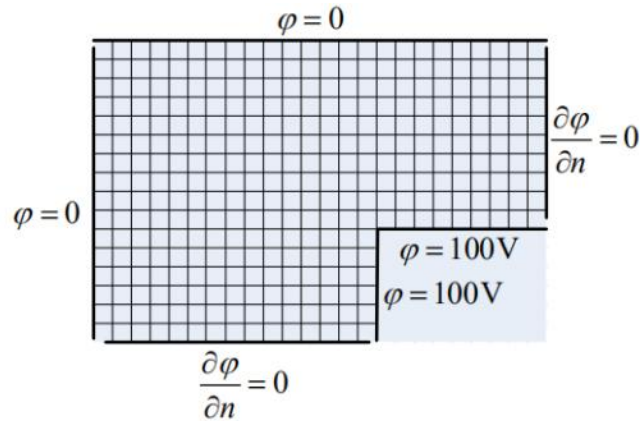


# 电磁场实验一实验报告

题目：试用高斯-塞德尔迭代法和超松弛迭代法确定二维静电场域的电位分布。



## 1. 高斯-塞德尔迭代法

程序：

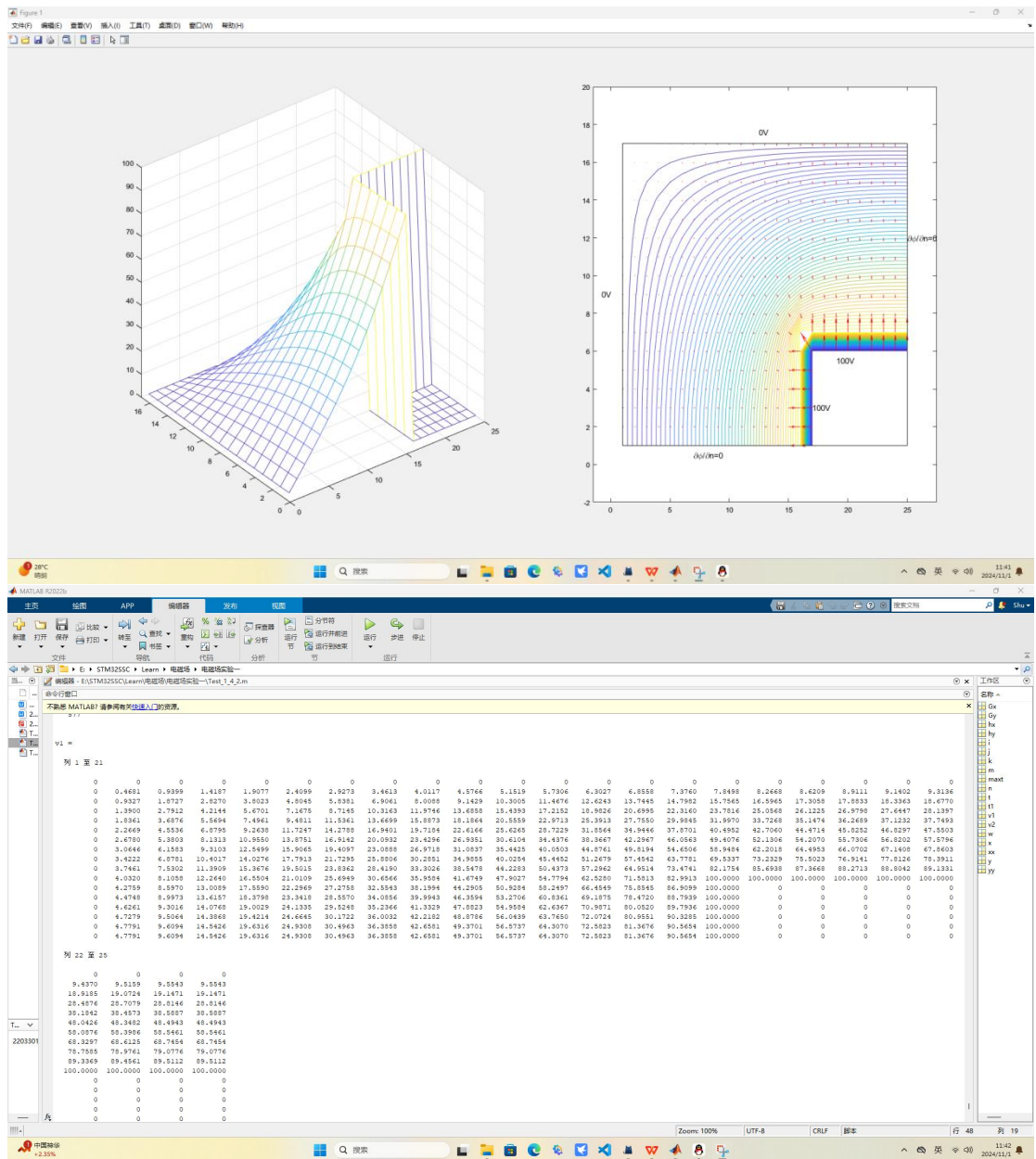
```
1. % 用高斯-赛德尔迭代法求二维静电场域的电位分布
2. hx=25;hy=17; % 设置网格节点数
3. v1=ones(hy,hx); % 设置行列二维数组
4. v1(12:17,17:25)=zeros(6,9); % 12 到 17 行的 17 到 25 列电位为 0
5. m=24;n=16; % 横纵向网格数
6.
7. % 边界的 Dirichlet 边界条件值
8. v1(1,1:25)=0; % 第一行电位为 0
9. v1(1:17,1)=0; % 第一列电位为 0
10. v1(11,16:25)=ones(1,10)*100; % 横向 100V 边界
11. v1(11:17,16)=ones(7,1)*100; % 纵向 100V 边界
12.
13. % 高斯-赛德尔迭代法
14. v2=v1;maxt=1;t=0; % 初始化
15. k=0;
16. while(maxt>1e-6) % 由 v1 迭代，算出 v2，迭代精度
    为 0.000001
17.     k=k+1 % 计算迭代次数
18.     maxt=0;
19.     for i=2:10 % 从 2 到 10 行循环
20.         for j=2:24 % 从 2 到 24 列循环
21.             v2(i,j)=(v1(i,j+1)+v1(i+1,j)+v2(i-1,j)+v2(i,j-1))/4;%拉普拉斯方程
                差分式
22.             t=abs(v2(i,j)-v1(i,j));
23.             if(t>maxt) maxt=t;
24.         end
25.     end
26. end
```

```

27.     for i=11:16                                % 从 11 到 16 行循环
28.         for j=2:15                              % 从 2 到 15 列循环
29.             v2(i,j)=(v1(i,j+1)+v1(i+1,j)+v2(i-1,j)+v2(i,j-1))/4;%拉普拉斯方程
