

## Лабораторная работа 1.3.2

### Изучение магнитного и электрического гистерезиса.

Зуева Надежда 594 группа

**Цель:** исследовать поляризацию сегнетоэлектриков в зависимости от напряжённости

приложенного электрического поля; исследовать намагниченность ферромагнетиков в

зависимости от напряжённости приложенного магнитного поля; изучить явление

гистерезиса на их примере.

**Оборудование:** электронный осциллограф, звуковой генератор, вольтметр, модуль ФПЭ-07, модуль ФПЭ-02.

## Часть 1

### Ферромагнетики

1. Определение напряжения магнитного поля  $H$  в тороидальной катушке. По теореме о циркуляции,  $2\pi r_{cp}H = I_1N_1$

Измеряем напряжение  $U_x$

на сопротивлении  $R_1$

в первичной катушке. Сигнал регистрируем по осциллографу (канал "x").

$$\text{Ток } I = \frac{U_x}{R_1}$$

, где  $R_1 = 51$

Ом

,  $N_1 = 100$

$$\text{, } r_{cp} = \frac{d_1 + d_2}{4}$$

,  $d_1 = 18 \cdot 10^{-4}$  м

,  $d_2 = 9 \cdot 10^{-4}$  м

$$H = \frac{U_x}{R_1} \cdot \frac{N_1}{2\pi r_{cp}}$$

### 2. Определение индукции

$B$  магнитного поля в тороидальной катушке.

Во вторичной катушке:  $E_{\text{инд}} = \frac{d\Phi}{dt}$ ,  $\Phi = SN_2B$

. Предполагая, что  $E_{\text{инд}} = U_C$ ,  $U_C = U_y$  (канал "y")

, получим:

$$U_C = \frac{1}{CH^2} \int SN_2 \frac{dB}{dt} = - \frac{1}{CH^2} SN_2 B$$

, здесь  $S$

-- площадь сечения тора.

3. Построим графики  $B(H)$ ,  $J(H)$ ,  
 $\mu(H)$ .

In [2]:

```
import numpy as np
```

```
import math
import matplotlib.pyplot as plt
%matplotlib inline
h = 0.1
S = 6 * 3.5 * 0.01
C2 = 0.047
R1 = 470 * 0.001
R2 = 11 * 0.001
def round_(f):
    return round(f, 5)
```

Запишем данные с приборов и построим *графики*:

In [3]:

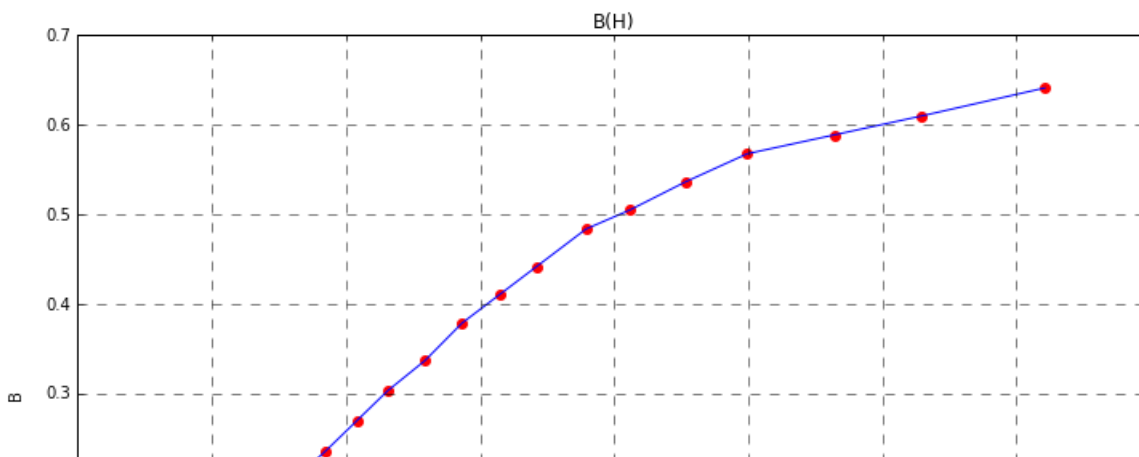
```
x_V = np.array([7.8*20, 6.8*20, 6.1*20, 5.4*20, 9.8*10, 8.9*10, 8.2*10, 7.4*10, 6.8*10, 6.2*10,
5.6*10, 5*10, 9*5, 8*5, 4.8*5, 3.8*5, 2.8*5, 1.6*5])
y_V = np.array([6.1*0.5, 5.8*0.5, 5.6*0.5, 5.4*0.5, 5.1*0.5, 4.8*0.5, 4.6*0.5, 4.2*0.5, 3.9*0.5, 3.6*0.5,
8*0.2, 7.2*0.2, 6.4*0.2, 5.6*0.2, 6.4*0.1,
4.8*0.1, 6.4*0.05, 3.2*0.05])
H = x_V/51*100/(2*np.pi*27/2*0.01)
B = y_V*0.22*10**(-6)*4.3*10**3/(200*0.005/2*0.009)
print("Результаты расчетов для B:\n", B)
m_0 = 4*np.pi*10**(-7)
J = (B / m_0) - H
plt.figure(figsize=(12,8))
plt.scatter(H, B, color = 'r', linewidth=2.5)
plt.plot(H, B)
plt.xlabel('H')
plt.title("B(H)")
plt.ylabel('B')
plt.grid(ls='--')
plt.show()

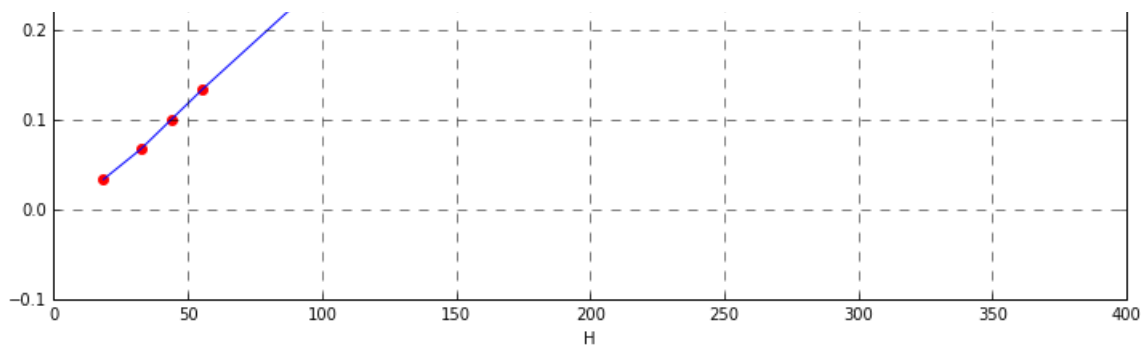
print("Результаты расчетов для J:\n", J)
plt.figure(figsize=(12,8))
plt.scatter(H, J, color = 'g', linewidth=2.5)
plt.plot(H, J)
plt.xlabel('H')
plt.title("J(H)")
plt.ylabel("J $\cdot 10^5$")
plt.grid(ls='--')
plt.show()

print("Результаты расчетов для mu:\n", B/(H*4*np.pi*10**(-7)))
plt.figure(figsize=(12,8))
plt.scatter(H, B/(H*4*np.pi*10**(-7)), linewidth=2.5, color = "orange")
plt.plot(H, B/(H*4*np.pi*10**(-7)))
plt.title("$\mu(H)$")
plt.ylabel("$\mu$")
plt.grid(ls='--')
plt.show()
```

Результаты расчетов для B:

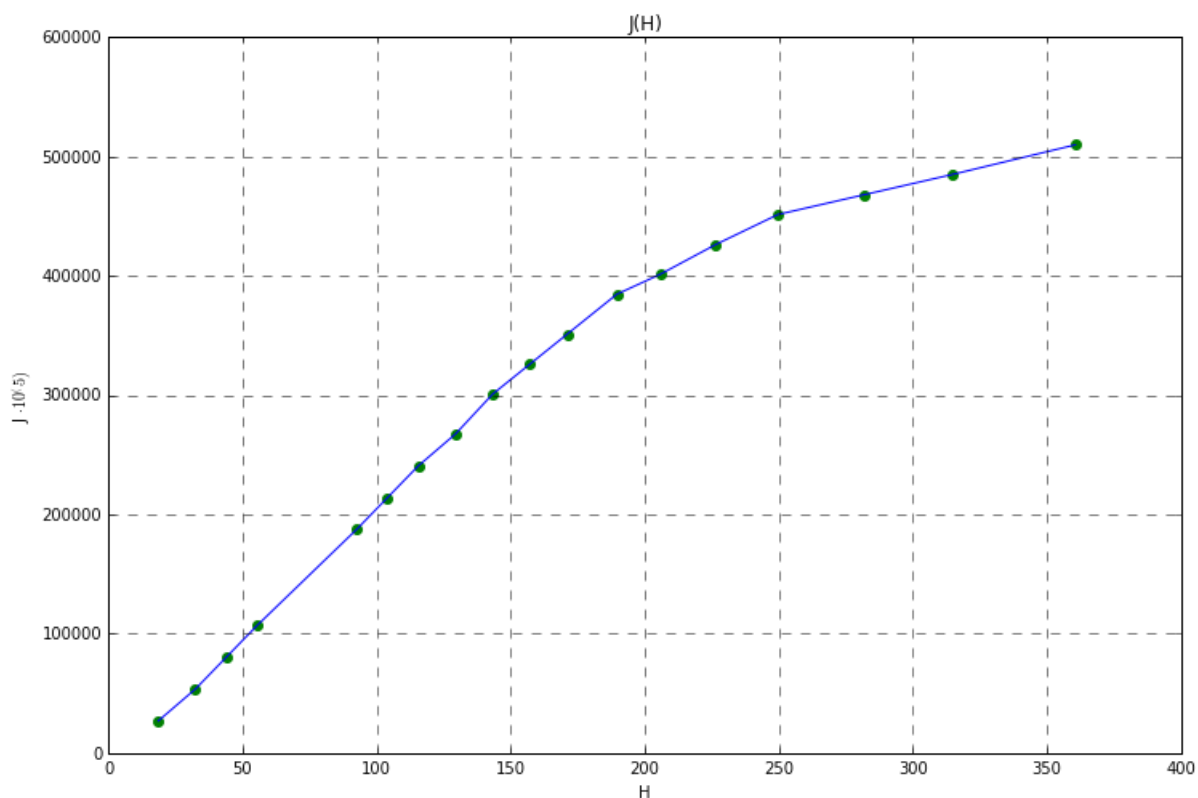
0.64117778	0.60964444	0.58862222	0.5676	0.53606667	0.50453333
0.48351111	0.44146667	0.40993333	0.3784	0.33635556	0.30272
0.26908444	0.23544889	0.13454222	0.10090667	0.06727111	0.03363556





