Numerical Heat Transfer
(数值传热学)
Chapter 10 Grid Generation Techniques for and Flow Simulation in irregular Regions



Instructor CHEN, Li

Key Laboratory of Thermo-Fluid Science & Engineering
Int. Joint Research Laboratory of Thermal Science & Engineering
Xi'an Jiaotong University
Innovative Harbor of West China, Xian

2026-Nov.-09~10

Contents

10.1 Grid Classification and Methods for Slightly Irregular Domains

10.2 Introduction to Body-Fitted Coordinates (BFC, 适体坐标系)

10.3 Discretization and Solution of Convection–Diffusion Problems in Body-Fitted Coordinates

10.1 Grid Classification and Methods for Slightly Irregular Domains

10.1.1 Classification of Grids

10.1.2 Methods for Treating Slightly Irregular Domains Using Orthogonal Curvilinear Coordinates

1) Stair-step (阶梯) boundary approximation to the real boundary

2) Special orthogonal curvilinear (正交曲线) coordinate systems

3) Block-structured (块结构化网格) grids

10.1 Grid Classification and Methods for Slightly Irregular Domains

10.1.1 Classification of grids

Grids available in the literature can generally be classified as that in Figure 10-1. Broadly, they fall into two major categories: **structured grids** and **unstructured grids**.

Structured grid (结构化网格): characterized by fixed, implicit connectivity （连接关系） between each node and its neighbours; no extra data is required to store and identify the neighbour relationships.

Unstructured grid (非结构化网格) : requires explicit storage of connectivity information. Although generating unstructured grids demands more computational effort, they have gained widespread adoption owing to their excellent capability to accommodate （适应） irregular geometries.

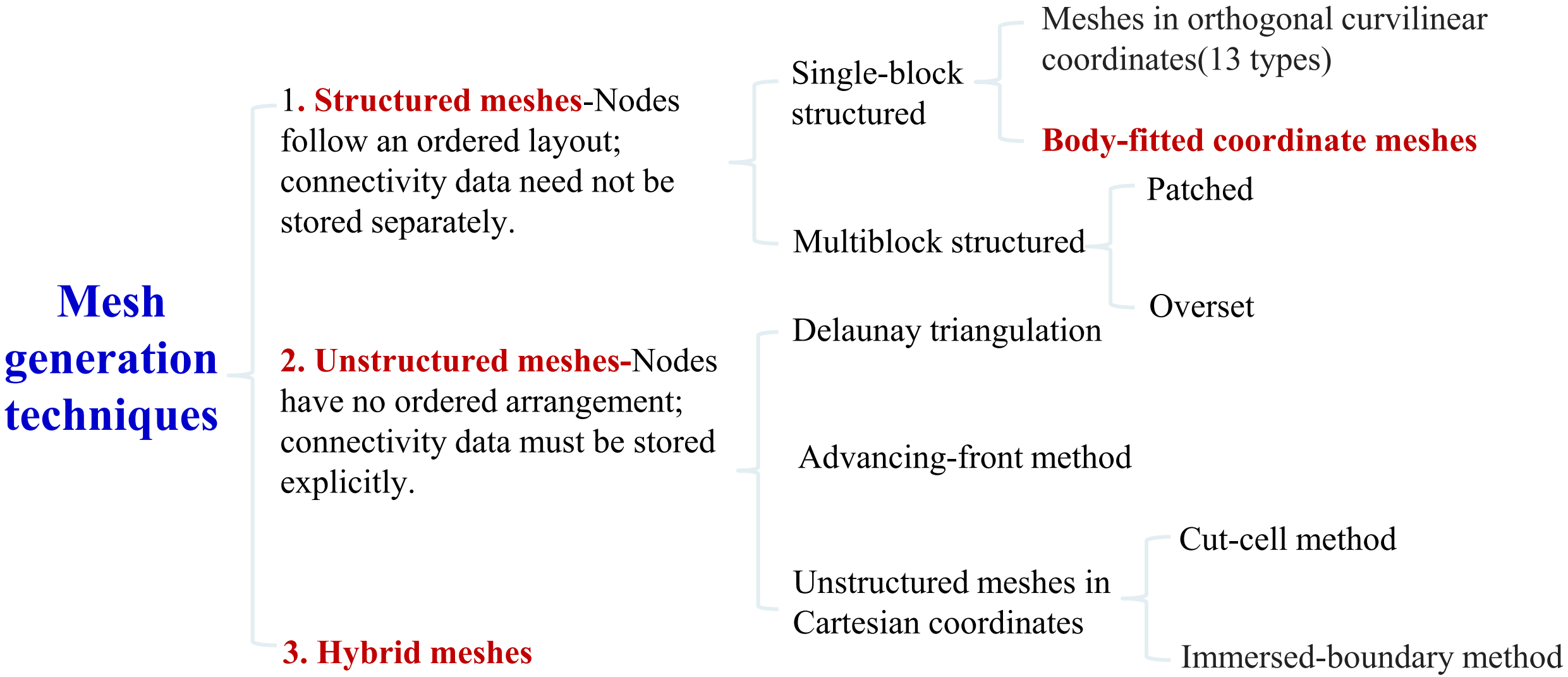
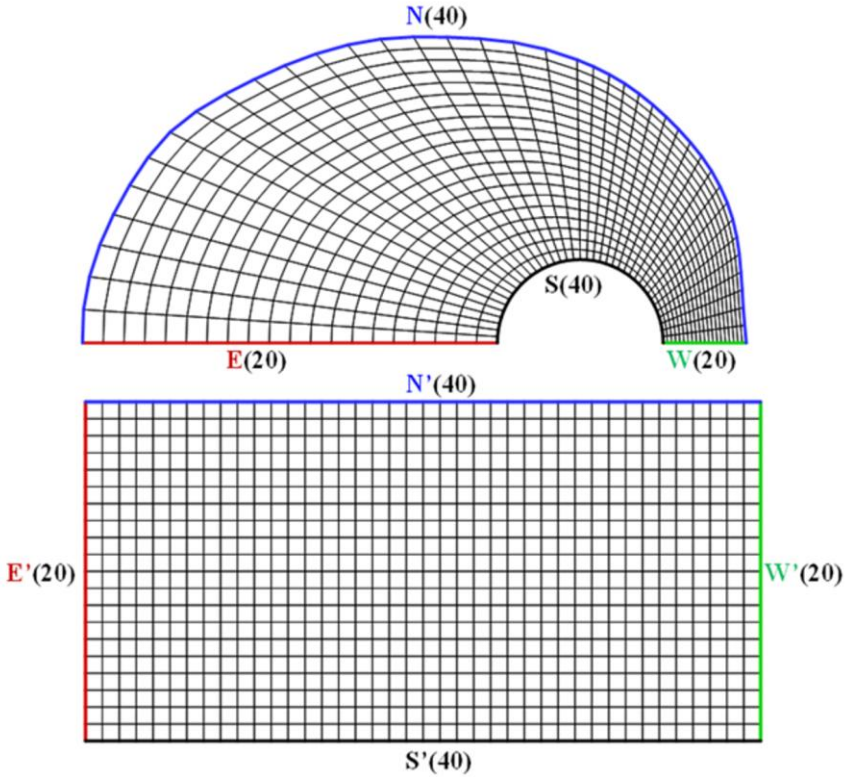
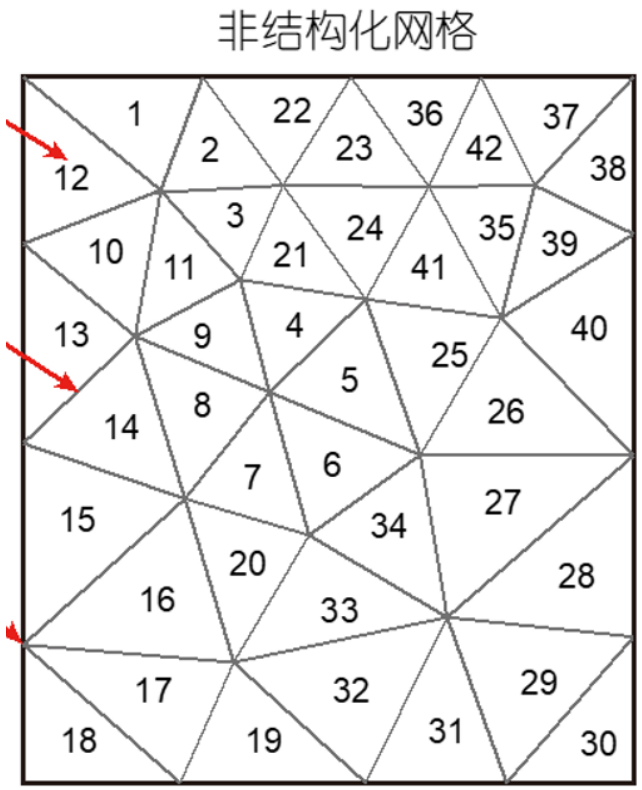


Figure 10–1. Classification of Meshes

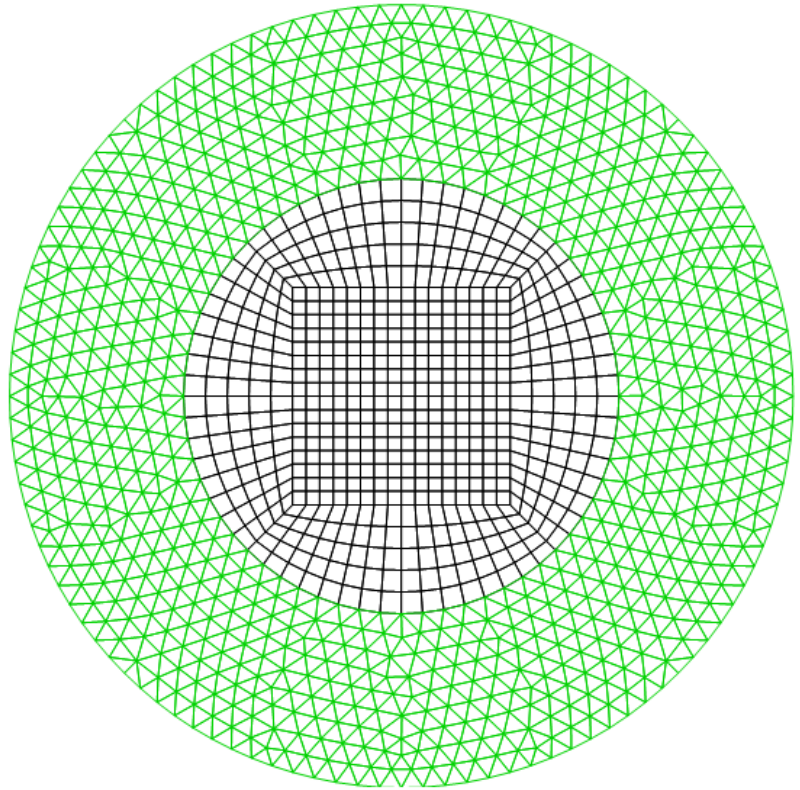


节点排列有序，节点间的联系信息不必专门存储。

1. **Structured meshes**-Nodes follow an ordered layout; connectivity data need not be stored separately.



2. Unstructured meshes-Nodes have no ordered arrangement; connectivity data must be stored explicitly.

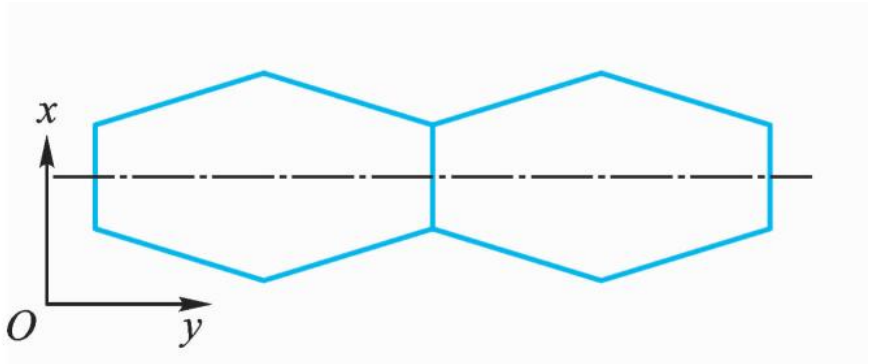


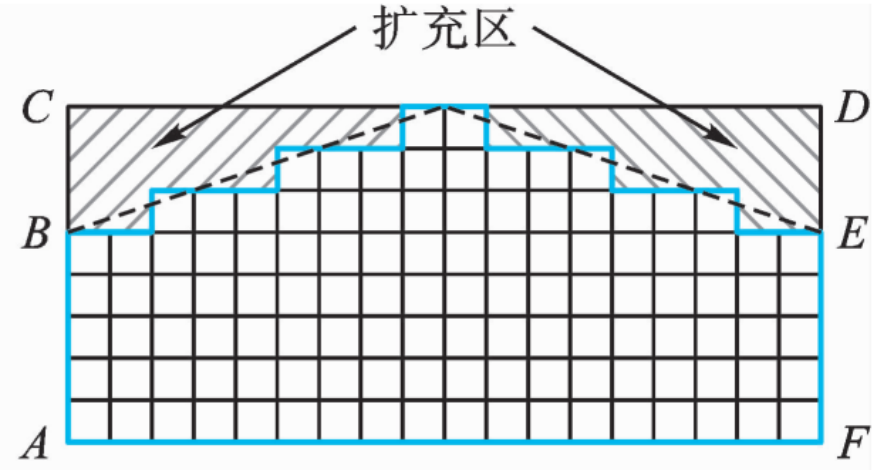
3. Hybrid meshes

10.1.2 Methods for Treating slightly Irregular Domains Using Orthogonal Curvilinear Coordinates

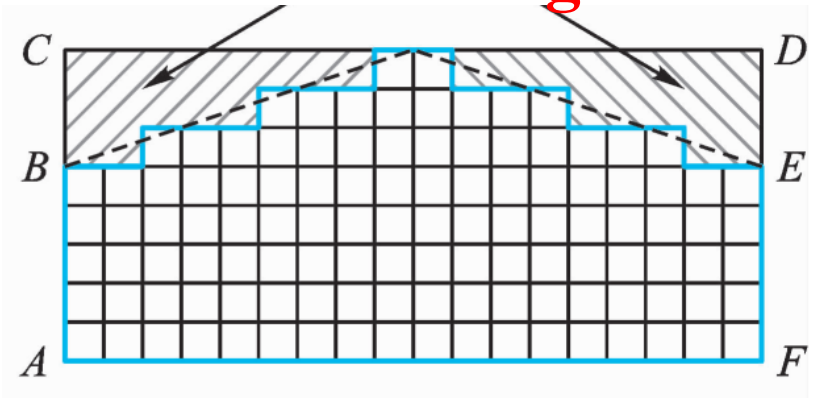
1) Stair-step (阶梯) boundary approximation to the real boundary

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 rectangular coordinate (直角坐标) within the extended domain (扩充区域).





Extended region



(1) Flow 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 （无限大粘度） in the extended region

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

(c) Set interface diffusivity by harmonic mean

(2) Heat transfer simulation

(a) First kind boundary condition with uniform temperature: The same as for velocity: boundary temperatures are given, and in the extended region a very large thermal conductivity (**无限大导热系数**) is set

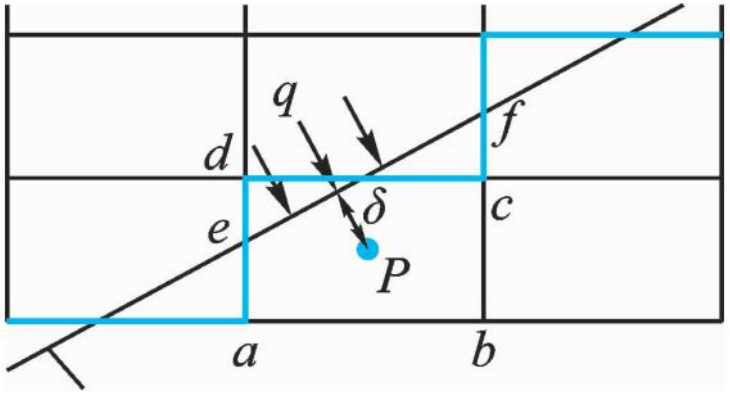
$$\lambda = 10^{25} \sim 10^{30}$$

(b) Second kind boundary condition by ASTM c

q: specified boundary heat flux distribution

(i) For CV. *P* adding additional source term:

$$S_{c,ad} = \frac{q \bullet \overline{ef}}{\Delta V_P}$$



True boundary

(ii) Set λ=0 for the extended region to avoid outward heat transfer.

