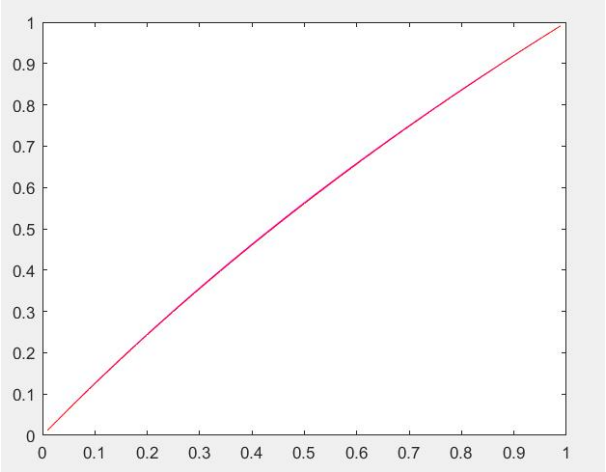
上机作业 1

1. 迭代次数比较

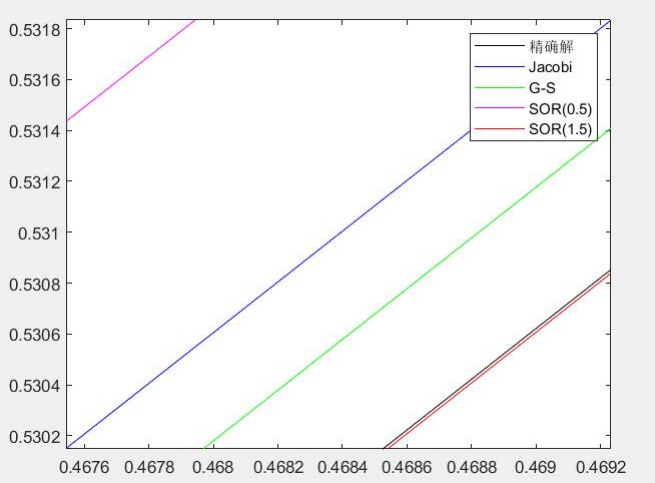
迭代次数	Jacobi 迭代	G-S 迭代	SOR(w=0.5)	SOR(w=1.5)
ε =1	11999	6380	15919	2473
ε =0.1	4237	2353	6004	908
ε =0.01	458	282	727	101
ε =0.0001	110	105	247	不收敛

注：初值皆为元素全为 1 的 n-1 维向量

2. 与精确解比较



粗略来看，不同迭代方法所得的结果相差不大，都与精确解相近。



放大来看，SOR(w=1.5)的效果最佳，SOR(w=0.5)的效果最差。

3. 源代码

```
e=1;a=0.5;n=100;h=1/n;
A=ones(n-1,n-1);A=A-A;
A(1,1)=-(2*e+h);
A(1,2)=e+h;
A(n-1,n-1)=-(2*e+h);
A(n-1,n-2)=e;
for i=2:n-2
    A(i,i-1)=e;
    A(i,i)=-(2*e+h);
    A(i,i+1)=e+h;
end
b=ones(n-1,1);
b(n-1)=a*h^2-e-h;
for i=1:n-2
    b(i)=a*h^2;
end
D = diag(diag(A));%求 A 的对角矩阵

L = -tril(A,-1);%求 A 的下三角矩阵

U = -triu(A,1);%求 A 的上三角矩阵
y_true=ones(n-1,1);
for i=1:n-1

y_true(i)=(1-a)*(1-exp(-i*h/e))/(1-exp(-1/e))+a*i*h;
end

%Jacobi
B=D\(L+U);
g=D\b;
y_old=2*ones(n-1,1);y_new=ones(n-1,1);
n1=0;
while norm(y_new-y_old)/norm(y_old)>10^(-6)
    y_old=y_new;
    y_new=B*y_old+g;
    n1=n1+1;
end
y_Jacobi=y_new;
%G-S
B=(D-L)\U;
g=(D-L)\b;
y_old=2*ones(n-1,1);y_new=ones(n-1,1);
```

```
n2=0;
while norm(y_new-y_old)/norm(y_old)>10^(-6)
    y_old=y_new;
    y_new=B*y_old+g;
    n2=n2+1;
end
y_G_S=y_new;
%SOR w=0.5
w=0.5;
B=(D-w*L)\((1-w)*D+w*U);
g=(D-w*L)\(w*b);
y_old=2*ones(n-1,1);y_new=ones(n-1,1);
n3=0;
while norm(y_new-y_old)/norm(y_old)>10^(-6)
    y_old=y_new;
    y_new=B*y_old+g;
    n3=n3+1;
end
y_SOR1=y_new;
%SOR w=1.5
w=1.5;
B=(D-w*L)\((1-w)*D+w*U);
g=(D-w*L)\(w*b);
y_old=2*ones(n-1,1);y_new=ones(n-1,1);
n4=0;
while norm(y_new-y_old)/norm(y_old)>10^(-6)
    y_old=y_new;
    y_new=B*y_old+g;
    n4=n4+1;
end
y_SOR2=y_new;

%compare
x=ones(n-1,1);
for i=1:n-1
    x(i)=i*h;
end
plot(x,y_true,'k');
hold on
plot(x,y_Jacobi,'b');
plot(x,y_G_S,'g');
plot(x,y_SOR1,'m');
plot(x,y_SOR2,'r');
```