Computational Physics

Homework 6

Cheng-ching Lin

1.

(a). Using Von Neumann stability analysis to show implicit FTSC is stable

Implicit FTSC in Schrödinger equation is

$$\frac{i\hbar}{\tau}(\psi^{n+1} - \psi^{n}) = H\psi^{n+1}$$

$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V$$

$$\frac{i\hbar}{\tau}\left(\psi_j^{n+1} - \psi_j^{n}\right) = \left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V\right)\psi_j^{n+1}$$

$$= -\left(\frac{\hbar^2}{2m}\right)\frac{\psi_{j+1}^{n+1} + \psi_{j-1}^{n+1} - 2\psi_j^{n+1}}{h^2} + V\psi_j^{n+1}$$

Von Neumann analysis is try to put $a_j^n = A^n e^{ikjh}$ into the function, and $x_j = jh, t_n = (n-1)\tau$.

Therefore, we put it into Schrödinger equation.

$$\frac{i\hbar}{\tau}\left(A^{n+1}e^{ikjh} - A^{n}e^{ikjh}\right) = -\left(\frac{\hbar^2}{2m}\right)\frac{A^{n+1}e^{ik(j+1)h} + A^{n+1}e^{ik(j-1)h} - 2A^{n+1}e^{ikjh}}{h^2} + VA^{n+1}e^{ikjh}$$

We suppose that $\xi = \frac{A^{n+1}}{A^n}$, and if $|\xi| > 1$, it means that the function is unstable.

$$\frac{i\hbar}{\tau}(A^{n+1} - A^{n}) = -\left(\frac{\hbar^2}{2m}\right)\frac{A^{n+1}e^{ikh} + A^{n+1}e^{-ikh} - 2A^{n+1}}{h^2} + VA^{n+1}$$

$$\frac{i\hbar}{\tau}\left(1 - \frac{1}{\xi}\right) = -\left(\frac{\hbar^2}{2m}\right)\frac{e^{ikh} + e^{-ikh} - 2}{h^2} + V$$

$$\frac{1}{\xi} = \left(\frac{\hbar\tau}{2im}\right)\frac{e^{ikh} + e^{-ikh} - 2}{h^2} - \frac{\tau V}{i\hbar} + 1$$

$$\frac{1}{\xi} = \left(-i\frac{\hbar\tau}{mh^2}\right)\cos(kh) + i\frac{\hbar\tau}{mh^2} + i\frac{\tau V}{\hbar} + 1$$

$$\frac{1}{\xi} = \mathrm{i}(\left(\frac{\hbar\tau}{mh^2}\right)(1 - \cos(kh)) + \frac{\tau V}{\hbar}) + 1$$

$$\frac{1}{|\xi|} = \sqrt{\left(\frac{\hbar\tau}{mh^2}(1 - \cos(kh)) + \frac{\tau V}{\hbar}\right)^2 + 1}$$

$$|\xi| = \frac{1}{\sqrt{\underbrace{\left(\frac{\hbar\tau}{mh^2}(1 - \cos(kh)) + \frac{\tau V}{\hbar}\right)^2}_{>0} + 1}} < 1$$

Therefore, implicit FTSC in Schrödinger equation is stable.

(b). Using Tylor series to show the accuracy of Crank-Nicolson scheme.

Crank-Nicolson scheme is below:

$$\psi^{n+1}=\left(\frac{1-\frac{i\tau}{2\hbar}H}{1+\frac{i\tau}{2\hbar}H}\right)\psi^{n}$$

$$\psi^{n+1}+\frac{i\tau}{2\hbar}\left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}+V\right)\psi^{n+1}=\ \psi^{n}-\frac{i\tau}{2\hbar}\left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}+V\right)\psi^{n}$$

$$\psi_j^{n+1}-\left(\frac{i\tau\hbar}{4m}\right)\frac{\psi_{\mathrm{j+1}}^{\mathrm{n+1}}+\psi_{j-1}^{n+1}-2\psi_j^{n+1}}{h^2}+\frac{i\tau}{2\hbar}V\psi_j^{n+1}=\ \psi^{n}+\left(\frac{i\tau\hbar}{4m}\right)\frac{\psi_{\mathrm{j+1}}^{\mathrm{n}}+\psi_{j-1}^{n}-2\psi_j^{n}}{h^2}-\frac{i\tau}{2\hbar}V\psi_j^{n}\ \dots(1)$$

Tylor series can be express by these terms:

$$\psi_j^{n+1}=\psi_j^n+\Delta t\left(\frac{\partial\psi_j^n}{\partial t}\right)+\frac{\Delta t^2}{2}\left(\frac{\partial^2\psi_j^n}{\partial t^2}\right)+\frac{\Delta t^3}{6}\left(\frac{\partial^3\psi_j^n}{\partial t^3}\right)+\frac{\Delta t^4}{24}\left(\frac{\partial^4\psi_j^n}{\partial t^4}\right)+O(\Delta t^5)$$

$$\psi_{j+1}^{n}=\psi_j^n+\Delta x\left(\frac{\partial\psi_j^n}{\partial x}\right)+\frac{\Delta x^2}{2}\left(\frac{\partial^2\psi_j^n}{\partial x^2}\right)+\frac{\Delta x^3}{6}\left(\frac{\partial^3\psi_j^n}{\partial x^3}\right)+\frac{\Delta x^4}{24}\left(\frac{\partial^4\psi_j^n}{\partial x^4}\right)+O(\Delta x^5)$$

$$\begin{aligned}\psi_{\mathrm{j+1}}^{\mathrm{n+1}}=\ &\psi_j^n+\Delta x\left(\frac{\partial\psi_j^n}{\partial x}\right)+\frac{\Delta x^2}{2}\left(\frac{\partial^2\psi_j^n}{\partial x^2}\right)+\frac{\Delta x^3}{6}\left(\frac{\partial^3\psi_j^n}{\partial x^3}\right)+\frac{\Delta x^4}{24}\left(\frac{\partial^4\psi_j^n}{\partial x^4}\right)+\Delta t\left(\frac{\partial\psi_j^n}{\partial t}\right)+\frac{\Delta t^2}{2}\left(\frac{\partial^2\psi_j^n}{\partial t^2}\right)\\&+\frac{\Delta t^3}{6}\left(\frac{\partial^3\psi_j^n}{\partial t^3}\right)+\frac{\Delta t^4}{24}\left(\frac{\partial^4\psi_j^n}{\partial t^4}\right)+\Delta x\Delta t\left(\frac{\partial^2\psi_j^n}{\partial x\partial t}\right)+\Delta x^2\Delta t\left(\frac{\partial^3\psi_j^n}{\partial x^2\partial t}\right)+\Delta x\Delta t^2\left(\frac{\partial^3\psi_j^n}{\partial x\partial t^2}\right)\\&+\Delta x^3\Delta t\left(\frac{\partial^4\psi_j^n}{\partial x^3\partial t}\right)+\Delta x^2\Delta t^2\left(\frac{\partial^4\psi_j^n}{\partial x^2\partial t^2}\right)+\Delta x\Delta t^3\left(\frac{\partial^4\psi_j^n}{\partial x\partial t^3}\right)\end{aligned}$$

