计算物理第一次作业

信息

- 韩懿杰
- 20213006413

解释说明

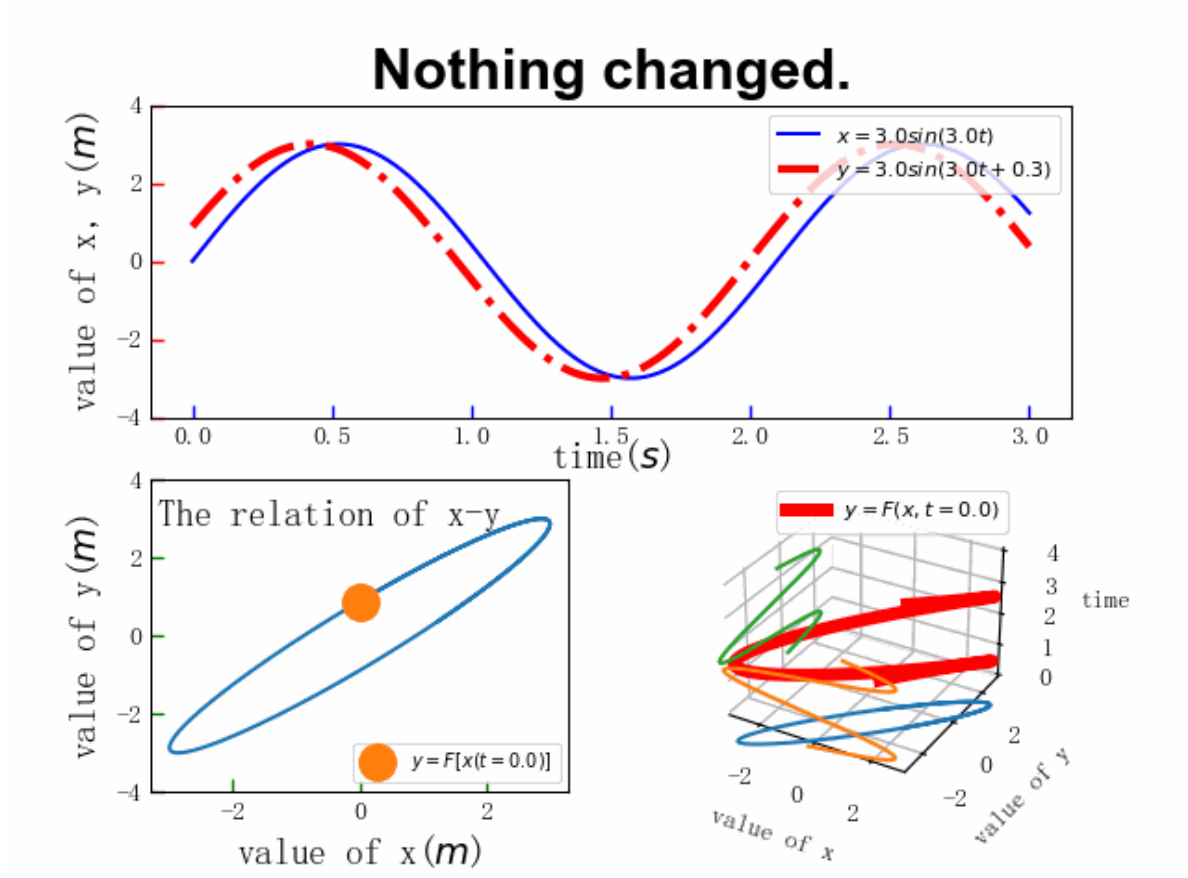
- 所演示函数为：

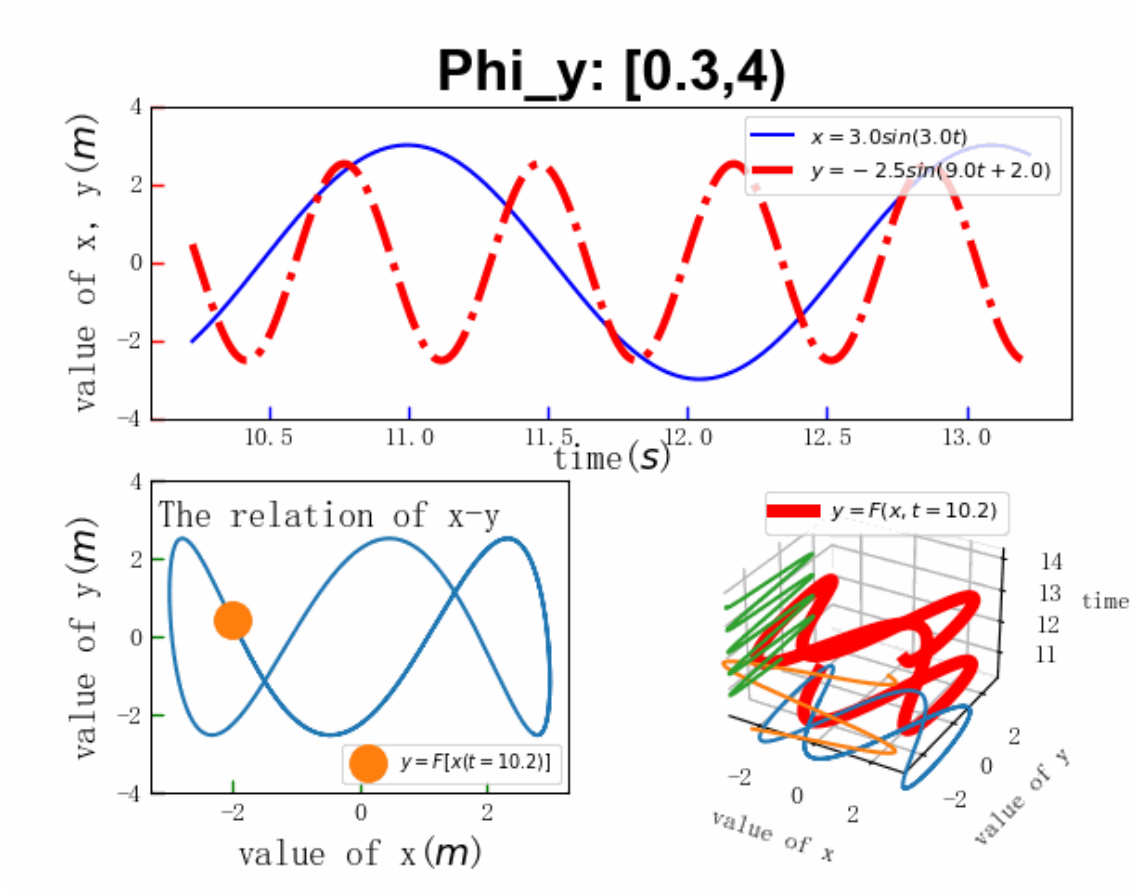
$$
\begin{aligned}
y_1 &= A_1 sin(w_1 \cdot t) \\
y_2 &= A_2 sin(w_2 \cdot t + \phi)
\end{aligned} \tag{1}
$$