(c) Third kind boundary conditions by ASTM (附加源项法，Chapter 3.3.2)

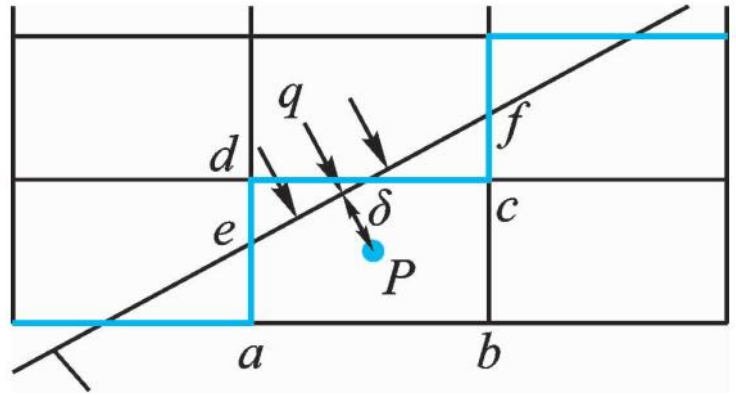
Specified external convective heat transfer coefficient and fluid temperature, h and T_f,

(i) For CV. P adding additional source term:

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

$$q = h(T - T_{\text{ref}})$$

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

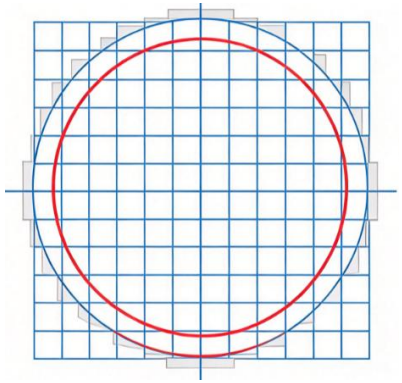


True boundary

(ii) Set λ=0 for the extended region to avoid outward heat transfer.

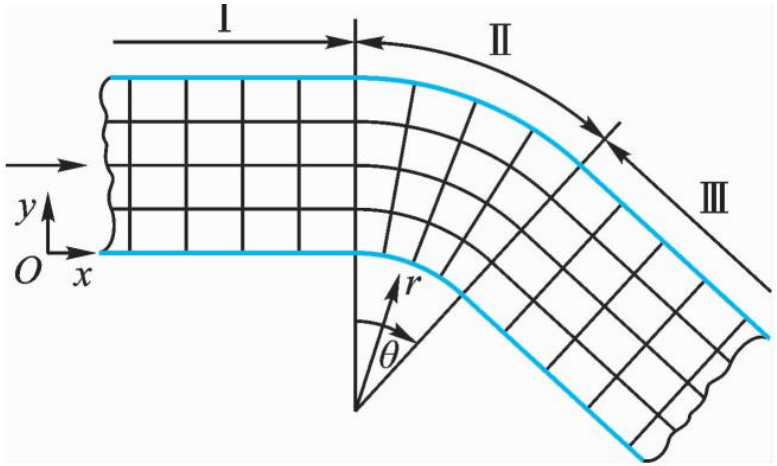
2) Special orthogonal curvilinear (正交曲线) coordinate systems

For simulating physical problems, the most desirable coordinate system is the one in which the boundaries of the solution domain are parallel to the coordinate axes（求解区域边界与网格线一一平行）.



3) Block-structured (块结构化网格) grids

The entire domain is divided into several blocks, for each block the most suitable 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）.



10.2 Introduction to Body-Fitted Coordinates

10.2.1 Basic Idea of Body-fitted Coordinates (BFC, 适体坐标网格) for Irregular Domains

10.2.2 Solving Diffusion Equation for Generating BFC

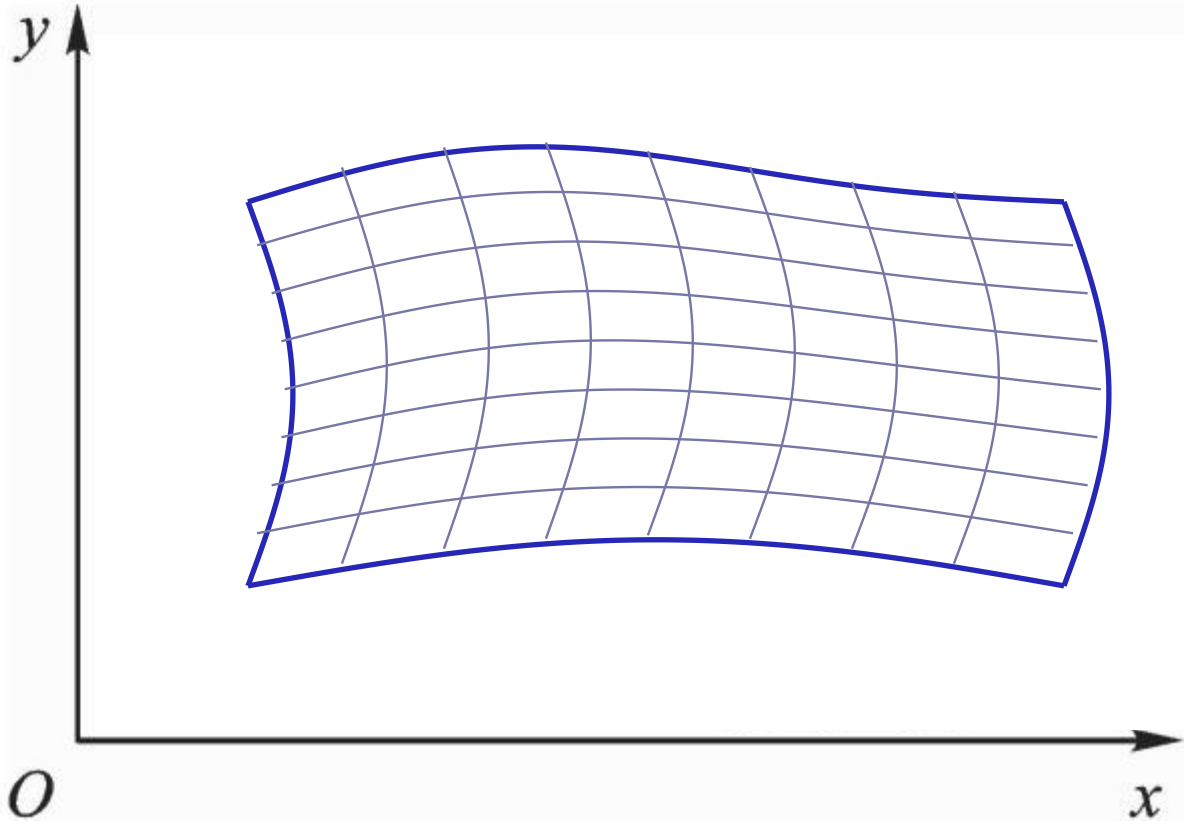
10.2.3 Control of Grid Spacing and Orthogonality （疏密与正交）

10.2 Introduction to Body-fitted Coordinates

10.2.1 Basic idea of body-fitted coordinates (BFC, 适体坐标网格) for irregular domains

1) In 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 (BFC, 适体坐标系): Cartesian coordinate is the body-fitted one for rectangles (笛卡尔坐标是矩阵的适体坐标系), and polar coordinate is the one for annular spaces (极坐标是环形空间的适体坐标系).

2) The existing orthogonal coordinates (直角坐标) can not deal with variety of complicated geometries (复杂结构) in different fields; Thus artificially (人为地) constructed body-fitted coordinates are necessary to meet the different practical requirements.

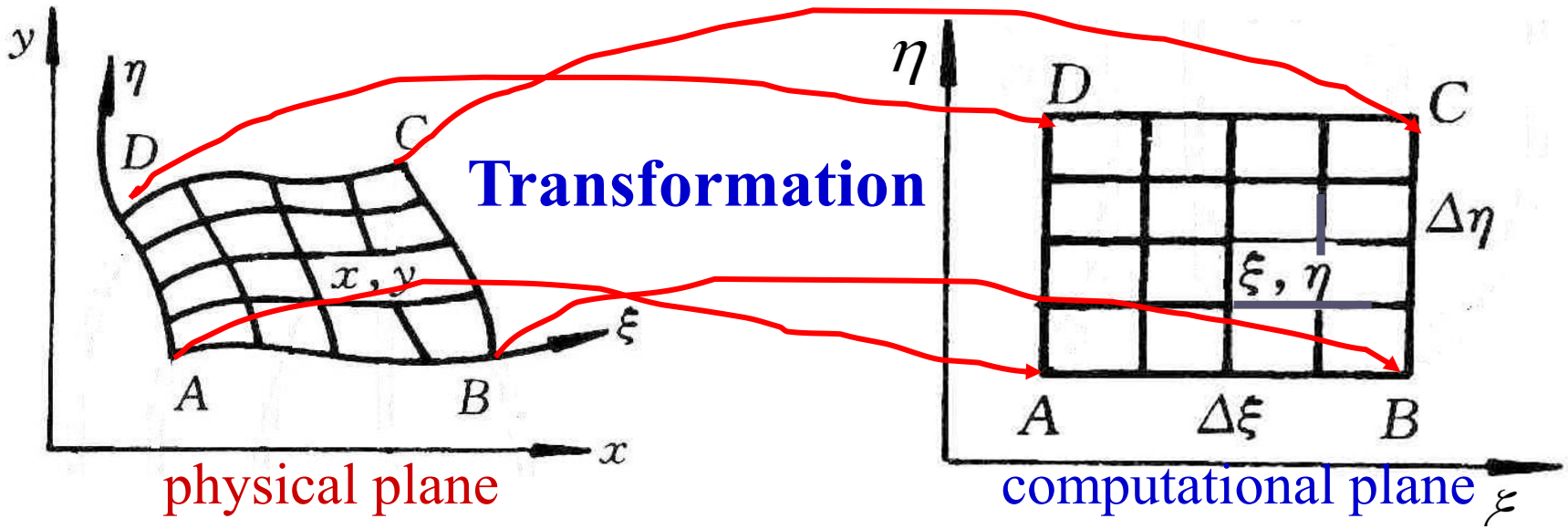


Irregular domain

Meshes with low quality

1) An irregular domain in the rectangular coordinates.
2) Generating meshes for the irregular domain.
3) The mesh is not regular, which means the mesh quality is not good, which will affect the solution accuracy. (网格质量不好，影响仿真精度)

Basic idea of BFC



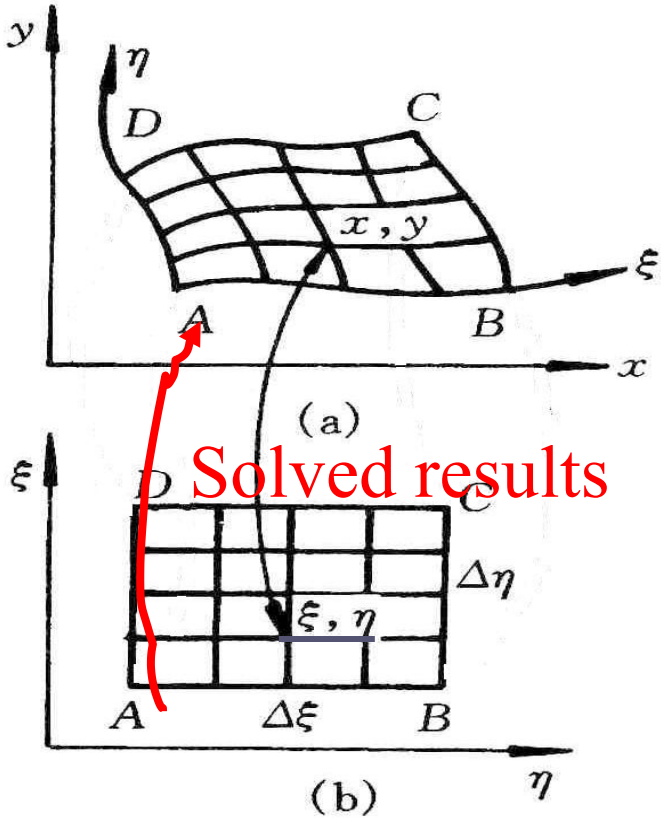
1. Assuming that a BFC has been constructed denoted by ξ and η;

2. Regarding ξ and η as the two coordinates of a Cartesian coordinate in an imaginary (假想的) computational plane, then the irregular geometry in physical plane (物理平面) transforms to a rectangle in the computational plane (计算平面).

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

4 . Simulation is first conducted in the computational plane , then the converged solution is transferred from the computational plane to physical plane.

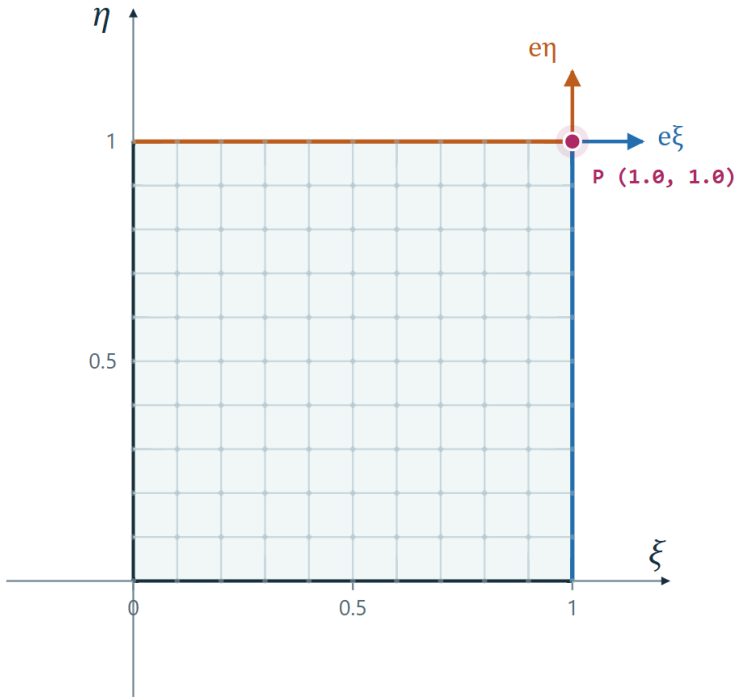
In such a way the simulation domain is greatly simplified.



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 (空间位置的对应关系）.

计算平面 (ξ, η)

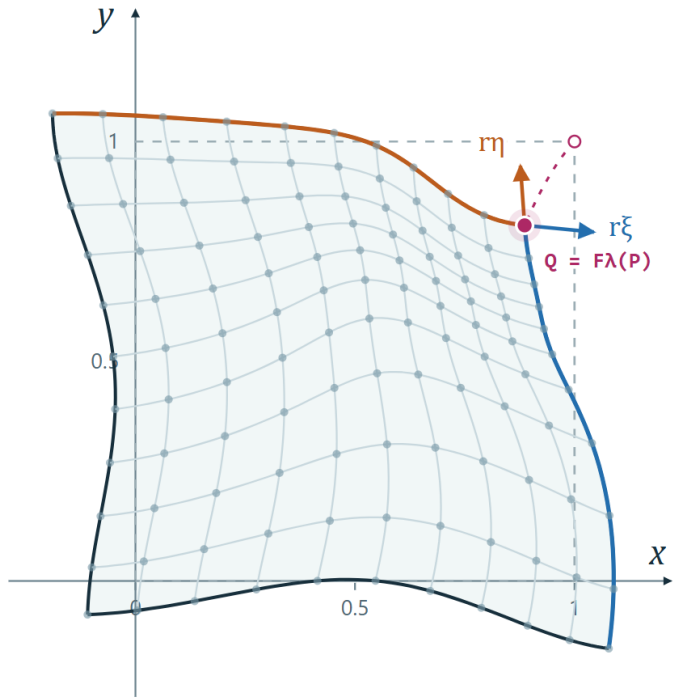
固定坐标 · 点击可选点



计算节点 P = (1.000, 1.000)

物理平面 (x, y)

抓住交点 → 拖动 → 松手

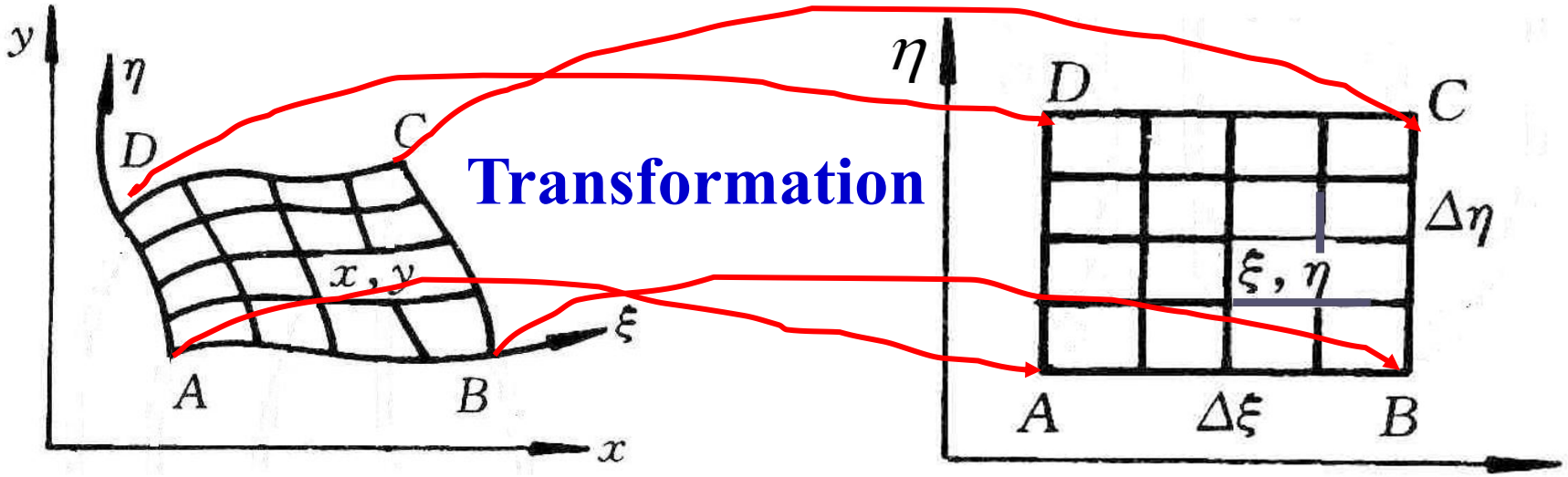


物理节点 Q = (0.886, 0.809)

