已知变量 φ 的输运过程由如下对流—扩散方程控制：

$$\frac{\mathrm{d}}{\mathrm{d}x}(\rho u\varphi)=\frac{\mathrm{d}}{\mathrm{d}x}\left(\Gamma\frac{\mathrm{d}\varphi}{\mathrm{d}x}\right) \tag{1}$$

$$\frac{\mathrm{d}(\rho u)}{\mathrm{d}x}=0 \tag{2}$$

对于如图 1 所示的计算域（$x\in[0,1]$），其对应的边界条件为：

$$\begin{cases}\varphi_0=1, & x=0\\ \varphi_L=0, & x=L\end{cases} \tag{3}$$

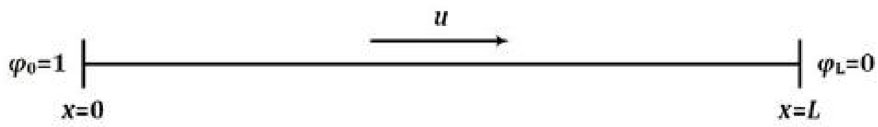


图 1 计算域示意图

该方程的精确解为：

$$\frac{\varphi-\varphi_0}{\varphi_L-\varphi_0}=\frac{\exp(\rho ux/\Gamma)-1}{\exp(\rho uL/\Gamma)-1} \tag{4}$$

若**已知**：L=1.0 m、ρ=1.0 kg/m^3、Γ=0.1 kg/(m·s)，**要求**：（1）将该计算域划分为 5 个均分的网格，分别采用中心差分格式、上风格式、混合格式（仅对 u=2.5 m/s）三种离散格式，在（i）u=0.1 m/s、（ii）u=2.5 m/s 的条件下进行求解；（2）采用 Fortran 语言编写计算机程序代码；（3）分别以图和表的形式，将上述三种离散格式的计算结果与精确解[式（4）]的计算结果进行比较。

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

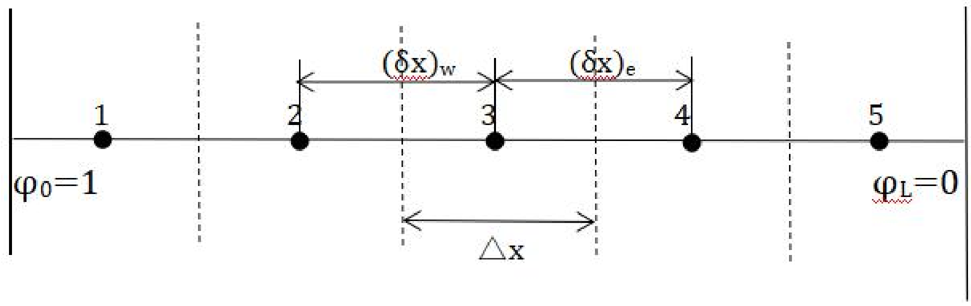


图 2 计算域网格划分

方程离散格式

由方程（1）可得

$$\int_w^e \frac{d}{dx}(\rho u\varphi)dx = \int_w^e \frac{d}{dx}(\Gamma\frac{d\varphi}{dx})dx \quad ,(5)$$

即：

$$(\rho u\varphi)_e - (\rho u\varphi)_w = \left(\Gamma_\varphi\frac{d\varphi}{dx}\right)_e - (\Gamma_\varphi\frac{d\varphi}{dx})_w ,$$

$$(\rho u\varphi)_e - (\rho u\varphi)_w = \Gamma_{\varphi,e}\frac{\varphi_E-\varphi_P}{(\delta x)_e} - \Gamma_{\varphi,w}\frac{\varphi_p-\varphi_w}{(\delta x)_w} \quad ,(6)$$

定义 Peclet 数 $P=F/D$，其中 $F=\rho u$，$D=\Gamma_\varphi/\delta x$

1.1 中心差分

界面处的φ值：

$$\begin{cases}\varphi_e=(\varphi_E+\varphi_P)/2\\ \varphi_w=(\varphi_W+\varphi_P)/2\end{cases}$$

由（6）可得：

$$(\rho u)_e\frac{(\varphi_E+\varphi_P)}{2} - (\rho u)_w\frac{(\varphi_W+\varphi_P)}{2} = \Gamma_{\varphi,e}\frac{\varphi_E-\varphi_P}{(\delta x)_e} - \Gamma_{\varphi,w}\frac{\varphi_p-\varphi_w}{(\delta x)_w} \quad ,(7)$$

化为标准形式：$a_P\varphi_P = a_W\varphi_W + a_E\varphi_E$,则有，

$a_E = D_e - F_e/2$, $a_W = D_w + F_w/2$, $a_P = a_E + a_W + (F_e - F_w)$,

已知$\varphi_0 = 1，\varphi_L = 0$，则有：

对于第一个控制体：

$$\begin{cases}\varphi_e = (\varphi_E + \varphi_P)/2 \\ \varphi_w = \varphi_0\end{cases},$$

