Electrostatic Potentials

Lý thuyết

Thế năng điện từ $U(\mathbf{x})$ có thể được dẫn ra bằng mật độ điện tích $\rho(\mathbf{x})$ sao cho thỏa phương trình $Poisson$

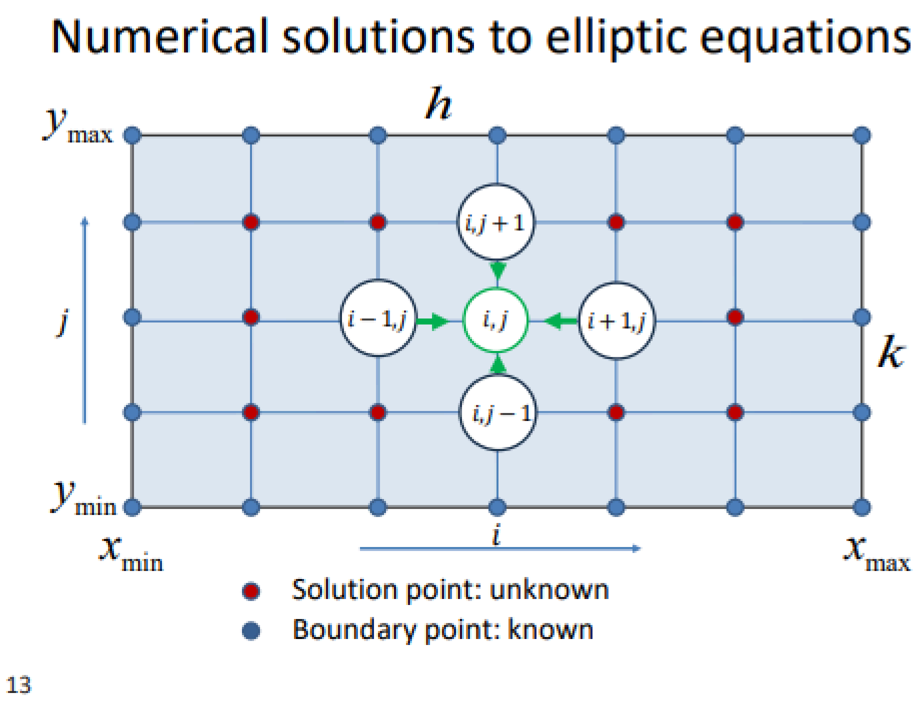
$$\nabla^2 U(\mathbf{x}) = -\frac{1}{\epsilon_0}\rho(\mathbf{x}), \tag{1}$$

trong đó $\rho(\mathbf{x})$ là mật độ điện tích. Trong vùng không có điện tích, $\rho(\mathbf{x}) = 0$, và thế năng ở phương trình (1) thỏa phương trình $Laplace$:

$$\nabla^2 U(\mathbf{x}) = 0. \tag{2}$$

Giả sử bài toán đang xét cho trường hợp 2-D, ta chia nhỏ không gian thành dạng của ô mạng tinh thể, và giải $U(\mathbf{x})$ cho từng điểm mạng.

Thuật toán



Lý thuyết

Ta khai triển Taylor cho phương trình (2)

$$u(x-\Delta x, y) = u(x,y) - \frac{\partial u}{\partial x} + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(\Delta x)^2 - \dots \tag{3}$$

$$u(x+\Delta x, y) = u(x,y) + \frac{\partial u}{\partial x} + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}(\Delta x)^2 + \dots \tag{4}$$

(3) + (4)

$$\Rightarrow \frac{\partial^2 u}{\partial x^2} \approx \frac{u(x-\Delta x, y) + u(x+\Delta x, y) + 2u(x,y)}{(\Delta x)^2} \tag{5}$$

Một cách tương tự ta có cho

$$\Rightarrow \frac{\partial^2 u}{\partial y^2} \approx \frac{u(x, y-\Delta y) + u(x, y+\Delta y) + 2u(x,y)}{(\Delta y)^2} \tag{6}$$

$$\frac{u(x-\Delta x, y) + u(x+\Delta x, y) + 2u(x,y)}{(\Delta x)^2} + \frac{u(x, y-\Delta y) + u(x, y+\Delta y) + 2u(x,y)}{(\Delta y)^2} = 0$$

Đặt $\Delta x \equiv h$, $\Delta y \equiv k$, cộng (5) và (6), ta có

$$2\left[\frac{h^2}{k^2} + 1\right]u(x,y) - [u(x+h,y) + u(x-h,y)] - \frac{h^2}{k^2}[u(x,y+k) + u(x,y-k)]$$
$$-\frac{h^2}{k^2}[u(x,y+k) + u(x,y-k)] = 0 \tag{7}$$

Finite Difference

$$x \to x_0 + ih \tag{8}$$
$$y \to y_0 + ik \tag{9}$$

và

$$u_{ij} \equiv u(x_i, y_j); \quad i = 1, \ldots n-1, ; \quad j = 1, \ldots, -1 \tag{10}$$

