|  |
| --- |
|  |
| 计算流体力学 |
| 本科生大作业 |
|  |

|  |
| --- |
|  |

目录

[第一部分 大作业内容和要求 2](#_Toc312873843)

[一、模拟对象 2](#_Toc312873844)

[二、控制方程（准一维Euler方程） 2](#_Toc312873845)

[三、差分格式（可以用有限体积格式） 2](#_Toc312873846)

[四、来流条件 3](#_Toc312873847)

[五、网格步长 3](#_Toc312873848)

[六、边界条件（两种流态） 3](#_Toc312873849)

[七、收敛准则（采用时间推进法） 3](#_Toc312873850)

[八、要求 3](#_Toc312873852)

[第二部分 实验报告 4](#_Toc312873853)

[一、变换控制方程 4](#_Toc312873854)

[二、差分格式补充方程 4](#_Toc312873855)

[三、设定网格步长 6](#_Toc312873856)

[四、编程思路 7](#_Toc312873857)

[第三部分 实验数据 8](#_Toc312873858)

[一、出口超音条件下静压P沿轴向分布 8](#_Toc312873859)

[二、出口超音条件下马赫数Ma沿轴向分布 8](#_Toc312873860)

[三、出口超音条件下残差减小过程 9](#_Toc312873861)

[四、出口亚音条件下静压P沿轴向分布 10](#_Toc312873862)

[五、出口亚音条件下马赫数Ma沿轴向分布 10](#_Toc312873863)

[六、出口亚音条件下残差减小过程 11](#_Toc312873864)

[七、出口超音条件下运算结果 12](#_Toc312873865)

[八、出口亚音条件下运算结果 12](#_Toc312873866)

[第四部分 C语言程序 13](#_Toc312873867)

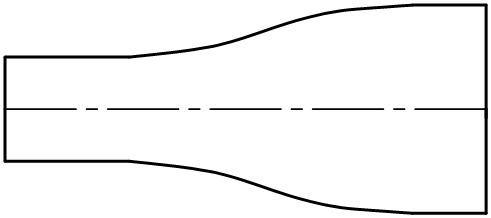
[一、出口超音条件 13](#_Toc312873868)

[二、出口亚音条件 17](#_Toc312873869)

第一部分 大作业内容和要求

用MC格式计算二维管流

一、模拟对象



inflow

outflow



二、控制方程（准一维Euler方程）

（一） 一维Euler方程



式中：

 ,  , 

其中，,是管道截面积。

补充状态方程

 ，

三、差分格式（可以用有限体积格式）

式中，α为人工粘性系数，α=0.1~1.0，可以根据具体情况再加大或减小。

四、来流条件

；；

进口：；出口：（单位：m）

五、网格步长



1，空间步长：

2，时间步长：



a为音速。

六、边界条件（两种流态）

1．进口超音，出口超音；

2．进口超音，出口亚音（）。

七、收敛准则（采用时间推进法）

残差：



八、要求

1.写出实验报告（数值实验）；

2.给出沿轴向分布；

3.给出残差Error的减小过程；

4.附上程序。

第二部分 实验报告

一、变换控制方程

为保证迭代过程中变量的一致性,便于迭代，需将一维Euler方程中的F与H变换为用U的分量表示的形式，同时将ρ、u、e、P用U的分量表示出来。

由和可变化得:





   。

二、差分格式补充方程

为保证迭代过程中变量的一致性，需将MC格式中的非U变量用U表示出来。故需补充方程：

1.预测步







2.校正步







3.残差表示



三、设定网格步长

沿轴向取101个点，共100个网格，每个网格步长



时间步长取

人工粘性系数初选α=0.5。

四、编程思路

1. 用数组表示各点的变量。
2. 给每个点的变量赋初值。
3. 按MC格式进行迭代，每次迭代各变量都要从0至100点循环一遍。
4. 用比较函数得出每次迭代各点的残差值中的最大的，当最大残差值小于等于0.0001时，循环结束。
5. 将P与M用U的终值求出。
6. 将每迭代一次的残差与最终P、M的值输出到txt文件中。