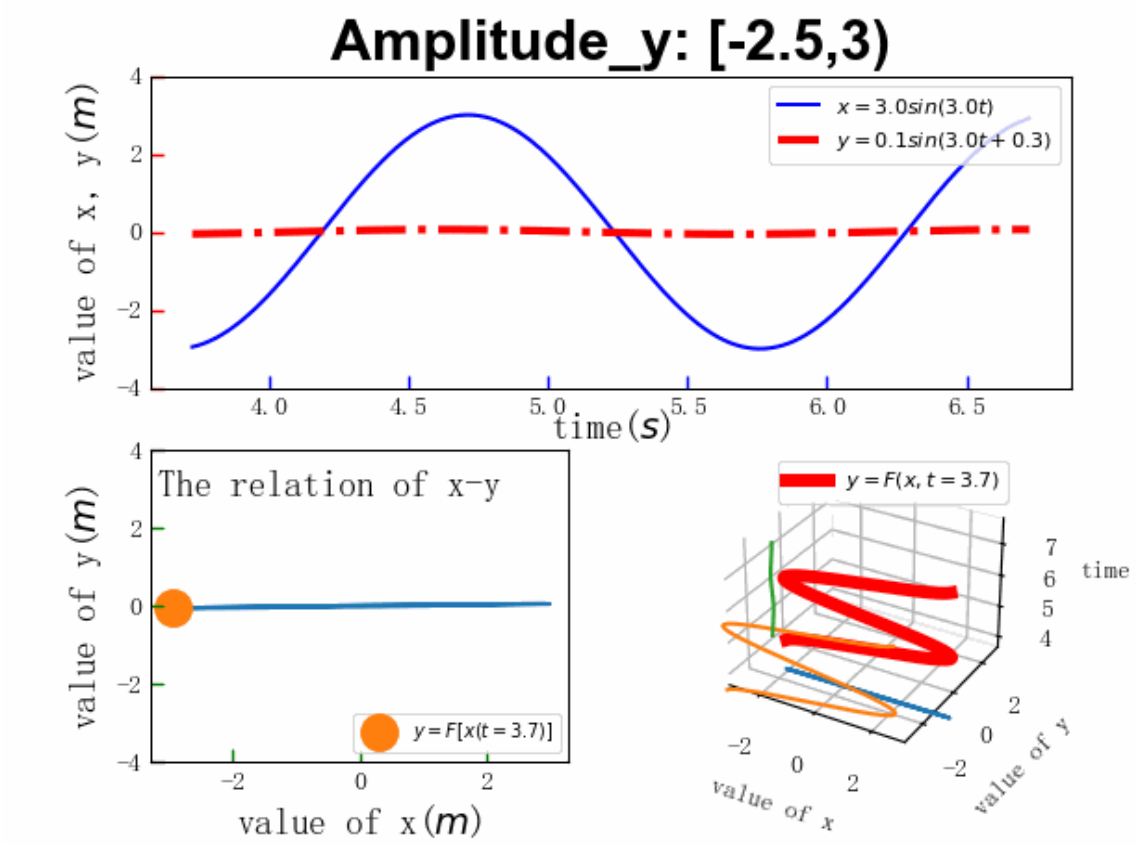
- 参数随时演化，观察x-t,y-t,x-y轨迹：
  - step 1: $A_1 = A_2 = 3, w_1 = w_2 = 0.3, \phi = 0.3$，由于w相同，轨迹为椭圆
  - step 2: 改变$y2$振幅A_2， 轨迹仍为椭圆，离心率和椭圆偏向有所改变。
  - step 3: 改变w_2, 轨迹为花环形状，花环做周期旋转运动， 且因为w_2持续变化，花环不闭合。
  - step 4: 改变ϕ，由于w固定，花环闭环。ϕ的改变对花环形状和周期旋转无改变，但对指定时间的状态在花环上的位置有影响。