Ta có:

$$2\left[\frac{h^2}{k^2} + 1\right] u_{ij} - [u_{i+1,j} + u_{i-1,j}] - \frac{h^2}{k^2}[u_{i,j+1} + u_{i,j-1}] \tag{11}$$

đặt $h = k$, từ đó ta có viết lại (11)

$$u_{ij} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) \tag{12}$$

Phương pháp ma trận

Ta xem "mạng" trên như là ma trận, với các điểm màu đỏ là điểm cần giải, mỗi điểm cách nhau một $h = 1/4$. Tại y_{max} tất cả giá trị bằng điều kiện đầu, như vậy với n điểm màu đỏ, thì ứng với n điểm màu xanh là điều kiện đầu, và có tổng cộng $n + 2$ điểm màu xanh.

Giả sử, ta xét ma trận V là ma trận có chỉ số i, j. Để tính được toàn bộ số điểm màu đỏ cần giải, ta cần phải chuyển đổi chỉ số từ ma trận V sang một ma trận một chiều U có chứa các thành phần ma trận tương ứng. Với một ma trận vuông $V_{N \times N}$, ta có

$$V_{i,j} = U_{i \times N + j}$$

$(v_{i,j})$		(u_k)
(i)	(j)	(k = i × N + j)
0	0	0
0	1	1
0	2	2
1	0	3
1	1	4
1	2	5
2	0	6
2	1	7
2	2	8

Như vậy ta hoàn toàn có thể biểu diễn ma trận V dưới dạng ma trận U.

Dẫn ra ma trận

Từ phương trình

$$Au = B,$$

xét 25 điểm mạng tương ứng với đó là 9 điểm u (điểm màu đỏ). Ta có hệ phương trình sau

$$\begin{aligned} 100 - 4u_0 + u_1 + \quad + u_3 + \quad + \quad &= 0 \\ 100 + u_0 - 4u_1 + u_2 + \quad + u_4 + \quad &= 0 \\ 100 + u_1 - 4u_2 + \quad + \quad + u_5 &= 0 \end{aligned} \tag{13}$$

$$\begin{aligned} u_0 \quad - 4u_3 + u_4 + \quad + u_6 &= 0 \\ u_1 \quad + u_3 - 4u_4 + u_5 + \quad u_7 &= 0 \\ u_2 \quad + u_4 - 4u_5 \quad u_8 &= 0 \end{aligned} \tag{14}$$

$$\begin{aligned} u_3 \quad - 4u_6 + u_7 \quad &= 0 \\ u_4 \quad + u_6 - 4u_7 + u_8 &= 0 \\ u_5 \quad + u_7 - 4u_8 &= 0 \end{aligned} \tag{15}$$

Như vậy A sẽ có dạng đường chéo

```
[[-4.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 1. -4.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  1. -4.  1.  0.  0.  0.  0.  0.  0.  0.  0. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  1. -4.  1.  0.  0.  0.  0.  0.  0.  0. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  1. -4.  1.  0.  0.  0.  0.  0.  0. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  1. -4.  1.  0.  0.  0.  0.  0. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  1. -4.  1.  0.  0.  0.  0. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  1. -4.  1.  0.  0.  0. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  1. -4.  1.  0.  0. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  1. -4.  0.  0. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 1.  0.  0.  0.  0.  0.  0.  0.  0.  0. -4.  1. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1. -4. ...  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 ...
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ... -4.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ...  1. -4.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ...  0.  0. -4.  1.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ...  0.  0.  1. -4.  1.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ...  0.  0.  0.  1. -4.  1.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ...  0.  0.  0.  0.  1. -4.  1.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ...  0.  0.  0.  0.  0.  1. -4.  1.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ...  0.  0.  0.  0.  0.  0.  1. -4.  1.  0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ...  0.  0.  0.  0.  0.  0.  0.  1. -4.  1.  0.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ...  0.  0.  0.  0.  0.  0.  0.  0.  1. -4.  1.  0.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ...  1.  0.  0.  0.  0.  0.  0.  0.  0.  1. -4.  1.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. ...  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  1. -4.]]
```

Hình trên là với 11 điểm chưa biết.

Ta có thể giải ma trận trên bằng phương trình hàm riêng trị riêng, với trị riêng U. Ta phải chuyển đổi chỉ số từ $U \to V$.

