電位分布（高校生のための）

2021 年 2 月 23 日

S.Matoike

目次

第 1 章　電位の分布 2
1.1　実験の目的 2
1.2　実験 2
1.2.1　実験の手順 2
1.2.2　-5[V] を作るには 2
1.2.3　等電位面（線）の作図 3

第 2 章　電界の強さと方向 5
2.1　電流の流れる方向 5

第 3 章　模擬実験 8
3.1　格子間隔を無限に小さく (微分) 8
3.2　内部の点の離散化 9

第 4 章　補助資料 13

第 1 章

電位の分布

1.1　実験の目的

1.2　実験

実験１　直流安定化電源の +5[V] と GND の端子に接続した「棒」を電極として電場を作る
実験２　直流安定化電源の +5[V] と GND の端子に接続した「板」を電極として電場を作る
実験３　+5[V],GND,−5[V] の 3 つの電圧端子を用意し、「棒」と「板」を電極として電場を作る
　　格子点上の電位分布を測定し、等電位線を作図する。また、電場（電界）の強さと向きも求める

1.2.1　実験の手順

実験１の場合の手順は次の通り（実験 2 と実験 3 の手順は各自で）

1. バットに水道の水を張って、浅い水槽をつくる
2. 直流安定化電源の出力が +5[V] となるように、回路計の値を読みながら調整する
3. 直流安定化電源の +5[V] と GND の端子を、電極にする棒と接続し水槽に立てる
4. 格子点の電位を回路計の値を読んで記録する
5. 等電位線を作図する

1.2.2　-5[V] を作るには

2 台の直流安定化電源装置を使って、＋５ [V]、０ [V]、ー５ [V]　の 3 つの電極を作る方法を考えてみます（実験 3 のために必要になる）

1.2.3　等電位面（線）の作図

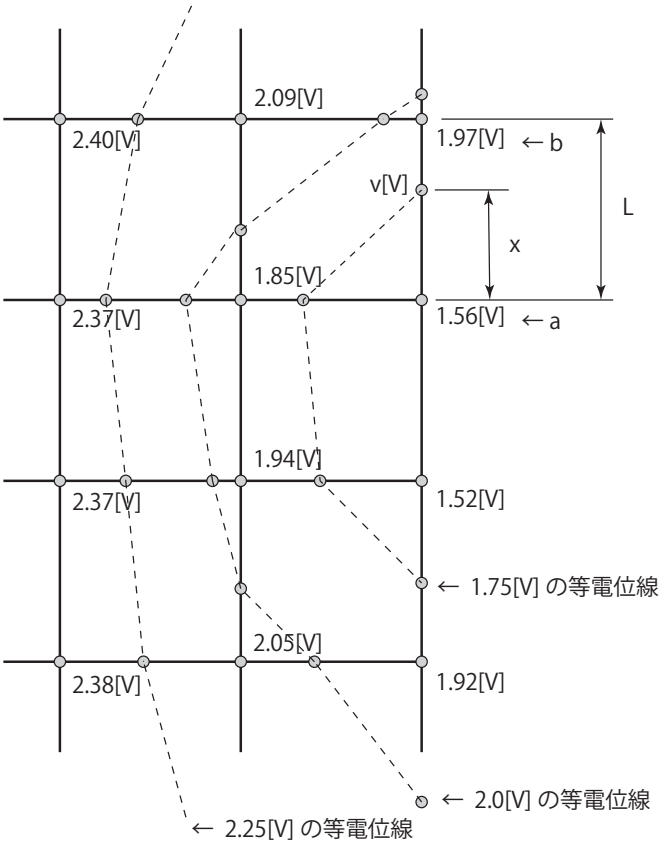
例として、$v = 1.75$[V] の等電位面（線）を作図する場合の手順を説明します

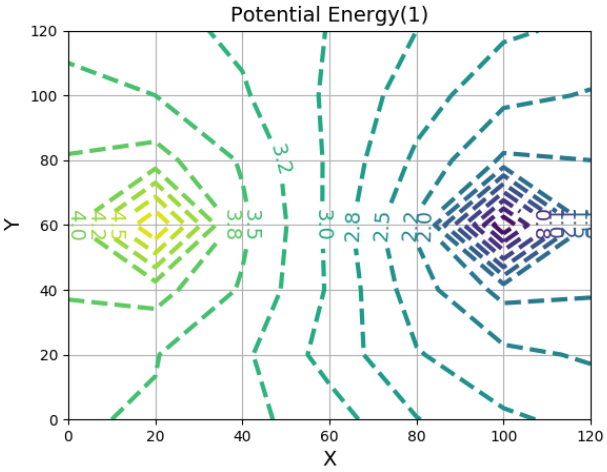
1. $v = 1.75$[V] が存在すると思われる格子線を見つけます
2. その格子線の端点で測定した電圧の値を見て、
   電圧の小さい方の端点の電圧を a[V]、大きい方の端点の電圧を b[V] とします
3. 格子線の長さ L は、この実験では共通です。実際に測定して求めます $L =$ [mm]
4. 比例配分によって、当該格子線上の $v = 1.75$[V] の場所 x[mm] を求めます (a< v < b)

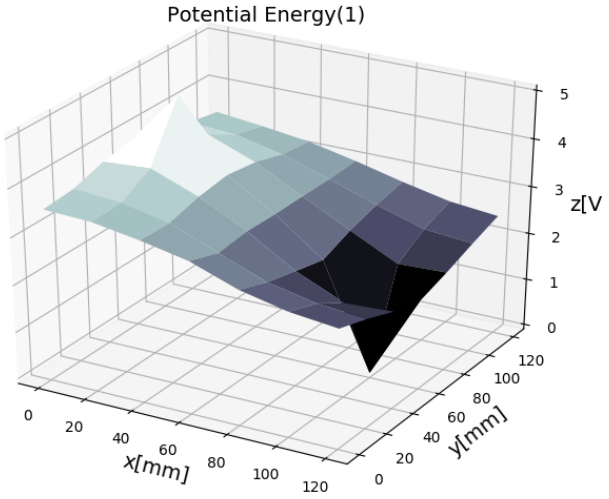
$$x = \frac{v - a}{b - a} \times L$$

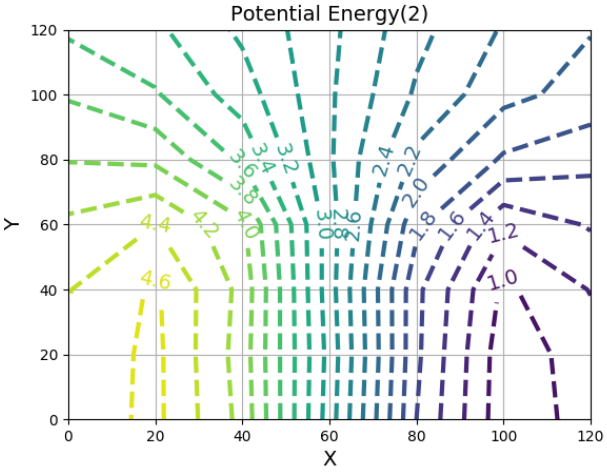
5. 電圧の小さい方の点 a から x[mm] 離れた格子線上の点を $v = 1.75$[V] の場所としてプロットします。この場合、 x は必ず 0< x < L になります

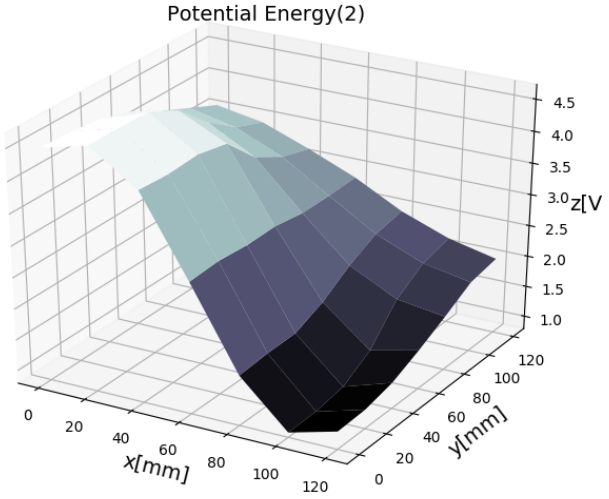
格子面上に $v = 1.75$[V] の所在が無くなるまで上記を繰り返し、見つけた点を直線で順に接続していくと、それが $v = 1.75$[V] の等電位面（線）になります。等電位面（線）は、0[V]〜5[V] を、0.25[V] 間隔（大変なら 0.5[V] 間隔でもいいかな）で描くことにします。

