Electronic Properties of a Two-Dimensional Electron Gas

Tianyi Zhou* and Fan Zhang
Department of Physics, Brown University
Providence, RI 02906, U.S.A
(Dated: December 12, 2017)

Two-dimensional electron gas (2DEG) can be formed on insulator-semiconductor interface by applying an external electric field to the system. We investigated a 2DEG formed in an n-channel MOSFET, measured the conductivity and its temperature dependence, studied the electrical transport properties of 2DEG in strong magnetic field, and determined the threshold voltage and the insulator thickness of the MOSFET. Quantum oscillation of conductivity was observed at the presence of $\sim$ 6T magnetic field. The threshold voltage is found to be (4.426 ± 0.425)V and the insulator thickness is (281.328 ± 2.747)nm. Conductivity of the semiconductor is larger at higher temperature. Electron mobility and mean free time peaks at certain energy level ($\sim$ 5V at 4.2K) and has non-linear dependence on energy.

PACS numbers: 85.30.Tv; 73.20.-r; 72.80.Ey.

I. INTRODUCTION

Semiconductors are physically interesting materials because they have unique electrical conductive behavior like either insulators or conductors depending on their composition and environment. The electric transport properties of semiconductors are well explained in solid state physics with band theory. There is a relatively small ($\sim$ 1eV) band gap between the filled conduction band and the empty valence band, and Fermi energy lies in the forbidden region formed between two levels. In a metal-oxide-semiconductor (MOS) structured system, external electrical field makes electrons accumulate on the interface of the semiconductor and insulator and form a "two-dimensional electron gas". When a positive gate potential is applied to an n-channel MOSFET(with p-type substrate), band bending occurs as shown in Fig.1. At some gate voltage V_G, E_F will be close enough to E_c so that the surface is no longer in depletion but at the threshold of inversion. V_G at this threshold condition is defined as threshold voltage V_T. In some texts, threshold voltage is defined in another condition. We will discuss this ambiguity in Section.IV.

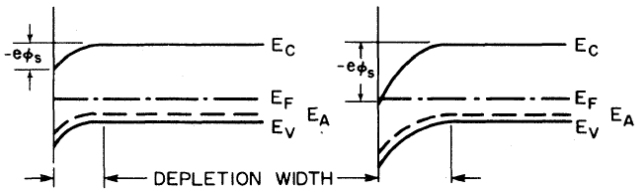


FIG. 1. The energy band diagram for a p-type semiconductor for (left) depletion of holes or ionization of neutral acceptors near the surface to form a depletion layer; (right) band bending strong enough to form an inversion layer of electrons at the surface.[1]

* Corresponding author: Email: tianyi_zhou@brown.edu

Under the inversion condition of a p-type semiconductor, conducting charges in the semiconductor are just electrons that form 2DEG. That is, conductivity of 2DEG σ is equivalent to that of the semiconductor, given by

$$\sigma = G \cdot L/W \tag{1}$$

where G is the conductance, L is the length of semiconductor from drain to source and W its width. When an electron is acted upon by a uniform electric field E, it will be accelerated until it reaches a constant drift velocity v_d. Electron mobility μ is then defined by

$$\mu = v_d/E \tag{2}$$

Ohm's law $\sigma = J/E$ where $J = eNv_d$ relates conductivity σ and mobility μ

$$\sigma = eN\mu \tag{3}$$

Electron density per area $N = Q/(e \cdot A_{2DEG})$ can be determined by assuming turned-on MOSFET as a parallel plate capacitor with capacitance $C = \epsilon_{ox}/t_{ox} \cdot A_{2DEG}$.

$$N = \frac{\epsilon_{ox}}{et_{ox}}(V_G - V_T) \tag{4}$$

In the mean free time model, the mean free time τ is related to mobility μ by

$$\tau = m^*\mu/e \tag{5}$$

where m^* is effective mass in 2DEG.

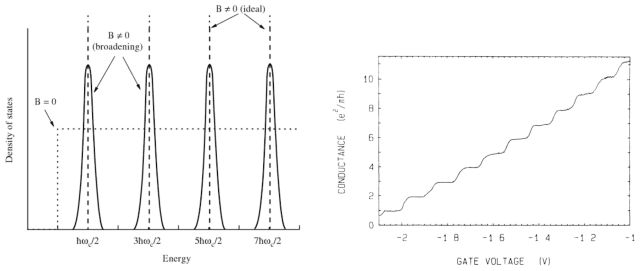
Electrons in a 2DEG can be approximated as free particles, and their density of state can be determined using the ideal gas model. We derive that 2DEG density of state(per unit area) $g(E) = m^*/(\pi\hbar^2)$ is a non-zero constant value as long as semiconductor Fermi energy E_F is above the bottom energy of the conduction band, i.e. $E_F > E_c$. Electron density is then given by $N = \int_E g(E)f(E)dE$. If we treat the Fermi distribution $f(E)$ as a step function taking $T \to 0$, N can be approximated by

$$N = \frac{m^*}{\pi\hbar^2}(E_F - E_c) \tag{6}$$

proportional to the energy. This changes when a magnetic field is applied perpendicular to the 2DEG. In a magnetic field, electrons form cyclotron orbits with quantized wave functions and can only occupy discrete energy levels called "Landau levels". Energy of 2DEG in a magnetic field is

$$E_F - E_c = (n + 1/2)\hbar\omega_c, \quad (n = 0, 1, 2, ...) \qquad (7)$$

where $\omega_c = eB/m^*$ is cyclotron frequency, and n is Landau index. Electrons now fill these discrete energy levels by "condensing" into groups, hence the density of state is a set of delta functions as shown in Fig.2(a). In this case,



(a) (b)

FIG. 2. (a)Density of state function vs energy for a 2DEG in a magnetic field [2](b) An illustration of plateau shaped conductance curve of a GaAs-AlGaAs heterostructure[5]

the electron density is no longer proportional to E_F but will rapidly increasing at Landau energy levels. Recall Eq.3, conductivity is proportional to N and will behave just as described above. Therefore, we are expecting to see a similar conductance curve as illustrated in Fig.2(b) once magnetic field is applied.