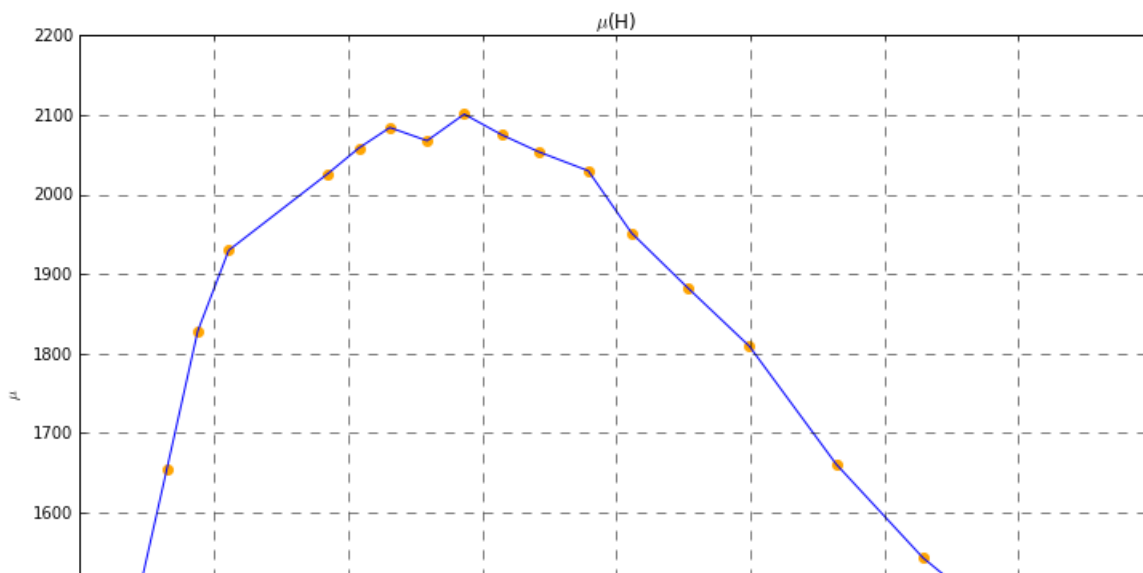
Результаты расчетов для J:

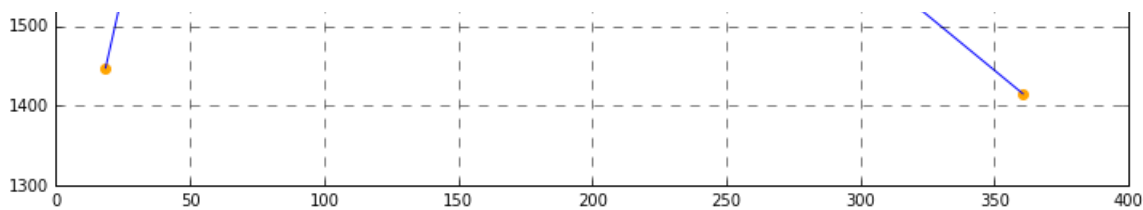
[ 509872.45116283	484825.25417473	468128.66392842	451432.07368211
426361.76050764	401289.13571455	384576.36413778	351136.95127244
326057.39162344	300977.83197443	267533.79587182	240781.34093209
214026.57437373	187271.80781536	107009.81975891	80255.05320055
53500.28664218	26747.83170246]		