第三部分 实验数据

一、出口超音条件下静压P沿轴向分布

轴向分布 / 0.1m

二、出口超音条件下马赫数Ma沿轴向分布

三、出口超音条件下最大残差减小过程

轴向分布 / 0.1m

迭代次数

迭代次数

四、出口亚音条件下静压P沿轴向分布

轴向分布 / 0.1m

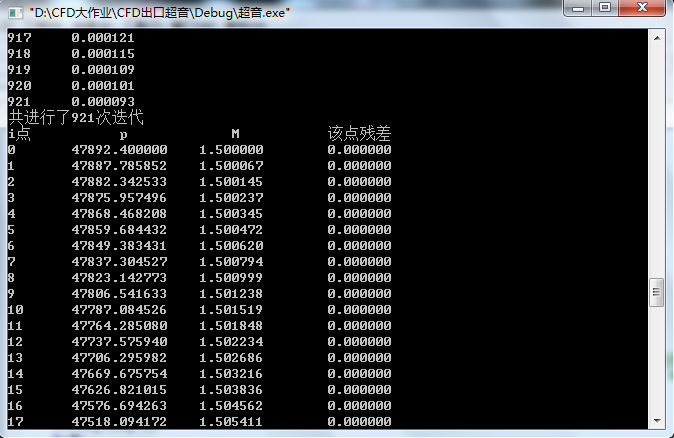
五、出口亚音条件下马赫数Ma沿轴向分布

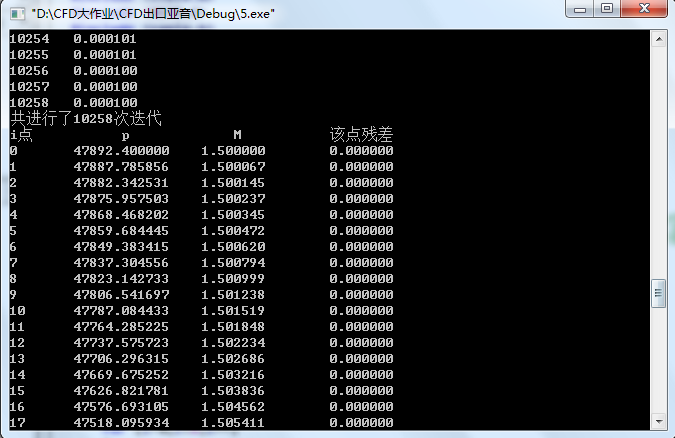
轴向分布 / 0.1m

六、出口亚音条件下最大残差减小过程

迭代次数

迭代次数

七、出口超音条件下运算结果

八、出口亚音条件下运算结果

第四部分 C语言程序

一、出口超音条件

#include <STDIO.H>

#include <MATH.H>

#include <conio.h>

#define W 101 //定义网格为100个点//

#define K 1.4 //绝热系数//

#define T 0.0001 //时间步长//

#define X 0.1 //空间步长//

#define a 0.5 //粘性系数//

void main()

{

FILE \*fout1,\*fout2;

fout1=fopen("d:\\超音残差.txt","w");//将数据输出到超音残差.txt

fout2=fopen("d:\\超音参数.txt","w");//将数据输出到超音参数.txt

double U1[W],U2[W],U3[W],UU1[W],UU2[W],UU3[W],F1[W],F2[W],F3[W],

FF1[W],FF2[W],FF3[W],H1[W],H2[W],H3[W],HH1[W],HH2[W],HH3[W],

//定义函数变量Ui、Fi、Hi以及加横杠的Ui、Fi、Hi//

u[W],p[W],b[W],e[W],M[W],A[W],AA[W],ERROR[W],max;

//b为密度，M为马赫数,A为面积函数值，AA为面积倒数值//

int i,j,k;