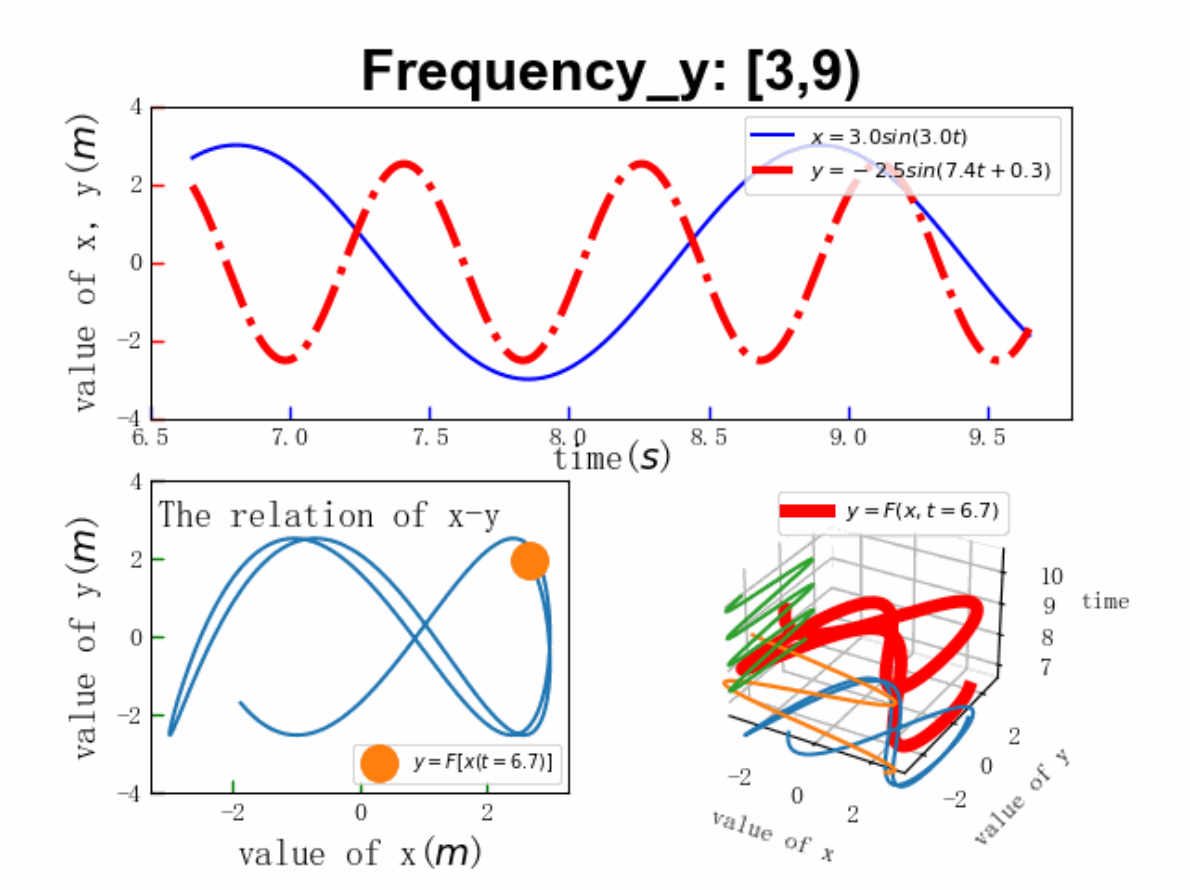
图片演示

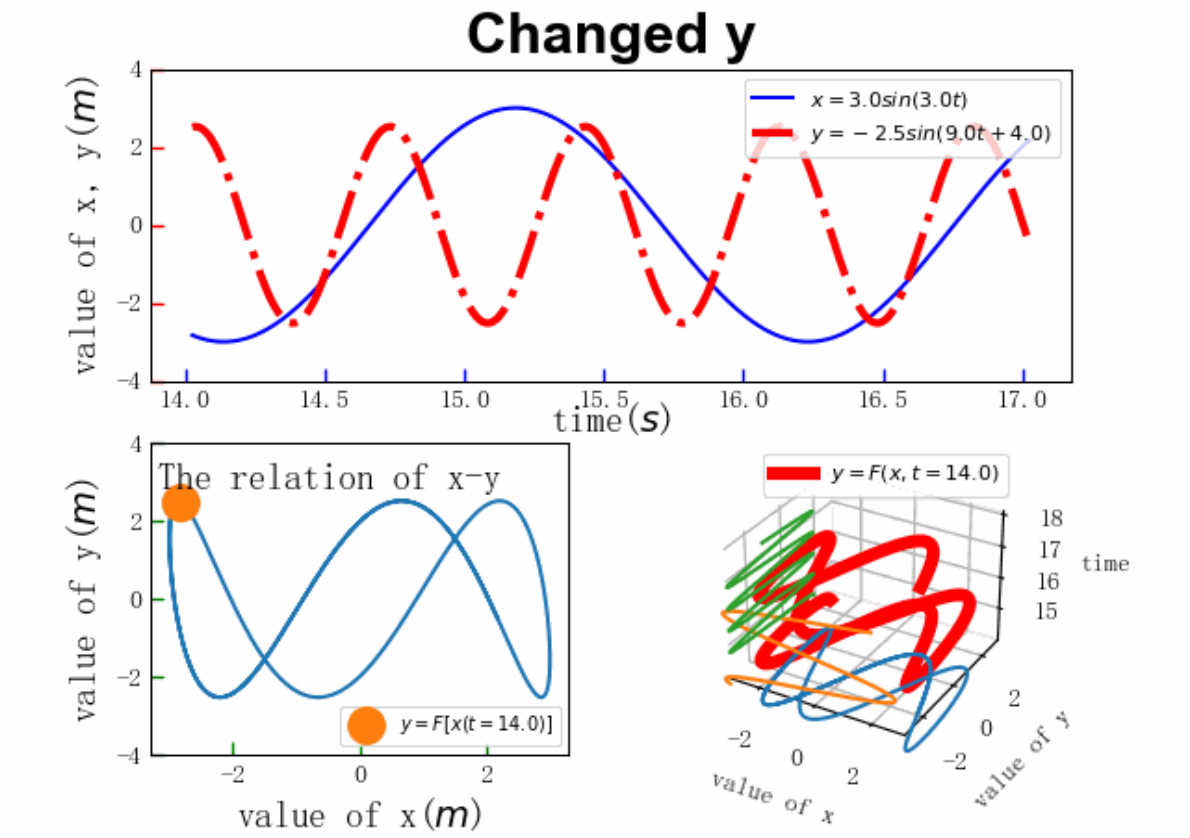
显然动图放不进PDF，任选若干张以作示范。




Phi_y: [0.3,4)
$x = 3.0sin(3.0t)$
$y = -2.5sin(9.0t + 2.0)$
value of x, y(m)
time(s)
The relation of x-y
value of y(m)
value of x(m)
$y = F[x(t = 10.2)]$
$y = F(x, t = 10.2)$
time
value of x
value of y


Amplitude_y: [-2.5,3)
$x = 3.0sin(3.0t)$
$y = 0.1sin(3.0t + 0.3)$
value of x, y(m)
time(s)
The relation of x-y
value of y(m)
value of x(m)
$y = F[x(t = 3.7)]$
$y = F(x, t = 3.7)$
time
value of x
value of y