则

$$a_P\varphi_P = a_W\varphi_W + a_E\varphi_E$$

$$a_E = D_e - \frac{F_e}{2},\ a_W\varphi_W = (2D_w + F_w)\varphi_0,\ a_P = D_e + \frac{F_e}{2} + 2D_w$$

同理对最后一个控制体有：

$$\begin{cases}\varphi_e = \varphi_L \\ \varphi_w = (\varphi_W + \varphi_P)/2\end{cases},$$

则有：

$$a_P\varphi_P = a_W\varphi_W + a_E\varphi_E$$

$$a_W = D_w - \frac{F_w}{2},\ a_E\varphi_E = (2D_e - F_e)\varphi_L,\ a_P = D_w - \frac{F_e}{2} + 2D_e$$

1.2 上风格式

对于界面处的φ有：

$$\varphi_e = \begin{cases}\varphi_P, & F_e > 0 \\ \varphi_E, & F_e < 0\end{cases},$$

$$\begin{cases}F_e\varphi_e = \varphi_P[\![F_e, 0]\!] - \varphi_E[\![-F_e, 0]\!] \\ F_w\varphi_w = \varphi_W[\![F_w, 0]\!] - \varphi_P[\![-F_w, 0]\!]\end{cases},$$

带入方程（6）则有：

$$(\varphi_e[\![F_e,0]\!] - \varphi_E[\![-F_e,0]\!]) - (\varphi_W[\![F_w,0]\!] - \varphi_W[\![-F_w,0]\!]) = \Gamma_{\varphi,e}\frac{\varphi_E - \varphi_P}{(\delta x)_e} - \Gamma_{\varphi,w}\frac{\varphi_p - \varphi_w}{(\delta x)_w}, \quad (8)$$

则,

$$a_P\varphi_P = a_W\varphi_W + a_E\varphi_E$$

$$a_E = D_e + [\![-F_e, 0]\!],\ a_W = D_w + [\![F_w, 0]\!],\ a_P = a_E + a_W + (F_e - F_w)$$

对于边界处的界面上φ值：

由边界条件可得对于第一个控制体有：

$$a_P\varphi_P = a_W\varphi_W + a_E\varphi_E$$

$a_E = D_e + [\![- F_e, 0]\!]$, $a_W\varphi_W = (2D_w + [\![F_w, 0]\!])\varphi_0$,

$a_P = [\![F_e, 0]\!] + [\![- F_w, 0]\!] + D_e + 2D_w$

对于最后一个控制体有：

$a_P\varphi_P = a_W\varphi_W + a_E\varphi_E$

$a_W = D_w + [\![F_w, 0]\!]$, $a_E\varphi_E = (2D_e + [\![- F_e, 0]\!])\varphi_L$,

$a_P = [\![F_e, 0]\!] + [\![- F_w, 0]\!] + 2D_e + 2D_w$

1.3 混合格式

控制体界面处的φ值：

$$\varphi_e = \begin{cases} \varphi_P, \; P_e > 2 \\ (\varphi_P + \varphi_E)/2, \; |P_e| \le 2 \\ \varphi_E, \; P_e < -2 \end{cases}$$

当$|P_e| \le 2$时，其形式与中心差分相同；当$|P_e| > 2$时，其形式与上风格式相同，整理可得，

$a_P\varphi_P = a_W\varphi_W + a_E\varphi_E$

$a_E = \left[\!\left[- F_e, D_e - \frac{F_e}{2}, 0 \right]\!\right]$, $a_W = \left[\!\left[F_w, D_w + \frac{F_w}{2}, 0 \right]\!\right]$, $a_P = a_E + a_W + (F_e - F_w)$

计算结果

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

表 1 不同格式计算结果

	φ_1	φ_2	φ_3	φ_4	φ_5
精 确 解	0.938793	0.796390	0.622459	0.410020	0.150545
中心差分	0.9390146	0.7967154	0.6227942	0.4102237	0.1504154
相对误差（%）	2.3612201 E-02	4.0819650 E-02	5.3796108 E-02	4.9803854 E-02	8.5985282 E-02
上风格式	0.9337334	0.7879469	0.6130031	0.4030705	0.1511514
相对误差（%）	5.3894101 E-01	1.0602032	1.52	1.695	0.403
混合格式	0.9390146	0.7967154	0.6227942	0.4102237	0.1504154

相对误差（%）	2.3612201 E-02	4.0819650 E-02	5.3796108 E-02	4.9803854 E-02	8.5985282 E-02

u=2.5 m/s

表 2 不同格式计算结果

	φ_1	φ_2	φ_3	φ_4	φ_5
精 确 解	1.000000	1.000000	0.999996	0.999447	0.917915
中心差分	1.004167	0.9916667	1.020833	0.9527779	1.111574
相对误差（%）	0.4	0.83	2.08	4.67	21.10
上风格式	0.9998425	0.9987401	0.9921260	0.9524110	0.7143308
相对误差（%）	0.0157	0.125	0.787	4.7032	22.18
混合格式	1.000000	1.000000	1.000000	1.000000	1.000000
相对误差（%）	0	0	3.7551067E-04	5.5337755E-02	8.9425504