Результаты расчетов для  $\mu$ :

[ 1414.90673077	1543.1625	1660.92786885	1809.225	1883.07091837
1951.52359551	2029.86219512	2053.71486486	2075.2875	2101.03548387
2067.68571429	2084.2272	2058.496	2026.332	1929.84
1828.26947368	1654.14857143	1447.38		

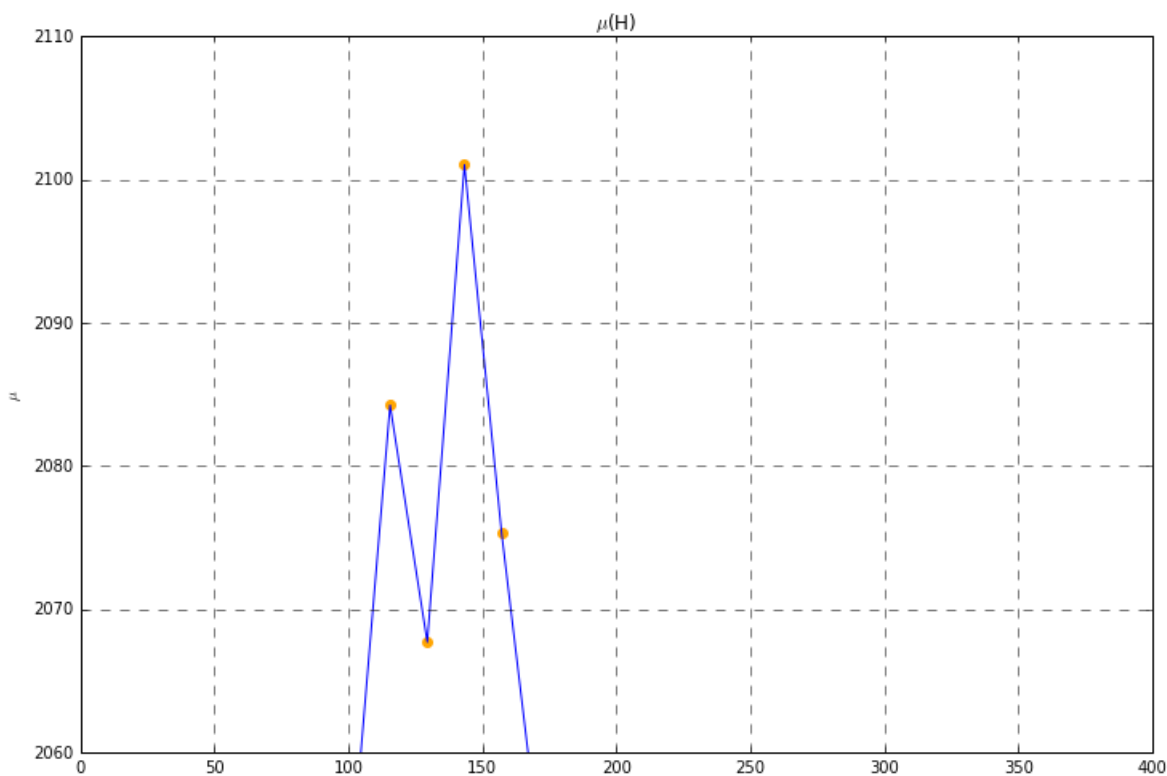




Увеличим график для  $\mu$   
вблизи максимума:

In [4]:

```
plt.figure(figsize=(12,8))
plt.scatter(H, B/(H**4*np.pi*10**(-7)), linewidth=2.5, color = "orange")
plt.plot(H, B/(H**4*np.pi*10**(-7)))
plt.title("$\mu$(H)")
plt.ylabel("$\mu$")
plt.ylim(2060, 2110)
plt.grid(ls='--')
plt.show()
```



высчитаем значения пика в 9й точке аналитически:

In [5]:

```
print ("намагниченность в момент насыщения:", round(M[9],4))
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-5-2110d92c49de> in <module>()
----> 1 print ("намагниченность в момент насыщения:", round(M[9],4))

NameError: name 'M' is not defined
```

In [6]:

```
mu = np.array(list(B/H))
mu = mu * 10**4
print(mu)
print("Максимальная магнитная проницаемость:", np.max(mu)/2)
```

```
[ 17.78024236  19.39195189  20.87183516  22.73539187  23.66336705
 24.52356876  25.50800064  25.80774213  26.07883186  26.40239056
 25.983305    26.19117144  25.86782364  25.4636389   24.25108467
 22.97471179  20.786644    18.1883135 ]
```

Максимальная магнитная проницаемость: 13.2011952821

In [7]:

```
print("B_r =", round(4*np.pi*2101.035,3), "Тл")
```

B\_r = 26402.384 Тл

## Часть 2

### Сегнетоэлектрики

Эта часть работы посвящена измерению  $E$

.

