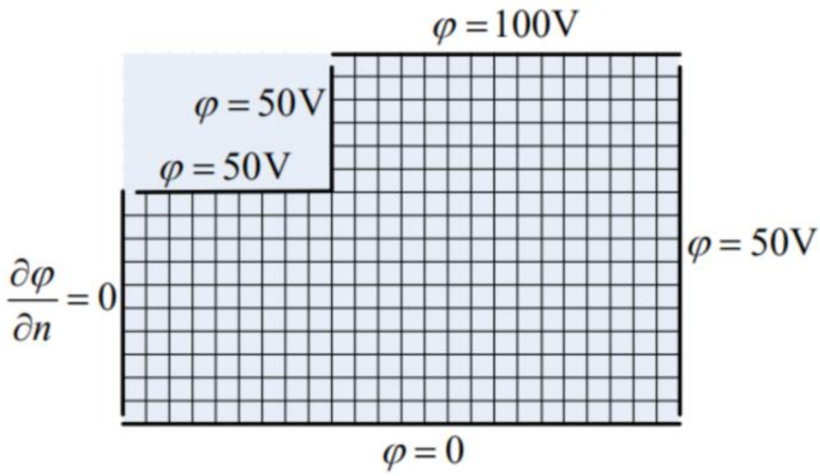
电磁场实验报告

实验一：静电场有限差分法简单计算的编程实现

2025 年 4 月 18 日

尾数 9：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。



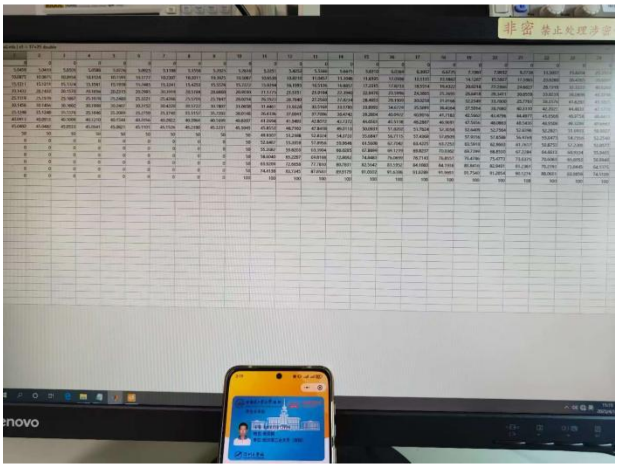
1. 高斯-塞德尔迭代法

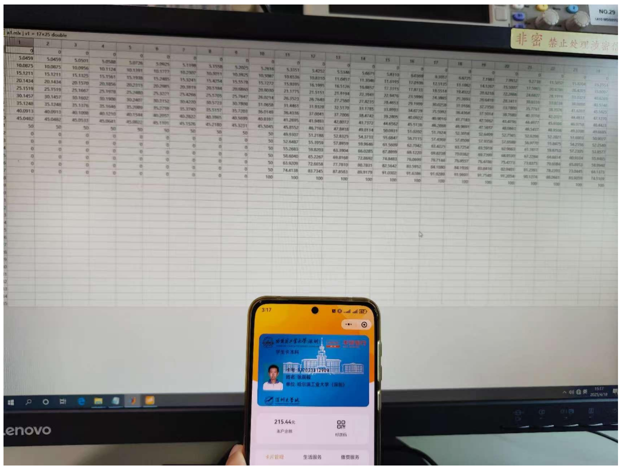
MATLAB 程序代码：

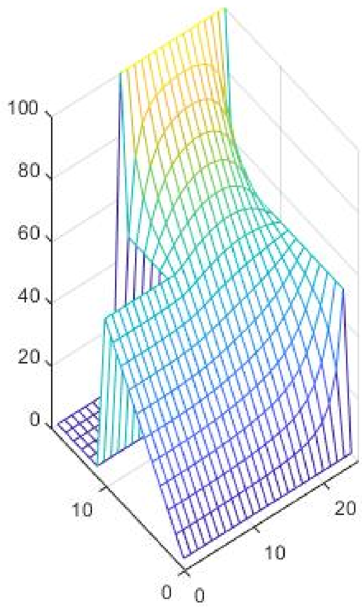
```
% 用高斯-赛德尔迭代法求二维静电场域的电位分布
hx=25;hy=17;                    % 设置网格节点数
v1=ones(hy,hx);                 % 设置行列二维数组
v1(1:6,1:9)=zeros(6,9);         % 1 到 6 行的 1 到 9 列电位为 0
m=24;n=16;                      % 横纵向网格数

% 边界的 Dirichlet 边界条件值
v1(1,10:25)=ones(1,16)*100;     % 第 1 行电位为 100V
v1(2:16,25)=ones(15,1)*50;      % 第 25 列电位为 50V
v1(17,1:25)=0;                  % 第 17 行电位为 0
v1(2:7,10)=ones(6,1)*50;        % 纵向 50V 边界
v1(7,2:10)=ones(1,9)*50;        % 横向 50V 边界

% 计算加速收敛因子
t1=(cos(pi/m)+cos(pi/n))/2;
w=2/(1+sqrt(1-t1*t1));

% 超松弛迭代法
v2=v1;maxt=1;t=0;               % 初始化
k=0;
```

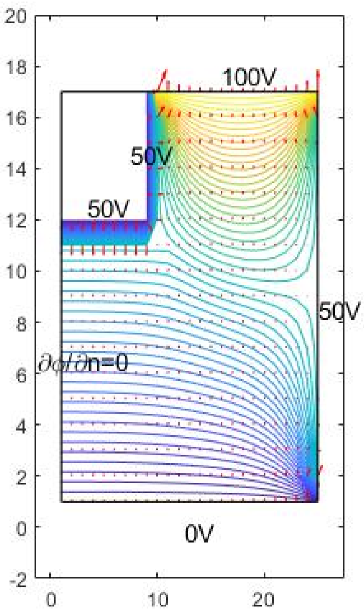
```
while(maxt>1e-6)                          % 由 v1 迭代，算出 v2，迭代精度为 0.000001
    k=k+1                                 % 计算迭代次数
    maxt=0;
    for i=2:7                             % 从 2 到 6 行循环
        for j=11:24                       % 从 11 到 24 列循环
            v2(i,j)=(v1(i,j+1)+v1(i+1,j)+v2(i-1,j)+v2(i,j-1))/4;%拉普拉斯方程差分式
            t=abs(v2(i,j)-v1(i,j));
            if(t>maxt) maxt=t;
                end
        end
    end
    for i=8:16                            % 从 7 到 16 行循环
        for j=2:24                        % 从 2 到 24 列循环
            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
            t=abs(v2(i,j)-v1(i,j));
            if(t>maxt) maxt=t;
                end
        end
    end
    v2(7:16,1)=v2(7:16,2);                % Neumann 条件处理
    v1=v2                                 % 迭代一次
end

v1=v2(hy:-1:1,:)
subplot(1,2,1),mesh(v1)                   % 画三维曲面图
axis([0,25,0,17,0,100])
subplot(1,2,2),contour(v1,50)             % 画等电位线图

hold on
x=1:1:hx;y=1:1:hy;
[xx,yy]=meshgrid(x,y);                    % 形成栅格
[Gx,Gy]=gradient(v1,0.6,0.6);             % 计算梯度
quiver(xx,yy,Gx,Gy,'r')                   % 根据梯度数据画箭头
axis([-1.5,hx+2.5,-2,20])                 % 设置坐标边框
plot([1,1,hx,hx,1],[1,hy,hy,1,1],'k')     % 画导体边框
text(12.5,-0.2,'0V','fontsize',11);
text(-1.2,6.5,'\partial\phi/\partialn=0','fontsize',11);
text(25,8.5,'50V','fontsize',11);
text(3.5,12.5,'50V','fontsize',11);
text(7.5,14.5,'50V','fontsize',11);
text(16,17.6,'100V','fontsize',11);
hold off
```

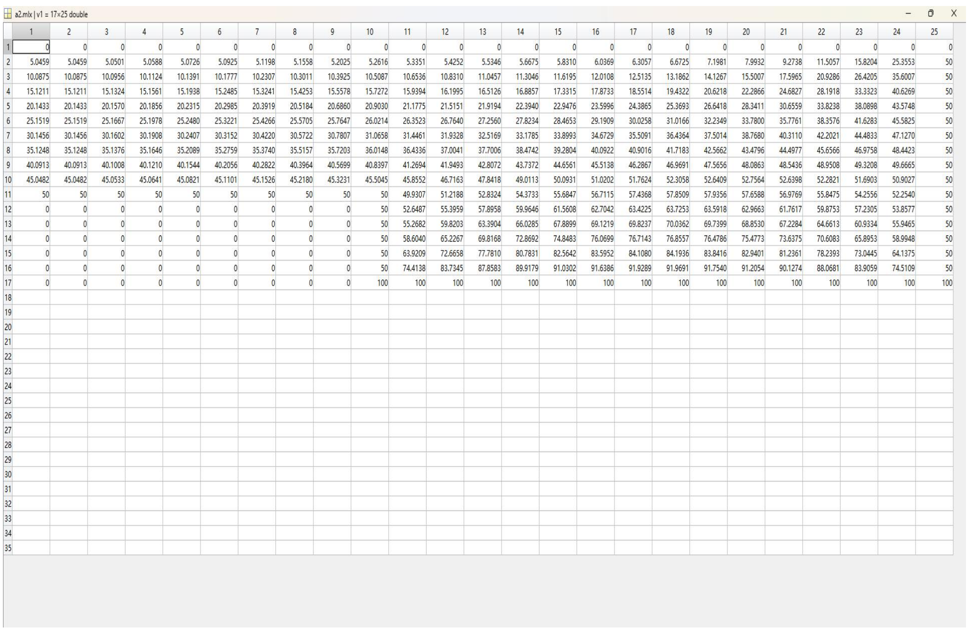
实验结果：











a2.mlx | v1 = 17×25 double

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	5.0459	5.0459	5.0501	5.0588	5.0726	5.0925	5.1198	5.1558	5.2025	5.2616	5.3351	5.4252	5.5346
3	10.0875	10.0875	10.0956	10.1124	10.1391	10.1777	10.2307	10.3011	10.3925	10.5087	10.6536	10.8310	11.0457
4	15.1211	15.1211	15.1324	15.1561	15.1938	15.2485	15.3241	15.4253	15.5578	15.7272	15.9394	16.1995	16.5126
5	20.1433	20.1433	20.1570	20.1856	20.2315	20.2985	20.3919	20.5184	20.6860	20.9030	21.1775	21.5151	21.9194
6	25.1519	25.1519	25.1667	25.1978	25.2480	25.3221	25.4266	25.5705	25.7647	26.0214	26.3523	26.7640	27.2560
7	30.1456	30.1456	30.1602	30.1908	30.2407	30.3152	30.4220	30.5722	30.7807	31.0658	31.4461	31.9328	32.5169
8	35.1248	35.1248	35.1376	35.1646	35.2089	35.2759	35.3740	35.5157	35.7203	36.0148	36.4336	37.0041	37.7006
9	40.0913	40.0913	40.1008	40.1210	40.1544	40.2056	40.2822	40.3964	40.5699	40.8397	41.2694	41.9493	42.8072
10	45.0482	45.0482	45.0533	45.0641	45.0821	45.1101	45.1526	45.2180	45.3231	45.5045	45.8552	46.7163	47.8418
11	50	50	50	50	50	50	50	50	50	50	49.9307	51.2188	52.8324
12	0	0	0	0	0	0	0	0	0	50	52.6487	55.3959	57.8958
13	0	0	0	0	0	0	0	0	0	50	55.2682	59.8203	63.3904
14	0	0	0	0	0	0	0	0	0	50	58.6040	65.2267	69.8168
15	0	0	0	0	0	0	0	0	0	50	63.9209	72.6658	77.7810
16	0	0	0	0	0	0	0	0	0	50	74.4138	83.7345	87.8583
17	0	0	0	0	0	0	0	0	0	100	100	100	100

	14	15	16	17	18	19	20	21	22	23	24	25
1	0	0	0	0	0	0	0	0	0	0	0	0
2	5.6675	5.8310	6.0369	6.3057	6.6725	7.1981	7.9932	9.2738	11.5057	15.8204	25.3553	50
3	11.3046	11.6195	12.0108	12.5135	13.1862	14.1267	15.5007	17.5965	20.9286	26.4205	35.6007	50
4	16.8857	17.3315	17.8733	18.5514	19.4322	20.6218	22.2866	24.6827	28.1918	33.3323	40.6269	50
5	22.3940	22.9476	23.5996	24.3865	25.3693	26.6418	28.3411	30.6559	33.8238	38.0898	43.5748	50
6	27.8234	28.4653	29.1909	30.0258	31.0166	32.2349	33.7800	35.7761	38.3576	41.6283	45.5825	50
7	33.1785	33.8993	34.6729	35.5091	36.4364	37.5014	38.7680	40.3110	42.2021	44.4833	47.1270	50
8	38.4742	39.2804	40.0922	40.9016	41.7183	42.5662	43.4796	44.4977	45.6566	46.9758	48.4423	50
9	43.7372	44.6561	45.5138	46.2867	46.9691	47.5656	48.0863	48.5436	48.9508	49.3208	49.6665	50
10	49.0113	50.0931	51.0202	51.7624	52.3058	52.6409	52.7564	52.6398	52.2821	51.6903	50.9027	50
11	54.3733	55.6847	56.7115	57.4368	57.8509	57.9356	57.6588	56.9769	55.8475	54.2556	52.2540	50
12	59.9646	61.5608	62.7042	63.4225	63.7253	63.5918	62.9663	61.7617	59.8753	57.2305	53.8577	50
13	66.0285	67.8899	69.1219	69.8237	70.0362	69.7399	68.8530	67.2284	64.6613	60.9334	55.9465	50
14	72.8692	74.8483	76.0699	76.7143	76.8557	76.4786	75.4773	73.6375	70.6083	65.8953	58.9948	50
15	80.7831	82.5642	83.5952	84.1080	84.1936	83.8416	82.9401	81.2361	78.2393	73.0445	64.1375	50
16	89.9179	91.0302	91.6386	91.9289	91.9691	91.7540	91.2054	90.1274	88.0681	83.9059	74.5109	50
17	100	100	100	100	100	100	100	100	100	100	100	100

