EPS 136A LAB 1

2021-01-05

Lingsen Meng and Tian Feng

1) Zoom

TA's Zoom ID is 846 868 2158, please join the Zoom meeting. Test your microphone and camera.

2) Anaconda Installation

Check website: https://docs.anaconda.com/anaconda/install/

Or https://www.anaconda.com/products/individual to download Individual Edition (Python 3.7 version).

3) Instruction on Jupyter Notebook

Open terminal (Linux/OS) and type 'Jupyter notebook' or open Jupyter Notebook desktop launcher (Windows)

If you had problems to start the Jupyter notebook, you can reinstall Jupyter with the reference https://test-jupyter.readthedocs.io/en/latest/install.html

After successfully starting Jupyter Notebook, you can get familiar with the JupyterLab Interface: https://jupyterlab.readthedocs.io/en/stable/user/interface.html

Meanwhile, try to create a new notebook with the button 'New' on the upper right corner, and you will find a .ipynb file in the current folder.

4) Install Dependent Packages

Open terminal and type the following commands
‘pip install obspy’
‘pip install keras’
‘pip install tensorflow’
‘pip install sklearn’
‘pip install numpy’
‘pip install pandas’
‘pip install scipy’
You can change the ‘pip’ into ‘pip3’ or ‘conda’ if had any errors.

For Windows users, open the Anaconda Prompt. For more details, check the following link https://datatofish.com/how-to-install-python-package-in-anaconda/

5) Run Test notebook

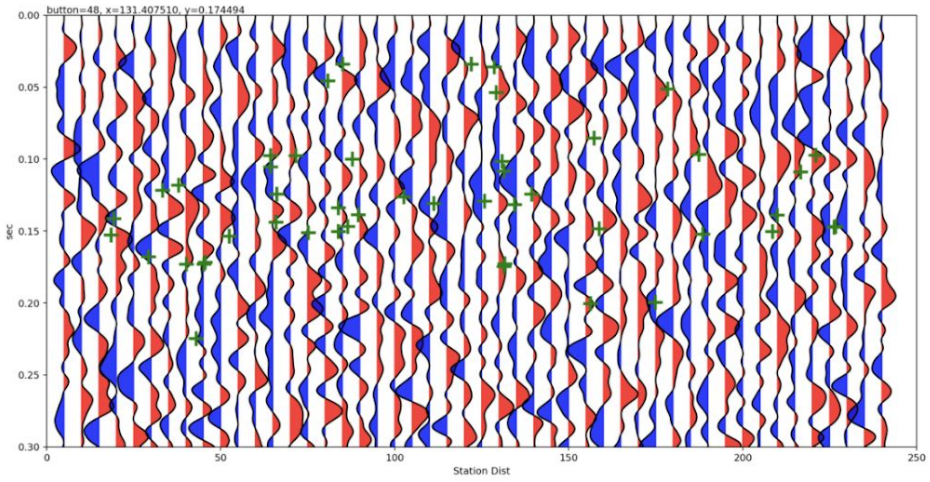
Open test.ipynb file, and run cells. Make sure the code is running without errors. Make sure you can click on the figures, and draw green crosses freely.
If everything is good, you will the output would like this:

In [12]:

```
import numpy as np
import matplotlib.pyplot as plt
import random

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

import matplotlib
import scipy.stats as stats
from obspy.signal.trigger import trigger_onset

# sklearn packages
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix, precision_recall_curve, roc_curve

# keras packages
from keras import backend as K
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.models import Sequential, Model
from keras.layers import Input, Conv1D, MaxPooling1D, UpSampling1D,Flatten,Dense,Dropout,BatchNormalization
from keras.utils import np_utils
from keras.optimizers import Adam

import math
from obspy.geodetics.base import gps2dist_azimuth
from scipy.optimize import curve_fit

import pandas as pd, numpy as np, matplotlib.pyplot as plt
from obspy.signal.filter import bandpass
from obspy.signal.detrend import polynomial
import math
from scipy.interpolate import interp1d

%matplotlib notebook

print('success')
```

```
success
```