II. EXPERIMENT

In this experiment, two circuits (Fig.3) were used to measure the conductance profile of the 2DEG. Measurement and calibration were conducted at three different temperatures: room, LN2 and LHe. Static magnetic field was applied to the semiconductor at LHe temperature.

In both circuits, function generator was used to apply gate voltage V_G to the MOSFET, which was set to a 0.03Hz triangle wave ranging in 0-6V at room, LN2 temperature and in 0-25V at LHe temperature. In the conductance circuit, lock-in amplifier provided an AC signal to create a voltage from the drain to source of the MOSFET, hence a current dependent on the conductance of the semiconductor. This AC current was fed back to the lock-in and after a low-pass filter inside we got the amplified output signal (measured voltage V_{meas}). So measurement with conductance circuits gave the relation between V_G and V_{meas}. Calibration was done with conductance circuit by replacing the semiconductor with a variable resistor, giving the relation between V_{meas} and

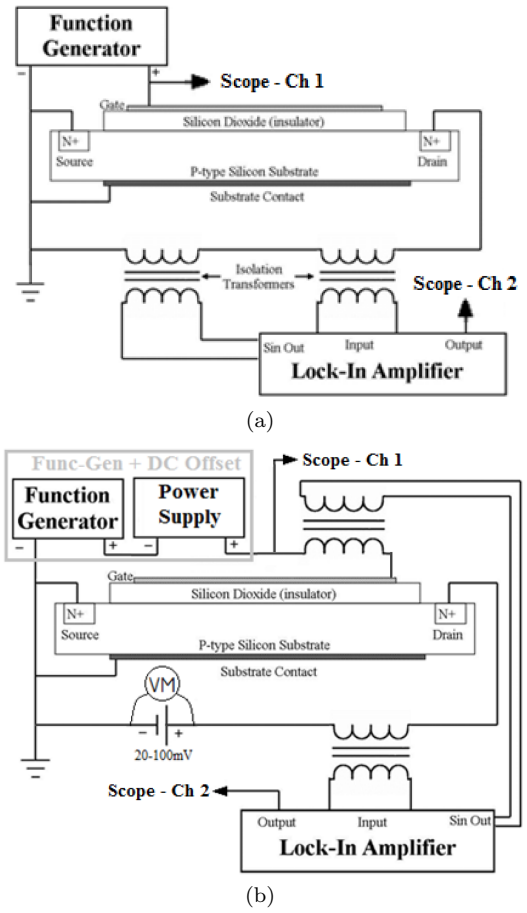


FIG. 3. Circuit for measuring (a) conductance (b) transconductance[3]

conductance G (just 1/R). When manipulating the lock-in amplifier, we had to optimize the output signal by adjusting the sensitivity, time constant and phase of the lock-in. The frequency we used in this experiment is about 92.4Hz. Although we noticed that the impedance of the circuit is dependent on the conductance of the semiconductor $G(V_G)$, hence the phase factor is a function of V_G, making it impossible for us to cancel out the relative phase between input and reference signal of the lock-in while V_G is sweeping. We can still accurately relate the conductance G and gate voltage V_G as long as we keep the settings of lock-in unchanged and the calibration curve gives a single-value correspondence. This will be discussed in detail in Sec.IV. In this case, we optimized signal at the saturated region (where the measured voltage almost does not changes as shown in Fig.II) by adjusting the phase.

We collected data from two channels of the oscilloscope in both conductance circuit and transconductance circuit at room and LN2 temperature. Then the LHe was added to the cryogenic dewar and the superconducting magnet was operated at this temperature. We set the current of magnet power supply to 33A, 30A, 27A and 25A, col-

lected conductance and transconductance data at every magnetic field. Typical conductance and transconductance signal are shown in Fig. 4. Sensitivity was set to be 5 mV in the conductance curve measurement and 100 μV in the transconductance measurement.

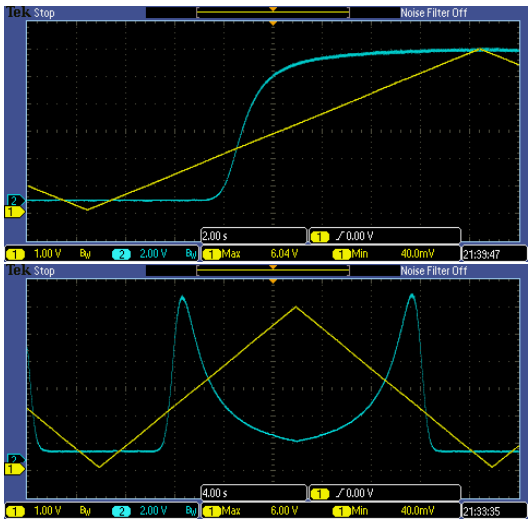


FIG. 4. Scope signal of conductance and transconductance circuit at room temperature, channel 1 is gate voltage and channel 2 is measured voltage.