ERROR[0]=0;

ERROR[100]=0;

k=0;

//给定入口初值条件//

p[0]=47892.40;b[0]=1.2218;M[0]=1.5;

u[0]=M[0]\*sqrt(K\*p[0]/b[0]);

e[0]=p[0]/(K-1)+0.5\*b[0]\*u[0]\*u[0];

//给定各点初值//

for (i=0;i<W;i++)

{

A[i]=1.398+0.347\*tanh(0.8\*X\*i-4.0);

AA[i]=0.347\*0.8\*(1-tanh(0.8\*X\*i-4.0)\*tanh(0.8\*X\*i-4.0));

U1[i]=A[i]\*b[0];

U2[i]=A[i]\*b[0]\*u[0];

U3[i]=A[i]\*e[0];

F1[i]=U2[0];

F2[i]=(3-K)\*U2[0]\*U2[0]/2/U1[0]+(K-1)\*U3[0];

F3[i]=(K\*U3[0]-(K-1)\*U2[0]\*U2[0]/2/U1[0])\*U2[0]/U1[0];

H1[i]=0;

H2[i]=AA[i]\*(K-1)\*(U3[0]-0.5\*U2[0]\*U2[0]/U1[0])/A[i];

H3[i]=0;

}

for (i=0;i<W;i++)

{

UU1[i]=U1[1]-T\*(F1[1]-F1[0])/X+H1[1]\*T+a\*T\*(U1[2]-2\*U1[1]+U1[0])/X/X; UU2[i]=U2[1]-T\*(F2[1]-F2[0])/X+H2[1]\*T+a\*T\*(U2[2]-2\*U2[1]+U2[0])/X/X UU3[i]=U3[1]-T\*(F3[1]-F3[0])/X+H3[1]\*T+a\*T\*(U3[2]-2\*U3[1]+U3[0])/X/X;

FF1[i]=UU2[1];

FF2[i]=(3-K)\*UU2[1]\*UU2[1]/2/UU1[1]+(K-1)\*UU3[1];

FF3[i]=(K\*UU3[1]-(K-1)\*UU2[1]\*UU2[1]/2/UU1[1])\*UU2[1]/UU1[1];

HH1[i]=0;

HH2[i]=AA[i]\*(K-1)\*(UU3[1]-0.5\*UU2[1]\*UU2[1]/UU1[1])/A[i];

HH3[i]=0;

}

//迭代求解//

do

{

for (i=1;i<W;i++)

{

F1[i]=U2[i];

F2[i]=(3-K)\*U2[i]\*U2[i]/2/U1[i]+(K-1)\*U3[i];

F3[i]=(K\*U3[i]-(K-1)\*U2[i]\*U2[i]/2/U1[i])\*U2[i]/U1[i];

H1[i]=0;

H2[i]=AA[i]\*(K-1)\*(U3[i]-0.5\*U2[i]\*U2[i]/U1[i])/A[i];

H3[i]=0;

}

//出口参数等于前一点参数//

U1[100]=U1[99];

U2[100]=U2[99];

U3[100]=U3[99];

//预测步//

for (i=1;i<W-1;i++)

{

UU1[i]=U1[i]-T\*(F1[i]-F1[i-1])/X+H1[i]\*T+a\*T\*(U1[i+1]-2\*U1[i]+U1[i-1])/X/X;

UU2[i]=U2[i]-T\*(F2[i]-F2[i-1])/X+H2[i]\*T+a\*T\*(U2[i+1]-2\*U2[i]+U2[i-1])/X/X;

UU3[i]=U3[i]-T\*(F3[i]-F3[i-1])/X+H3[i]\*T+a\*T\*(U3[i+1]-2\*U3[i]+U3[i-1])/X/X;

}

for (i=1;i<W;i++)

{

FF1[i]=UU2[i];

FF2[i]=(3-K)\*UU2[i]\*UU2[i]/2/UU1[i]+(K-1)\*UU3[i];