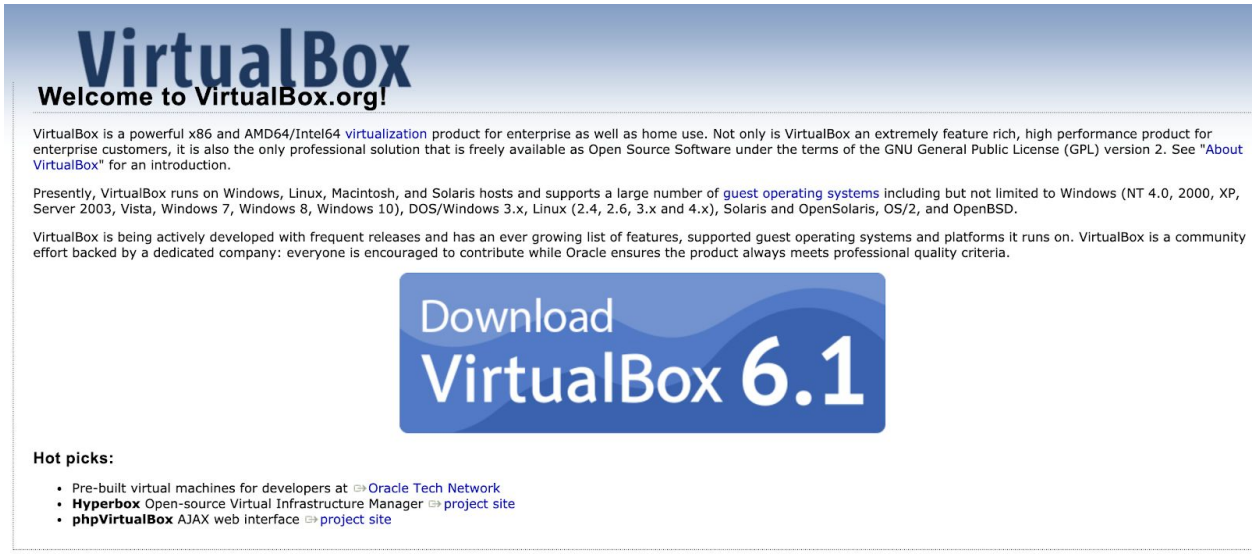
In [9]:
```
nostations = 48
nodata = 3000
dt = 0.001
dx = 5
seis1 = np.random.rand(nodata,nostations)

def plotseis(seis,doublecolor=False):
    fig =plt.figure(figsize=(16,8))
    ax = fig.add_subplot(111)
    for i in range(nostations):
    #     tr.normalize()
        tr=seis[:nodata,i]
    #     tr = minmax_scale(tr, feature_range=(-1,1))
        tr=polynomial(tr, order=3, plot=False)
        tr=bandpass(tr,0.1,40,1/dt)
        tr=tr/tr.max()*5
        dist = dx*i+dx
        y = np.arange(nodata)*dt
        x = tr+dist
        ax.plot(x,y,'k')
        ax.fill_betweenx(y,x,dist,x > dist, color='r', alpha = 0.8)
        if doublecolor:
            ax.fill_betweenx(y,x,dist,x < dist, color='b', alpha = 0.8)

    ax.set_xlabel("Station Dist")
    ax.set_ylabel("sec")
    ax.set_ylim(0,0.3)
    ax.set_xlim(0,250)
#     plt.tight_layout()
    plt.gca().invert_yaxis()

    return fig, ax

def onclick(event):
#     tx = 'button=%d, x=%d, y=%d, xdata=%f, ydata=%f' % (event.button, event.x, event.y, event.xdata, event.ydata)
    if len(picks) == nostations:
        np.save(filename,np.array(sorted(picks)))
        fig.canvas.mpl_disconnect(cid)
        return
    picks.append((event.xdata,event.ydata))
    tx = 'button=%d, x=%f, y=%f ' % (len(picks), event.xdata, event.ydata)
    text.set_text(tx)
    plt.plot(event.xdata,event.ydata,'g+',markersize=15,markeredgewidth=3)
    plt.show()

picks=[]
fig, ax =plotseis(seis1,doublecolor=True)
filename='tt1.npy'
text=ax.text(0,0, "", va="bottom", ha="left")
cid = fig.canvas.mpl_connect('button_press_event', onclick)
```



If you see the error like “ ModuleNotFoundError: No module named ‘xxxx’ “, open the terminal and type ‘pip install xxxx’ or ‘pip3 install xxxx’ or ‘conda install xxxx’ .

