Presentation of Assignment 5

Quantum Information and Computing, A.Y. 2022/2023

Paolo Zinesi
10/12/2022

Theoretical introduction

The time-dependent Hamiltonian $\hat{H}(t)$ is obtained from the 1D harmonic oscillator Hamiltonian $\hat{\tilde{H}}$ with a time-dependent potential

$$\hat{\tilde{H}} = \frac{\hat{\tilde{p}}^2}{2m} + \frac{m}{2}\tilde{\omega}^2 \left(\hat{\tilde{q}} - q_0(t)\right)^2 \xrightarrow[\tilde{\omega}=2\omega]{\hbar=1,\, 2m=1} \hat{H} = \hat{p}^2 + \omega^2 \left(\hat{q} - q_0(t)\right)^2$$

with $q_0(t) = t/T$, $t \in [0, T]$.
Time evolution is performed using the split operator method,

$$\hat{\mathcal{U}}(\Delta t) = \exp\left[-i\,\Delta t\,\frac{\hat{V}}{2}\right] \mathcal{F}^{-1} \exp\left[-i\,\Delta t\,\hat{\mathcal{T}}\right] \mathcal{F} \exp\left[-i\,\Delta t\,\frac{\hat{V}}{2}\right] + \mathcal{O}(\Delta t^3)$$

$\mathcal{F}, \mathcal{F}^{-1}$: Fourier and Inverse Fourier Transforms

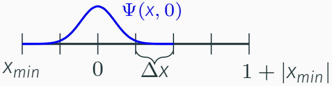
The initial wavefunction $\Psi(x, 0)$, obtained numerically in the previous assignment, is evolved N_t times using the propagator $\hat{\mathcal{U}}(\Delta t)$,

$$\Psi(x, t = N\,\Delta t) = \hat{\mathcal{U}}(\Delta t)^N\,\Psi(x, 0) \quad \text{with} \quad \Psi(x, 0) = \langle x | n = 0 \rangle$$

Discrete formulation

Tunable parameters:

- N_t: number of discretized points in time
- ω: frequency of the oscillator
- T: total time of evolution



$$L = 1 + 2\,|x_{min}| \quad x_{min} =< 0$$

A delicate point is the **adaptation** of the initial condition $\Psi(x, 0)$ to a different grid. **Solution**: use the same Δx defined in $\Psi(x, 0)$ and simply extend the system to the bigger size L (it is not possible to tune N_x directly!).

Initial condition adaptation (TD_HarmOsc_1D.f90)

```
! find indices of init WF to transfer to the total system
idx_minL = MAXLOC(psi0_x, MASK=psi0_x<(1.0D-10)*MAXVAL(psi0_x) .AND. (psi0_xgrid .LT. 0.D0), DIM=1)
idx_minR = MAXLOC(psi0_x, MASK=psi0_x<(1.0D-10)*MAXVAL(psi0_x) .AND. (psi0_xgrid .GT. 0.D0), DIM=1)
xmin = psi0_xgrid(idx_minL)

! Nx, Ltot depends directly on xmin
Ltot = 1.D0 + 2.D0*ABS(xmin)
Nx = CEILING(Ltot/deltax)

! fill wavefunctions in the x domain
psi_x%elem_fftw(1:(idx_minR - idx_minL + 1)) = psi0_x(idx_minL:idx_minR)
psi_x%elem_fftw((idx_minR - idx_minL + 2):) = 0.D0
```

Code development

A new module, **Zwavefunc_mod**, has been written to take care of the common operations performed on a double complex wavefunction, such as:

- Initialization (optionally with proper FFTW memory alignment)
- Grid definition
- Export to file
- Integration with Simpson's rule (optionally with a multiplicative function to compute momenta)

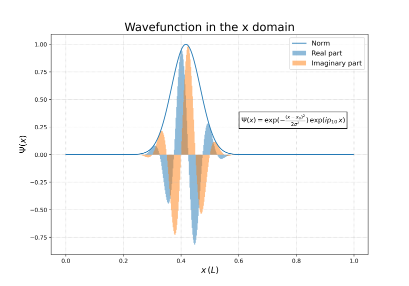
Implementation of the split operator method using Zwavefunc_mod (TD_HarmOsc_1D.f90)