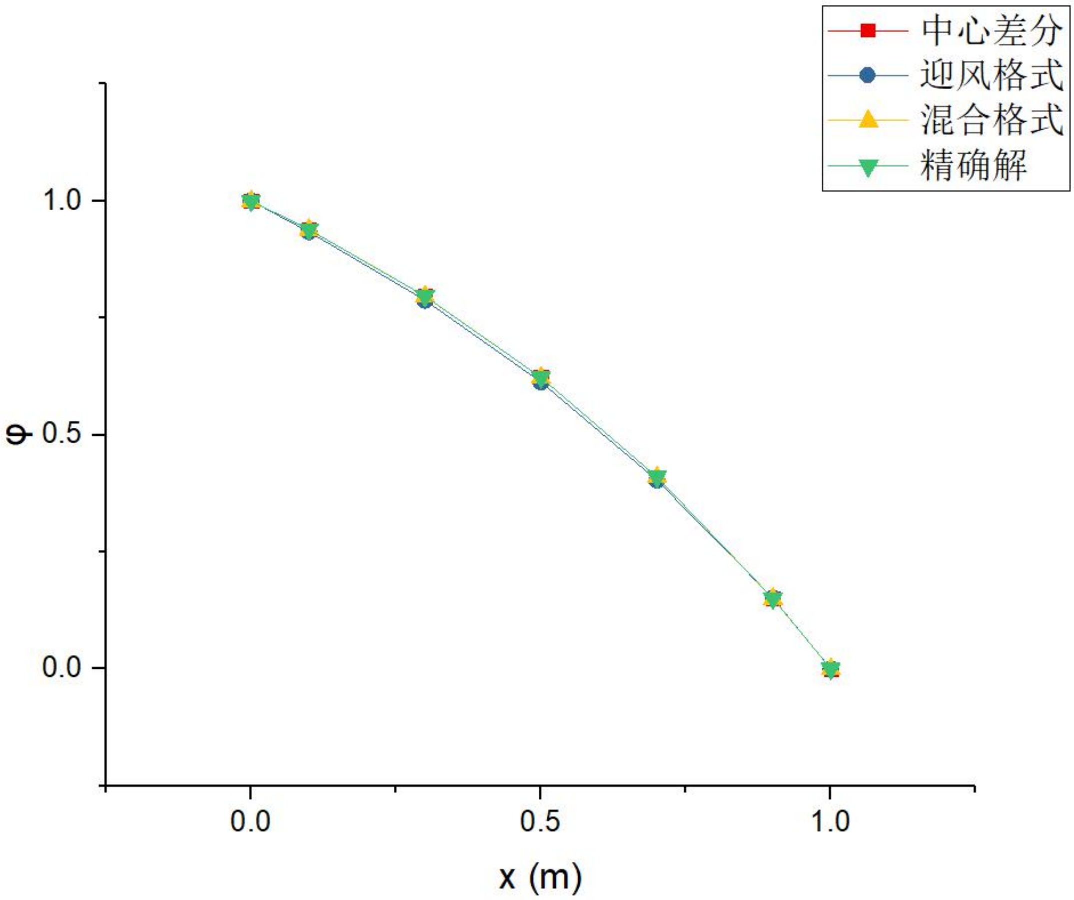


图 3 u=0.1m/s

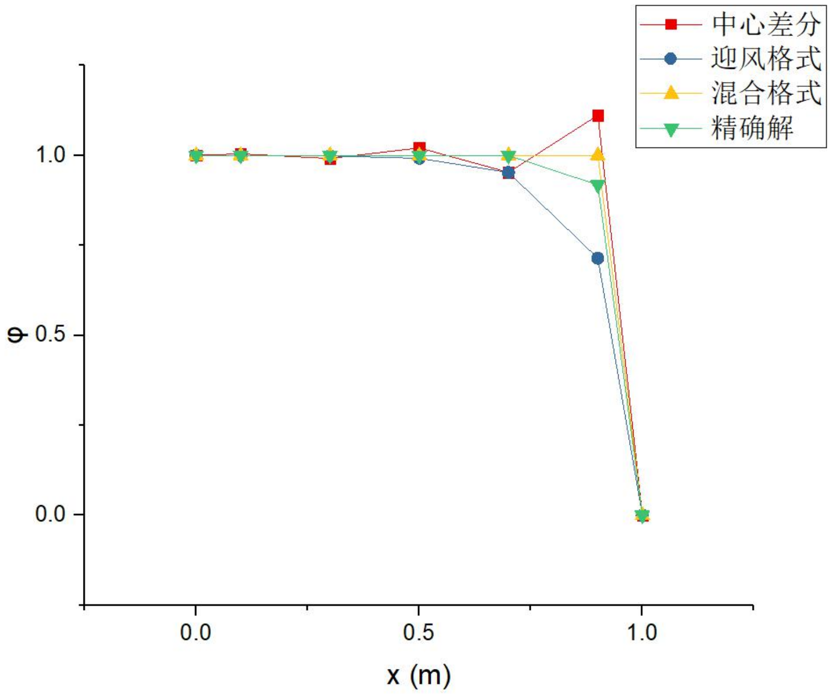


图 4 u=2.5m/s

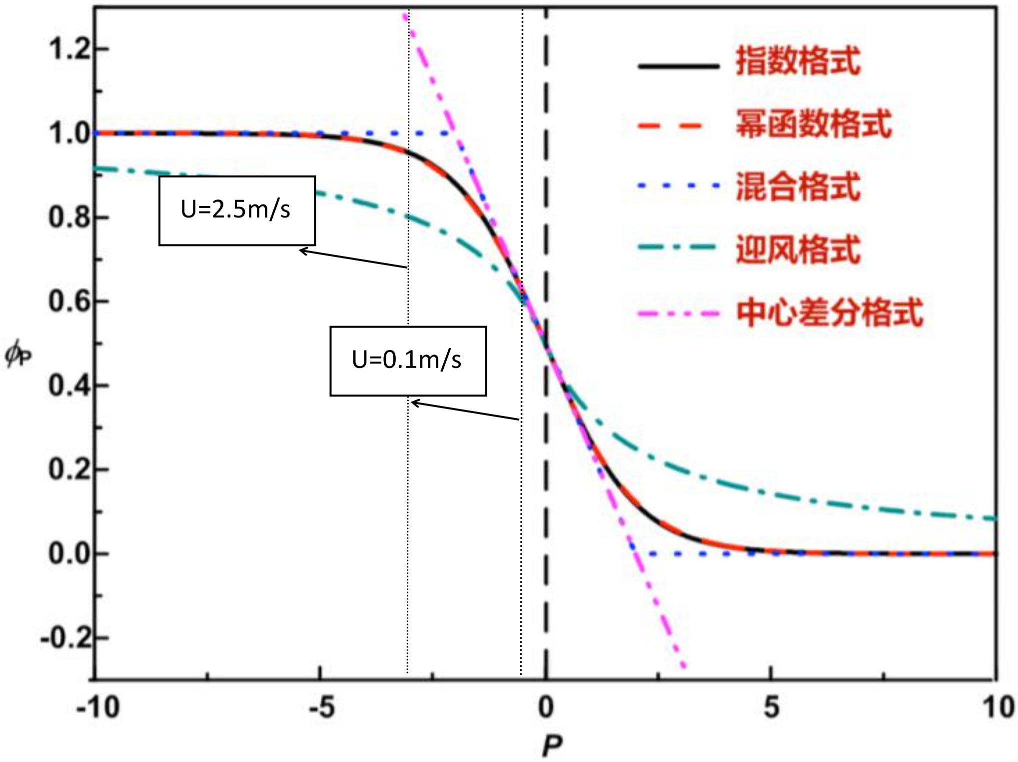


图 5Peclet 数对计算结果的影响

结果分析

本图是课堂上讨论的不同离散格式下的不同 Peclet 下的精确度，可知 u=0.1m/s、u=2.5m/s 情况分别等效于如图所示的两个位置，结合计算结果可知，相吻合，u=2.5m/s 时，三种格式的计算结果均与精确解有较大偏差，且中心差

分格式偏大，混合格式略偏大，迎风格式偏小，图 4 中的结果符合预期结果。

为进一步分析计算结果,速度取 u=2m/s、u=5m/s，可看出 u=1m/s 时 P=2，此时混合格式与中心差分格式相同，此时混合格式与准确解的偏差达到最大，这是因为此时混合格式仍然与中心格式是等同的，u=5m/s 时，Pelclet 数较大，此时，迎风格式优于中心差分格式，后者不稳定性加剧，而混合格式已经接近精确解，这是因为混合格式忽略了扩散项，而迎风格式的扩散项假设节点之间温度呈线性分布，该假设造成了很大的偏差。