FF3[i]=(K\*UU3[i]-(K-1)\*UU2[i]\*UU2[i]/2/UU1[i])\*UU2[i]/UU1[i];

HH1[i]=0;

HH2[i]=AA[i]\*(K-1)\*(UU3[i]-0.5\*UU2[i]\*UU2[i]/UU1[i])/A[i];

HH3[i]=0;

}

//出口参数等于前一点参数//

UU1[100]=U1[100]-T\*(F1[100]-F1[99])/X+H1[100]\*T+a\*T\*(U1[100]-2\*U1[100]+U1[99])/X/X;

UU2[100]=U2[100]-T\*(F2[100]-F2[99])/X+H2[100]\*T+a\*T\*(U2[100]-2\*U2[100]+U2[99])/X/X;

UU3[100]=U3[100]-T\*(F3[100]-F3[99])/X+H3[100]\*T+a\*T\*(U3[100]-2\*U3[100]+U3[99])/X/X;

//校正步//

for (i=1;i<W-1;i++)

{

U1[i]=0.5\*(U1[i]+UU1[i]-T\*(FF1[i+1]-FF1[i])/X+HH1[i]\*T+a\*T\*(UU1[i+1]-2\*UU1[i]+UU1[i-1])/X/X);

U2[i]=0.5\*(U2[i]+UU2[i]-T\*(FF2[i+1]-FF2[i])/X+HH2[i]\*T+a\*T\*(UU2[i+1]-2\*UU2[i]+UU2[i-1])/X/X);

U3[i]=0.5\*(U3[i]+UU3[i]-T\*(FF3[i+1]-FF3[i])/X+HH3[i]\*T+a\*T\*(UU3[i+1]-2\*UU3[i]+UU3[i-1])/X/X);

//求残差//

ERROR[i]=fabs((-U3[i]+UU3[i]-T\*(FF3[i+1]-FF3[i])/X+HH3[i]\*T+a\*T\*(UU3[i+1]-2\*UU3[i]+UU3[i-1])/X/X)/A[i]);

//即(2\*U3n+1 -U3n)-U3n+1 等价于U3n+1 - U3n//

}

//求最大残差//

max=ERROR[1];

for (j=1;j<W-1;j++)

{

if (max<ERROR[j]) max=ERROR[j];

}

k=k+1; //迭代计数//

//显示该次运算最大残差//

printf("%d\t%f\n",k,max);

fprintf(fout1,"%d\t%f\n",k,max);

}

while (max>=0.0001);

//循环结束//

//显示迭代次数//

printf("共进行了%d次迭代\n",k);

//显示运算结果//

printf("i点 \t p \t M \t该点残差\n");

for (i=0;i<W;i++)

{

p[i]=(K-1)\*(U3[i]-0.5\*U2[i]\*U2[i]/U1[i])/A[i];

b[i]=U1[i]/A[i];

u[i]=U2[i]/U1[i];

e[i]=U3[i]/A[i];

M[i]=u[i]/sqrt(K\*p[i]/b[i]);

printf("%d\t%f\t%f\t%f\n",i,p[i],M[i],ERROR[i]);

fprintf(fout2,"%d\t%f\t%f\t%f\n",i,p[i],M[i],ERROR[i]);

}

printf("程序结束\n");

}

二、出口亚音条件

#include <STDIO.H>

#include <MATH.H>

#include <conio.h>

#define W 101 //定义网格为100个点//

#define K 1.4 //绝热系数//

#define T 0.0001 //时间步长//

#define X 0.1 //空间步长//

#define a 0.5 //粘性系数//

void main()

