NB: The graded, first version of the report must be returned if you hand in a second time!

H3b: Time dependent quantum mechanics

Victor Nilsson and Simon Nilsson (930128-1854)

December 22, 2016

Task Nº €	Points	Avail. points
Σ		

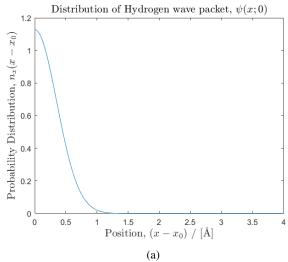
Introduction

Problem 1

The initial distribution of the Gaussian wave packet,

$$\psi(x;0) = \frac{1}{(\pi d^2)^{1/4}} \exp(-\frac{(x-x_0)^2}{2d^2}) \exp(ip_0(x-x_0)/\hbar), \tag{1}$$

where d is the witdth of the packet, x_0 and p_0 are the corresponding initial position and momenta.



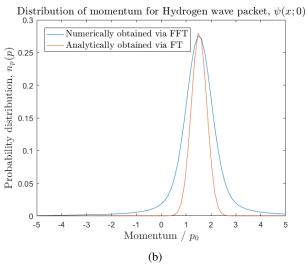


Figure 1: (a): * (b): *

1

Fourier of the gaussian,

$$\mathcal{F}\left[\psi(x;0)\right](p) = \int_{-inf}^{inf} e^{-ipx} \psi(x;0) dx \tag{2}$$

$$=\frac{1}{(\pi d^2)^{1/4}} \int_{-inf}^{inf} e^{-ipx} e^{-\frac{(x-x_0)^2}{2d^2}} e^{ip_0(x-x_0)/\hbar} dx \tag{3}$$

$$\left\{ a = \frac{e^{-ipx_0}}{(\pi d^2)^{1/4}} \right\} = a \int_{-inf}^{inf} e^{-ip(x-x_0)} e^{-\frac{(x-x_0)^2}{2d^2}} e^{ip_0(x-x_0)/\hbar} dx \tag{4}$$

$$\begin{cases}
 x' = x - x_0 \\
 dx' = dx \\
 x' \rightarrow x
\end{cases} = a \int_{-inf}^{inf} e^{-i(p - p_0/\hbar)x} e^{-\frac{x^2}{2d^2}} dx \tag{5}$$

$$\begin{cases}
p' &= p - p_0/\hbar \\
b &= \frac{1}{2d^2} \\
x' &= x - \frac{ip}{2b} \\
dx' &= dx \\
x' &\to x
\end{cases} = a \int_{-inf}^{inf} e^{-bx^2} e^{-p'^2/4b} dx \tag{6}$$

$$= ae^{-p'^2/4b} \int_{-inf}^{inf} e^{-bx^2} dx$$
 (7)

$$\left\{ \int_{-int}^{int} e^{-bx^2} dx = \sqrt{\frac{\pi}{2b}} \right\} = ae^{-p'^2/4b} \sqrt{\frac{\pi}{2b}}$$
 (8)

$$= (\pi d^2)^{1/4} e^{-ipx_0 - (p\hbar - p_0)^2 d^2/2\hbar^2}$$
(9)

References

A Source code

A.1 main1.m

```
% HOMEWORK 3B IN COMP.PHYS. - TASK 1
                       \% By Victor Nilsson, Simon Nilsson
                       % 2016
                       % Length scale: 1
                       % Time scale: 1 fs = 1e-15 s
                       % Energy scale: 1 eV
                        clear all, clc, close all
  11
  12
                                               ---- SIMULATION PARAMETERS -----
  13
                       hbar
                                                                                         = 1.054/1.602; % JS -> f eV s
                                                                                          = 0.5;
  16
                                                                                           = 1.66/1.6*1e2;
  17
                                                                                           = sqrt(0.1*2*m);
  18
                       x 0
  19
                       dx
                                                                                         = 0.01;
 20
                                                                                           = 1024;
                       n_points
                                                                                           = 2*pi/(n_points*dx);
23
                       % ----- VARIABLES -----
                       % space samples
                       x = x_0 + dx*(0:n_points-1);
                        % and the corresponding samples in momentum space
                       p = dp*((0:n_points-1)-n_points/2);
                         % ---- Functions handles
 29
                       Gaussian\_Wave\_Packet = @(x)1/(pi*d^2)^(1/4)*exp(-(x-x_0).^2/(2*d^2)).*exp(1i*p_0 \leftarrow 1/4)*exp(-(x-x_0).^2/(2*d^2)).*exp(1i*p_0 \leftarrow 1/4)*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-x_0).^2/(2*d^2)).*exp(-(x-
                                                   *(x-x_0)/hbar);
                       % Fourier transform obtained via Mathematica as the 'Inverse Fourier
                        % Transform', due to differences in FT defintion
                        Gaussian\_Packet\_Fourier = @(p) \underbrace{exp(-1i*p*x\_0 - (d^2*(p*hbar - p\_0).^2)./(2*hbar \leftarrow p_0).^2)}_{-1} + \underbrace{(d^2*(p*hbar - p\_0).^2)./(2*hbar \leftarrow p_0).^2)}_{-1} + \underbrace{(d^2*(p*hbar - p_0).^2)./(2*hbar \leftarrow p_0).^2)}_{-1} + \underbrace{(d^2*(p*hbar - p_0).^2)./(2*hbar \leftarrow p_0).^2)}_{-1} + \underbrace{(d^2*(p*hbar - p_0).^2)./(2*hbar \leftarrow p_0)}_{-1} + \underbrace{(d^2*(p*hbar - p_0)).^2)}_{-1} + \underbrace{(d^2*(p*hbar - p_0)).^2}_{-1} + \underbrace{(d^2*(p*hbar - p_0)).^2)}_{-1} + \underbrace{(d^2*(p*hbar - p_0)).^2}_{-1} + \underbrace{(d^2*(p*hbar - p_0)).^2)}_{-1} + \underbrace{(d^2*(p*hbar - p_0)).^2}_{-1} + \underbrace{(d^2*(p*hbar - p_0)).^2}_{-1} + \underbrace{(d^2*(p*hbar - p_0)).^2}_{-1} + \underbrace{(d^2*(p*hbar - p_0))}_{-1} + \underbrace{(d^2*(p*hbar - p_0))}_{-1} + \underbrace{(d^2*(p*hbar - p_0))}_{-1} + \underbrace{(d^2*(p*hbar - p_0))}_{-1} + \underbrace{(d^2*(p*hbar - p_0)}
                                                     ^2))*(pi*d^2)^(1/4);
 35
36
                       % Sample-discretize the wave packet function
 37
                        wave_packet = Gaussian_Wave_Packet(x)*dx;
                         theoretic_prob = abs(Gaussian_Packet_Fourier(p)).^2;
```

```
40
    prob = abs(wave_packet/dx).^2;
41
     fft_prob_momentum = abs(fftshift(fft(wave_packet))).^2*dp;
42
43
    % Plot prob.distr. in momentum space
    figure(1); clf;
plot(p/p_0, fft_prob_momentum)
44
46
     hold on
47
     plot(p/p_0, theoretic_prob)
48
     hold off
    xlim([-5 5])
xlabel('Momentum / $p_0$', 'interpreter', 'latex', 'fontsize', 14)
49
50
     ylabel('Probability distribution, $n_p(p)$','interpreter','latex', 'fontsize', ↔
51
52
     title('Distribution of momentum for Hydrogen wave packet, <math>x;0,'\leftarrow
    interpreter','latex','fontsize',14)
L = legend({'Numerically obtained via FFT', 'Analytically obtained via FT'},'
    interpreter','latex','fontsize',12);
set(L, 'location', 'northwest')
53
55
     % Plot prob.distr. in normal space
56
57
     figure(2); clf;
    plot(x, prob)
xlim([0 4])
58
59
    ylabel('Probability Distribution, $n_x(x-x_0)$','interpreter','latex', 'fontsize↔
60
     xlabel('Position, $(x-x_0)$ / [\AA]', 'interpreter', 'latex', 'fontsize', 14)
61
     title('Distribution of Hydrogen wave packet, $\psi(x;0)$','interpreter','latex',$ 
           'fontsize',14)
```

A.2 main2.m

```
% HOMEWORK 3B IN COMP.PHYS. - TASK 2
 4
    \% By Victor Nilsson, Simon Nilsson
    % 2015
 6
     % Length scale: 1
    % Time scale: 1 \text{ fs} = 1e-15 \text{ s}
    % Energy scale: 1 eV
 g
10
11
     clear all, clc, close all
12
     % ----- SIMULATION PARAMETERS ----
13
14
    hbar
               = 1.054/1.602; % JS -> f eV s
15
     d
                  = 0.5;
16
    m
                  = 1.66/1.6*1e2;
17
    p_0
                  = sqrt(0.1*2*m);
18
                  = 0:
    x_0
19
                  = 0.1;
    dx
                   = 2^10;
20
    n_points
21
                   = 2*pi/(n_points*dx);
     dp
                  = 1;
22
23
24
25
    % ----- VARIABLES -----
26
    % space samples
    x = x_0+dx*(0:n_points-1);
     % and the corresponding samples in momentum space
    p = ((0:n_points-1)-n_points/2)*dp;
29
30
     % Functions handles
     Gaussian\_Wave\_Packet = @(x)1/(pi*d^2)^(1/4)*exp(-(x-x_0).^2/(2*d^2)).*exp(1i*p_0 \leftarrow 1.0)
31
           *(x-x_0)/hbar);
     Potential_Function = @(x) 0;
32
33
34
     step_three=Gaussian_Wave_Packet(x);
35
36
     potential = Potential Function(x):
    exp_potential = exp(-1i/hbar.*potential*dt);
inv_pot = exp(-1i/hbar*(hbar^2*p.^2./(2*m))*dt);
37
38
39
40
41
    % Plot initial prob.distr.
     figure(1); clf;
42
43
    plot(x(1:n_points/2), abs(step_three(1:n_points/2).^2))
     xlim([0 2])
    xlabel('Position / [\AA]', 'interpreter', 'latex', 'fontsize', 14)
ylabel('Probability distribution', 'fontsize', 14)
title('$\left| \psi (0) \right|^2$', 'interpreter', 'latex', 'fontsize', 18)
45
46
47
48
49
     % Plot the rest in a separate figure
50
     figure(2); clf;
     step_one = step_three;
52
     step_two = fftshift(fft(step_one.*exp_potential));
53
     step_three = ifft(ifftshift(inv_pot.*step_two));
```

```
plotHandle = plot(x(1:n_points/2), abs(step_three(1:n_points/2).^2));
       riotrandle = prot(x(1.n=points/z), abs(step_intee(1.n=points/z), 2)),
xlabel('Position / [\AA]', 'interpreter', 'latex', 'fontsize', 14)
ylabel('Probability distribution', 'fontsize', 14)
title('$\left| \psi (t = 256 \mathrm{fs}) \right|^2$', 'interpreter', 'latex', '\left
fontsize', 18)
56
57
58
60
        for j=1:n_points/4-1
61
              step_one = step_three;
62
              step_two = fftshift(fft(step_one.*exp_potential));
step_three = ifft(ifftshift(inv_pot.*step_two));
63
64
65
               set(plotHandle, 'YData', abs(step_three(1:n_points/2).^2))
              pause (0.01)
```

A.3 main3.m

```
% HOMEWORK 3B IN COMP.PHYS. - TASK 2
    % By Victor Nilsson, Simon Nilsson
    % 2015
6
    % Length scale: 1
% Time scale: 1 fs = 1e-15 s
    % Energy scale: 1 eV
10
11
    clear all, clc, close all
12
    % ----- SIMULATION PARAMETERS -----
13
14
    hbar
                 = 1.054/1.602; % JS -> f eV s
15
                 = 0.5;
    d
                 = 1.66/1.6*1e2;
17
    p_0
                 = sqrt(0.12*2*m);
                 = 0;
18
    x_0
                 = 0.1;
= 2<sup>1</sup>0;
19
    dх
    n_points
20
                 = 2*pi/(n_points*dx);
    dp
23
    v_0=0.1;
24
    alpha=2.0;
25
26
27
            ----- VARIABLES -----
    % space samples
29
    x = x_0+dx*(0:n_points-1);
30
    \ensuremath{\text{\%}} and the corresponding samples in momentum space
31
    p = ((0:n_points-1)-n_points/2)*dp;
    % Functions handles
32
    Gaussian_Wave_Packet = @(x)1/(pi*d^2)^(1/4)*exp(-(x-x_0).^2/(2*d^2)).*exp(1i*p_0\leftrightarrow
          *(x-x_0)/hbar);
34
    Potential_Function = @(x) v_0*cosh(x/alpha).^(-2);
35
    step_three=Gaussian_Wave_Packet(x);
36
37
38
    potential = Potential_Function(x);
    exp_potential = exp(-1i/hbar.*potential*dt);
inv_pot = exp(-1i/hbar*(hbar^2*p.^2./(2*m))*dt);
40
41
42
    for j=1:n_points/4
43
         step_one = step_three;
44
45
         step_two = fftshift(fft(step_one.*exp_potential));
46
         step_three = ifft(ifftshift(inv_pot.*step_two));
47
48
         plot(x-max(x)/2,abs(ifftshift(step_three)).^2)
         pause (0.01)
49
    end
```

A.4 main4.m

```
clear all, clc, close all

clear all, clc, close all

% ----- SIMULATION PARAMETERS -----
hbar = 1.054/1.602; % JS -> f eV s
d = 0.5;
m = 1.66/1.6*1e2;
p_0 = sqrt(0.12*2*m);
x_0 = 0;
dx = 0.1;
n_points = 2^10;
dp = 2*pi/(n_points*dx);
dt = 1;
a = 0.3;
b = 0.4;
d = 0.7;
c = 0.05; % 0.1;

tmp = V_d(0,a,b,c,d);
[R,v]=eig(tmp);
```