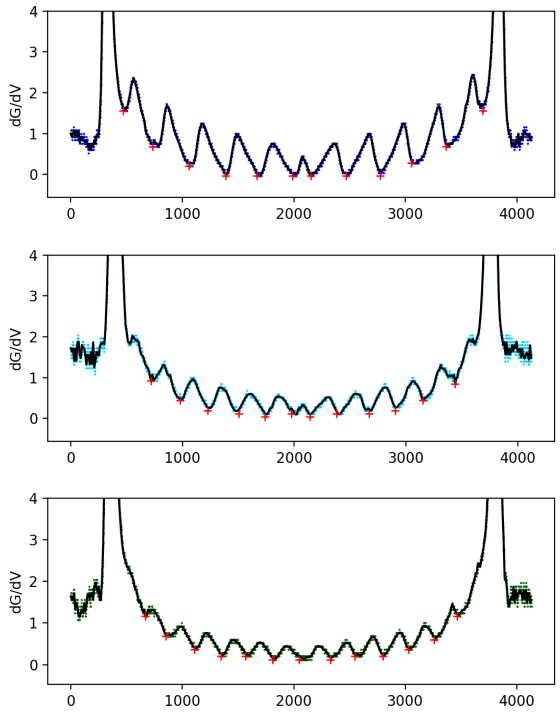


FIG. 5. Post-averaging way. Local minimum detection at current of the magnet power supply set to be 33A, 27A, 25A. x axis of the plot is the number of data points.

III. DATA ANALYSIS AND RESULTS

As described in Sec.I, when magnetic field is applied, we expect to see a plateau shaped conductance curve. However, the change in conductance is so tiny that we can hardly see that characteristic from conductance curve, Transconductance circuit measures the dG/dV_G, which allows us to see an oscillation with increasing V_G. Thus, the local minimum of the oscillation signal indicates the gate voltage where conductance changes rapidly, i.e. at the Landau level. Here we process with the transconductance data in two ways. The first is post-averaging way (Fig.5)- we locate the V_G of local minimum V_{meas} in one full cycle and then take the average of the detected V_G. Since the output waveform data from the digital oscilloscope was massive and noisy, we come up with the second way - pre-averaging, In this method, we average all V_{meas} at same V_G and get a set of data in a rising V_G, i.e. we project the dropping V_G part signal to the rising part and take the average. In both methods, we find the local minimum with the same peak-detect algorithm.

By relating Eq. 4, 6, 7, we can get the relation between V_G and the Landau indx n.

$$V_G(E_F) = V_T + \frac{e^2 t_{ox} B \omega_c}{\pi \hbar^2 \epsilon_{ox}}(n + \frac{1}{2}) \tag{8}$$

By comparing the detected local minimum and the measured curve, we determined the starting Landau indices. In both methods, the Landau index starts at 0, 1, 1 for 33A, 27A, 25A data. Thus, the value of V_G when $n = -1/2$ is the threshold voltage and three lines should converge to on point if we choose the correct starting Landau index. The slope of the line gives the value of the insulator thickness t_{ox}. Fig.7 shows the result of fitting (landau index, V_G) data points to straight line. The error is the error in averaging plus the error in peak-detecting(0.2V since the scope outputs V_G in step of 0.2V). In the post-averaging way, $V_T = 4.192 \pm 0.369$ V, $t_{ox} = 281.274 \pm 3.170$nm. In the pre-averaging way, $V_T = 4.178 \pm 0.383$ V, $t_{ox} = 281.904 \pm 3.368$nm. We can see from the result that the value is almost consistent in two averaging methods, but the error is slightly smaller in the post-averaging way. In the following calculation, we will use the mean value of two result, which is $V_T = 4.185 \pm 0.376$ V, $t_{ox} = 281.589 \pm 3.269$nm.

Now we are ready to process the conductance data and examine the electric transport properties of the 2DEG. One key point in the data processing is the correspondence of calibration and measurement data. The calibration curve we measured has rapid change in conductance at higher measurement voltage. But in the case of MOSFET data measurement, higher measurement voltage indicates saturated conductance, which should not change too much. So we chose a cutoff measurement voltage $V_T = 11V$ and taking all data that beyond this voltage to be the same value. Using interpolation spline we plotted the calibration curve in a V_{meas} space. Then we process

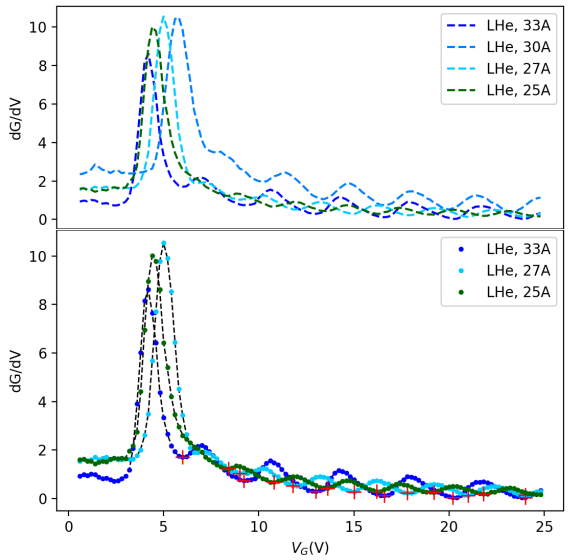
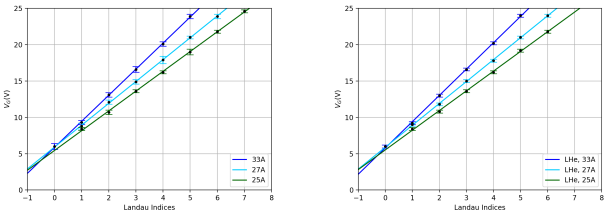


FIG. 6. Pre-averaging way. Upper figure shows the original data curve. In lower figure, dots are averaged data points and black dashed lines are fitted curve. Local minimum shown in red cursor. Data at 30A was removed when calculating.