6) VirtualBox Setup

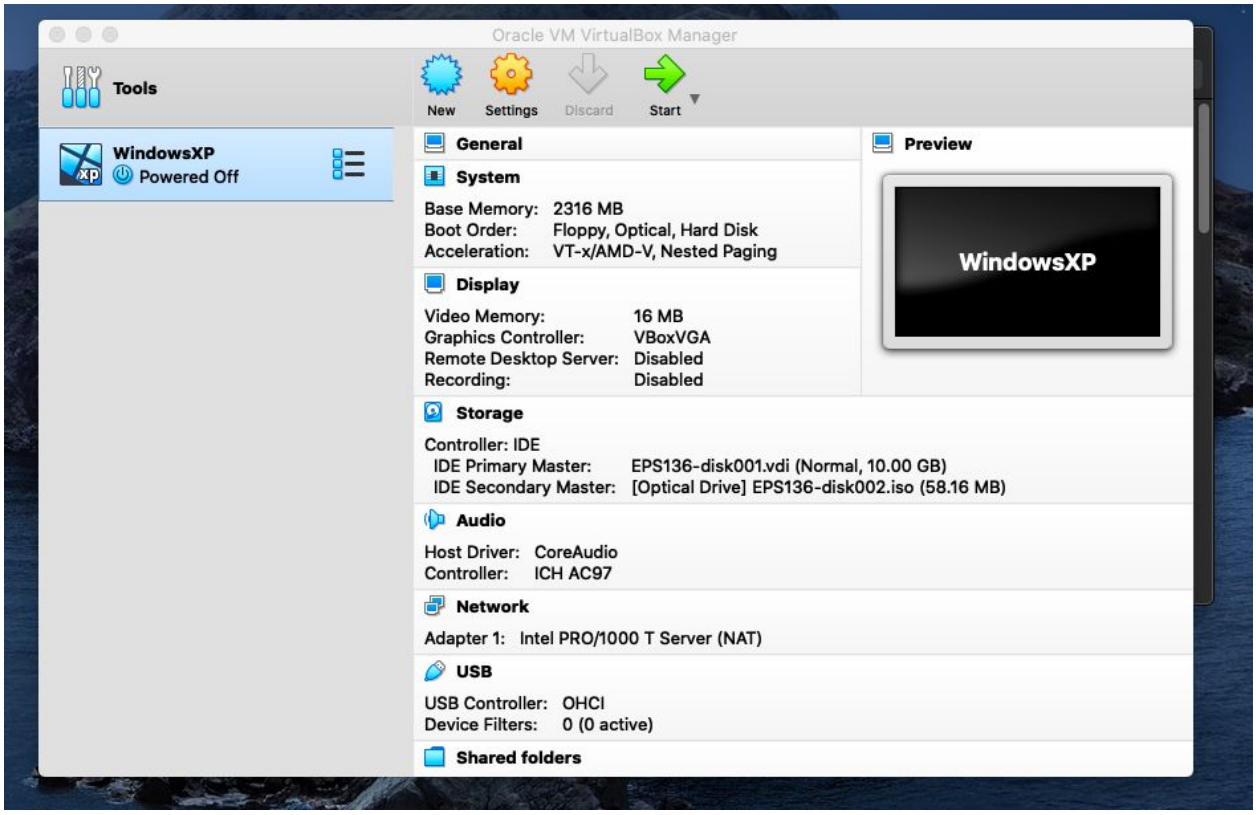
VirtualBox is a free cross-platform virtual machine (VM) software developed by Oracle. Please download and install VirtualBox 6.1 (https://www.virtualbox.org/) Choose the right variant of VirtualBox for your own operating system (Mac / Windows / Linux)