ξ = 1.00：同一列　η = 1.00：同一行　当前节点的变形轨迹　原始方形

Click here to turn on the APP

nonlinear-coordinate-mapping-draggable.html



physical plane
物理平面
The plane we are familiar with
The shape is strange
Not friendly for generating mesh

computational plane
计算平面
The plane we are not familiar with
The shape is simple
Friendly for generating mesh

1. **Position in this plane?** ⟷ **Position in this plane?**
2. **Equations in this plane** ⟶ **Equations in this plane?**
3. **Results in physical plane?** ⟵ **Simulation conducted in this plane**

Procedure of solving problem by BFC

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

2. **Equation transformation**: transform governing eqs. and boundary conditions from **physical plane** to **computational plane;**

3. **Solving**: discretize gov. eq. and solving the algebraic equations in the **computational plane.**

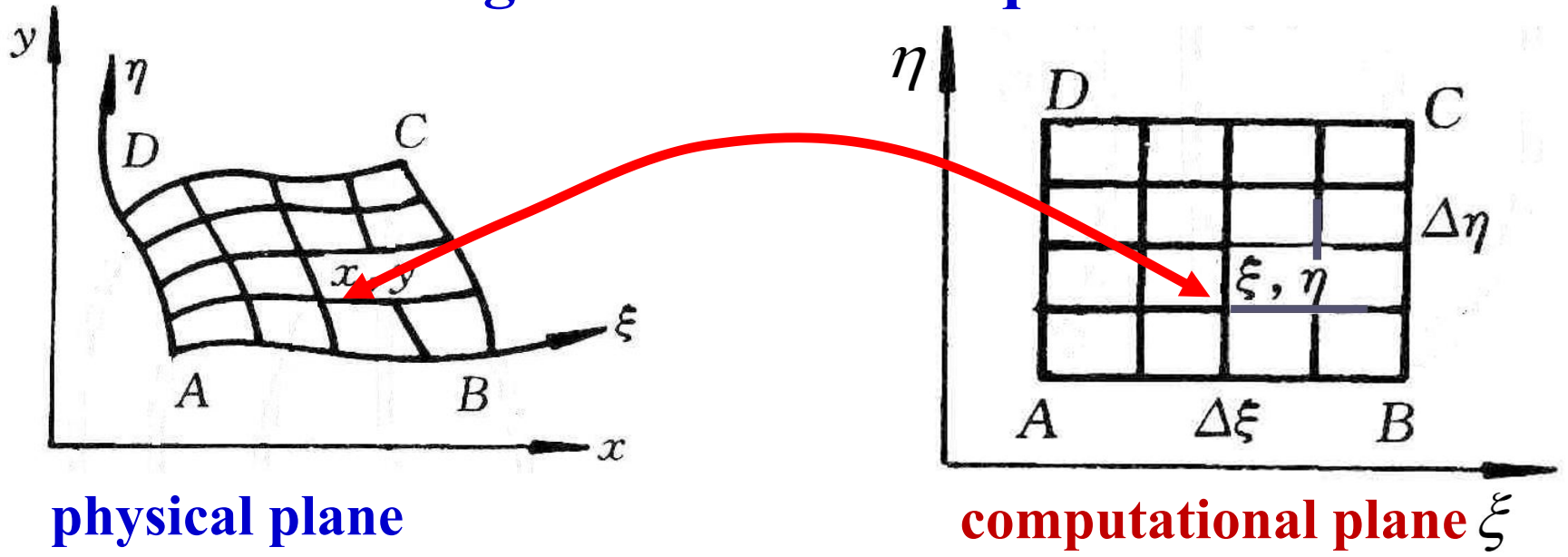
4. **Result back transfer**: transfer solutions from the **computational plane** to the **physical plane**.

Grid generation or the relationship between two planes

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

$(\xi, \eta) \longrightarrow (x, y)$

Grid generation technique



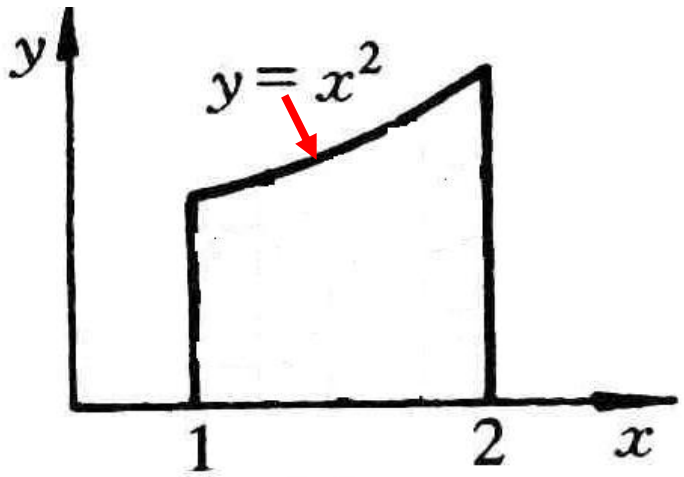
physical plane

computational plane

For some simple cases we can obtain body-fitted coordinates just by **boundary normalization** (**边界规范化**).

Example 1: 2-D nozzle

A plane nozzle is given by the profile $y = x^2$, its body-fitted coordinates can be constructed as follows:



$\xi = x$

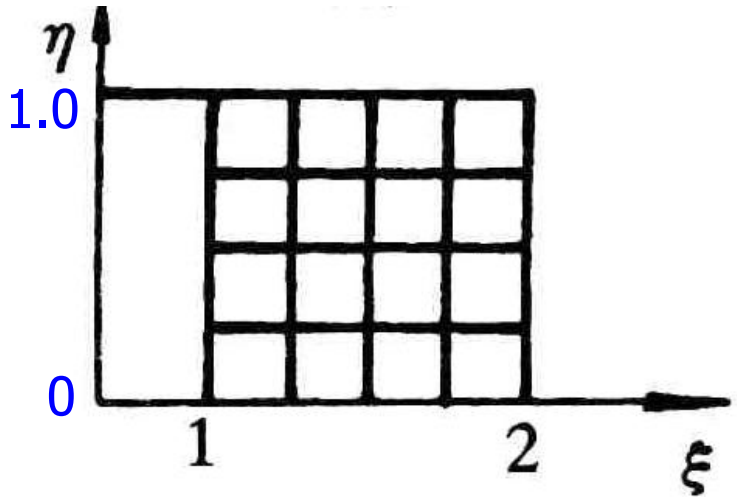
$\eta = y / y_{\max}$

normalization

$y_{\max} = x^2$

$x = \xi$

$y = \eta \xi^2$

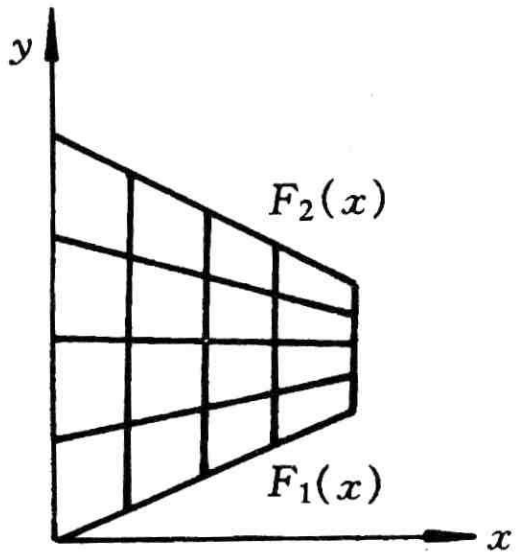


Example 2: Trapezoid (梯形) enclosure

Functions of two inclined boundaries are given by:

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

The grid in the trapezoid enclosure is generated.

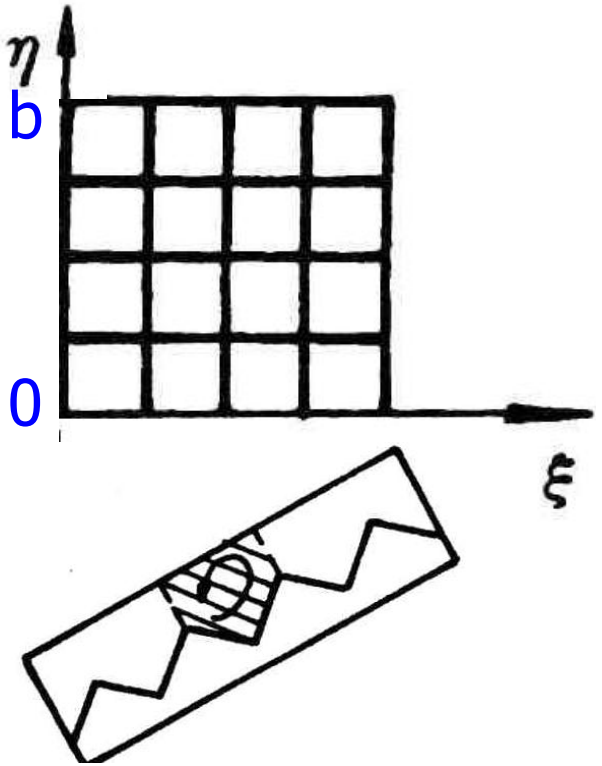


$$\xi = ax$$

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

normalization

Normalized by the distance between top and bottom



Solar collector

For some **simple cases** we can obtain body-fitted coordinates just by **boundary normalization** (**边界规范化**).

Q1: what is the general method for complicated cases?

10.2.2 Solving diffusion equation for generating BFT

1) Known conditions and task of grid generation by PDE

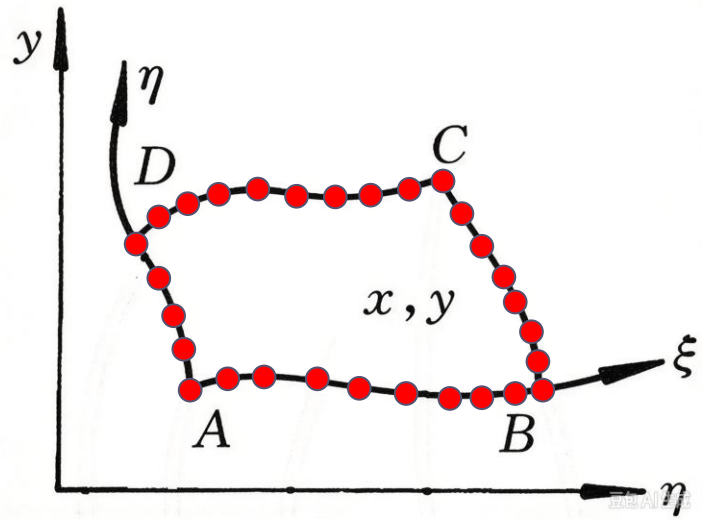
Known conditions

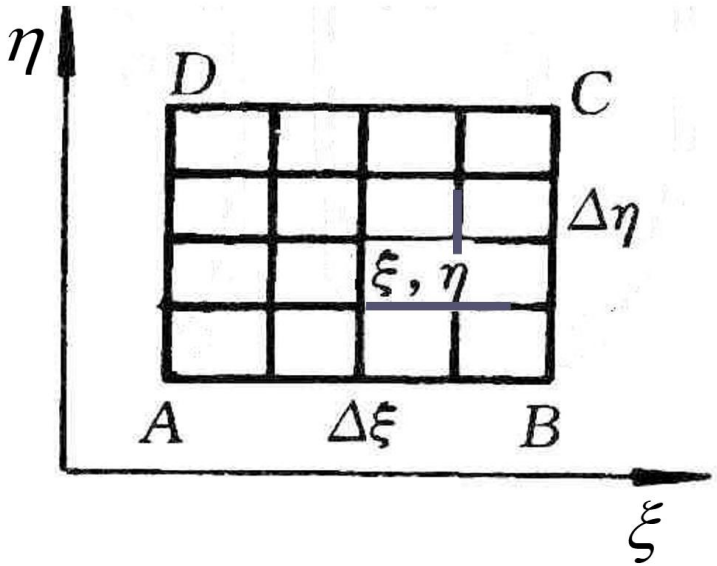
1. The **grid distribution in computational plane** is given; Usually it is a rectangular and very uniform.

2. The **grid arrangement on the physical boundary** is predefined by the researchers according to the flow and heat transfer processes studied.

