Numerical Heat Transfer

(数值传热学)

Chapter 6 Primitive Variable Methods for Elliptic Flow and Heat Transfer（2）

Instructor Tao, Wen-Quan

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

Xi'an, 2017-Oct.-25

Brief review of 2017-10-25 lecture key points

1. Three kinds of numerical instabilities

1) Instability of transient explicit scheme---oscillating

2) Instability of solution procedure of ABEqs.---no solution

3) Instability of discretized convective term---oscillating

2. Sign preservation principle

1) The iterative solution procedure of the discretized diffusion-convection equation is modeled by the marching process of the explicit scheme of an initial problem;

2) The studied scheme is used to discretize the convection term and CD for the diffusion term; The discrete disturbance method is used to analyze the transfer of

a disturbance based on this discretized equation;

3）Stability of the scheme requires that the effect of any disturbance at any time level on the neighboring point at the next time level must have the same sign.

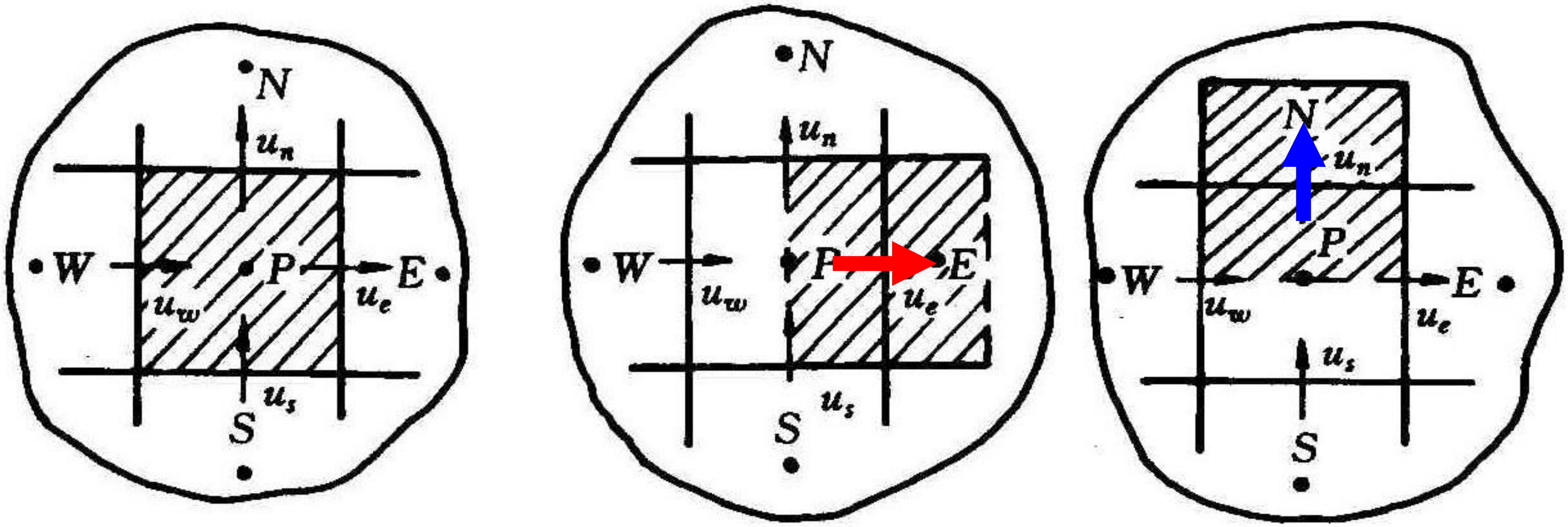
$$\phi_{i\pm1}^{n+1} / \varepsilon_i^n \geq 0$$ **(Sign preservation principle，SPP)**

3. Two key issues in numerically solving discretized momentum equations of incompressible flows

(1) On the conventional grid system the discretized momentum equation can not detect an unphysical pressure profile.