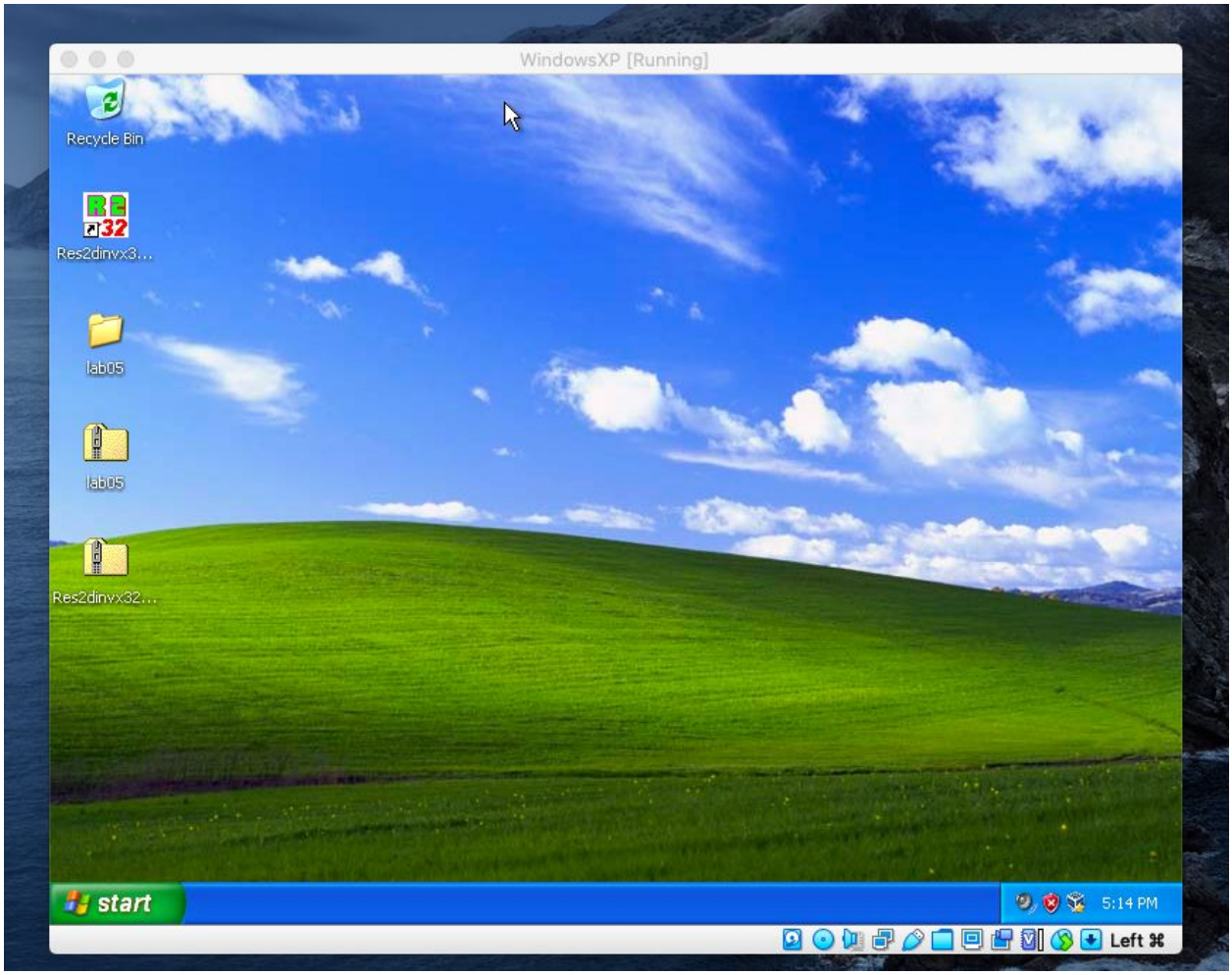
Download the class Linux system image(EPS136.ova) from CCLE.

Import the Linux system in VirtualBox: File->import appliance-> choose the file EPS136.ova (wherever it is downloaded)

It will take a few minutes to import the VM. The VM takes about 10 GB of disk space. Make sure you have enough resources.



If everything is set up correctly, you should see the Windows XP desktop in the VM. It has a software named Res2dinv installed as well as a folder named lab05.



You have now successfully set up the remote lab environment.

7) Taking Screen Shots

The best way to take screenshots for your lab report figures is to use the host OS's ability to do this.