3. At the boundary, relationship between **(ξ, η) —— (x, y) is known.**

Physical plane



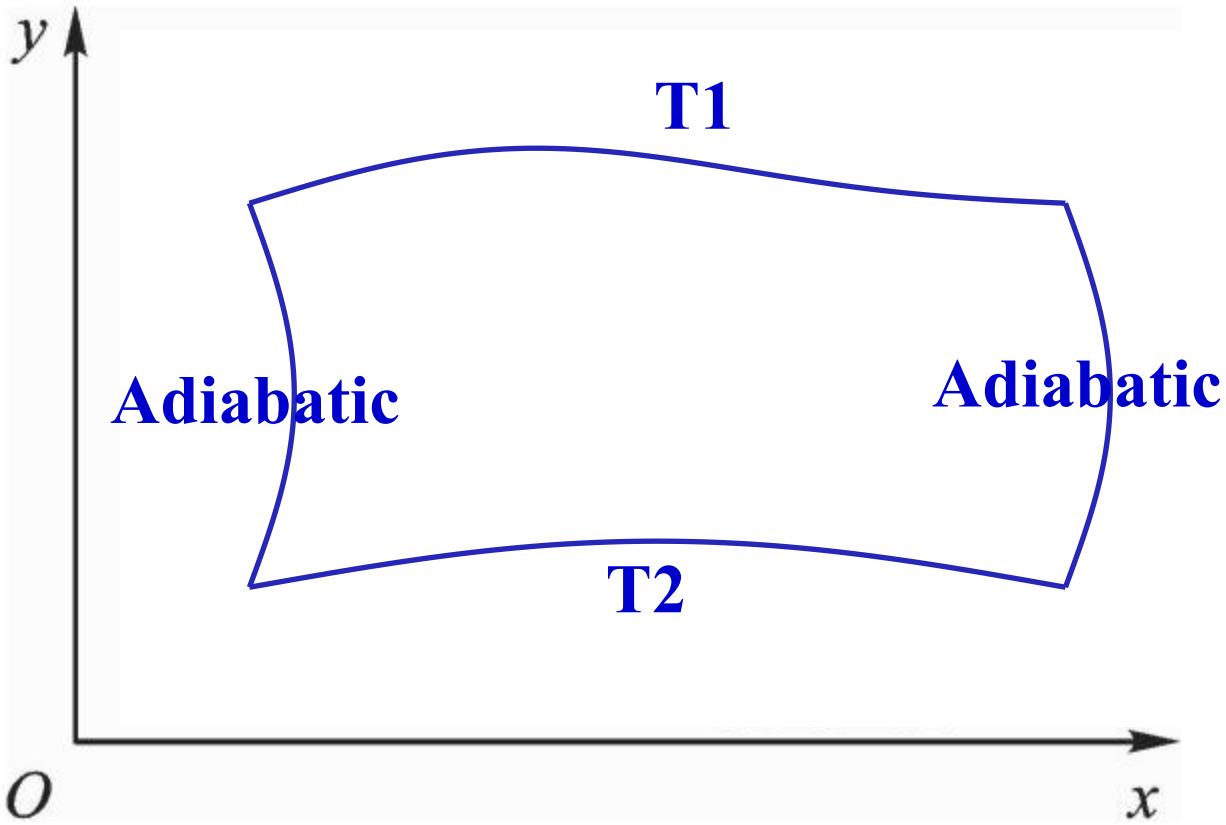


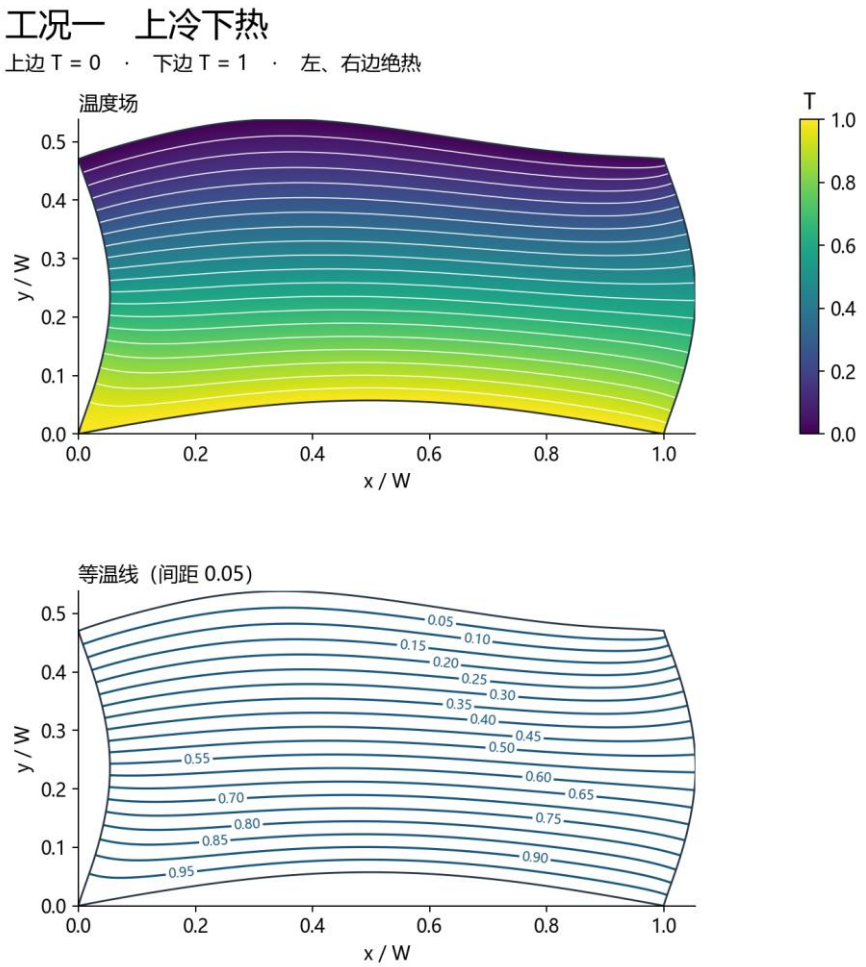
Computational plane

2) Inspiration from pure heat conduction process

$$\nabla(\lambda\nabla T)=0 \longrightarrow \nabla(\nabla T)=0$$

constant λ

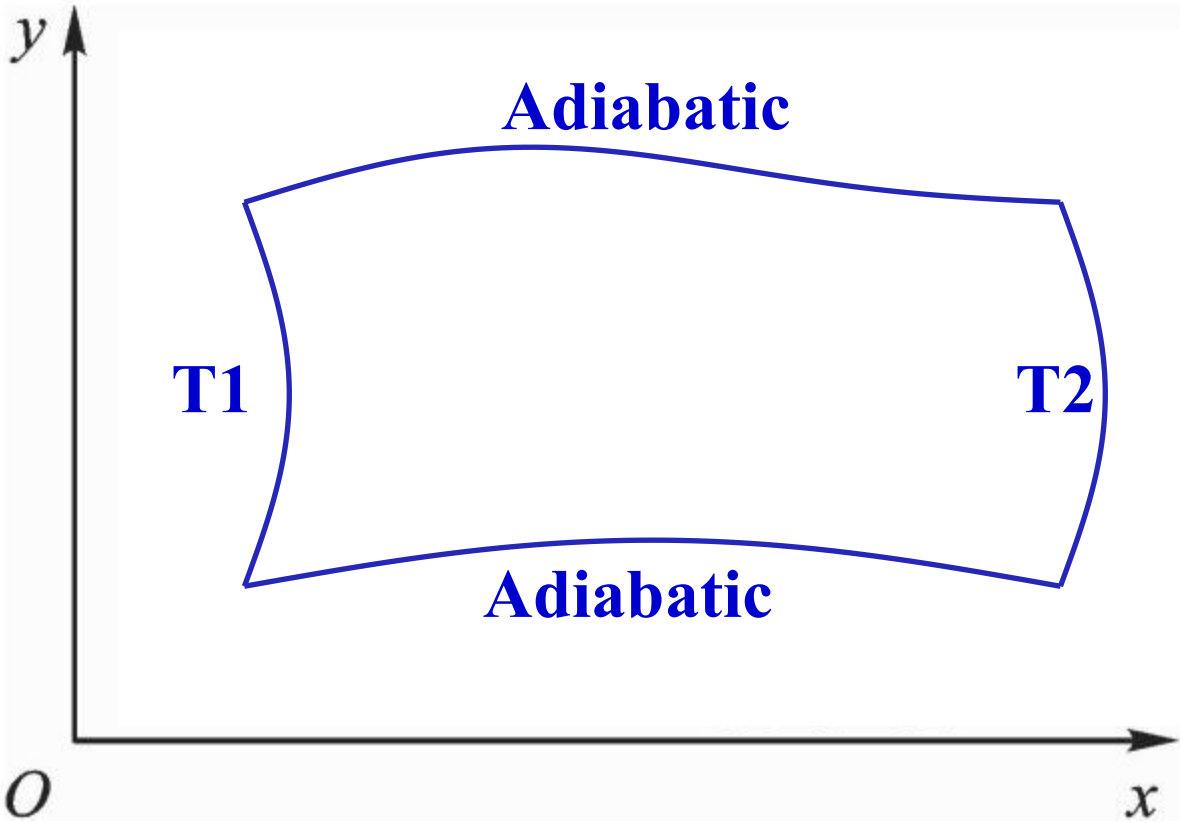


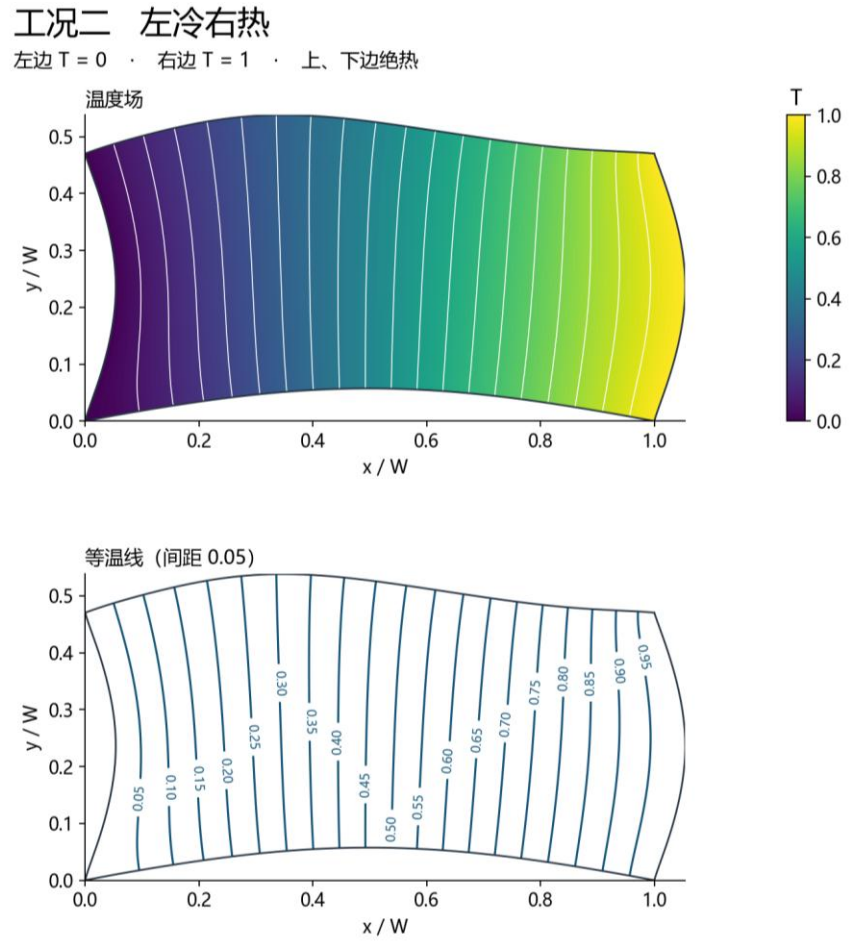


Pure heat conduction process

$$\nabla(\lambda\nabla T)=0 \longrightarrow \nabla(\nabla T)=0$$

constant λ



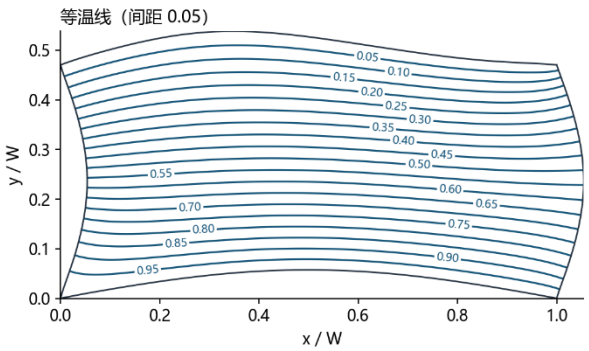


Pure heat conduction process

$$\nabla(\lambda\nabla T)=0 \longrightarrow \nabla(\nabla T)=0$$

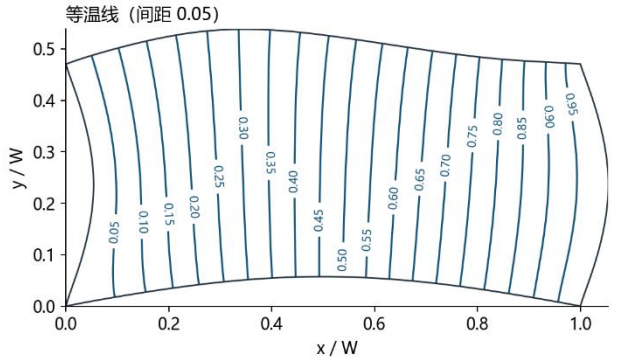
constant λ

1. Such equation is called **Laplace equation**, which is a diffusion-type equation.
2. With boundary values given, the temperature field can be numerically predicted.
3. The isothermal lines (等温线） present as **smooth curves**, which are free from self-intersection and overlapping (等温线非常光滑，既不重叠，也不交叉）。



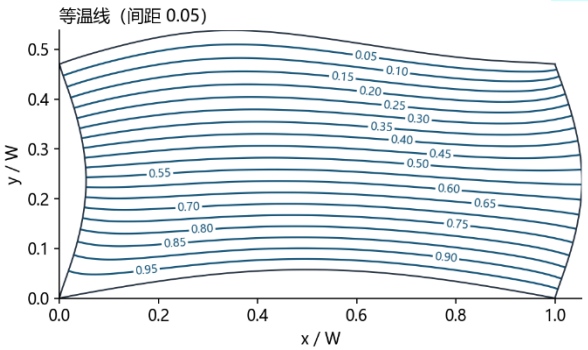
Isothermal lines

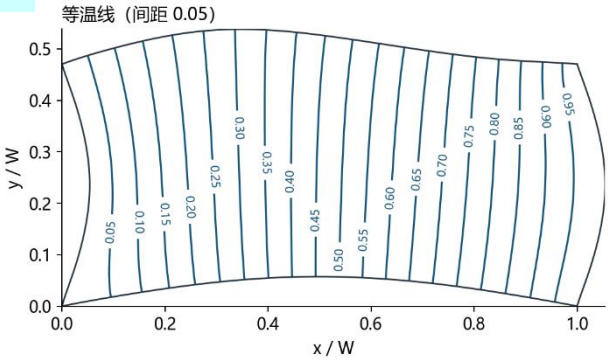
Grid lines



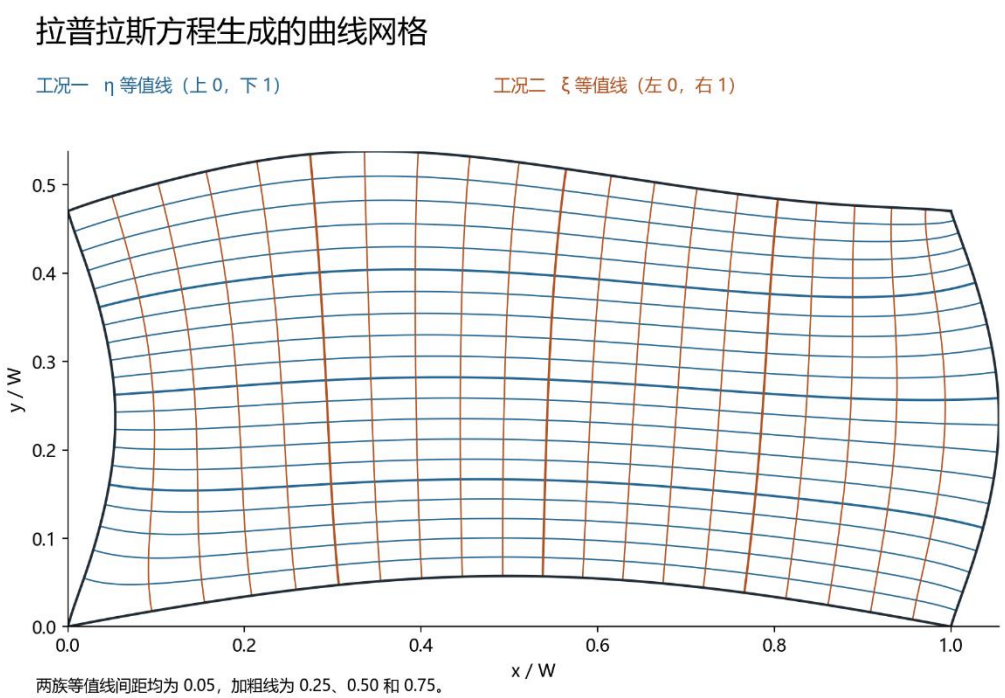
The above characteristics are exactly what a good grid requires.

Isothermal lines





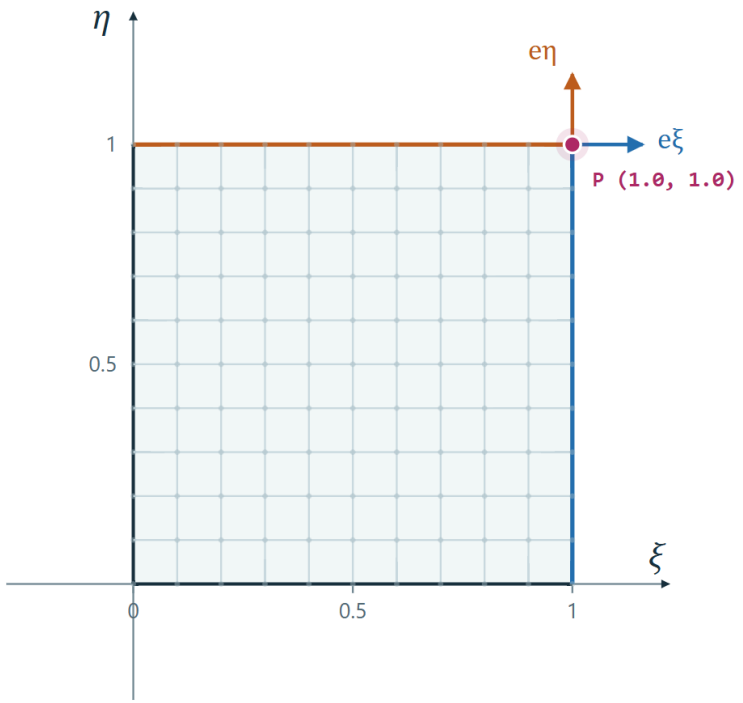
Grid lines



Therefore, Laplace equation is solved to generate grid lines

计算平面 (ξ, η)

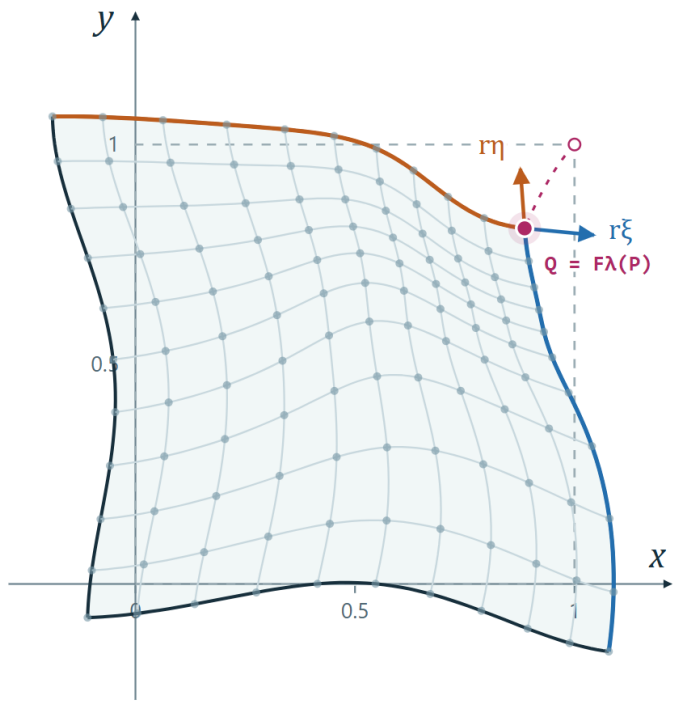
固定坐标 · 点击可选点



计算节点 P = (1.000, 1.000)