Напряженность в поле конденсатора может быть определена как  $E = \frac{U_{C_1}}{h_1}$ , где  $U_{C_1}$

-- разность потенциалов между пластинами конденсатора, а  $h_1$

-- расстояние между ними.

На канал  $X$

подается напряжение, равное

$$U_x = I_2 \cdot R_2 = \frac{U}{R_1 + R_2} \cdot R_2 = \frac{R_2}{R_1 + R_2} (U_{C_1} + U_{C_2})$$

,

где  $U = U_{C_1} + U_{C_2}$

-- суммарное напряжение на конденсаторах.

Параметры схемы таковы, что  $C_1 \ll C_2$

, поэтому  $\frac{U_{C_1}}{U_{C_2}} = \frac{C_1}{C_2} \gg 1$

, откуда  $U \approx U_{C_1}$

$$\text{Тогда } E = \frac{1}{300} \cdot \frac{R_1 + R_2}{R_2} \cdot \frac{U_x}{h_1} = \frac{\sqrt{2}V}{300 \cdot h_1}$$

в СГС.

Пусть  $V$

-- показания вольтметра

In [12]:

```
N1 = 100
N2 = 200
R1 = 51
R2 = 4300
C = 0.00000022
r1 = 0.018
r2 = 0.009
Hc = round_(2*N1*5*0.02/(5*((r1+r2)/2)*R1))
Bh = round_(R2*C*19*0.5/(5*N2*0.45) * 10000000)
Hh = round_(2*N1*15.5*0.02/(5*((r1+r2)/2)*R1) * 10)
print("M_h = ", round_((Bh - Hh)/(4*np.pi)))
Br = round_(R2*C*7*0.5/(5*N2*0.45) * 1000000)
print("H_c = ", Hc)
print("B_r = ", Br)
```

M\_h = 1.56047

H\_c = 5.80973

B\_r = 7.35778

Получим на осциллографе изображения сигналов. Причем, канал  $X$  пропорционален  $E$

, а канал  $Y$

-- пропорционален  $D$

в сегнетоэлектрическом образце. Наблюдаем изменение по каналам в зависимости от амплитуды сигнала.

In [13]:

```
import numpy as np
import math
import matplotlib.pyplot as plt
N1 = 100
N2 = 200
R1 = 51
R2 = 4300
C = 0.00000022
r1 = 0.018
r2 = 0.009
```

In [48]:

```
l = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
am_15 = np.array([6.6*20, 4.2*50, 5*50, 5.5*50, 5.7*50, 5.8*50, 5.8*50, 5.6*50, 5.1*50, 4.2*50])
am_5 = np.array([4.1*10, 6.2*10, 7.7*10, 4.2*20, 4.3*20, 4.4*20, 4.4*20, 4.2*20, 3.8*20, 6.4*10 ])
print("max 15:", np.max(am_15))
print("max 5:", np.max(am_5))
```

```
max 15: 290.0
max 5: 88.0
```

In [47]:

```
am_5_2 = np.array([4.1*10, 6.3*10, 3.8*20, 4.2*20, 4.3*20, 4.4*20, 4.4*20, 4.2*20, 3.8*20, 6.4*10])
am_15_2 = np.array([6.4*20, 4*50, 4.9*50, 5.4*50, 5.6*50, 5.6*50, 5.6*50, 5.4*50, 5*50, 4.1*50])
print("max 15:", np.max(am_15_2))
print("max 5:", np.max(am_5_2))
```

```
max 15: 280.0
max 5 88.0
```

На канал  $Y$

подается напряжение с конденсатора  $C_2$

, включенного последовательно с  $C_1$

. Заряды на них по модулю равны **в любой момент времени**.

$$D = 12\pi \cdot 10^3 \frac{h_1 C_2}{S_1} U_y$$

в СГС

1

.Найдем  $D_r$

и  $E_c$

по параметрам петли, измеренным в пункте 18.