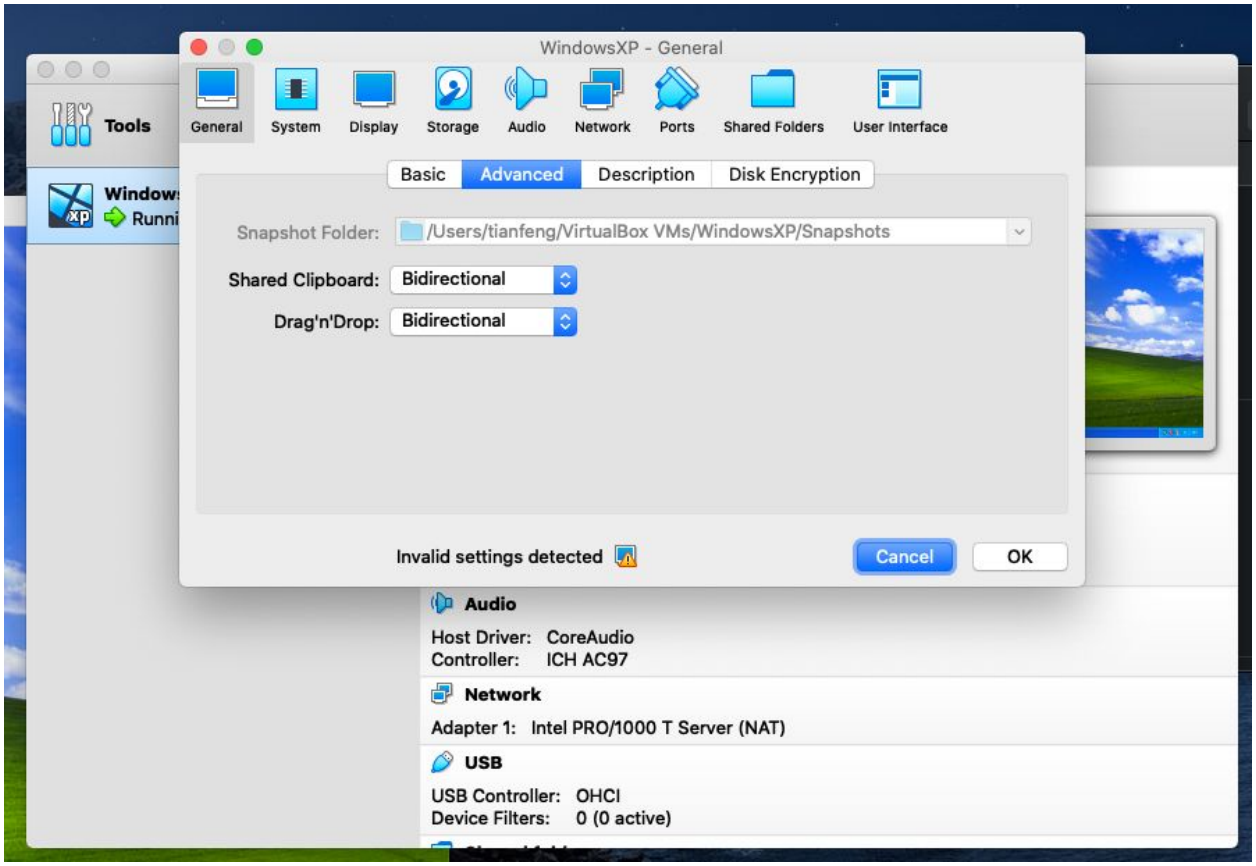
On a mac, hit Command+shift+4 for a selectable area of the screen or command+shift+3 for a fullscreen screenshot. (you might have to put the VirtualBox in the background and make sure no other programs are covering the VirtualBox screen to activate the screen shooting.)

On a Windows 10 machine, ctrl+shift+s will allow you to select an area to screenshot, and which is saved to the clipboard, or users can use the snipping tool. For taking

screenshots inside the VM, use Gnome Screenshot utility (you can hit the activity menu on the top-left corner open a search, then type “screenshot” to open the utility. This is the Linux equivalent of using the windows snipping tool.

