数值传热学

第十二章 网格生成技术

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12月7日, 西安

Numerical Heat Transfer
(数值传热学)
Chapter 12 Grid Generation Techniques

Instructor Tao, Wen-Quan

CFD-NHT-EHT Center
Key Laboratory of Thermo-Fluid Science & Engineering
Xi'an Jiaotong University
Xi'an, 2016-12-07

Chapter 12 Grid Generation Techniques

12.1 Treatments of Irregular Domain in FDM,FVM

12.2 Introduction to Body-Fitted Coordinates

12.3 Algebraic Methods for Generating Body-Fitted Coordinates

12.4 PDE Method for Generating Body-Fitted Coordinates

12.5 Control of Grid Distribution

12.6 Transformation and Discretization of Governing Eq. and Boundary Conditions

12.7 SIMPLE Algorithm in Computational Plane

12.8 Post-Process and Examples

12.1 Treatments of Irregular Domain in FDM,FVM

12.1.1 Conventional orthogonal coordinates can not deal with variety of complicated geometries

12.1.2 Methods in FDM,FVM to deal with complicated geometries

1. Structured grid (结构化网格)

   1) Domain extension method

   2) Special orthogonal coordinates

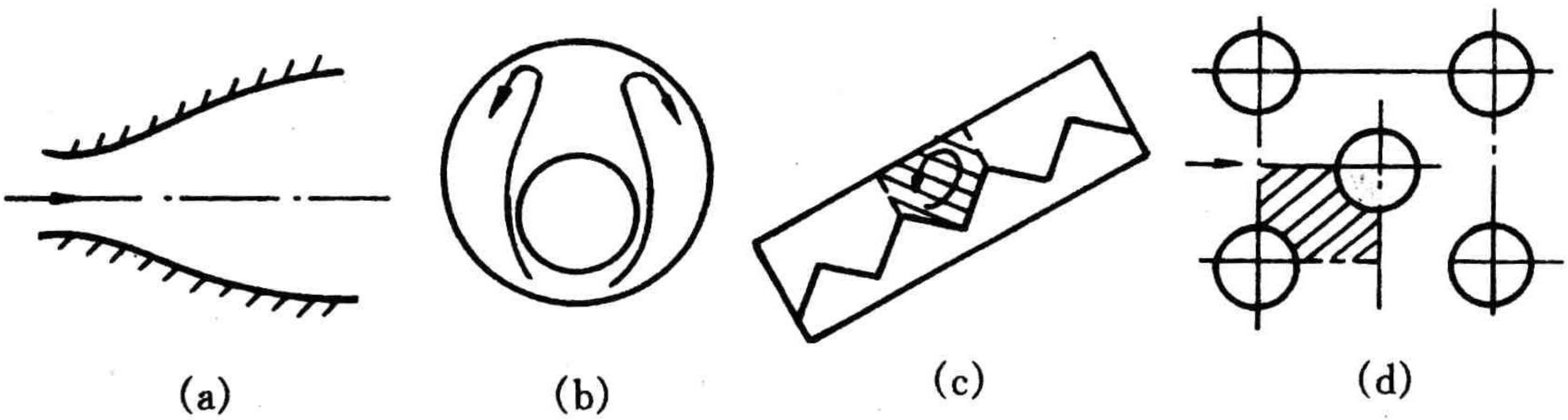
   3) Composite grid (组合网格)

   4) Body-fitted coordinate (适体坐标系)

2. Unstructured grid (非结构化)

12.1 Treatments of Irregular Domain in FDM,FVM

12.1.1 Conventional orthogonal (正交)coordinates can not deal with variety of complicated geometries



Plane nozzle

Eccentric annulus (偏心圆环)

Solar collector

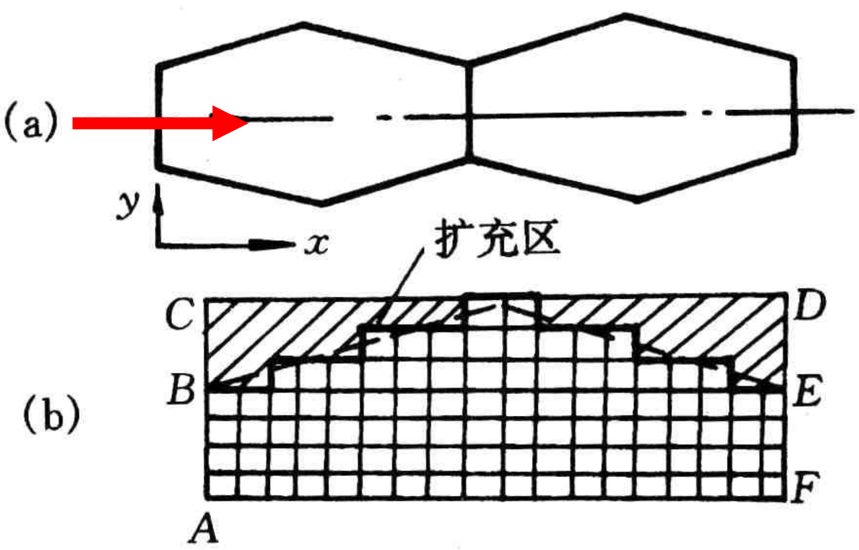
Tube bank

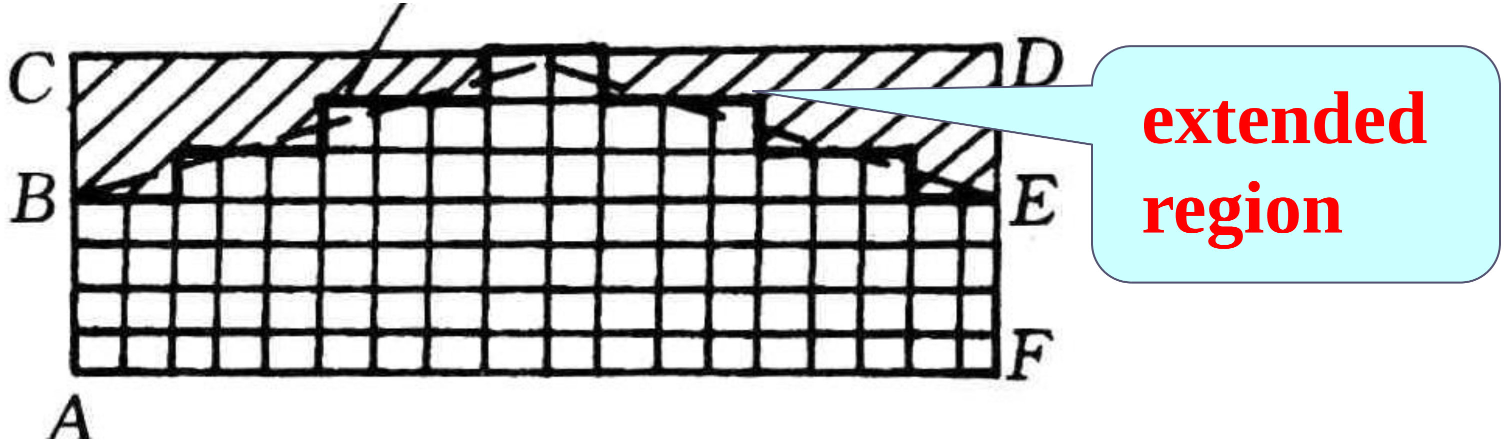
12.1.2 Methods in FDM,FVM to deal with complicated geometries

1. Structured grid (结构化网格)

1) Domain extension method(区域扩充法)

An irregular domain is extended to a regular one, the irregular boundary is replaced by a step-wise approximation, and simulation is performed in a conventional coordinate.





(1) Flow field simulation

(a) Set zero velocity at the boundaries of extended region at B-C-D-E: u=v=0;

(b) Set a very large viscosity in the extended region

$$\eta = 10^{25} \sim 10^{30};$$

(c) Set interface diffusivity by harmonic mean

(2) Temperature field prediction

(a) First kind boundary condition with uniform temperature: **The same as for velocity： in the extended region the thermal conductivity is set to e very large, $\lambda = 10^{25} \sim 10^{30}$ and boundary temperatures are given**

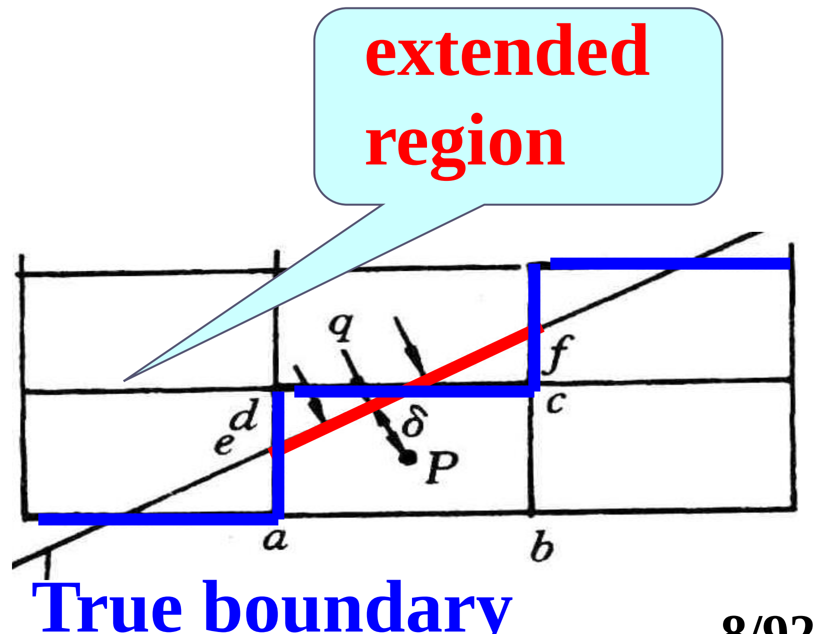
(b) Second kind boundary conditions by ASTM

Specified boundary heat flux distribution (not necessary uniform)

For CV P adding additional source term:

$$S_{C,ad} = \frac{q \cdot \overline{ef}}{\Delta V_P};$$

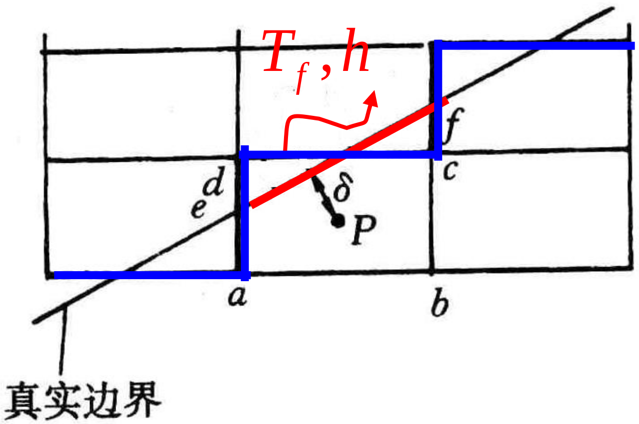
And setting zero conductivity for the extended region to avoid heat transfers outward.



(c) Third kind boundary conditions by ASTM

Specified external convective heat transfer coefficient and temperature, h and T_f,

For CV. P following source term is added



$$S_{C,ad} = \frac{\overline{ef}}{\Delta V_P} \frac{T_f}{1/h + \delta/\lambda};$$

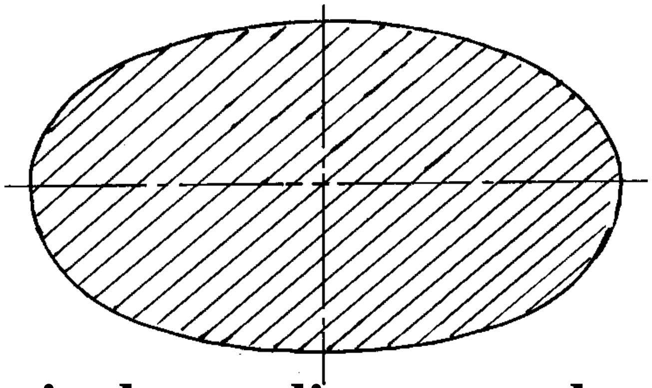
$$S_{P,ad} = -\frac{\overline{ef}}{\Delta V_P} \frac{1}{1/h + \delta/\lambda};$$

And setting zero conductivity ($\lambda = 0$) for the extended region to avoid heat transfers outward.

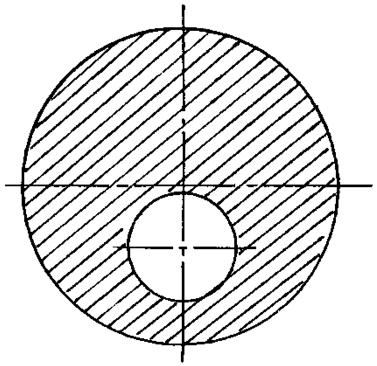
For not very complicated geometries, is is a convenient method.

2) Special orthogonal (正交的) coordinates

There are 14 orthogonal coordinates, and they can be used to deal with some irregular regions



Elliptical coordinate can be used to simulate flow in elliptic tube

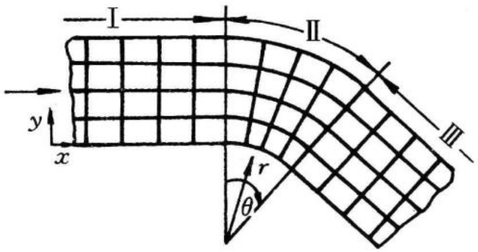


Bi-polar coordinate (双极坐标） can be used for flow in a biased annulus(偏心环)

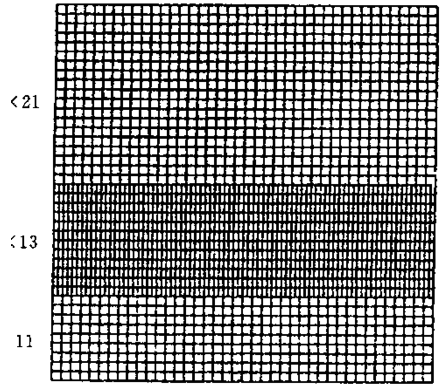
3) Composite coordinate（block structured）

The entire domain is composed of several blocks, for each block individual coordinate is adopted and solutions are exchanged at the interfaces between different blocks.

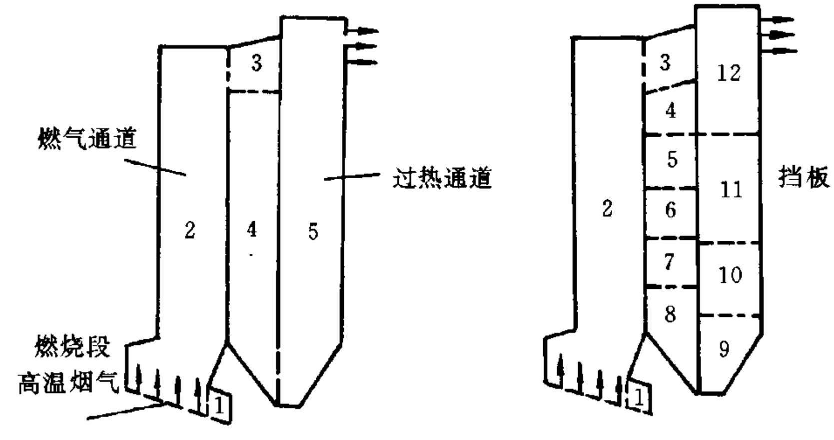
Mathematically it is called domain decomposition method (区域分解法）.



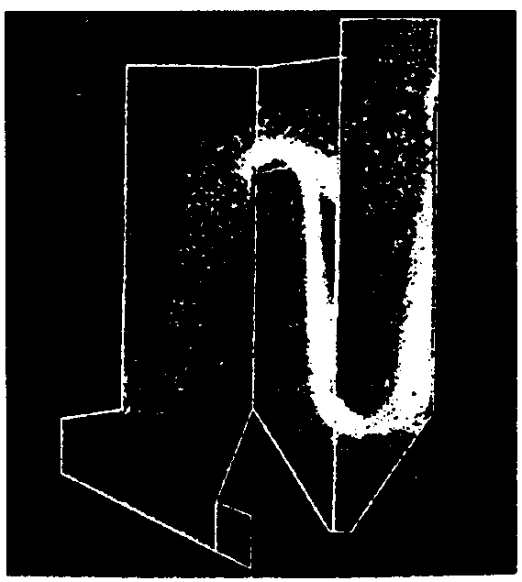
Grid lines are continuous. The entire domain can be solved by ADI.



Grid lines are discontinuous

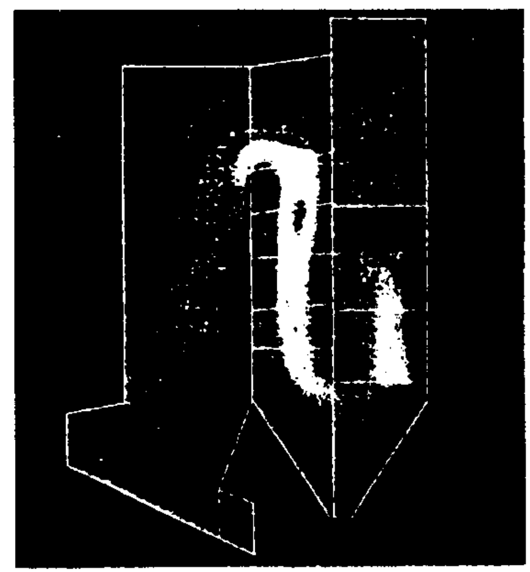


Application example



(a) 原设计

Original design



(b) 改型设计

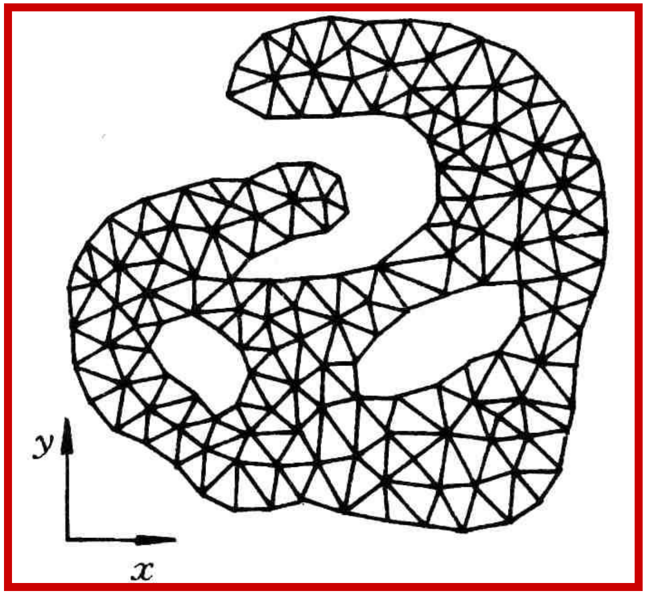
Improved design

4) Body-fitted coordinates(适体坐标)

In such coordinates the coordinates are fitted with(适应) the domain boundaries; The generation of such coordinates by numerical methods is the major concern of this chapter.