物理平面 (x, y)

抓住交点 → 拖动 → 松手



物理节点 Q = (0.886, 0.809)

ξ = 1.00：同一列　η = 1.00：同一行　当前节点的变形轨迹　原始方形

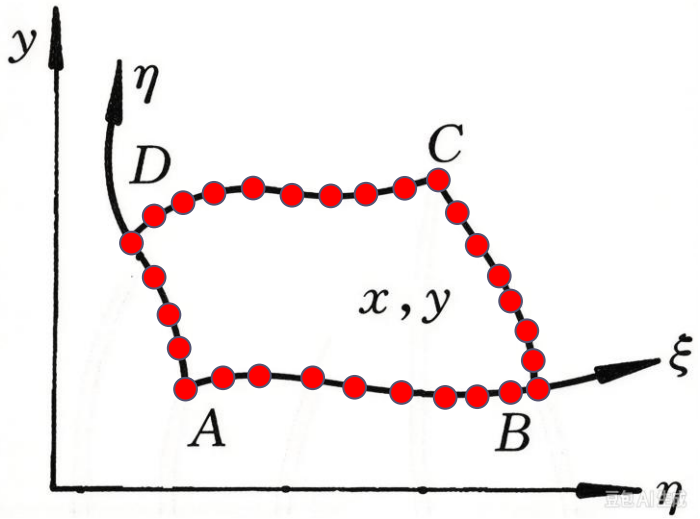
Click here to turn on the APP

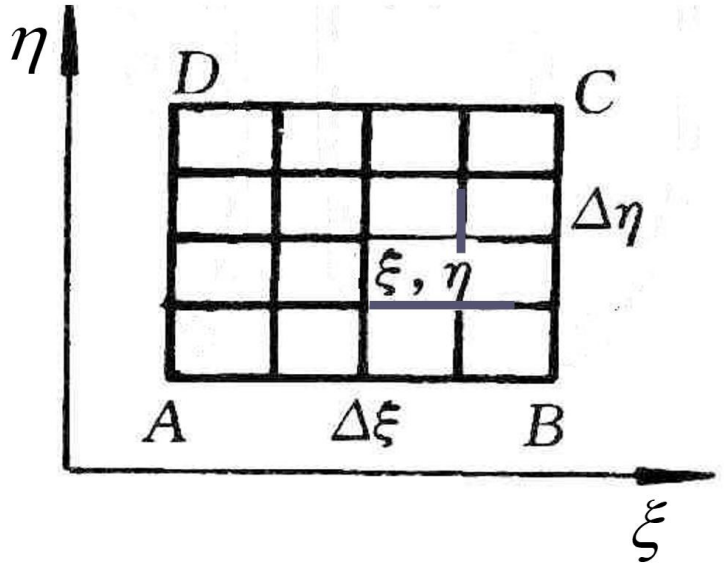
nonlinear-coordinate-mapping-draggable.html

3) Equations for grid generation technique

Therefore, similar to the Laplace equation for pure heat conduction, Laplace equation is developed for the grid generation.

Physical plane





Computational plane

Method 1

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

Method 1: simulate from the physical plane to the computational domain

$$\nabla^2\xi = 0;\ \nabla^2\eta = 0 \quad i.e., \quad \frac{\partial^2\xi}{\partial x^2} + \frac{\partial^2\xi}{\partial y^2} = 0; \quad \frac{\partial^2\eta}{\partial x^2} + \frac{\partial^2\eta}{\partial y^2} = 0$$

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

ξ_B, η_B are given（i.e., ξ, η of boundary nodes are known)

Find values of **(ξ , η)** for any inner point **(x , y)** within the solution region in physical plane. It is essentially the same as the pure heat conduction equation.

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!

Method 2: simulate from the computational domain to physical plane

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

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

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

cross-derivative term

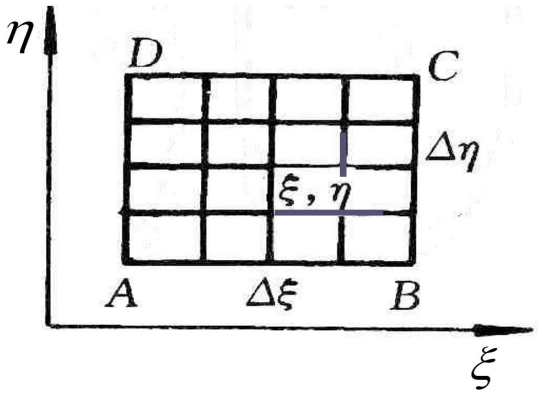
$$\alpha = x_\eta^2 + y_\eta^2; \qquad \beta = x_\xi x_\eta + y_\xi y_\eta; \qquad \gamma = x_\xi^2 + y_\xi^2$$

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

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

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

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



The equation is relatively complex, although the coordinates are simple.

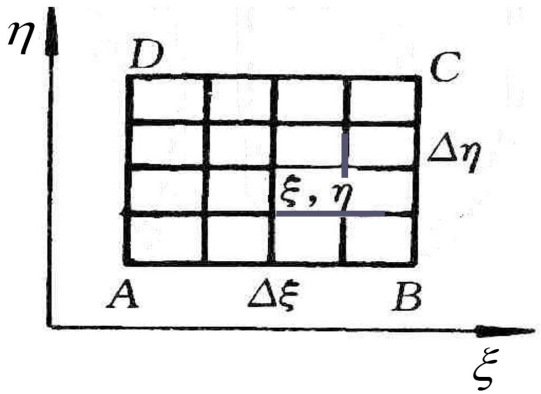
The above two differential equations are non-linear and can be regarded as two non-isotropic (各向异性） diffusion equations.

Thus, the essence (本质) of grid generation by PDE is to solve two boundary value diffusion problems in the computational domain! The boundary value problems are set up by elliptic partial differential equations.

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

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

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



One may wonder why the equation for from computational domain to the physical plane is as simple as follows:

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

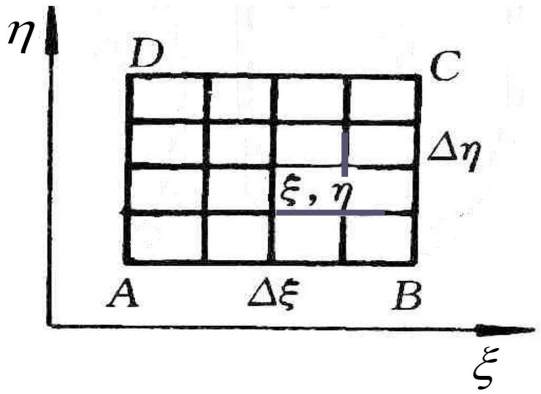
This is because the coordinate transformation between two planes should obey basic mathematical rules.

3) How to get the equation

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

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

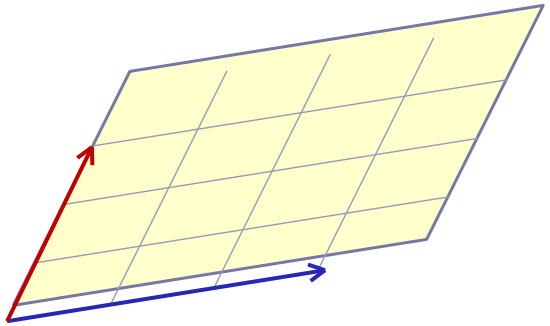
$$\alpha = x_\eta^2 + y_\eta^2; \qquad \beta = x_\xi x_\eta + y_\xi y_\eta; \qquad \gamma = x_\xi^2 + y_\xi^2$$



Coordinate Mappings, Jacobian and Geometric Coefficients

6)-1 Forward and inverse mappings

6)-2 Geometry of a local cell



Jacobian Inversion and the First-Order Chain Rule

01 Differentiate the inverse identities

02 Assemble and invert the Jacobian

Derivatives of function and its inverse function (反函数)



Matrix inversion expresses every inverse-map derivative in terms of forward-map derivatives.

First-Order Differential Operators

01 Chain-rule operators

02 Direct substitution is valid but algebraically longer

Apply the second-order chain rule to the physical coordinate identities.

The identities for physical x and y add no assumption about Laplace equations in the computational plane.

Second-Order Chain Rule (1): Differentiation in Physical x

01 First derivative and product rule

02 Apply the chain rule inside each product

The two equal mixed-derivative products combine to give the factor 2.

Second-Order Chain Rule (2): Differentiation in Physical y

Collect the sum, using the physical coordinate identity

Add the physical second derivatives and collect terms by the same forward-map derivative.

Harmonic Coordinate Conditions and Term Elimination

01 Gradient notation in physical Cartesian coordinates

02 The complete composite Laplacian for the coordinate x

Both highlighted terms vanish by the harmonic coordinate conditions.

Only the original Laplace equations justify eliminating the final two chain-rule terms.

Elliptic Body-Fitted Grid Generation Equations

Substitute into the reduced equation

Repeat the same chain-rule derivation for the coordinate y.

The inverse-gradient dot product supplies the minus sign in the mixed-derivative coefficient.

Boundary Conditions and Mathematical Properties

01 Dirichlet boundary data

Boundary node positions are prescribed.
The PDEs determine the interior positions.

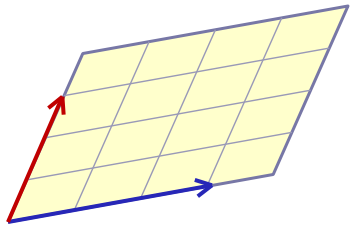
02 Quasilinear, coupled equations

The physical-plane Laplace equations are linear.
After inversion, the coefficients depend on
first derivatives of the unknown mapping.

Update the coefficients and iterate for both coordinates.

03 Elliptic principal part

04 Mixed derivatives and the local angle



The mixed term vanishes for orthogonal tangents.

Known boundary coordinates and iterative solution of the coupled interior equations produce the grid.

Derivation Summary and Two Distinct Grid Models

Derivation route

↓

Jacobian inversion

↓

First-order chain rule

↓

Second-order chain rule

↓

Apply harmonic conditions

↓

Geometric coefficients

Method A: physical coordinates are harmonic

A linear Laplace model in the computational plane.

Method B: inverse coordinates are harmonic

Jacobian inversion links the derivatives.
The second-order chain rule transforms the PDEs.
Iterate for interior nodes with fixed boundary data.

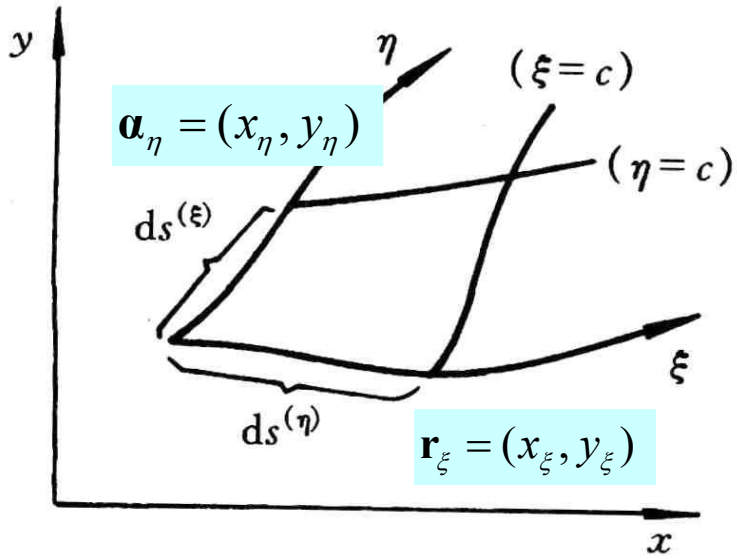
The two methods impose different assumptions. Neither follows by simply omitting coefficients.

4) Explanation of α, β and γ

The Lamé coefficients (拉梅系数) and/or metric coefficients (度量张量)

For a coordinate system in the computational domain, (ξ, η) do not necessarily represent physical lengths; they may merely serve as indices like grid numbers: $\xi = 0,1,2,3,4\ldots$, $\eta = 0,1,2,3,4\ldots$.

- $\mathbf{r}_\xi$: the change of x and y in the physical plane (x, y) along the ξ -grid line when ξ increases by one unit.
- $\boldsymbol{\alpha}_\eta$: the change of x and y travelled in the physical plane (x, y) along the η -grid line when η increases by one unit.

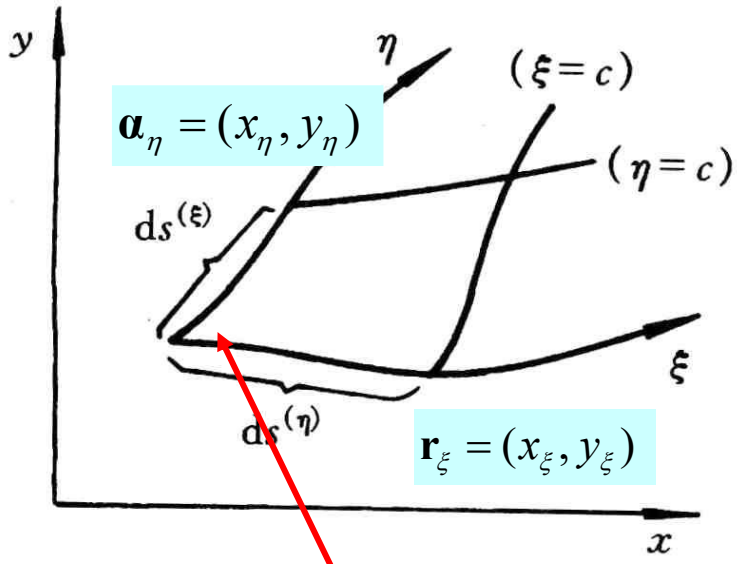


$$\mathbf{r}_\xi = (x_\xi, y_\xi) \qquad \boldsymbol{\alpha}_\eta = (x_\eta, y_\eta)$$

4) Explanation of α, β and γ

The Lamé coefficients (拉梅系数) and/or meric coefficients (度量系数)

- $\sqrt{\gamma}$ the **true length** (实际长度） travelled in the physical plane (x, y) along the ξ-grid line when ξ increases by one unit.
- $\sqrt{\alpha}$ the **true length** （实际长度）travelled in the physical plane (x, y) along the η-grid line when η increases by one unit.



