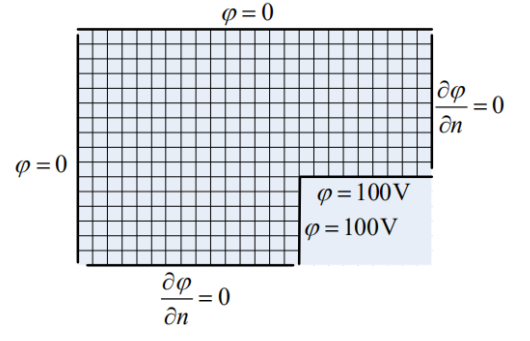
电磁场实验一实验报告

**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。**

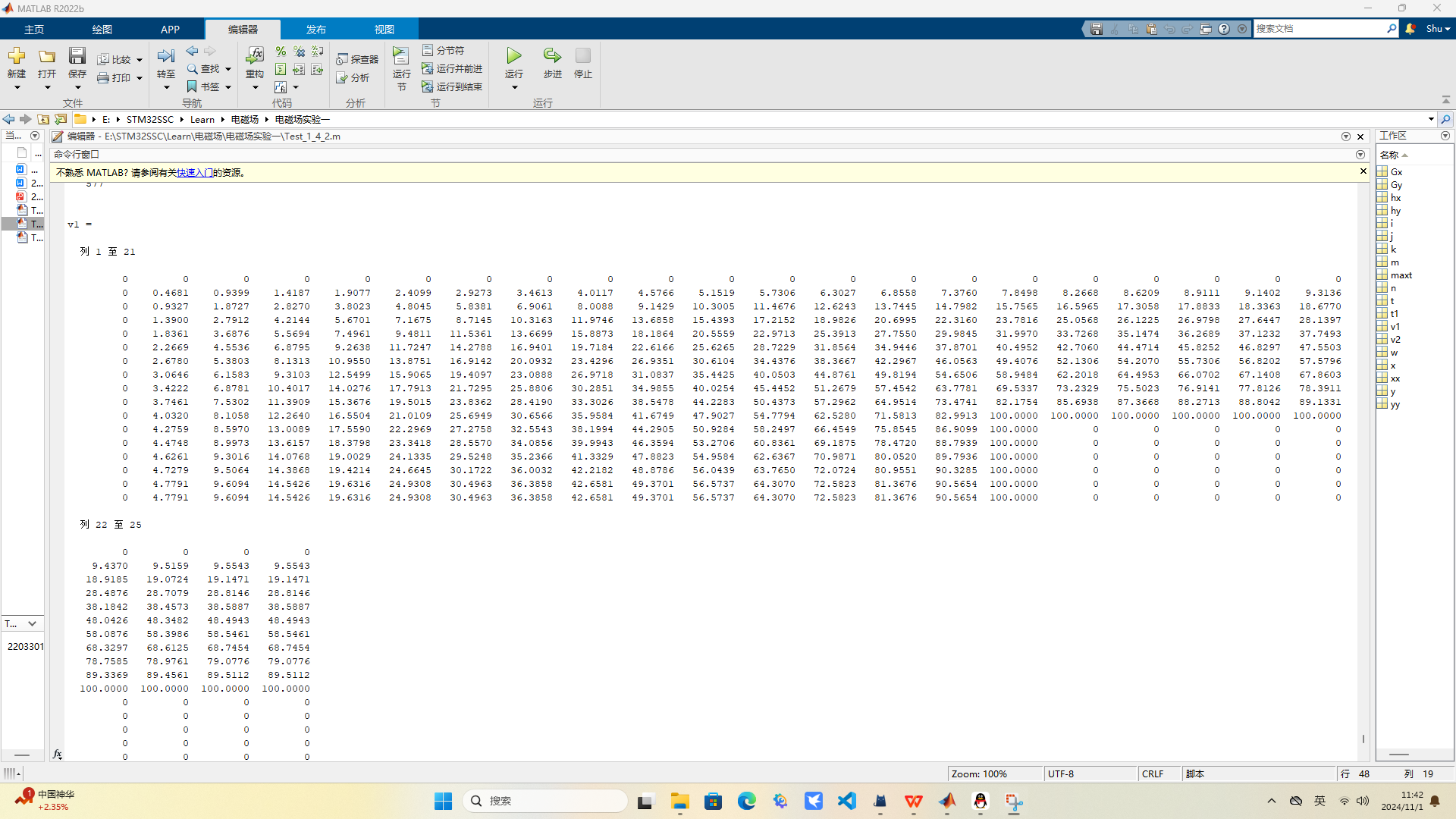
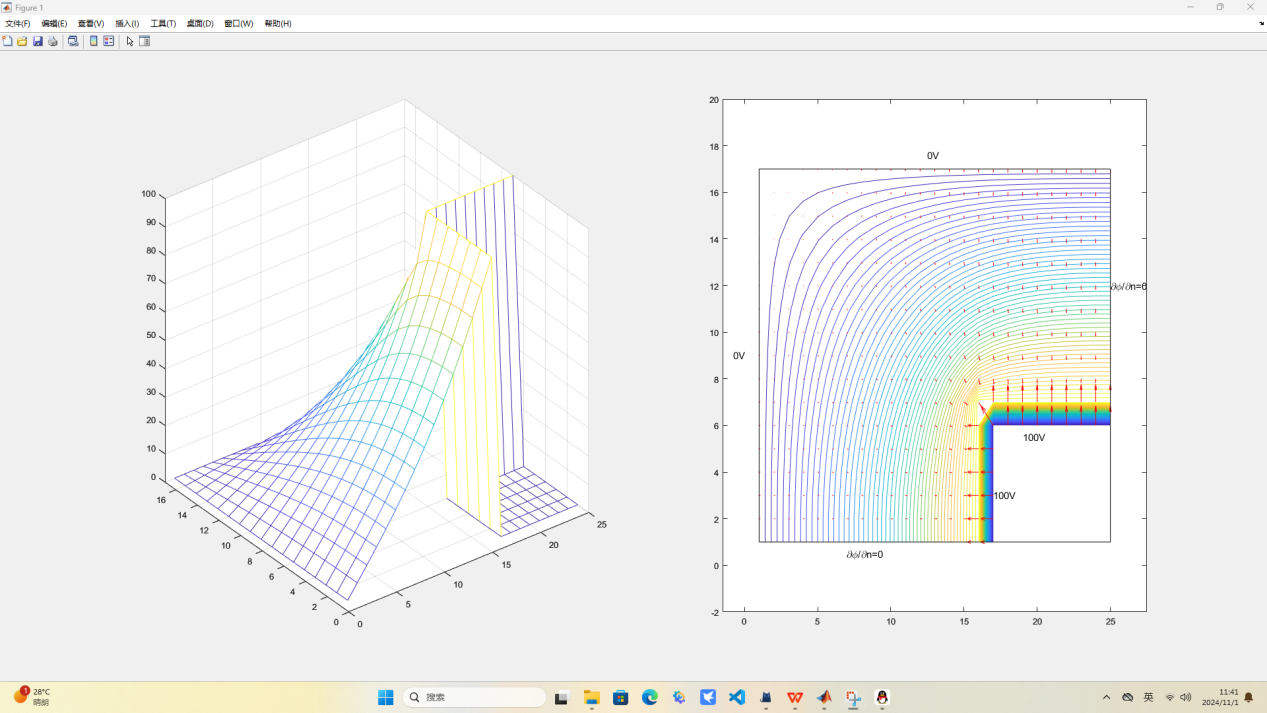