8) File Transfer

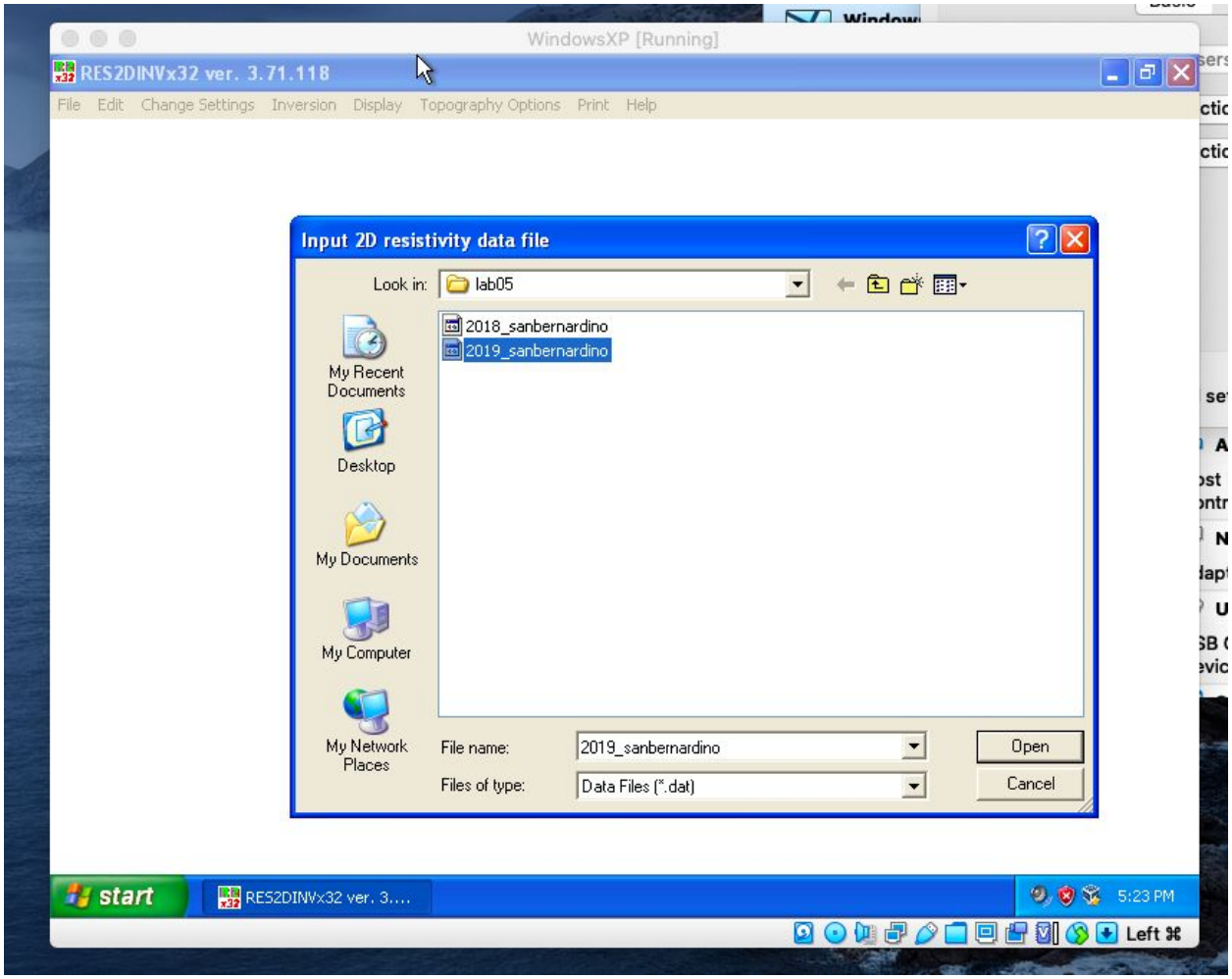
We can transfer files between host and guest machines in 3 ways (https://www.isunshare.com/blog/3-ways-to-transfer-files-between-windows-and-virtualbox/). The easiest way is to drag and drop. Make sure you select Bidirectional in the menu Devices >> Drag and Drop >> Bidirectional.