In [17]:

```
Dr = 12*np.pi*1000*h*C2*11*5/(5*S)
Ec = (R1+R2)/(300*R2*5*h)*12
print("D_r =", round_(Dr), "E_c =", round_(Ec))
```

```
D_r = 9281.1623 E_c = 0.08095
```

2

. Построим график зависимости  $D_m$

от  $E_m$

In [143]:

```
plt.figure(figsize=(20,16))
plt.subplot(2,2,1)
print("am_15,1")

D1 = np.array(list(map(round_, 12*np.pi*1000*h*C2*am_15/(5*S))))
E = np.array(list(map(round_, np.sqrt(2)*1/(300*h))))
plt.scatter(E, D, color = "violet", linewidth = 5)
```

```

plt.plot(E, D,linewidth=3)
plt.xlabel("E",size=10)
plt.ylabel("D", size=10)
plt.title("D(E)",size=10)
plt.grid(ls='--')

plt.subplot(2,2,2)

print("am_5,1")
D2 = np.array(list(map(round_, 12*np.pi*1000*h*C2*am_5/(5*S))))
plt.scatter(E, D2, color = 'g', linewidth = 5)
plt.plot(E, D2,linewidth=3)
plt.xlabel("E",size=10)
plt.ylabel("D", size=10)
plt.title("D(E)",size=10)
plt.grid(ls='--')

plt.subplot(2, 2, 3)

print("am_15_2,1")
D3 = np.array(list(map(round_, 12*np.pi*1000*h*C2*am_15_2/(5*S))))
plt.scatter(E, D3, color = "orange", linewidth = 5)
plt.plot(E, D3,linewidth=3)
plt.xlabel("E",size=10)
plt.ylabel("D", size=10)
plt.title("D(E)",size=10)
plt.grid(ls='--')

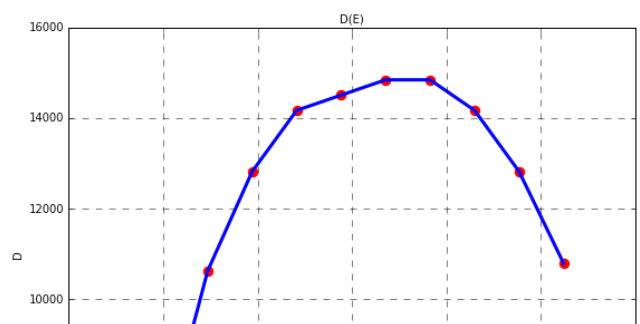
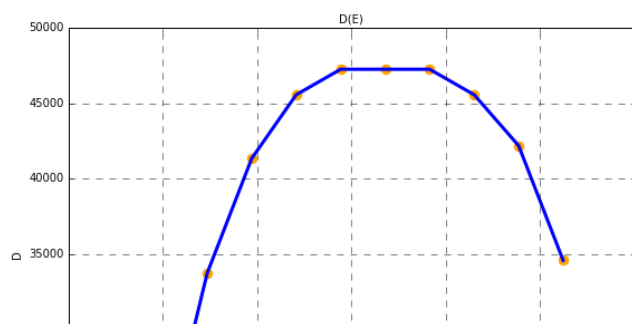
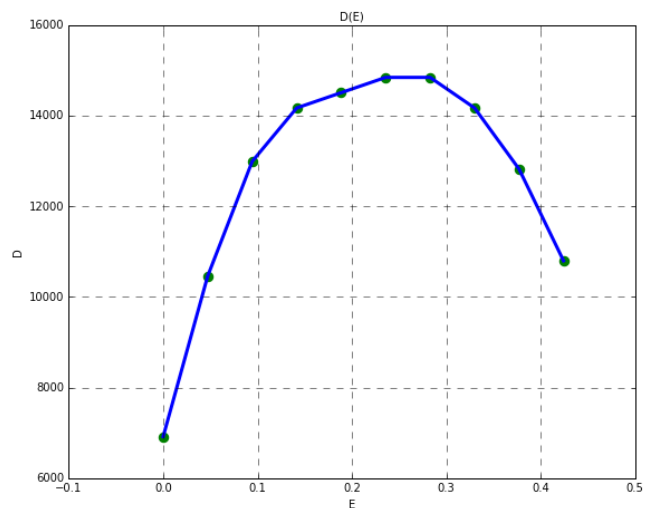
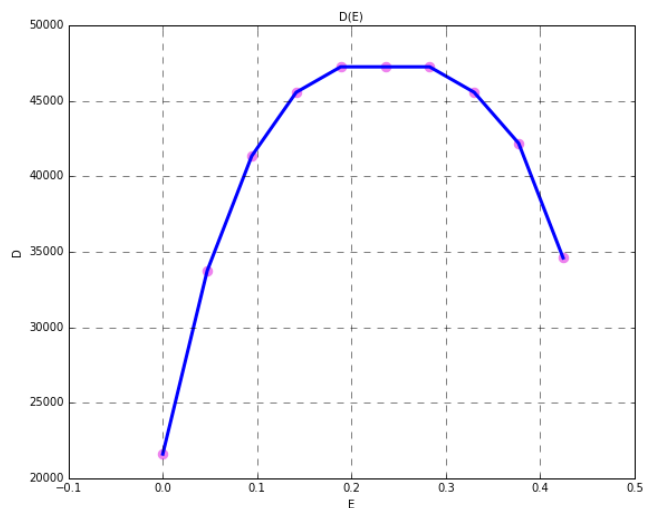
plt.subplot(2, 2, 4)

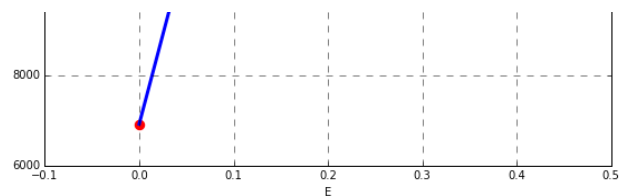
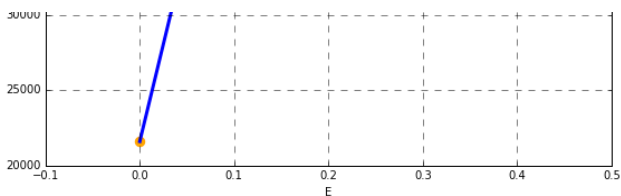
D4 = np.array(list(map(round_, 12*np.pi*1000*h*C2*am_5_2/(5*S))))
plt.scatter(E, D4, color = 'r', linewidth = 5)
plt.plot(E, D4,linewidth=3)
plt.xlabel("E",size=10)
plt.ylabel("D", size=10)
plt.title("D(E)",size=10)
plt.grid(ls='--')
print("am_5_2,1")

plt.show()

