电 子 科 技 大 学 实 验 报 告

课程名称： 数学建模实验

实验地点： 科A229

指导教师： 张勇

评 分：

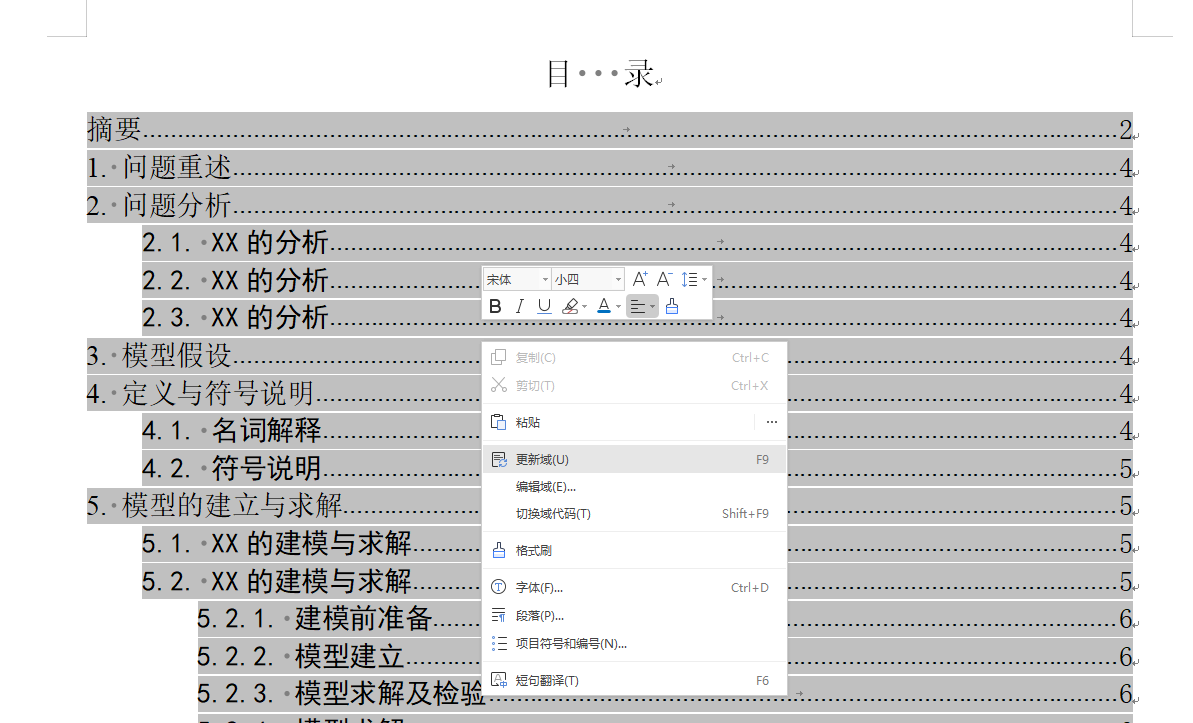
完成实验学生信息：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **姓名** | **学号** | **选课**  **序号** | **贡献百分比/%** | **备注（主要工作**） |
| 1 | 张镕麒 | 2019081301026 | 28 | 100 |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
|  |  |  |  |  |  |

1. 学生人数按照任课教师要求限定；
2. 对于“评价、改进、总结和体会”都要认真填写，和其他内容是评价实验成绩的重要参考。

**注意：**

1. 本次文档增加了“目录”，标题会自动编号。
2. 在目录下方点击“鼠标右键”出现菜单条，点击“更新域”自动更新目录。如下图所示。



1. 排版时标题1-标题3使用“格式刷”保证格式一致。

(排版时，删除本页)

**实验题目名称: 椭球面上两点之间的最短距离**

1. **实验内容**

已知一球面方程,其中,.又已知球面上2点的(300,320,),(450,800,).请编程计算和二点在球面上的最短距离.这里,均大于0.