Therefore, we can bring these terms into equation(1). On the left hand side:

$$\begin{aligned}LHS=\ &\psi_j^n+\Delta t\left(\frac{\partial\psi_j^n}{\partial t}\right)+\frac{\Delta t^2}{2}\left(\frac{\partial^2\psi_j^n}{\partial t^2}\right)+\frac{\Delta t^3}{6}\left(\frac{\partial^3\psi_j^n}{\partial t^3}\right)+\frac{\Delta t^4}{24}\left(\frac{\partial^4\psi_j^n}{\partial t^4}\right)\\&-\left(\frac{i\tau\hbar}{4mh^2}\right)\left[\Delta x^2\left(\frac{\partial^2\psi_j^n}{\partial x^2}\right)+\frac{\Delta x^4}{12}\left(\frac{\partial^4\psi_j^n}{\partial x^4}\right)+2\Delta x^2\Delta t\left(\frac{\partial^3\psi_j^n}{\partial x^2\partial t}\right)+2\Delta x^2\Delta t^2\left(\frac{\partial^4\psi_j^n}{\partial x^2\partial t^2}\right)\right]\\&+\frac{i\tau}{2\hbar}V[\psi_j^n+\Delta t\left(\frac{\partial\psi_j^n}{\partial t}\right)+\frac{\Delta t^2}{2}\left(\frac{\partial^2\psi_j^n}{\partial t^2}\right)+\frac{\Delta t^3}{6}\left(\frac{\partial^3\psi_j^n}{\partial t^3}\right)+\frac{\Delta t^4}{24}\left(\frac{\partial^4\psi_j^n}{\partial t^4}\right)]\end{aligned}$$

On the right hand side:

$$\mathrm{RHS}=\ \psi_j^n+\left(\frac{i\tau\hbar}{4mh^2}\right)\left[\Delta x^2\left(\frac{\partial^2\psi_j^n}{\partial x^2}\right)+\frac{\Delta x^4}{12}\left(\frac{\partial^4\psi_j^n}{\partial x^4}\right)\right]-\frac{i\tau}{2\hbar}V\psi_j^n$$

Also, $\Delta\mathrm{t}=\tau,\Delta\mathrm{x}=\mathrm{h}$:

$$\begin{aligned}0=\ &\tau\left(\frac{\partial\psi_j^n}{\partial t}\right)+\frac{\tau^2}{2}\left(\frac{\partial^2\psi_j^n}{\partial t^2}\right)+\frac{\tau^3}{6}\left(\frac{\partial^3\psi_j^n}{\partial t^3}\right)+\frac{\tau^4}{24}\left(\frac{\partial^4\psi_j^n}{\partial t^4}\right)\\&-\left(\frac{i\tau\hbar}{4m}\right)\left[2\left(\frac{\partial^2\psi_j^n}{\partial x^2}\right)+\frac{\mathrm{h}^2}{6}\left(\frac{\partial^4\psi_j^n}{\partial x^4}\right)+2\tau\left(\frac{\partial^3\psi_j^n}{\partial x^2\partial t}\right)+2\tau^2\left(\frac{\partial^4\psi_j^n}{\partial x^2\partial t^2}\right)\right]+\frac{i\tau}{2\hbar}V[2\psi_j^n\\&+\tau\left(\frac{\partial\psi_j^n}{\partial t}\right)+\frac{\tau^2}{2}\left(\frac{\partial^2\psi_j^n}{\partial t^2}\right)+\frac{\tau^3}{6}\left(\frac{\partial^3\psi_j^n}{\partial t^3}\right)+\frac{\tau^4}{24}\left(\frac{\partial^4\psi_j^n}{\partial t^4}\right)]\end{aligned}$$

$$= -i\hbar\left(\frac{\partial \psi_j^n}{\partial t}\right) + \frac{-i\hbar\tau}{2}\left(\frac{\partial^2 \psi_j^n}{\partial t^2}\right) + \frac{-i\hbar\tau^2}{6}\left(\frac{\partial^3 \psi_j^n}{\partial t^3}\right) + \frac{-i\hbar\tau^3}{24}\left(\frac{\partial^4 \psi_j^n}{\partial t^4}\right)$$

$$- \left(\frac{\hbar^2}{4m}\right)\left[2\left(\frac{\partial^2 \psi_j^n}{\partial x^2}\right) + \frac{\mathrm{h}^2}{6}\left(\frac{\partial^4 \psi_j^n}{\partial x^4}\right) + 2\tau\left(\frac{\partial^3 \psi_j^n}{\partial x^2 \partial t}\right) + 2\tau^2\left(\frac{\partial^4 \psi_j^n}{\partial x^2 \partial t^2}\right)\right] + \frac{1}{2}V[2\psi_j^n + \tau\left(\frac{\partial \psi_j^n}{\partial t}\right)$$

$$+ \frac{\tau^2}{2}\left(\frac{\partial^2 \psi_j^n}{\partial t^2}\right) + \frac{\tau^3}{6}\left(\frac{\partial^3 \psi_j^n}{\partial t^3}\right) + \frac{\tau^4}{24}\left(\frac{\partial^4 \psi_j^n}{\partial t^4}\right)]$$