```

am\_15,1  
am\_5,1  
am\_15\_2,1  
am\_5\_2,1





3

. Построим график зависимости диэлектрической проницаемости

$\epsilon$  от

$E$  при ее увеличении. Знаем, что  $\epsilon = \frac{D}{E}$

In [158]:

```
am_15 = np.array([6.6*20, 4.2*50, 5*50, 5.5*50, 5.7*50, 5.8*50])
am_5 = np.array([4.1*10, 6.2*10, 7.7*10, 4.2*20, 4.3*20, 4.4*20])
am_5_2 = np.array([4.1*10, 6.3*10, 3.8*20, 4.2*20, 4.3*20, 4.4*20])
am_15_2 = np.array([6.4*20, 4*50, 4.9*50, 5.4*50, 5.6*50, 5.6*50])
D1 = np.array(list(map(round_, 12*np.pi*1000*h*C2*am_15/(5*S))))
D2 = np.array(list(map(round_, 12*np.pi*1000*h*C2*am_5/(5*S))))
D3 = np.array(list(map(round_, 12*np.pi*1000*h*C2*am_15_2/(5*S))))
D4 = np.array(list(map(round_, 12*np.pi*1000*h*C2*am_5_2/(5*S))))
```

In [162]:

```
def draw_eps_plot(D,c):
    D = np.array(D[1::])
    _l = np.array([0, 1, 2, 3, 4, 5])
    E = np.array(np.sqrt(2)*_l/(300*h))
    E = np.array(E[1::])
    print(len(E), len(D))
    eps=D/E/2/100
    plt.scatter(E, D/E, color = c, linewidth = 3)
    plt.plot(E, D/E,linewidth=2)
    plt.xlabel("E",size=10)
    plt.ylabel("D/E", size=10)
    plt.title("D/E(E)",size=10)
    plt.grid(ls='--')
    print("Максимальная диэлектрическая проницаемость: ", round(eps.max()))
plt.figure(figsize=(20,16))
plt.subplot(2,2,1)
draw_eps_plot(D1, c='r')
plt.subplot(2,2,2)
draw_eps_plot(D2, c='g')
plt.subplot(2,2,3)
draw_eps_plot(D3,c='b')
plt.subplot(2,2,4)
draw_eps_plot(D4,c="orange")
print("Максимальное значение диэлектрической проницаемости -- 3759")
```