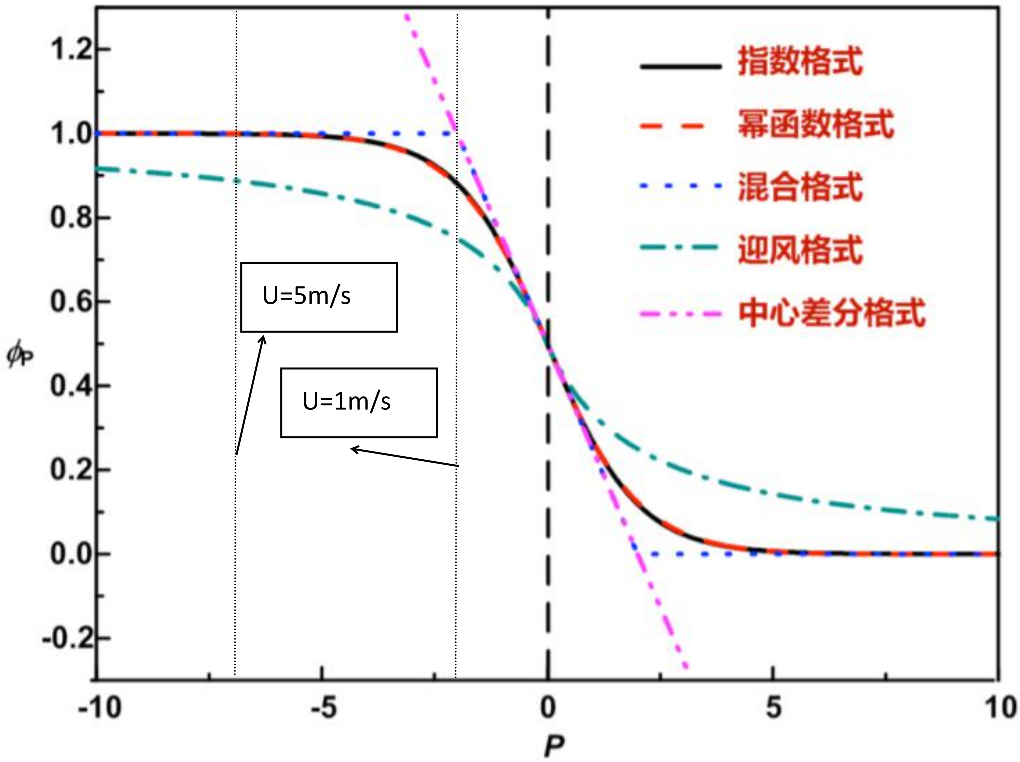


图 6 探究 u=1m/s 与 u=5m/s

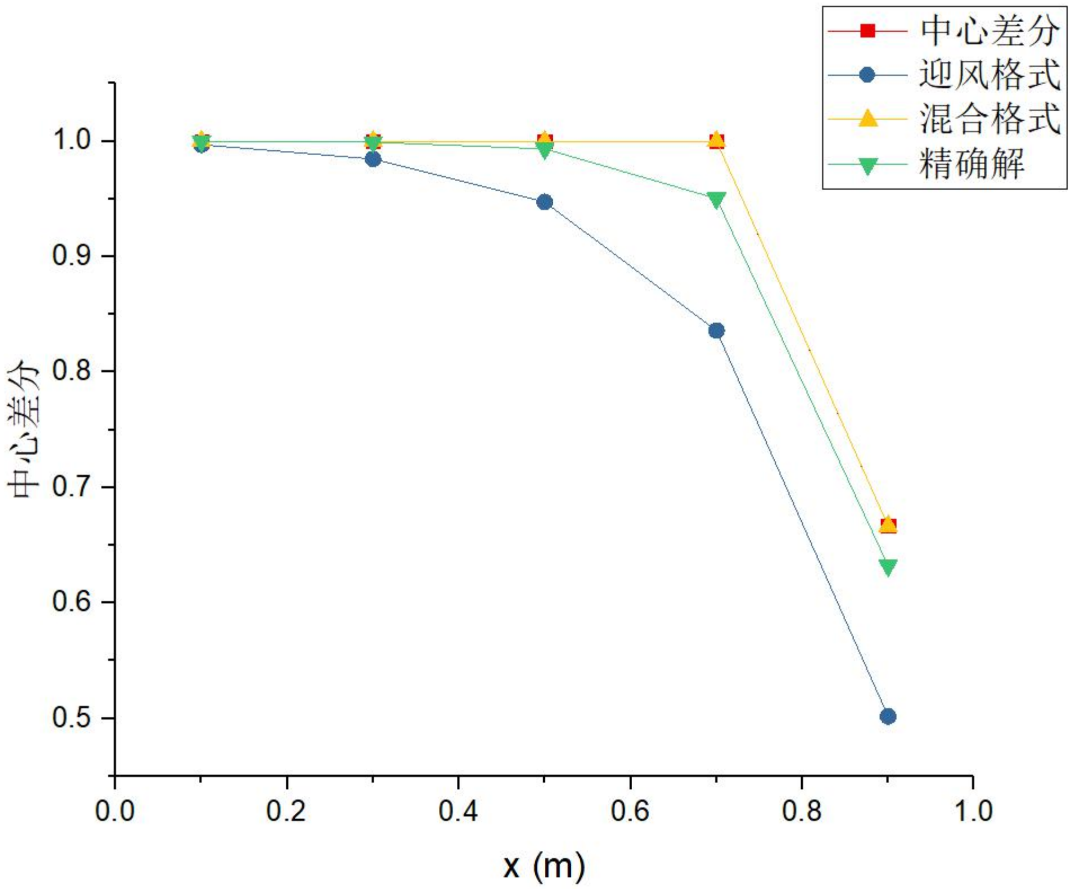