(2) Pressure does not have its own governing equation. To improve an assumed pressure field, a specially-designed algorithm has to be proposed.

4. Staggered grid is adopted to eliminate the occurrence of checkerboard pressure field



Main grid $\qquad$ ***u*** **-grid** $\qquad$ ***v*** **-grid**

5. Interpolations in staggered grid

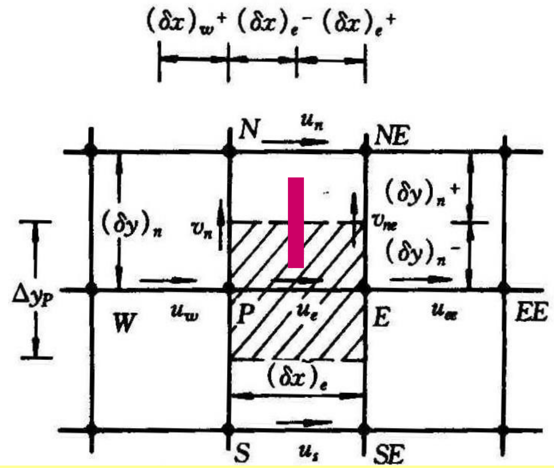
$$D_{n-e} = \frac{(\delta x)_{e^-}}{\dfrac{(\delta y)_n}{\Gamma_n}} + \frac{(\delta x)_{e^+}}{\dfrac{(\delta y)_n}{\Gamma_{ne}}}$$

Parallel conductances

$$D_{n-e}=\frac{(\delta x)_{e^-}}{\dfrac{(\delta y)_n}{\Gamma_n}}+\frac{(\delta x)_{e^+}}{\dfrac{(\delta y)_n}{\Gamma_{ne}}}=\frac{(\delta x)_{e^-}}{\dfrac{(\delta y)_{n^-}}{\Gamma_P}+\dfrac{(\delta y)_{n^+}}{\Gamma_N}}+\frac{(\delta x)_{e^+}}{\dfrac{(\delta y)_n}{\Gamma_{ne}}}$$

Resistances in series

$$=\frac{(\delta x)_{e^-}}{\dfrac{(\delta y)_{n^-}}{\Gamma_P}+\dfrac{(\delta y)_{n^+}}{\Gamma_N}}+\frac{(\delta x)_{e^+}}{\dfrac{(\delta y)_{n^-}}{\Gamma_E}+\dfrac{(\delta y)_{n^+}}{\Gamma_{NE}}}$$



6. Basic idea of pressure correction methods

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

Chapter 6 Primitive Variable Methods for Elliptic Flow and Heat Transfer

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

6.2 Staggered grid system and discretization of momentum equation

6.3 Pressure correction methods for N-S equation

6.4 Approximations in SIMPLE algorithm

6.5 Discussion on SIMPLE algorithm and criteria for convergence

6.6 Developments of SIMPLE algorithm

6.7 Boundary condition treatments for open system

6.8 Fluid flow & heat transfer in a closed system

6.4 Approximations in SIMPLE algorithm

6.4.1 Calculation procedure of SIMPLE algorithm

6.4.2 Approximations in SIMPLE algorithm

1.Inconsistenacy (不一致性)of initial field assumptions

2.Overestimating (夸大) the effects of pressure correction of neighboring nodes

6.4.3 Numerical example

6.4 Approximations in SIMPLE Algorithm

6.4.1 Calculation procedure of SIMPLE algorithm

1. Assuming initial velocity fields, u^0 and v^0, to determine coefficients of momentum equations;

2. Assuming an initial pressure field, p^*;

3. Solving discretized momentum equation based on p^*, obtaining u^*, v^*;

4. Solving pressure correction equation, obtaining p';

5. Revising pressure and velocities by p': $p = p^* + \alpha p'$

$$u = u_e^* + u_e^{'} = u_e^* + d_e \Delta p_e^{'} \qquad v = v_n^* + v_n^{'} = v_n^* + d_n \Delta p_n^{'}$$

6a.Solving other scalar variables coupled with velocity;

6b.Starting next iteration with $u = u_e^* + u_e^{'}$, $v = v_n^* + v_n^{'}$ **and** $p = p^* + \alpha_p p^{'}$ **as the solutions of the present iteration.**

In the following discussion focus will be paid on the solution of flow field, and step 6a will be ignored. The entire solution procedure is composed of six steps.