代码

- 注意，输出已被注释
- 运行代码下载，复制会格式紊乱：https://github.com/hanyijie1/Homework_of_computate_physics

```
#Library
## system function
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
## Self-function
'''
This function is used to undate and plot;
input: iterations, A,w,\phi,original time t.
Output: None
'''
def update(n,A,w,phi,t):
    #Definition of variable
    A_=A
    w_=w
    phi_=phi
    #clear
    ax[0].clear()
    ax[1].clear()
    ax3.clear()
    #Updata
    ## title
    ax[1].set_title("The relation of x-y",\

 fontname='FangSong',fontsize=14,weight='bold',x=0.43,y=0.8)
    if n<100:
        ax[0].set_title("Nothing
changed.",fontname='Arial',fontsize=22,weight='bold')
    elif n<193:
        A_=-A*(n-150)/50
        ax[0].set_title("Amplitude_y:
[-2.5,3)",fontname='Arial',fontsize=22,weight='bold')
    elif n<294:
        A_=-A*(192-150)/50
        w_=w+w*(n-193)/50
        ax[0].set_title("Frequency_y:
[3,9)",fontname='Arial',fontsize=22,weight='bold')
    elif n<544:
        A_=-A*(192-150)/50
        w_=w+w*(293-193)/50
        phi_=phi+phi*(n-293)/20
        ax[0].set_title("Phi_y:
[0.3,4)",fontname='Arial',fontsize=22,weight='bold')
    else:
        A_=-A*(192-150)/50
        w_=w+w*(293-193)/50
        phi_=phi+phi*(544-294)/20
        ax[0].set_title("Changed
y",fontname='Arial',fontsize=22,weight='bold')
    ## date
    t_up=t+n/40
    vary1=A*np.sin(w*(t_up))
    vary2=A_*np.sin(w_*(t_up)+phi_)
    ## plot
    ax[0].plot(t_up, vary1,\
```

```
        color='b',linewidth=1.5,linestyle='-',\
            label='$x={0:.1f}sin({1:.1f}t)$'.format(A,w),zorder=1)
    ax[0].plot(t_up, vary2,color='r',linestyle='-.',linewidth=3,\
            label='$y={0:.1f}sin({1:.1f}t+{2:.1f})$'.format(A_,w_,phi_))
    ax[1].plot(vary1,vary2,linestyle='-',zorder=0.5)
    ax[1].plot(vary1[0], vary2[0],linestyle='',label='$y=F[x(t=
{0:.1f})]$'.format(t_up[0]),\
                marker='o',markersize=14,zorder=1)
    ax3.plot(vary1,vary2,t_up,color='r',linewidth=5,\
             label='$y=F(x,t={:.1f})$'.format(t_up[0]))
    ax3.plot(vary1,vary2,t_up[0]-1) #y-x投影
    ax3.plot(vary1,np.zeros(1000)-A-0.5,t_up) #x-t
    ax3.plot(np.zeros(1000)-A-0.5,vary2,t_up) #y-t投影
    # legend
    ax[0].legend(loc='upper right',prop = {'size':8})
    ax[1].legend(loc='lower right',prop = {'size':7})
    ax3.legend(loc='upper right',prop = {'size':8})
    # label
    ax[0].set_xlabel("time($s$)",fontsize=14,labelpad=-4)
    ax[0].set_ylabel("value of x, y($m$)",fontsize=14)
    ax[1].set_xlabel("value of x($m$)",fontsize=14)
    ax[1].set_ylabel("value of y($m$)",fontsize=14)
    ax3.set_xlabel('value of x')
    ax3.set_ylabel('value of y')
    ax3.set_zlabel('time')
    # axis
    ## tick

 ax[0].tick_params(axis='both',direction='in',color='blue',length=5,width=1
)
    ax[0].tick_params(axis='y',direction='in',color='red',length=5,width=1)

 ax[1].tick_params(axis='both',direction='in',color='green',length=5,width=
1)
    ## limit
    ax[1].set_ylim([-A-1,A+1])
    ax[0].set_ylim([-A-1,A+1])
    ax3.set_zlim([t_up[0],t_up[999]+1])
    ax3.set_xlim([-A,A])
    ax3.set_ylim([-A,A])
#---------------------------The following is the text----------------------
----#

# The parameter to input
A=3
w=3
phi=0.3
# Original Data
t = np.linspace(0, 3, 1000) #The lastest time range to display
# Graph
## figure
ax=[0,0]
fig = plt.figure()
ax[0] = fig.add_subplot(2,1,1)
ax[1]= fig.add_subplot(2,2,3)
```

```
ax3 = fig.add_subplot(2,2,4, projection='3d')
## plot
ani = FuncAnimation(fig, update,fargs=(A, w, phi,t),\
                    frames=700, interval=20, blit=False, repeat=True)
# Output
plt.show()
#ani.save('computational_physics.gif',writer='pillow',fps=24)
#plt.savefig('savefig_example.eps')
```