

**网格划分**

对计算域进行网格划分，共划分为5个网格，见图2。

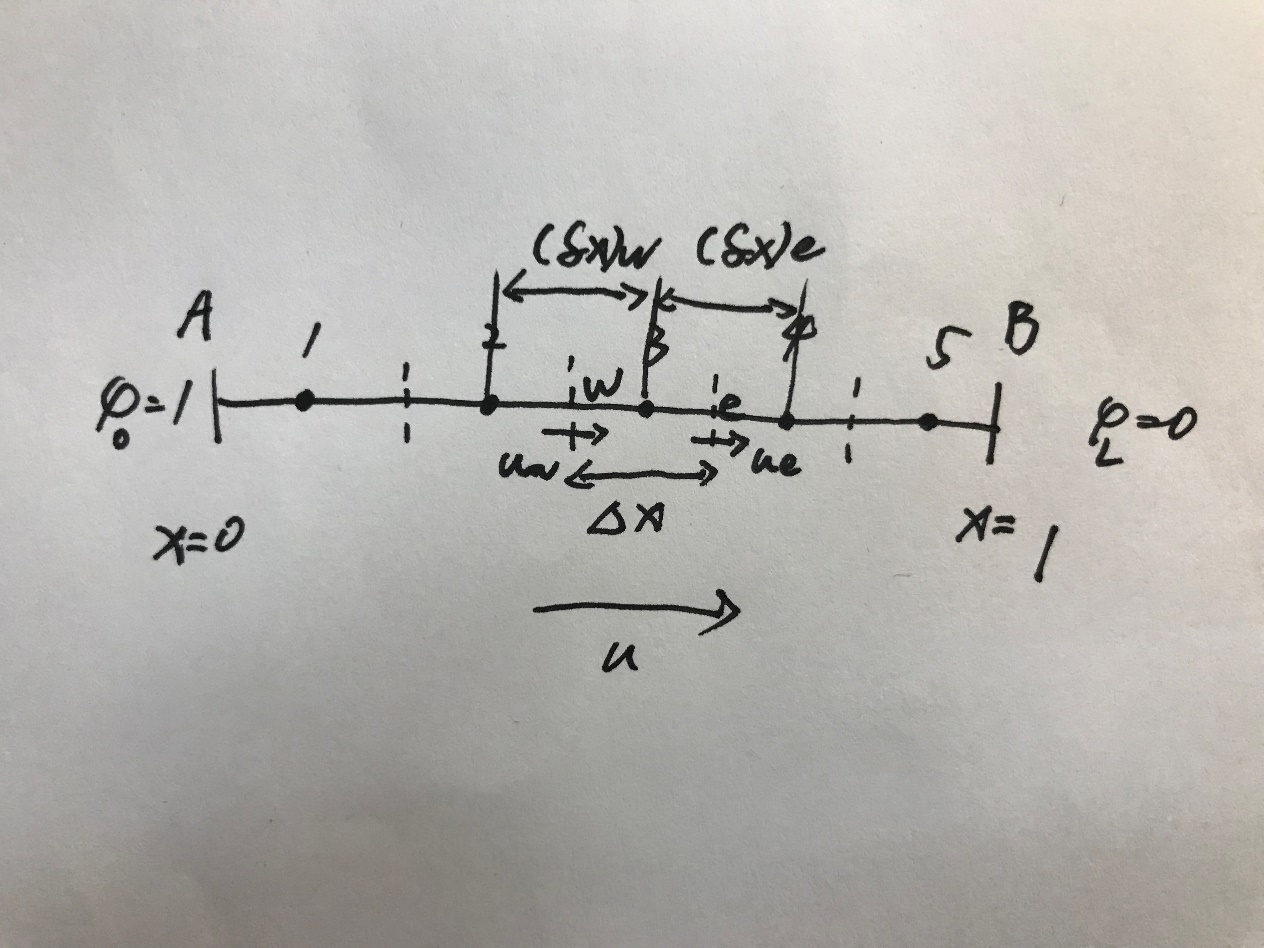


图2 计算域网格划分

**方程离散格式**

由方程（1）可得

,(5)

即：

,

, (6)

定义Peclet数*P=F*/*D*，其中*F=ρu*，*D=Γφ*/*δx*

1.1 中心差分

界面处的值：

由（6）可得：

,(7)

化为标准形式：,则有，

, , ,

已知，则有：

对于第一个控制体：

,

则

, ,

同理对最后一个控制体有：

,

则有：

, ,

1.2 上风格式

对于界面处的有：

,

,

带入方程（6）则有：

=,(8)

则,

, ,

对于边界处的界面上值：

由边界条件可得对于第一个控制体有：

, ,

对于最后一个控制体有：

, ,

1.3 混合格式

控制体界面处的值：

当时，其形式与中心差分相同；当时，其形式与上风格式相同，整理可得，

, ,

**计算结果**

***u*=0.1 m/s**

中心差分计算矩阵为：

上风格式计算矩阵为：

精确解

表1不同格式计算结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 精 确 解 | 0.938793 | 0.796390 | 0.622459 | 0.410020 | 0.150545 |
| 中心差分 | 0.942100 | 0.800601 | 0.627646 | 0.416256 | 0.157890 |
| 上风格式 | 0.933733 | 0.787947 | 0.613003 | 0.403071 | 0.151151 |



图3 速度为0.1m/s时计算结果与精确解的比较

定义误差为

表2速度为0.1m/s时计算误差

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 中心差分 | 0.003307 | 0.004211 | 0.005187 | 0.006236 | 0.007345 |
| 上风格式 | 0.00506 | 0.008443 | 0.009456 | 0.006949 | 0.000606 |



图4 速度为0.1m/s时计算误差

可见在0.1 m/s 的速度下，两种离散格式的计算结果与精确解符合的程度都比较好。在计算范围内，两者计算误差也都很小，在x<0.7 的范围内中心差分的格式的误差相对较小，在x>0.7的范围内上风格式的误差相对较小。

***u*=2.5 *m/s***

中心差分计算矩阵为：

上风格式计算矩阵为：

混合格式计算矩阵为：

精确解

表3不同格式计算结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 精 确 解 | 1.000000 | 1.000000 | 0.999996 | 0.999447 | 0.917915 |
| 中心差分 | 1.035630 | 0.869355 | 1.257331 | 0.352053 | 2.464370 |
| 上风格式 | 0.999843 | 0.998740 | 0.992126 | 0.952411 | 0.714331 |
| 混合格式 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 0.714286 |



图5 速度为2.5m/s时计算结果与精确解的比较

表4 速度为2.5 m/s时计算误差

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 中心差分 | 0.03563 | 0.13065 | 0.257335 | 0.64739 | 1.546455 |
| 上风格式 | 0.000157 | 0.00126 | 0.00787 | 0.047036 | 0.203584 |
| 混合格式 | 0 | 0 | 0.000004 | 0.00055 | 0.203629 |



图6 速度为2.5m/s时的计算误差

表3和图5为速度为2.5 m/s不同格式的计算结果。可以看出，当速度较大时，上风格式和混合格式的计算结果基本相同，与精确解较为接近，且在x较小的时候拟合的较好，随着x的增加，误差较大。而采用中心差分格式则得到发散的结果；从误差图也可看出，随着x增加，三种格式的误差都有所增加，但上风格式和混合格式的误差都比较小，而中心差分格式的误差增长迅速，远超过其他两种格式。

附件：程序代码

program main

implicit none

integer,parameter::N=5

integer method

real\*4,dimension(N)::Fe,Fw,De,Dw,AE,AW,AP,p,y,x,f,u,a

real\*4,parameter::r=1.0,g=0.1,lt=1.0,pA=1.0,pB=0.0

real\*4 V

real\*4 dx,L(2:N),d(1:N-1),c(1:N-1),e(2:N),F0,D0

integer::i

write(\*,\*) '输入选择:1. 中心差分；2. 上风格式；3. 混合格式'

read(\*,\*) method

write(\*,\*) '速度 V='

read(\*,\*) V

dx=lt/N

F0=r\*V

D0=g/dx

do i=1,N

Fe(i)=F0

De(i)=D0

Fw(i)=F0

Dw(i)=D0

end do

De(N)=2\*D0

Dw(1)=2\*D0

!write(\*,\*) (DW(i),i=1,N)

select case(method)

case(1)

do i=1,N

AW(i)=Dw(i)+0.5\*Fw(i)

AE(i)=De(i)-0.5\*Fe(i)

end do

AW(1)=Dw(1)+Fw(1)

AE(N)=De(N)-Fe(N)

!write(\*,\*) (AW(i),i=1,N)

case(2)

do i=1,N

AW(i)=Dw(i)+max(Fw(i),0.0)

AE(i)=De(i)+max(-Fe(i),0.0)

end do

AW(1)=Dw(1)+Fw(1)

AE(N)=De(N)

case(3)

do i=1,N

AW(i)=max(Fw(i),Dw(i)+0.5\*Fw(i),0.0)

AE(i)=max(-Fe(i),De(i)-0.5\*Fe(i),0.0)

end do

AW(1)=Dw(1)+Fw(1)

AE(N)=De(N)

end select

do i=1,N

AP(i)=AE(i)+AW(i)+(Fe(i)-Fw(i))

end do

write(\*,200) AW,AE,AP

200 format('AW=',5(/F10.5),/,'AE=',5(/F10.5),/,'AP=',5(/F10.5))

do i=1,N

p(i)=1-(exp(r\*V\*(i-0.5)\*dx/g)-1)/(exp(r\*V/g)-1)

end do

do i=1,N

a(i)=AP(i)

f(i)=0

end do

f(1)=AW(1)\*pA

f(N)=AE(N)\*pB

write(\*,210) f

210 format('f=',5(/F10.5))

do i=1,N-1

c(i)=-AE(i)

end do

do i=2,N

e(i)=-AW(i)

end do

do i=1,N-1

d(i)=c(i)

end do

u(1)=a(1)

do i=2,N

L(i)=e(i)/u(i-1)

u(i)=a(i)-L(i)\*c(i-1)

end do

y(1)=f(1)

do i=2,N

y(i)=f(i)-L(i)\*y(i-1)

end do

x(n)=y(n)/u(n)

do i=n-1,1,-1

x(i)=(y(i)-c(i)\*x(i+1))/u(i)

end do

write(\*,100) x,p

100 format(/,'数值解：',/,5(/F10.6),/,'解析解：',/,5(/F10.6))

end program