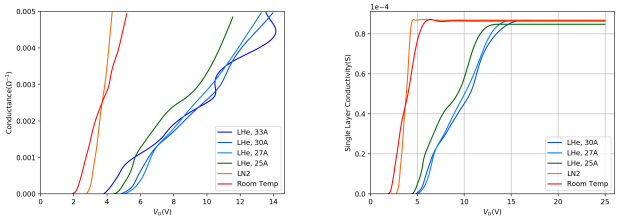
(a) (b)

FIG. 7. Straight line fitting result of (a)post-averaging way (b)pre-averaging way

the data taken in conductance circuit. Using the "pre-averaging" method mentioned before, we get a set of data relating V_{meas} to single V_G output from function generator at step of 0.2V. Spline interpolation gives the V_G profile in the same V_{meas} space. Now we successfully relates conductance G and gate voltage V_G. We noticed that LHe, 33A conductance curve actually shows the plateau-shaped characteristic(Fig.8(a)), but this brings difficulty to the data analysis process. Therefore we removed this data set in the following calculation. Conductivity is calculated by Eq.1, given that the dimension of the semiconductor is $W = 0.572$mm and $L = 10\mu$m. We can see from Fig.8(b) that, at room and LN2 temperature, the 2DEG starts to conduct at a lower V_G but at LHe temperature, higher V_G is needed to turn on the MOSFET. Eq.4 gives the electron density per unit area. And Eq.3 gives the electron mobility μ:

$$\mu = \sigma/(eN) \tag{9}$$

Then Eq.5 relates mean free time τ and mobility μ with



(a) (b)

FIG. 8. (a)Conductance vs V_G before the cutoff treatment, plateau shaped conductance curve is shown at LHe temperature when current set to 33A. (b) Conductivity vs V_G curve.

effective mass in 2DEG in x and y direction $m_{eff} = 0.19m_e$. The plots of electron mobility, mean free time, and density are shown in Fig.9.

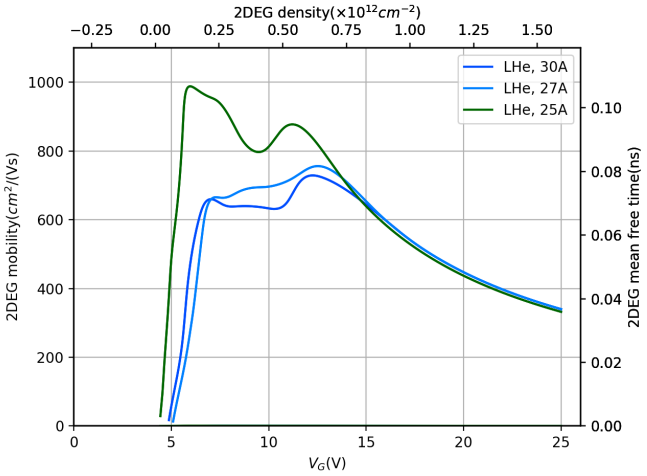


FIG. 9. Electron mobility μ and mean free time τ dependence on V_G and density N at LHe temperature with different magnetic fields where the current set to be 25A, 27A, 30A.

We can see that the 2DEG mobility is no longer constant at LHe temperature when magnetic field is applied, indicating the electron density N is not linearly increasing with the gate voltage. This is consistent with what we expected. Since 2DEG mean free time is just proportional to mobility, its behavior should be the same as that of mobility.

IV. DISCUSSION

In some texts, threshold voltage is defined as the gate voltage applied in the threshold condition when Fermi energy of the semiconductor E_F crosses the intrinsic energy E_i, where $E_i = (E_c + E_v)/2$ is just the mid-gap energy level of a semiconductor. This is different from what we described in Sec.I. This ambiguity on threshold voltage definition comes from the different definition of term "de-

pletion". In this alternative definition, transition of depletion condition and inversion condition happens when the electron densityn_c exceeds the hole density p_v, which corresponds to the condition when $E_F = E_i$. This condition can be understood by equation of density difference in extrinsic semiconductor[4]:

$$\frac{\Delta n}{n_i} = 2sinh[\beta(E_F - E_i)] \tag{10}$$

where $\Delta n = n_c - p_v$. But according to Ando, "depletion condition" refers to depletion of holes or ionization of neutral acceptors near the surface. The rest of the energy $E_c - E_i$ is used to ionize the neutral acceptors. In the aspect of band theory, "depletion" can be understood as any condition that E_F is far away from both bottom of conduction band E_c and top of the valence bandE_v, and n_c, p_v are both small.

In our experiment, the use of lock-in and the parameters setting are quite crucial to getting the optimized signal. The isolated transformers in the circuit are just inductors with inductance L. Thus the current running through the circuit will lag phase ϕ, where $\tan\phi = \omega L/R$. ω is the frequency we choose in the lock-in amplifier and R is the resistance of the semiconductor. Since we are measuring the R dependence of V_G, i.e. $R(V_G)$, the phase in the circuit is also dependent on gate voltage $\phi(V_G)$. But fortunately, we are not to make use of exact value of measurement voltage, treating it as a bridge that relates V_G and conductance G. As long as we get a single value calibration profile, this one-to-one correspondence is valid. In this experiment, we first set lock-in frequency to be 207 Hz and got a double-valued calibration curve. We tested that calibration data is a monotonous increasing curve when lock-in frequency is $\sim$ 90Hz or $\sim$ 400Hz and our data were collected at the frequency of 92.4Hz.

V. CONCLUSION

We set up two measurement circuits, i.e. the conductance circuit and the transconductance circuit, to study the electronic properties of a 2DEG gas. At LHe temperature, the magnetic field was applied to the semiconductor and quantum oscillation of conductivity was observed. The threshold voltage is found to be (4.426 $\pm$ 0.425)V and the insulator thickness is (281.328 $\pm$ 2.747)nm. The We used two averaging methods, i.e. post-averaging method and pre-averaging method to manipulate the oscilloscope output waveform data. The mobility and mean free time rises to a peak value and drops nearly linearly with increasing gate voltage.