2. Unstructured grid (非结构化网格)

There are no fixed rules for the relationship between different nodes, and such relationship should be specially stored for each node. Computationally very expensive. Adopted for very complicated geometries.



12.2 Introduction to Body-Fitted Coordinates

12.2.1 Basic idea for solving physical problems by BFC

12.2.2 Why domain can be simplified by BFC

12.2.3 Methods for generation of BFC

12.2.4 Requirements for grid system constructed by BFC

12.2.5 Basic solution procedure by BFC

12.2 Introduction to Body-Fitted Coordinates

12.2.1 Basic idea for solving physical problems by BFC

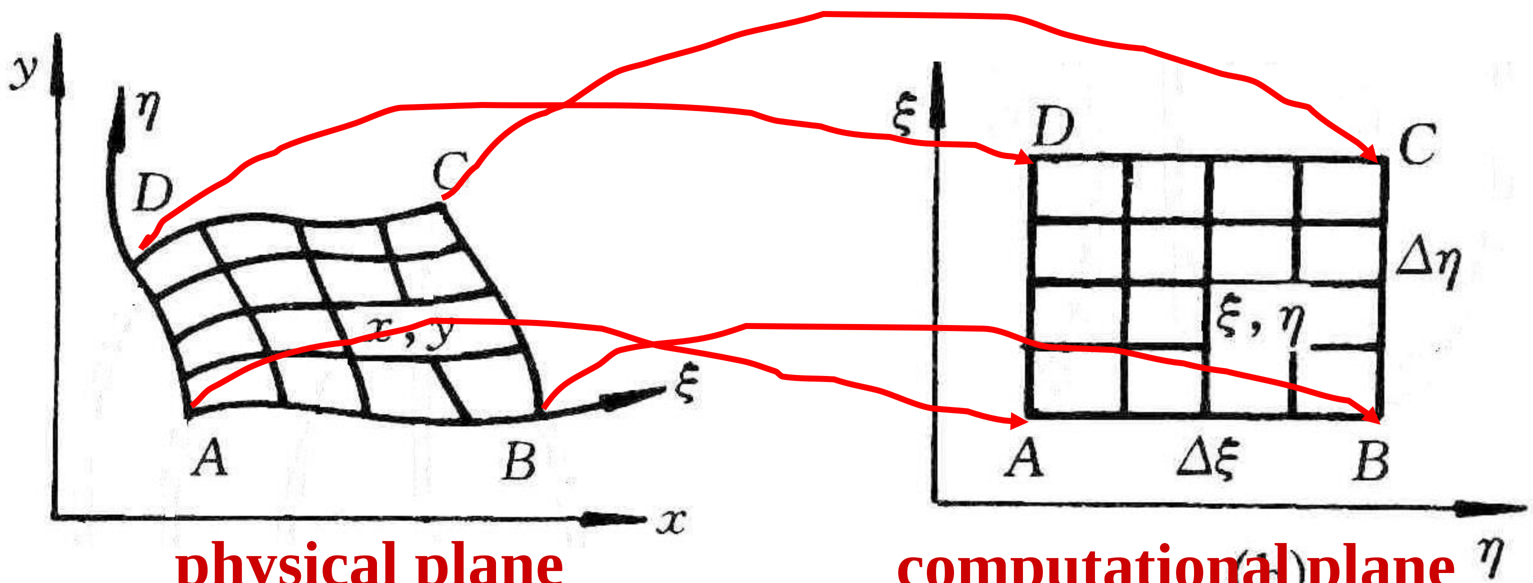
1.In the numerical simulation of physical problems the most ideal coordinate is the one which fits with the boundaries of the studied problem, called body-fitted coordinates(适体坐标系): Cartesian coordinate is the body-fitted one for rectangles, polar coordinate is the one for annular spaces.

2.The existing orthogonal coordinates can not deal with variety of complicated geometries in different engineering ; Thus body-fitted coordinates artificially constructed are necessary to meet the different practical requirements.

12.2.2 Why domain can be simplified by BFC

1.Assuming that a BFC has been constructed in Cartesian coordinate x-y, denoted by $\xi-\eta$;

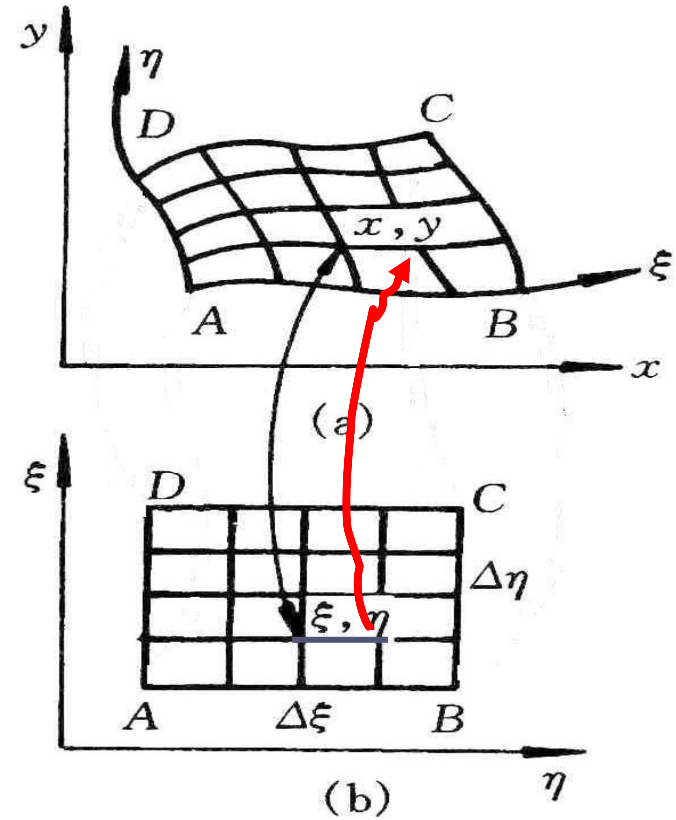
2.Regarding ξ and η as the two coordinates of a Cartesian coordinate in a **computational plane**, then the irregular geometry in physical plane transforms to a rectangle in the computational plane



physical plane **computational plane**

3.The **grids in computational plane are always uniformly distributed**, thus once grid number is given, the grid system in computational plane can be constructed with ease.

4.**Simulation is first conducted in the computational plane** , then the converged solution is transferred from the computational plane to physical one. In such a way the simulation domain is greatly simplified.



5.In order to transfer solutions from computational domain to physical domain, **it is necessary to obtain the corresponding relations of nodes between the two planes;**

The so-called grid generation technique refers to the methods by which from (ξ, η) in the computational plane the corresponding (x, y) in Cartesian coordinate can be obtained.

12.2.3 Methods for generation of BFC

1. Conforming mapping(保角变换法)

2. Algebraic method(代数法)

The correspondent relations between grids of two planes are represented by algebraic equations.

3. PDE method(微分方程法)

The relations are obtained through solving PDE. Three kinds of PDE, hyperbolic, parabolic and elliptic, all can be used to provide such relations.

12.2.4 Requirements for grid system constructed by BFC

1. The nodes in two planes should be one to one correspondent（一一对应）．

2. Grid lines in physical plane should be normal to the boundary .

3. The grid spacing in the physical plane can be controlled easily.

12.2.5 Procedure of solving problem by BFC

1. Generating grid: find the one to one correspondence between $(\xi,\eta)\leftrightarrow(x,y)$

2. Transforming governing eqs. and boundary conditions from physical plane to computational plane;

3. Discretizing gov. eq. and solving the ABEqs. in

computational plane.

4. Transferring solutions to the physical plane.

12.3 Algebraic Methods for Generating Body-Fitted Coordinates

12.3.1 Boundary normalization（边界规范化）

1. 2 D nozzle

2. Trapezoid enclosure（梯形封闭空腔）

3. Eccentric annular space（偏心圆环）

4. Plane duct with one irregular boundary

12.3.2 Two-boundary method（双边界法）

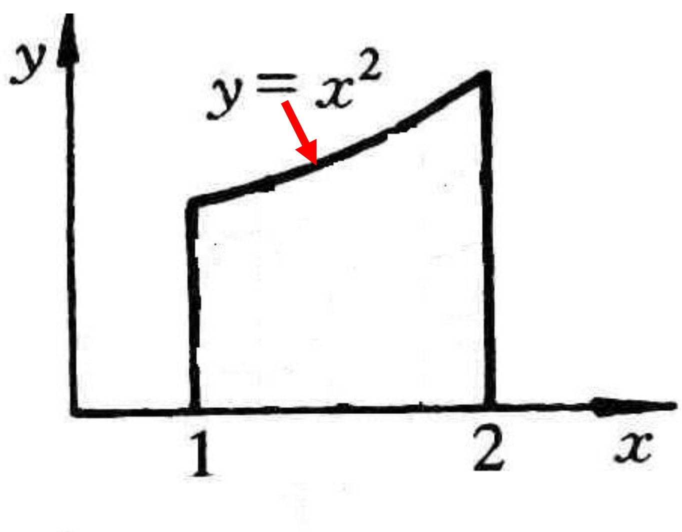
12.3 Algebraic Methods for Generating Body-Fitted Coordinates

12.3.1 Boundary normalization（边界规范化）

1. 2 D nozzle

A plane nozzle is given by following profile

$$y = x^2$$

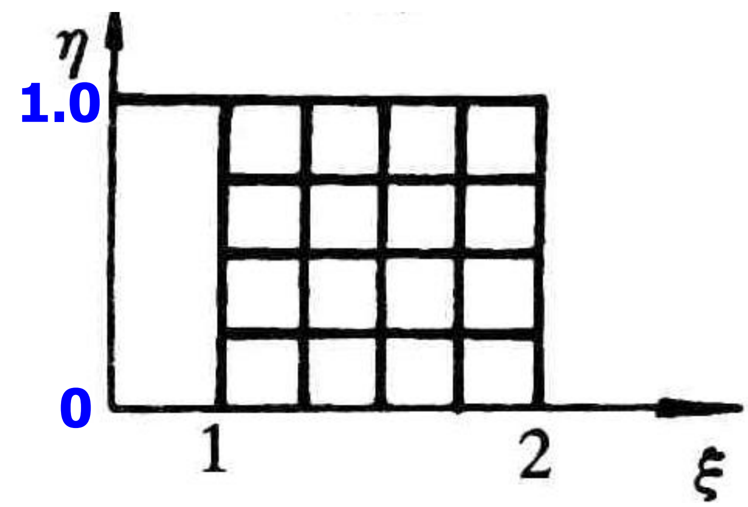


$$\boxed{\xi = x}$$

$$\boxed{\eta = y / y_{\max}}$$

normalization

$$y_{\max} = x^2$$

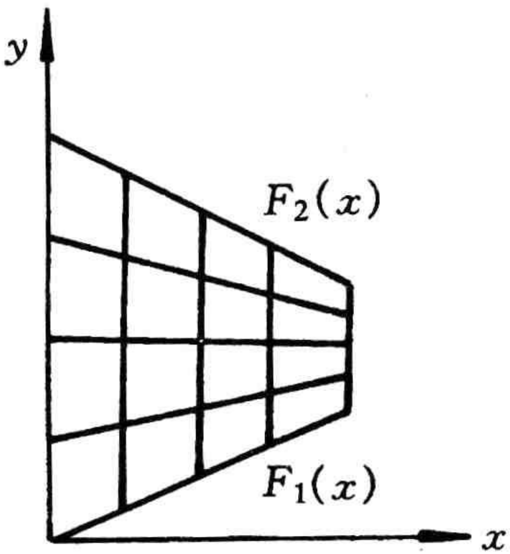


2. Trapezoid (梯形) enclosure

Functions of two tilted boundaries are given by：

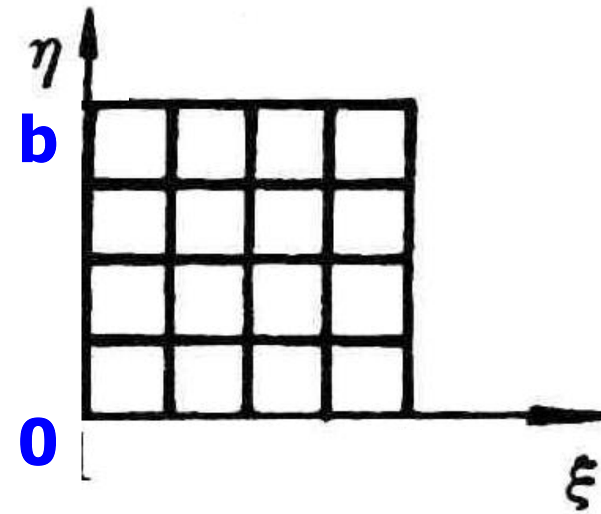
$$F_1(x), F_2(x)$$

The grid in the trapezoid enclosure is generated.



$$\boxed{\xi = ax}$$

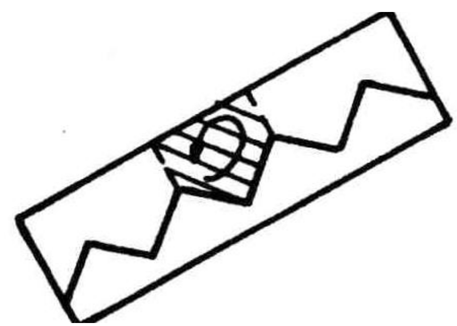
$$\boxed{\eta = b\frac{y - F_1(x)}{F_2(x) - F_1(x)}}$$



normalization

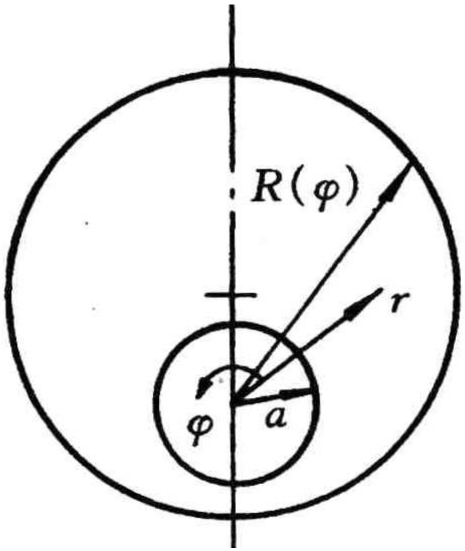
Normalized by the distance between top and bottom

Solar collector



3. Eccentric annular space

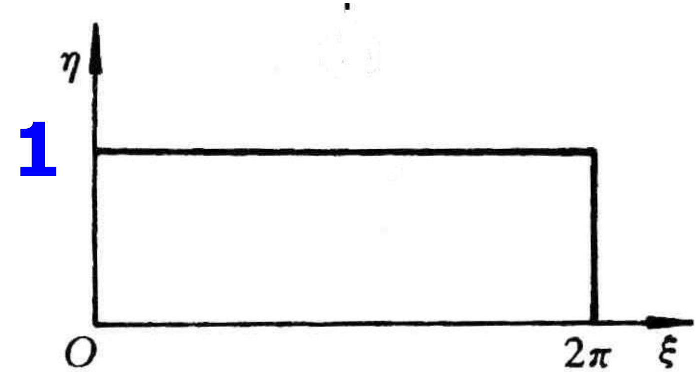
Given two radiuses (R,a) and the eccentric distance



$$\boxed{\xi = \varphi}$$

$$\boxed{\eta = \frac{r-a}{R(\varphi)-a}}$$

normalization

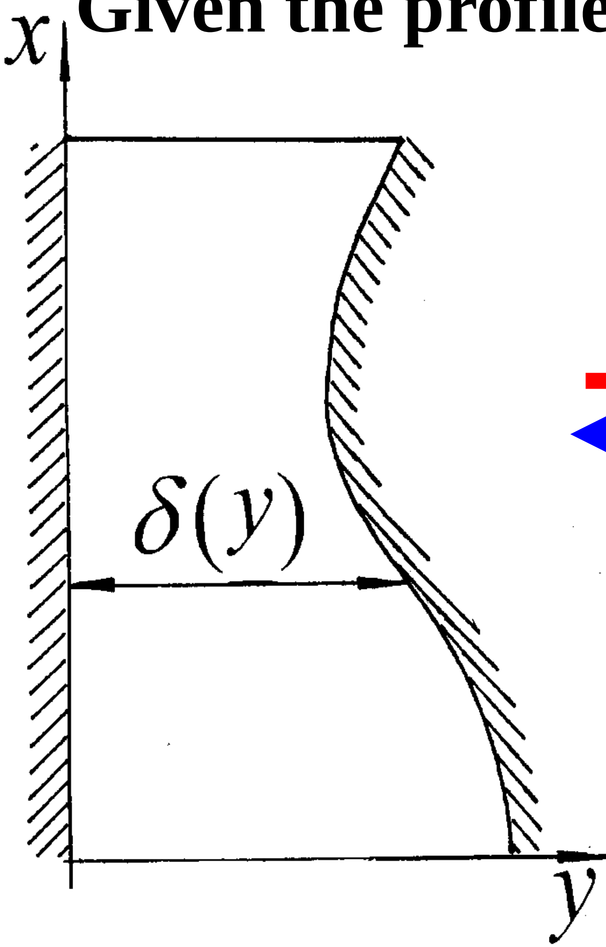


Normalized by the distance between outer and inner circles

Prusa,Yao, ASME J H T, 1983， 105:105-116

4. Plane duct with one irregular boundary

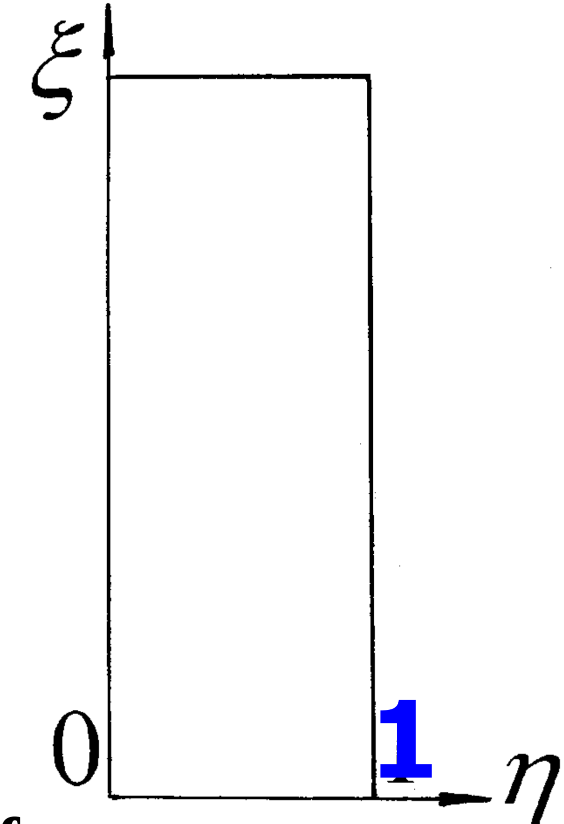
Given the profile of the irregular boundary $\delta(y)$



$$\boxed{\xi = x}$$

$$\boxed{\eta = \frac{y}{\delta(x)}}$$

normalization



Normalized by the distance between left and right boundaries

Sparrow-Faghri-Asako, p.479 of Textbook

12.3.2 Two-boundary method

1. Method for transforming an irregular quadrilateral（四边形） in physical plane to a rectangle in computational plane.

Implementing procedure:

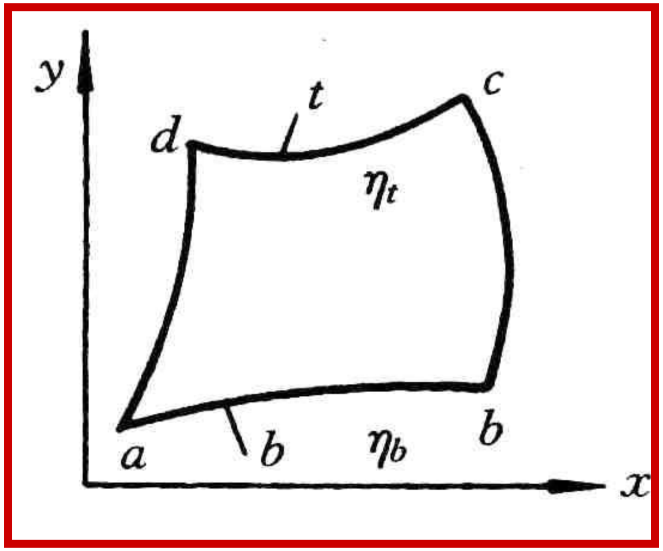
1) Setting values of η for two opposite (相对的) boundaries:

say： $\eta)_{ab} = \eta_b = 0;\ \eta)_{cd} = \eta_t = 1$

2) Setting the rules of how x,y vary with ξ on the two boundaries：

$$x_b = x_b(\xi), y_b = y_b(\xi)$$

$$x_t = x_t(\xi), y_t = y_t(\xi)$$



3) For any pair of (x,y) and (ξ,η) within the domain taking following interpolations

$$x(\xi,\eta) = x_b(\xi,0)\,[1 - f_1(\eta)] + f_1(\eta)\,x_t(\xi,1)$$

$$y(\xi,\eta) = y_b(\xi,0)[1 - f_1(\eta)] + f_1(\eta)y_t(\xi,1)$$

where $f_1(\eta)$ must satisfy following conditions:

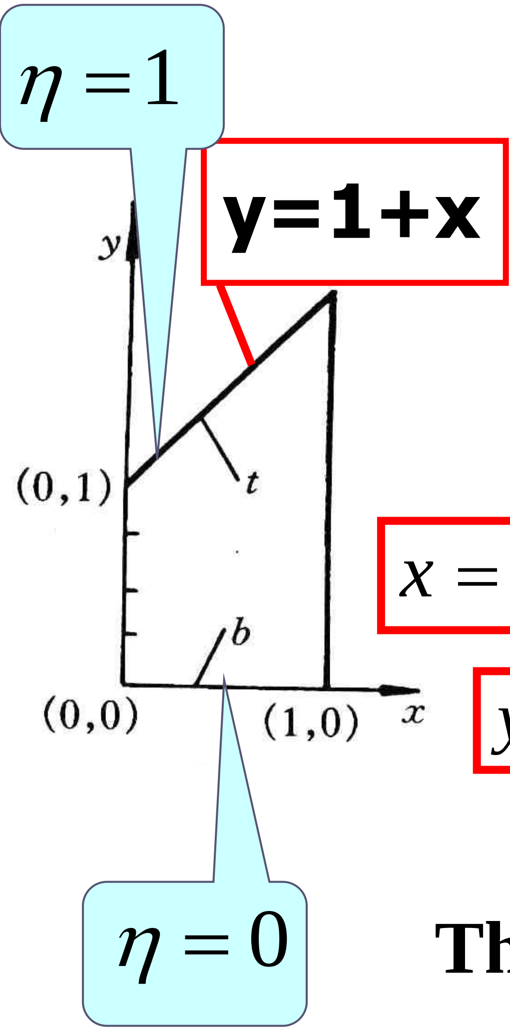
$$\eta = 0, x(\xi,\eta) = x_b(\xi), y(\xi,\eta) = y_b(\xi)$$

$$\eta = 1, x(\xi,\eta) = x_t(\xi), y(\xi,\eta) = y_t(\xi)$$

The most simple interpolation which satisfies such conditions is

$$f_1(\eta) = \eta$$

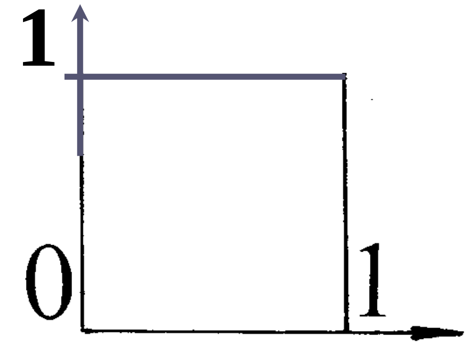
2. Example



$$x(\xi,\eta) = x_b \bullet (1-\eta) + x_t \bullet \eta$$

$$y(\xi,\eta) = y_b \bullet (1-\eta) + y_t \bullet \eta$$

$$x_b = \xi,\ y_b = 0;\ x_t = \xi,\ y_t = 1+\xi$$



$$x = \xi \bullet (1-\eta) + \xi \bullet \eta = \xi \qquad x = \xi$$

$$y = 0 \bullet (1-\eta) + (1+\xi) \bullet \eta \qquad y = \eta(1+\xi)$$

That is: $\begin{cases} \xi = x \\ \eta = \dfrac{y}{1+x} \end{cases}$

The same as that by boundary normalization method.

12.4 PDE Method for Generating Body-Fitted Coordinates

12.4.1 Known conditions and task of grid generation by PDE

12.4.2 Problem set up of grid generation by PDE

1. Starting from physical plane

2. Starting from computational plane

12.4.3 Procedure of grid generation by solving an Elliptic-PDE

12.4.4 The metric identity should be satisfied

12.4 PDE Method for Generating Body-Fitted Coordinates

12.4.1 Known conditions and task of grid generation by PDE

1. The grid distribution in computational plane is given;

2. The grid arrangement on the physical boundary is given.

Find: the one to one correspondence between $(x, y),(\xi,\eta)$

i,e: $(x, y) \leftrightarrow (\xi,\eta)$

12.4.2 Problem set up of grid generation by PDE

1. Starting from physical plane

Regarding (ξ,η) as two dependent variables to be solved in physical plane; then above given conditions are equivalent to: Given boundary values of the two dependent variables:

$$\xi_B = f^{\xi}(x_B, y_B), \eta_B = f^{\eta}(x_B, y_B)$$

Find values of (ξ, η) for any inner point (x, y) within the solution region in physical plane.

This is a boundary value problem in physical plane. The most simple governing equation is Laplace eq.:

$$\begin{cases} \nabla^2 \xi = 0;\ \nabla^2 \eta = 0 \quad \textbf{or} \quad \xi_{xx} + \xi_{yy} = 0,\ \eta_{xx} + \eta_{yy} = 0 \\ \xi_B, \eta_B \ \textbf{given (i.e.,}\ \xi, \eta \ \textbf{of boundary nodes are known)} \end{cases}$$

However, this problem should be solved for a domain in physical plane, which is irregular! Thus we have the same difficulty as for the original problem!

2. Starting from computational plane

Now we regard (x, y) as the dependent variables in computational domain, the above conditions are equivalent to solve a boundary value problem in computational domain: with given boundary values of x and y

$$x_B = f^x(\xi_B, \eta_B), y_B = f^y(\xi_B, \eta_B)$$

it is required to find (x, y) for any inner point (ξ, η) in computational plane.

This is a boundary value problem in a regular computational domain. **This treatment greatly simplify the problem because in computational plane the solution region is either a rectangle or a square.**

It should be noted that the boundary value problem in computational domain can not be simply expressed

as:

$$x_{\xi\xi} + x_{\eta\eta} = 0; \quad y_{\xi\xi} + y_{\eta\eta} = 0$$

According to mathematical rules the correspondent expression are:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0; \quad \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0$$

$$\alpha = x_\eta^2 + y_\eta^2; \quad \beta = x_\xi x_\eta + y_\xi y_\eta; \quad \gamma = x_\xi^2 + y_\xi^2$$

where subscript stands for derivative, and parameter β represents the orthogonality (正交性) of grid lines in physical plane： two orthogonal lines have zero value.

Thus the essence of grid generation is to solve a boundary value problem in computational domain! The boundary value problem is formulated by an elliptic partial differential equation.

12.4.3 Procedure of grid generation by solving an elliptic-PDE

1. Determining the number of nodes in physical plane and constructing grid network in computational plane;

2. Setting boundary nodes in physical plane according to given conditions;

3. Solving two boundary value problems in computational plane, by regarding them as non-isotropic and nonlinear diffusion problems with source term.

4. Calculating $x_\xi, x_\eta, y_\xi, y_\eta$ after getting the correspondence between (ξ, η) and (x, y) .

12.4.4 The metric identity should be satisfied

$$\frac{\partial\phi}{\partial x}=\frac{\partial\phi}{\partial\xi}\left(\frac{\partial\xi}{\partial x}\right)+\frac{\partial\phi}{\partial\eta}\left(\frac{\partial\eta}{\partial x}\right)=\phi_{\xi}\xi_{x}+\phi_{\eta}\eta_{x}=\frac{1}{J}[(\phi y_{\eta})_{\xi}-(\phi y_{\xi})_{\eta}]$$

where: $J=x_{\xi}y_{\eta}-x_{\eta}y_{\xi}$ **, called Jakobi factor.**

When ϕ **is uniform** $\frac{\partial\phi}{\partial x}=0,$ **thus:** $(\phi y_{\eta})_{\xi}=(\phi y_{\xi})_{\eta}$

For uniform field: $\boxed{y_{\eta\xi}=y_{\xi\eta}}$

This equation is called metric identity(度规恒等式). In the procedure of grid generation this identity should be satisfied. Otherwise artificial source will be introduced.

In the transformation of govern, eq. from physical plane to computational plane such kind of derivatives will be introduced.

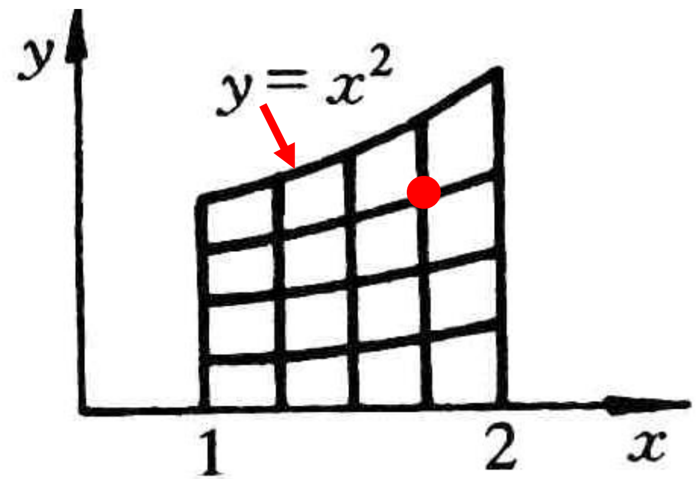
In order to guarantee the satisfaction of metric identity Thompson et al. proposed following method:

(1) All derivatives with respect to geometric position must be determined by discretized form;

(2) Any such kind of derivative must be computed directly, no interpolation can be used.

Example

[Find] y_ξ, y_η **for the position of x=1.75, y=2.2969 in the 2D nozzle problem.**



(a)

[Calculation] (1) The position of this point (ξ,η) **in computational plane is determined:**

$$\xi = x = 1.75;\ \eta = y / y_{\max} = 2.2969/1.75^2 = 0.75$$

(2) According to definition:

$$y_\eta = \frac{\partial y}{\partial \eta}\Big)_{\xi=cons} = \frac{y(\xi,\eta+\Delta\eta)-y(\xi,\eta-\Delta\eta)}{2\Delta\eta} =$$

$$\frac{y[1.75,(0.75+0.25)]-y[1.75,(0.75-0.25)]}{2\times 0.25}=$$

$$\frac{y(1.75,1.0)-y(1.75,0.5)}{0.5}\xrightarrow[\xi=x]{y=\eta x^2}\frac{1\times 1.75^2-0.5\times 1.75^2}{0.5}=3.0625$$

$$y_\xi=\frac{\partial y}{\partial \xi})_{\eta=cons}=\frac{y(\xi+\Delta\xi,\eta)-y(\xi-\Delta\xi,\eta)}{2\Delta\xi}=$$

$$\frac{y[(1.75+0.25),0.75]-y[(1.75-0.25),0.75]}{2\times 0.25}=$$

$$=\frac{y(2.0,0.75)-y(1.5,0.75)}{0.5}\xrightarrow[\xi=x]{y=\eta x^2}=\frac{0.75\times 2.0^2-0.75\times 1.5^2}{0.5}=2.6250$$

$$y_\eta=3.0625;\quad y_\xi=2.6250$$

12.5 Control of Grid Distribution

12.5.1 Major features of grid system generated by Laplace equation

12.5.2 Grid system generated by Poisson equation

12.5.3 Thomas-Middlecoff method for determining P,Q function

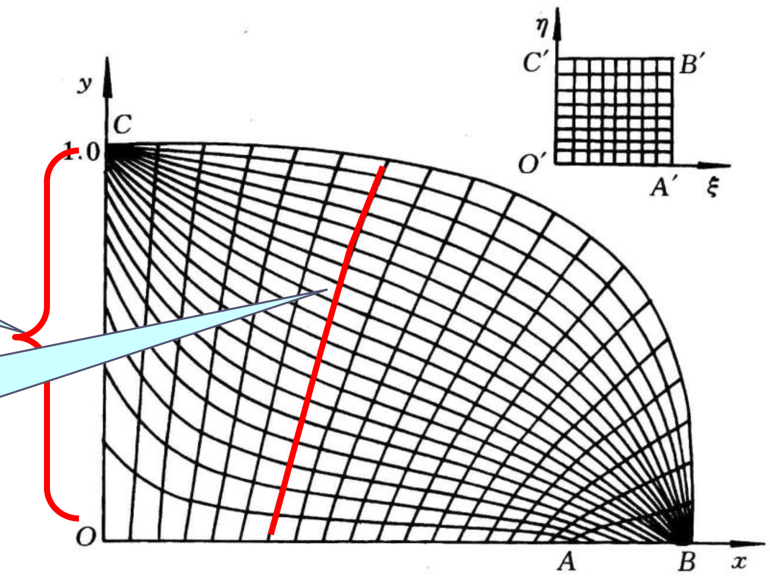
12.5 Control of Grid Distribution

12.5.1 Major features of grid system generated by Laplace equation

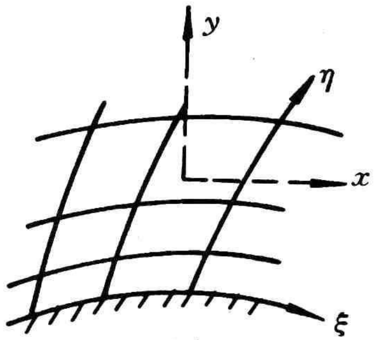
1.The grid distribution is automatically unified within the solution domain

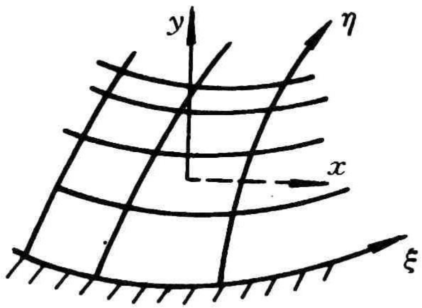
Strongly non-uniform distribution at left boundary

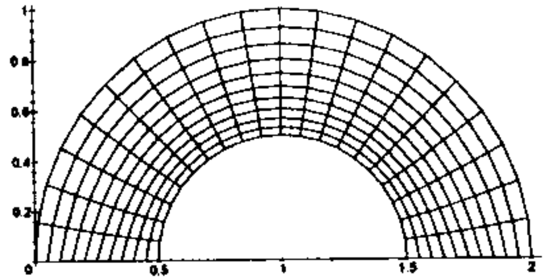
In the domain grid distribution has been unified.