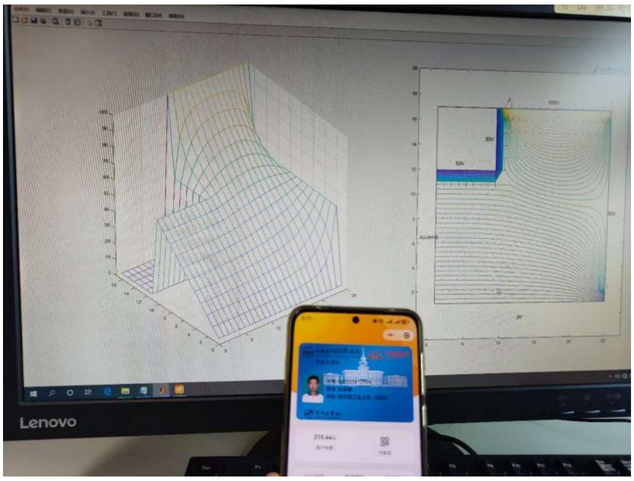
2.超松弛迭代法

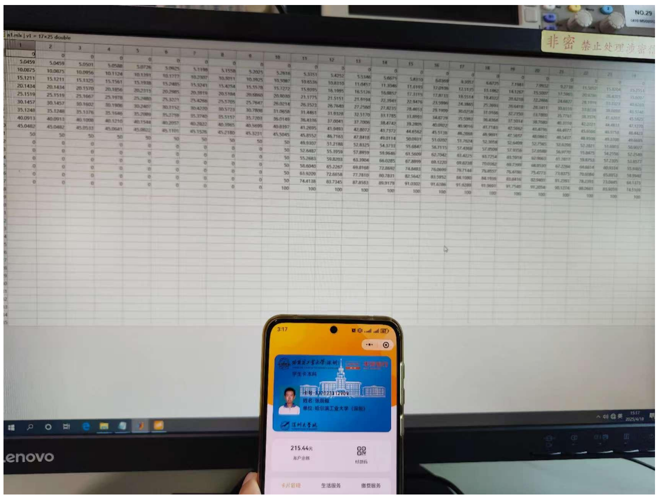
MATLAB 程序代码：

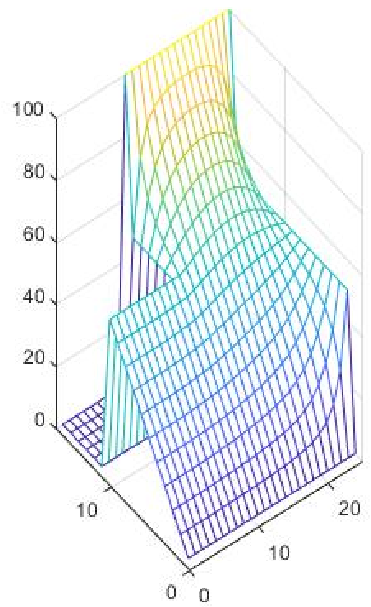
```
% 用超松弛迭代法求二维静电场域的电位分布
hx=25;hy=17;                              % 设置网格节点数
v1=ones(hy,hx);                           % 设置行列二维数组
v1(1:6,1:9)=zeros(6,9);          % 1 到 6 行的 1 到 9 列电位为 0
m=24;n=16;                                % 横纵向网格数

% 边界的 Dirichlet 边界条件值
v1(1,10:25)=ones(1,16)*100;          % 第 1 行电位为 100V
v1(2:16,25)=ones(15,1)*50;           % 第 25 列电位为 50V
v1(17,1:25)=0;               % 第 17 行电位为 0
v1(2:7,10)=ones(6,1)*50;             % 纵向 50V 边界
v1(7,2:10)=ones(1,9)*50;             % 横向 50V 边界

% 计算加速收敛因子
t1=(cos(pi/m)+cos(pi/n))/2;
w=2/(1+sqrt(1-t1*t1));

% 超松弛迭代法
v2=v1;maxt=1;t=0;                               % 初始化
k=0;
while(maxt>1e-6)                                % 由 v1 迭代，算出 v2，迭代精度为 0.000001
    k=k+1                                         % 计算迭代次数
    maxt=0;
    for i=2:7                                 % 从 2 到 6 行循环
        for j=11:24                             % 从 11 到 24 列循环
            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
            t=abs(v2(i,j)-v1(i,j));
            if(t>maxt) maxt=t;
                end
        end
    end
    for i=8:16                                  % 从 7 到 16 行循环
        for j=2:24                              % 从 2 到 24 列循环
            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
            t=abs(v2(i,j)-v1(i,j));
            if(t>maxt) maxt=t;
                end
        end
    end
```

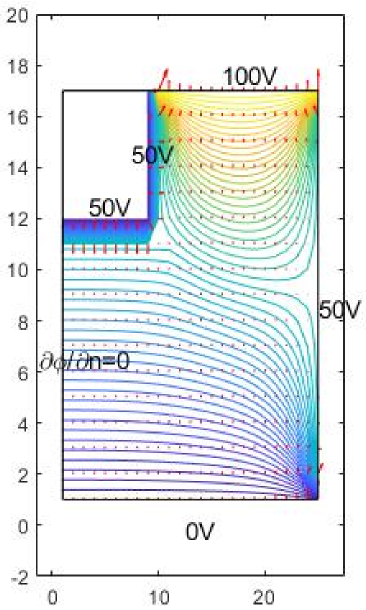
```
    v2(7:16,1)=v2(7:16,2);                    % Neumann 条件处理
    v1=v2                                      % 迭代一次
end

v1=v2(hy:-1:1,:)
subplot(1,2,1),mesh(v1)                                    % 画三维曲面图
axis([0,25,0,17,0,100])
subplot(1,2,2),contour(v1,50)                              % 画等电位线图

hold on
x=1:1:hx;y=1:1:hy;
[xx,yy]=meshgrid(x,y);                                     % 形成栅格
[Gx,Gy]=gradient(v1,0.6,0.6);                              % 计算梯度
quiver(xx,yy,Gx,Gy,'r')                                    % 根据梯度数据画箭头
axis([-1.5,hx+2.5,-2,20])                                  % 设置坐标边框
plot([1,1,hx,hx,1],[1,hy,hy,1,1],'k')                      % 画导体边框
text(12.5,-0.2,'0V','fontsize',11);
text(-1.2,6.5,'\partial\phi/\partialn=0','fontsize',11);
text(25,8.5,'50V','fontsize',11);
text(3.5,12.5,'50V','fontsize',11);
text(7.5,14.5,'50V','fontsize',11);
text(16,17.6,'100V','fontsize',11);
hold off
```