Lamé coefficients

$$\sqrt{\alpha} = \sqrt{x_\eta^2 + y_\eta^2} \qquad \sqrt{\gamma} = \sqrt{x_\xi^2 + y_\xi^2}$$

Metric coefficients

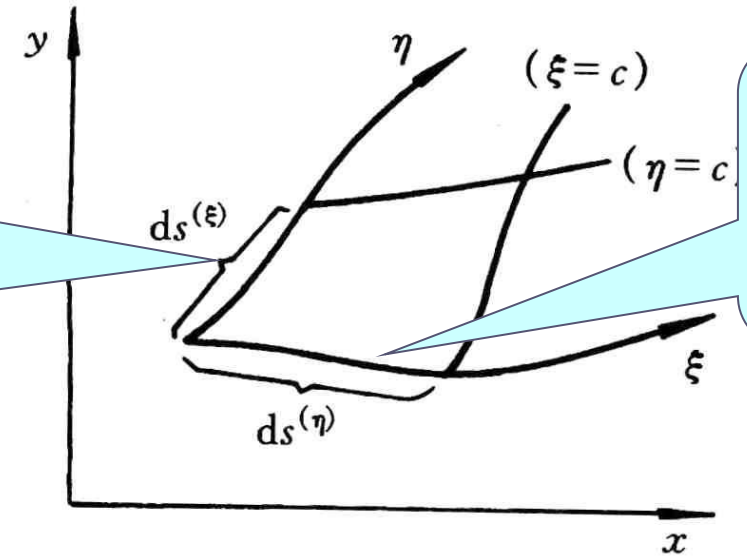
$$\gamma = x_\xi^2 + y_\xi^2 \qquad \alpha = x_\eta^2 + y_\eta^2;$$

Orthogonality (正交性)

$$\beta = \mathbf{r}_\xi \cdot \boldsymbol{\alpha}_\eta = (x_\xi, y_\xi) \cdot (x_\eta, y_\eta) = x_\xi x_\eta + y_\xi y_\eta$$

$$\beta = 0, \; if \; = 90^\circ$$

$ds^{(\xi)} = \sqrt{\alpha}\,d\eta$ is a differential arc length (弧长) in curve with constant ξ



$ds^{(\eta)} = \sqrt{\gamma}\,d\xi$ is a differential arc length in curve with constant η

Lamé coefficients
（拉梅系数）

$$\sqrt{\alpha} = \sqrt{x_\eta^2 + y_\eta^2} \qquad \sqrt{\gamma} = \sqrt{x_\xi^2 + y_\xi^2}$$

Metric coefficients
(度量系数）

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

Orthogonality
(正交性）

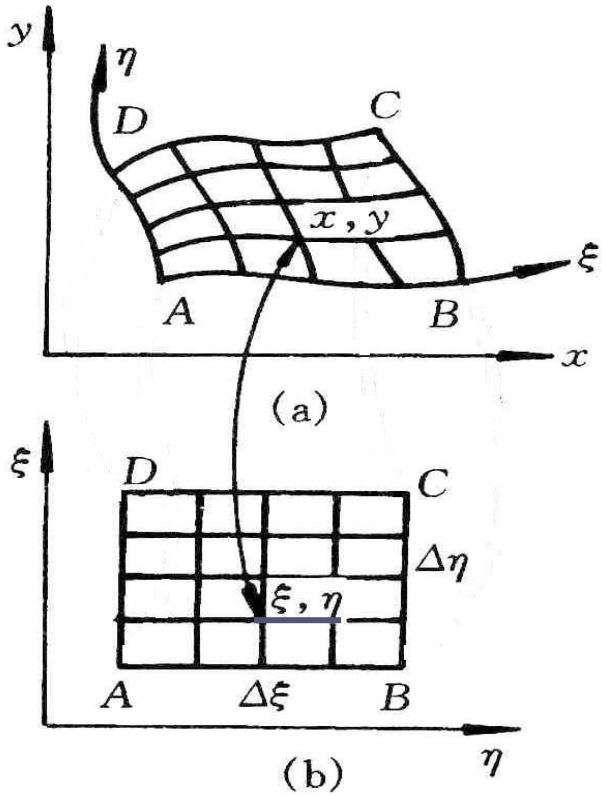
$$\beta = \mathbf{r}_\xi \cdot \boldsymbol{\alpha}_\eta = (x_\xi, y_\xi) \cdot (x_\eta, y_\eta) = x_\xi x_\eta + y_\xi y_\eta$$

5) 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;

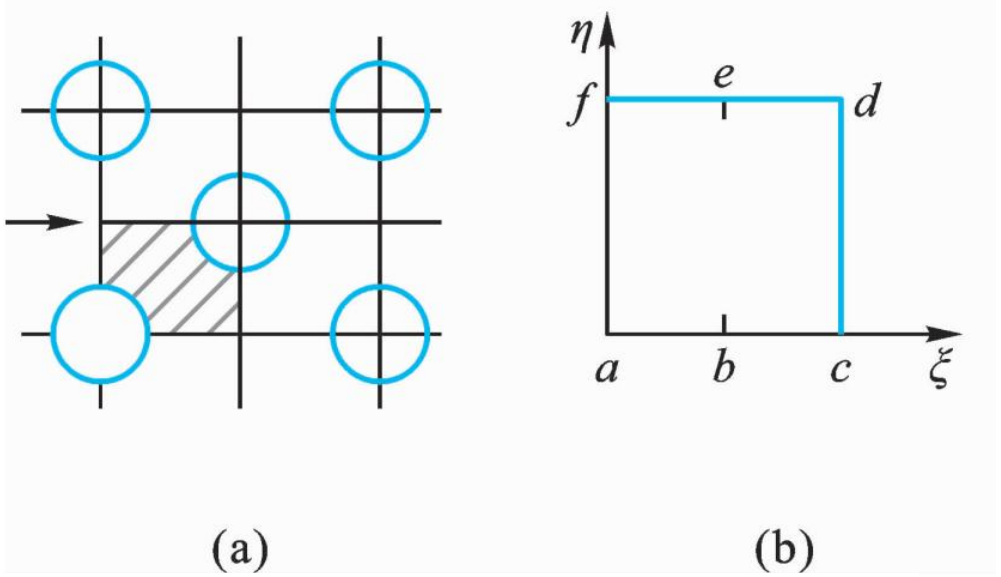
2. Setting boundary nodes in physical plane according to given conditions;

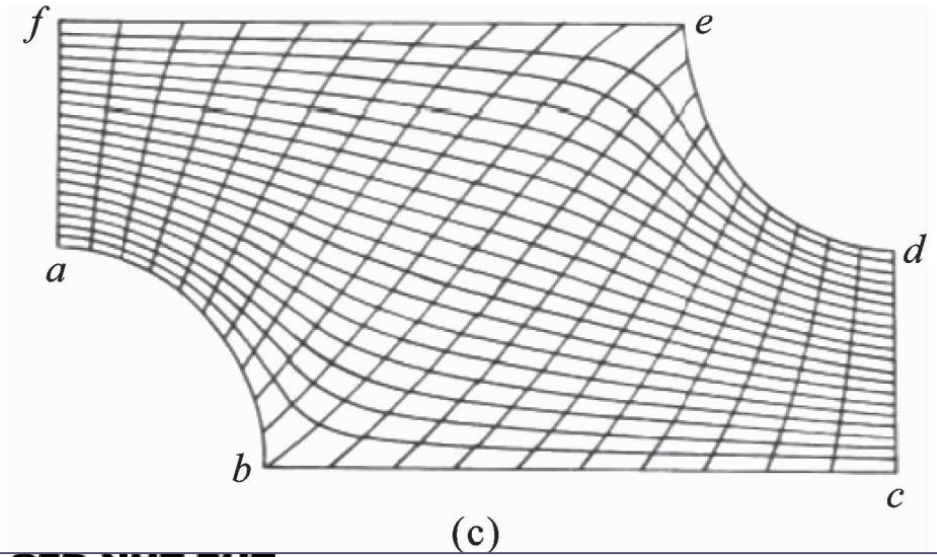
3. Numerically solving the two boundary value problems in computational plane, by regarding them as non-isotropic and nonlinear diffusion problems with non-constant source term.

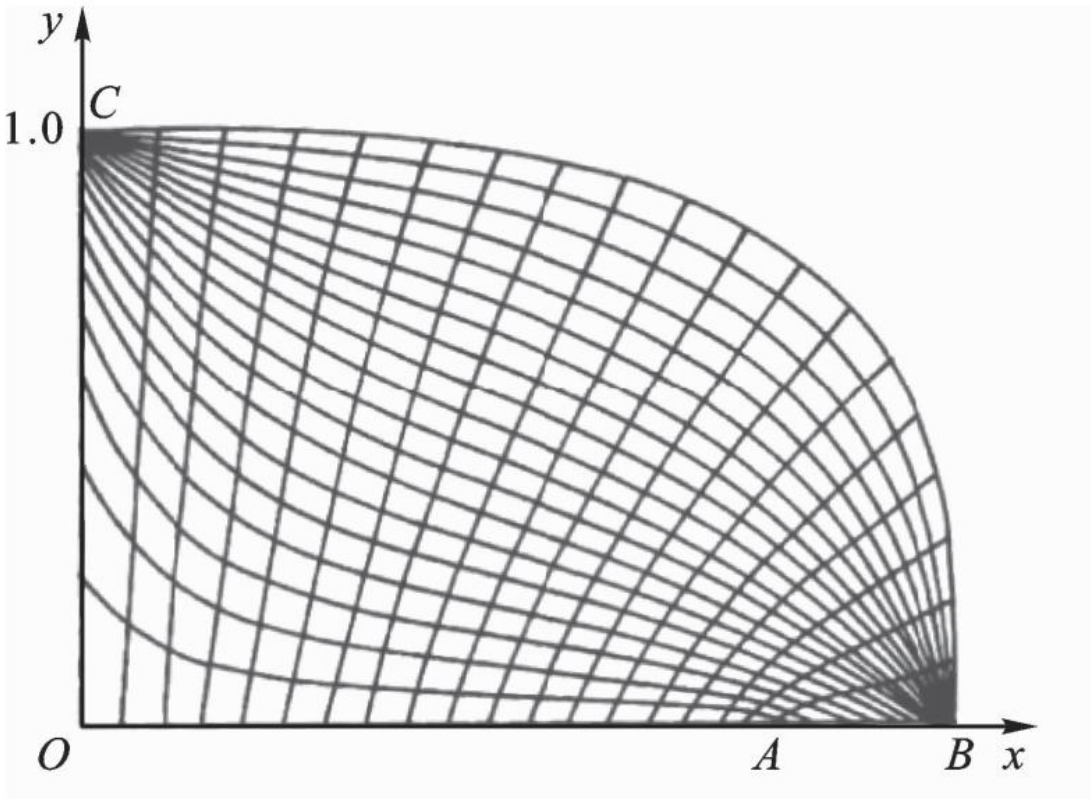


4. Calculating $x_\xi, x_\eta, y_\xi, y_\eta$ after getting the correspondence between (ξ, η) and (x, y).

7) Examples







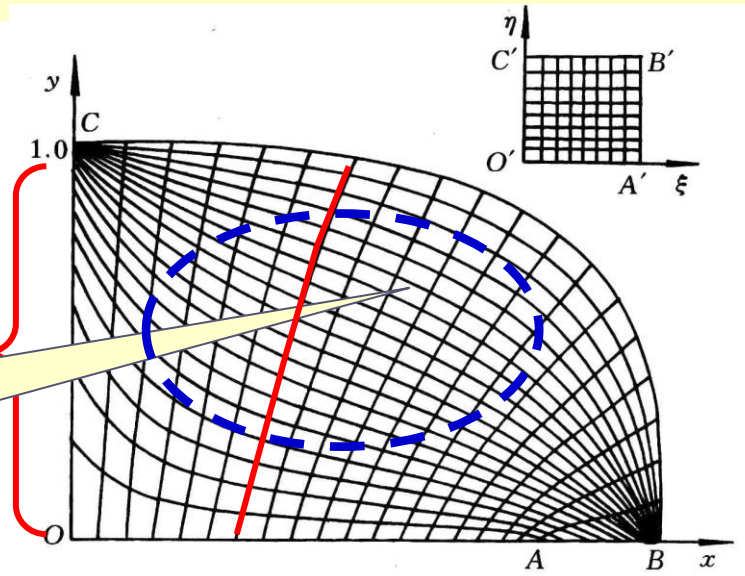
10.2.3 Control of Grid Spacing and Orthogonality （疏密与正交控制）

1）Major problems of grid system generated by Laplace equation

(1) The given grid distribution specified along the boundary in physical plane will be automatically ununiformed (or smoothed out, 被抹平) within the solution domain, which is not required. (物理平面边界上给定的网格分布，会在求解域内部自动被抹平/趋于均匀）。

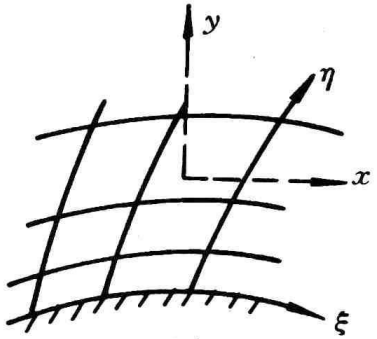
The given strongly non-uniform distribution at left boundary

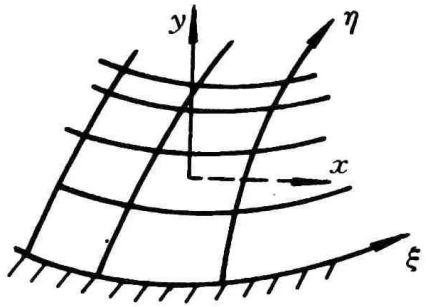
In the domain grid distribution has been unified (smoothed out).

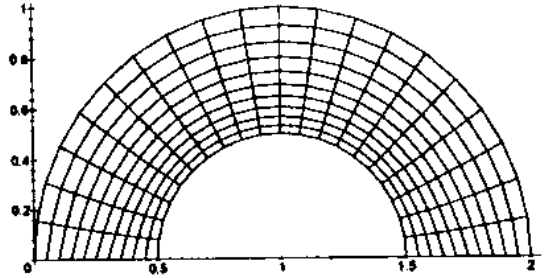


The specified boundary grid feature is not preserved （保持） within the domain.

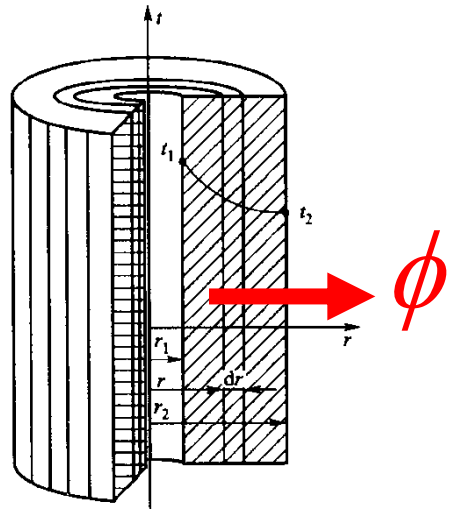
(2) Along the normal to a curved wall spacing between grid lines changes automatically.







Such features are inherently related to diffusion process: For steady 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 grid lines with boundary.**

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. (添加源项控制等温线)**

In physical plane Poisson equation is：

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

In computational plane, it becomes:

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

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

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

3) Thomas-Middlecoff method for determining P,Q

P* and *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) Manually constructed source terms (人为构造源项) by assuming

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

Controlling the orthogonality of boundary grid line（控制网格正交）

Controlling grid density within domain---transmitting the specified density on the boundary to inner region (控制网格疏密）

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

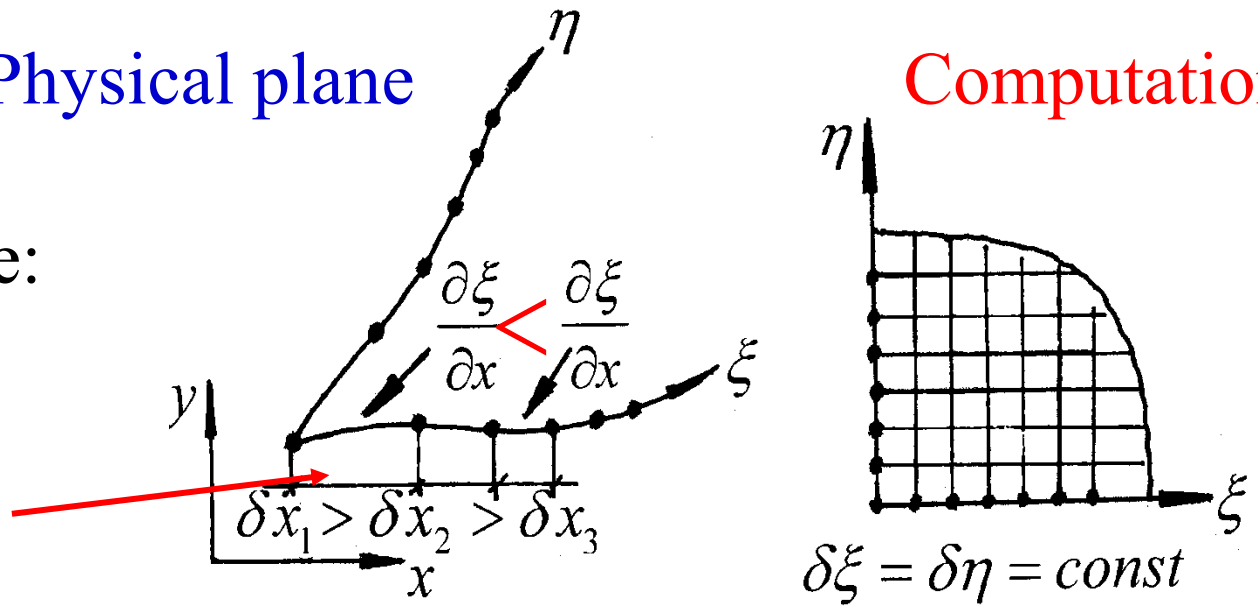
Controlling grid density within domain---transmitting the specified density on the boundary to inner region (控制网格疏密）

Physical plane

Computational plane

Along the ξ coordinate:

$$\frac{\partial\xi}{\partial x})_1<\frac{\partial\xi}{\partial x})_2<\frac{\partial\xi}{\partial x})_3$$

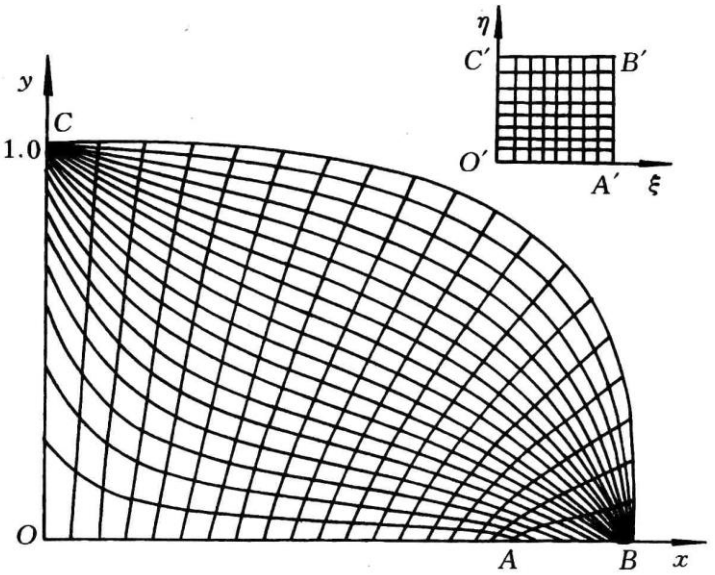


The first derivatives of ξ,η with respect to x, y, $\xi_x,\eta_x,\xi_y,\eta_y$ in the physical plane reflect the rate of changes. Thus, $(\xi_x^2+\xi_y^2)$ represents grid density distribution!

