time after which the well shifts to the x = 1 position. After finding the ground state at t = 0 by finding the first eigenfunction of the corresponding Hamiltonian[1], the wavefunction evolution with time is described by Schrodinger's equation. In particular the wavefunction at time $t+\Delta t$ will be given by

$$\psi(t+\Delta t) = e^{-iH\Delta t}\psi(t) = e^{-i(T+V)\Delta t}\psi(t)$$

where V is the potential operator and it will be diagonal in real space while T is the kinetic energy operator and it will be diagonal in the momentum space. It is possible to shift to momentum space and back to real space by applying the Fourier and anti-Fourier transform respectively. This method of time evolution is called **Split method operator** and it is given by

$$\psi(t+\Delta t) = e^{-\frac{iV}{2}\Delta t}\mathcal{F}^{-1}e^{-iT\Delta t}\mathcal{F}e^{-\frac{iV}{2}\Delta t}\psi(t)$$

II. CODE DEVELOPMENT

The previous code has been modified to not only compute the eigenvalues and eigenfunctions for a potential well centered in zero but also compute its time evolution with the split operator method. The previous input was modified to also include the T parameter and the discretization in time given by n_T

```
  read (5,*) n_eig,mass,omega,L,n_T,T

  CALL FATAL(   n_eig  <= 0 ,      "main","
    Number of eigenvalues must be positive" )
  CALL FATAL(   MOD(n_eig,2)/= 0 , "main","
    Number of eigenvalues must be even" )
  CALL FATAL(   mass   <= 0 ,      "main","Mass
    must be positive" )
  CALL FATAL(   L      <= 0 ,      "main","
    Length must be positive" )
  CALL FATAL(   n_T    <= 0 ,      "main","
    Number of time intervals must be positive" )
  CALL FATAL(   T      <= 0 ,      "main","T
    must be positive" )
```

The time interval [0,T] is divided in n_T steps and the matrix Time_Evolution contains the wave function at each time step in the interval. After computing and normalizing the ground state as done in the last report, the first column of the matrix is initialized with the ground state

```
temp_norm = 0

DO  j = 1,n_eig
  temp_norm = temp_norm + (matrix_1(j,1)**2)*dx
END DO
GS = cmplx(matrix_1(:,1)/SQRT(temp_norm))
Time_Evol(:,1) = GS
```

Each column is then iteratively computed starting from the last one using the Split operator method as described in the theory section

```
DO  i = 2,n_T
  CALL dfftw_plan_dft_1d(plan,n_eig,Time_Evol(:,
    i),temp_mom_space,FFTW_FORWARD,FFTW_ESTIMATE
    )
  !Evolve in real space
  DO  j = 1,n_eig
    Time_Evol(j,i) = Time_Evol(j,i-1)* cexp(
    cmplx( 0.0 , -0.25*(omega**2*(-1.0*L + (j-1)
    *dx - t_step*(i-1)/T)**2)*t_step) )
  END DO

  temp_mom_space = 0

  !pass to momentum space
  CALL dfftw_execute_dft(plan, Time_Evol(:,i),
    temp_mom_space)
  CALL dfftw_destroy_plan(plan)
  temp_mom_space = 1.0*temp_mom_space/n_eig

  !evolve in momentum space
  DO  j = 1,n_eig
    IF (j <= 1.0*n_eig/2) THEN
      temp_mom_space(j) = temp_mom_space(j)*
    cexp( cmplx( 0.0 , -2.0*((PI*j/2.0/L)**2)/
    mass*t_step) )
    ELSE
      temp_mom_space(j) = temp_mom_space(j)*
    cexp( cmplx( 0.0 , -2.0*((PI*(j- 1.0*n_eig)
    /2.0/L)**2)/mass*t_step) )
    END IF
  END DO

  !back to real space
  CALL dfftw_plan_dft_1d(plan,n_eig,
    temp_mom_space,Time_Evol(:,i),FFTW_BACKWARD,
    FFTW_ESTIMATE)
```

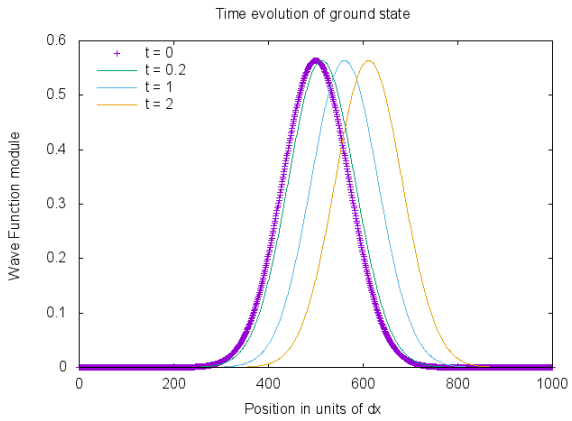

FIG. 1. Time evolution for T = 2s

```
  CALL dfftw_execute_dft(plan, temp_mom_space,
    Time_Evol(:,i))
  CALL dfftw_destroy_plan(plan)
  !Time_Evol(:,i) = Time_Evol(:,i)/ sqrt(DBLE(
    n_eig))

  DO  j = 1,n_eig
    Time_Evol(j,i) = Time_Evol(j,i)* cexp( cmplx
    ( 0.0 , -0.25*(omega**2*(-1.0*L + (j-1)*dx -
     t_step*(i-1)/T)**2)*t_step) )
  END DO

END DO
```

The norm for the wavefunction at all steps is computed and saved in a vector and all the wavefunctions are then normalized

```
DO  i = 1,n_T
  temp_norm = 0
  DO  j = 1,n_eig
    temp_norm = temp_norm + (Time_Evol(j,i)*
    CONJG(Time_Evol(j,i)))*dx
  END DO
  norms(i) = SQRT(temp_norm)
  Time_Evol(:,i) = Time_Evol(:,i)/norms(i)
END DO
```

Finally the probability density is computed for each position in real space and for each time stamp and it is saved on the corresponding txt file so that to each column correspond the proababilty density at a specific time and so they are easily plotted by gnuplot.

```
  DO  i = 1,n_T
    DO j = 1,n_eig
       Time_Evol(j,i) = Time_Evol(j,i)*CONJG(
    Time_Evol(j,i))
    END do
  END DO
```

```
  DO i = 1,n_eig
        write(11,*)REAL(Time_Evol(i,:))
  END DO
```

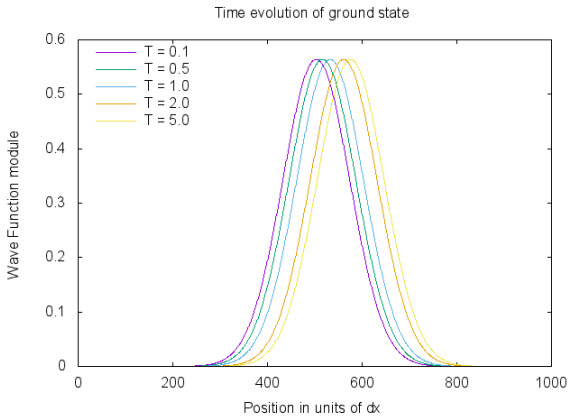

FIG. 2. Probability densities at t = 0.5 T for different values of T

III. RESULTS

The code has been tested in the interval [-5,5] with 1000 points in real space and 1000 time steps. The mass and omega have been left equal to 1 while the T parameter has been put equal to 2. The time evolution of the wavefunction is shown in Fig.1 and as expected it is just an horizontal shift of L = 1. Notice that the x axis is given in units of dx and so to the value of zero corresponds a position of -5 and to the value of 1000 a position of 5. In Fig. 2 wavefunctions obtained during different runs with different T parameters are all plotted at t = 0.5 T

IV. SELF-EVALUATION

The FFTW3 library has been installed and its application has been tested, in particular regarding the shift in the order of the Fourier components after transforming to momentum space. The split operator has been implemented and tested resulting in reasonable results for the time evolution of the wavefunction. Further studies on the evolution of the wavefunction when approaching the boundary of the position interval could be performed as well as different forms for the potential or a moving space interval that follows the well along

[1] N. Lonigro - Quantum information and computing: Report 6.