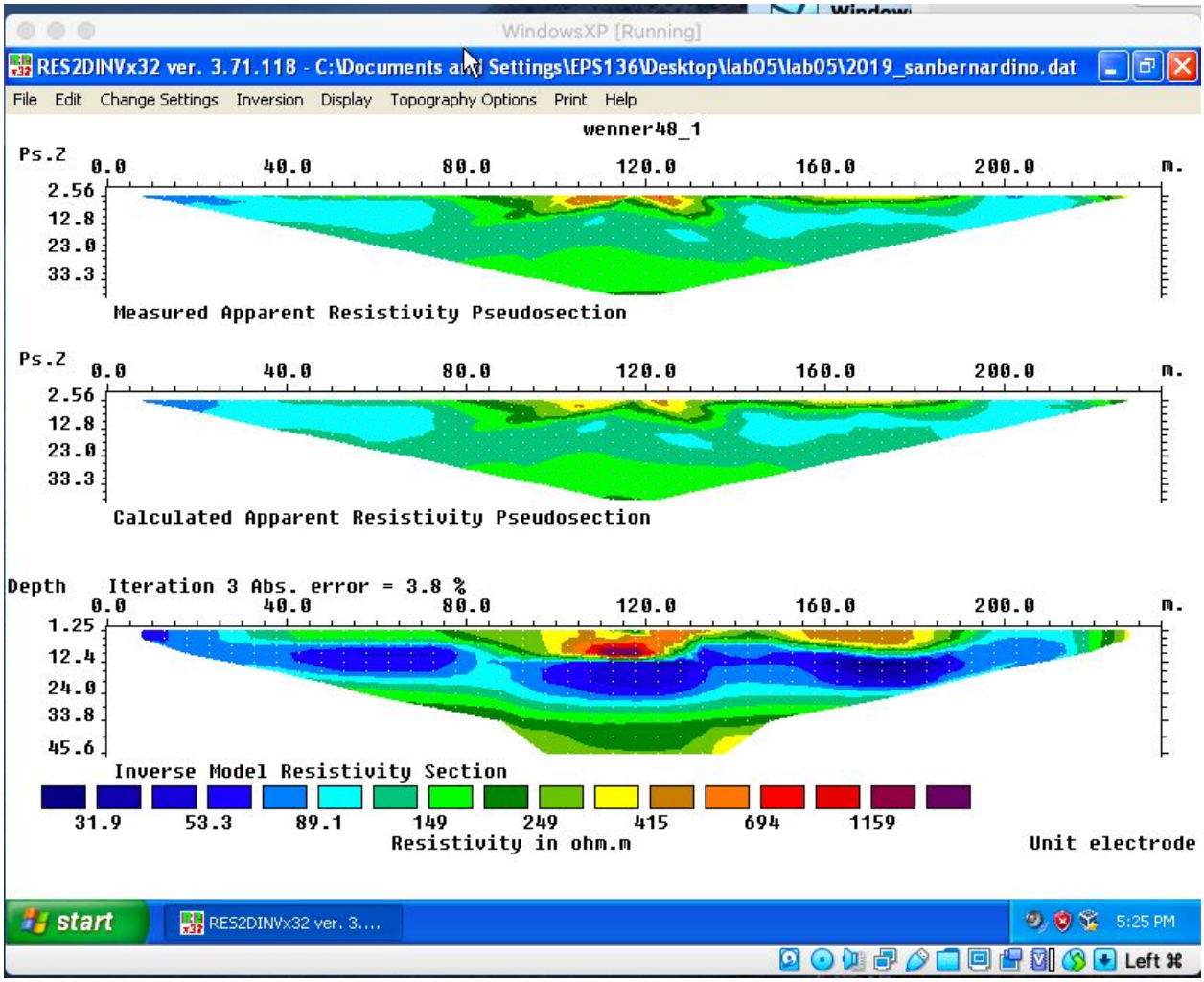
You can test dragging and dropping files between host and guest machines. For example, drag and drop the screenshots you made in step 8. If you had problems, try to click “Insert Guest Additions CD Image” in the “Devices” menu. It will help you install some extension programs. If you have some bugs during installation, just click continue. If the installation process is too slow, reboot, and try to reinstall.