(2) How to determine ϕ and ψ

It can be found from the grid lines generated by the Laplace equation that the grid lines are not orthogonal (不正交）

An orthogonal grid means β=0, leading to all the terms with β in the NS equations as 0, which will reduce the numerical diffusion and better numerical stability.



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

Therefore, the function of ϕ **and** ψ **is constructed to** controlling the orthogonality of boundary grid line（控制网格正交）.
However, note that that totally orthogonality is not always possible.

(2) How to determine ϕ and ψ

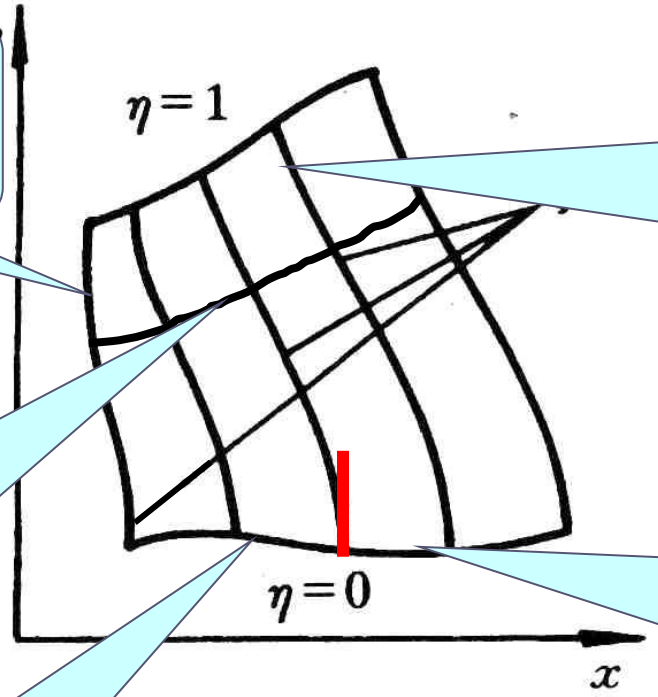
1) ϕ is first determined for the bottom and top boundaries where η is constant; ψ is first determined for the left and right boundaries where ξ is constant. (how to obtain such boundary values will be introduced later)

The boundary values of ϕ and ψ should satisfy following conditions: 1) the local gridlines are straight and 2) normal to the relative boundary (局部网格线是直线且垂直边界).

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 = Constant, **determining** ψ

On the constant ξ line ϕ is linearly interpolated with respect to η



On the constant η line ϕ is linearly interpolated with respect to ξ

Locally straight and orthogonal to the boundary

η = Constant, **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) Ways for determining ϕ on η=0 and η=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 following Poisson equation in computational plane:

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

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

and rewriting the equations in terms of ϕ,ψ, obtaining following 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

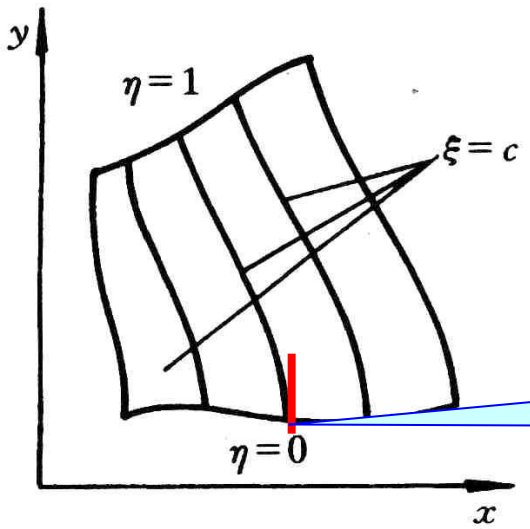
$$\alpha[y_\eta(x_{\xi\xi}+\phi x_\xi)-x_\eta(y_{\xi\xi}+\phi y_\xi)]=$$

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

0

Straight and normal

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



Locally straight and normal(局部平直正交)

On the local straight line, we have:

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

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

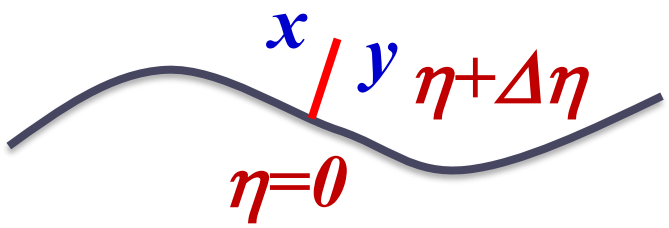
3) Summarizing: (i) Local orthogonality leads to β=0, (ii) local straight requires $(x_\eta/y_\eta)_\eta=0$.Thus the right hand side of the above equation equals zero:

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

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.

Local straight
***x/y*=constant**



Local orthogonality
***β*=0**

Thus, we have no way to calculate x_η/y_η; In order to determine this term following condition is utilized:

$$\text{From} \quad \beta = x_\xi x_\eta + y_\xi y_\eta = 0 \longrightarrow \frac{x_\eta}{y_\eta} = -\frac{y_\xi}{x_\xi}$$

y_ξ / x_ξ can be computed on the line of **η=constant**

Substituting into:

$$x_{\xi\xi} + \phi x_\xi = \left(\frac{x_\eta}{y_\eta}\right)(y_{\xi\xi} + \phi y_\xi) \longrightarrow$$

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

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

Finally:

$$\phi = -\frac{y_\xi y_{\xi\xi} + x_\xi x_{\xi\xi}}{x_\xi^2 + y_\xi^2} \quad (\text{on } \eta = 0, \eta = 1 \quad \text{boundaries})$$

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)

Summary

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = P(\xi,\eta) = \phi(\xi,\eta)(\xi_x^2 + \xi_y^2)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = Q(\xi,\eta) = \psi(\xi,\eta)(\eta_x^2 + \eta_y^2)$$

Controlling the orthogonality

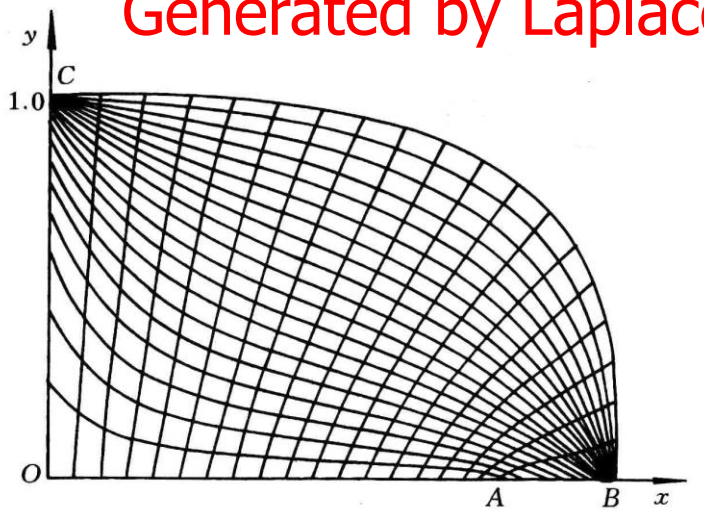
Controlling grid density within domain

$$\phi = -\frac{y_\xi y_{\xi\xi} + x_\xi x_{\xi\xi}}{x_\xi^2 + y_\xi^2}$$

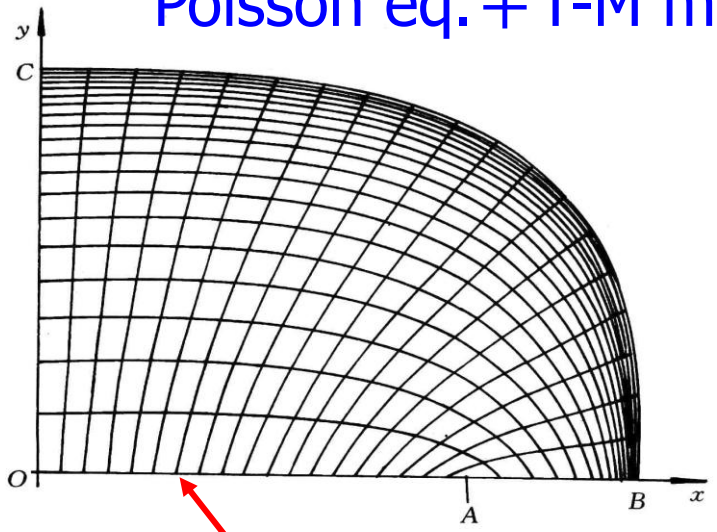
$$\psi = -\frac{y_\eta y_{\eta\eta} + x_\eta x_{\eta\eta}}{x_\eta^2 + y_\eta^2}$$

Application example of Thomas－Middlecoff method

Generated by Laplace eq.



Poisson eq.＋T-M method



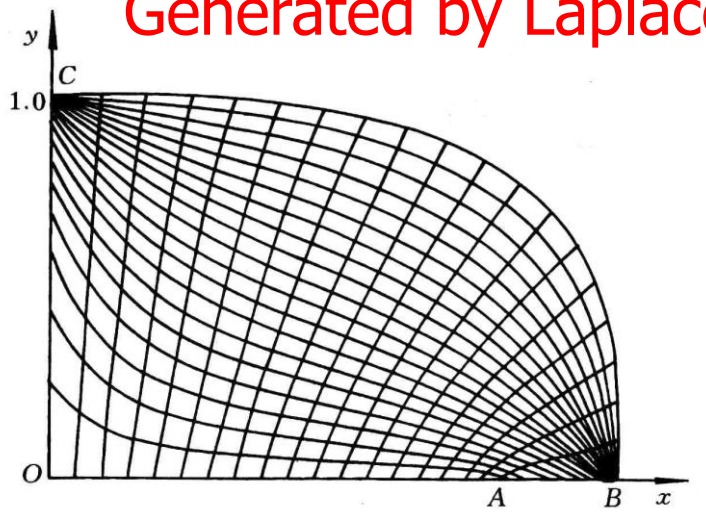
1) Mesh strategy assigned at the boundary is well preserved inside the domain.

2) The local orthogonality at the boundary is improved.

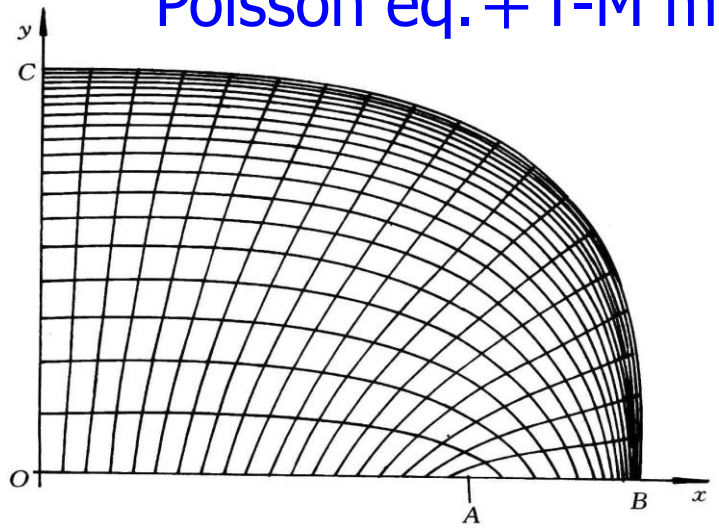
3) **However, the local orthogonality is not strictly satisfied. This is because the source term only driving the grid line to meet orthogonality. (驱动网格满足正交性，但没有能力严格保证正交性）**

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-mathematicians!

10.3 Discretization and Solution of Convection–Diffusion Problems in Body-Fitted Coordinates

10.3.1 Transformation of Boundary Conditions

10.3.2 Transformation of Governing Equation

10.3.3 Discretization in Computational Plane

10.3.4 SIMPLE Algorithm in the Computational Domain

10.3.5 Post-Processing the Results

10.3 Discretization and Solution of Convection–Diffusion Problems in Body-Fitted Coordinates

10.3.1 Transformation of Boundary Conditions

General expression of B.C. in physical plane

The three kinds of boundary conditions can be unified by following general form, where A, B and C are constants:

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

A = 0: second kind

B = 0: first kind

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

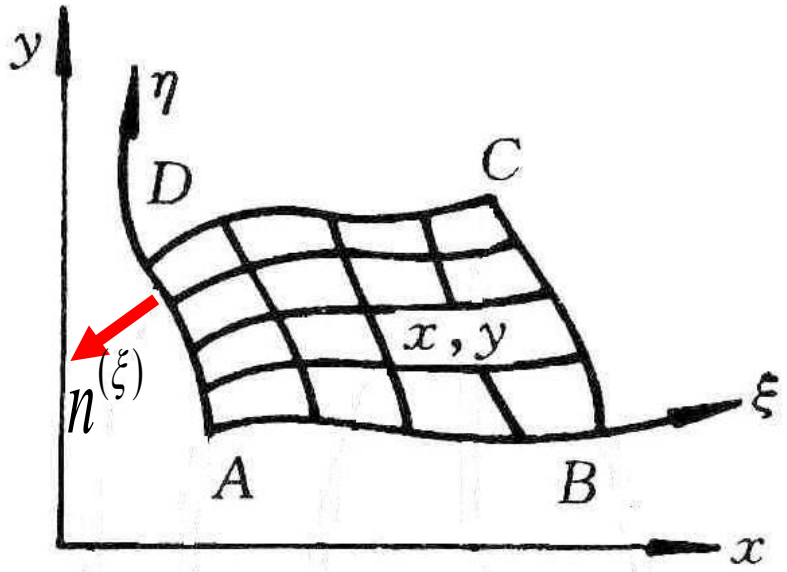
Since A,B, C and Gama are constants, this part in the two planes should have the same value

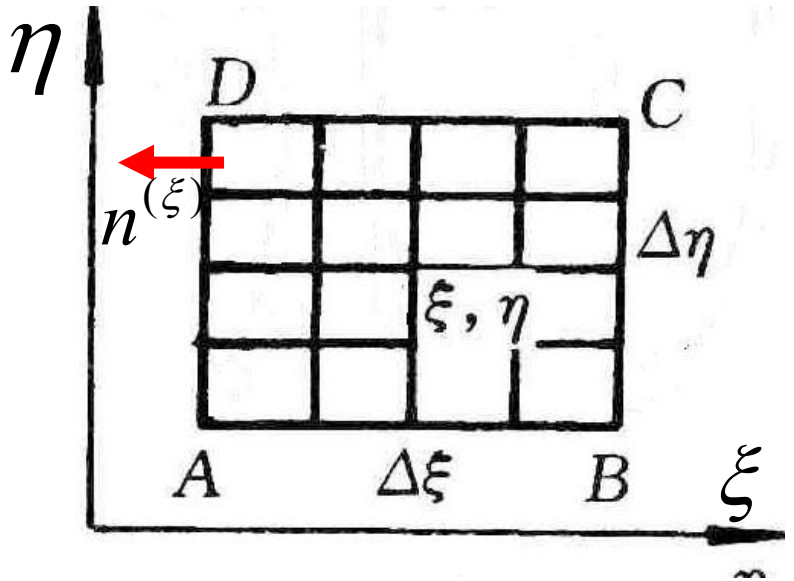
During the transformation from **physical plane** to **computational plane**

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

(2) Physical properties /constants remain unchanged.