SIMPLE=Semi-implicit method for pressure-linked equations(求解压力耦合问题的半隐方法)-

where “semi-implicit” refers to the neglect of velocity correction effects of neighboring grids.

6.4.2 Approximations in SIMPLE algorithm

SIMPLE is the dominant algorithm for solving incompressible flows. It was proposed in 1972. Since then many variants(改进方案) were proposed to improve following two assumptions

1.Inconsistency (不一致性) of initial field assumptions

In SIMPLE u^0,v^0,and p^* are assumed independently. Actually there is some inherent (固有的） relation between velocity and pressure;

2.Overestimating the effects of pressure correction of neighboring nodes. Because u_e' is caused by both the pressure correction and velocity corrections of its

neighboring nodes. The neglect of velocity corrections of neighboring nodes attributes (归结于）the driving force of u_e' totally to pressure correction, thus exaggerating (夸大) the action of pressure correction.

6.4.3 Numerical example

[Example 6-1]

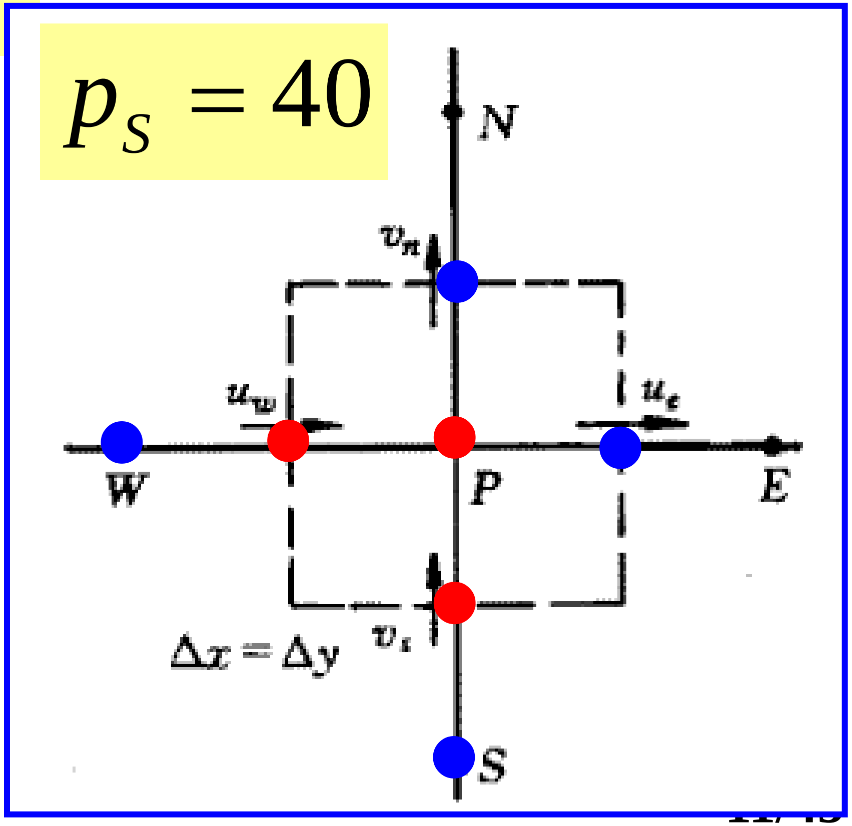
Known:

$$p_W, p_S, u_e, v_n$$

$$u_w = 0.7(p_W - p_P)$$

$$v_s = 0.6(p_S - p_P)$$

Find: p_P, u_w, v_s



The key to solve Example 6-1: how to understand :

$$u_w = 0.7(p_W - p_P)$$

$$v_s = 0.6\,(p_S - p_P)$$

They should be regarded as follows

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

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

For this example

$$u_w = 0 + 0.7(p_W - p_P) \quad \overline{u_w^*} = 0 \quad \boxed{d_w = 0.7}$$

Similarly, $\boxed{d_s = 0.6}$

6.5 Discussion on SIMPLE and Convergence Criteria of Flow Field Iteration

6.5.1 Discussion on SIMPLE algorithm

1.Can the simplification approximations affect the computational results?

2.Mathematically what type does the boundary condition of the pressure correction equation belong to ?

3.How to adopt the underrelaxation method in the flow filed iteration process?

6.5.2 Convergence criteria of flow field iteration

6.5 Discussion on SIMPLE and Convergence Criteria of Flow Field Iteration

6.5.1 Discussion on SIMPLE algorithm

1.Can the simplification approximations affect the computational results?

The approximations of SIMPLE will not affect the converged solution , but do affect the convergence speed for the following reasons:

(1) The inconsistency between u^0, v^0, p^* **will be gradually eliminated with the proceeding of iteration;**

(2) The term $\sum u'_{nb}$ **in** u'_e **will gradually disappear (消失) with the proceeding of iteration.**

2. What type does the boundary condition of the pressure correction equation belong to？

(1) Mathematically the boundary condition of the pressure correction equation is Newmann condition，

$$\frac{\partial p'}{\partial n} = 0$$

—Gresho question（1991： A simple question to SIMPLE users)

(2) The adiabatic type boundary condition of the pressure correction equation can uniquely(唯一地) define an incompressible flow problem, because pressure exists in the N-S equation in terms of gradient!

$$\left.\begin{aligned} &\vec{U} \bullet \nabla \vec{U} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{U} \\ &\nabla \bullet \vec{U} = 0 \\ &\text{No slip on the boundary} \end{aligned}\right\}$$

can uniquely define a flow field.

(3) The boundary condition of the pressure correction equation makes the ABEqs. being linearly dependent (线性相关)，and the coefficient matrix is singular (奇异)；In order to get a unique solution the compatibility condition (相容性条件) must be satisfied：the sum of the right terms of the ABEqs. should be zero.

$$a_P p_P' = a_E p_E' + a_W p_W' + a_N p_N' + a_S p_S' + b$$

$$a_P p_P' - (a_E p_E' + a_W p_W' + a_N p_N' + a_S p_S') = b$$

Right term

$$\sum b_{i,j,k} = 0$$

Mass conservation of the entire domain.