So,

$$i\hbar\left(\frac{\partial \psi_j^n}{\partial t}\right) + \left(\frac{\hbar^2}{2m}\right)\left(\frac{\partial^2 \psi_j^n}{\partial x^2}\right) - V\psi_j^n = 0$$

$$= \frac{-i\hbar\tau}{2}\left(\frac{\partial^2 \psi_j^n}{\partial t^2}\right) + \frac{-i\hbar\tau^2}{6}\left(\frac{\partial^3 \psi_j^n}{\partial t^3}\right) + \frac{-i\hbar\tau^3}{24}\left(\frac{\partial^4 \psi_j^n}{\partial t^4}\right)$$

$$- \left(\frac{\hbar^2}{4m}\right)\left[\frac{\mathrm{h}^2}{6}\left(\frac{\partial^4 \psi_j^n}{\partial x^4}\right) + 2\tau\left(\frac{\partial^3 \psi_j^n}{\partial x^2 \partial t}\right) + 2\tau^2\left(\frac{\partial^4 \psi_j^n}{\partial x^2 \partial t^2}\right)\right]$$

$$+ \frac{1}{2}V\left[\tau\left(\frac{\partial \psi_j^n}{\partial t}\right) + \frac{\tau^2}{2}\left(\frac{\partial^2 \psi_j^n}{\partial t^2}\right) + \frac{\tau^3}{6}\left(\frac{\partial^3 \psi_j^n}{\partial t^3}\right) + \frac{\tau^4}{24}\left(\frac{\partial^4 \psi_j^n}{\partial t^4}\right)\right]$$

$$\approx \frac{-i\hbar\tau}{2}\left(\frac{\partial^2 \psi_j^n}{\partial t^2}\right) - \frac{(\hbar h)^2}{24m}\left(\frac{\partial^4 \psi_j^n}{\partial x^4}\right) + \frac{V\tau}{2}\left(\frac{\partial \psi_j^n}{\partial t}\right) + O(\tau^3) + O(h^5)$$

We can see that Crank-Nicolson scheme accurate to the second order.

(2).Find the retarded time

We suppose that the charge orbit is $\vec{r_1}(t) = (ct, 0, v_0 t)$, and the observer is on $\vec{r}(t) = (ct, 1, r_y)$.

The distance between is:

$$R^\mu = \vec{r}(t) - \vec{r_1}(t_{ret}) = (c(t - t_{ret}), 1, r_y - v_0 t_{ret})$$

We know electromagnetic field propagate by speed of light c = 1, so R is lightlike.

$$R^\mu R_\mu = 0 = -(t - t_{ret})^2 + 1 + \left(r_y - v_0 t_{ret}\right)^2$$

We rewrite the formula, and $v_0 = \frac{1}{2}$.

$$\frac{3}{4} t_{ret}^2 + (r_y - 2t) t_{ret} + t^2 - r_y^2 - 1 = 0$$

$$\mathrm{t_{ret}} = \frac{2}{3}\left((2t - r_y) \pm \sqrt{(2t - r_y)^2 + 3(1 + r_y^2 - t^2)} \right)$$

There is a constraint that $t_{ret} < t$, for we cannot receive the message from future.

In another way, we also use Newton method to find retarded time. Newton method is that to use first derivation to get closer the real answer. We suppose x_{ans} is real answer of $f(x)$, and x_i is the answer what we guess.

$$f(x_{ans}) = 0 = f(x_0) - \delta x_0 f'(x_0)$$

$$\delta x_0 = \frac{f(x_0)}{f'(x_0)}$$

$$x_1 = x_0 - \delta x_0$$

This process will continue until $\delta x_i < \Delta_{accuracy}$. Therefore, in this case, δt_{ret} will become this terms:

$$\delta t_{ret} = \frac{\frac{3}{4} t_{ret}^2 + (r_y - 2t) t_{ret} + t^2 - r_y^2 - 1}{\frac{3}{2} t_{ret} + r_y - 2t}$$

First, I create the two functions in the code.

Code:

```
import matplotlib.pyplot as plt
import math
acc = 0.01 #In the second question,the answer of accuracy are 0.01

def analytic(ry,t): #analytic solution
    ans = ((2*t-ry)+math.sqrt(4*pow(ry,2)+pow(t,2)-4*ry*t+3))/1.5
    if ans >t: #retarded time cannot be bigger than observer time
        ans = ((2 * t - ry) - math.sqrt(4 * pow(ry, 2) + pow(t, 2) - 4 * ry * t + 3)) / 1.5
    return ans

def delta(ry,t,gu): #Newton method; This function is create the delta
    ans = (0.75*pow(gu,2)+(ry-2*t)*gu+pow(t,2)-pow(ry,2)-1)/(1.5*gu+ry-2*t)
    return ans
```

Next, we will try to plot the retarded time from t=-10~10, with condition r_y =1,0,-1.

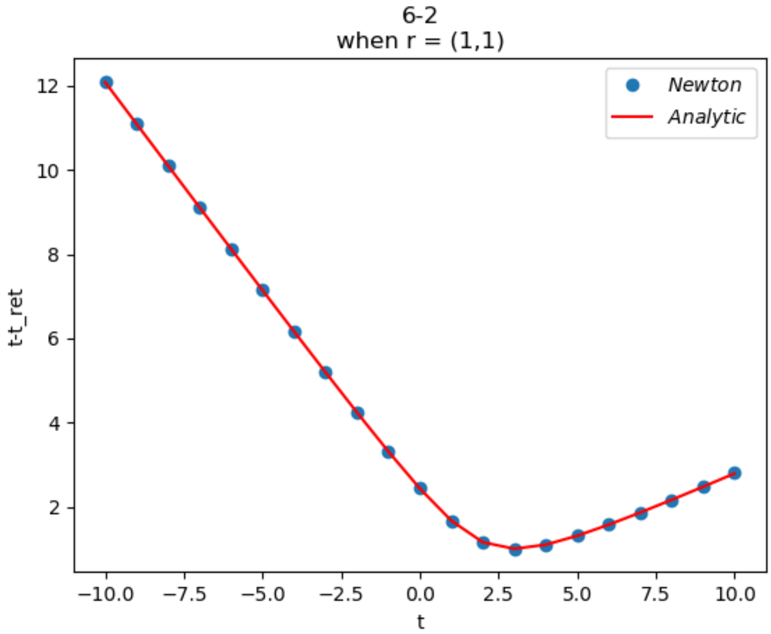
Code:

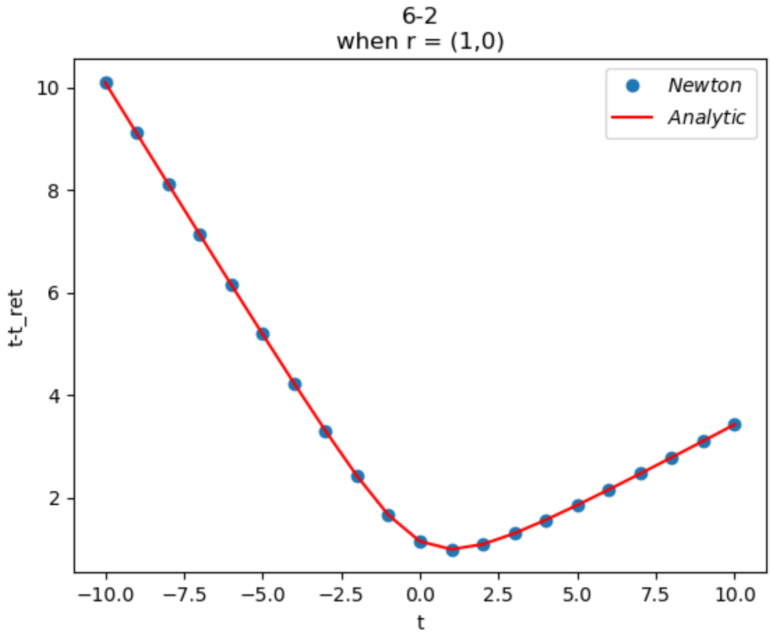
```
def test_1():
    ry = 1
    degu = 0
    ana = []
    new = []
    tt = []
    for idx in range(21):
        t = -10+idx
        tempa = analytic(ry,t)
        temp = t-tempa
        ana.append(temp)
        tt.append(t)

    for jdx in range(21):
        idx = 1
        t = -10 + jdx
        gu = t-10
        while idx>=0:
            gu -= degu
            tempnew = delta(ry,t,gu)
            if abs(tempnew) > acc:
                degu = tempnew
            else:
                idx = -1
        temp = t-gu
        new.append(temp)
    plt.plot(tt, new, "o", label="$Newton$")
    plt.plot(tt, ana, "r", label="$Analytic$")
    plt.title("6-2\nwhen r = (1,1)")
    plt.xlabel("t")
    plt.ylabel("t-t_ret")
    plt.legend()
    plt.show()

test_1()

def test_2():
    ry = 0
    degu = 0
    ana = []
```

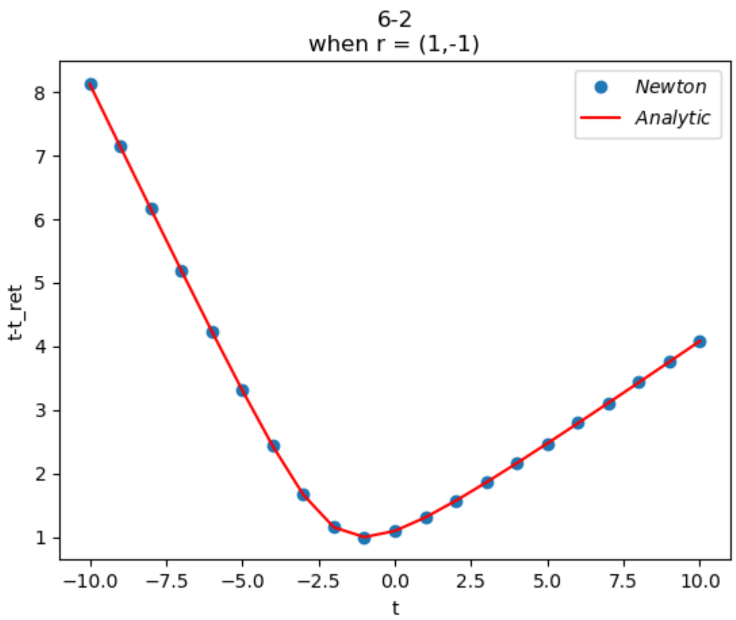
```
    new = []
    tt = []
    for idx in range(21):
        t = -10+idx
        tempa = analytic(ry,t)
        temp = t-tempa
        ana.append(temp)
        tt.append(t)

    for jdx in range(21):
        idx = 1
        t = -10 + jdx
        gu = t-10
        while idx>=0:
            gu -= degu
            tempnew = delta(ry,t,gu)
            if abs(tempnew) > acc:
                degu = tempnew
            else:
                idx = -1
        temp = t-gu
        new.append(temp)

    plt.plot(tt, new, "o", label="$Newton$")
    plt.plot(tt, ana, "r", label="$Analytic$")
    plt.title("6-2\nwhen r = (1,0)")
    plt.xlabel("t")
    plt.ylabel("t-t_ret")
    plt.legend()
    plt.show()

test_2()

def test_3():
    ry = -1
    degu = 0
    ana = []
    new = []
    tt = []
    for idx in range(21):
        t = -10+idx
        tempa = analytic(ry,t)
```

```
        temp = t-tempa
        ana.append(temp)
        tt.append(t)

    for jdx in range(21):
        idx = 1
        t = -10 + jdx
        gu = t-10
        while idx>=0:
            gu -= degu
            tempnew = delta(ry,t,gu)
            if abs(tempnew) > acc:
                degu = tempnew
            else:
                idx = -1
        temp = t - gu
        new.append(temp)

    plt.plot(tt,new,"o",label = "$Newton$")
    plt.plot(tt, ana, "r", label="$Analytic$")
    plt.title("6-2\nwhen r = (1,-1)")
    plt.xlabel("t")
    plt.ylabel("t-t_ret")
    plt.legend()
    plt.show()

test_3()
```

Result:


6-2
when r = (1,1)
Newton
Analytic
t-t_ret
t


6-2
when r = (1,0)
Newton
Analytic
t-t_ret
t


6-2
when r = (1,-1)
Newton
Analytic
t-t_ret
t

(3). Charge in circular motion

(a) Calculating retarded time

In this question, the condition is following: $q = 1, c = 1, \omega = 0.1\ or\ 0.9$, and charge orbit $\overrightarrow{\mathrm{r_q}}(t) = (\cos(\omega t), \sin(\omega t))$, calculating in the grid $\overrightarrow{\mathrm{r_{i,j}}} = \big(-10 + 0.4(i-1), -10 + 0.4(j-1)\big)$ at t = 0.