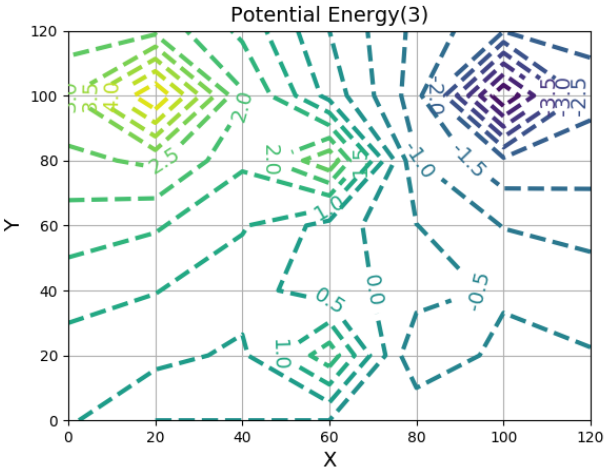


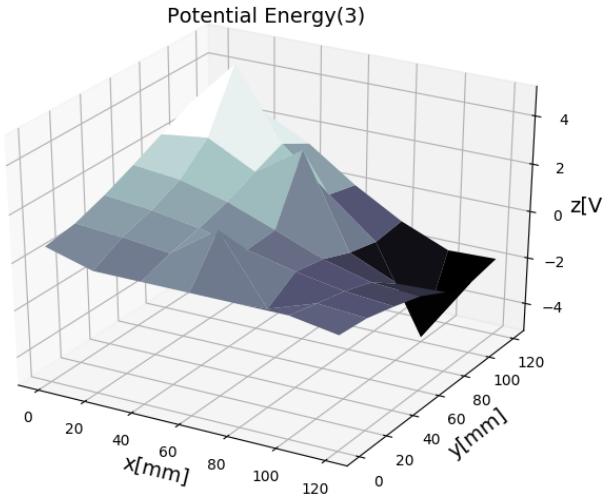

Potential Energy(1)
X
Y


Potential Energy(1)
x[mm]
y[mm]
z[V


Potential Energy(2)
X
Y


Potential Energy(2)
x[mm]
y[mm]
z[V


Potential Energy(3)
X
Y


Potential Energy(3)
x[mm]
y[mm]
z[V

第 2 章

電界の強さと方向

予め何らかの方法によって作られた電気の力の場（電界）がある。その場の中に 1[Coulomb] の電荷を置いた。この時、その電荷が受ける力の大きさが E[Newton] であったとすると、その電荷を置いた場所 $\boldsymbol{x}(x,y)$ における電界の強さは $\boldsymbol{E}(\boldsymbol{x})$[Newton/Coulomb] であると言う。

場所 $\boldsymbol{x}(x,y)$ における電界の強さは、その周囲の電位 $\phi(\boldsymbol{x})$ の様子から知ることができる。具体的には、場所 $\boldsymbol{x}(x,y)$ の周辺における電位 $\phi(\boldsymbol{x})$ の変化が大きいほど、その場所に置いた電荷が受ける力（＝電界の強さ）が大きい。逆に、その場所 $\boldsymbol{x}(x,y)$ の周辺の電位 $\phi(\boldsymbol{x})$ の変化が小さい場合には、その場所に置いた電荷が受ける力（＝電界の強さ）は小さい。

また、電界から電荷が力を受けると、電荷はその力の方向に向かって動かされる事になる。この電荷の動きは、電流の流れに他ならない。電位の変化の無い（＝電位が等しい）場所では、電荷が受ける力（＝電界の強さ）はゼロであるため電流は流れない。

2.1　電流の流れる方向

実験を行った格子面上で、例として⑧の場所における電界の強さと方向を求める手順を見てみよう。

格子面上の⑧の場所 $\boldsymbol{x}(x,y)$ における電位 $\phi(\boldsymbol{x})$ の変化を、⑧の周辺における電位の測定値である ϕ_0、ϕ_x、ϕ_y を使って近似的に求めることにする。

まず、電界の強さ E の x 方向成分である $E_x(x)$ について考えることにしよう。x 方向（横方向）の電位の変化は、近傍の 2 つの格子点（y 座標は同じだが x 座標は隣り合う格子点）上の電位、ϕ_x と ϕ_0 の差、（即ち $\phi_0 - \phi_x$）によって知ることにする。次に、電界の強さ E の y 方向成分である $E_y(y)$ について考えると、y 方向（縦方向）の電位の変化は、近傍の 2 つの格子点（x 座標は同じだが y 座標は隣り合う格子点）上の電位、ϕ_y と ϕ_0 の差、（即ち $\phi_y - \phi_0$）によって知ることにする。

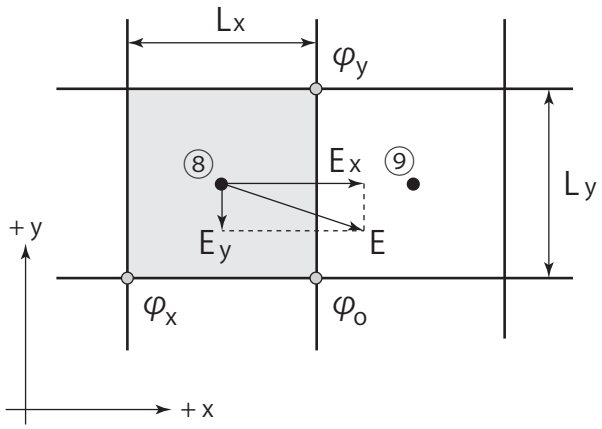
電位の差が同じ大きさであったとしても、それが短い距離の間で急に生じている変化量なのか、長い格子間隔でようやく生じた変化量なのかを考えると、前者は大きな（＝急激な）変化であり、後者は小さな（＝ゆるやかな）変化であると捉えることができる。

以上の考察により、電界の強さ $\boldsymbol{E}$ の x 成分 E_x と、y 成分 E_y は、次の様にして求めることができる。

$$E_x = -\frac{\phi_0 - \phi_x}{L_x}$$

$$E_y = -\frac{\phi_y - \phi_0}{L_y}$$

$$|E| = \sqrt{{E_x}^2 + {E_y}^2}$$

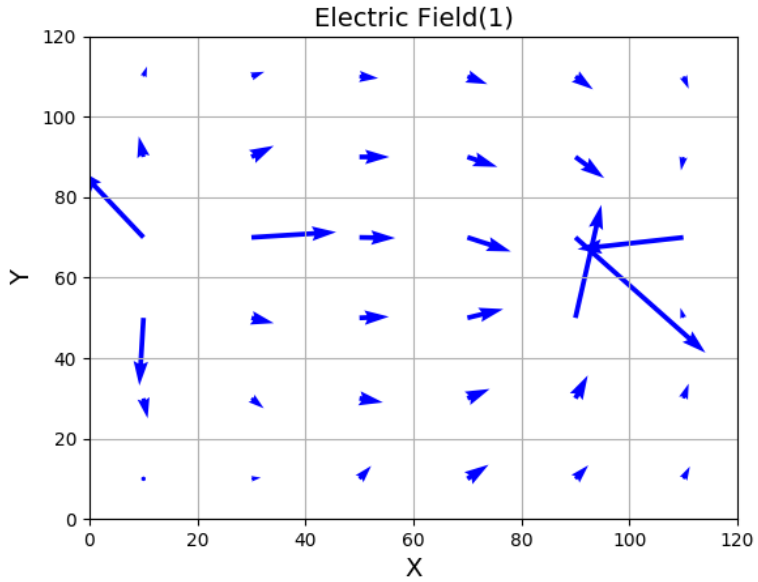


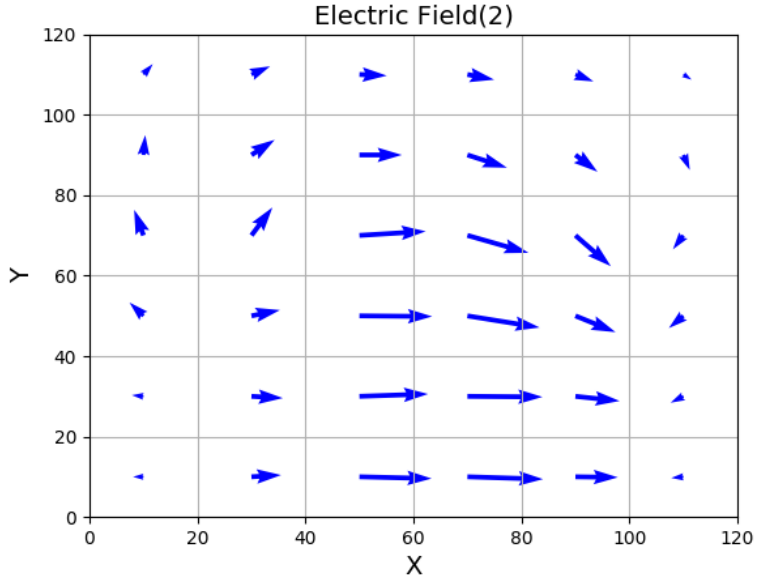
点⑧に置かれた 1[Coulomb] の電荷は、
x 方向（横方向）に E_x の力を受けると同時に、y 方向（縦方向）にも E_y の力を受ける。
その結果、電荷は E の方向へ動こうとする。電荷の移動、これは即ち「電流」の流れである。