There are several kinds of origin inducing errors. When we did the calibration and measurement, we found that it was hard to set the frequency of lock-in amplifier exactly the same value by adjusting the variable resistor and there is always a $\sim$0.3Hz off. Also, the precision can be improved by improving the precision of the oscilloscope since all the data was collected from the scope.

ACKNOWLEDGEMENT

We would like to thank Prof. Richard Gaitsgell for valuable discussion and instruction about the experiment. We are also grateful to Dean Hudek, Robert Horton and teaching assistant Erick Garcia for great help in transferring liquid helium and other procedures in the lab. Lastly, we want to thank Brown University Department of Physics for supporting experimental instruments.

[1] T. Ando, A. Fowler and F. Stern, Rev.Mod.Phys.(1982) 54: 437, *Electronic properties of two-dimensional systems*,

[2] B. Wees, H. Houten *Quantized Conductance of Point Contacts in a Two-Dimensional Electron Gas*, Physical Review Letters, (1988) 60:9

[3] *lab manual*, available at `https://wiki.brown.edu/confluence/display/PhysicsLabs/PHYS+2010+Lab+Files`.

[4] N. Ashcroft, N. Mermin *Solid State Physics*, (Holt, Rinehart and Winston, 1976).

[5] J. Duart, R.Paima, F. Rueda. *Nanotechnology for microelectronics and optoelectronics*, (Elsevier, 2006)

Appendix

A.1 Python codes of calculation and plotting

Transconductance:

```
import numpy as np
import scipy.optimize as opt
from scipy import interpolate
import matplotlib.pyplot as plt
from peakdetect import peakdetect

h_bar = 1.055e-34       # Planck constant (J*s)
e_vacc = 8.854e-12      # Permittivity in vaccum (F/m)
e_SiO2 = 3.9 * e_vacc   # Dielectric const for SiO2
q_e = 1.60218e-19       # Electron charge (C)


def f(x, a, b):
    return a * x + b


# Prepare parameters in the loop

file_list = ('T_33A_LHe.csv', 'T_30A_LHe.csv', 'T_27A_LHe.csv', 'T_25A_LHe.csv')
label_list = ('33A', '30A', '27A', '25A')
I_list = [33, 30, 27, 25]
color_list = ((0, 0, 1), (0, 0.5, 1), (0, 0.8, 1), (0, 0.4, 0))
LIdxStart_list = [0, 1, 1, 1]

V_T = []
errV_T = []
V_T_def2 = []
errV_T_def2 = []
t_ox = []
V_G = [[], [], [], []]

for i in range(0,4):

# Import transconductance data
    # Select data in one cycle

    Data = np.loadtxt(file_list[i], delimiter=',', skiprows=16, usecols=(1, 3))

    minVGidx = Data[:, 0].argmin()
    Data = Data[minVGidx::]
    minVGidx = Data[::-1, 0].argmin()
    Data = Data[:(len(Data) - minVGidx):]

    N = len(Data)
    Idx = np.linspace(0, N, N)

    x = Idx
    y = Data[:, 1]

    # plt.figure('Original')
    # plt.plot(x, y, 'k.', markersize=1)
    # plt.xlabel(r'Index')
    # plt.ylabel(r'dG/dV')

    # Find local minimum between 2 peaks using peakdetect.py
    # Delete other minimum outside 2 peaks
```

```
    (locMax, locMin) = peakdetect(y, lookahead=63)
    locMax = np.array(locMax)
    locMin = np.array(locMin)

    sortedMaxIdx = np.argsort(locMax[:, 1])
    peakIdx1 = locMax[sortedMaxIdx[-1], 0]
    peakIdx2 = locMax[sortedMaxIdx[-2], 0]
    idxStart = min(peakIdx1, peakIdx2)
    idxEnd = max(peakIdx1, peakIdx2)
    print(Data[int(peakIdx1), 0])
    ThV_def2 = (Data[int(peakIdx1), 0] + Data[int(peakIdx2), 0])/2
    errThV_def2 = 0.2

    condition = (locMin[:, 0] > idxStart) & (locMin[:, 0] < idxEnd)
    locMin = locMin[condition]

# Smooth signal using interpolate spline
    # Set smoothing factor to 0.3

    x2 = x[::10]
    y2 = y[::10]
    spl = interpolate.UnivariateSpline(x2, y2)
    spl.set_smoothing_factor(0.3)
    ySmooth = spl(x2)

    plt.figure(label_list[i])
    plt.plot(x, y, '.', color=color_list[i], markersize=1)
    plt.plot(x2, ySmooth, 'k')
    plt.xlabel(r'Time(arbitrary unit)')
    plt.ylabel(r'dG/dV')
    plt.ylim(ymax=4)
    plt.plot(locMin[:, 0], locMin[:, 1], 'r+')

# Find corresponding gate voltage using index
    # Deal with asymmetric peakdetect result
    if i == 1:
        continue
    idxGV = [int(x) for x in locMin[:, 0]]

    gateV = np.sort(Data[idxGV, 0])

    if np.mod(len(gateV), 2) == 1:
        gateV = np.append(gateV, gateV[-1])
    gateV = np.reshape(gateV, (int(len(gateV) / 2), 2))

    print(gateV)

    errGV = [np.std(x)+0.2 for x in gateV]
    meanGV = [np.mean(x) for x in gateV]
    V_G[i] = np.append(V_G[i], meanGV)
    print('\nGate Voltage is {}'.format(meanGV))
    print('Gate Voltage error is {}'.format(errGV))

    LandauIdx = np.arange(LIdxStart_list[i], len(gateV) + LIdxStart_list[i])

# Fit gate voltage points to a straight line
    # and plot four lines

    LIdx_range = np.arange(-1, 10)
    popt, pcov = opt.curve_fit(f, LandauIdx, meanGV)
```

```
    perr = np.sqrt(np.diag(pcov))
    print('Fitting param {}, param error {}'.format(popt, perr))

    plt.figure('QO')
    plt.plot(LIdx_range, f(LIdx_range, *popt), color=color_list[i],
             label=label_list[i])
    plt.errorbar(LandauIdx, meanGV, yerr=errGV, fmt='k.', ecolor=color_list[i], capsize=5)
    plt.axis((-1, 8, 0, 25))
    plt.grid('on')
    plt.xlabel(r'Landau Indices')
    plt.ylabel(r'$V_G$(V)')
    plt.legend(loc='lower right')

# Calculate Threshold voltage V_T
    # V_T = f(-1/2)

    ThV = f(-1/2, popt[0], popt[1])
    errThV = perr[0]/2 + 0.2
    V_T = np.append(V_T, ThV)
    errV_T = np.append(errV_T, errThV)
    V_T_def2 = np.append(V_T_def2, ThV_def2)
    errV_T_def2 = np.append(errV_T_def2, errThV_def2)

# Calculate thickness t_ox
    # t_ox = slope * pi * h_bar * epsilon / e^2 / B

    B = 1737 * I_list[i] * 1e-4
    tox = popt[0] * np.pi * h_bar * e_SiO2 / (q_e**2 * B)
    t_ox = np.append(t_ox, tox)


print('\nThreshold voltage', V_T)
print('Error in threshold voltage', errV_T)
print('\nThreshold voltage(def2)', V_T_def2)
print('Error in threshold voltage(def2)', errV_T_def2)
sd_VT = np.std(V_T)
V_T = np.mean(V_T)
errV_T = np.mean(errV_T)
V_T_def2 = np.mean(V_T_def2)
errV_T_def2 = np.mean(errV_T_def2)

print('\nThickness is', t_ox)
sd_t_ox = np.std(t_ox)
t_ox = np.mean(t_ox)

print('\nThreshold voltage is {:5.3f} +/- {:5.3f} V'.format(V_T, errV_T+sd_VT))
print('\nThreshold voltage(def2) is {:5.3f} +/- {:5.3f} V'.format(V_T_def2, errV_T_def2))
print('Thickness is {:5.3f} +/- {:5.3f} nm'.format(t_ox * 1e9, sd_t_ox * 1e9))  # Convert m
to nm


# Save gate voltage to a file
transfile = open('transfile.csv', 'w')

for i in range(0, 4):
    for j in V_G[i]:
        transfile.write(str(j)+',')
    transfile.write('\n')
transfile.write('\n')
# Calculate number of electrons in all energy levels per unit N
    # and plot N(V_gate)
```

```
for i in range(0, 4):
    N = [e_SiO2 * (x - V_T) / (q_e * t_ox) * 1e-4 for x in V_G[i]]    # Convert m^-2 to cm^-2
    for j in N:
        transfile.write(str(j)+',')

    transfile.write('\n')
    #plt.figure('Electron Density N')
    #plt.plot(V_G[i], N, 'o', markersize=5, color=color_list[i], label=label_list[i])
    #plt.xlabel(r'$V_G$(V)')
    #plt.ylabel(r'2DEG density($cm^{-2}$)')
    #plt.legend(loc='lower right')


transfile.close()
plt.show()
```