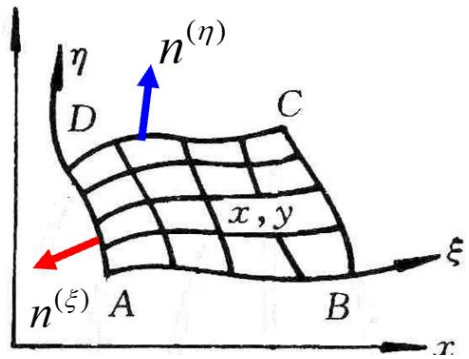
What different is the derivative normal to a boundary in physical plane and in computational plane, i.e.,:





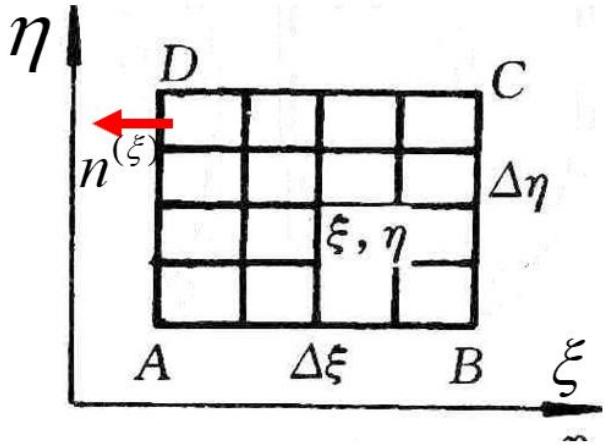
Boundary normal derivative in physical space is not equal to boundary normal derivative in computational space. (物理空间边界上导数和计算空间边界上导数不相等)

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



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

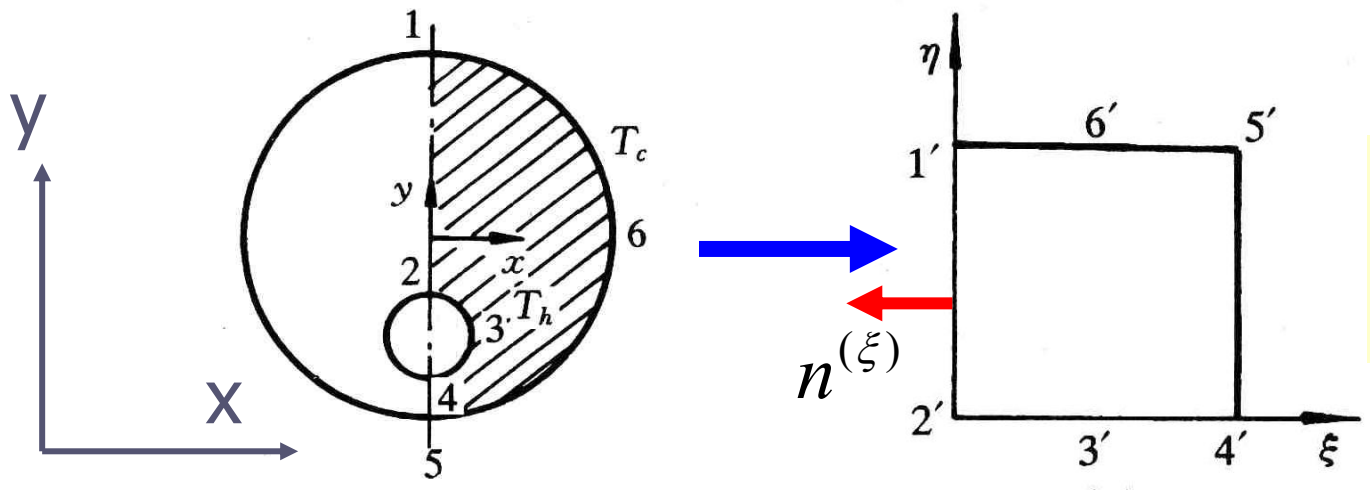
$$\frac{\partial \phi}{\partial n^{(\eta)}} = \frac{\gamma\phi_\eta - \beta\phi_\xi}{J\sqrt{\gamma}}$$



Boundary normal derivative in physical space

ϕ_ξ and ϕ_η are boundary normal derivatives in computational space

Example of B.C. transformation: eccentric (偏心） annulus space



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

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

10.3.2 Transformation of Governing Equations

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:

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

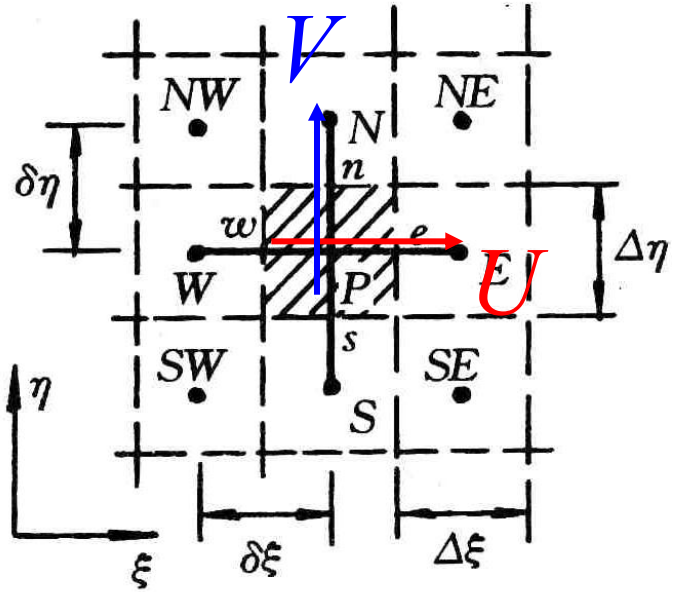
2) J: Jakobi factor，representing variation of volume during transformation:

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

Physical space volume

Computational. space volume

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



3) Coordinate transformation for partial derivatives

$$\frac{\partial \phi}{\partial y} = \xi_y \frac{\partial \phi}{\partial \xi} + \eta_y \frac{\partial \phi}{\partial \eta} = -\frac{x_\eta}{J}\frac{\partial \phi}{\partial \xi} + \frac{x_\xi}{J}\frac{\partial \phi}{\partial \eta}$$

$$\frac{\partial \phi}{\partial x} = \xi_x \frac{\partial \phi}{\partial \xi} + \eta_x \frac{\partial \phi}{\partial \eta} = \frac{y_\eta}{J}\frac{\partial \phi}{\partial \xi} - \frac{y_\xi}{J}\frac{\partial \phi}{\partial \eta}$$

2) Transformation of 2-D diffusion-convection equation in physical Cartesian coordinate to computational plane

Physical:

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

Computational:

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

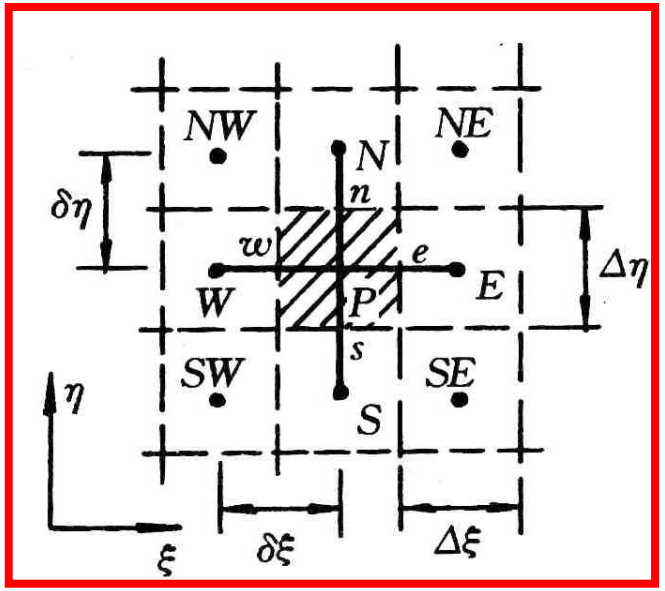
Velocity transformation

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

U, V are velocities in ξ,η direction respectively in comput. plane, called **contravariant velocity** (**逆变速度**);

10.3.3 Discretization in computational plane

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



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

$$[(\rho U\phi)_e-(\rho U\phi)_w]\Delta\eta+[(\rho V\phi)_n-(\rho V\phi)_s]\Delta\xi=$$
$$[\frac{\Gamma_\phi}{J}(\alpha\phi_\xi-\beta\phi_\eta)]_e\Delta\eta-[\frac{\Gamma_\phi}{J}(\alpha\phi_\xi-\beta\phi_\eta)]_w\Delta\eta+$$
$$[\frac{\Gamma_\phi}{J}(-\beta\phi_\xi+\gamma\phi_\eta)]_n\Delta\xi-[\frac{\Gamma_\phi}{J}(-\beta\phi_\xi+\gamma\phi_\eta)]_s\Delta\xi+S\bullet J\bullet\Delta\eta\bullet\Delta\xi$$

cross derivatives
交叉导数项

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

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

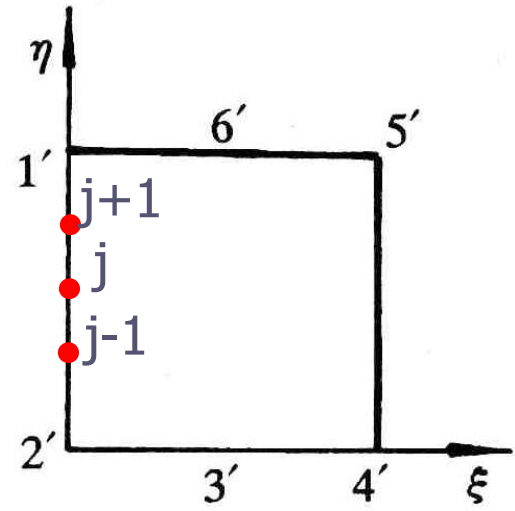
$$b = S_\phi J\Delta\eta\Delta\xi - [(\frac{\Gamma_\phi}{J}\beta\phi_\eta)_w^e\Delta\eta + (\frac{\Gamma_\phi}{J}\beta\phi_\xi)_s^n\Delta\xi]$$

The pressure gradient term is temporarily included in S_ϕ.

Discretization of boundary condition

The key is boundary derivative, for example

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



10.3.4 SIMPLE Algorithm in Computational Plane

1) Choice of velocity in computational space

a. Components in physical plane (u, v)

b. Contravariant velocity (U, V) (逆变分量) $U = uy_\eta - vx_\eta,\quad V = vx_\xi - uy_\xi$

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.

Computational plane:

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

2) Discretized momentum equation in computational plane

(1) Separating pressure gradient from source term

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

Note： cross derivatives occur.

(2) Discretized momentum equation in physical plane

We will follow the discretized form in physical plan:

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

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

Three parts (neighbor, pressure gradient and source) with coefficient.

(3) Discretized *u, v* equations in computational plane

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

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

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

(a) (u_P, v_P) are the velocities at respective locations in the staggered grid.

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

(4) Velocity correction in computational plane

Adopting the same solution procedure as that in the physical plane,

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

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

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

$$U^* = u_P^* y_\eta - v_P^* x_\eta \qquad V^* = v_P^* x_\xi - v_P^* y_\xi$$

The 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：

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

Similarly

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

Omitting the effects of neighboring nodes：

yielding velocity correction：

$$u_P' = B^u p_\xi' + C^u p_\eta'$$

$$v_P' = B^v p_\xi' + C^v p_\eta'$$

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

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

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

0

New assumption: cross derivatives in contravariant velocity are neglected

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

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

At location of V_P

At location of U_P

(5) Pressure correction equation in computational plane

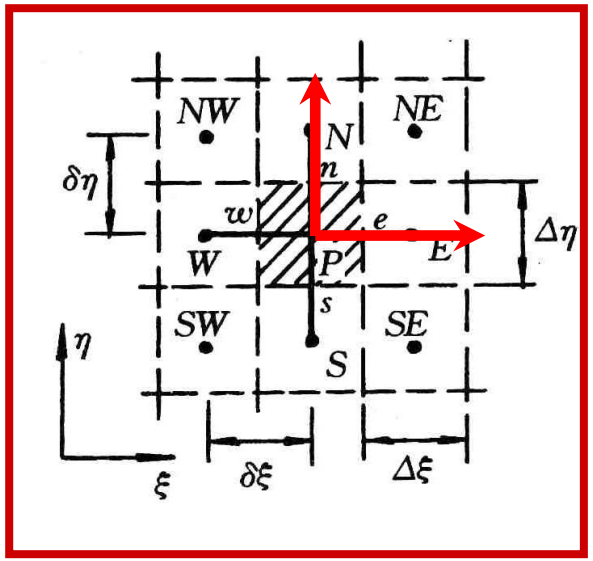
Discretized mass conservation in computational plane

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

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

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':

$$A_P p'_P = A_E p'_E + A_W p'_W + A_N p'_N + A_S p'_S + b$$

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

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

(6) Solution procedure of SIMPLE algorithm in computational plane

1. Assuming 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

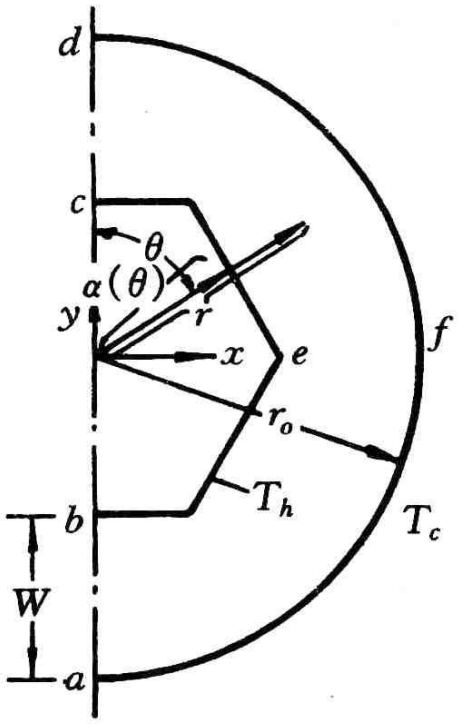
6. Starting next iteration with improved velocity and pressure.

10.3.6. Post-Process and Examples

Data reduction (post process, 后处理) should be conducted for the solutions in the physical plane. The results in the computational plane can not be directly adopted for data reduction by using definition in physical plane.

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

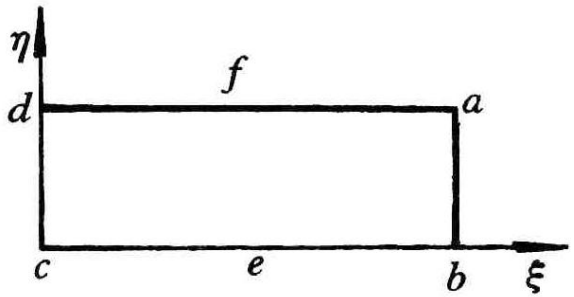
(Physical plane)



$$\xi = \theta$$

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

(Computational domain)



The averaged Nusselt number can be obtained by integration of Nu_i over the inner surface.

$$Nu_i = \frac{h_i W}{\lambda} = \frac{W}{\lambda}[-\lambda(\frac{\partial T}{\partial n})_i \frac{1}{T_h - T_c}] = -[\frac{\partial(\frac{T-T_c}{T_h-T_c})}{\partial(\frac{n}{W})^{(\eta)}}]_i$$

$$= -[\frac{\partial \Theta}{\partial n^{(\eta)}}]_i = -[\frac{\gamma\Theta_\eta - \beta\Theta_\xi}{J\sqrt{\gamma}}]_i$$

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

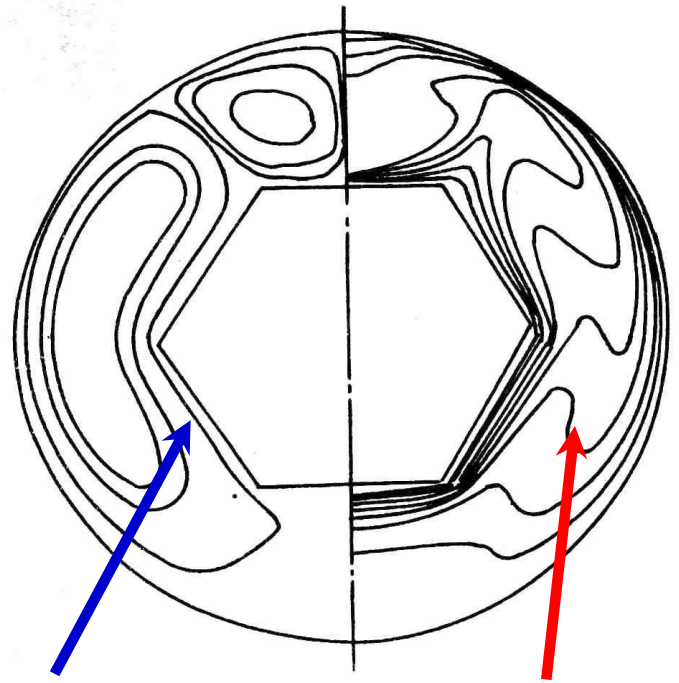
$$\frac{\partial \phi}{\partial n^{(\eta)}} = \frac{\gamma\phi_\eta - \beta\phi_\xi}{J\sqrt{\gamma}}$$

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

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

$$Nu_i = -[\frac{\gamma\Theta_\eta - \beta\Theta_\xi}{J\sqrt{\gamma}}]_i \rightarrow Nu_i = -(\frac{\gamma\Theta_\eta}{J\sqrt{\gamma}})_i$$

$Ra = 9.2\times10^4$



Stream lines Isotherms

