《传热学》数值计算大作业

二维稳态导热问题的解析解与数值解

姓 名： 刘铭

学 号：2017151613

班 级： 20161516

任课教师： 谭思超

哈尔滨工程大学

核科学与技术学院

2019年10月9日

# 第一题

## 问题描述





设图中，，L1=100.0，L2=80.0

## 一、建立控制方程及定解条件

对上述问题的微分方程及其边界条件为：

x=0，T=T1=tw1

x=L1，T=T1=tw1

y=0，T=T1=tw1

y=L2，T=T2=tw1+tw2\*sin(pi\*x/L1)

该问题的解析解：

通过附件一中C++代码编译运行，将解出的数据通过绘图软件绘制。

## 二、数值离散

区域离散x方向总节点数为N，y方向总节点数为M，区域内任一节点用i,j表示。

## 三、建立代数方程

对于图中所有的内部节点方程可写为：

用i,j节点的二阶中心差分代替上式中的二阶导数，得：



上式整理成迭代形式：



(i=2,3……,N-1)，(j=2,3……,M-1)

## 四、设立迭代初场

补充四个边界上的第一类边界条件得：  (j=1,2,3……,M)

 (j=1,2,3……,M)

 (i=1,2,3……,N)

 (i=1,2,3……,N)

## 五、迭代求解

通过由附件二中C++编写的迭代程序，计算9598次迭代，相对偏差小于代码中所设定的误差最大值.得出的数据通过绘图软件绘制。

## 六、解的分析

将解析解的图形与数值解的图形进行对比，其结果极其相似。

# 第二题

## 问题描述



## 解析解

## 数值解

其边界条件为：

x=0，T=T1=tw1

x=L1，T=T1=tw1

y=0，T=T1=tw1

y=L2，T=T2= tw2

对于所有的内部节点方程可写为：

用i,j节点的二阶中心差分代替上式中的二阶导数，得：



上式整理成迭代形式：



(i=2,3……,N-1)，(j=2,3……,M-1)

## 解的分析

# 第三题

## 问题描述



附件一、问题一解析解代码（C++）

1. #include<bits/stdc++.h>
2. #define pi 3.1415926535
3. #define N 100
4. #define M 100
5. **using** **namespace** std;
6. **int** main()
7. {
8. ofstream fout;
9. **double** L1=100,L2=80,x=0,y=0;
10. **double** tw1=25,tw2=5;
11. **float** len1=L1/(N-1),len2=L2/(M-1);
12. **double** t[M][N],T[M][N];
13. memset(t,0,**sizeof**(t));
14. memset(T,0,**sizeof**(T));
15. **for**(**int** i=0;i<N;i++)
16. {
17. t[i][0]=tw1;
18. t[i][N-1]=tw1+tw2\*sin(pi\*(i-1)/(N-1));
19. }
20. **for**(**int** j=0;j<M;j++)
21. {
22. t[0][j]=tw1;
23. t[M-1][j]=tw1;
24. }
25. **for** (**int** i=1;i<M-1;i++)
26. **for** (**int** j=1;j<M-1;j++)
27. t[i][j]=tw1+tw2\*sin(pi\*(j-1)/(N-1))\*sinh(pi\*(i-1)/(N-1))/sinh(pi\*L2/L1);
28. **for**(**int** j=M-1;j>=0;j--)
29. {
30. **for**(**int** i=0;i<N;i++)
31. fout<<t[i][j]<<',';
32. fout<<endl;
33. }
34. fout.close();
35. **return** 0;
36. }

附件二、问题一数值解代码（C++）

1. #include<bits/stdc++.h>
2. #define N 100
3. #define M 100
4. #define pi 3.1415926535
5. **using** **namespace** std;
6. **int** main()
7. {
8. ofstream fout;
9. fout.open("Problem one data(Numerical Solution).csv");
10. **int** i,j,l,num=1;
11. **float** cha,x,y,x2,y2;
12. **float** t[N][M],a[N][M];
13. memset(t,0,**sizeof**(t));
14. **for**(j=0;j<M;j++)
15. {
16. t[0][j]=25;
17. t[M-1][j]=25;
18. }
19. **for**(i=0;i<N;i++)
20. {
21. t[i][0]=25;
22. t[i][N-1]=25+5\*sin(pi\*i/N);
23. }
24. x=1.0/(N-1);
25. y=1.0/(M-1);
26. cha=1;
27. **while**(cha>1e-6)
28. {
29. **for**(i=0;i<N;i++)
30. **for**(j=0;j<M;j++)
31. a[i][j]=t[i][j];
32. **for**(i=1;i<N-1;i++)
33. **for**(j=1;j<M-1;j++)
34. {
35. x2=pow(x,2);y2=pow(y,2);
36. t[i][j]=0.5\*y2\*(t[i+1][j]+t[i-1][j])/(x2+y2)+0.5\*x2\*(t[i][j+1]+t[i][j-1])/(x2+y2);
37. }
38. cha=0;
39. **for**(i=0;i<N;i++)
40. **for**(j=0;j<M;j++)
41. cha=cha+abs(a[i][j]-t[i][j]);
42. cha=cha/(N\*M);
43. cout<<num++<<' '<<cha<<endl;
44. }
45. l=0;
46. **for**(j=M-1;j>=0;j--)
47. {
48. **for**(i=0;i<N;i++)
49. fout<<t[i][j]<<',';
50. fout<<endl;
51. }
52. fout.close();
53. **return** 0;
54. }

附件三、问题二解析解代码

附件四、问题二数值解代码（C++）

1. #include<bits/stdc++.h>
2. #define N 100
3. #define M 100
4. #define pi 3.1415926535
5. **using** **namespace** std;
6. **int** main()
7. {
8. ofstream fout;
9. fout.open("Problem two data(Numerical Solution).csv");
10. **int** i,j,l,num=1;
11. **float** cha,x,y,x2,y2;
12. **float** t[N][M],a[N][M];
13. **for**(i=0;i<N;i++)
14. **for**(j=0;j<M;j++)
15. t[i][j]=0;
16. **for**(j=0;j<M;j++)
17. {
18. t[0][j]=25;
19. t[M-1][j]=25;
20. }
21. **for**(i=0;i<N;i++)
22. {
23. t[i][0]=25;
24. t[i][N-1]=5;
25. }
26. x=1.0/(N-1);
27. y=1.0/(M-1);
28. cha=1;
29. **while**(cha>1e-6)
30. {
31. **for**(i=0;i<N;i++)
32. **for**(j=0;j<M;j++)
33. a[i][j]=t[i][j];
34. **for**(i=1;i<N-1;i++)
35. **for**(j=1;j<M-1;j++)
36. {
37. x2=pow(x,2);y2=pow(y,2);
38. t[i][j]=0.5\*y2\*(t[i+1][j]+t[i-1][j])/(x2+y2)+0.5\*x2\*(t[i][j+1]+t[i][j-1])/(x2+y2);
39. }
40. cha=0;
41. **for**(i=0;i<N;i++)
42. **for**(j=0;j<M;j++)
43. cha=cha+abs(a[i][j]-t[i][j]);
44. cha=cha/(N\*M);
45. cout<<num++<<' '<<cha<<endl;
46. }
47. **for**(j=M-1;j>=0;j--)
48. {
49. **for**(i=0;i<N;i++)
50. fout<<t[i][j]<<',';
51. fout<<endl;
52. }
53. fout.close();
54. **return** 0;
55. }