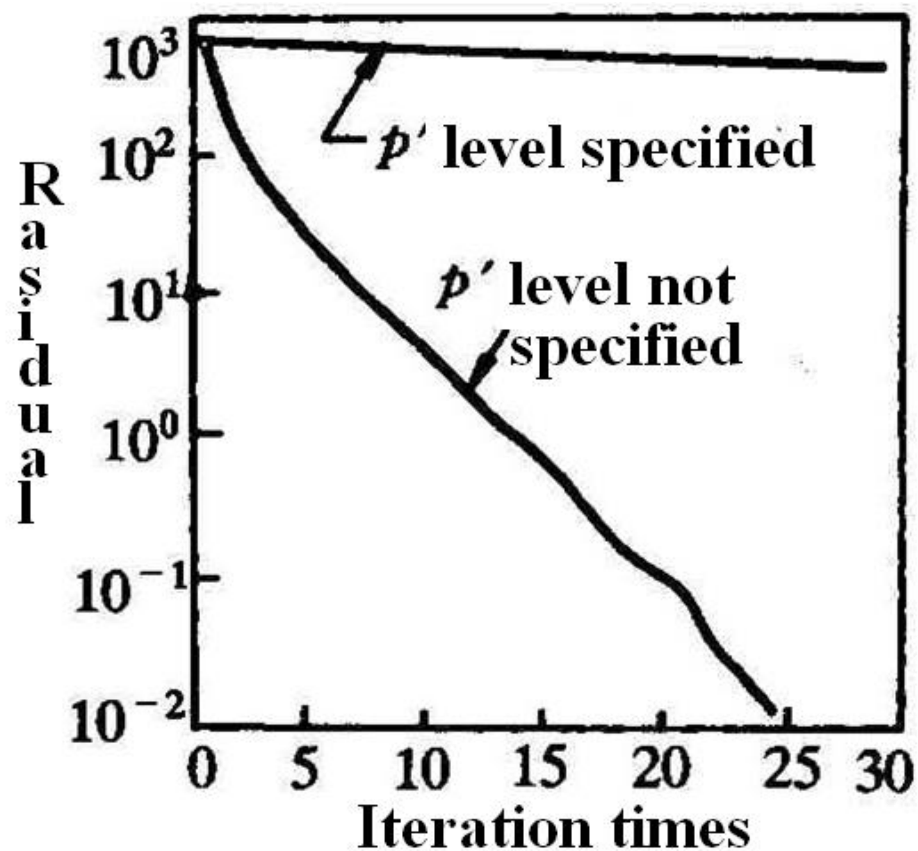
Thus the requirement of mass conservation at each iteration level corresponds to the execution of Neumann boundary condition.

In our teaching program **RMAX**, **SSUM** represent b_{max} and $\sum b_{i,j}$,respectively.

(4) Determination of absolute pressure

For Neumann condition, p ’should be determined by computation , rather than specified in advance.

After receiving the converged solution, selecting some point as a reference and using the relative results as output.



3.How to adopt the underrelaxation in solving flow fields?

(1) Underrelaxation of pressure correction *p* ’ :

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

α_p **--pressure underrelaxation factor**

(2) Underrelaxation of velocity is organized into the solution procedure:

Iteration process is generally expressed as:

$$\phi_P = \phi_P^0 + \alpha\left[\frac{\sum a_{nb}\phi_{nb} + b}{a_P} - \phi_P^0\right]$$

$$\underline{\left(\frac{a_P}{\alpha}\right)}\phi_P = \sum a_{nb}\phi_{nb} + \underline{b + (1-\alpha)\frac{a_P}{\alpha}\phi_P^0}$$

a_P **of new eq.**

b **of new eq.**

The obtained numerical results are underrelaxed!

Discussion: Can the direct underrelaxation be used for velocity?

$$u = u^* + \alpha_u u'$$ **No! ! !**

Reason: The velocity correction is obtained through mass conservation requirement. Its underrelaxation will violate (破坏) mass conservation condition. Thus incorporating (纳入) the underrelaxation of velocity into solution procedure is necessary!