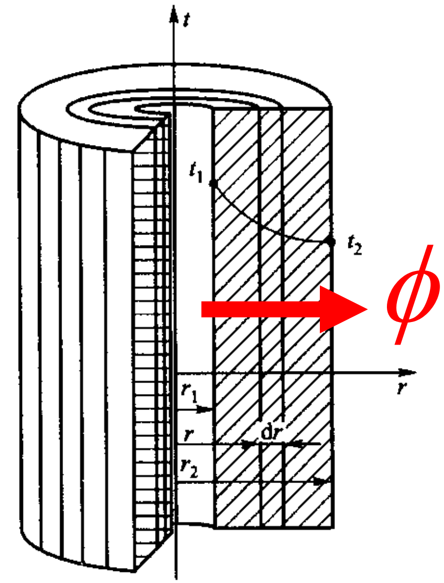
2.Along the normal to a curved wall spacing between grid lines changes automatically.







Such features are inherently related to diffusion process: For heat conduction through a cylindrical wall heat flux gradually deceases along radius and spacing between two isothermals increases.



Thus it is needed to develop techniques for controlling grid distribution: grid density and the orthogonality of gridline with boundary.

12.5.2 Grid generation by Poisson equation

1.Heat transfer theory shows that high heat flux leads to dense isothermal（等温线） distribution. If gridlines are regarded as isothermals， then their density can be controlled by heat source. Heat conduction with source term is governed by Poisson equation.

In physical plane Poisson equation is：

$$\nabla^2\xi = P(\xi,\eta); \nabla^2\eta = Q(\xi,\eta)$$

In computational plane, it becomes:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2[P(\xi,\eta)x_\xi + Q(\xi,\eta)x_\eta]$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2[P(\xi,\eta)y_\xi + Q(\xi,\eta)y_\eta]$$

$$\alpha = x_\eta^2 + y_\eta^2; \quad \beta = x_\xi x_\eta + y_\xi y_\eta; \quad \gamma = x_\xi^2 + y_\xi^2$$

12.5.3 Thomas-Middlecoff method for P,Q

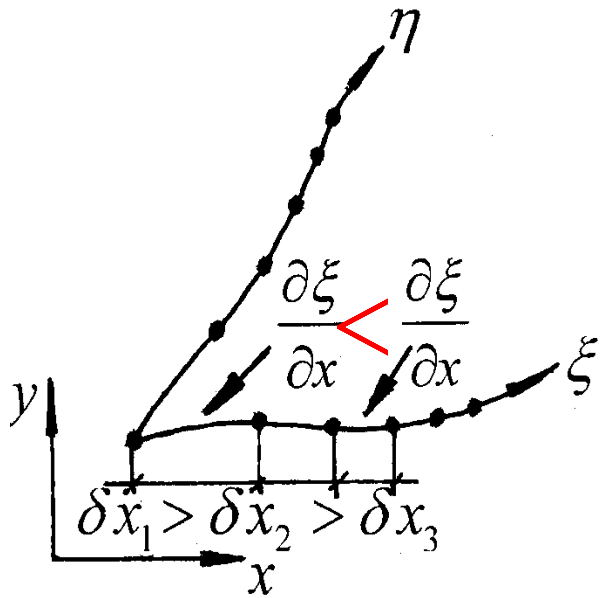
P,Q are source function for controlling density and orthogonality, and can be constructed by different methods. Thomas－Middlecoff method is very meaningful and easy to be implemented. Its implementation procedure is introduced as follows .

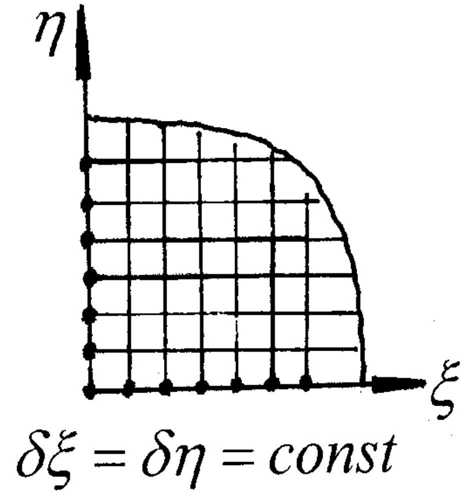
1.Assuming that

$$P(\xi,\eta)=\underline{\phi(\xi,\eta)}(\underline{\xi_x^2+\xi_y^2});Q(\xi,\eta)=\psi(\xi,\eta)(\eta_x^2+\eta_y^2)$$

Controlling the orthogonality of boundary grid line

Controlling grid density within domain---transmitting the specified density on the boundary to inner region





The first derivatives of ξ, η with respect x, y , ξ_x, η_x, in the physical plane reflect the rate of changes.

For grid generation, $\xi_x, \xi_y, \eta_x, \eta_y$ are known along the boundary;

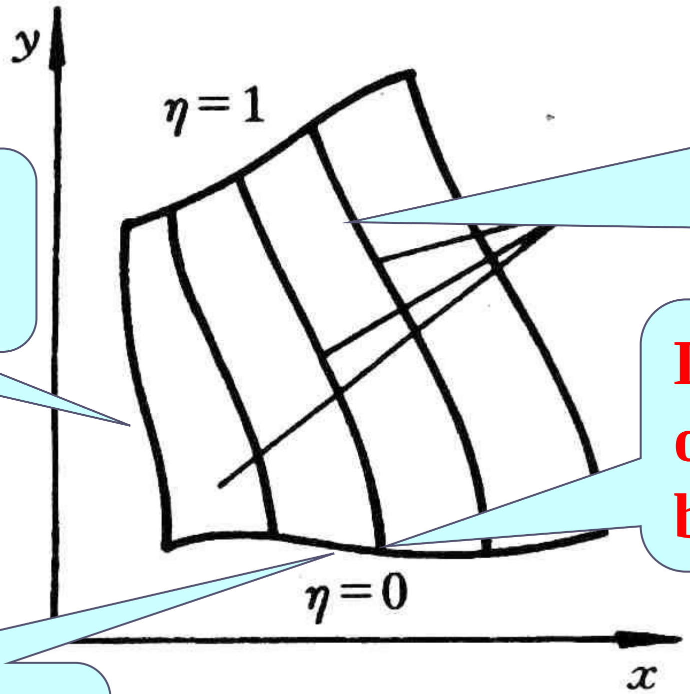
The key is to determine ϕ, ψ

2. Ways for determining ϕ and ψ

1) ϕ is first determined for the bottom and top boundaries where η is constant; ψ is first determined for the left and right boundaries where ξ is constant.

The boundary values of ϕ and ψ should satisfy following conditions: the local gridlines are straight and normal to the relative boundary (直线且垂直边界).

2) On the constant ξ lines between bottom and top, the values of ϕ are linearly interpolated with respect to η ; On the constant η lines between left and right boundaries the values of ψ are interpolated linearly with respect to ξ .



ξ = C, determining ψ

On the line ϕ is linearly interpolated with respect to η

Locally straight and orthogonal to the boundary

η = C, determining ϕ

Then our task is to determine ϕ for $\eta = 0$ and $\eta = 1$; and determine ψ for $\xi = 0$ and $\xi = 1$.

3. Way for determining ϕ on $\eta=0, \eta=1$

1) Substituting

$$P(\xi,\eta)=\phi(\xi,\eta)(\xi_x^2+\xi_y^2);Q(\xi,\eta)=\psi(\xi,\eta)(\eta_x^2+\eta_y^2)$$

into the Poisson equation in computational plane

$$\alpha x_{\xi\xi}-2\beta x_{\xi\eta}+\gamma x_{\eta\eta}=-J^2[\underline{P(\xi,\eta)}x_\xi+\underline{Q(\xi,\eta)}x_\eta]$$

$$\alpha y_{\xi\xi}-2\beta y_{\xi\eta}+\gamma y_{\eta\eta}=-J^2[\underline{P(\xi,\eta)}y_\xi+\underline{Q(\xi,\eta)}y_\eta]$$

Rewriting above equations in terms of ϕ, ψ ,

obtaining two simultaneous equations:

$$\alpha(y_{\xi\xi}+\phi y_\xi)-2\beta y_{\xi\eta}+\gamma(y_{\eta\eta}+\psi y_\eta)=0$$

$$\alpha(x_{\xi\xi}+\phi x_\xi)-2\beta x_{\xi\eta}+\gamma(x_{\eta\eta}+\psi y_\eta)=0$$

2) Eliminating ψ from above two equations, obtaining equation of ϕ

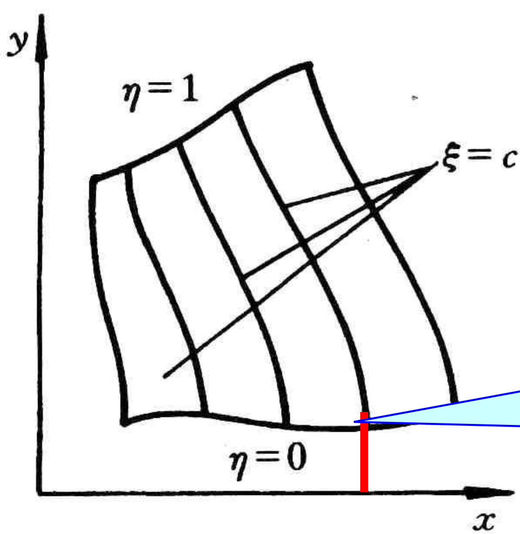
$$\alpha[y_\eta(x_{\xi\xi}+\phi x_\xi)-x_\eta(y_{\xi\xi}+\phi y_\xi)]=$$

$$y_\eta^2[2\beta(x_\eta / y_\eta)_\xi+\gamma\underline{(x_{\eta\eta}y_\eta-y_{\eta\eta}x_\eta)/y_\eta^2}]$$

0

Straight and normal

$=(x_\eta / y_\eta)_\eta$



Locally straight and normal(局部平直正交)

$$\frac{dy}{dx}=const \longrightarrow \frac{dx}{dy}=const \longrightarrow \frac{dx/d\eta}{dy/d\eta}=(x_\eta/y_\eta)=const$$

Thus $$(x_\eta/y_\eta)_\eta=\frac{d}{d\eta}(x_\eta/y_\eta)=\frac{d}{d\eta}(const)\equiv 0$$

3) Summarizing: Local orthogonality leads to $\beta=0$ **,** **Local straight requires** $(x_\eta/y_\eta)_\eta=0$ **.Thus the right hand side of the above equation equals zero.**

$$\alpha[y_\eta(x_{\xi\xi}+\phi x_\xi)-x_\eta(y_{\xi\xi}+\phi y_\xi)]=0$$

Further: $$x_{\xi\xi}+\phi x_\xi=(\frac{x_\eta}{y_\eta})(y_{\xi\xi}+\phi y_\xi)$$

We are now working on the boundary with constant η **.**

Thus we have no way to calculate x_η / y_η; **In order to determine this term following transformation is made**

From $\beta = x_\xi x_\eta + y_\xi y_\eta = 0 \longrightarrow \dfrac{x_\eta}{y_\eta} = -\dfrac{y_\xi}{x_\xi}$

y_ξ / x_ξ **can be computed on the line of** $\eta = \text{cons tan t}$

Thus substituting into: $x_{\xi\xi} + \phi x_\xi = (\dfrac{x_\eta}{y_\eta})(y_{\xi\xi} + \phi y_\xi) \longrightarrow$

$$x_{\xi\xi} + \phi x_\xi = -(\frac{y_\xi}{x_\xi})(y_{\xi\xi} + \phi y_\xi) \longrightarrow$$

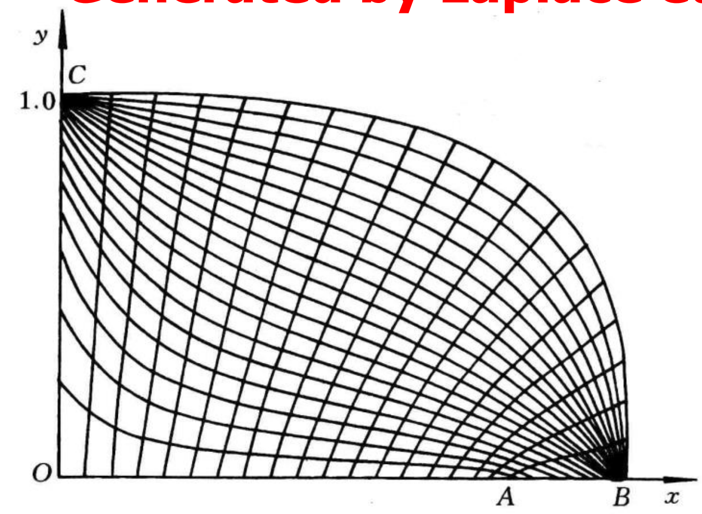
$$x_\xi(x_{\xi\xi} + \underline{\phi x_\xi}) = -y_\xi(y_{\xi\xi} + \underline{\phi y_\xi})$$

Finally: $\phi = -\dfrac{y_\xi y_{\xi\xi} + x_\xi x_{\xi\xi}}{x_\xi^2 + y_\xi^2}$ <u>(**on** $\eta = 0, \eta = 1$ **boundaries**)</u>

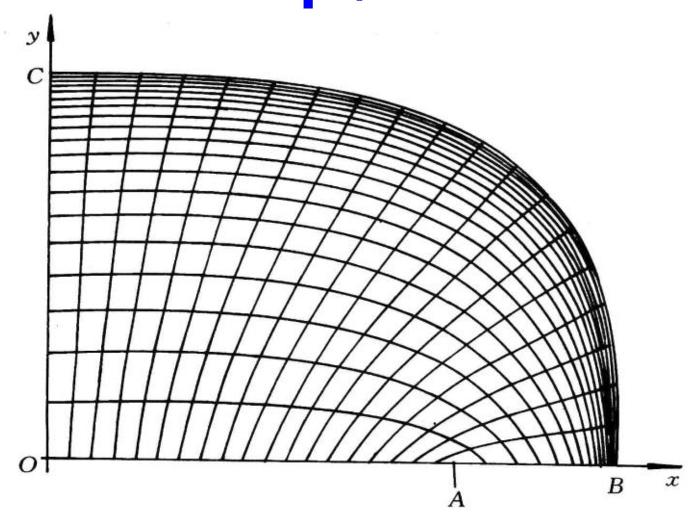
Similarly: $\psi = -\dfrac{y_\eta y_{\eta\eta} + x_\eta x_{\eta\eta}}{x_\eta^2 + y_\eta^2}$ **(On $\xi = 0, \xi = 1$ boundaries)**

Application example of Thomas－Middlecoff method

Generated by Laplace eq.



Poisson eq.＋T-M method



Thomas－Middlecoff method for determining source functions of P,Q is a good example of creative numerical method proposed by non-mathematician!

12.6 Transformation and Discretization of Governing Eq. and Boundary Conditions

12.6.1 Transformation of Governing Equation

12.6.2 Transformation of Boundary Conditions

12.6.3 Discretization in computational plane

12.6 Transformation and Discretization of Governing Eq. and Boundary Conditions

12.6.1 Transformation of Governing Equation

1.Mathematical tools used for transformation

1)Chain rule for composite function（复合函数链导法）

$$u(x,y)=u(x(\xi,\eta),y(\xi,\eta)) \quad v(x,y)=v(x(\xi,\eta),y(\xi,\eta))$$

$$\begin{vmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{vmatrix} = \begin{vmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial u}{\partial \eta} \\ \frac{\partial v}{\partial \xi} & \frac{\partial v}{\partial \eta} \end{vmatrix} \times \begin{vmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{vmatrix}$$

yielding: $$\frac{\partial u}{\partial x}=\frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial x}+\frac{\partial u}{\partial \eta}\frac{\partial \eta}{\partial x}$$

2) Derivatives of function and its inverse function(反函数)

$\xi(x,y),\eta(x,y)$ **are the inverse function of** $x(\xi,\eta),y(\xi,\eta)$

Their derivatives have following relation：

$$\xi_x=\frac{1}{J}y_\eta;\ \eta_x=-\frac{1}{J}y_\xi;\xi_y=-\frac{1}{J}x_\eta;\eta_y=\frac{1}{J}x_\xi$$

2.Results of transformation of 2-D diffusion-convection equation in physical Cartesian coordinate

$$\frac{\partial(\rho u\phi)}{\partial x}+\frac{\partial(\rho v\phi)}{\partial y}=\frac{\partial}{\partial x}(\Gamma_\phi\frac{\partial\phi}{\partial x})+\frac{\partial}{\partial y}(\Gamma_\phi\frac{\partial\phi}{\partial y})+R_\phi(x,y)$$

Results：

$$\underline{\frac{1}{J}\frac{\partial}{\partial\xi}(\rho U\phi)+\frac{1}{J}\frac{\partial}{\partial\eta}(\rho V\phi)}=\underline{\frac{1}{J}\frac{\partial}{\partial\xi}[(\frac{\Gamma_\phi}{J}(\alpha\phi_\xi-\beta\phi_\eta)]}+$$

$$\underline{\frac{1}{J}\frac{\partial}{\partial\eta}[\frac{\Gamma_\phi}{J}(-\beta\phi_\xi+\gamma\phi_\eta)]}+\underline{S_\phi(\xi,\eta)}$$

3. Explanation for results

1) Velocity U, V: $U = uy_\eta - vx_\eta, V = vx_\xi - uy_\xi$

U, V are velocities in ξ, η direction in computational plane, called contravariant velocity (逆变速度);

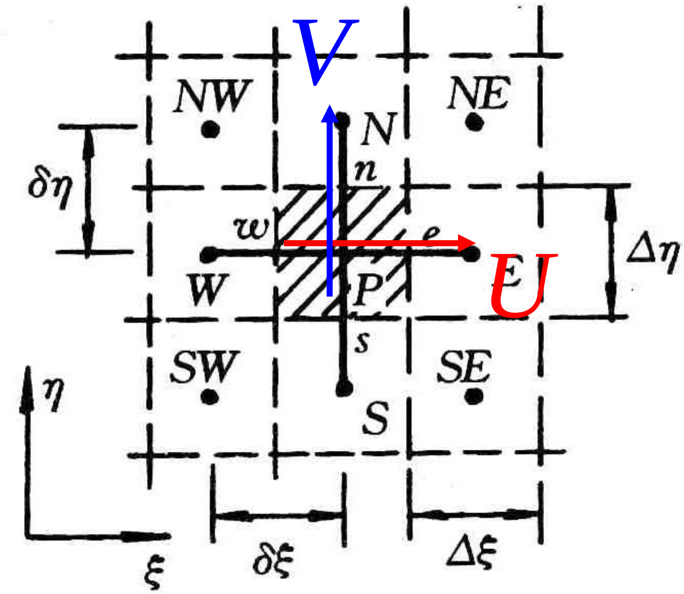
2) J: Jakobi factor, representing variation of volume during transformation

$$dV = Jd\xi d\eta d\varsigma$$

Physical space volume

Computational. space volume

Factor of volume change: Larger than 1 means volume in computational space is reduced.