We need to calculate δx because we will use Newton method to solve this question. First, we need to find the distance R:

$$R = \overrightarrow{r_{i,j}} - \overrightarrow{r_q}(t) = (-t_{ret}, r_x - \cos(\omega t_{ret}), r_y - \sin(\omega t_{ret}))$$

R is lightlike, so:

$$R^{\mu}R_{\mu} = -t_{ret}^2 + (r_x - cos(\omega t_{ret}))^2 + \left(r_y - sin(\omega t_{ret})\right)^2 = 0$$

$$= -t_{ret}^2 + r_x^2 + r_y^2 - 2r_x\, cos(\omega t_{ret}) - 2r_y sin(\mathrm{\omega t_{ret}})$$

And the delta is:

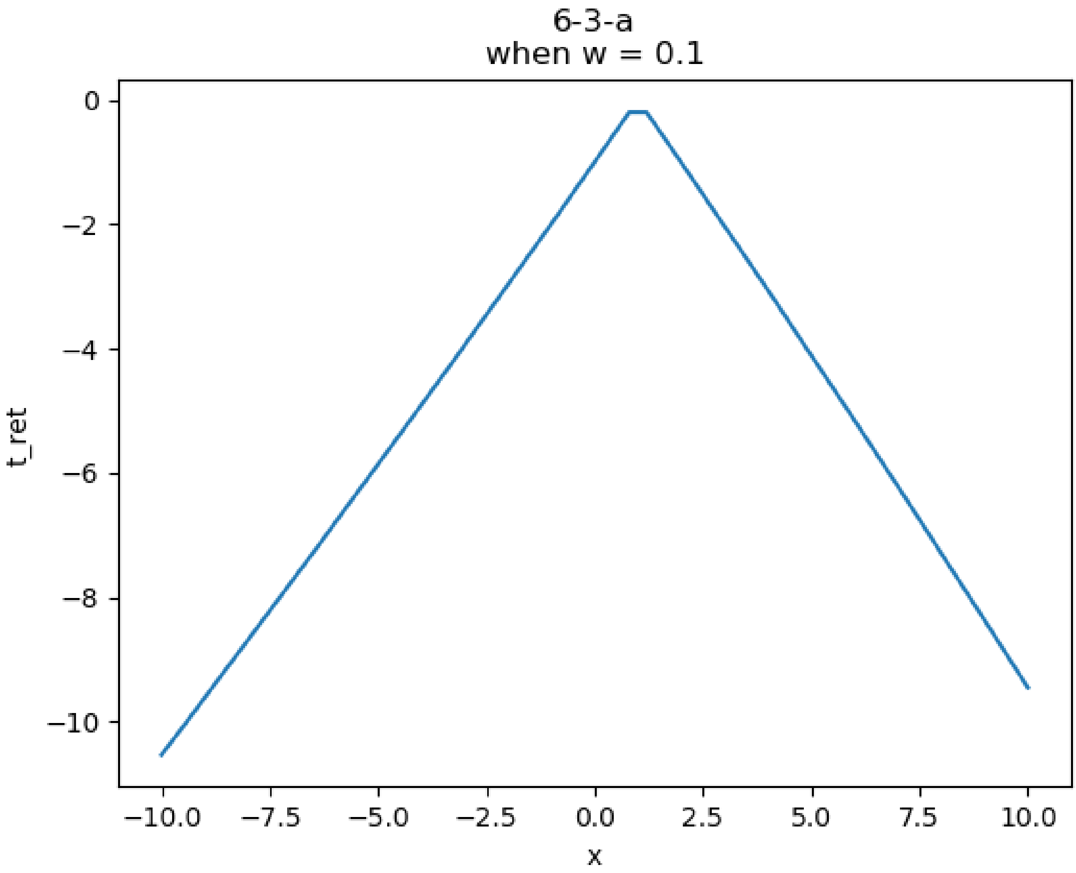
$$\delta t_{ret} = \frac{-t_{ret}^2 + r_x^2 + r_y^2 - 2r_x\, cos(\omega t_{ret}) - 2r_y si\,n(\mathrm{\omega t_{ret}})}{-2t_{ret} + 2r_x\omega\sin(\omega t_{ret}) - 2r_y\omega\cos(\omega t_{ret})}$$

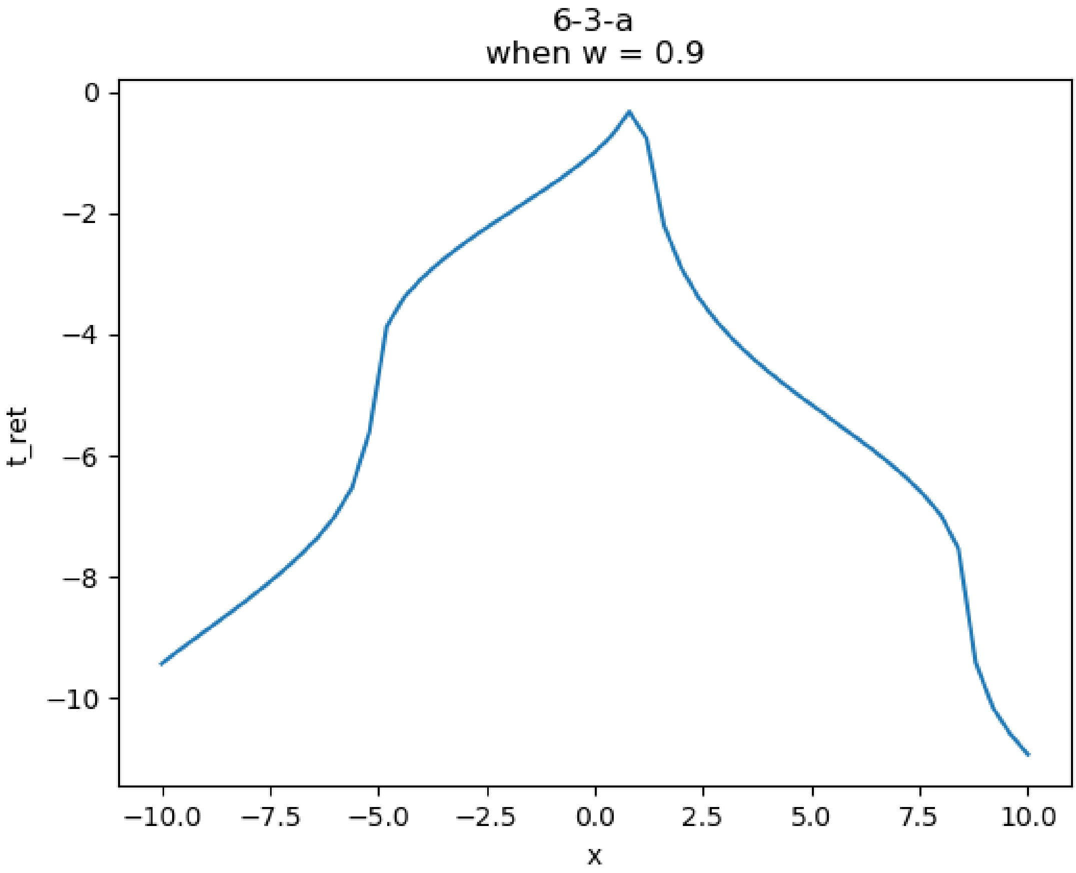
Code:

```
import matplotlib.pyplot as plt
import numpy as np
import math
import pdb

acc = 0.01 #In the second question,the answer of accuracy are 0.01

def delta(x,y,w,gu): #Newton method; This function is create the delta
    s = math.sin(w*gu)
    c = math.cos(w*gu)
    ans = (pow(gu,2)+2*x*c+2*y*s-pow(x,2)-pow(y,2)-1)/(2*gu-2*w*x*s+2*w*y*c)
    return ans

def test_1():
    w=0.1
    degu = 0
    tret = np.zeros((51,51))
    xx = []
    for idx in range(51): #create the grid from x=-10~10
        x = -10+0.4*idx
        xx.append(x)
        for jdx in range(51): #create the grid from y=-10~10
            y = -10+0.4*jdx
            gu = -10
            i = 1
            while i >=0:
                gu-= degu
```

```
                temp = delta(x, y, w, gu)
                if abs(temp)>acc: #check delta accuracy
                    degu = temp
                else:
                    i = -1
            tret[idx, jdx] = gu
    plt.plot(xx,tret[:,25])
    plt.title("6-3-a\nwhen w = 0.1")
    plt.xlabel("x")
    plt.ylabel("t_ret")
    plt.show()
test_1()

def test_2():
    w=0.9
    degu = 0
    tret = np.zeros((51,51))
    xx = []
    for idx in range(51):
        x = -10+0.4*idx
        xx.append(x)
        for jdx in range(51):
            y = -10+0.4*jdx
            gu = -10
            i = 1
            while i >=0:
                gu-= degu
                temp = delta(x, y, w, gu)
                if abs(temp)>acc:
                    degu = temp
                else:
                    i = -1
            tret[idx,jdx] = gu
    plt.plot(xx,tret[:,25])
    plt.title("6-3-a\nwhen w = 0.9")
    plt.xlabel("x")
    plt.ylabel("t_ret")
    plt.show()
test_2()
```