6.5.2 Convergence criteria of flow field iteration

1.Two different iterations

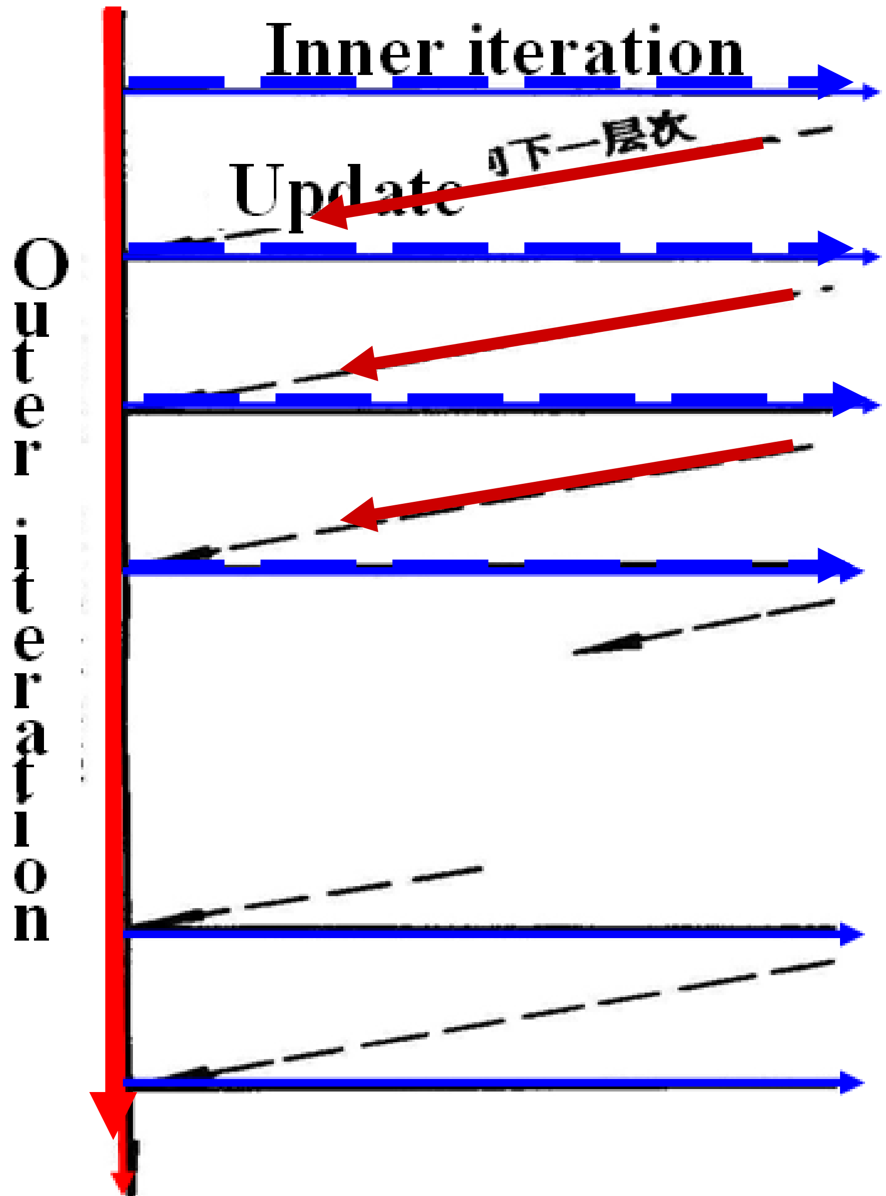
(1) Iteration for solving ABEqs.－ Inner iteration

This is the solution procedure for ABEqs. with specified coefficients and source term.

(2) Iteration for non-linear problem－Outer iteration

This is the process in which the coefficients and source term are updated.

Linear algebraic



2. Criteria for terminating inner iteration

The major solution work for flow field is in the *p* ’eqs. . Terminating too early is not in favor of (不利于)mass conservation ,while too late is not economic. Three criteria may be used：

(1) Specify the number of iteration cycles：One cycle means that the dependent variables at all nodes have been updated. ----Simple but not rational(合理的)；

(2) Specify a threshold (阈值）for the norm(范数) of residual (余量) of p’ eqs.

$$\{\sum[\underline{(a_P p_P^{'} - \sum a_{nb} p_{nb}^{'} - b)^{(k)}}]^2\}^{1/2} = R_p^{(k)}$$

$$\boxed{R_p^{(k)} \le \varepsilon}$$

Zero if converged

Residual may be negative

Resume to original dimension

(3) Specify a threshold for the ratio of residuals (余量) of p' equations:

$$R_p^{(k)} / R_p^{(0)} \le r_p,\ r_p = 0.05 \sim 0.25$$

3. Criteria for terminating outer iteration

(1) Specify a threshold of relative deviation of some quantity

$$\left| \frac{Nu^{(k+n)} - Nu^{(k)}}{Nu^{(k+n)}} \right| \le \varepsilon \qquad n = 1 \sim 100$$