图 7 u=1m/s

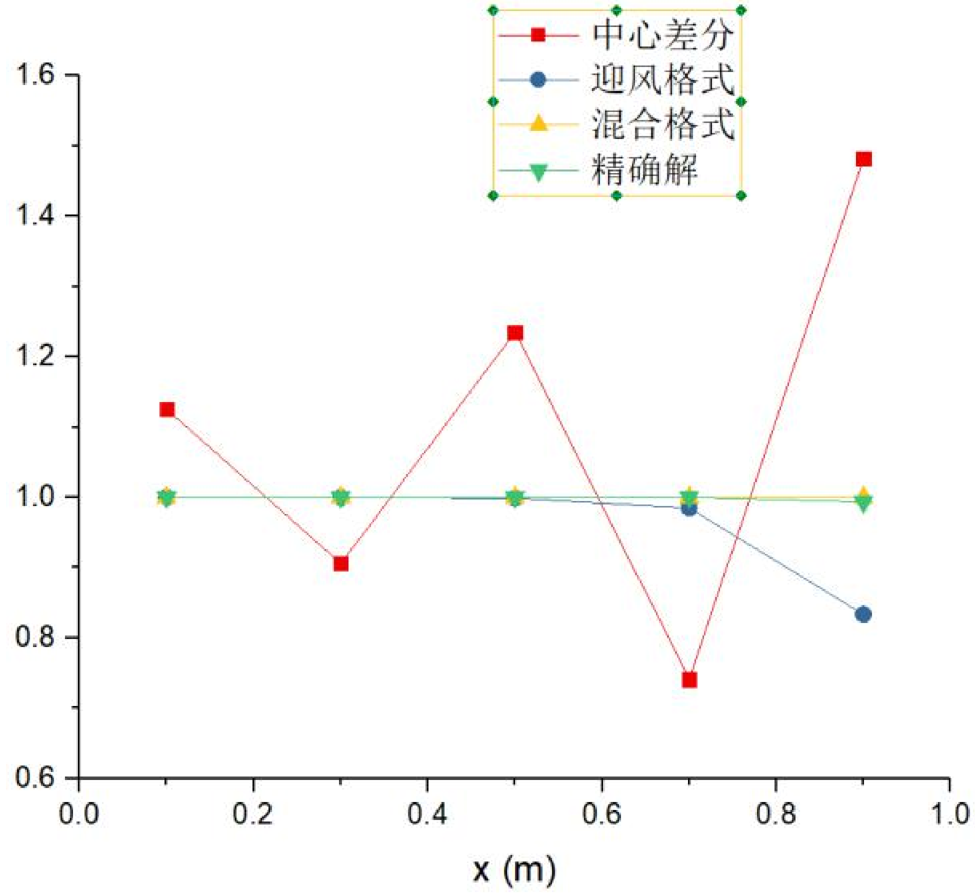
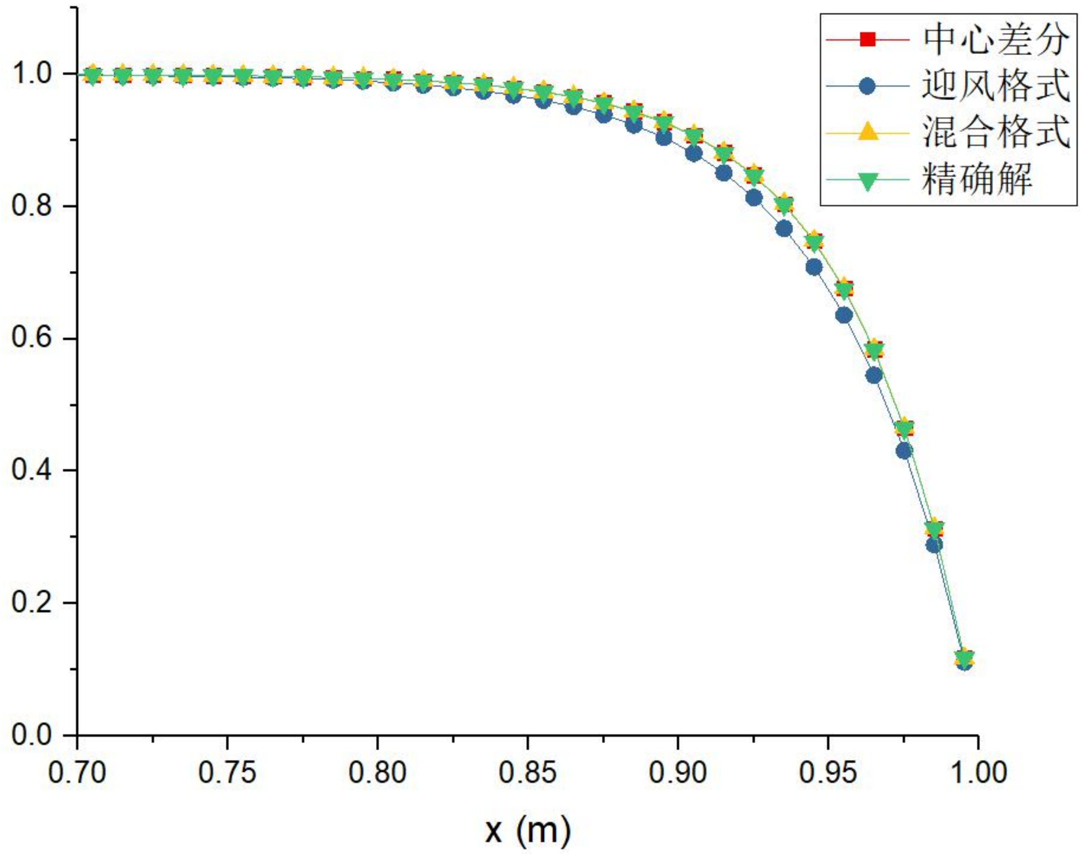


图 8 u=5m/s

还可以将网格细分为 100 个观察此时结果，取 u=2.5m/s，此时 Pelclet 数为 0.25，由图中可看出，迎风格式仍然存在较大偏差，可见，虽然迎风格式绝对稳定，但需要在 Pelclet 数很小或很大时才能接近解析解。



附件：程序代码

1 追赶法解线性方程

```
MODULE share_data
REAL::L,rou,lamda,u,fai0,faiL  !fai0,faiL 边界条件
save
INTEGER::K        !网格数
REAL::x           !网格间距
REAL::P,D,F       !P=F/D,F=rou*u,D=lamda/x
END MODULE                      !共享数据

REAL FUNCTION select_bigger(x,y)   !max()函数
IMPLICIT NONE
REAL,INTENT(IN)::x,y
IF(x>y) THEN
select_bigger=x
ELSE
select_bigger=y
END IF
END FUNCTION
```

```fortran
REAL FUNCTION A1(P)                !三种系数函数,中心差分格式
IMPLICIT NONE
REAL,INTENT(IN)::P
REAL::select_bigger
A1=1.0-0.5*ABS(P)+select_bigger(-P,0.0)
END FUNCTION A1

REAL FUNCTION A2(P)               !迎风格式
IMPLICIT NONE
REAL,INTENT(IN)::P
REAL::select_bigger
A2=1.0+select_bigger(-P,0.0)
END FUNCTION A2


REAL FUNCTION A3(P)               !混合格式
IMPLICIT NONE
REAL,INTENT(IN)::P
REAL::select_bigger
A3=select_bigger(0.0,1.0-0.5*ABS(P))+select_bigger(-P,0.0)
    END FUNCTION A3


SUBROUTINE pattern_chasing(fai,A)      !追赶法
USE share_data
IMPLICIT NONE
REAL,INTENT(OUT)::fai(K)
REAL,EXTERNAL::A
REAL::diag(K),right(K-1),left(K-1),b(K)
INTEGER::i
diag(1)=2.0*A(P/2.0)+A(P)+P
right(1)=-A(P)
b(1)=2.0*(P/2.0+A(P/2.0))*fai0
diag(K)=A(P)+2.0*A(P/2.0)+P
left(K-1)=-P-A(P)
b(K)=2.0*A(P/2.0)*failL
DO i=2,K-1
diag(i)=2*A(P)+P
right(i)=-A(P)
left(i-1)=-A(P)-P
b(i)=0
END DO
DO i=1,K-1
```

```
left(i)=left(i)/diag(i)
diag(i+1)=diag(i+1)-left(i)*right(i)
b(i+1)=b(i+1)-left(i)*b(i)
end do
fai(K)=b(K)/diag(K)
DO i=K-1,1,-1
fai(i)=(b(i)-right(i)*fai(i+1))/diag(i)
end do
END SUBROUTINE

PROGRAM Convection_diffusion
USE share_data
IMPLICIT NONE
REAL,EXTERNAL::A1,A2,A3
REAL::j
REAL,ALLOCATABLE,DIMENSION(:)::z
REAL,ALLOCATABLE,DIMENSION(:)::fai1,fai2,fai3,fai4      !1 中心差分，2 迎风格式，3 混合格式 4 解析解
REAL::time_begin,time_over
WRITE(*,*)'输入网格数'
READ(*,*) K
WRITE(*,*)'输入速度'
READ(*,*)u
CALL CPU_TIME(time_begin)
ALLOCATE(fai1(K))
ALLOCATE(fai2(K))
ALLOCATE(fai3(K))
ALLOCATE(fai4(K))
ALLOCATE(z(K))
L=1.0
rou=1.0
lamda=0.1
fai0=1.0
failL=0.0
x=L/K
F=rou*u
D=lamda/x
P=F/D
CALL pattern_chasing(fai1,A1)     !追赶法，求三种格式线性方程组
CALL pattern_chasing(fai2,A2)
CALL pattern_chasing(fai3,A3)
DO j=1,K
z(j)=(j-0.5)*x
fai4(j)=1.0-(exp(F*z(j)/lamda)-1.0)/(exp(F*L/lamda)-1.0)
```

```
end DO
CALL CPU_TIME(time_over)
WRITE(*,*)'    坐标     ','    中心差分    ','  迎风格式    ','    混合格式    ','  解
析解    '
Do j=1,K
WRITE(*,*) z(j),fai1(j),fai2(j),fai3(j),fai4(j)
END DO
open(1,file='test.out',mode ='write')
WRITE(1,*)'    坐标     ','    中心差分    ','  迎风格式    ','    混合格式    ','  解
析解    '
Do j=1,K
WRITE(1,*) z(j),fai1(j),fai2(j),fai3(j),fai4(j)
END DO
Do j=1,K
WRITE(1,*)
(fai1(j)-fai4(j))/fai4(j),(fai2(j)-fai4(j))/fai4(j),(fai3(j)-fai4(j))/fai4(j)
END DO
close(1)
WRITE(*,*)'运行时间',time_over-time_begin
DEALLOCATE(z)
DEALLOCATE(fai1)
DEALLOCATE(fai2)
DEALLOCATE(fai3)
DEALLOCATE(fai4)
END PROGRAM convection_diffusion
```

2. 高斯赛达尔方法解线性方程

```
MODULE share_data
REAL::L,rou,lamda,u,fai0,failL  !fai0,failL 边界条件
INTEGER::K   !网格数
REAL::x          !网格间距
REAL::P,D,F      !P=F/D,F=rou*u,D=lamda/x
REAL::w         !松弛因子
END MODULE                      !共享数据

REAL FUNCTION select_bigger(x,y)   !max()函数
IMPLICIT NONE
REAL,INTENT(IN)::x,y
IF(x>y) THEN
select_bigger=x
ELSE
select_bigger=y
END IF
END FUNCTION
```

```
REAL FUNCTION A1(P)               !三种系数函数,中心差分格式
IMPLICIT NONE
REAL,INTENT(IN)::P
REAL::select_bigger
A1=1.0-0.5*ABS(P)+select_bigger(-P,0.0)
END FUNCTION A1

REAL FUNCTION A2(P)              !迎风格式
IMPLICIT NONE
REAL,INTENT(IN)::P
REAL::select_bigger
A2=1.0+select_bigger(-P,0.0)
END FUNCTION A2


REAL FUNCTION A3(P)              !混合格式
IMPLICIT NONE
REAL,INTENT(IN)::P
REAL::select_bigger
A3=select_bigger(0.0,1.0-0.5*ABS(P))+select_bigger(-P,0.0)
    END FUNCTION A3

SUBROUTINE pattern_gauss_siedal(fai,A)          !求解线性方程组，fai 为输出量，A 为系数函
数 A（P）
USE share_data
IMPLICIT NONE
REAL,INTENT(OUT)::fai(K)
REAL,EXTERNAL::A
INTEGER::i,j
REAL::fai_former(K)
fai=0.0
fai_former=1.0
DO                                  !高斯—塞达尔迭代解法
fai(1)=(A(P)*fai(2)/(A(P)+P+2.0*A(P/2.0))+(P+2.0*A(P/2.0))*fai0/(A(P)+P+2.0*A(P/2.0)))*
w+fai(1)*(1-w)
DO i=2,K-1
fai(i)=((A(P)+P)*fai(i-1)/(2.0*A(P)+P)+A(P)*fai(i+1)/(2.0*A(P)+P))*w+fai(i)*(1-w)
END DO
fai(K)=((A(P)+P)*fai(K-1)/(A(P)+P+2.0*A(P/2.0))+2*A(P/2.0)*failL/(A(P)+P+2.0*A(P/2.0)))*
w+fai(K)*(1-w)
IF (maxval(abs(fai-fai_former))<0.000001) exit              !收敛条件
fai_former=fai
END DO
```

```
END SUBROUTINE pattern_gauss_siedal


PROGRAM Convection_diffusion
USE share_data
IMPLICIT NONE
REAL,EXTERNAL::A1,A2,A3
INTEGER::j
REAL,ALLOCATABLE,DIMENSION(:)::z
REAL,ALLOCATABLE,DIMENSION(:)::fai1,fai2,fai3,fai4      !1 中心差分，2 迎风格式，3 混合格式 4 解析解
REAL::time_begin,time_over
WRITE(*,*)'输入网格数'
READ(*,*) K
WRITE(*,*)'输入速度'
READ(*,*) u
WRITE(*,*)'松弛因子'
READ(*,*) w
Call CPU_TIME(time_begin)
ALLOCATE(z(K))
ALLOCATE(fai1(K))
ALLOCATE(fai2(K))
ALLOCATE(fai3(K))
ALLOCATE(fai4(K))
L=1.0
rou=1.0
lamda=0.1
fai0=1.0
faiL=0.0
x=L/K
F=rou*u
D=lamda/x
P=F/D
CALL pattern_gauss_siedal(fai1,A1)     !高斯塞达尔法，求三种格式线性方程组
CALL pattern_gauss_siedal(fai2,A2)
CALL pattern_gauss_siedal(fai3,A3)
DO j=1,K
z(j)=(j-0.5)*x
fai4(j)=1.0-(exp(F*z(j)/lamda)-1.0)/(exp(F*L/lamda)-1.0)
end DO
CALL CPU_TIME(time_over)
Do j=1,K
WRITE(*,*) z(j),fai1(j),fai2(j),fai3(j),fai4(j)
END DO
```

```
WRITE(*,*) '运行时间',time_over-time_begin
open(1,file='test.out',mode ='write')
Do j=1,K
WRITE(1,*) z(j),fai1(j),fai2(j),fai3(j),fai4(j),(fai1(j)-fai4(j))/fai4(j)
END DO
close(1)
DEALLOCATE(z)
DEALLOCATE(fai1)
DEALLOCATE(fai2)
DEALLOCATE(fai3)
DEALLOCATE(fai4)
END PROGRAM convection_diffusion
```