Conductance:

```
import numpy as np
import xlrd
from scipy import interpolate
import scipy.optimize as opt
import matplotlib.pyplot as plt


def find_nearest(array,value):
    idx = (np.abs(array-value)).argmin()
    return idx


e_vacc = 8.854e-12      # Permittivity in vaccum (F/m)
e_SiO2 = 3.9 * e_vacc   # Dielectric const for SiO2
q_e = 1.60218e-19       # Electron charge (C)
m_e = 9.10938e-31       # Electron mass (kg)
m_eff = 0.19 * m_e      # Effective mass (kg)

L = 1e-5                # Length of substrate (m)
W = 5.72e-4             # Width of substrate (m)

t_ox = 281.589e-9        # Thickness of dioxide (m)
V_T = 4.185             # Threshold voltage (V)

# Prepare parameters in the loop

file_list = ('C_33A_LHe.csv', 'C_30A_LHe.csv', 'C_27A_LHe.csv', 'C_25A_LHe.csv', 'C_LN2.csv',
'C_RoomTemp.csv')
label_list = ('LHe, 33A', 'LHe, 30A', 'LHe, 27A', 'LHe, 25A', 'LN2', 'Room Temp')
color_list = ((0, 0, 1), (0, 0.3, 1), (0, 0.5, 1), (0, 0.4, 0), (1, 0.4, 0), (1, 0, 0))

smooth1_list = [0.2, 0.5, 0.5, 0.5, 0.005, 0.001]

book1 = xlrd.open_workbook('transfile.xlsx')
sheet1 = book1.sheet_by_index(0)

fig, ax1 = plt.subplots()

for i in range(0, 6):

# Import Calibration and Conductance data
```

```
    Cali_data = np.loadtxt('Calibration.csv', delimiter=',', skiprows=10, usecols=(1, i+2))
    Data = np.loadtxt(file_list[i], delimiter=',', skiprows=16, usecols=(1, 3))
    VarG = Cali_data[::-1, 0]
    MeasV = Cali_data[::-1, 1]

# Plot Calibration curve: condunctance v.s. V_gate

    SatVm = 11

    spl = interpolate.UnivariateSpline(MeasV, VarG)
    Vm_space = np.linspace(0.2, 11.2, 1000)
    spl.set_smoothing_factor(0.00000001)
    G_spline = spl(Vm_space)

    for Vm in Vm_space:
        if Vm > SatVm:
            idx_sat = list(Vm_space).index(Vm)
            break
    G_sat = G_spline[idx_sat - 1]
    G_spline[idx_sat:] = [G_sat] * len(G_spline[idx_sat:])

    plt.figure('Calibration')
    plt.plot(Vm_space, G_spline, color=color_list[i], label=label_list[i])
    plt.plot(MeasV, VarG, 'k.', markersize=2)
    plt.xlabel(r'Measured V(V)')
    plt.ylabel(r'Conductance($\Omega^{-1}$)')


# Plot original data points

    plt.figure('Scope data')
    plt.plot(Data[:, 1], Data[:, 0], '.', color=color_list[i],  markersize=1)
    plt.xlabel(r'Measured V(V)')
    plt.ylabel(r'$V_G$(V)')

# Plot averaged curve
    # Select data with mono-increasing V_measure
    #  by deleting starting zero values
    #  and cut off by saturation V_measure
    # Then using spline interpolation to get smooth curve

    idx = np.argsort(Data[:, 0])
    sorted = Data[idx, 0]
    V1, idx_start, count = np.unique(sorted, return_counts=True, return_index=True)
    res = np.split(idx, idx_start[1:])
    V2 = np.ones(len(res))
    for (i_uniq_value, j) in zip(res, range(0, len(res))):
        unique_value = np.mean(Data[i_uniq_value, 1])
        V2[j] = unique_value

    print('V1 is', V1, '\nV2 is', V2)

    V1V2 = np.transpose([V1, V2])
    condition = ((V1V2[:, 1] - V1V2[0, 1]) > 0.05)
    V1V2 = V1V2[condition]

    idx = np.argsort(V1V2[:, 1])
    sorted = V1V2[idx, 1]
    V2, idx_start, count = np.unique(sorted, return_counts=True, return_index=True)
    res = np.split(idx, idx_start[1:])
    V1 = np.ones(len(res))
```

```
    for (i_uniq_value, j) in zip(res, range(0, len(res))):
        unique_value = np.mean(V1V2[i_uniq_value, 0])
        V1[j] = unique_value
    V1V2 = np.transpose([V1, V2])

    V1V2lt10 = V1V2[np.where(V1V2[:, 1] < 10)]
    V1V2gt10 = V1V2[np.where(V1V2[:, 1] > 10)]
    print(V1V2gt10)
    V1V2 = np.append(V1V2lt10, V1V2gt10[::6], axis=0)
    print(V1V2)

    spl = interpolate.UnivariateSpline(V1V2[:, 1], V1V2[:, 0])
    spl.set_smoothing_factor(smooth1_list[i])
    VG_spline = spl(Vm_space)

# Plot V_gate v.s. V_measure

    plt.figure('Averaged scope curve')
    plt.plot(Vm_space, VG_spline, color=color_list[i], label=label_list[i])
    plt.plot(V1V2[:, 1], V1V2[:, 0], 'k+', markersize=5)
    plt.xlabel(r'Measured V(V)')
    plt.ylabel(r'$V_G$(V)')
    plt.legend()



# Plot Conductance Curve: conductance v.s. V_gate
    plt.figure('VGspline-Gspline')