            差分式
30.             t=abs(v2(i,j)-v1(i,j));
31.             if(t>maxt) maxt=t;
32.         end
33.     end
34. end
35.     v2(17,2:15)=v2(16,2:15);                    % Neumann 条件处理
36.     v2(2:10,25)=v2(2:10,24);                    % Neumann 条件处理
37.     v1=v2                                          % 迭代一次
38. end
39.
40. v1=v2(hy:-1:1,: )
41. subplot(1,2,1),mesh(v1)                          % 画三维曲面图
42. axis([0,25,0,17,0,100])
43. subplot(1,2,2),contour(v1,50)                    % 画等电位线图
44.
45. hold on
46. x=1:1:hx;y=1:1:hy;
47. [xx,yy]=meshgrid(x,y);                          % 形成栅格
48. [Gx,Gy]=gradient(v1,0.6,0.6);                  % 计算梯度
49. quiver(xx,yy,Gx,Gy,'r')                          % 根据梯度数据画箭
            头
50. axis([-1.5,hx+2.5,-2,20])                        % 设置坐标边框
51. plot([1,1,hx,hx,1],[1,hy,hy,1,1], 'k')          % 画导体边框
52. text(12.5,17.6,'0V','fontsize',11);
53. text(-0.8,9,'0V','fontsize',11);
54. text(17,3,'100V','fontsize',11);
55. text(19,5.5,'100V','fontsize',11);
56. text(25,12,'\partial\phi/\partial n=0','fontsize',11);
57. text(7,0.5,'\partial\phi/\partial n=0','fontsize',11);
58. hold off