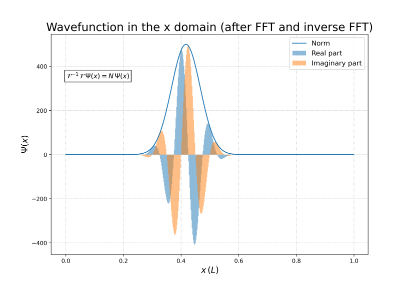
```fortran
! V/2 propagation
operator = EXP(COMPLEX(0.D0,-1.D0)*deltat*(omega**2)*0.5D0*(psi_x%grid - (t_idx*1.D0)/Nt)**2)
psi_x%elem_fftw = psi_x%elem_fftw * operator

! Fourier Transform
CALL fftw_execute_dft(plan_direct, psi_x%elem_fftw, psi_p%elem_fftw)

! T propagation
operator = EXP(COMPLEX(0.D0,-1.D0)*deltat*(psi_p%grid)**2)
psi_p%elem_fftw = psi_p%elem_fftw * operator

! Inverse Fourier Transform
CALL fftw_execute_dft(plan_inverse, psi_p%elem_fftw, psi_x%elem_fftw)
psi_x%elem_fftw = psi_x%elem_fftw / Nx

! V/2 propagation
operator = EXP(COMPLEX(0.D0,-1.D0)*deltat*(omega**2)*0.5D0*(psi_x%grid - (t_idx*1.D0)/Nt)**2)
psi_x%elem_fftw = psi_x%elem_fftw * operator
```

Results - FFTW testing

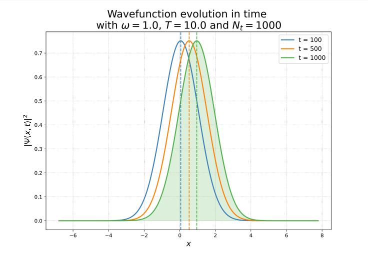
Testing of FFTW and Zwavefunc_mod (fftw_test.f90)

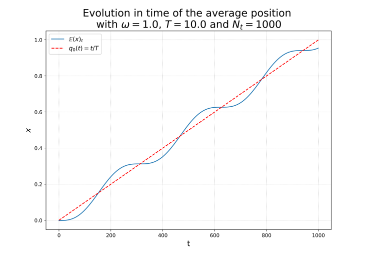
```fortran
! WF declarations and allocations
TYPE(Zwavefunc) :: psi_x, psi_p
psi_x = Zwavefunc(length=NN, need_fftw_alloc=.TRUE.)
psi_p = Zwavefunc(length=NN, need_fftw_alloc=.TRUE.)

! creation of plans
plan_direct = fftw_plan_dft_1d(NN, psi_x%elem_fftw,psi_p%elem_fftw, FFTW_FORWARD,FFTW_MEASURE)
plan_inverse = fftw_plan_dft_1d(NN, psi_p%elem_fftw,psi_x%elem_fftw, FFTW_BACKWARD,FFTW_MEASURE)

! FFT and inverse FFT execution
CALL fftw_execute_dft(plan_direct, psi_x%elem_fftw, psi_p%elem_fftw)
CALL fftw_execute_dft(plan_inverse, psi_p%elem_fftw, psi_x%elem_fftw)
```

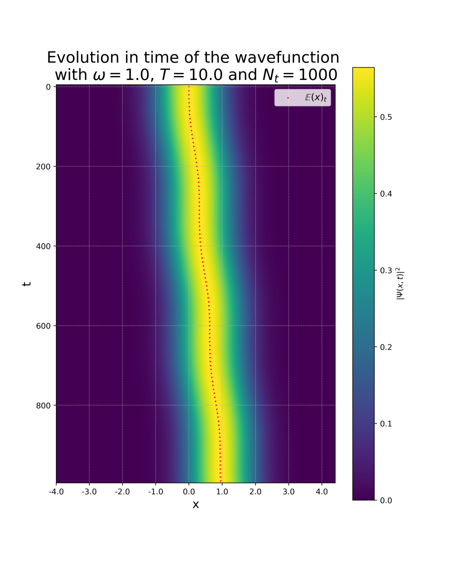




Results


Wavefunction evolution in time
with $\omega = 1.0$, $T = 10.0$ and $N_t = 1000$
t = 100
t = 500
t = 1000
$|\Psi(x,t)|^2$
x
−6 −4 −2 0 2 4 6 8
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7


Evolution in time of the average position
with $\omega = 1.0$, $T = 10.0$ and $N_t = 1000$
$\mathbb{E}(x)_t$
$q_0(t) = t/T$
x
t
0 200 400 600 800 1000
0.0 0.2 0.4 0.6 0.8 1.0


Evolution in time of the wavefunction
with $\omega = 1.0$, $T = 10.0$ and $N_t = 1000$
$\mathbb{E}(x)_t$
t
0 200 400 600 800
x
-4.0 -3.0 -2.0 -1.0 0.0 1.0 2.0 3.0 4.0
$|\Psi(x,t)|^2$
0.0 0.1 0.2 0.3 0.4 0.5