1. **实验目的**

求解椭圆两点的最短距离。

1. **实验过程**

建模->求解->验证

目 录

[1. 实验1：椭球面上两点之间的最短距离 5](#_Toc13989)

[1.1. 问题分析 5](#_Toc29906)

[1.2. 模型假设 5](#_Toc12843)

[1.3. 模型建立与求解(实验过程) 5](#_Toc3380)

[1.4. 实验结果分析 5](#_Toc4816)

[1.5. 优缺点及改进方向 5](#_Toc3593)

[1.6. 心得体会与总结 5](#_Toc18342)

[2. 对本次实验的设计提出改进意见 5](#_Toc27299)

[附件 6](#_Toc23834)

[附件1.求椭圆两点最短距离的MATLAB程序 6](#_Toc21147)

# 实验1：椭球面上两点之间的最短距离

## 问题分析

所求问题即为对于椭球上给定两点，求其最短的椭球表面距离。

## 模型假设

椭球上足够近的两点，其表面距离可以近似视作实际距离。

## 模型建立与求解(实验过程)

对两点在其xy轴内部的空间内，随机撒点，对于足够近的点，连一条权值为两点实际距离的边，利用SPFA算法求解最短路，求得对应终点的距离值即可近似视作最初两点的距离值。

## 实验结果分析

程序运行结果为ans = 781.7520，即对于题目给定的两点，其近似距离约为782。

## 优缺点及改进方向

1. 应该改进SPFA算法实现的细节，降低时间复杂度常数，从而高效处理更多点的情况，使两点间的距离更加近似等于实际距离，算出更加精确的数值结果。
2. SPFA算法在最短路的实现上更加容易，相较于dijkstra算法所需要的小根堆，队列实现起来更加便捷。
3. 没有考虑对于椭圆长轴点附近的两点结果，不能简单通过在两点的xy坐标范围内撒点获得近似路径经过节点。

## 心得体会与总结

1. 学习到了数值计算方法在生活中的应用。
2. 对最短路算法有了更深刻的理解。

# 对本次实验的设计提出改进意见

可以设计求椭圆中一对近似为对顶点的距离，其最短路径难以轻易判断大致位置，可以使随机撒点求最短路难度更大。

# 附件

## 附件1.求椭圆两点最短距离的MATLAB程序

close all

clear

a=6000;

b=5000;

x=[2200 2900];

y=[3600 3300];

z=b\*sqrt(1-(x.\*x+y.\*y)/(a\*a)); %计算P1, P2的z坐标

f=@(x,y)b\*sqrt(1-(x.\*x+y.\*y)/(a\*a));

v1=[x(1) y(1) z(1)];% 向量 OP1

v2=[x(2) y(2) z(2)];% 向量 OP2

[theta,alpha]=meshgrid(linspace(0,pi/2,50), linspace(0,2\*pi,50));

z=b\*sin(theta);% 根据椭球面参数方程绘制半椭球面

x=a\*cos(theta).\*cos(alpha);

y=a\*cos(theta).\*sin(alpha);

[xp,yp]=meshgrid(min(x):50:max(x),min(y):50:max(y));

zp = real(f(xp,yp));

hold on

tot=100;

x=[];

y=[];

d=[];

st=tot\*tot+1;

ed=tot\*tot+2;

que=[];

dx=(v2(1)-v1(1))/tot;

dy=(v2(2)-v1(2))/tot;

for i=1:tot\*tot+2

d(i)=1000000000;

end

d(st)=0;

for i=1:tot

xx(i)=dx\*i+v1(1)+rand()\*dx;

yy(i)=dy\*i+v1(2)+rand()\*dy;

end

[XX,YY]=meshgrid(xx,yy);

ZZ=real(f(XX,YY));

%mesh(XX,YY,ZZ);

h=0;

t=1;

eps=15;

que(t)=st;

inq=[];

inq()

inq(ed)=0;

inq(st)=1;

XX=reshape(XX,1,length(XX(:)));

YY=reshape(YY,1,length(YY(:)));

ZZ=reshape(ZZ,1,length(ZZ(:)));

XX=[XX v1(1) v2(1)];

YY=[YY v1(2) v2(2)];

ZZ=[ZZ v1(3) v2(3)];

flag=[];

while h~=t

h=h+1;

inq(h)=0;

u=que(h);

for i=1:tot\*tot+2

if(sqrt((ZZ(i)-ZZ(u))^2+(YY(i)-YY(u))^2+(XX(i)-XX(u))^2)<eps)

if d(i)>d(u)+sqrt((ZZ(i)-ZZ(u))^2+(YY(i)-YY(u))^2+(XX(i)-XX(u))^2)

d(i)=d(u)+sqrt((ZZ(i)-ZZ(u))^2+(YY(i)-YY(u))^2+(XX(i)-XX(u))^2);

flag(i)=u;

if(inq(i)==0)

t=t+1;

que(t)=i;

inq(i)=1;

end

end

end

end

end

d(ed)