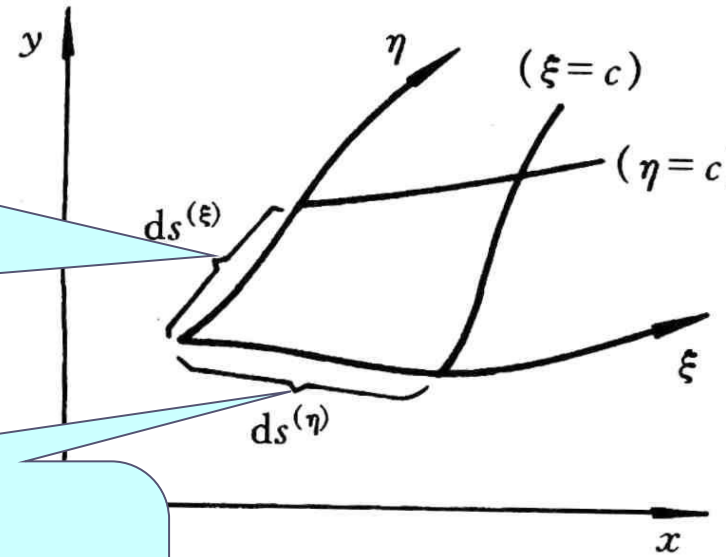


3) α, γ are metric (度规) coefficients in η, ξ direction

$\sqrt{\alpha}, \sqrt{\gamma}$ are called Lame coefficient in η, ξ direction.

$$ds^{(\xi)} = \sqrt{\alpha}\, d\eta$$

is a differential arc length in curve with constant ξ



$$ds^{(\eta)} = \sqrt{\gamma}\, d\xi$$

is a differential arc length in curve with constant η

4) β represents local orthogonality

12.6.2 Transformation of boundary condition

1.Uniform expression of B.C. in physical plane

A＝0: second kind

B＝0: first kind

$$A\phi + B\Gamma_{\phi}\frac{\partial \phi}{\partial n} = C$$

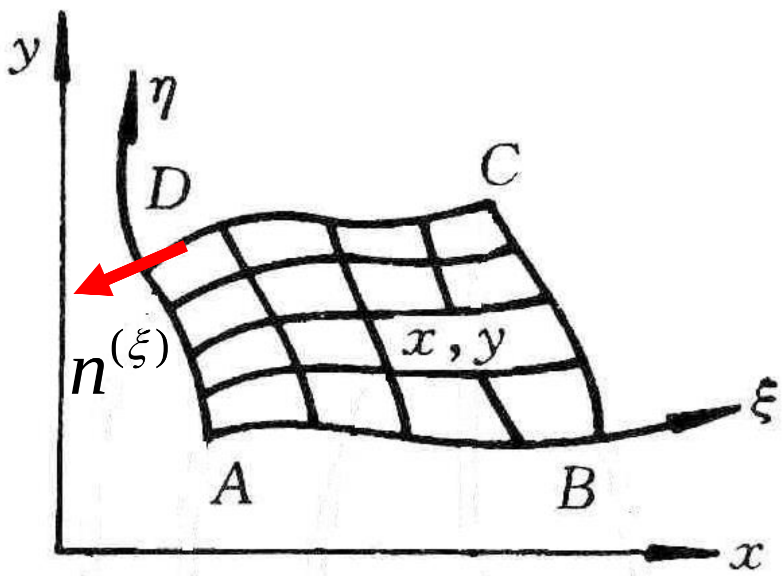
A， B are not zero: 3rd kind boundary condition

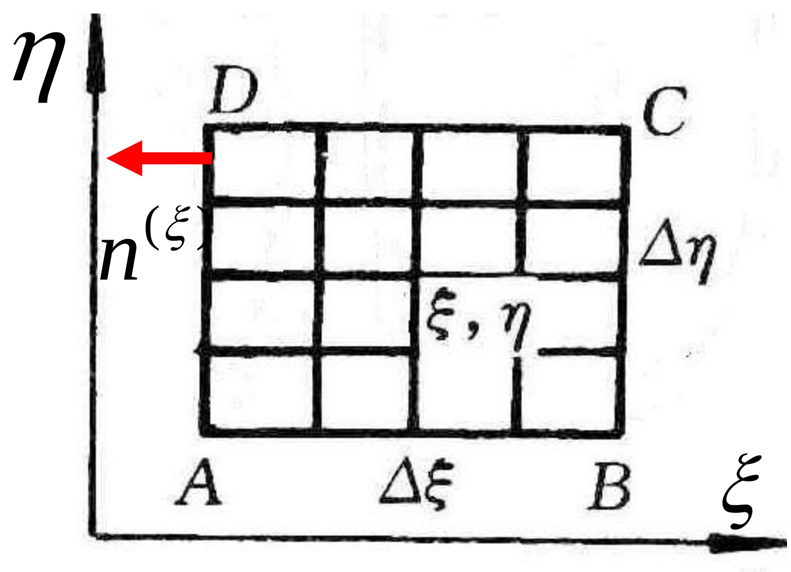
During the transformation from physical plane to computational plane

(1)The values of physical variables at correspondent positions remain unchanged

(2)Physical properties /constant remain unchanged.

What different is the derivative normal to a boundary in physical plane and in computational plane:

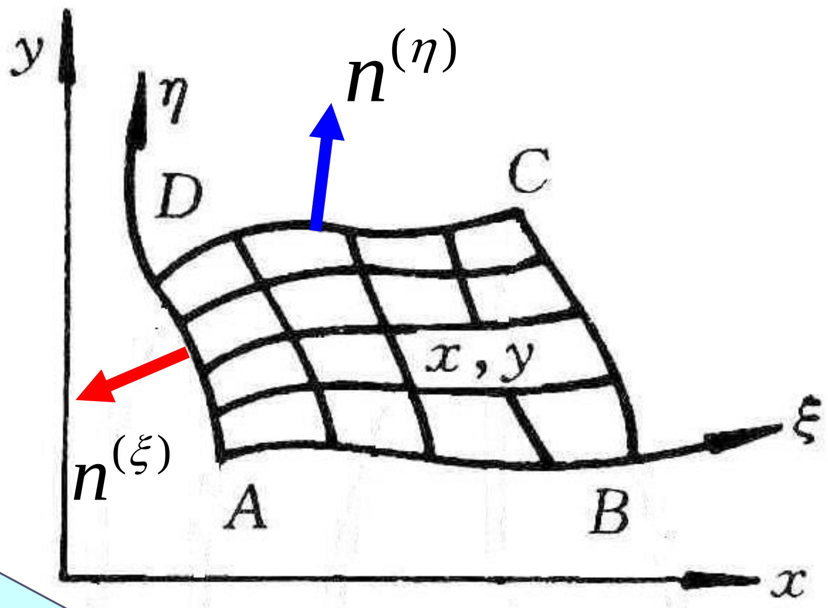




$$\left[\frac{\partial \phi}{\partial n^{(\xi)}}\right]_{Phy} \neq \left[\frac{\partial \phi}{\partial n^{(\xi)}}\right]_{Comp}$$

It can be shown that

$$\begin{cases} \dfrac{\partial \phi}{\partial n^{(\xi)}} = \dfrac{\alpha\phi_\xi - \beta\phi_\eta}{J\sqrt{\alpha}}; \\ \dfrac{\partial \phi}{\partial n^{(\eta)}} = \dfrac{\gamma\phi_\eta - \beta\phi_\xi}{J\sqrt{\varphi}} \end{cases}$$

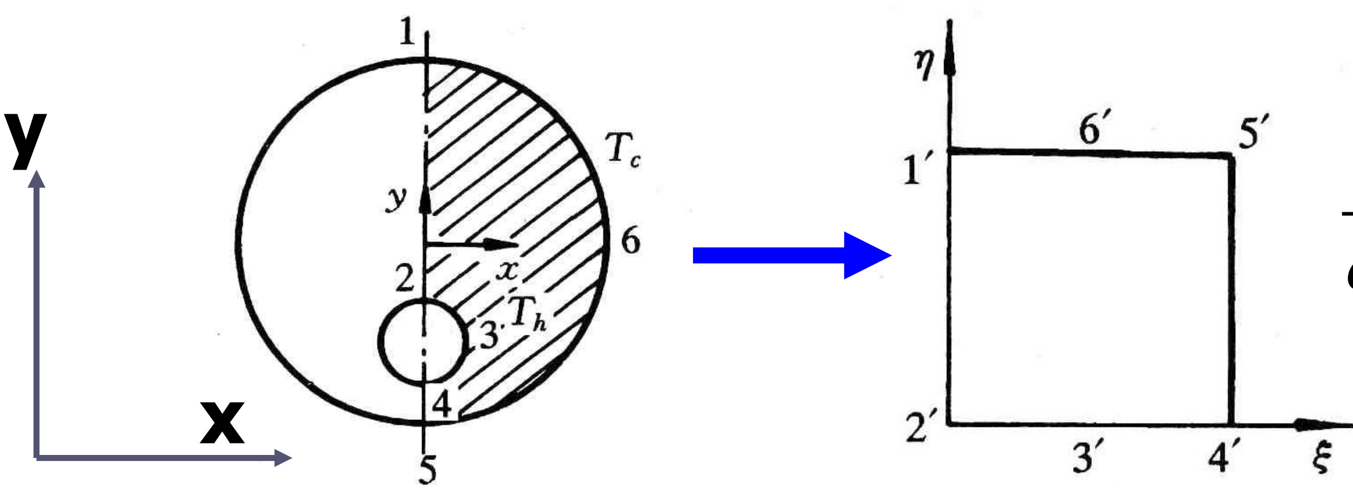


Boundary normal derivative in physical space

ϕ_ξ and ϕ_η **are boundary normal derivative in computational space**

Example of boundary condition transformation

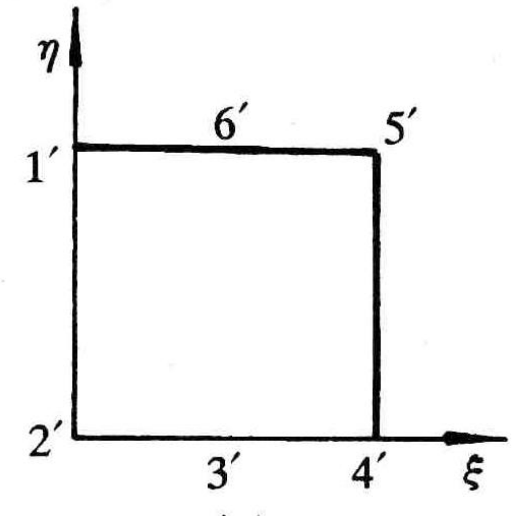
Boundary	Condition -Physical	Condition-Computational
1-2	$u=0,\dfrac{\partial v}{\partial x}=\dfrac{\partial T}{\partial x}=0$	$u=0;\alpha v_{\xi}-\beta v_{\eta}=\alpha T_{\xi}-\beta T_{\eta}=0$
2-3-4	$u=v=0,T=T_h$	$u=v=0,T=T_h$
4-5	$u=0,\dfrac{\partial v}{\partial x}=\dfrac{\partial T}{\partial x}=0$	$u=0;\alpha v_{\xi}-\beta v_{\eta}=\alpha T_{\xi}-\beta T_{\eta}=0$
5-6-1	$u=v=0,T=T_c$	$u=v=0,T=T_c$



$$\frac{\partial \phi}{\partial n^{(\xi)}}=\frac{\alpha\phi_{\xi}-\beta\phi_{\eta}}{J\sqrt{\alpha}};$$

Implementation of boundary condition at 1'-2'

$$\alpha T_\xi - \beta T_\eta = 0 \longrightarrow T_\xi = \frac{\beta T_\eta}{\alpha}$$

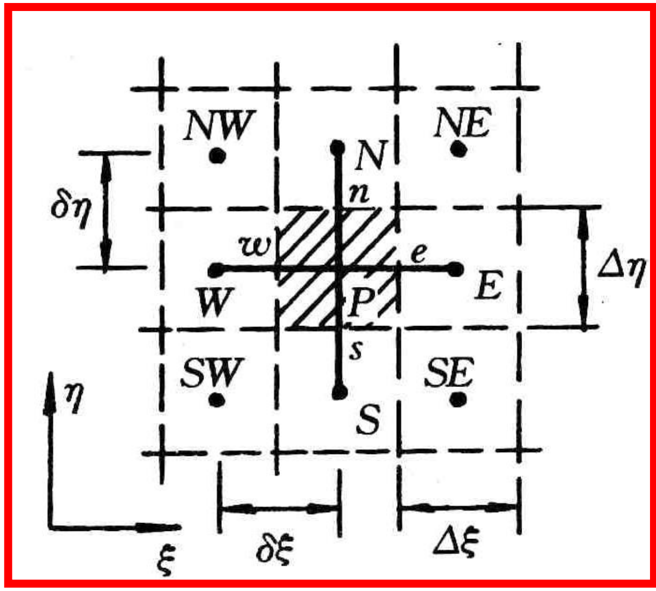


This is second kind boundary in computational plane, and can be implemented by ASTM.

12.6.3 Discretization in computational plane

1.Discretization of G.E.

Multiplying two sides of the Gov.Eqs. by J, and integrating it over a CV at staggered grid system:



$$[(\rho U\phi)_e-(\rho U\phi)_w]\Delta\eta+[(\rho V\phi)_n-(\rho V\phi)_s]\Delta\xi=$$

$$[\frac{\Gamma_\phi}{J}(\alpha\phi_\xi-\beta\phi_\eta)]_e\Delta\eta-[\frac{\Gamma_\phi}{J}(\alpha\phi_\xi-\beta\phi_\eta)]_w\Delta\eta+$$

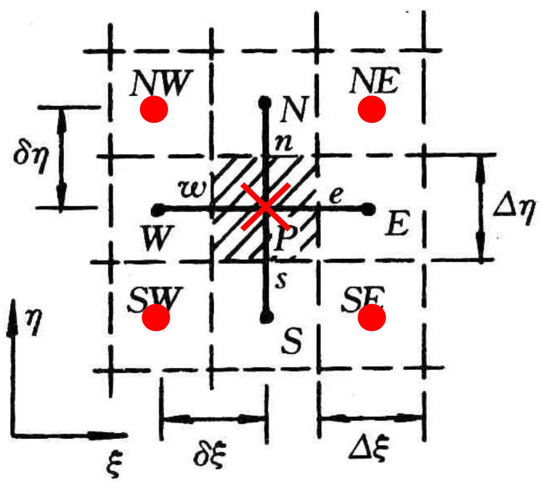
$$[\frac{\Gamma_\phi}{J}(-\beta\phi_\xi+\gamma\phi_\eta)]_n\Delta\xi-[\frac{\Gamma_\phi}{J}(-\beta\phi_\xi+\gamma\phi_\eta)]_s\Delta\xi+S\bullet J\bullet\Delta\eta\bullet\Delta\xi$$

Note：Cross derivatives(交叉导数) occurs in diffusion terms.

2) Discretization of convective term –the same as in physical space.

3) Cross derivatives in diffusion term

Say: $$(\phi_\eta)_e=\frac{(\phi_N+\phi_{NE})-(\phi_S+\phi_{SE})}{4\Delta\eta}$$



leading to 9-point scheme of 2-D case.

Putting the cross derivatives into source term, obtaining following results:

$$a_P\phi_P = a_E\phi_E + a_W\phi_W + a_W\phi_W + a_N\phi_N + b$$

$$b = S_\phi J\Delta\eta\Delta\xi - [(\underline{\frac{\Gamma_\phi}{J}\beta\phi_\eta})_w^e\Delta\eta + (\underline{\frac{\Gamma_\phi}{J}\beta\phi_\xi})_s^n\Delta\xi]$$

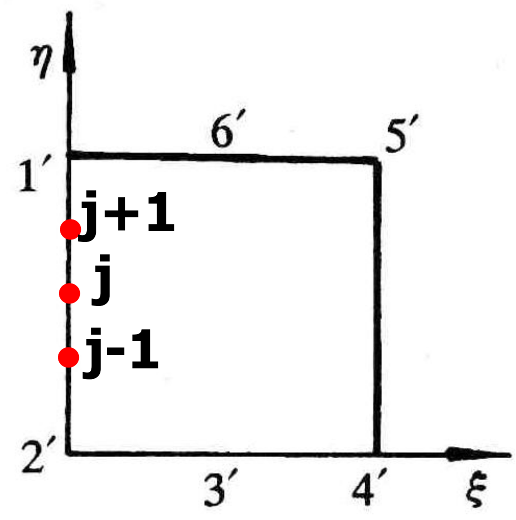
The pressure gradient term is temporary included in S_ϕ .