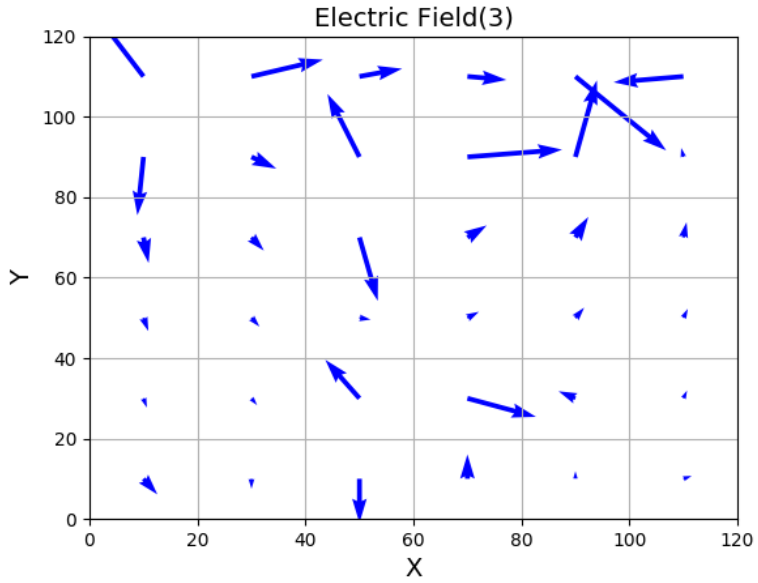
電界の強さは、1[Coulomb] 当たりの電荷に働く力 [Newton] なので、E の単位は [Newton/Coulomb] である。一方これまでの考察から電界 E_x の大きさは、x 方向の単位長さ 1 メートル [Meter] 当たりの電圧 ϕ[Volt] の変化量、そして電界 E_y の大きさも、y 方向の単位長さ 1 メートル [Meter] 当たりの電圧 ϕ[Volt] の変化量ですから、E の単位は [Volt/Meter] と表すこともできる。

もっと正確に説明すると、1[Volt] = 1[Joule/Coulomb] を用いることによって、電界 E の単位は、1[Newton/Coulomb] = 1[(Joule/Meter)/Coulomb] = 1[Volt/Meter] となります。こちらの単位の方がよく使われます。

これまで、1[Coulomb] の電荷に対して作用する力 E[Newton] が分布する場（電界）について考察してきたが、q[Coulomb] の電荷を置いた場合は、クーロンの法則にもある通り、その電荷は qE[Newton] の力を受けることになる。


Electric Field(1)
120
100
80
60
40
20
0
Y
0
20
40
60
80
100
120
X


Electric Field(2)
120
100
80
60
40
20
0
Y
0
20
40
60
80
100
120
X


Electric Field(3)
120
100
80
60
40
20
0
Y
0
20
40
60
80
100
120
X

第 3 章

模擬実験

3.1　格子間隔を無限に小さく (微分)

x 方向（横方向）の電位 ϕ の変化、及び y 方向（縦方向）の電位 ϕ の変化は、数学ではそれを（偏）微分の演算子を使って記述できます。格子間隔の L_x や L_y を極限まで短くして、非常に細かな格子にしたときに以下のような微分演算子による電場の記述が可能になります。また、ϕ は場所 (x, y) によって決まる値を取る。（このことを「ϕ は x と y の関数である」と言い、$\phi(x, y)$ と記述します）

x と y の関数である $\phi(x, y)$ において、y を同じ値のまま変えずに x だけの変化量を得るには、偏微分の演算子 $\partial/\partial x$ を用いて $\partial\phi/\partial x$ と書きます。同様に、x を同じ値のまま変えずに y だけの変化量を知る場合も、偏微分の演算子 $\partial/\partial y$ を使って $\partial\phi/\partial y$ と書きます。（本実習では、L_x も L_y も長くてとても粗い格子ですが、次の式の右辺の様な大雑把な概算をして電場のイメージを描画してみたものです。）

$$E_x = -\frac{\partial}{\partial x}\phi \approx -\frac{\phi_0 - \phi_x}{L_x}$$

$$E_y = -\frac{\partial}{\partial y}\phi \approx -\frac{\phi_y - \phi_0}{L_y}$$

ここに現れた偏微分の演算子を、まとめて ∇（ナブラと読む）という記号で書くことがあります。

$$\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$$

この演算子 ∇ を、電位 ϕ のようなスカラ量に対して作用させる場合、特に grad（グラディエントと読む。日本語では「勾配」）という記号を使います。これを使うと、上記の電界の強さ E_x や E_y が電位 ϕ の変化量だとする式は次の様に書けます。

$$\boldsymbol{E} = \nabla(-\phi) = -\mathrm{grad}\phi$$

また同じ演算子 ∇ を、電界 $\boldsymbol{E} = (E_x, E_y)$ のようなベクトル量に対して作用させる場合は、特に div（ダイバージェンスと読む。日本語では「発散」）という記号が使われます。例えば次の式は、電荷を内部に含む微少体積を貫いて出入りする電気力線（電界 $\boldsymbol{E}$ の方向と一致）の収支が、その微少体積内部の電荷量に比例するとするガウスの法則の微分形式です。（ρ は電荷密度分布、ε_0 は真空の誘電率、$\boldsymbol{D}$ は電束密度（電気変位とも言う））もし右辺がゼロならば、微少体積に入ってくる電気力線は、全てそのまま出て行く事になり、変化量は無し、収支ゼロになりますが、電界では収支ゼロになりません。（これを「湧きだしがある」といいます）因みに磁界の場合、磁束密度 $\boldsymbol{B}$ の収支はゼロになります。（つまり、$\mathrm{div}\boldsymbol{B} = 0$、

湧きだしなしです）

$$\nabla \boldsymbol{E} = \mathrm{div}\boldsymbol{E} = \frac{\rho}{\varepsilon_0}, \qquad \mathrm{div}\boldsymbol{D} = \rho \qquad \text{ここで } \boldsymbol{D} = \varepsilon_0 \boldsymbol{E}$$

∇ も grad も div も、どれも「〜の（位置座標に関する）変化量」という意味では同じです。つまり、gradϕ は「ϕ の変化量」という意味であり、div$\boldsymbol{E}$ は「$\boldsymbol{E}$ の変化量」という意味になります。

grad も div も ∇ と同じ格好をした演算子ですから、grad に div を作用させることを、あたかもナブラ ∇ 同士の内積と同じ様に扱って、形式的に演算を進めることができます。

$$\mathrm{div}\boldsymbol{E} = \mathrm{div}(-\mathrm{grad}\phi) = -\nabla \cdot \nabla\phi = -\nabla^2\phi = -(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})\phi = -\Delta\phi = \frac{\rho}{\varepsilon_0}$$

演算子 Δ は、ラプラシアン（ラプラスの演算子）と呼ばれています。

$$\Delta = \nabla \cdot \nabla = (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})$$

次の式で、前者はラプラス (Laplace) の方程式、後者はポアソン (Poisson) の方程式と呼ばれています。

$$\Delta\phi = 0$$

$$\Delta\phi = -\frac{\rho}{\varepsilon_0}$$

これが、電界（電場）を記述している式の 1 つになります。

3.2　内部の点の離散化

ここまでは実験を通して実際に測定した格子点上の電位を元に、等電位面（線）を求めてきました。

ここからは、ポアソンの方程式、あるいは電荷密度の分布していない電場を表すラプラスの方程式からスタートして、計算機による模擬実験（シミュレーション）を行い、電位の分布つまり等電位面（線）の描画をしてみます。

計算機による模擬実験が、どの程度まで実際の現象を再現できるものなのか、実験データによって測定した電位分布と、ラプラスの方程式を計算機で解いて求めた電位分布とを比べてみることにしましょう。

ポアソンの方程式を再掲すると次の通りです。

$$-\varepsilon\Delta\phi = -\varepsilon(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})\phi = \rho$$

この方程式を計算機で解くためには、この式を離散化するという作業が必要になります。ここでは差分法による離散化を行います。

下の図に示す様に、格子点 P に着目したコントロール・ボリューム（斜線部）における収支を考えます。z 方向に単位長さ 1 を持つ面 w と、同様の面 s から流入する量はそれぞれ次の様になります。

$$-\varepsilon\frac{\partial\phi}{\partial x}\,|_w\,(\Delta y \cdot 1), \qquad -\varepsilon\frac{\partial\phi}{\partial y}\,|_s\,(\Delta x \cdot 1)$$

同じ様にして、面 e と面 n から流出する量は、それぞれ次の通り表せます。

$$-\varepsilon\frac{\partial\phi}{\partial x}\,|_e\,(\Delta y \cdot 1), \qquad -\varepsilon\frac{\partial\phi}{\partial y}\,|_n\,(\Delta x \cdot 1)$$

この 4 つの総和がここで着目しているコントロール・ボリュームにおける収支であり、今はこの中に電荷分布の存在を考えていませんから、4 つの総和はゼロになります。これはラプラスの方程式ですね。

$$-\varepsilon\left\{(\Delta y\cdot 1)(\frac{\partial\phi}{\partial x}\mid_w-\frac{\partial\phi}{\partial x}\mid_e)+(\Delta x\cdot 1)(\frac{\partial\phi}{\partial y}\mid_s-\frac{\partial\phi}{\partial y}\mid_n)\right\}=0$$