{

FILE \*fout1,\*fout2;

fout1=fopen("d:\\亚音残差.txt","w");//将数据输出到亚音残差.txt

fout2=fopen("d:\\亚音参数.txt","w");//将数据输出到亚音参数.txt

double U1[W],U2[W],U3[W],UU1[W],UU2[W],UU3[W],F1[W],F2[W],F3[W],

FF1[W],FF2[W],FF3[W],H1[W],H2[W],H3[W],HH1[W],HH2[W],HH3[W],

//定义函数变量Ui、Fi、Hi以及加横杠的Ui、Fi、Hi//

u[W],p[W],b[W],e[W],M[W],A[W],AA[W],ERROR[W],max;

//b为密度，M为马赫数,A为面积函数值，AA为面积倒数值//

int i,j,k;

ERROR[0]=0;

ERROR[100]=0;

k=0;

//给定入口初值条件//

p[0]=47892.40;b[0]=1.2218;M[0]=1.5;

u[0]=M[0]\*sqrt(K\*p[0]/b[0]);

e[0]=p[0]/(K-1)+0.5\*b[0]\*u[0]\*u[0];

//给定出口初值条件//

u[100]=119;

//给定各点初值//

for (i=0;i<W;i++)

{

A[i]=1.398+0.347\*tanh(0.8\*X\*i-4.0);

AA[i]=0.347\*0.8\*(1-tanh(0.8\*X\*i-4.0)\*tanh(0.8\*X\*i-4.0));

U1[i]=A[i]\*b[0];

U2[i]=A[i]\*b[0]\*u[0];

U3[i]=A[i]\*e[0];

F1[i]=U2[0];

F2[i]=(3-K)\*U2[0]\*U2[0]/2/U1[0]+(K-1)\*U3[0];

F3[i]=(K\*U3[0]-(K-1)\*U2[0]\*U2[0]/2/U1[0])\*U2[0]/U1[0];

H1[i]=0;

H2[i]=AA[i]\*(K-1)\*(U3[0]-0.5\*U2[0]\*U2[0]/U1[0])/A[i];

H3[i]=0;

}

for (i=0;i<W;i++)

{

UU1[i]=U1[1]-T\*(F1[1]-F1[0])/X+H1[1]\*T+a\*T\*(U1[2]-2\*U1[1]+U1[0])/X/X;

UU2[i]=U2[1]-T\*(F2[1]-F2[0])/X+H2[1]\*T+a\*T\*(U2[2]-2\*U2[1]+U2[0])/X/X;

UU3[i]=U3[1]-T\*(F3[1]-F3[0])/X+H3[1]\*T+a\*T\*(U3[2]-2\*U3[1]+U3[0])/X/X;

FF1[i]=UU2[1];

FF2[i]=(3-K)\*UU2[1]\*UU2[1]/2/UU1[1]+(K-1)\*UU3[1];

FF3[i]=(K\*UU3[1]-(K-1)\*UU2[1]\*UU2[1]/2/UU1[1])\*UU2[1]/UU1[1];

HH1[i]=0;

HH2[i]=AA[i]\*(K-1)\*(UU3[1]-0.5\*UU2[1]\*UU2[1]/UU1[1])/A[i];

HH3[i]=0;

}

//迭代求解//

do

{

for (i=1;i<W;i++)

{

F1[i]=U2[i];

F2[i]=(3-K)\*U2[i]\*U2[i]/2/U1[i]+(K-1)\*U3[i];

F3[i]=(K\*U3[i]-(K-1)\*U2[i]\*U2[i]/2/U1[i])\*U2[i]/U1[i];

H1[i]=0;

H2[i]=AA[i]\*(K-1)\*(U3[i]-0.5\*U2[i]\*U2[i]/U1[i])/A[i];

H3[i]=0;

}

//出口速度恒定，其他参数等于前一点参数//

b[99]=U1[99]/A[99];

e[99]=U3[99]/A[99];

U1[100]=A[100]\*b[99];

U2[100]=A[100]\*b[99]\*u[100];

U3[100]=A[100]\*e[99];

//预测步//

for (i=1;i<W-1;i++)

{

UU1[i]=U1[i]-T\*(F1[i]-F1[i-1])/X+H1[i]\*T+a\*T\*(U1[i+1]-2\*U1[i]+U1[i-1])/X/X;