4.Discretization of boundary condition

The key is boundary derivative,

As shown in the above example:

$$T_\xi = \frac{\beta T_\eta}{\alpha} \longrightarrow (T_\xi)_j = \frac{\beta}{\alpha}\frac{T_{B(j+1)} - T_{B(j-1)}}{2\Delta\eta}$$



12.7 SIMPLE Algorithm in Computational Plane

12.7.1 Choice of velocity in computational space

12.7.2 Discretized momentum equation in computational plane

12.7.3 Velocity correction in computational plane

12.7.4 Pressure correction equation in computational plane

12.7.5 Solution procedure of SIMPLE in computational plane

12.7 SIMPLE Algorithm in Computational Plane

12.7.1 Choice of velocity in computational space

1. Three kinds of velocity

1) Components in physical plane (u, v)

2) Contravariant velocity (U, V) **(逆变分量)**

$$U = uy_\eta - vx_\eta, \quad V = vx_\xi - uy_\xi$$

3) Covariant velocity $(\overline{U}, \overline{V})$**(协变分量)**

$$\overline{U} = ux_\xi + vy_\xi, \quad \overline{V} = ux_\eta + vy_\eta$$

All the three kinds of velocity were adopted in refs.

According to W. Shyy（史维）： following combination can satisfy the conservation condition the best: taking u, v as solution variables and U, V as the velocity in computational plane. We will take this practice.

12.7.2 Discretized momentum equation in computational plane

1.Separating pressure gradient from source term

$$\frac{\partial p}{\partial x}=\frac{\partial p}{\partial \xi}\frac{\partial \xi}{\partial x}+\frac{\partial p}{\partial \eta}\frac{\partial \eta}{\partial x}=\frac{1}{J}\left(\frac{\partial p}{\partial \xi}y_{\eta}-\frac{\partial p}{\partial \eta}y_{\xi}\right)=\frac{1}{J}(p_{\xi}y_{\eta}-p_{\eta}y_{\xi})$$

Note： cross derivatives occur.

2. Discretized momentum equation in physical plane

$$a_e u_e = \sum a_{nb} u_{nb} + b - \Delta y \bullet \delta x \left(\frac{p_E - p_P}{\delta x}\right) = \sum a_{nb} u_{nb} + b - \Delta y \bullet \delta x \bullet p_x$$

$$u_e = \sum \left(\frac{a_{nb}}{a_e}\right) u_{nb} + \left(\frac{-\Delta y \bullet \delta x}{a_e}\right) p_x + \left(\frac{b}{a_e}\right)$$

Subscript here denotes derivative

3. Discretized u,v equations in computational plane

Mimicking the above form for ***u,v*** in physical plane for computational plane following form is taken:

$$u_P = \sum A^u_{nb} u_{nb} + (B^u p_\xi + C^u p_\eta) + D^u$$

$$v_P = \sum A^v_{nb} v_{nb} + (B^v p_\xi + C^v p_\eta) + D^v$$

1) (u_P, v_P) are the velocities at respective locations of staggered grid.

2) A,B,C,D are coefficients and constants generated during discretization.

12.7.3 Velocity correction in computational plane

1. u’,v’ equations in computational plane

From assumed *p, yielding *u**, *v**:**

$$u_P^* = \sum A_{nb}^u u_{nb}^* + (B^u p_\xi^* + C^u p_\eta^*) + D^u$$

$$v_P^* = \sum A_{nb}^v v_{nb}^* + (B^v p_\xi^* + C^v p_\eta^*) + D^v$$

The correspondent *U*,V may not satisfy mass conservation, and improvement of pressure is needed.**

Denoting pressure correction by *p’*, and the correspondent velocity corrections by *u’,v’*;

According to the SIMPLE practice, (*p+*p*'), (*u**+*u*'), and (*v**+*v*') also satisfy momentum equation:**

$$(u_P^* + u_P^{'}) = \sum A_{nb}(u_{nb}^* + u_{nb}^{'}) + [B^u(p_\xi^* + p_\xi^{'}) + C^u(p_\eta^* + p_\eta^{'})] + D^u$$

$$u_P^* = \sum A_{nb}^u u_{nb}^* + (B^u p_\xi^* + C^u p_\eta^*) + D^u$$

Subtraction of the two equations:

$$u_P^{'} = \overbrace{\sum A_{nb}^u u_{nb}^{'}}^{0} + B^u p_\xi^{'} + C^u p_\eta^{'}$$

Similarly $$v_P^{'} = \overbrace{\sum A_{nb}^v v_{nb}^{'}}^{0} + B^v p_\xi^{'} + C^v p_\eta^{'}$$

Omitting the effects of neighboring nodes:

yielding velocity correction:

$$\begin{cases} u_P^{'} = B^u p_\xi^{'} + C^u p_\eta^{'} \\ v_P^{'} = B^v p_\xi^{'} + C^v p_\eta^{'} \end{cases}$$

2. U',V' equations in computational plane

By definition: $U = uy_\eta - vx_\eta, \quad V = vx_\xi - uy_\xi$

Thus $U' = u'y_\eta - v'x_\eta = y_\eta(\underline{B^u p'_\xi} + \underline{C^u p'_\eta}) - x_\eta(\underline{B^v p'_\xi} + \underline{C^v p'_\eta})$

$$U'_P = \underline{p'_\xi}(B^u y_\eta - B^v x_\eta) + \underline{p'_\eta}\underbrace{(C^u y_\eta - C^v x_\eta)}_{0}$$

New assumption: cross derivatives in contravariant velocity are neglected

Thus: $U'_P = p'_\xi\underline{(B^u y_\eta - B^v x_\eta)} = (Bp'_\xi)_{U_P} \quad , B = B^u y_\eta - B^v x_\eta$

Similarly: $V'_P = p'_\eta(C^v x_\xi - C^u y_\xi) = (Cp'_\eta)_{V_P}$

At location of U_P

At location of V_P

12.7.4 Pressure correction equation in computational plane

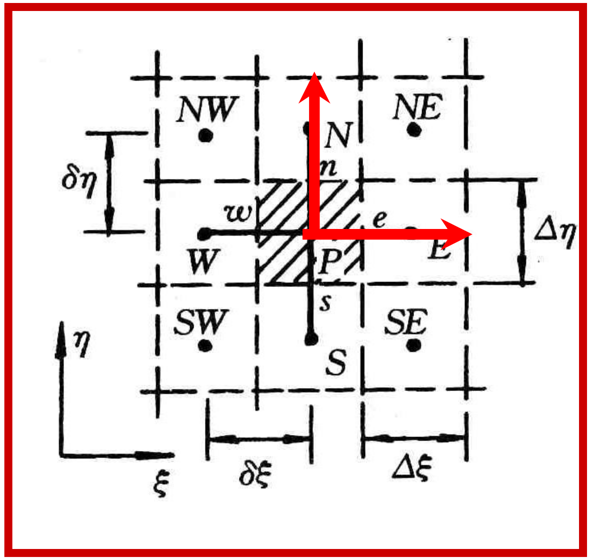
1. Discretized mass conservation in computational plane

From mass conservation in physical plane:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0$$

Its correspondent form in computational plane can be obtained:

$$\frac{\partial U}{\partial \xi}+\frac{\partial V}{\partial \eta}=0$$



Integrating over control volume P

$$(\rho U\Delta\eta)_e-(\rho U\Delta\eta)_w+(\rho V\Delta\xi)_n-(\rho V\Delta\xi)_s=0$$

2. Pressure correction equation in computational plane

Substituting $(U^*+U'),(V^*+V'),U'=Bp'_\xi,V'=Bp'_\eta$

into mass conservation eq., and re-writing in terms of *p'*:

$$A_P p_P^{'} = A_E p_E^{'} + A_W p_W^{'} + A_N p_N^{'} + A_S p_S^{'} + b$$

$$b = (\rho U^* \Delta\eta)_e - (\rho U^* \Delta\eta)_w + (\rho V^* \Delta\xi)_n - (\rho V^* \Delta\xi)_s$$

$$A_E = (\rho B \frac{\Delta\eta}{\delta\xi})_e, \quad A_W = (\rho B \frac{\Delta\eta}{\delta\xi})_w, A_N = (\rho C \frac{\Delta\xi}{\delta\eta})_n, \quad A_S = (\rho C \frac{\Delta\xi}{\delta\eta})_s$$

3. Boundary condition of pressure correction equation

Homogeneous Neumann condition:

boundary coefficient = 0

12.7.5 Solution procedure of SIMPLE in computational plane

1. Assuming velocity field of u,v ,calculating *U,V* by definition and discretization coefficients;

2. Assuming pressure field *p and solving for** (u_P^*, v_P^*) **;**

3. From u^*, v^* calculating (U_P^*, V_P^*) by definition;

4. Solving pressure correction eq., yielding p';

5. Determining revised velocities

$$u_P = u_P^* + (B^u p_\xi' + C^u p_\eta') \qquad \boxed{u_P' = B^u p_\xi' + C^u p_\eta'}$$

$$v_P = v_P^* + (B^v p_\xi' + C^v p_\eta') \qquad \boxed{v_P' = B^v p_\xi' + C^v p_\eta'}$$

$$U_P = U_P^* + (B^u y_\eta + C^u x_\eta) p_\xi' \qquad \boxed{U_P' = p_\xi' (B^u y_\eta - B^v x_\eta)}$$

$$V_P = V_P^* + (C^v x_\xi + C^v y_\xi) p_\eta' \qquad \boxed{V_P' = p_\eta' (C^v x_\xi - C^u y_\xi)}$$

$$p = p^* + \alpha_p p'$$

6.Starting next iteration with improved velocity and pressure.

12.8 Post-Process and Examples

12.8.1 Data reduction should be conducted in physical plane

12.8.2 Examples

1. Example 1—Natural convection in a circle with hexagon (六边形)
2. Example 2—Forced flow over a bank of tilted plates
3. Example 3—Periodic forced convection in a duct with roughness elements
4. Example 4—Periodic forced convection in a wavy channel

12.8 Post-Process and Examples

12.8.1 Data reduction should be conducted in physical plane

Data reduction (post process, 后处理) should be conducted for the solutions in physical plane.

The results in computational plane can not be directly adopted for data reduction by using definition in physical plane.

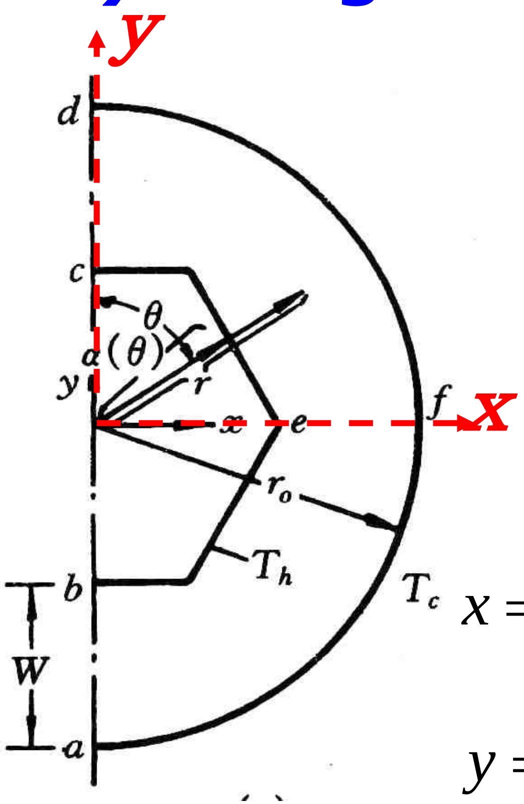
For example，the volume of a control volume is：

$\Delta V = J d\xi d\eta d\varsigma$ **rather than:** $d\xi d\eta d\varsigma$

12.8.2 Four examples

1. Example 1—Natural convection in a circle with an inner hexagon(六边形)

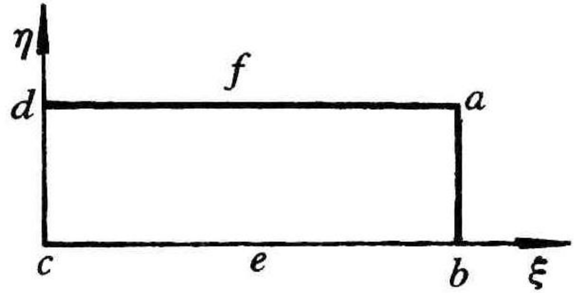
1) Grid generation—algebraic method



$$\xi = \theta$$

$$\eta = \frac{r - a(\theta)}{r_0 - a(\theta)}$$

(Polar coordinate)



$$x = [a(\xi) + \eta[r_0 - a(\xi)]\cos(\frac{\pi}{2} - \xi)$$

$$y = [a(\xi) + \eta[r_0 - a(\xi)]\sin(\frac{\pi}{2} - \xi)$$

(Cartesian coordinate)

2) Local Nusselt on inner surface

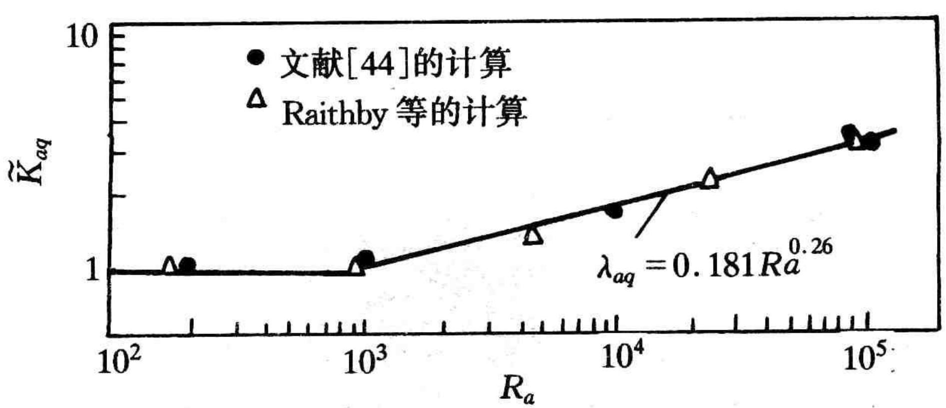
$$Nu_i=\frac{h_iW}{\lambda}=\frac{W}{\lambda}[-\lambda(\frac{\partial T}{\partial n})_i\frac{1}{T_h-T_c}]=-[\frac{\partial(\frac{T-T_c}{T_h-T_c})}{\partial(\frac{n}{W})^{(\eta)}}]_i=-[\frac{\partial\Theta}{\partial n^{(\eta)}}]_i=-[\frac{\gamma\Theta_\eta-\beta\Theta_\xi}{J\sqrt{\gamma}}]_i$$

On inner surface $\eta=0,\Theta)_{\eta=0}\equiv 1$

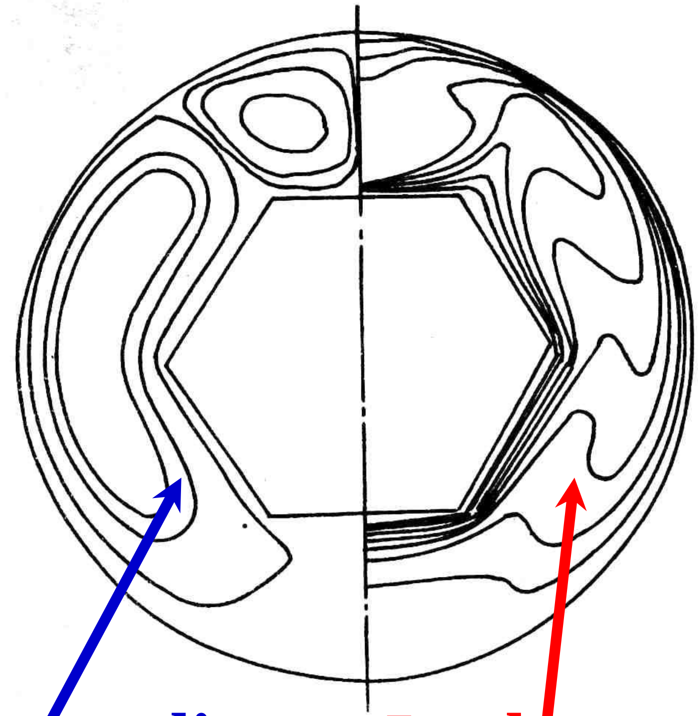
$$\Theta_\xi=\frac{\partial\Theta}{\partial\xi})_{\eta=0}=0$$