    plt.plot(VG_spline[:idx_sat:], G_spline[:idx_sat:], color=color_list[i],
label=label_list[i])
    plt.xlabel(r'$V_G$(V)')
    plt.ylabel(r'Conductance($\Omega^{-1}$)')
    plt.xlim(xmin=0)
    plt.ylim(ymin=0, ymax=0.005)
    plt.legend()

    if i == 0:
        continue


    VG_spline = np.delete(VG_spline, list(range(idx_sat-1, idx_sat+9)))
    G_spline = np.delete(G_spline, list(range(idx_sat-1, idx_sat+9)))

    VG = np.append(VG_spline[:980:12], VG_spline[984::4])
    G = np.append(G_spline[:980:12], G_spline[984::4])

    VG_append = np.linspace(VG[-1]+2, 25, int((25-VG_spline[idx_sat-1])/2))
    G_append = [G_sat] * len(VG_append)

    VG = np.append(VG, VG_append)
    G = np.append(G, G_append)

    print(VG)
    VG_space = np.linspace(min(VG), max(VG), 1000)
    spl = interpolate.UnivariateSpline(VG, G)
    spl.set_smoothing_factor(0)
    Cond = spl(VG_space)

    plt.figure('Conductance Curve')
    plt.plot(VG_space, Cond, color=color_list[i], label=label_list[i])
    plt.plot(VG, G, 'k.', markersize=2)
```

```
    plt.xlabel(r'$V_G$(V)')
    plt.ylabel(r'Conductance($\Omega^{-1}$)')
    plt.xlim(xmin=0)
    plt.ylim(ymin=0, ymax=0.006)
    plt.ticklabel_format(style='sci', axis='y', scilimits=(0,0))
    plt.legend()

# Plot Conductivity v.s. VG

    sigma = L / W * Cond
    plt.figure('Conductivity')
    plt.plot(VG_space, sigma, color=color_list[i], label=label_list[i])
    plt.xlabel(r'$V_G$(V)')
    plt.ylabel(r'Single Layer Conductivity(S)')
    plt.xlim(xmin=0)
    plt.ylim(ymin=0)
    plt.ticklabel_format(style='sci', axis='y', scilimits=(0,0))
    plt.legend()

# Calculate N
    N = [e_SiO2 * (x - V_T) / (q_e * t_ox) * 1e-4 for x in VG_space]    # Convert m^-2 to
cm^-2


# Plot mobility v.s. VG
    if i in [0, 4, 5]:
        continue

    VG_discrete = sheet1.row_values(i)
    N_discrete = sheet1.row_values(i+4)
    VG_discrete = list(filter(None, VG_discrete))
    N_discrete = list(filter(None, N_discrete))
    print(VG_discrete)
    nearest_idx = [find_nearest(VG_space, x) for x in VG_discrete]

    miu_discrete = [sigma[x1] / q_e / x2 for (x1, x2) in zip(nearest_idx, N_discrete)]  # in
unit cm^2/(Vs)
    miu = [x1 / q_e / x2 for (x1, x2) in zip(sigma, N)]
    #fig, ax1 = plt.subplots()
    plt.grid('on')
    #plt.figure('Mobility')
    #plt.plot(VG_discrete, miu_discrete, 'D', color=color_list[i])
    ax1.plot(VG_space, miu, color=color_list[i], label=label_list[i])
    ax1.set_xlabel(r'$V_G$(V)')
    ax1.set_ylabel(r'2DEG mobility($cm^2$/(Vs)')
    ax1.set_xlim([0, 26])
    ax1.set_ylim([0, 1.1e3])
    #ax1.ticklabel_format(style='sci', axis='y', scilimits=(0,0))
    ax1.grid('on')
    ax1.legend()

# Plot mean free time tau v.s. VG
    tau_discrete = [m_eff * x * 1e-4 / q_e for x in miu_discrete]  # Convert miu back to
m^2/(Vs)
    tau = [m_eff * x * 1e-4 / q_e for x in miu]                    # Convert s to ns
    tau_lim = m_eff * 1.1e6 * 1e-4 * 1e9 / q_e
    #plt.figure('Mean free time')
    #plt.plot(VG_discrete, tau_discrete, 'D', color=color_list[i])
    ax2 = ax1.twinx()
    ax2.plot(VG_space, [x*1e9 for x in tau], color=color_list[i], label=label_list[i])
    ax2.set_ylabel(r'2DEG mean free time(ns)')
```

```
    ax2.set_ylim([0, tau_lim])
    #ax2.ticklabel_format(style='sci', axis='y', scilimits=(0,0))
    #plt.legend()

# Plot mobility v.s. density
    plt.figure('Mobility v.s. Density')
    plt.grid('on')
    #plt.plot(N_discrete, miu_discrete, 'D', color=color_list[i])
    plt.plot(N, miu, color=color_list[i], label=label_list[i])
    plt.xlabel(r'2DEG density($cm^{-2}$)')
    plt.ylabel(r'2DEG mobility($cm^2$/(Vs)')
    plt.ticklabel_format(style='sci', axis='y', scilimits=(0,0))
    plt.legend()

    ax3 = ax1.twiny()
    N_lim_min = e_SiO2 * (0 - V_T) / (q_e * t_ox) * 1e-4 * 1e-12      # gate limit_min set to
be 0V
    N_lim_max = e_SiO2 * (26 - V_T) / (q_e * t_ox) * 1e-4 * 1e-12      # gate limit_max set
to be 26V
    ax3.plot([x*1e-12 for x in N], miu, color=color_list[i], label=label_list[i])
    ax3.set_xlabel(r'2DEG density($\times 10^{12} cm^{-2}$)')
    ax3.set_xlim([N_lim_min, N_lim_max])
    #ax3.ticklabel_format(style='sci', axis='x', scilimits=(0, 0))

plt.show()
```