```

实验结果:



## 2. 超松弛迭代法

程序：

1. % 用超松弛迭代法求二维静电场域的电位分布
2. `hx=25;hy=17;` % 设置网格节点数
3. `v1=ones(hy,hx);` % 设置行列二维数组
4. `v1(12:17,17:25)=zeros(6,9);` % 12 到 17 行的 17 到 25 列电位为 0
5. `m=24;n=16;` % 横纵向网格数
- 6.
7. % 边界的 Dirichlet 边界条件值
8. `v1(1,1:25)=0;` % 第一行电位为 0
9. `v1(1:17,1)=0;` % 第一列电位为 0
10. `v1(11,16:25)=ones(1,10)*100;` % 横向 100V 边界
11. `v1(11:17,16)=ones(7,1)*100;` % 纵向 100V 边界
- 12.

```

13. % 计算加速收敛因子
14. t1=(cos(pi/m)+cos(pi/n))/2;
15. w=2/(1+sqrt(1-t1*t1));
16.
17. % 超松弛迭代法
18. v2=v1;maxt=1;t=0; % 初始化
19. k=0;
20. while(maxt>1e-6) % 由 v1 迭代, 算出 v2, 迭代精度
    为 0.000001
21.     k=k+1 % 计算迭代次数
22.     maxt=0;
23.     for i=2:10 % 从 2 到 10 行循环
24.         for j=2:24 % 从 2 到 24 列循环
25.             v2(i,j)=v1(i,j)+(v1(i,j+1)+v1(i+1,j)+v2(i-1,j)+v2(i,j-1)-4*v1(i,
                j))*w/4;%拉普拉斯方程差分式
26.             t=abs(v2(i,j)-v1(i,j));
27.             if(t>maxt) maxt=t;
28.         end
29.     end
30. end
31. for i=11:16 % 从 11 到 16 行循环
32.     for j=2:15 % 从 2 到 15 列循环
33.         v2(i,j)=v1(i,j)+(v1(i,j+1)+v1(i+1,j)+v2(i-1,j)+v2(i,j-1)-4*v1(i,
            j))*w/4;%拉普拉斯方程差分式
34.         t=abs(v2(i,j)-v1(i,j));
35.         if(t>maxt) maxt=t;
36.     end
37. end
38. end
39. v2(17,2:15)=v2(16,2:15); % Neumann 条件处理
40. v2(2:10,25)=v2(2:10,24); % Neumann 条件处理
41. v1=v2 % 迭代一次
42. end
43.
44. v1=v2(hy:-1:1,:);
45. subplot(1,2,1),mesh(v1) % 画三维曲面图
46. axis([0,25,0,17,0,100])
47. subplot(1,2,2),contour(v1,50) % 画等电位线图
48.
49. hold on
50. x=1:1:hx;y=1:1:hy;
51. [xx,yy]=meshgrid(x,y); % 形成栅格
52. [Gx,Gy]=gradient(v1,0.6,0.6); % 计算梯度

```

53. `quiver(xx,yy,Gx,Gy, 'r')` % 根据梯度数据画箭头

54. `axis([-1.5,hx+2.5,-2,20])` % 设置坐标边框

55. `plot([1,1,hx,hx,1],[1,hy,hy,1,1], 'k')` % 画导体边框

56. `text(12.5,17.6, '0V', 'fontsize',11);`

57. `text(-0.8,9, '0V', 'fontsize',11);`

58. `text(17,3, '100V', 'fontsize',11);`

59. `text(19,5.5, '100V', 'fontsize',11);`

60. `text(25,12, '\partial\phi/\partial x=0', 'fontsize',11);`

61. `text(7,0.5, '\partial\phi/\partial y=0', 'fontsize',11);`

62. `hold off`

实验结果:

