

# 数值传热学

## 第六章 求解椭圆型流动与换热的原始变量法 (1)



**主讲 陶文铨**

西安交通大学能源与动力工程学院  
热流科学与工程教育部重点实验室  
2016年10月26日, 西安

# Numerical Heat Transfer

## (数值传热学)

### Chapter 6 Primitive Variable Methods for Elliptic Flow and Heat Transfer (1)



**Instructor Tao, Wen-Quan**

**CFD-NHT-EHT Center**  
**Key Laboratory of Thermo-Fluid Science & Engineering**  
**Xi'an Jiaotong University**  
**Xi'an, 2016-Oct.-26**

## Chapter 6 Primitive Variable Methods for Elliptic Flow and Heat Transfer

- 6.1 Source terms in momentum equations and two key issues in numerically solving momentum equation
- 6.2 Staggered grid system and discretization of momentum equation
- 6.3 Pressure correction methods for N-S equation
- 6.4 Approximations in SIMPLE algorithm
- 6.5 Discussion on SIMPLE algorithm and criteria for convergence
- 6.6 Developments of SIMPLE algorithm
- 6.7 Methods for accelerating convergence of pressure - correction methods
- 6.8 Boundary condition treatments for open system
- 6.9 Fluid flow & heat transfer in a closed system
- 6.10 SIMPLE-series algorithm at collocated grid system

## **6.1 Source terms in momentum equations and two key issues in numerically solving momentum equation**

### **6.1.1 Introduction**

### **6.1.2 Source in momentum equations**

### **6.1.3 Two key issues in solving flow field**

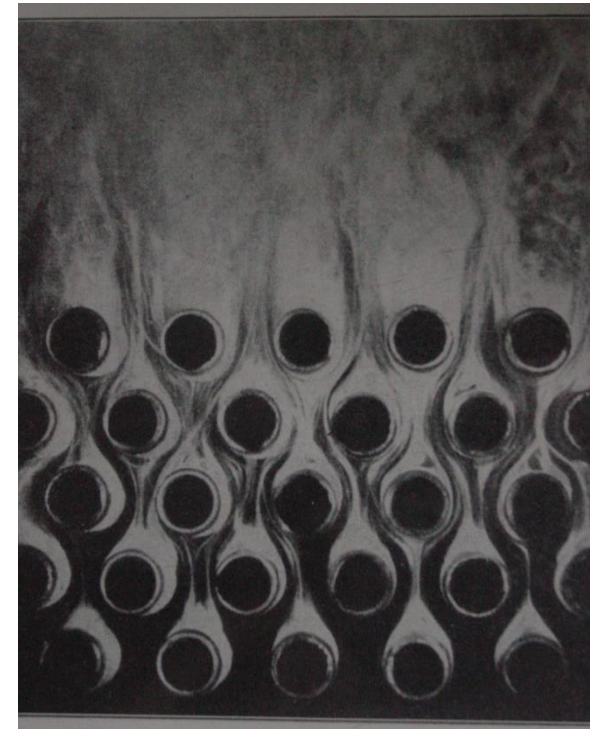
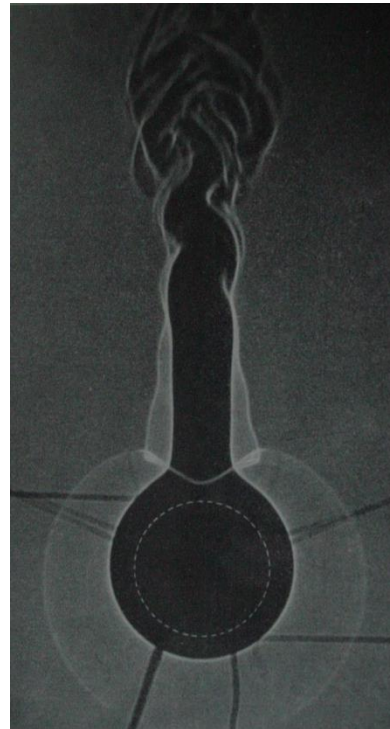
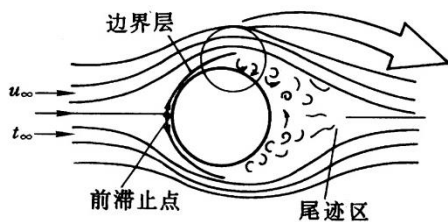
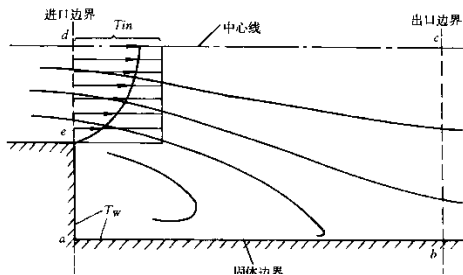
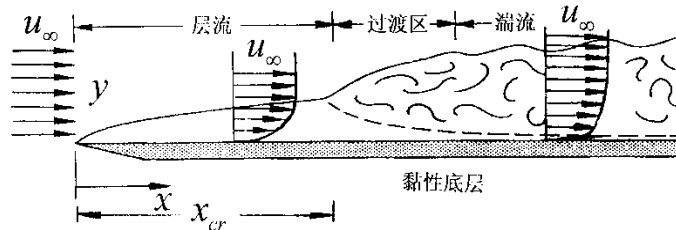
**1. The conventional methods may lead to oscillating pressure field**

**2. Pressure has no governing equation-To improve an assumed pressure field a specially designed algorithm is needed**

# 6.1 Source terms in momentum equations and two key issues in numerically solving momentum equation

## 6.1.1 Introduction

1 . Two kinds of most often encountered engineering flows: boundary layer type and recirculation type



**2. Flow field solution is the most important step for convective heat transfer simulation**

**3. Numerical approaches for solution of incompressible flow field**

Simultaneously solutions of different dependent variables ( $u, v, w, p, T$ )..

Segregated solutions (分离式求解) of different dependent variables

In such approaches no special algorithm is needed. The only requirement is **an extremely large computer resource**

Primitive variable method (原始变量法,  $u, v, w, p$ ), Pressure correction method is the most widely used one

Non-primitive variable Method. Vortex-stream function method is the most widely used one

## 6.1.2 Source terms in momentum equations

The general governing equation is:

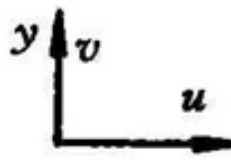
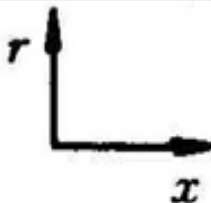

$$\frac{\partial(\rho\phi)}{\partial t} + \text{div}(\rho\vec{U}\phi) = \text{div}(\Gamma_{\phi}\text{grad}\phi) + S_{\phi}$$

Comparing N-S equations in the three coordinates with the above general governing equation, the related source terms can be obtained, where both physical source term (such as gravitation) and numerical source term are included;

Treatment of source term is very important in numerical simulation of momentum equations.

### Table 6 Source terms of 2-D incompressible flow

( $\eta = \text{const.}$  No gravitation)

Coordinates	u-equation	v-equation
Cartesian 	0	0
Axi-symmetric cylindrical 	0	$-\frac{\eta v}{r^2}$
Polar 	$-\frac{\rho uv}{r} + \frac{2\eta}{r^2} \frac{\partial v}{\partial \theta} - \frac{\eta u}{r^2}$	$\frac{\rho u^2}{r} - \frac{2\eta}{r^2} \frac{\partial u}{\partial \theta} - \frac{\eta v}{r^2}$



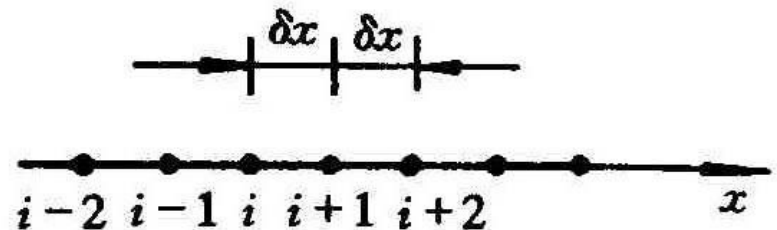
## 6.1.3 Two key issues in solving flow field

1. Conventional discretization method for pressure gradient in momentum equation may lead to oscillating pressure field.

Conventionally, one grid system is used to store all information: pressure, velocity, temperature,..... The discretized momentum equations can not detect unreasonable pressure field.

At node  $i$  the 1-D steady momentum equation

$$\rho u \frac{du}{dx} = -\frac{dp}{dx} + \eta \frac{d^2u}{dx^2}$$



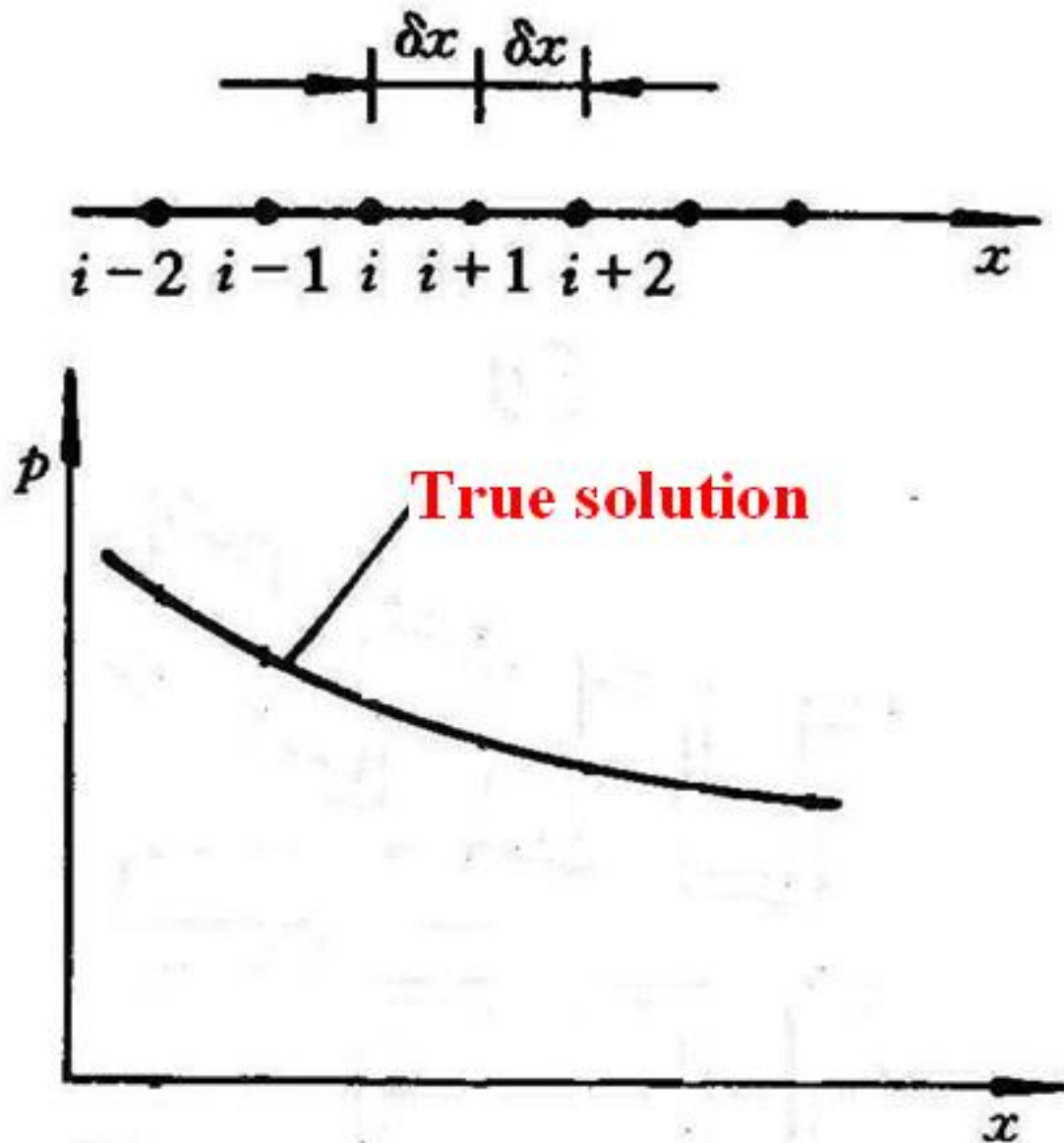
can be discretized by FDM as follows:

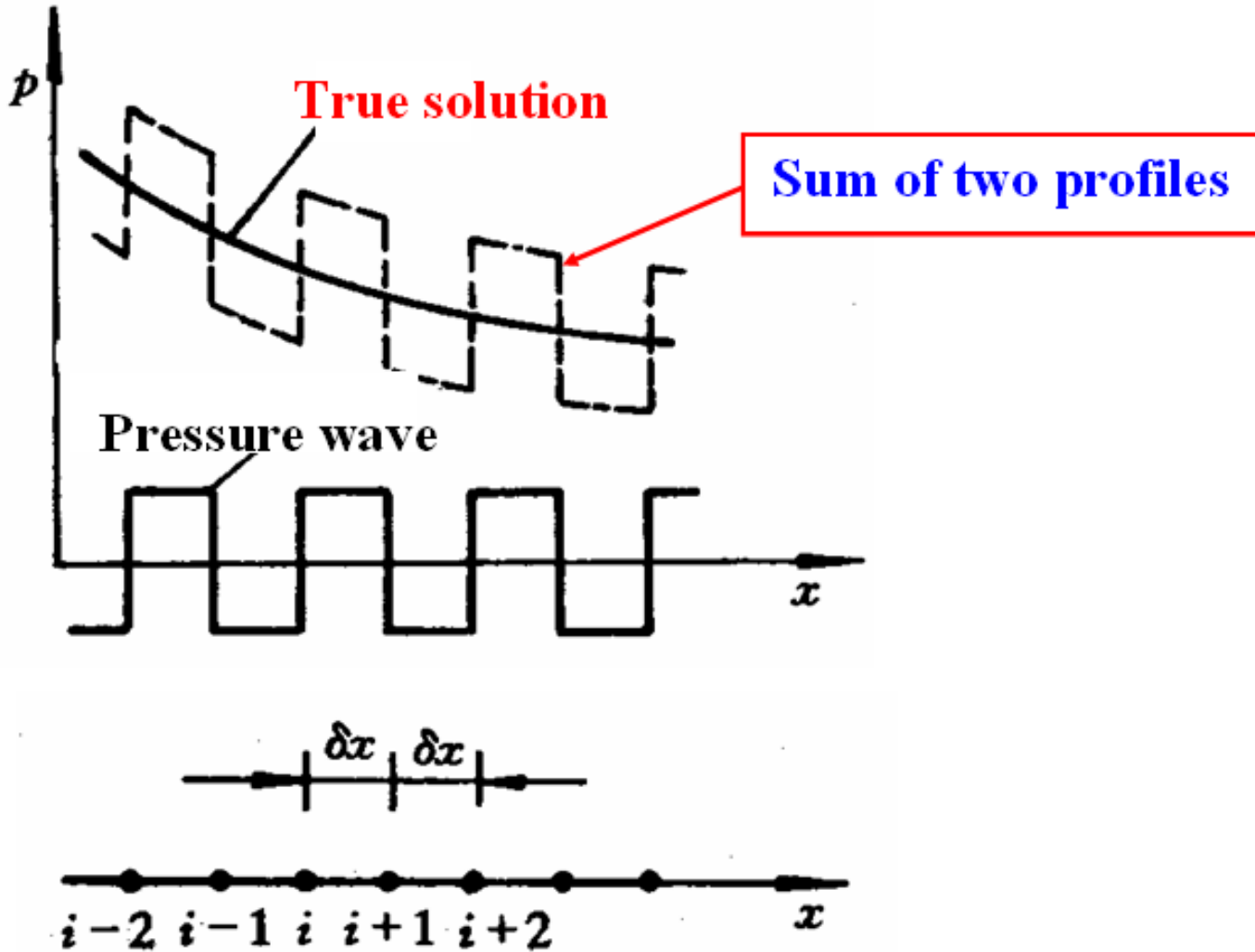
$$\rho u_i \frac{u_{i+1} - u_{i-1}}{2\delta x} = - \frac{p_{i+1} - p_{i-1}}{2\delta x} + \eta \frac{u_{i+1} - 2u_i + u_{i-1}}{(\delta x)^2}$$

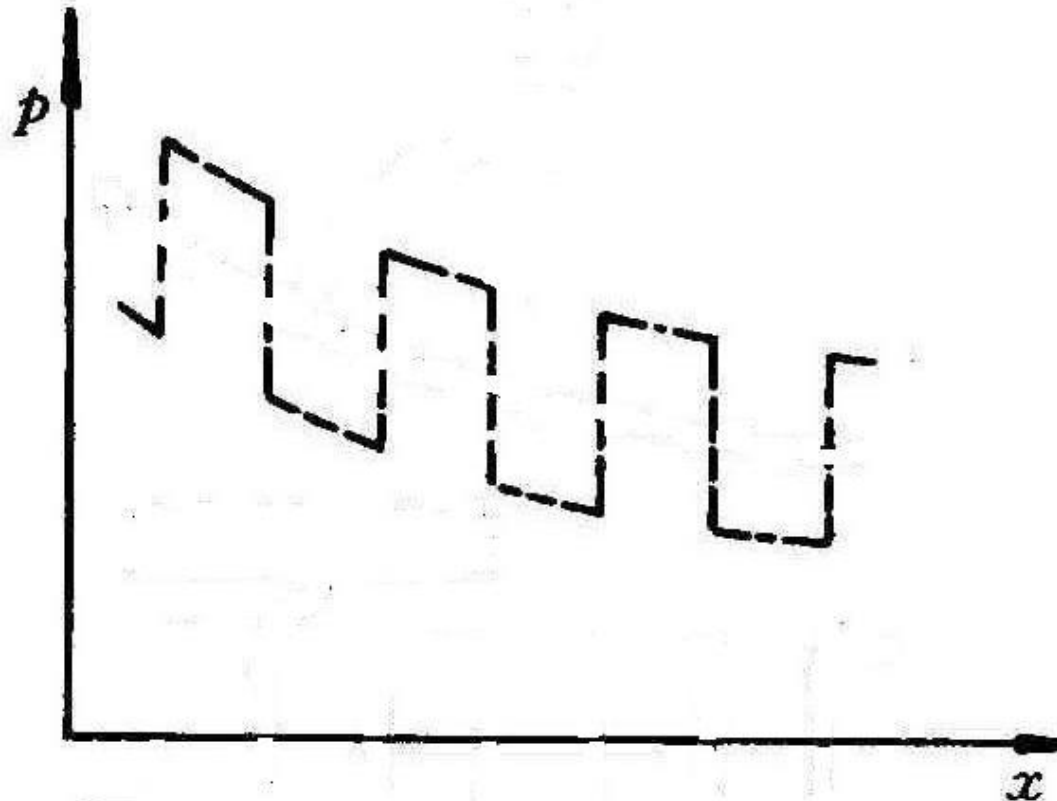
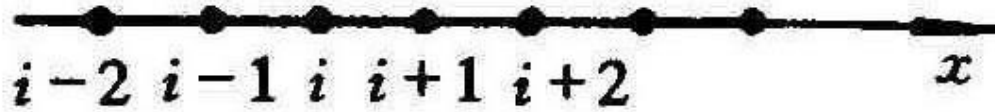
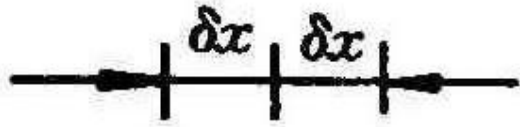
**CD**                      **CD**                      **CD**

**Discussion:** this is the discretized momentum equation for node  $i$ , it does not contain the pressure at node  $i$ , while includes the pressure difference between two nodes positioned two-steps apart, leading to following result: the discretized momentum equation can not detect an unreasonable pressure solution! Because it is the pressure gradient rather than pressure itself that occurs in the momentum equation.

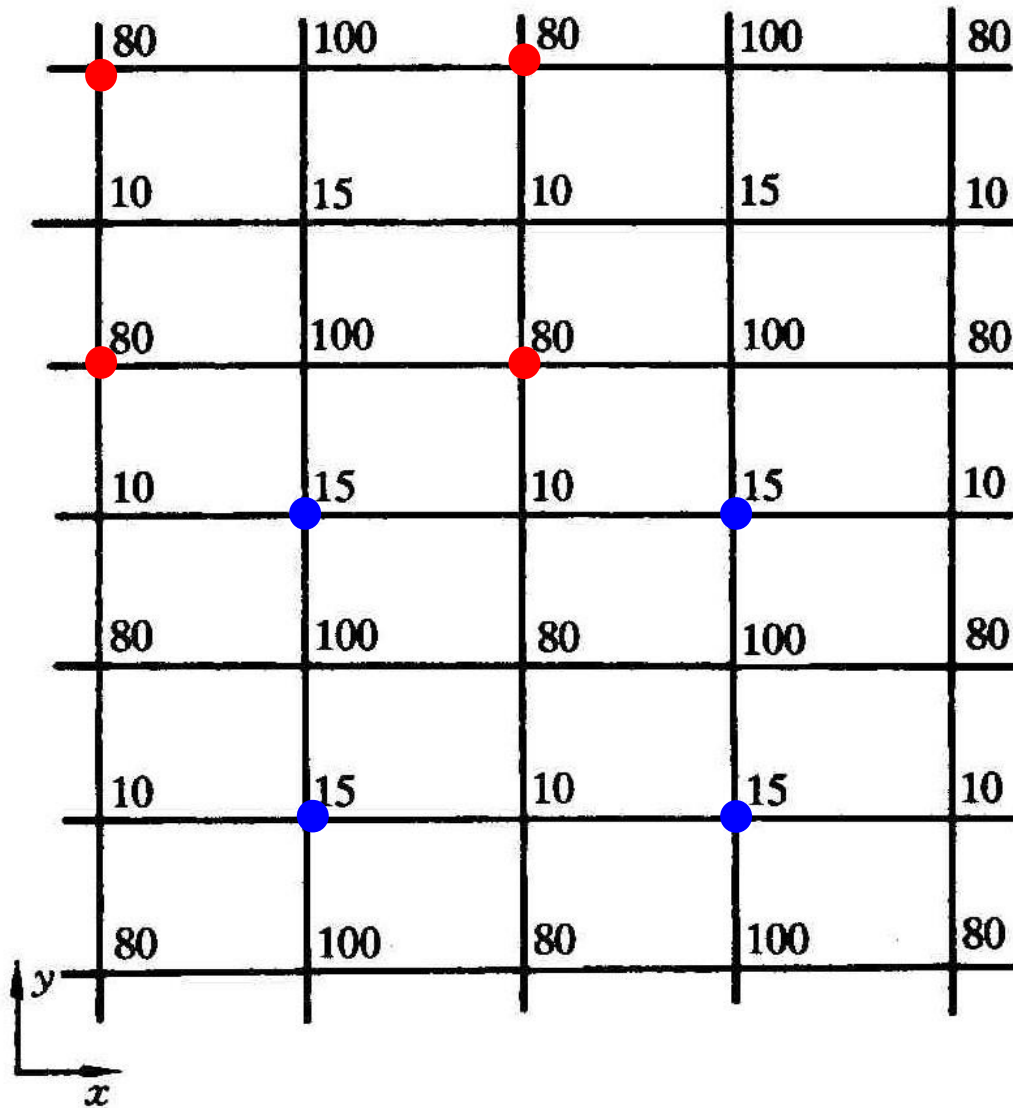
Pressure difference over two steps is called  $2 - \delta x$  pressure difference.







**Momentum equations based on this wave pressure field are the same as the true pressure field. Because only pressure difference over  $2 - \delta$ , is included in the discretized equations. For the true and false pressure profiles this difference is the same.**



The pressure difference over  $2\text{-}\Delta$  of the true field is the same as that of the superimposed (叠加的) pressure field!

2-D checkerboard pressure field (二维波形压力场)

**2. In the momentum equation, pressure gradient is the source term. It does not have its own governing equation. To improve an assumed pressure field, a specially-design algorithm(算法) has to be proposed.**

**At the beginning of iteration of the momentum equations, pressure field can be assumed. With the proceeding of iteration the assumed pressure field has to be improved. How to improve it ? Because pressure does not has its own governing equation, a special algorithm should be designed.**

**The first issue is to overcome the checkerboard by introducing staggered grid system(交叉网格), and the second one is by pressure correction algorithm.**

## 6.2 Staggered grid system and discretization of momentum equation

### 6.2.1 Staggered grid(交叉网格)

### 6.2.2 Discretization of momentum equation in staggered grid

### 6.2.3 Interpolation in staggered grid

1. Flow rate at a node

2. Density at interface

3. Conductance at interface

### 6.2.4 Remarks



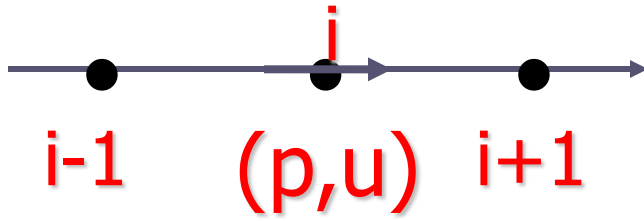
## 6.2 Staggered grid system and discretization of momentum equation

### 6.2.1 Staggered grid

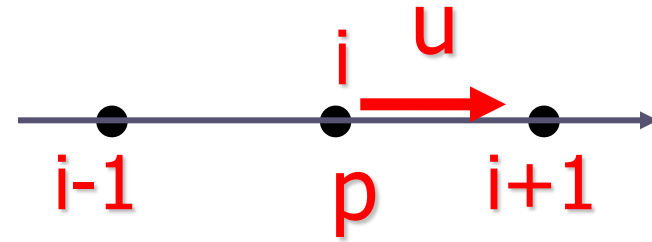
#### 1. Basic consideration

In the discretized momentum equation  $1-\delta$  pressure difference should be used, rather than  $2-\delta$  pressure difference; in addition the discretized pressure gradient should be of 2<sup>nd</sup> order according to Pascal principle (帕斯卡原理) in fluid mechanics.

Such requirements :1-Delta pressure difference is of 2<sup>nd</sup> order accuracy of pressure gradient can be easily achieved by moving the velocity to the interface :

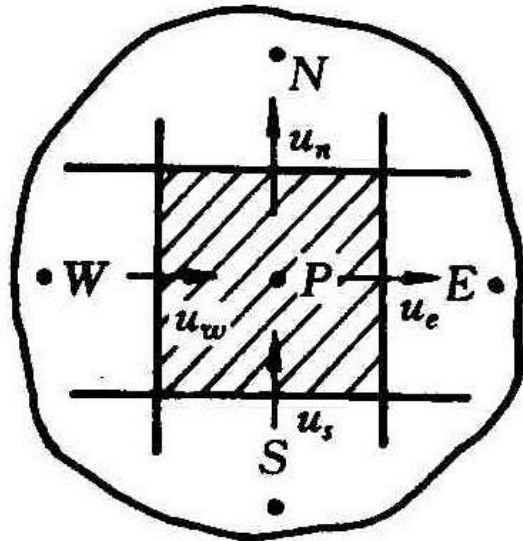


Conventional grid

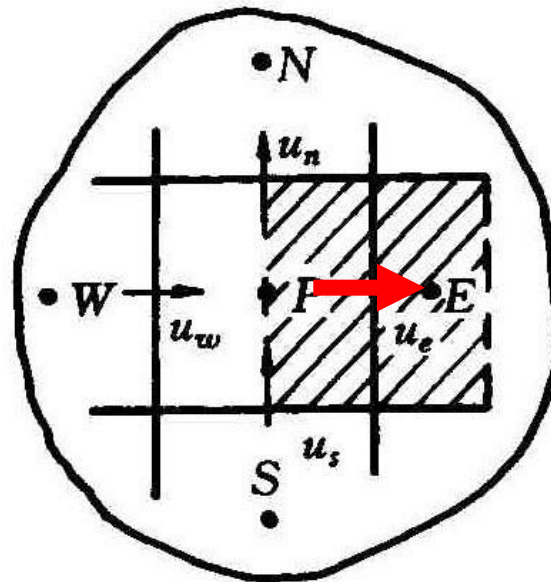


Staggered

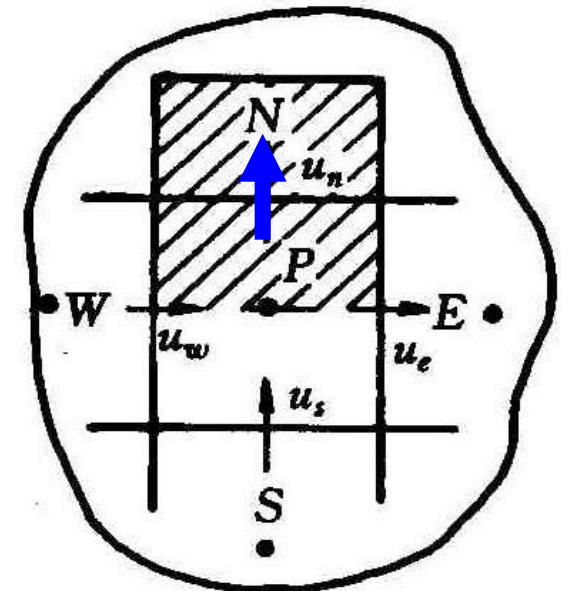
2. 2-D staggered grid



Main grid (Mass conservation)



u -grid



v -grid

## 6.2.2 Discretization of momentum equation

Discretization of other variables are the same as in the conventional grid. For velocities :

- 1)  $u, v$  –discretized based on own CV;
- 2) Pressure gradient is separated

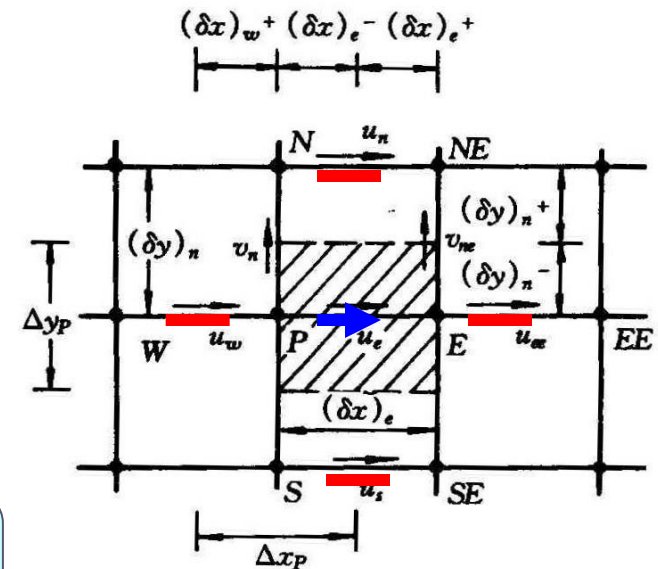
(分离) from source term:

$$\int_s^e \int_n^e -\frac{\partial p}{\partial x} dx dy = -\int_s^n (p)_P^E dy \cong$$

$$-(p_E - p_P)\Delta y = (p_P - p_E)\Delta y$$

**E-W boundary points for u-eq.: node P and E!**

$$a_e u_e = \sum a_{nb} u_{nb} + b + (p_P - p_E) A_e$$



**Neighboring points of  $u$**

## 6.2.3 Interpolations

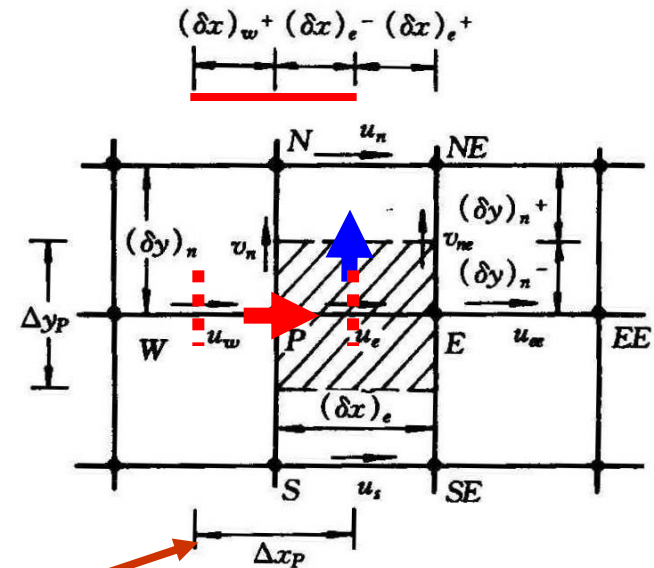
All thermal physical properties are stored at nodes, while velocities are at the interfaces. Interpolations are needed:

### 1. Flow rate at nodes ( $F_P$ , $F_{n-e}$ )

$$F_P = F_e \frac{(\delta x)_{w^+}}{\Delta x_P} + F_w \frac{(\delta x)_{e^-}}{\Delta x_P}$$

$$\Delta x_P = (\delta x)_{w^+} + (\delta x)_{e^-}$$

$$F_P = (\rho u \Delta y)_e \frac{(\delta x)_{w^+}}{\Delta x_P} + (\rho u \Delta y)_w \frac{(\delta x)_{e^-}}{\Delta x_P}$$

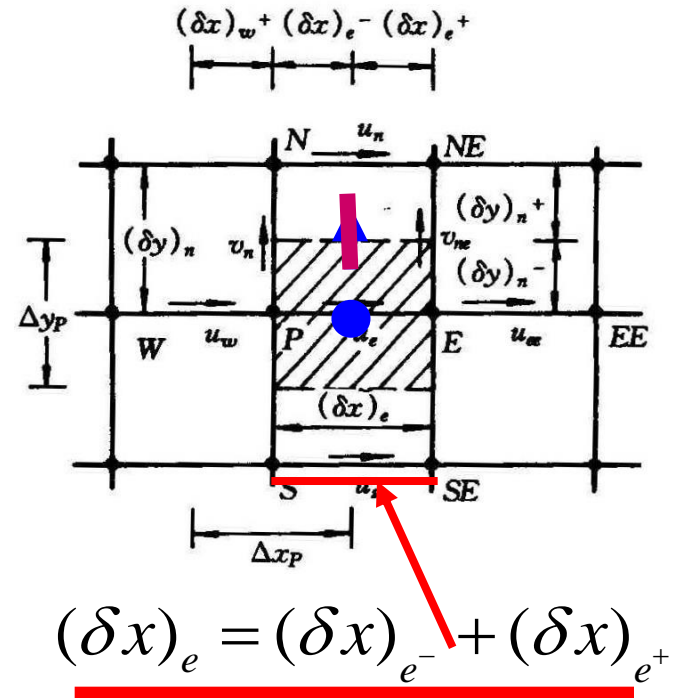


$$F_{n-e} = (\rho v)_n (\delta x)_{e^-} + (\rho v)_{ne} (\delta x)_{e^+}$$

## 2. Density at velocity position

$$\rho_e = \rho_E \frac{(\delta x)_{e^-}}{(\delta x)_e} + \rho_P \frac{(\delta x)_{e^+}}{(\delta x)_e}$$

$\frac{(\delta x)_{e^-}}{(\delta x)_e}, \frac{(\delta x)_{e^+}}{(\delta x)_e}$  interpolation functions for density



## 3. Conductance(扩导) at velocity interface

$$D_{n-e}$$

Parallel conductances

$$D_{n-e} = \frac{(\delta x)_{e^-}}{(\delta y)_n} + \frac{(\delta x)_{e^+}}{(\delta y)_n}$$

$$\Gamma_n \quad \Gamma_{ne}$$

$$D_{n-e} = \frac{(\delta x)_{e^-}}{(\delta y)_n} + \frac{(\delta x)_{e^+}}{(\delta y)_n} = \frac{(\delta x)_{e^-}}{(\delta y)_{n^-} + (\delta y)_{n^+}} + \frac{(\delta x)_{e^+}}{(\delta y)_n}$$

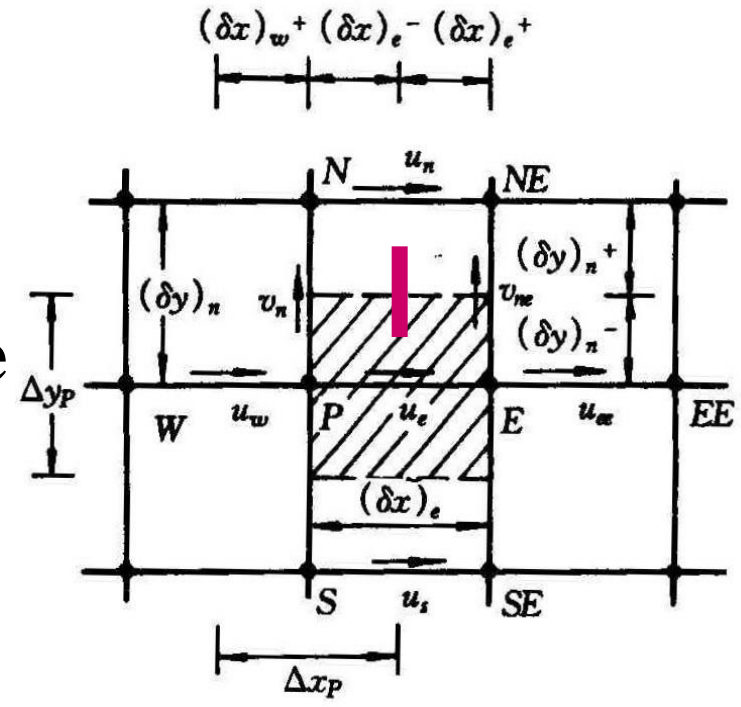
$\Gamma_n$                        $\Gamma_{ne}$                        $\Gamma_P$                        $\Gamma_N$                        $\Gamma_{ne}$

**Resistances in series(串联)**

$$= \frac{(\delta x)_{e^-}}{(\delta y)_{n^-} + (\delta y)_{n^+}} + \frac{(\delta x)_{e^+}}{(\delta y)_{n^-} + (\delta y)_{n^+}}$$

$\Gamma_P$                        $\Gamma_N$                        $\Gamma_E$                        $\Gamma_{NE}$

**Adopting the summation principle for resistances in series and the conductances in parallel (并联) to get  $D_{n-e}$  for the CV of  $u_e$ .**



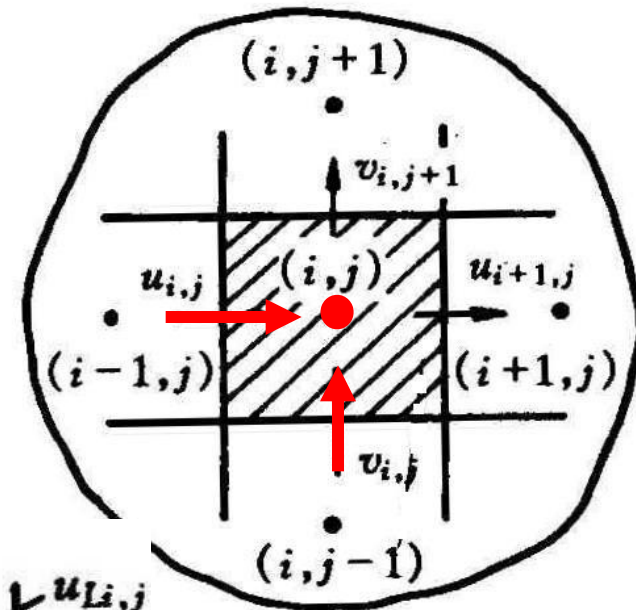
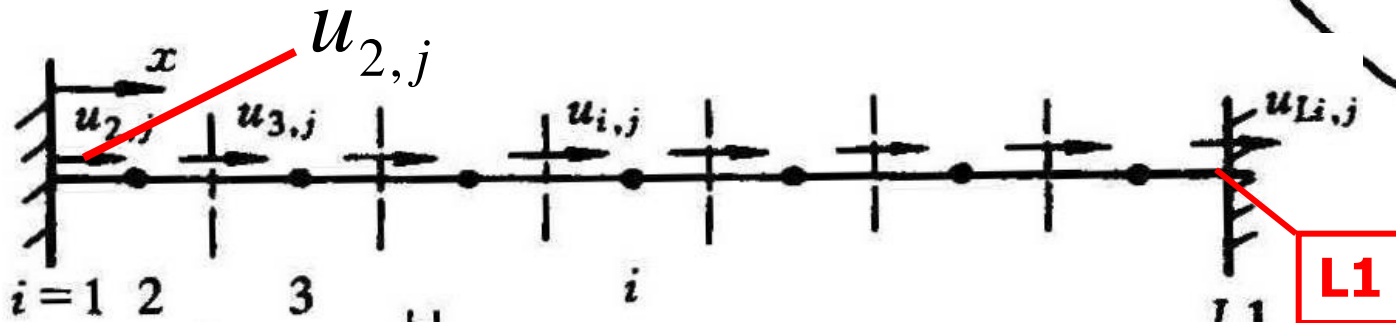
## 6.2.4 Remarks (注意事项)

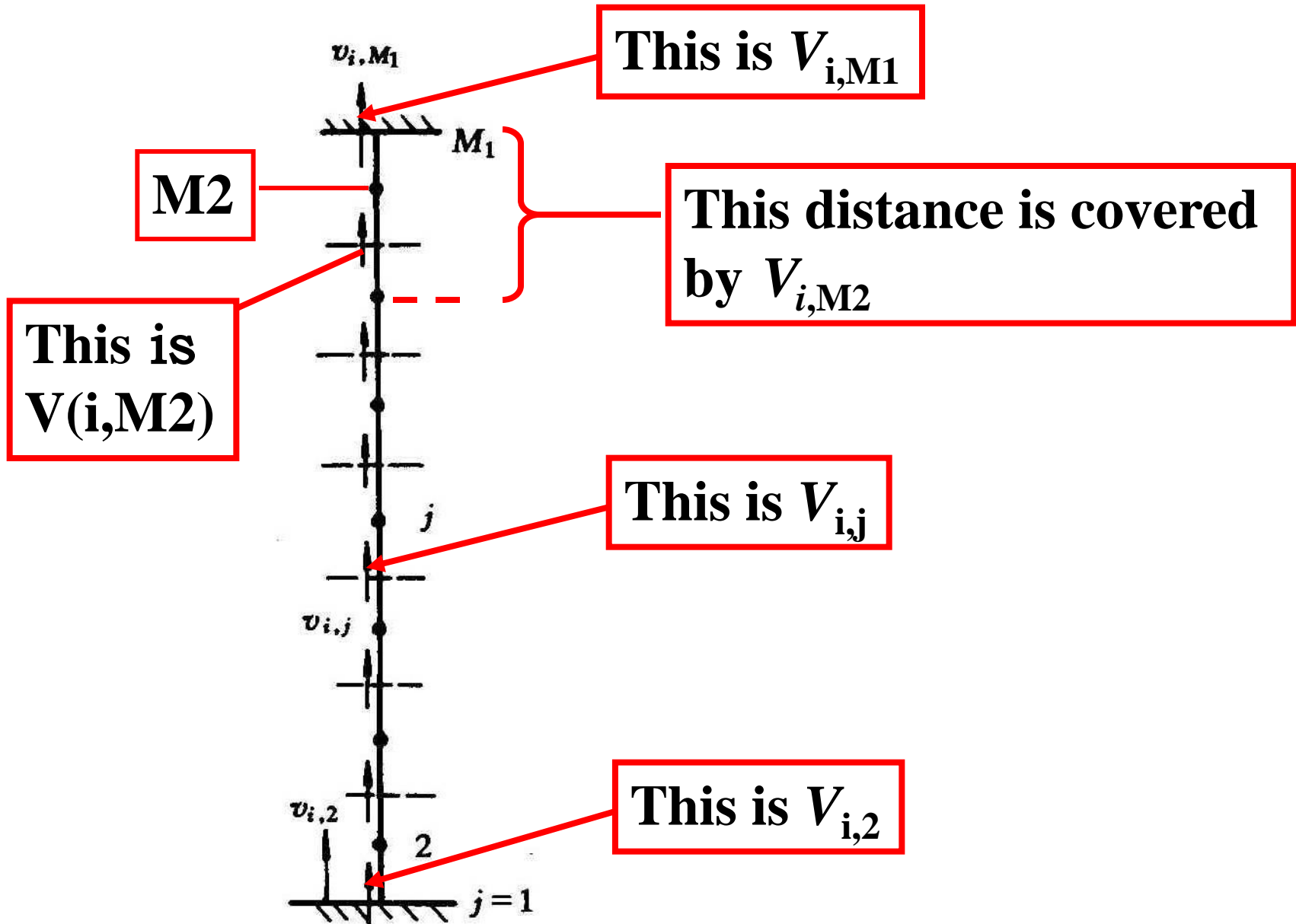
1. Three variables should be numbered consistently —  
The number of the node toward which the velocity arrow directs is the number of the velocity

Such numbering system(编号系统) has following consequences:

The starting value of E-W direction of  $u$  is  $i = 2$ ;

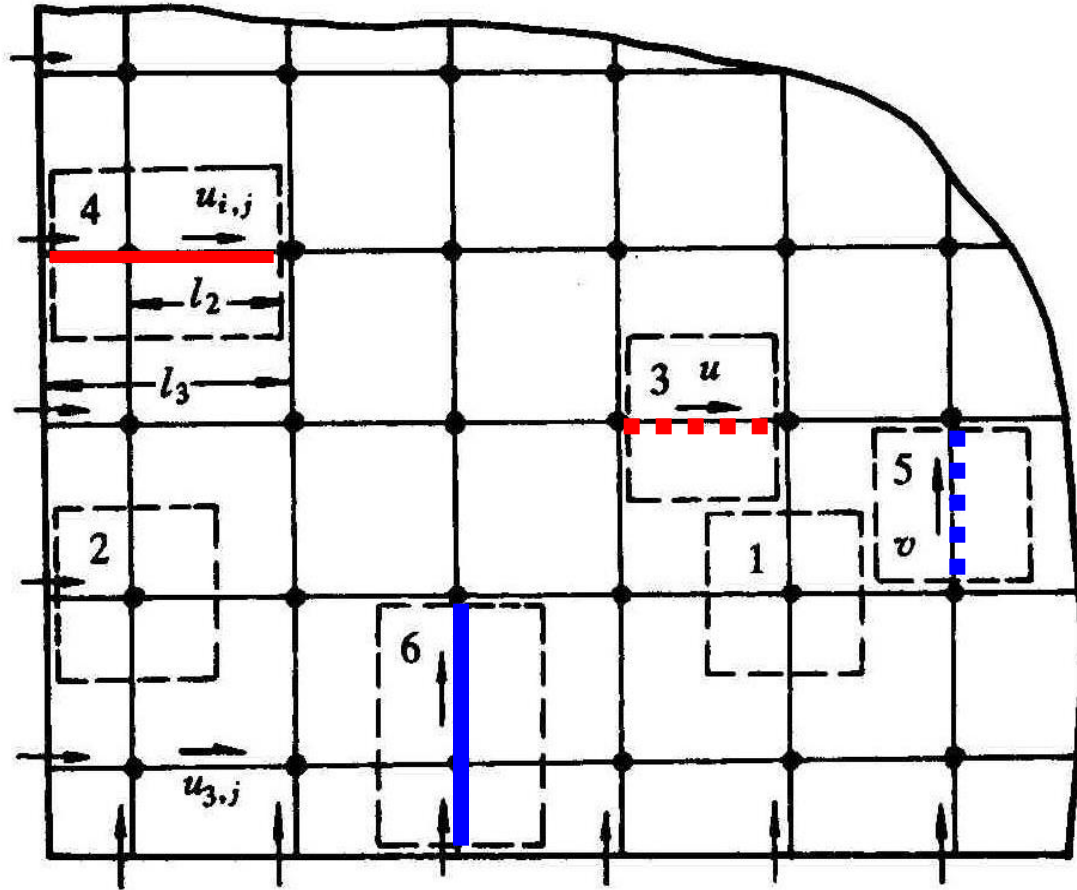
That for S-N direction of  $v$  is  $j = 2$ .







## 2. Velocity CV neighboring with boundary is different from inner ones in order to cover the whole domain



No	1	2
Type	$\phi$ (Inner)	$\phi$ (B.)

3	4	5	6
$u$ (In.)	$u$ (B.)	$v$ (In.)	$v$ (B.)

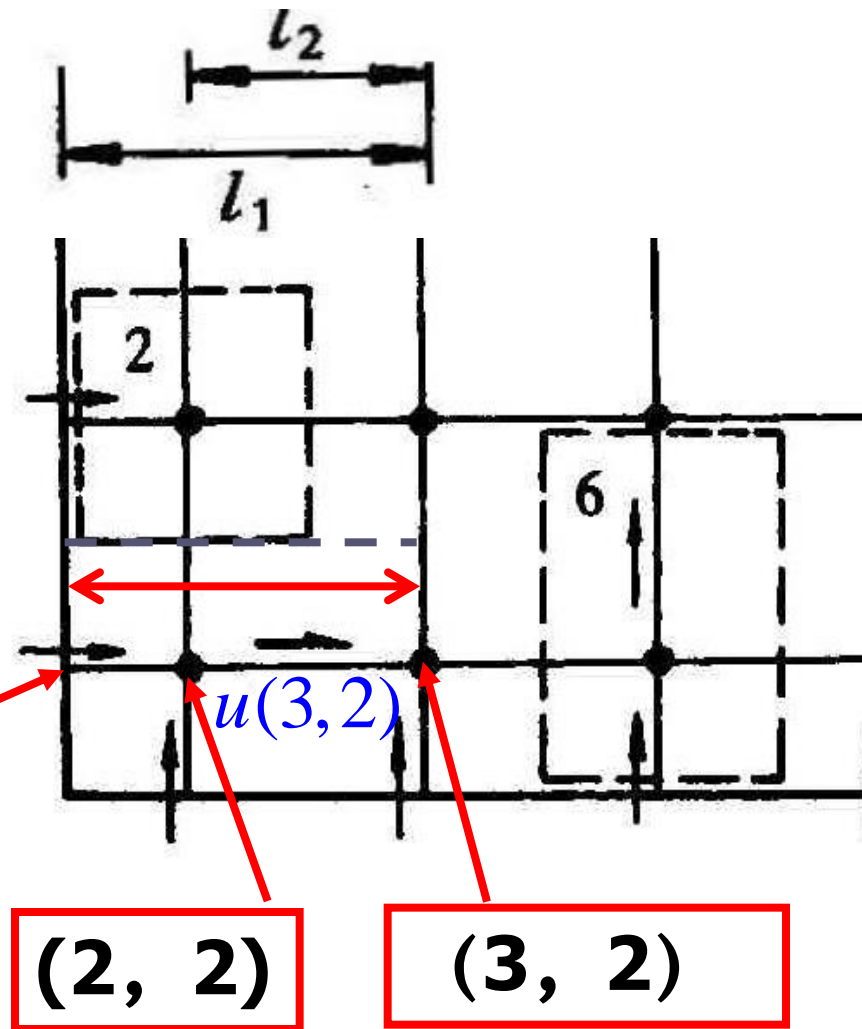
### 3. Pressure difference for velocity CV neighboring with boundary

In the iteration process the boundary pressure is not updated (更新) until convergence.

In the iteration process the pressure difference for  $u(3,2)$  is interpolated from inner data as follows:

$$p(3,2) - p(1,2) =$$

$$[p(3,2) - p(2,2)] \frac{l_1}{l_2}$$



## 6.3 Pressure correction methods for N-S equation

6.3.1 Basic idea of pressure correction methods

6.3.2 Equations for velocity corrections of  $u'$ ,  $v'$

6.3.3 Derivation of equation of pressure correction  $p'$

6.3.4 Boundary condition for pressure correction

## 6.3 Pressure correction methods for N-S equation

### 6.3.1 Basic idea of pressure correction methods

**At each iteration level after a converged velocity field is obtained based on the existing pressure field correction for the pressure field should be conducted such that the velocities corresponding to the corrected pressure field satisfy the mass conservation condition.**

- 1. Assuming a pressure field, denoted by  $p^*$ ;**
- 2. Solving the discretized momentum equations based on  $p^*$ , the results may not satisfy mass conservation;**
- 3. Improving pressure field according to mass conservation and yielding  $p'$ , for which following**

condition should be satisfied: velocities  $(u^* + u')$ ,  $(v^* + v')$  corresponding to  $(p^* + p')$  satisfy mass conservation condition;

4. Taking  $(p^* + p')$ ,  $(u^* + u')$ ,  $(v^* + v')$  as the solutions of this level for the next iteration.

### Two explanations:

(1) “Level” means a computational period during which the coefficients and source term are unchanged; Different level corresponds different coefficients and source term;

(2) Solutions  $u^*$ ,  $v^*$  based on  $p^*$  satisfy the momentum equations at that level, but do not satisfy

mass conservation condition; While the revised velocities satisfy the mass conservation but do not satisfy the momentum equation. In the iteration process both the mass conservation and the momentum equation can be satisfied gradually.

The key of pressure correction method is how to get equations for determining  $p'$ ,  $u'$ ,  $v'$ .

6.3.2 How to determine  $u'$ ,  $v'$  based on  $p'$

Suppose that the corrected pressure and velocity satisfy momentum equation

$$a_e(u_e^* + u_e') = \sum a_{nb}(u_{nb}^* + u_{nb}') + b + A_e[(p_P^* + p_P') - (p_E^* + p_E')] \quad (1)$$

Meanwhile,  $p^*$ ,  $u^*$ ,  $v^*$  satisfy the momentum equation of this level

$$a_e u_e^* = \sum a_{nb} u_{nb}^* + b + A_e (p_P^* - p_E^*) \quad (2)$$

Subtracting the two equations:

$$a_e u_e' = \sum a_{nb} u_{nb}' + A_e (p_P' - p_E')$$

Effects of neighboring velocity corrections on  $u_e'$

Effects of neighboring pressure corrections on  $u_e'$

**Analysis:** With given  $p'$  solving  $(u', v')$  from above equations is very complicated: since every velocity has its neighbors, and finally the simultaneous solution of  $u'$  and  $v'$  for the entire domain is necessary. For problems with grid number around  $10^5 \sim 10^6$ , this is unmanageable(无法实施)。

It may be expected: the effects of pressure correction are dominant, thus the effects of velocity corrections of the neighboring nodes **may be neglected, thus**

$$a_e u'_e = \sum a_{nb} u'_{nb} + A_e (p'_P - p'_E) \rightarrow a_e u'_e = A_e (p'_P - p'_E)$$

$$u'_e = \frac{A_e}{a_e} (p'_P - p'_E) = d_e (p'_P - p'_E), \quad d_e = \frac{A_e}{a_e}$$

**Similarly:**

$$v'_n = \frac{A_n}{a_n} (p'_P - p'_N) = d_n (p'_P - p'_N), \quad d_n = \frac{A_n}{a_n}$$

**The corrected velocities are:**

$$u_e = u_e^* + d_e (p'_P - p'_E) \quad v_n = v_n^* + d_n (p'_P - p'_N)$$



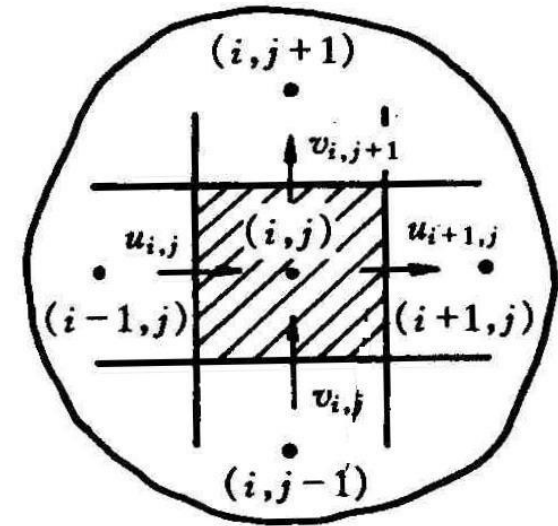
## 6.3.3 Derivation of pressure correction equation

### 1. Discretizing mass conservation

Integrating 
$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0$$

over CV P, yielding

$$\frac{\rho_P - \rho_P^0}{\Delta t} \Delta x \Delta y + [(\rho u)_e - (\rho u)_w] \Delta y + [(\rho v)_n - (\rho v)_s] \Delta x = 0$$



**Notice:** on the staggered grid system there are velocities at interfaces making the integration very easy.

## 2. Substituting the velocity correction equations

$$u_e = u_e^* + d_e (p'_P - p'_E)$$

$$u_w = u_w^* + d_w (p'_W - p'_P)$$

$$\frac{\rho_P - \rho_P^0}{\Delta t} \Delta x \Delta y + [(\rho u)_e - (\rho u)_w] \Delta y + [(\rho v)_n - (\rho v)_s] \Delta x = 0$$

$$v_n = v_n^* + d_n (p'_P - p'_N)$$

$$v_s = v_s^* + d_s (p'_S - p'_P)$$

Finally:

$$a_P p'_P = a_E p'_E + a_W p'_W + a_N p'_N + a_S p'_S + b$$

$$a_P = a_E + a_W + a_N + a_S$$

$$a_E = d_e A_e \rho_e \quad a_W = d_w A_w \rho_w \quad a_n = d_n A_n \rho_n \quad a_S = d_s A_s \rho_s$$

$$b = \frac{(\rho_P^0 - \rho_P) \Delta x \Delta y}{\Delta t} + [(\rho u^*)_w - (\rho u^*)_e] A_e + [(\rho v^*)_s - (\rho v^*)_n] A_n$$

**Remarks: If mass conservation condition was satisfied in the previous iteration, then  $b = 0$ ; Thus the  $b$  term in the  $p'$  equation reflects whether the mass conservation of each CV is satisfied, and can serve as a criterion for convergence of the iteration.**

## 6.3.4 Boundary condition of pressure correction equation

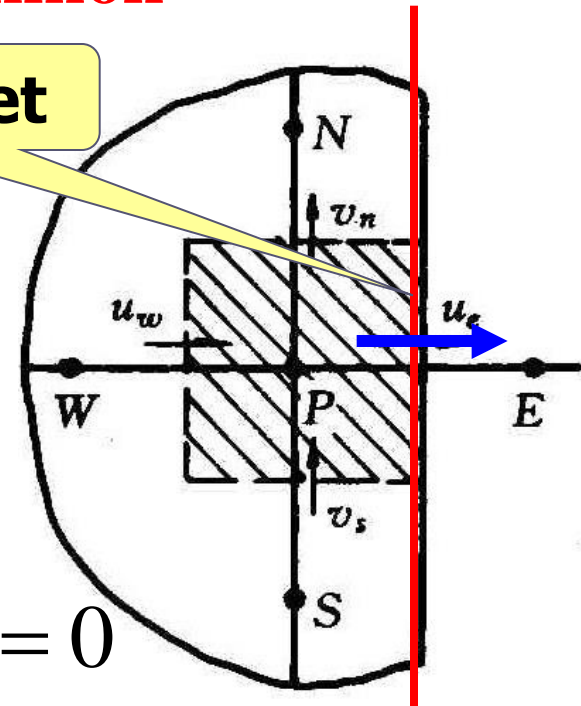
In essence, equations for pressure correction is a kind of discretized mass conservation equation. **All differential equations should have their boundary conditions for unique solutions. So does the pressure correction equation. There are two common boundary conditions:**

### 1. Given normal velocity

For given  $u_e$   $\xrightarrow{u_e^*}$   $u_e = u_e^* + u_e'$   $\xrightarrow{u_e^* \text{ takes given value}}$   $u_e' \equiv 0$

$u_e' = d_e \Delta p_e' \xrightarrow{u_e' \equiv 0}$   $d_e \Delta p_e' \equiv 0$   $\xrightarrow{\text{equi. to}}$   $d_e = 0$

Flow outlet



$$a_E = d_e \rho_e A_e \rightarrow a_E \equiv 0$$

## 2. Given boundary pressure

For given B.  $p$

$$p_E = p_E^* + p_E' \rightarrow p_E' \equiv 0$$

$p_E^*$  taking the given value

In the pressure correction eq.

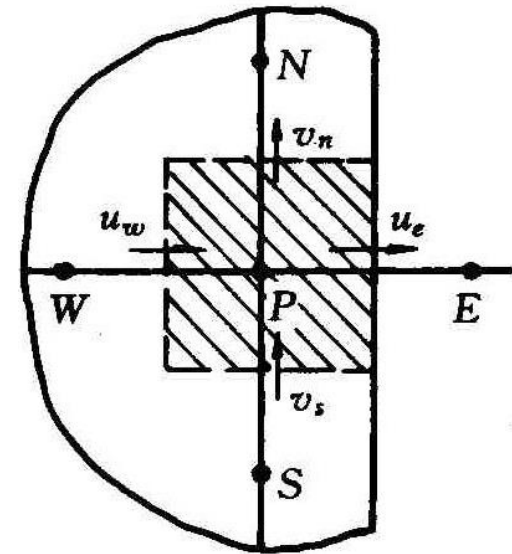
$a_E$  appears in term of  $a_E p_E$

$$a_E p_E' = 0 \rightarrow a_E \equiv 0!$$

Equi. to

The two boundary conditions both lead to

$$a_E \equiv 0.$$



# 同舟共济 渡彼岸!

People in the same  
boat help each  
other to cross to the  
other bank, where....