> Để chéo hoá ma trận A, ta có thể sử dụng một số phương pháp để chéo hoá như là phương trình hàm riêng trị riêng , v.v. Một trong số đó là cách giải số theo phương pháp Jacobian và Gaussian-Seidel. Đồng thời phải đảm bảo tính ma trận chéo trội

$$|a_{ij}| > \sum_{i \neq j; j=1}^{N} |a_{ij}| \tag{16}$$

Jacobian Iterative Method

Phương pháp Jacobian cho ta kết quả là nghiệm duy nhất của cả hệ phương trình (13), (14), (15). Bằng cách cô lập biến u_1 cho phương trình thứ nhất và u_2 cho phương trình thứ 2 và cứ thế tiếp tục ta có được nghiệm là

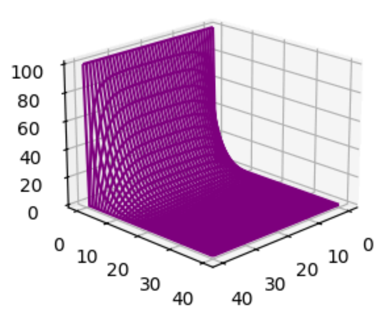
$$\begin{aligned} u_1 &= \frac{1}{a_{11}}(b_1 - a_{12}u_2 - a_{13}u_3 - \ldots a_{1n}u_n) \\ u_2 &= \frac{1}{a_{22}}(b_2 - a_{21}u_1 - a_{23}u_3 - \ldots a_{2n}u_n) \\ u_n &= \frac{1}{a_{nn}}(b_n - a_{n1}u_1 - a_{n2}u_2 - \ldots a_{n,n-1}u_{n-1}) \end{aligned}$$

ta viết lại thành

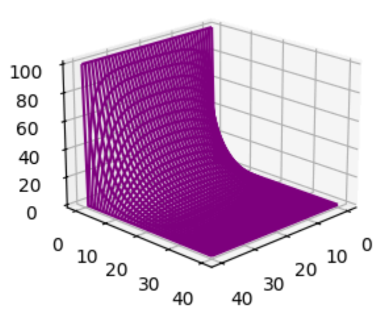
$$u_i = \frac{1}{a_{ii}}\left[\sum_{j=1, i \neq j}^{N} -a_{ij}u_j + b_i\right] \tag{17}$$

Kết quả

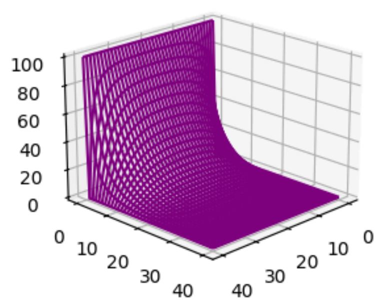
Jacobian with matrī at 1600 unknow points



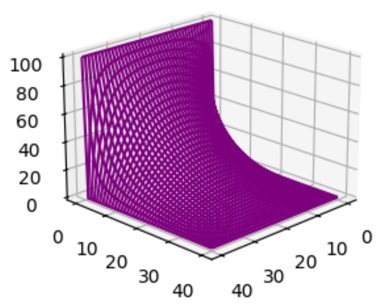
Gaussian-Seidel with matrix at 1600 unknow points



Gaussian-Seidel Iterative at 1600 unknow points



Nghiem giai tich su dung matrix at 1600 unknow points



ElectricPotentials.txt > data

n	i	j	Jacobian	Gaussian
0	0	0	49.36655532640504	49.66088632064772
1	0	1	68.50456274411086	69.09207657841027
2	0	2	77.18547434178444	78.06024885647292
3	0	3	81.67754445229511	82.82971134166625
4	0	4	84.24002059929701	85.65561646686552
5	0	5	85.80161532003424	87.46408125047721
6	0	6	86.79533589376811	88.68529435023913
7	0	7	87.44441212003738	89.54147711534429
8	0	8	87.87537098515062	90.15783507751522
9	0	9	88.16337945877805	90.60996023399895
10	0	10	88.35661047679919	90.94604656727705
11	0	11	88.48570632724756	91.19816437643146
12	0	12	88.57177990518194	91.38837867997769
13	0	13	88.62850944511348	91.53226049548874
14	0	14	88.66570425126773	91.64099730419332
15	0	15	88.68962572346368	91.72270993055987
16	0	16	88.70483164416105	91.78329790183751
17	0	17	88.71410323803381	91.82699168028799
18	0	18	88.71940486027	91.85671400964071
19	0	19	88.72178424073337	91.87431028226776
20	0	20	88.72178424073337	91.88068307786055
21	0	21	88.71940486027	91.87585057516586
22	0	22	88.71410323803381	91.85893787982852
23	0	23	88.70483164416105	91.82810169020354
24	0	24	88.68962572346368	91.78038009115316
25	0	25	88.66570425126773	91.7114485741843
26	0	26	88.62850944511348	91.61524787423376
27	0	27	88.57177990518194	91.48342436890206
28	0	28	88.48570632724756	91.3044813280639
29	0	29	88.35661047679919	91.06246299135023
30	0	30	88.16337945877805	90.73484958395284
31	0	31	87.87537098515062	90.28905599082756
32	0	32	87.44441212003738	89.67632748512261
33	0	33	86.79533589376811	88.82048315700901
34	0	34	85.80161532003424	87.59572099354065
35	0	35	84.240020599297	85.77923803529907
36	0	36	81.67754445229511	82.94030200707508
37	0	37	77.18547434178444	78.15231282154
38	0	38	68.50456274411086	69.15971625849225
39	0	39	49.36655532640504	49.697901358145195
40	1	0	28.98649952717637	29.565324612410716

Source code

Source code on Github