9) Test software

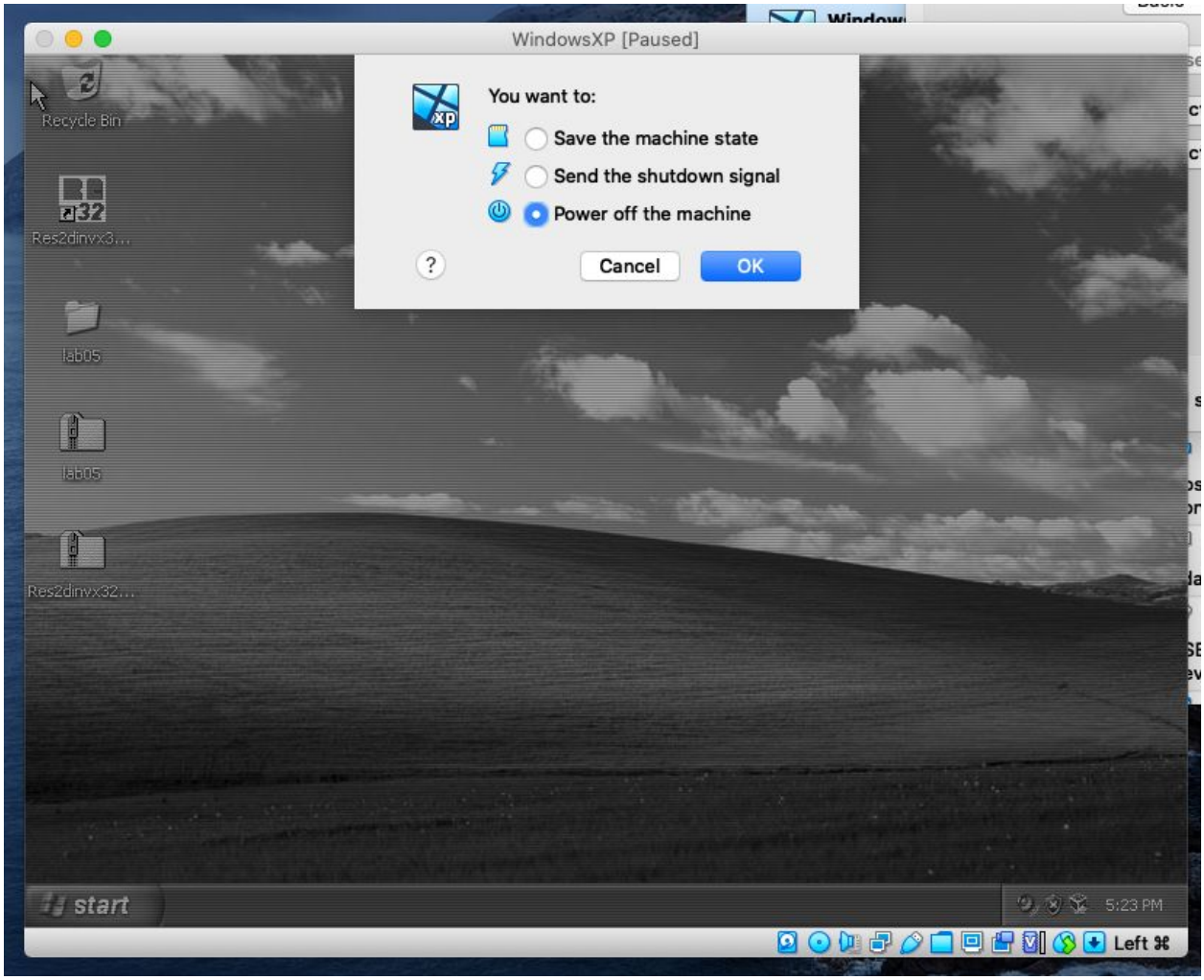
In this step, you will test the software. Open RES2DINV and select File >> Read data file, read any data in lab05.



Then select Inversion >> Carry out Inversion. You should see the output like the following figure.



Now, you have almost finished this lab. When you need to quit the VM, I find click the red button on the top-left, and choose “Power of the machine” is the most convenient.



10) Install Matlab

It's our first time to make labs in Python. Previously, all labs are written in Matlab. In case there are any unexpected problems or bugs, we hope every student can install Matlab.

Matlab is currently available for UCLA students to use free of charge. Follow the link below to get your Matlab https://softwarecentral.ucla.edu/matlab-getmatlab

You will be asked to create an account with Mathworks to be able to download, install, and activate Matlab.

Important! Make sure you also check the boxes of Signal Processing, Imaging, and Mapping toolboxes when installing Matlab.