Result:


6-3-a
when w = 0.1
t_ret
x
0
−2
−4
−6
−8
−10
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


6-3-a
when w = 0.9
t_ret
x
0
−2
−4
−6
−8
−10
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

(b) Finding the potential

The retarded potential is like this:

$$A_\mu(x) = -\frac{u_\mu(t_{ret})}{R \cdot u(t_{ret})}$$

$$u_\mu(t_{ret}) = \frac{1}{\gamma}\left(\frac{\partial \overrightarrow{r_q}}{\partial t}\right)_{t=t_{ret}}$$

$$= (1-\omega^2)(1, -\omega\, sin(\omega t_{ret}), \omega\, cos(\omega t_{ret}))$$

$$R \cdot u(t_{ret}) = (1-\omega^2)(t_{ret} - r_x \omega \sin(\omega t_{ret}) + r_y \omega \cos(\omega t_{ret}))$$

Therefore,

$$A_\mu(x) = \frac{(-1, \omega\, sin(\omega t_{ret}), -\omega\, cos(\omega t_{ret}))}{t_{ret} - r_x \omega \sin(\omega t_{ret}) + r_y \omega \cos(\omega t_{ret})}$$

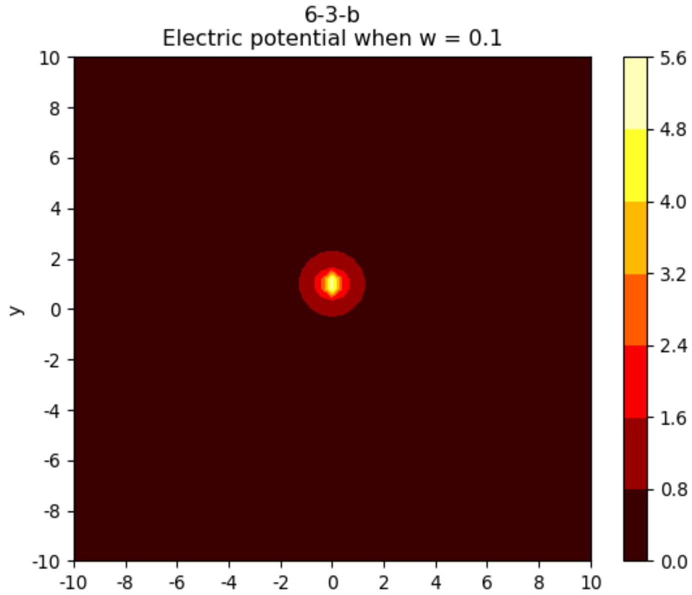
Code:

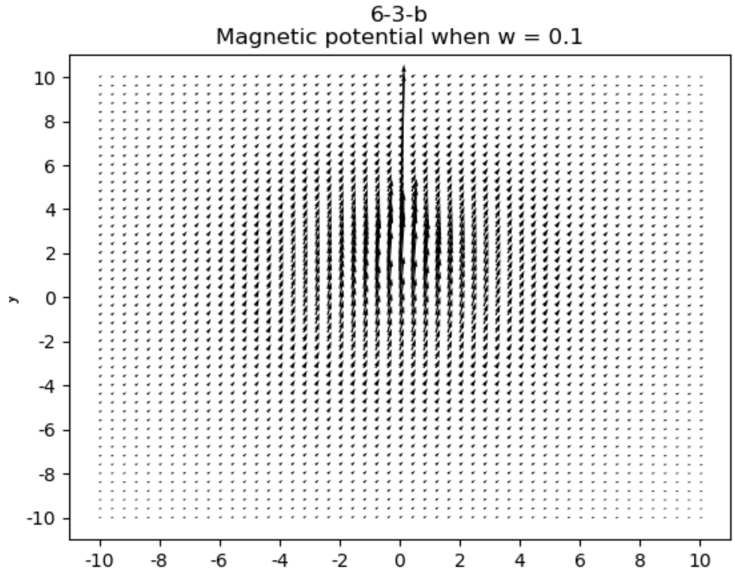
```
def test_3():
    w = 0.1
    degu = 0
    A0 = np.zeros((51,51))
    Ax = np.zeros((51,51))
    Ay = np.zeros((51,51))
    for idx in range(51): #create the grid from x=-10~10
        x = -10 + 0.4 * idx
        for jdx in range(51): #create the grid from y=-10~10
            y = -10 + 0.4 * jdx
            gu = -10
            i = 1
            while i >= 0:
                gu -= degu
                temp = delta(x, y, w, gu)
                if abs(temp) > acc:
                    degu = temp
                else:
                    i = -1
            A0[idx, jdx] = potential(x, y, w, gu, 0)
            Ax[idx, jdx] = potential(x, y, w, gu, 1)
            Ay[idx, jdx] = potential(x, y, w, gu, 2)

    bb = plt.contourf(A0, cmap=plt.cm.hot)
    plt.colorbar(bb, orientation="vertical")
    plt.xticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.yticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.title("6-3-b\nElectric potential when w = 0.1")
    plt.xlabel("x")
```