$$\boxed{Nu_i=-(\frac{\gamma\Theta_\eta}{J\sqrt{\gamma}})_i}$$

3) Partial results



Ra=9.2×10^4

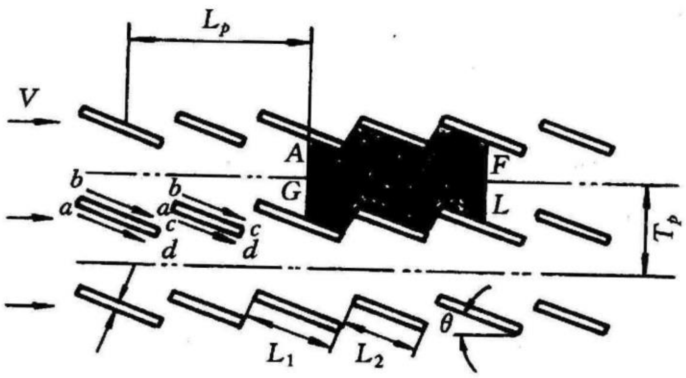


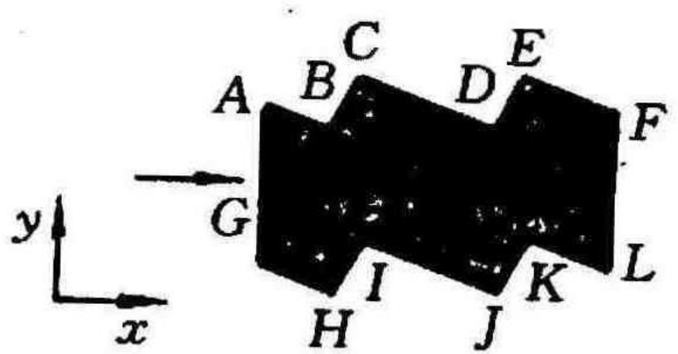
Stream lines   Isotherms

Zhang H L et al. Journal of Thermal Science, 1992, 1(4):249-258

2. Example 2—Forced flow over a bank of tilted plates

1) Grid generation—algebraic method

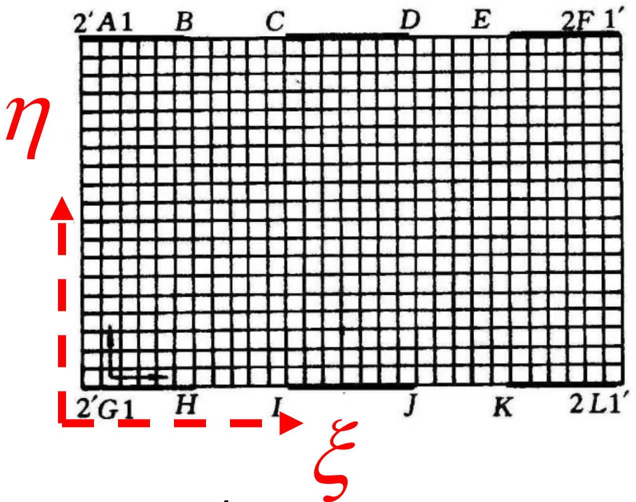


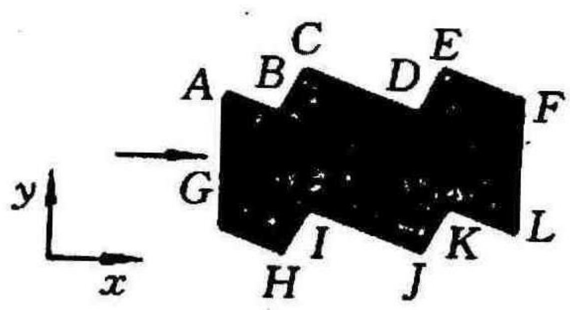


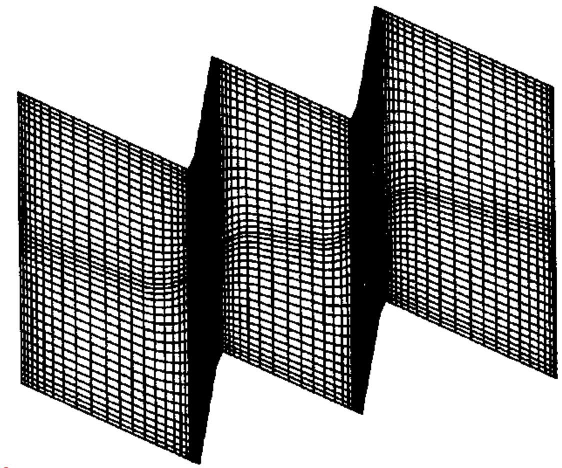
Data reduction is conducted for one cycle:

A-G-H-I-J-K-L-F-E-D-C-B-A

2) Calculation procedure







$$\boxed{dy = ds^{(\xi)} = d\eta\sqrt{\alpha}}$$

$$T_b)_{\overline{AG}} = \frac{\int_G^A T(x,y)u(x,y)dy}{\int_G^A u(x,y)dy} \longrightarrow \frac{\int_{\eta_b}^{\eta_t} T(\xi,\eta)u(\xi,\eta)\sqrt{\alpha}\,d\eta}{\int_{\eta_b}^{\eta_t} u(\xi,\eta)\sqrt{\alpha}\,d\eta}$$

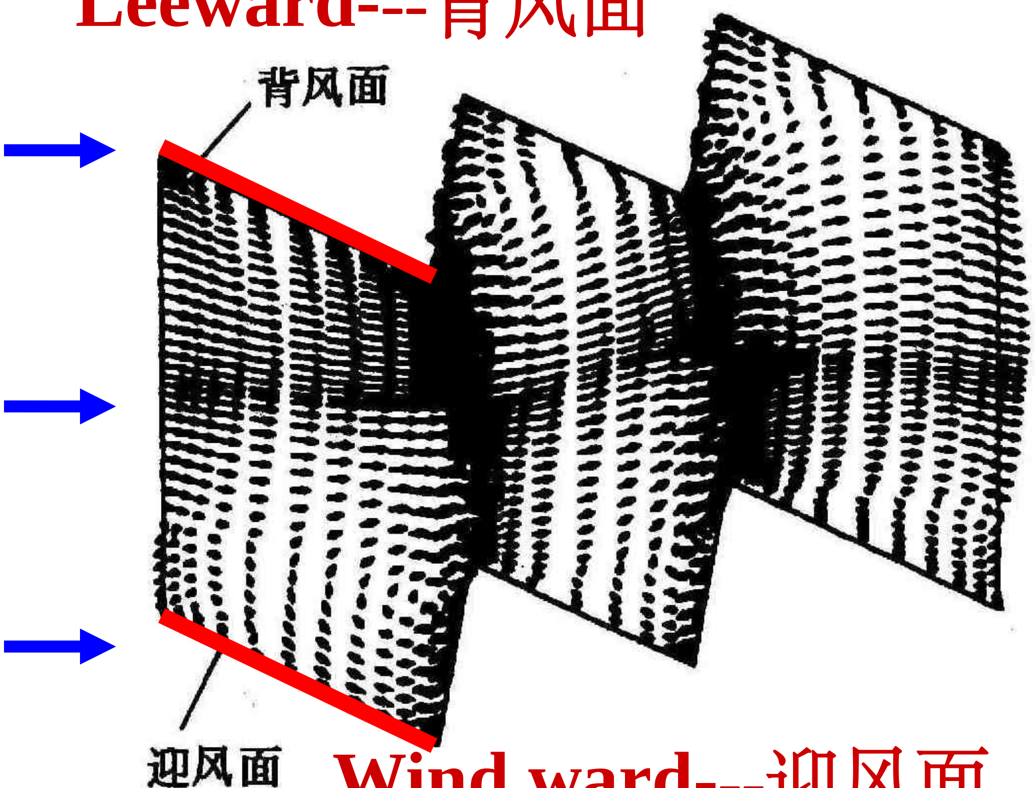
$$q_{A-F} = \frac{\int_{\xi_A}^{\xi_B} q^{(\eta_t)}\sqrt{\gamma}\,d\xi + \int_{\xi_C}^{\xi_D} q^{(\eta_t)}\sqrt{\gamma}\,d\xi + \int_{\xi_E}^{\xi_F} q^{(\eta_t)}\sqrt{\gamma}\,d\xi}{\int_{\xi_A}^{\xi_B} \sqrt{\gamma}\,d\xi + \int_{\xi_C}^{\xi_D} \sqrt{\gamma}\,d\xi + \int_{\xi_E}^{\xi_F} \sqrt{\gamma}\,d\xi}$$

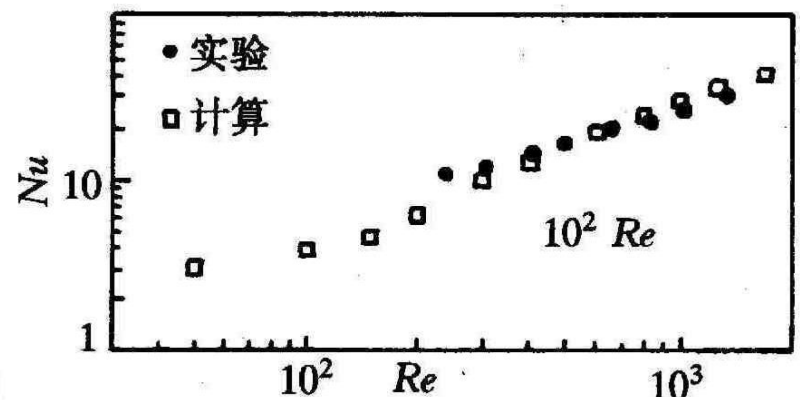
$$\boxed{ds^{(\eta)} = d\xi\sqrt{\gamma}}$$

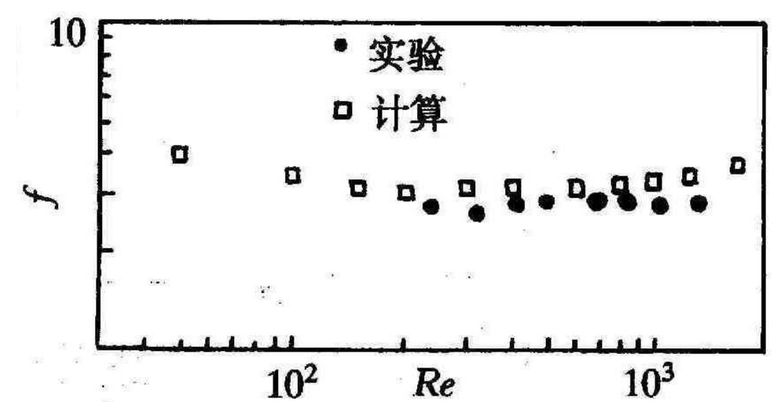
Local heat flux calculation should be conducted as shown in example 1.

3) Partial results

Leeward---背风面





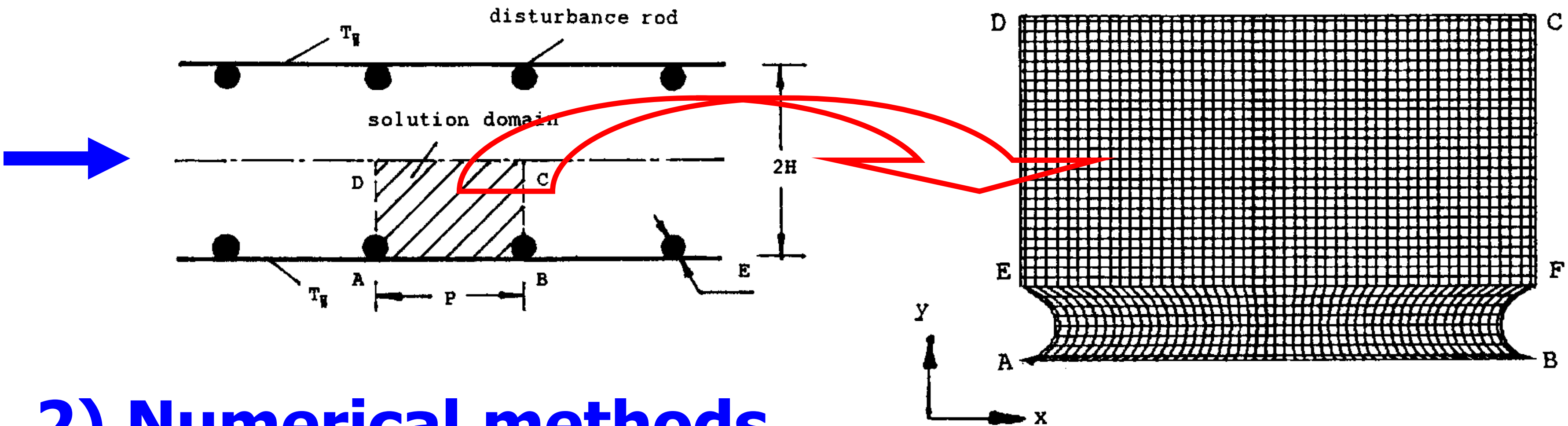


Wind ward---迎风面

Wang L B, et al. ASME Journal of Heat Transfer, 1998, 120:991-998

3. Example 3—Periodic forced convection in a duct with roughness elements

1) Grid generation—Boundary normalization



2) Numerical methods

(1) Steady vs. unsteady—Unsteady governing equation is used to get a steady solution for the case of (H/E=5, P/E=20,Re=700). The results are compared with those from steady equation. The differences are small: Nu-3％， f –less than 1％. Thus steady eq. is used.

(1) Scheme of convection term — PLS was used. Reviewer required : it should be shown that false diffusion effect could be neglected. Simulation with CD was conducted and comparison was made.

Table I. Comparison of results using PLDS and CDS

	Re	50	100	200	400	700
Nu	PLDS	7.811	8.166	8.988	10.648	12.776
	CDS	7.811	8.172	8.925	10.354	12.994
f	PLDS	2.3980	1.2197	0.6319	0.3352	0.1999
	CDS	2.3980	1.2198	0.6298	0.3329	0.2089

3) Partial results

Yuan Z X, et al. Int Journal Numerical Methods in Fluids, 1998, 28:1371-1378

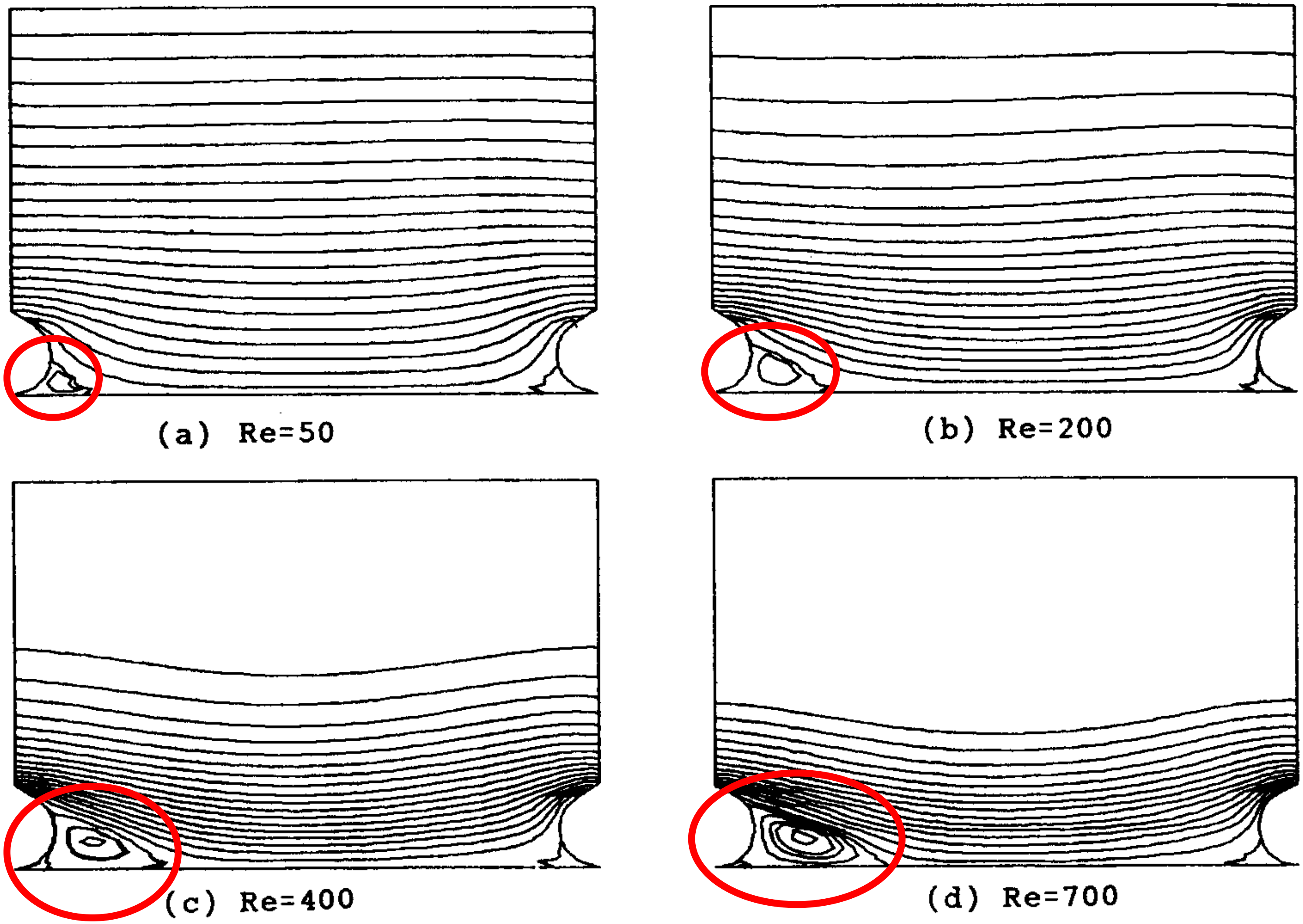
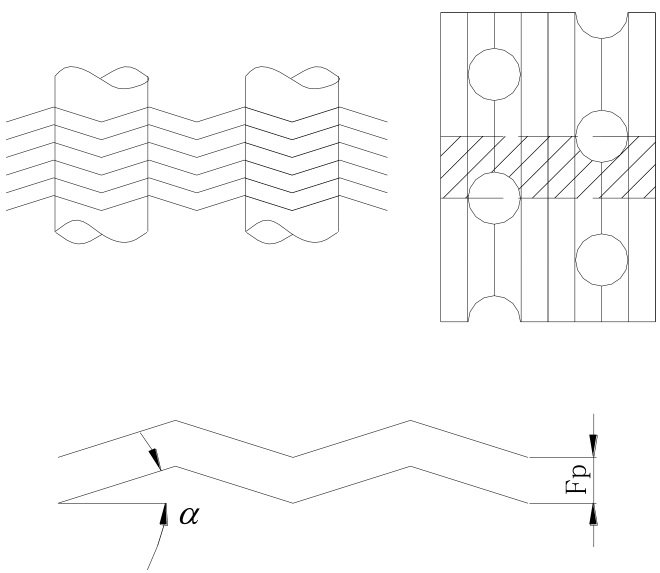
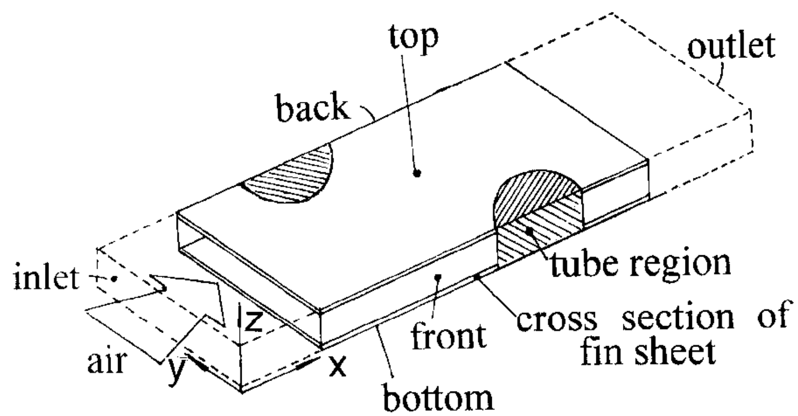


Figure 4. Flow patterns at different Reynolds numbers ($H/E = 5$, $P/E = 7.5$). (a) $Re = 50$; (b) $Re = 200$; (c) $Re = 400$; (d) $Re = 700$.