UU2[i]=U2[i]-T\*(F2[i]-F2[i-1])/X+H2[i]\*T+a\*T\*(U2[i+1]-2\*U2[i]+U2[i-1])/X/X;

UU3[i]=U3[i]-T\*(F3[i]-F3[i-1])/X+H3[i]\*T+a\*T\*(U3[i+1]-2\*U3[i]+U3[i-1])/X/X;

}

for (i=1;i<W;i++)

{

FF1[i]=UU2[i];

FF2[i]=(3-K)\*UU2[i]\*UU2[i]/2/UU1[i]+(K-1)\*UU3[i];

FF3[i]=(K\*UU3[i]-(K-1)\*UU2[i]\*UU2[i]/2/UU1[i])\*UU2[i]/UU1[i];

HH1[i]=0;

HH2[i]=AA[i]\*(K-1)\*(UU3[i]-0.5\*UU2[i]\*UU2[i]/UU1[i])/A[i];

HH3[i]=0;

}

//保障出口速度恒定//

UU1[100]=U1[100]-T\*(F1[100]-F1[99])/X+H1[100]\*T+a\*T\*(U1[100]-2\*U1[100]+U1[99])/X/X;

UU2[100]=U2[100]-T\*(F2[100]-F2[99])/X+H2[100]\*T+a\*T\*(U2[100]-2\*U2[100]+U2[99])/X/X;

UU3[100]=U3[100]-T\*(F3[100]-F3[99])/X+H3[100]\*T+a\*T\*(U3[100]-2\*U3[100]+U3[99])/X/X;

//校正步//

for (i=1;i<W-1;i++)

{

U1[i]=0.5\*(U1[i]+UU1[i]-T\*(FF1[i+1]-FF1[i])/X+HH1[i]\*T+a\*T\*(UU1[i+1]-2\*UU1[i]+UU1[i-1])/X/X);

U2[i]=0.5\*(U2[i]+UU2[i]-T\*(FF2[i+1]-FF2[i])/X+HH2[i]\*T+a\*T\*(UU2[i+1]-2\*UU2[i]+UU2[i-1])/X/X);

U3[i]=0.5\*(U3[i]+UU3[i]-T\*(FF3[i+1]-FF3[i])/X+HH3[i]\*T+a\*T\*(UU3[i+1]-2\*UU3[i]+UU3[i-1])/X/X);

//求残差//

ERROR[i]=fabs((-U3[i]+UU3[i]-T\*(FF3[i+1]-FF3[i])/X+HH3[i]\*T+a\*T\*(UU3[i+1]-2\*UU3[i]+UU3[i-1])/X/X)/A[i]);

//即(2\*U3n+1 -U3n)-U3n+1 等价于U3n+1 - U3n//

}

//求最大残差//

max=ERROR[1];

for (j=1;j<W-1;j++)

{

if (max<ERROR[j]) max=ERROR[j];

}

k=k+1; //迭代计数//

//显示该次运算最大残差//

printf("%d\t%f\n",k,max);

fprintf(fout1,"%d\t%f\n",k,max);

}

while (max>=0.0001);

//循环结束//

//显示迭代次数//

printf("共进行了%d次迭代\n",k);

//显示运算结果//

printf("i点 \t p \t M \t该点残差\n");

for (i=0;i<W;i++)

{

p[i]=(K-1)\*(U3[i]-0.5\*U2[i]\*U2[i]/U1[i])/A[i];

b[i]=U1[i]/A[i];

u[i]=U2[i]/U1[i];

e[i]=U3[i]/A[i];

M[i]=u[i]/sqrt(K\*p[i]/b[i]);

printf("%d\t%f\t%f\t%f\n",i,p[i],M[i],ERROR[i]);

fprintf(fout2,"%d\t%f\t%f\t%f\n",i,p[i],M[i],ERROR[i]);

}

printf("程序结束\n");

}