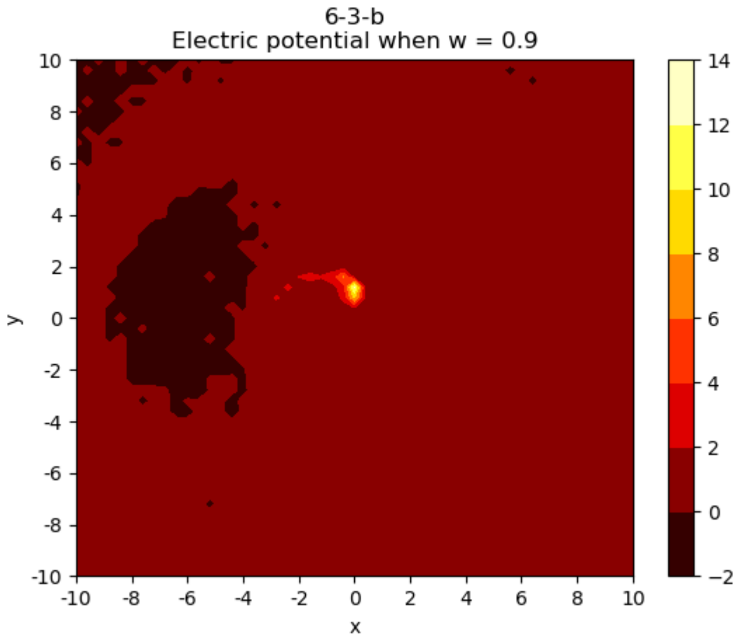
```python
    plt.ylabel("y")
    plt.show()
    plt.quiver(Ax,Ay)
    plt.xticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.yticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.title("6-3-b\nMagnetic potential when w = 0.1")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.show()
test_3()

def test_4():
    w = 0.9
    degu = 0
    A0 = np.zeros((51,51))
    Ax = np.zeros((51,51))
    Ay = np.zeros((51,51))
    for idx in range(51):
        x = -10 + 0.4 * idx
        for jdx in range(51):
            y = -10 + 0.4 * jdx
            gu = -10
            i = 1
            while i >= 0:
                gu -= degu
                temp = delta(x, y, w, gu)
                if abs(temp) > acc:
                    degu = temp
                else:
                    i = -1
            A0[idx, jdx] = potential(x, y, w, gu, 0)
            Ax[idx, jdx] = potential(x, y, w, gu, 1)
            Ay[idx, jdx] = potential(x, y, w, gu, 2)

    bb = plt.contourf(A0, cmap=plt.cm.hot)
    plt.colorbar(bb, orientation="vertical")
    plt.xticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.yticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.title("6-3-b\nElectric potential when w = 0.9")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.show()
```

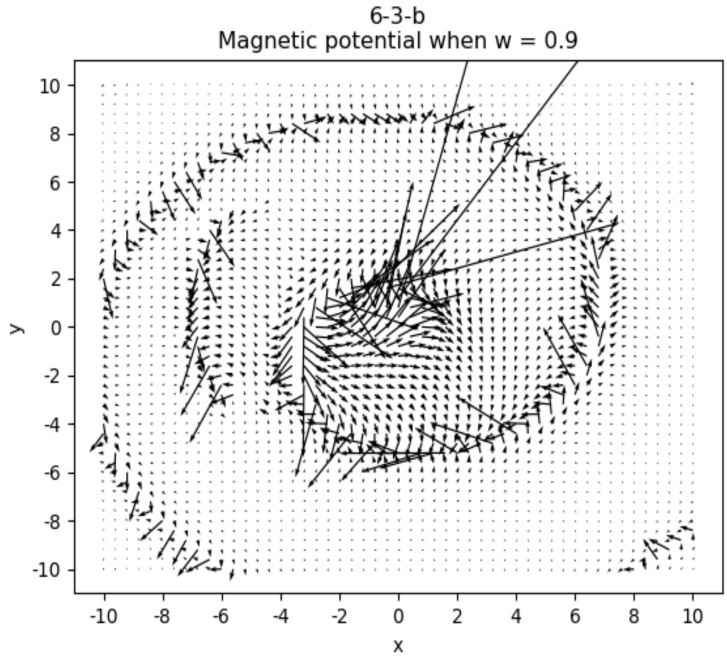
```
    plt.quiver(Ax,Ay)
    plt.xticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.yticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.title("6-3-b\nMagnetic potential when w = 0.9")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.show()
test_4()
```

Result:









(c).Calculating retarded electric field

Retarded electric field

$$\vec{E} = \underbrace{\left(\frac{1}{\gamma^2}\frac{(\hat{R}-\beta)}{R^2\left(1-\beta\cdot\hat{R}\right)^3}\right)}_{Comlomb\,(near\ field)} + \underbrace{\frac{\hat{R}\times\left[\left(\hat{R}-\beta\right)\times\dot{\beta}\right]}{|R|\left(1-\beta\cdot\hat{R}\right)^3}}_{Radiation\,(far\ field)}$$

$$= \frac{1}{\gamma^2}\frac{(\hat{R}-\beta)}{R^2\left(1-\beta\cdot\hat{R}\right)^3} + \frac{(\hat{R}\cdot\dot{\beta})(\hat{R}-\beta)}{|R|\left(1-\beta\cdot\hat{R}\right)^3} - \frac{\dot{\beta}}{|R|\left(1-\beta\cdot\hat{R}\right)^2}$$

And we know that $|R| = t_{ret}, \frac{1}{\gamma^2} = 1-\omega^2, \beta = \gamma u.$

So,

$$\hat{R} = \frac{R}{|R|} = \left(\frac{r_x - cos(\omega t_{ret})}{t_{ret}}, \frac{r_y - sin(\omega t_{ret})}{t_{ret}}\right)$$

$$\dot{\beta} = \left(-\omega^2\, cos(\omega t_{ret}), -\omega^2\, sin(\omega t_{ret})\right)$$

$$1-\beta\cdot\hat{R} = 1 - \frac{\omega}{t_{ret}}\left(-r_x\, sin(\omega t_{ret}) + r_y\, cos(\omega t_{ret})\right)$$

$$\hat{R}-\beta = \left(\frac{r_x - cos(\omega t_{ret})}{t_{ret}} - \omega\sin(\omega t_{ret}), \frac{r_y - sin(\omega t_{ret})}{t_{ret}} + \omega\cos(\omega t_{ret})\right)$$