4. Example 4—Periodic forced convection in a wavy channel

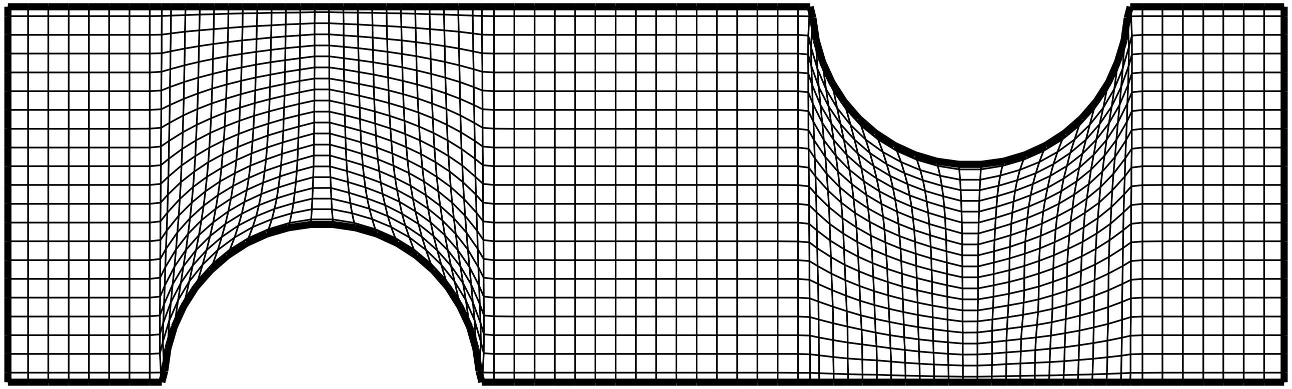
1) Grid generation－ (Block structured＋3D Poisson)





(Taking plain channel as an example)

$$\begin{cases} \alpha_{11}x_{\xi\xi}+\alpha_{22}x_{\eta\eta}+\alpha_{33}x_{\zeta\zeta}+2\alpha_{12}x_{\xi\eta}+2\alpha_{13}x_{\xi\zeta}+2\alpha_{23}x_{\eta\zeta}+J^2(Px_\xi+Qx_\eta+Rx_\zeta)=0 \\ \alpha_{11}y_{\xi\xi}+\alpha_{22}y_{\eta\eta}+\alpha_{33}y_{\zeta\zeta}+2\alpha_{12}y_{\xi\eta}+2\alpha_{13}y_{\xi\zeta}+2\alpha_{23}y_{\eta\zeta}+J^2(Py_\xi+Qy_\eta+Ry_\zeta)=0 \\ \alpha_{11}z_{\xi\xi}+\alpha_{22}z_{\eta\eta}+\alpha_{33}z_{\zeta\zeta}+2\alpha_{12}z_{\xi\eta}+2\alpha_{13}z_{\xi\zeta}+2\alpha_{23}z_{\eta\zeta}+J^2(Pz_\xi+Qz_\eta+Rz_\zeta)=0 \end{cases}$$



2) Grid-independence examonation

One row

$102(x)\times 22(y)\times 10(z)$

Two-row

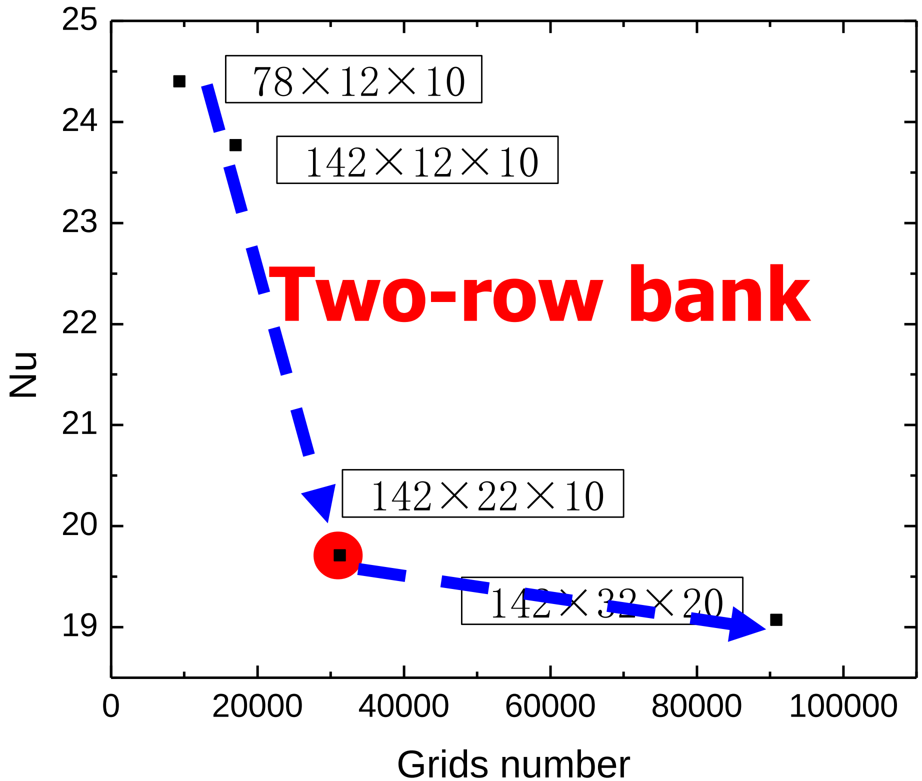
$142\times 22\times 10$

Three-row

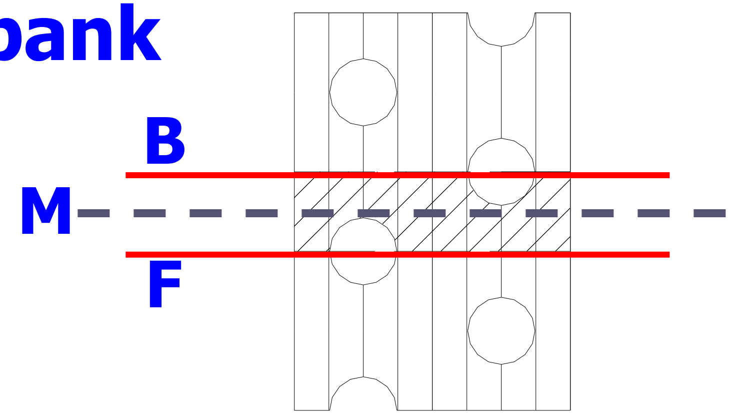
$182\times 22\times 10$

Four-row

$192\times 22\times 10$

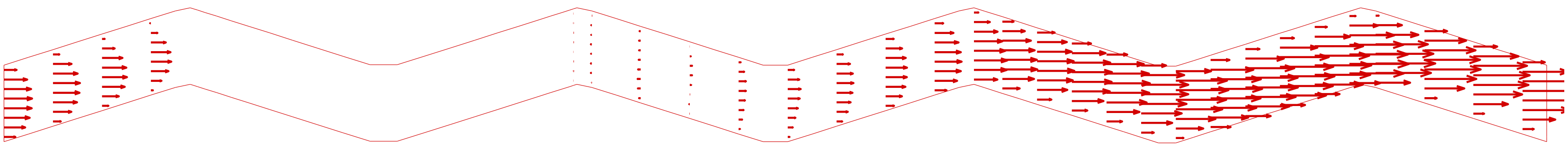


3) Partial results of two-row bank

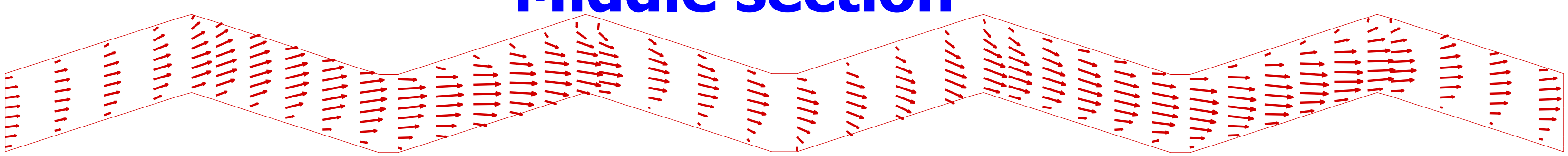


Velocity distributions of three sections

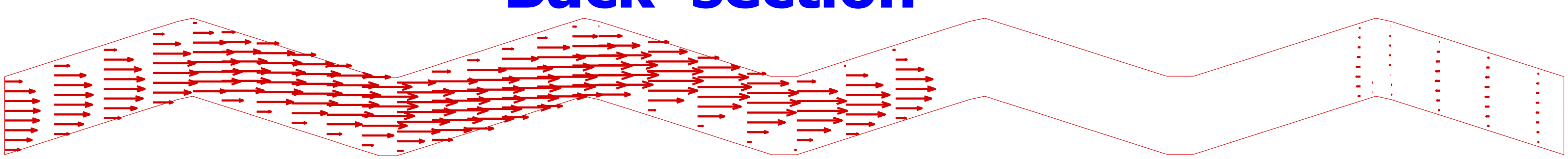
Front section



Middle section



Back section



Tao Y B, et al. Int Journal Heat Mass Transfer, 2007, 50:1163-1175

Key to three difficult problems

Problem # 3-4

There is a heat exchanger pipe with fully developed velocity field, where temperature field is described by the following equation

$$u\frac{\partial T}{\partial x}=\frac{\nu}{P_r}\frac{\partial^2 T}{\partial y^2} \rightarrow \frac{\partial T}{\partial x}=(u\frac{\nu}{P_r})\frac{\partial^2 T}{\partial y^2}$$

Using explicit format for discretization of the given equation and find out the stability condition.

Key: Regarding x as the one-way coordinate like time!

Problem # 5-9

Define the 3rd-order upwind scheme by interface interpolation method and verify the consistence with derivative definition.

From derivative definition:

$$u\frac{\partial \phi}{\partial x})_i = \frac{u_i}{6\Delta x}(2\phi_{i+1} + 3\phi_i - 6\phi_{i-1} + \phi_{i-2}), u > 0$$

$$u\frac{\partial \phi}{\partial x})_i = \frac{u_i}{6\Delta x}(2\phi_{i-1} + 3\phi_i - 6\phi_{i+1} + \phi_{i+2}), u < 0$$

$$\phi_e = \begin{cases} \dfrac{\phi_P + \phi_E}{2} - \dfrac{\phi_E - 2\phi_P + \phi_W}{6}, u > 0 \\ \dfrac{\phi_P + \phi_E}{2} - \dfrac{\phi_P - 2\phi_E + \phi_{EE}}{6}, u < 0 \end{cases}$$

Mimicking QUICK scheme, set:

$$\varphi_e = \begin{cases} \dfrac{\varphi_P + \varphi_E}{2} - \dfrac{\varphi_E - 2\varphi_P + \varphi_W}{a}, u > 0 \\ \dfrac{\varphi_P + \varphi_E}{2} - \dfrac{\varphi_P - 2\varphi_E + \varphi_{EE}}{a}, u < 0 \end{cases}$$

where coefficient *a* is to be determined. First work for

***u*>0**

$$\frac{1}{\Delta x}\int_e^w \left(\frac{\partial \phi}{\partial x}\right) dx = \frac{\phi_e - \phi_w}{\Delta x} \overset{u>0}{=}$$

$$\frac{\dfrac{\phi_E + \phi_P}{2} - \dfrac{\phi_W - 2\phi_P + \phi_E}{a} - \dfrac{\phi_P + \phi_W}{2} + \dfrac{\phi_{WW} - 2\phi_W + \phi_P}{a}}{\Delta x}$$

Compared with the derivative definition with u>0

yielding: a=6

Substituting *a*=6 to interface definiton for u<0 set up above, yielding:

$$\left.\frac{\partial \phi}{\partial x}\right)_i = \frac{1}{6\Delta x}(2\phi_{i-1} + 3\phi_i - 6\phi_{i+1} + \phi_{i+2}), \; u < 0$$

Problem # 6-1

As mentioned in section 6.1, the problem of segregated algorithm for fluid flow, there is no independent governing equation for pressure. In order to deal with the problem of the coupling between the pressure and velocity, SIMPLE and a series of algorithms are introduced. But, on the other hand, the pressure Poisson equation can be derived from the momentum equation and continuity equation, for example, as shown below is the equation of two-dimensional rectangular coordinates for incompressible fluid:

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=2\left[\left(\frac{\partial u}{\partial x}\right)\left(\frac{\partial v}{\partial y}\right)-\left(\frac{\partial v}{\partial x}\right)\left(\frac{\partial u}{\partial y}\right)\right]$$

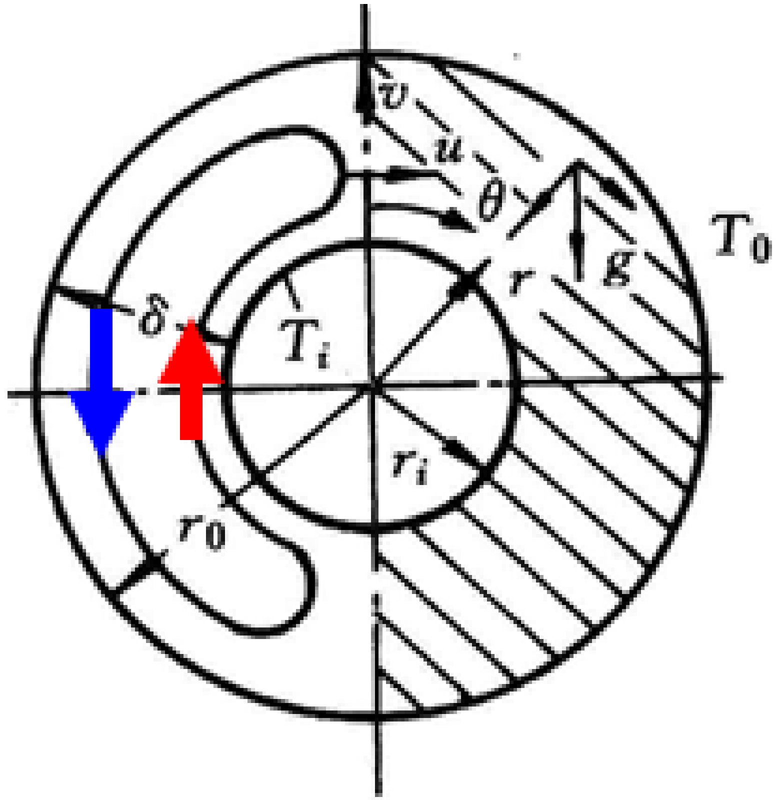
Somebody thinks that we can get the pressure equation and momentum equation simultaneous to solve the flow, namely, solving the u equation, the v equation and pressure equation in turn (u, v are known, and can be the source term of the pressure equation). This is the one iteration of the separation method, thus, without using SIMPLE algorithm. Try to derive the pressure Poisson equation and comment on the view.

No! In such a method we have no way to guarantee mass conservation, while when the Poisson Eq. of pressure is derived we have several times adopted the mass conservation condition.

Computer-Aided Project of Numerical Heat Transfer
Xi'an Jiaotong University, 2016-12-14

1. Project formulation

For air natural convection within an annular space as shown in Fig. 1 , following conditions are given: $\delta / r_o = 0.4$, flow is laminar and the average air temperature is 50 $^\circ C$. For $\mathrm{Ra} = g\beta\Delta T\delta^3 \nu / a^2 = 10^2, 10^3, 10^4, 10^5$, determine the relative thermal conductivity: $\lambda_{\mathrm{eq}} / \lambda_{air}$ .The temperature difference between inner wall and outer wall is not large, so the Boussinesq assumption can be adopted. By using Tecplot or other software, display the isotherms and streamlines and the variation of $\lambda_{\mathrm{eq}} / \lambda_{air}$ vs. Ra.



$(T_i > T_0)$

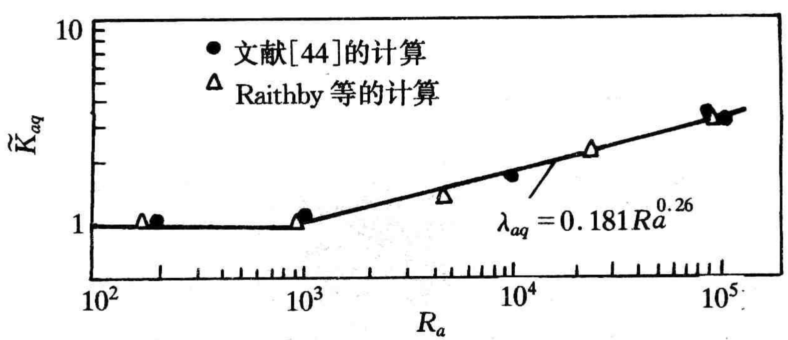
$$\delta / r_o = 0.4$$

$$\mathrm{Ra} = g\beta\Delta T\delta^3 \nu / a^2$$

$$10^2, 10^3, 10^4, 10^5$$

Air average temperatyure

$$50^\circ C$$



Ra= 9.2×10^4

2. Suggestions and Requirements

1） Considering the symmetry of the geometry, only half of the structure should be simulated.

2） The solution should be grid-independent.

3） The project report should be written in the format of the Journal of Xi'an Jiaotong University. Both Chinese and English can be accepted.

4） It is encouraged to use the teaching code, yet commercial software may also be used.

5） When the teaching code is adopted, please submit in the USER part developed by yourself for solving the problem.

6） The project report should be due in before April 30, 2017 to room 204 of East 3rd Building.

同舟共济
渡彼岸!

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