5 5

Максимальная диэлектрическая проницаемость: 3759.0

5 5

Максимальная диэлектрическая проницаемость: 1110.0

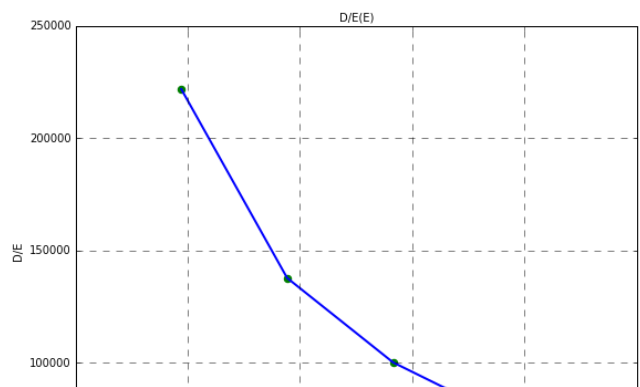
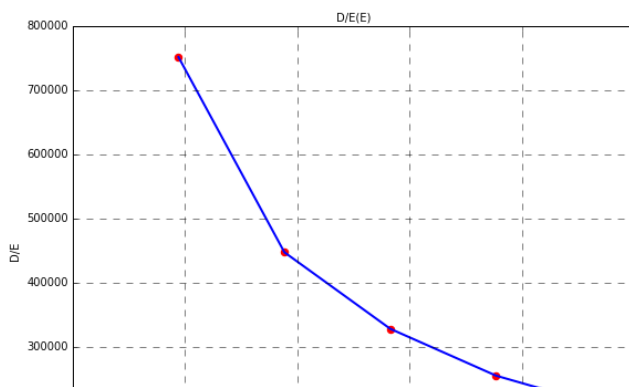
5 5

Максимальная диэлектрическая проницаемость: 3580.0

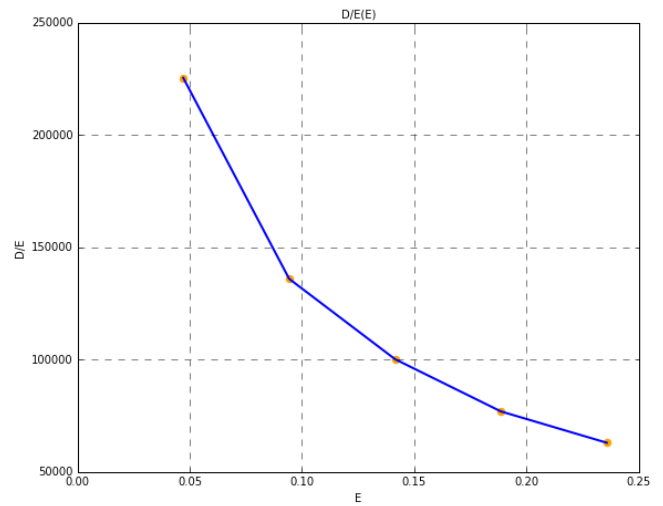
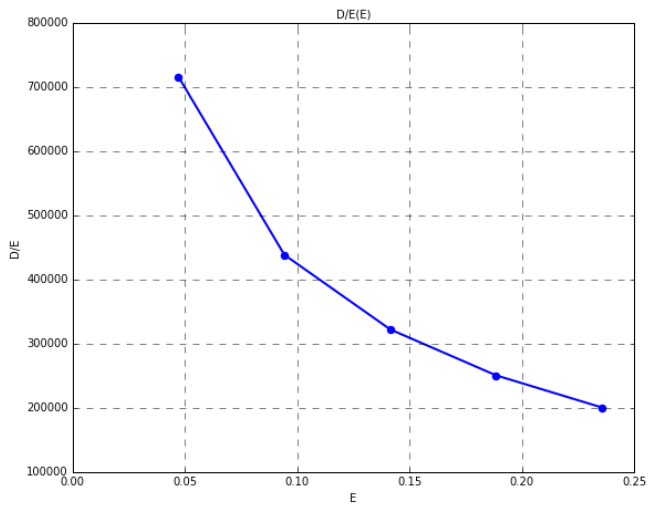
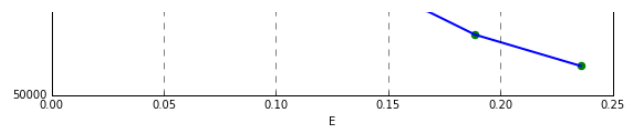
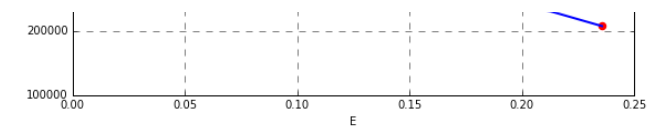
5 5

Максимальная диэлектрическая проницаемость: 1128.0

Максимальное значение диэлектрической проницаемости -- 3759







In [161]:

```
Dh = 12*np.pi*1000*h*C2*13.5*5/(5*S)
Eh = (R1+R2)/(300*R2*5*h)*15
print("Поляризация в насыщенном состоянии: ", round((Dh-Eh)/(4*np.pi),2))
```

Поляризация в насыщенном состоянии: 906.42

In [ ]:

```
Dr = 12*np.pi*1000*h*C2*11*5/(5*S)
Ec = (R1+R2)/(300*R2*5*h)*12
print("D_r: ", Dr, '\nE_c: ', Ec)
```