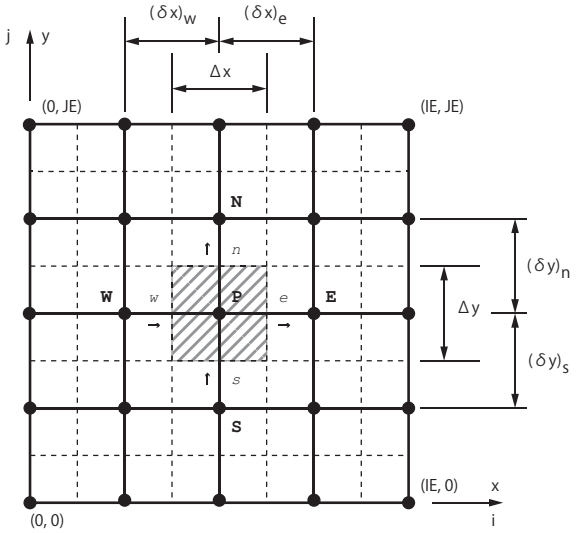
これまでの実習の中で行ってきたのと同じ様にして、この式に現れた偏微分（変化量）を、近傍の格子点の値の差を使って表すことにします。

$$\frac{\partial\phi}{\partial x}\mid_w=\frac{\phi_P-\phi_W}{(\delta x)_w}$$

$$\frac{\partial\phi}{\partial x}\mid_e=\frac{\phi_E-\phi_P}{(\delta x)_e}$$

$$\frac{\partial\phi}{\partial y}\mid_s=\frac{\phi_P-\phi_S}{(\delta y)_s}$$

$$\frac{\partial\phi}{\partial y}\mid_n=\frac{\phi_N-\phi_P}{(\delta y)_n}$$



これを先ほどの、コントロール・ボリュームにおける収支はゼロという式に代入します。ただし、格子は規則的に設けることにして、$(\delta x)_w=(\delta x)_e=\Delta x$、$(\delta y)_s=(\delta y)_n=\Delta y$ と置き換えて計算を進めることにします。

$$\Delta y(\frac{\phi_P-\phi_W}{\Delta x}-\frac{\phi_E-\phi_P}{\Delta x})+\Delta x(\frac{\phi_P-\phi_S}{\Delta y}-\frac{\phi_N-\phi_P}{\Delta y})=0$$

P 点の電位 ϕ_P について整理すると、

$$\phi_P=\frac{\Delta y^2}{2(\Delta x^2+\Delta y^2)}(\phi_E+\phi_W)+\frac{\Delta x^2}{2(\Delta x^2+\Delta y^2)}(\phi_S+\phi_N)$$

この式を見ると、P 点の電位 ϕ_P は、P 点を取り囲む東西南北の各格子点、E 点の電位 ϕ_E、W 点の電位 ϕ_W、S 点の電位 ϕ_S、N 点の電位 ϕ_N の 4 つから（平して）求めるという形になっています。

ここまでの状態でプログラムして計算させることもできますが、実際にはこれを次の様に、無次元化と呼ばれる式の変形を施した上でプログラムするようにします。

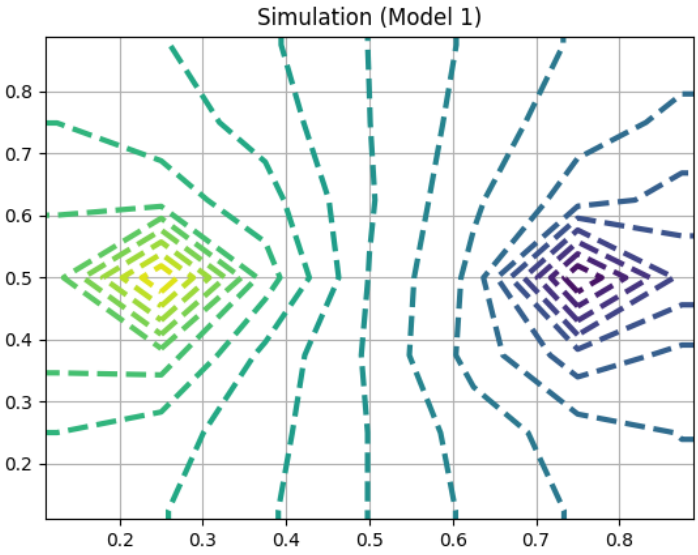
$$T_{i,j}=\frac{A^2\Delta Y^2}{2(\Delta X^2+A^2\Delta Y^2)}(T_{i-1,j}+T_{i+1,j})+\frac{A^2\Delta X^2}{2(\Delta X^2+A^2\Delta Y^2)}(T_{i,j-1}+T_{i,j+1})$$

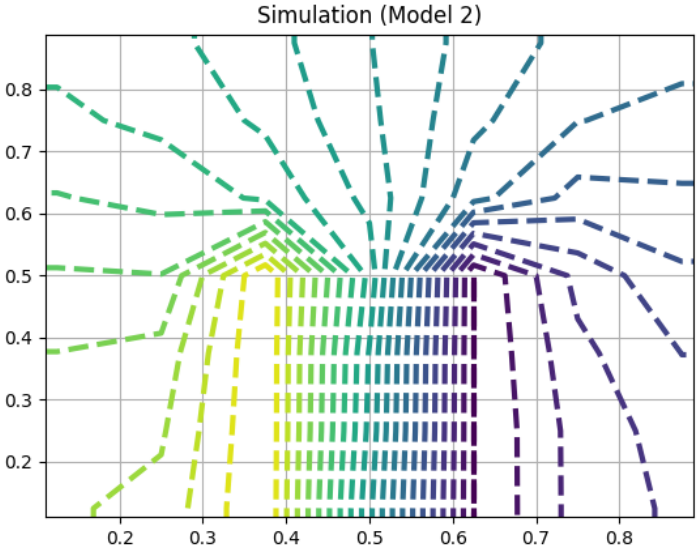
ここで、$A=y_{max}/x_{max}$（縦横比）、$\Delta X=1/IE$、$\Delta Y=1/JE$ とします。

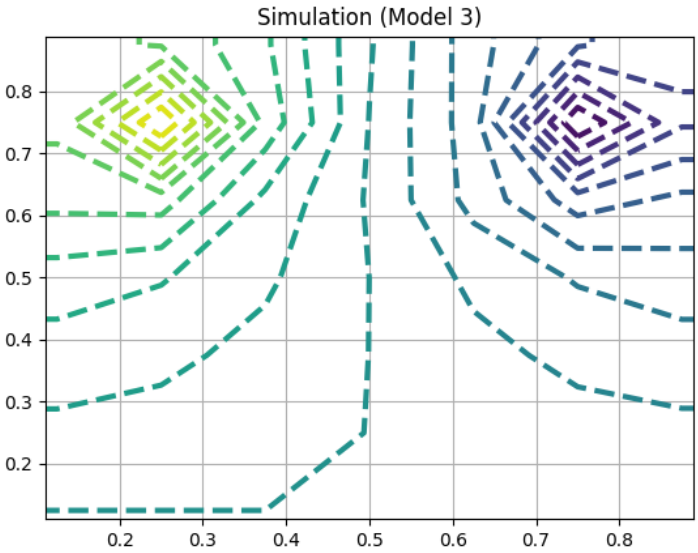
また、$T_{i,j}=(\phi_{i,j}-\phi_{min})/(\phi_{max}-\phi_{min})$、$X=x/x_{max}$、$Y=y/y_{max}$ なので、これによって模擬実験対象のモデルの領域は $0<X\leq 1$、$0<Y\leq 1$、$0<T\leq 1$ の範囲で計算されることになります。

計算は、まず対象領域内部の $T_{i,j}$ の初期値を適当に仮定し、次に上の式で $T_{i,j}$（$i=1$〜$IE-1$、$j=1$〜$JE-1$）を順次修正していきます。各点の値が必要な桁数に収束するまで、繰り返しこの式で格子点の値を平していくことになります。

計算結果 $T_{i,j}$ から元の物理量 $\phi_{i,j}$ を得るには、$\phi_{i,j} = T_{i,j} \times (\phi_{max} - \phi_{min}) + \phi_{min}$ とすれば求まります。


Simulation (Model 1)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.2
0.3
0.4
0.5
0.6
0.7
0.8