A.2 Inventory Sheet

Quantum Oscillations, B&H room 203
Inventory Sheet
10/24/17

Start Up	#	Inventory	Close Out
✓	1	Connector Box	✓
✓	1	Probe & Mosfet Holder	✓
✓	1	Transformer Box	✓
✓	1	10 Turn Pot Box	✓
✓	1	Cryogenic Cylinder	✓
✓	1	Helium Level Indicator	✓
✓	1	Persistent Switch	✓
✓	1	Programmer/Monitor	✓
✓	1	Cryomagnetic System Manual	✓
✓	1	Digital Oscilloscope with printer	✓
✓	1	D. C. Power Supply	✓
✓	2	D. C. Power Supply	✓
✓	1	Function Generator with manual	✓
✓	1	New Mosfet (SK9158)	✓
✓	1	Lock-in Amplifier with manual	✓
✓	1	Decade Box	✓
✓	1	Helium Transfer Tube	✓
✓	2	Pair Cryogen gloves and Safety Glasses	✓
✓	1	Roll of yellow tape	✓
✓	1	**Safety Manual In Room 203**	—

Comments:

AT TIME OF CHECKOUT THIS AREA WAS NEAT AND ORDERLY AND THE FOLLOWING ITEMS WERE DISCUSSED: ✓ THE LAB DOOR, ✓ NO FOOD OR DRINK IN LAB, ✓ WHERE TO FIND CABLES AND CONNECTORS, ✓ HOW TO GET AND USE LIQUID NITROGEN, ✓ HOW TO GET AND USE LIQUID HELIUM (TRANSFER TUBE), ✓ SAFE HANDLING OF CRYOGENS, AND ✓ LAB SAFETY.

Tianyi Zhou 11/16/17
(Student signature) (Date)

Fan Zhang 11/16/2017
(Student signature) (Date)

AT TIME OF START UP/CHECK OUT THIS AREA WAS NEAT AND ORDERLY AND ALL INVENTORY ITEMS WERE ACCOUNTED FOR.

Start Up:

Eric Bauer 11/16/17
(Staff signature) (Date)

Check Out:

Fan Zhang 11/11/17
(Staff signature) (Date)