实验结果：









a1.mlx | v1 = 17×25 double

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	5.0459	5.0459	5.0501	5.0588	5.0726	5.0925	5.1198	5.1558	5.2025	5.2616	5.3351	5.4252	5.5346
3	10.0875	10.0875	10.0956	10.1124	10.1391	10.1777	10.2307	10.3011	10.3925	10.5087	10.6536	10.8310	11.0457
4	15.1211	15.1211	15.1325	15.1561	15.1938	15.2485	15.3241	15.4254	15.5578	15.7272	15.9395	16.1995	16.5126
5	20.1434	20.1434	20.1570	20.1856	20.2315	20.2985	20.3919	20.5184	20.6860	20.9030	21.1775	21.5151	21.9194
6	25.1519	25.1519	25.1667	25.1978	25.2480	25.3221	25.4266	25.5705	25.7647	26.0214	26.3523	26.7640	27.2560
7	30.1457	30.1457	30.1602	30.1908	30.2407	30.3152	30.4220	30.5723	30.7808	31.0658	31.4461	31.9328	32.5170
8	35.1248	35.1248	35.1376	35.1646	35.2089	35.2759	35.3740	35.5157	35.7203	36.0149	36.4336	37.0041	37.7006
9	40.0913	40.0913	40.1008	40.1210	40.1544	40.2057	40.2822	40.3965	40.5699	40.8397	41.2695	41.9493	42.8072
10	45.0482	45.0482	45.0533	45.0641	45.0822	45.1101	45.1526	45.2180	45.3231	45.5045	45.8552	46.7163	47.8418
11	50	50	50	50	50	50	50	50	50	50	49.9307	51.2188	52.8325
12	0	0	0	0	0	0	0	0	0	50	52.6487	55.3959	57.8959
13	0	0	0	0	0	0	0	0	0	50	55.2683	59.8203	63.3904
14	0	0	0	0	0	0	0	0	0	50	58.6040	65.2267	69.8168
15	0	0	0	0	0	0	0	0	0	50	63.9209	72.6658	77.7810
16	0	0	0	0	0	0	0	0	0	50	74.4138	83.7345	87.8583
17	0	0	0	0	0	0	0	0	0	100	100	100	100
18													
19													
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													
31													
32													
33													
34													
35													

	14	15	16	17	18	19	20	21	22	23	24	25
1	0	0	0	0	0	0	0	0	0	0	0	0
2	5.6675	5.8310	6.0369	6.3057	6.6725	7.1981	7.9932	9.2738	11.5057	15.8204	25.3553	50
3	11.3046	11.6195	12.0108	12.5135	13.1862	14.1267	15.5007	17.5965	20.9286	26.4205	35.6007	50
4	16.8857	17.3315	17.8733	18.5514	19.4322	20.6218	22.2866	24.6827	28.1919	33.3323	40.6269	50
5	22.3941	22.9476	23.5996	24.3865	25.3693	26.6418	28.3411	30.6559	33.8238	38.0898	43.5748	50
6	27.8235	28.4653	29.1909	30.0258	31.0166	32.2350	33.7800	35.7761	38.3576	41.6283	45.5825	50
7	33.1785	33.8993	34.6729	35.5092	36.4364	37.5014	38.7680	40.3110	42.2021	44.4833	47.1270	50
8	38.4742	39.2805	40.0922	40.9016	41.7183	42.5662	43.4796	44.4977	45.6566	46.9758	48.4423	50
9	43.7372	44.6562	45.5138	46.2868	46.9691	47.5657	48.0863	48.5437	48.9508	49.3208	49.6665	50
10	49.0114	50.0931	51.0202	51.7624	52.3058	52.6409	52.7565	52.6398	52.2821	51.6903	50.9027	50
11	54.3733	55.6847	56.7115	57.4368	57.8509	57.9356	57.6588	56.9770	55.8475	54.2556	52.2540	50
12	59.9646	61.5609	62.7042	63.4225	63.7254	63.5918	62.9663	61.7617	59.8753	57.2305	53.8577	50
13	66.0285	67.8899	69.1220	69.8238	70.0362	69.7399	68.8530	67.2284	64.6614	60.9334	55.9465	50
14	72.8692	74.8483	76.0699	76.7144	76.8557	76.4786	75.4773	73.6375	70.6084	65.8953	58.9948	50
15	80.7831	82.5642	83.5952	84.1080	84.1936	83.8416	82.9401	81.2361	78.2393	73.0445	64.1375	50
16	89.9179	91.0302	91.6386	91.9289	91.9691	91.7540	91.2054	90.1274	88.0681	83.9059	74.5109	50
17	100	100	100	100	100	100	100	100	100	100	100	100
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												
32												
33												
34												
35												