Therefore, Electric field is following:

$$E_x = \frac{(1-\omega^2)\left(\frac{r_x - cos(\omega t_{ret})}{t_{ret}} - \omega\, sin(\omega t_{ret})\right)}{t_{ret}^2\left(1-\frac{\omega}{t_{ret}}\left(-r_x\, sin(\omega t_{ret}) + r_y\, cos(\omega t_{ret})\right)\right)^3}$$
$$+ \frac{\left(\frac{\omega}{t_{ret}}\left(-r_x\, sin(\omega t_{ret}) + r_y\, cos(\omega t_{ret})\right)\right)\left(\frac{r_x - cos(\omega t_{ret})}{t_{ret}} - \omega\, sin(\omega t_{ret})\right)}{t_{ret}\left(1-\frac{\omega}{t_{ret}}\left(-r_x\, sin(\omega t_{ret}) + r_y\, cos(\omega t_{ret})\right)\right)^3}$$
$$+ \frac{\omega^2\, cos(\omega t_{ret})}{t_{ret}\left(1-\frac{\omega}{t_{ret}}\left(-r_x\, sin(\omega t_{ret}) + r_y\, cos(\omega t_{ret})\right)\right)^2}$$

$$E_y = \frac{(1-\omega^2)\left(\frac{r_y - sin(\omega t_{ret})}{t_{ret}} + \omega\cos(\omega t_{ret})\right)}{t_{ret}^2\left(1-\frac{\omega}{t_{ret}}\left(-r_x\, sin(\omega t_{ret}) + r_y\, cos(\omega t_{ret})\right)\right)^3}$$
$$+ \frac{\left(\frac{\omega}{t_{ret}}\left(-r_x\, sin(\omega t_{ret}) + r_y\, cos(\omega t_{ret})\right)\right)\left(\frac{r_y - sin(\omega t_{ret})}{t_{ret}} + \omega\cos(\omega t_{ret})\right)}{t_{ret}\left(1-\frac{\omega}{t_{ret}}\left(-r_x\, sin(\omega t_{ret}) + r_y\, cos(\omega t_{ret})\right)\right)^3}$$
$$+ \frac{\omega^2\, sin(\omega t_{ret})}{t_{ret}\left(1-\frac{\omega}{t_{ret}}\left(-r_x\, sin(\omega t_{ret}) + r_y\, cos(\omega t_{ret})\right)\right)^2}$$

Code:

```
def Efield(x,y,w,tret,mu,fi): #calculate electric field
    c = math.cos(w * tret)
    s = math.sin(w * tret)
    a = w*(c*y-x*s)/tret
    if mu == 1: #Ex
        m = ((x-c)/tret)+w*s
        if fi == 1: #near field
            ans = (((1-pow(w,2))*m)/(pow(tret,2)*pow((1-a),3)))
        else: #far field
            ans =(a * m / (tret * pow((1 - a), 3))) + (pow(w, 2) * c / (tret * pow((1 - a), 2)))
    elif mu == 2: #Ey
        m = ((y-s)/tret)-w*c
        if fi ==1: #near field
            ans = (((1 - pow(w, 2)) * m) / (pow(tret, 2) * pow((1 - a), 3)))
        else: #far field
            ans = (a * m / (tret * pow((1 - a), 3))) + (pow(w, 2) * s / (tret * pow((1 - a), 2)))
    else:
        print("error")
        pdb.set_trace()
    return ans


def test_5():
    w = 0.1
    degu = 0
    Exn = np.zeros((51, 51))
    Exf = np.zeros((51, 51))
    Eyn = np.zeros((51, 51))
    Eyf = np.zeros((51, 51))
    for idx in range(51): #grid x=-10~10
        x = -10 + 0.4 * idx
        for jdx in range(51): #grid y=-10~10
            y = -10 + 0.4 * jdx
            gu = -10
            i = 1
            while i >= 0:
                gu -= degu
                temp = delta(x, y, w, gu)
                if abs(temp) > acc: #check the accuracy of delta
                    degu = temp
                else:
                    i = -1
```

```python
            tempxn = Efield(x, y, w, gu, 1,1)
            tempxf = Efield(x, y, w, gu, 1, 2)
            tempyn = Efield(x, y, w, gu, 2,1)
            tempyf = Efield(x, y, w, gu, 2, 2)
            Exn[idx,jdx] = tempxn
            Exf[idx, jdx] = tempxf
            Eyn[idx, jdx] = tempyn
            Eyf[idx, jdx] = tempyf

    plt.quiver(Exn, Eyn)
    plt.xticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.yticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.title("6-3-c\nElectric near field in w = 0.1")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.show()
    plt.quiver(Exf, Eyf)
    plt.xticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.yticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.title("6-3-c\nElectric far field in w = 0.1")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.show()

test_5()

def test_6():
    w = 0.9
    degu = 0
    Exn = np.zeros((51, 51))
    Exf = np.zeros((51, 51))
    Eyn = np.zeros((51, 51))
    Eyf = np.zeros((51, 51))
    for idx in range(51):
        x = -10 + 0.4 * idx
        for jdx in range(51):
            y = -10 + 0.4 * jdx
            gu = -10
            i = 1
            while i >= 0:
                gu -= degu
                temp = delta(x, y, w, gu)
```

```
            if abs(temp) > acc:
                degu = temp
            else:
                i = -1
        tempxn = Efield(x, y, w, gu, 1, 1)
        tempxf = Efield(x, y, w, gu, 1, 2)
        tempyn = Efield(x, y, w, gu, 2, 1)
        tempyf = Efield(x, y, w, gu, 2, 2)
        Exn[idx, jdx] = tempxn
        Exf[idx, jdx] = tempxf
        Eyn[idx, jdx] = tempyn
        Eyf[idx, jdx] = tempyf

    plt.quiver(Exn, Eyn)
    plt.xticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.yticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.title("6-3-c\nElectric near field in w = 0.9")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.show()
    plt.quiver(Exf, Eyf)
    plt.xticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.yticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50], [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10])
    plt.title("6-3-c\nElectric far field in w = 0.9")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.show()

test_6()
```

Result:

We can see that near field is just Coulomb part, so it will like a charge in the middle, and because Coulomb part is $\sim\frac{1}{r^2}$, it vanished very quickly. Far field is radiation part, so it seems like a dipole in the middle, and also because radiation part is $\sim\frac{1}{r}$, it will seems like affect all the space.