1. **高斯-塞德尔迭代法**

**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. % 边界的 Dirichlet 边界条件值
7. v1(1,1:25)=0;                   % 第一行电位为0
8. v1(1:17,1)=0;                   % 第一列电位为0
9. v1(11,16:25)=ones(1,10)\*100;    % 横向100V边界
10. v1(11:17,16)=ones(7,1)\*100;     % 纵向100V边界
11. % 高斯-赛德尔迭代法
12. v2=v1;maxt=1;t=0;                       % 初始化
13. k=0;
14. while(maxt>1e-6)                        % 由 v1 迭代，算出 v2，迭代精度为 0.000001
15. k=k+1                               % 计算迭代次数
16. maxt=0;
17. for i=2:10                          % 从 2 到 10 行循环
18. for j=2:24                      % 从 2 到 24 列循环
19. 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
20. t=abs(v2(i,j)-v1(i,j));
21. if(t>maxt) maxt=t;
22. end
23. end
24. end
25. for i=11:16                          % 从 11 到 16 行循环
26. for j=2:15                       % 从 2 到 15 列循环
27. 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
28. t=abs(v2(i,j)-v1(i,j));
29. if(t>maxt) maxt=t;
30. end
31. end
32. end
33. v2(17,2:15)=v2(16,2:15);             % Neumann 条件处理
34. v2(2:10,25)=v2(2:10,24);             % Neumann 条件处理
35. v1=v2                                % 迭代一次
36. end
37. v1=v2(hy:-1:1,:)
38. subplot(1,2,1),mesh(v1)                                     % 画三维曲面图
39. axis([0,25,0,17,0,100])
40. subplot(1,2,2),contour(v1,50)                               % 画等电位线图
41. hold on
42. x=1:1:hx;y=1:1:hy;
43. [xx,yy]=meshgrid(x,y);                                      % 形成栅格
44. [Gx,Gy]=gradient(v1,0.6,0.6);                               % 计算梯度
45. quiver(xx,yy,Gx,Gy,'r')                                     % 根据梯度数据画箭头
46. axis([-1.5,hx+2.5,-2,20])                                   % 设置坐标边框
47. plot([1,1,hx,hx,1],[1,hy,hy,1,1],'k')                       % 画导体边框
48. text(12.5,17.6,'0V','fontsize',11);
49. text(-0.8,9,'0V','fontsize',11);
50. text(17,3,'100V','fontsize',11);
51. text(19,5.5,'100V','fontsize',11);
52. text(25,12,'\partial\phi/\partialn=0','fontsize',11);
53. text(7,0.5,'\partial\phi/\partialn=0','fontsize',11);
54. hold off

**实验结果：**



1. **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. % 边界的 Dirichlet 边界条件值
7. v1(1,1:25)=0;                   % 第一行电位为0
8. v1(1:17,1)=0;                   % 第一列电位为0
9. v1(11,16:25)=ones(1,10)\*100;    % 横向100V边界
10. v1(11:17,16)=ones(7,1)\*100;     % 纵向100V边界
11. % 计算加速收敛因子
12. t1=(cos(pi/m)+cos(pi/n))/2;
13. w=2/(1+sqrt(1-t1\*t1));
14. % 超松弛迭代法
15. v2=v1;maxt=1;t=0;                       % 初始化
16. k=0;
17. while(maxt>1e-6)                        % 由 v1 迭代，算出 v2，迭代精度为 0.000001
18. k=k+1                               % 计算迭代次数
19. maxt=0;
20. for i=2:10                          % 从 2 到 10 行循环
21. for j=2:24                      % 从 2 到 24 列循环
22. 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
23. t=abs(v2(i,j)-v1(i,j));
24. if(t>maxt) maxt=t;
25. end
26. end
27. end
28. for i=11:16                          % 从 11 到 16 行循环
29. for j=2:15                       % 从 2 到 15 列循环
30. 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
31. t=abs(v2(i,j)-v1(i,j));
32. if(t>maxt) maxt=t;
33. end
34. end
35. end
36. v2(17,2:15)=v2(16,2:15);             % Neumann 条件处理
37. v2(2:10,25)=v2(2:10,24);             % Neumann 条件处理
38. v1=v2                                % 迭代一次
39. end
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. hold on
45. x=1:1:hx;y=1:1:hy;
46. [xx,yy]=meshgrid(x,y);                                      % 形成栅格
47. [Gx,Gy]=gradient(v1,0.6,0.6);                               % 计算梯度
48. quiver(xx,yy,Gx,Gy,'r')                                     % 根据梯度数据画箭头
49. axis([-1.5,hx+2.5,-2,20])                                   % 设置坐标边框
50. plot([1,1,hx,hx,1],[1,hy,hy,1,1],'k')                       % 画导体边框
51. text(12.5,17.6,'0V','fontsize',11);
52. text(-0.8,9,'0V','fontsize',11);
53. text(17,3,'100V','fontsize',11);
54. text(19,5.5,'100V','fontsize',11);
55. text(25,12,'\partial\phi/\partialn=0','fontsize',11);
56. text(7,0.5,'\partial\phi/\partialn=0','fontsize',11);
57. hold off

**实验结果：**