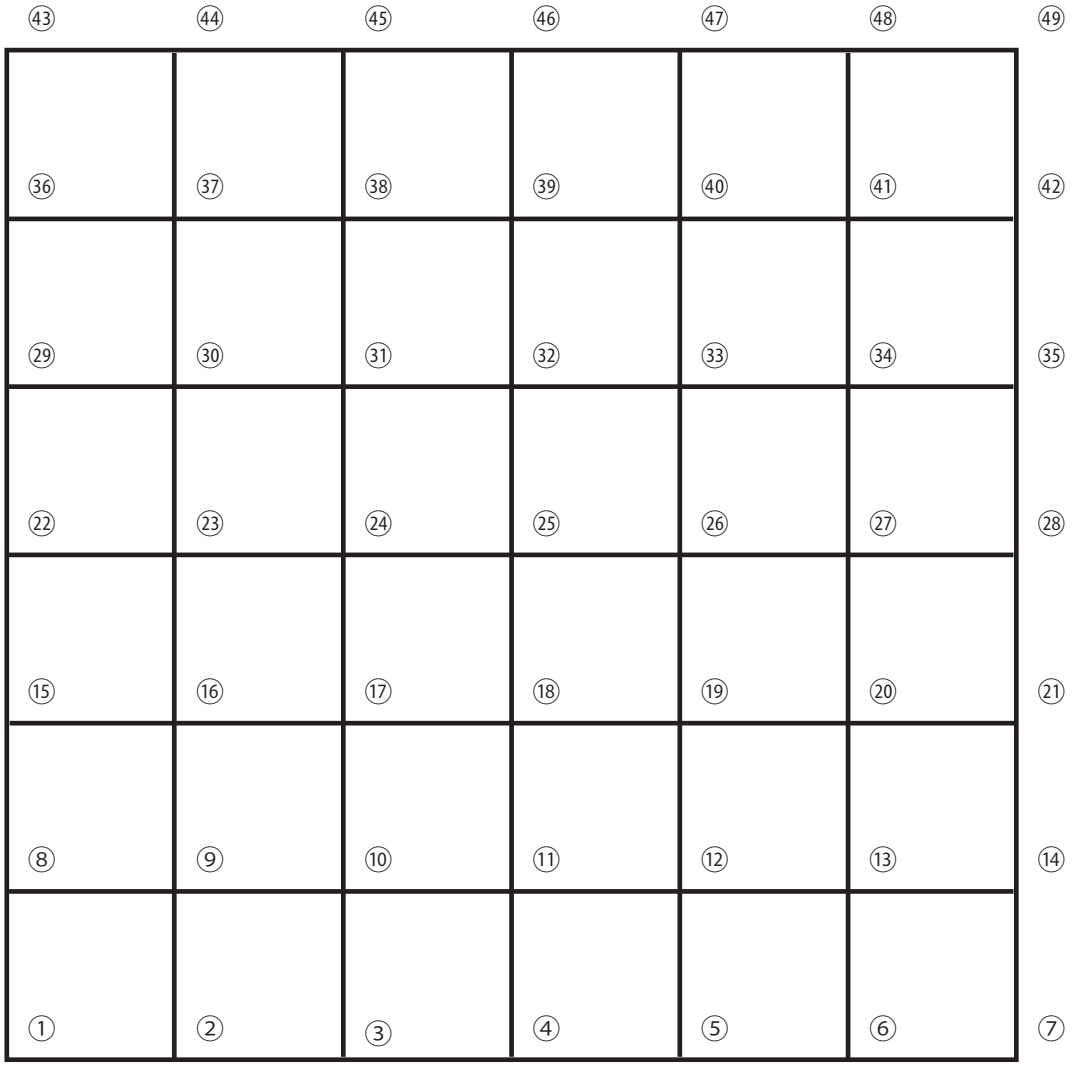

Simulation (Model 2)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.2
0.3
0.4
0.5
0.6
0.7
0.8


Simulation (Model 3)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.2
0.3
0.4
0.5
0.6
0.7
0.8

第 4 章

補助資料

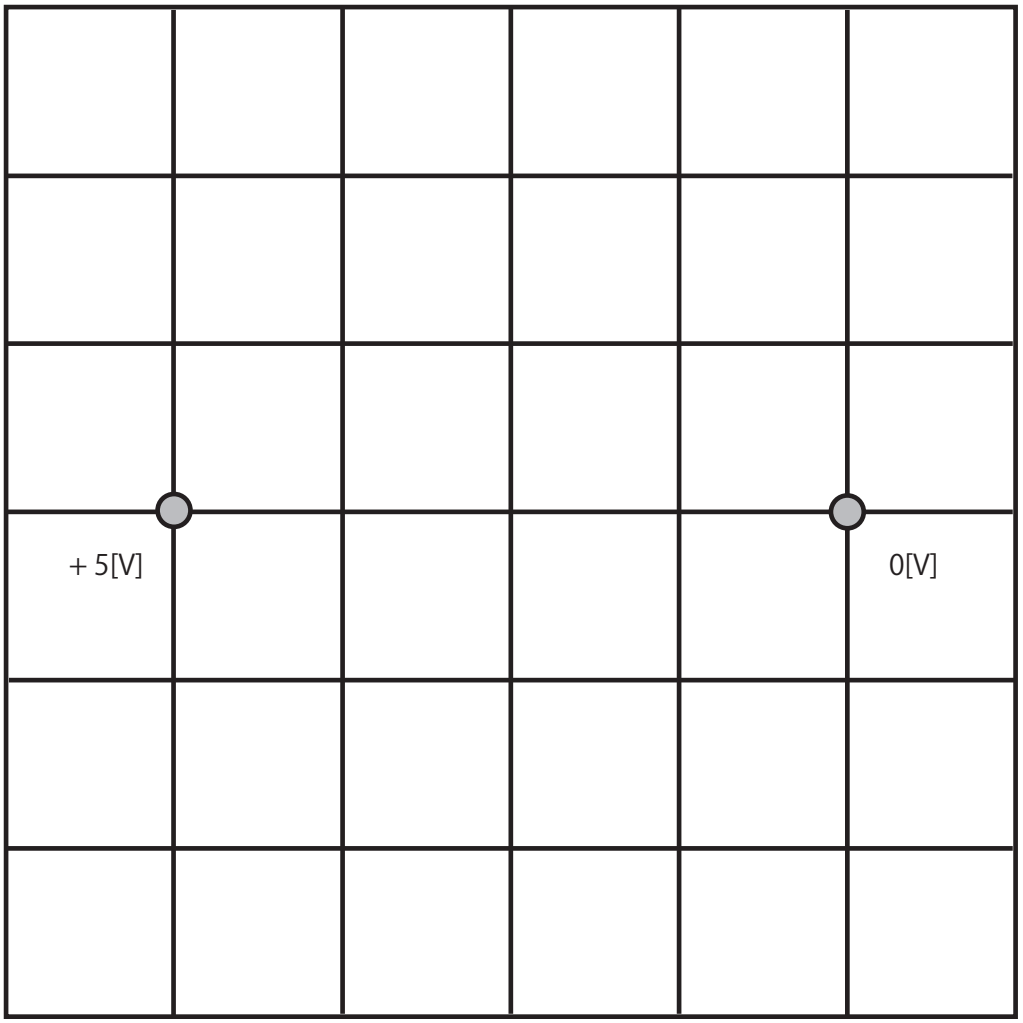
【電位分布】記録様式



①		⑪		㉑		㉛		㊶	
②		⑫		㉒		㉜		㊷	
③		⑬		㉓		㉝		㊸	
④		⑭		㉔		㉞		㊹	
⑤		⑮		㉕		㉟		㊺	
⑥		⑯		㉖		㊱		㊻	
⑦		⑰		㉗		㊲		㊼	
⑧		⑱		㉘		㊳		㊽	
⑨		⑲		㉙		㊴		㊾	
⑩		⑳		㉚		㊵			

単位 [V]

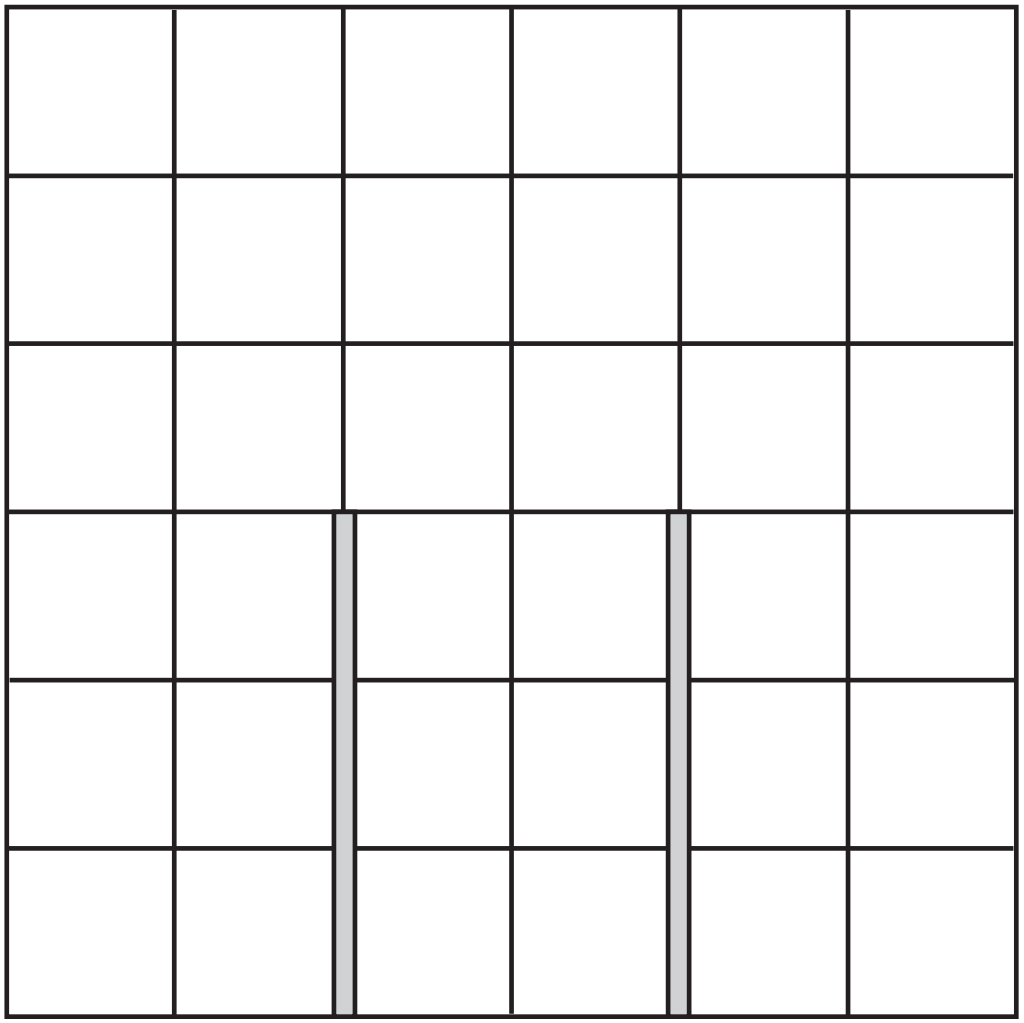
【実験 1 ：電位分布】



①		⑪		㉑		㉛		㊶	
②		⑫		㉒		㉜		㊷	
③		⑬		㉓	5.0	㉝		㊸	
④		⑭		㉔		㉞		㊹	
⑤		⑮		㉕		㉟		㊺	
⑥		⑯		㉖		㊱		㊻	
⑦		⑰		㉗	0.0	㊲		㊼	
⑧		⑱		㉘		㊳		㊽	
⑨		⑲		㉙		㊴		㊾	
⑩		⑳		㉚		㊵			

単位 [V]

【実験２：電位分布】

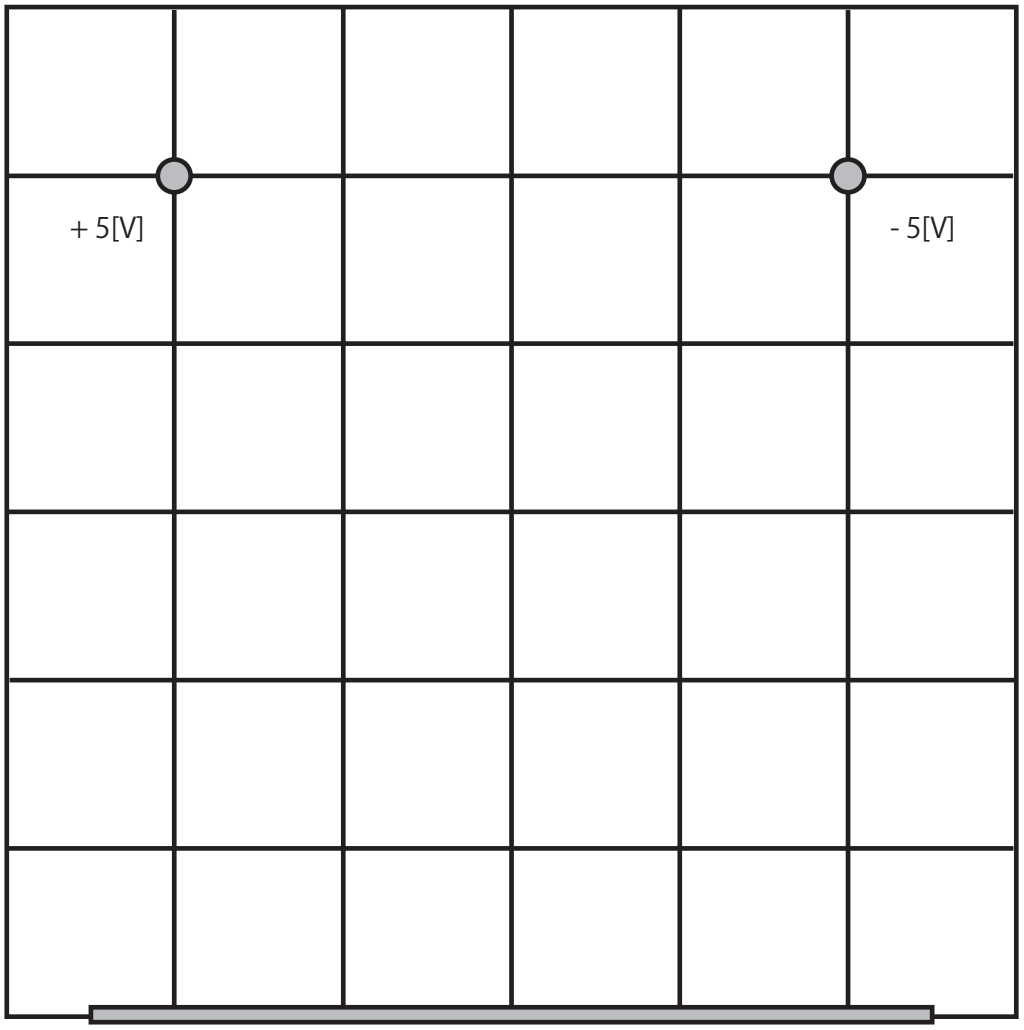


＋5[V]　　0[V]

①		⑪		㉑		㉛		㊶	
②		⑫	0.0	㉒		㉜		㊷	
③	5.0	⑬		㉓		㉝		㊸	
④		⑭		㉔	5.0	㉞		㊹	
⑤	0.0	⑮		㉕		㉟		㊺	
⑥		⑯		㉖	0.0	㊱		㊻	
⑦		⑰	5.0	㉗		㊲		㊼	
⑧		⑱		㉘		㊳		㊽	
⑨		⑲	0.0	㉙		㊴		㊾	
⑩	5.0	⑳		㉚		㊵			

単位 [V]

【実験３：電位分布】



①	0.0	⑪		㉑		㉛		㊶	- 5.0
②	0.0	⑫		㉒		㉜		㊷	
③	0.0	⑬		㉓		㉝		㊸	
④	0.0	⑭		㉔		㉞		㊹	
⑤	0.0	⑮		㉕		㉟		㊺	
⑥	0.0	⑯		㉖		㊱		㊻	
⑦	0.0	⑰		㉗		㊲	5.0	㊼	
⑧		⑱		㉘		㊳		㊽	
⑨		⑲		㉙		㊴		㊾	
⑩		⑳		㉚		㊵			

単位 [V]

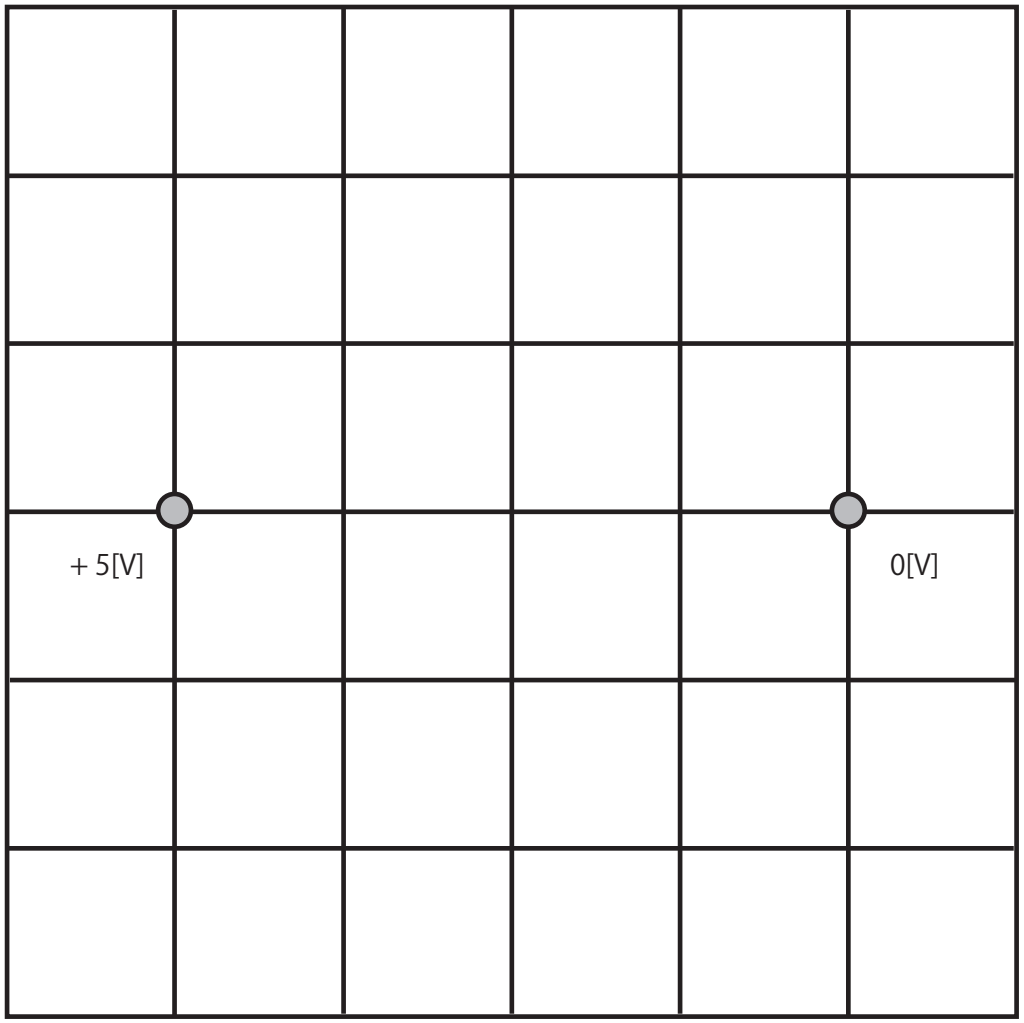
【電界の大きさと向き】記録様式

㉛	㉜	㉝	㉞	㉟	㊱
㉕	㉖	㉗	㉘	㉙	㉚
⑲	⑳	㉑	㉒	㉓	㉔
⑬	⑭	⑮	⑯	⑰	⑱
⑦	⑧	⑨	⑩	⑪	⑫
①	②	③	④	⑤	⑥

単位 [V/m]

㉛		㉜		㉝		㉞		㉟		㊱	
㉕		㉖		㉗		㉘		㉙		㉚	
⑲		⑳		㉑		㉒		㉓		㉔	
⑬		⑭		⑮		⑯		⑰		⑱	
⑦		⑧		⑨		⑩		⑪		⑫	
①	E_x= E_y=	②		③		④		⑤		⑥	

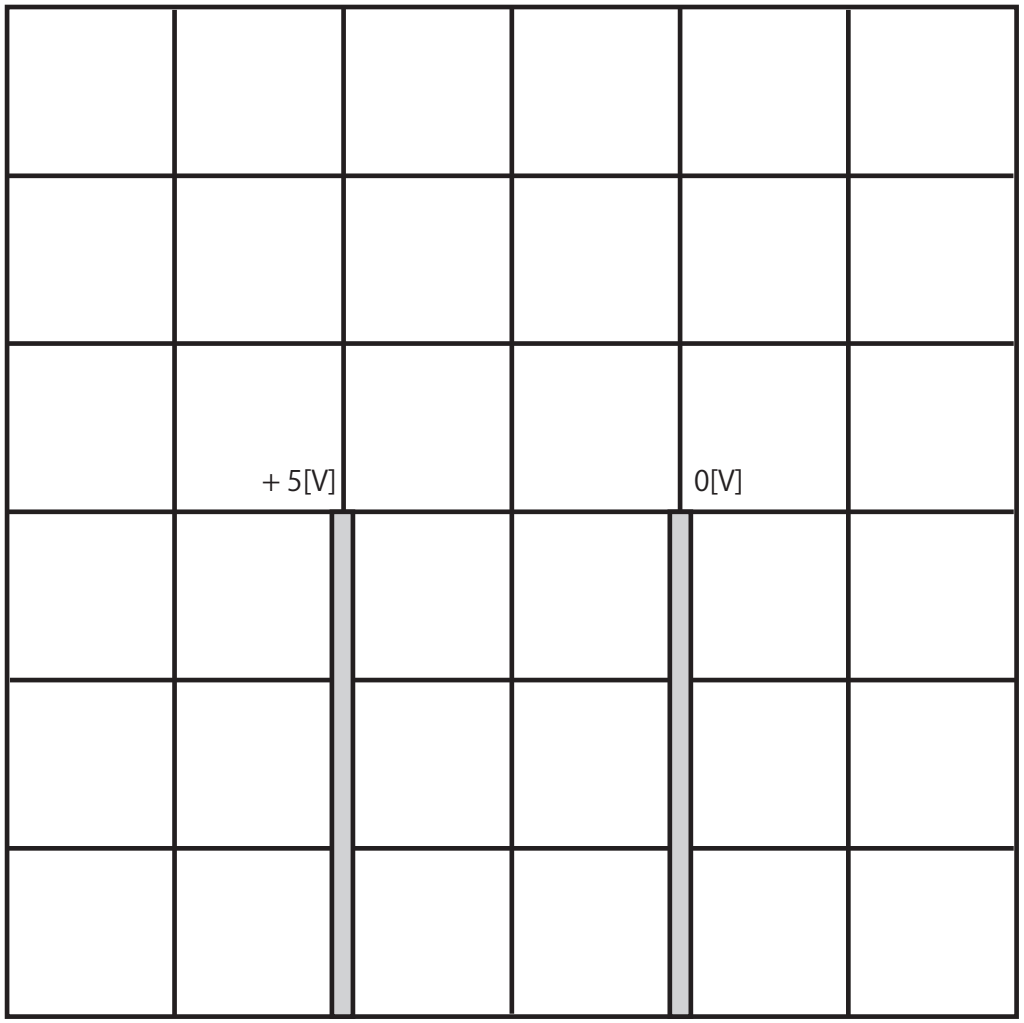
【実験１：電界の大きさと向き】



単位 [V/m]

㉛		㉜		㉝		㉞		㉟		㊱	
㉕		㉖		㉗		㉘		㉙		㉚	
⑲		⑳		㉑		㉒		㉓		㉔	
⑬		⑭		⑮		⑯		⑰		⑱	
⑦		⑧		⑨		⑩		⑪		⑫	
①	E_x= E_y=	②		③		④		⑤		⑥	

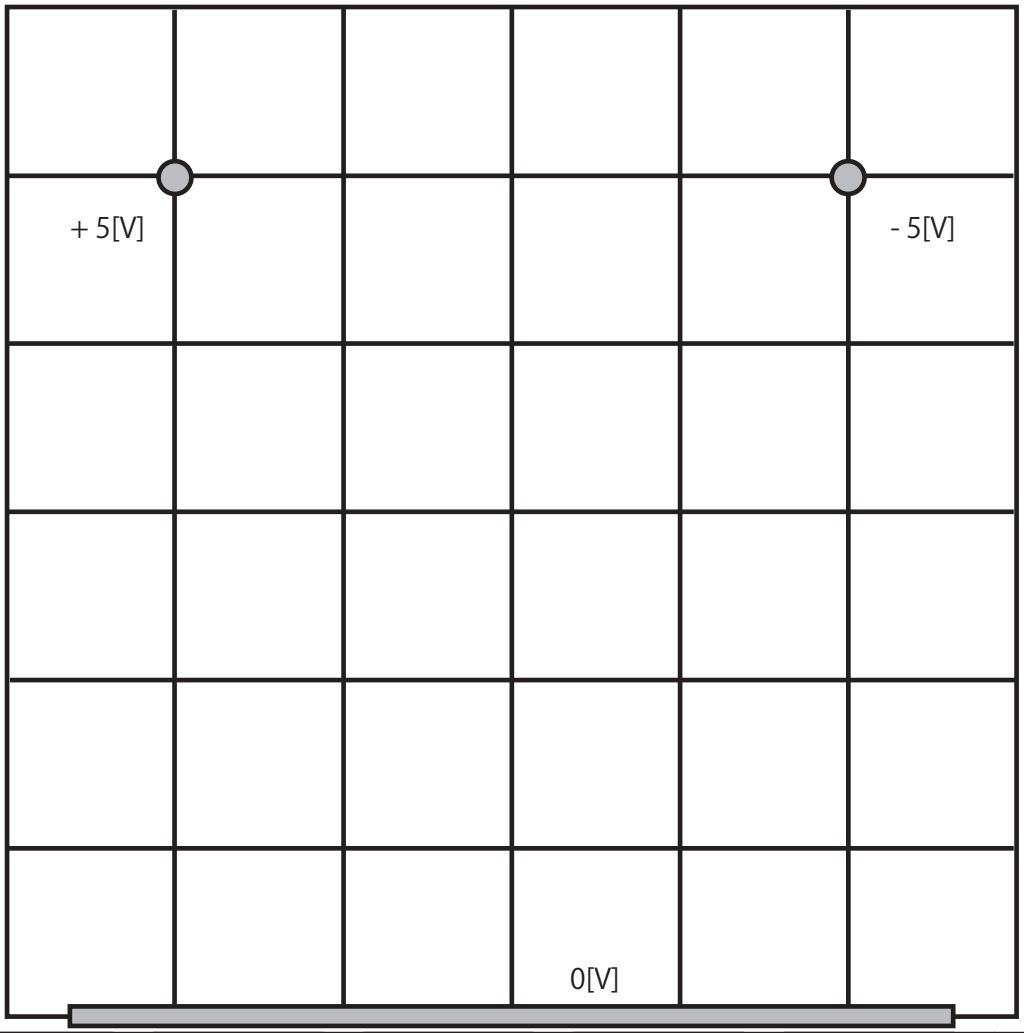
【実験２：電界の大きさと向き】



単位 [V/m]

㉛		㉜		㉝		㉞		㉟		㊱	
㉕		㉖		㉗		㉘		㉙		㉚	
⑲		⑳		㉑		㉒		㉓		㉔	
⑬		⑭		⑮		⑯		⑰		⑱	
⑦		⑧		⑨		⑩		⑪		⑫	
①	E_x= E_y=	②		③		④		⑤		⑥	

【実験３：電界の大きさと向き】



単位 [V/m]

㉛		㉜		㉝		㉞		㉟		㊱	
㉕		㉖		㉗		㉘		㉙		㉚	
⑲		⑳		㉑		㉒		㉓		㉔	
⑬		⑭		⑮		⑯		⑰		⑱	
⑦		⑧		⑨		⑩		⑪		⑫	
①	E_x= E_y=	②		③		④		⑤		⑥	

ソースコード 4.1　電位分布測定結果から等電位線の作図

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# 電位分布の実験より、等電位面（線）の作図
vx, vy = np.linspace(0, 120, 7), np.arange(0, 130, 20, dtype=float)
X, Y = np.meshgrid(vx, vy)
Z1 = np.array([[3.52,3.48,3.32,3.12,2.76,2.54,2.43],
  [3.58,3.51,3.30,2.90,2.52,2.31,2.22],[3.78,3.85,3.47,2.97,2.36,1.92,1.97],
  [3.91,5.00,3.56,2.95,2.21,0.00,1.80],[3.77,3.85,3.47,2.96,2.45,1.96,2.00],
  [3.56,3.50,3.28,2.96,2.62,2.32,2.23],[3.44,3.33,3.20,2.99,2.75,2.54,2.44]])
Z2 = np.array([[4.47,4.65,4.14,2.90,1.60,0.87,1.08],
  [4.45,4.65,4.11,2.93,1.64,0.88,1.10],[4.39,4.63,4.14,2.89,1.65,0.96,1.21],
  [4.24,4.40,4.04,2.90,1.85,1.24,1.42],[3.99,3.96,3.56,2.83,2.15,1.77,1.66],
  [3.78,3.62,3.30,2.83,2.37,2.06,1.92],[3.57,3.44,3.16,2.85,2.46,2.18,2.01]])
Z3 = np.array([[0.76,0.52,0.31,0.065,-0.19,-0.41,-0.58],
  [0.88,0.61,0.36,0.0,-0.24,-0.52,-0.68],
  [1.1,0.87,0.53,0.081,-0.039,-0.013,-0.96],
  [1.66,1.46,0.93,0.17,-0.54,-1.06,-1.23],
  [2.24,2.31,1.35,1.49,-1.02,-2.06,-1.83],
  [2.72,5.00,1.84,1.61,-1.42,-5.0,-2.14],
  [2.3,2.35,1.46,0.3,-0.78,-1.62,-1.65]])
Z=Z1
cont = plt.contour(X,Y,Z,levels=20,linestyles='dashed',linewidths=3)
cont.clabel(fmt='%1.1f', fontsize=14)
plt.title('Potential Energy(1)', size=14)
plt.xlabel('X', fontsize=14)
plt.ylabel('Y', fontsize=14)
plt.grid()
plt.savefig('contr1.png')
plt.show()

# X, Y, Zをワイヤフレーム(あるいはサーフェス)で表示
ax = Axes3D( plt.figure() ) # Axes3D のオブジェクトを ax に取得
#ax.plot_wireframe(X, Y, Z)
ax.plot_surface(X, Y, Z, cmap='bone', antialiased=True)
#ax.plot_trisurf(X, Y, Z, cmap = "bwr")
ax.set_title('Potential Energy(1)', size=14)
ax.set_xlabel("x[mm]", size = 14)
ax.set_ylabel("y[mm]", size = 14)
ax.set_zlabel("z[V]", size = 14)
#ax.contour(X, Y, Z, colors = "black", offset = -3)
#ax.view_init(45, 45)  # 視点の設定（仰角＝水平0〜垂直90）（回転角＝0〜360）
plt.grid()
plt.savefig('contr2.png')
plt.show()
```

ソースコード 4.2　電位分布測定結果から電気力線の作図

```
import numpy as np
import matplotlib.pyplot as plt
# 電位分布の実験より、電流の大きさと方向の作図
Z1 = np.array(
        [[3.52,3.48,3.32,3.12,2.76,2.54,2.43],
        [3.58,3.51,3.30,2.90,2.52,2.31,2.22],
        [3.78,3.85,3.47,2.97,2.36,1.92,1.97],
        [3.91,5.00,3.56,2.95,2.21,0.00,1.80],
        [3.77,3.85,3.47,2.96,2.45,1.96,2.00],
        [3.56,3.50,3.28,2.96,2.62,2.32,2.23],
        [3.44,3.33,3.20,2.99,2.75,2.54,2.44]])
Z2 = np.array(
        [[4.47,4.65,4.14,2.90,1.60,0.87,1.08],
        [4.45,4.65,4.11,2.93,1.64,0.88,1.10],
        [4.39,4.63,4.14,2.89,1.65,0.96,1.21],
        [4.24,4.40,4.04,2.90,1.85,1.24,1.42],
        [3.99,3.96,3.56,2.83,2.15,1.77,1.66],
        [3.78,3.62,3.30,2.83,2.37,2.06,1.92],
        [3.57,3.44,3.16,2.85,2.46,2.18,2.01]])
Z3 = np.array(
        [[0.76,0.52,0.31,0.065,-0.19,-0.41,-0.58],
        [0.88,0.61,0.36,0.00,-0.24,-0.52,-0.68],
        [1.10,0.87,0.53,0.081,-0.039,-0.013,-0.96],
        [1.66,1.46,0.928,0.17,-0.54,-1.06,-1.23],
        [2.24,2.31,1.35,1.49,-1.02,-2.06,-1.83],
        [2.72,5.00,1.84,1.61,-1.42,-5.0,-2.14],
        [2.3,2.35,1.46,0.3,-0.78,-1.62,-1.65]])
Z=Z1
X, Y = np.mgrid[10:125:20, 10:125:20]
U,V=[],[]
for j in range(len(Z)-1):
  for i in range(Z[j].size-1):
    x, y = -(Z[j][i+1] - Z[j][i]), -(Z[j+1][i+1] - Z[j][i+1])
    U.append(x)
    V.append(y)

plt.title('Electric Field (1)',size=14)
plt.xlabel('X', size=14)
plt.ylabel('Y', size=14)
plt.xlim(0,120)
plt.ylim(0,120)
plt.quiver(Y, X, U, V, units='width', color='blue')
plt.grid()
plt.savefig('Efield1.png')
plt.show()
```

ソースコード 4.3　電位分布の模擬実験

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

def solv(CX, CY, IE, JE, CT, boundary):
    def cvol():
        return CX*(CT[j][i-1]+CT[j][i+1])+CY*(CT[j-1][i]+CT[j+1][i])
    for j in range(1,JE+1):
        for i in range(1,IE+1):
            CT[j][i] = cvol()
    boundary(IE,JE,CT)
    for j in reversed(range(1,JE+1)):
        for i in reversed(range(1,IE+1)):
            CT[j][i] = cvol()
    boundary(IE,JE,CT)
    for i in range(1,IE+1):
        for j in reversed(range(1,JE+1)):
            CT[j][i] = cvol()
    boundary(IE,JE,CT)
    for i in reversed(range(1,IE+1)):
        for j in range(1,JE+1):
            CT[j][i] = cvol()
    boundary(IE,JE,CT)

def fixedb3(IE,JE,CT,CTH,CTC):
    for i in range(0,IE+2):
        CT[1][i] = 0.5
    CT[6][2] = 1.0
    CT[6][6] = 0.0

def fixedb2(IE,JE,CT,CTH,CTC):
    for j in range(5):
        CT[j][3]=1.0
        CT[j][5]=0.0

def fixedb1(IE,JE,CT,CTH,CTC):
    CT[4][2] = 1.0
    CT[4][6] = 0.0

def fixedb0(IE,JE,CT,CTH,CTC):
    for i in range(0,IE+2):
        CT[0][i] = CTH
        CT[JE+1][i] = CTC
    for j in range(0,JE+1):
        CT[j][0] = CTC
        CT[j][IE+1] = CTC

def bound3(IE,JE,CT):
    for i in range(0,IE+2):
        CT[1][i] = 0.5
        CT[JE+1][i] = CT[JE][i]
    for j in range(0,JE+1):
        CT[j][0] = CT[j][1]
        CT[j][IE+1] = CT[j][IE]
    fixedb3(IE,JE,CT,CTH,CTC)

def bound2(IE,JE,CT):
```

```
57      for i in range(0,IE+2):
58          CT[0][i] = CT[1][i]
59          CT[JE+1][i] = CT[JE][i]
60      for j in range(0,JE+1):
61          CT[j][0] = CT[j][1]
62          CT[j][IE+1] = CT[j][IE]
63      fixedb2(IE,JE,CT,CTH,CTC)
64
65  def bound1(IE,JE,CT):
66      for i in range(0,IE+2):
67          CT[0][i] = CT[1][i]
68          CT[JE+1][i] = CT[JE][i]
69      for j in range(0,JE+1):
70          CT[j][0] = CT[j][1]
71          CT[j][IE+1] = CT[j][IE]
72      fixedb1(IE,JE,CT,CTH,CTC)
73
74  def bound0(IE,JE,CT):
75      fixedb0(IE,JE,CT,CTH,CTC)
76
77  def init(IE,JE,CT,WK,fixedb):
78      for j in range(0,JE+2):
79          for i in range(0,IE+2):
80              CT[j][i] = WK
81      fixedb(IE,JE,CT,CTH,CTC)
82
83  def debug(IE,JE,CT):
84      for j in range(0,JE+2):
85          print( j,CT[j] )
86
87  CTH = 1.0
88  CTC = 0.0
89  AR = 1.0/1.0
90  IE = 7
91  JE = 7
92  LAST = 15
93  CT = np.array([[0.0]*(IE+2)]*(JE+2))
94
95  DX = 1.0/float(IE)
96  DY = 1.0/float(JE)
97  DX2 = DX * DX
98  DY2 = DY * DY * AR * AR
99  CX = 0.5 * DY2 / (DX2 + DY2)
100 CY = 0.5 * DX2 / (DX2 + DY2)
101 WK = 0.5 * (CTH + CTC)
102
103 model = 1 # 0 or 1 or 2 or 3
104 if model==1:
105     bound = bound1
106     fixedb = fixedb1
107 elif model==2:
108     bound = bound2
109     fixedb = fixedb2
110 elif model==3:
111     bound = bound3
112     fixedb = fixedb3
113 else:
114     bound = bound0
115     fixedb = fixedb0
```

```
116
117 init(IE,JE,CT,WK,fixedb)
118 # debug(IE,JE,CT)
119
120 iter = 0
121 while iter<LAST:
122     solv(CX, CY, IE, JE, CT, bound)
123     # debug(IE,JE,CT)
124     iter = iter + 1
125
126 X,Y=np.meshgrid(np.linspace(0.0,1.0,IE+2),np.linspace(0.0,1.0,JE+2))
127 #cont = plt.pcolormesh(X, Y, CT, cmap = "Blues")
128 cont=plt.contour(X,Y,CT,levels=20,linestyles='dashed',linewidths=3)
129 #cont.clabel(fmt='%1.1f', fontsize=12)
130 #plt.colorbar(cont)
131 plt.title('Simulation (Model '+str(model)+')')
132 plt.xlim(1.0/(IE+2),1.0-1.0/(IE+2))
133 plt.ylim(1.0/(JE+2),1.0-1.0/(JE+2))
134 plt.grid()
135 plt.savefig('simu'+str(model)+'.png')
136 plt.show()
```