Remarks:

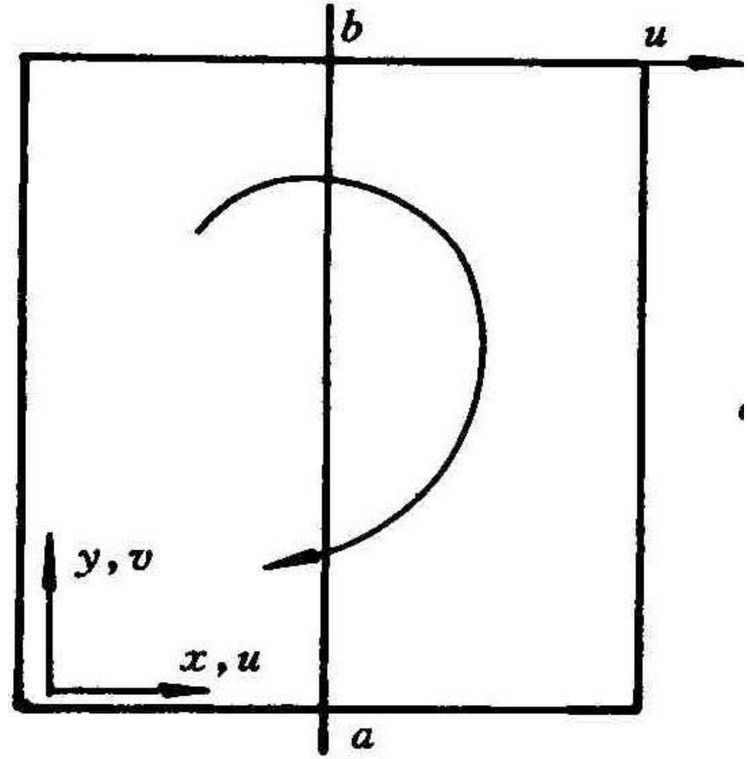
The smaller the α, the smaller the value of ε should be.

(2) Specify thresholds for SSUM and RMAX, respectively:

$$\boxed{\frac{|R_{SSUM}|}{q_m} \le \varepsilon_1; \quad \frac{R_{MAX}}{q_m} \le \varepsilon_2}$$

q_m **-reference flow rate; For open system the inlet flow rate may be used; for closed system, following definition may be used:**

$$q_m = \int_a^b \rho |u| dy$$



For open system if the mass conservation is forced to be satisfied, then $R_{SUM}/q_m \le \varepsilon_1$ **can not be used as a convergence criterion.**

(3) Relative norm (范数)of mass conservation residual less than an allowed value:

$$\frac{\sqrt{\sum (b)^2}}{q_m} \le \varepsilon$$

Remarks:

Residual of mass conservation is:

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

Residual of *p* ' equation is: $(a_P p_P' - \sum a_{nb} p_{nb}' - b)$

(4) Relative norm of momentum equation residual less than an allowed value:

Norm of momentum equation residual $=(\sum\{a_e u_e-[\sum a_{nb}u_{nb}+b+A_e(p_P-p_E)]\}^2)^{1/2}$

$$(\sum\{a_e u_e-[\sum a_{nb}u_{nb}+b+A_e(p_P-p_E)]\}^2)^{1/2}/(\rho u_{in}^2)\le\varepsilon$$

Zero if converged

Residual may be less than 0

dimensionless

Resume to original dimension

$$\boxed{\varepsilon\approx 10^{-3}\sim 10^{-6}}$$

A better criterion is: relative norms of both mass conservation and momentum equation less than allowed values.

6.6 Developments of SIMPLE algorithm

6.6.1 SIMPLER—Overcoming 1st assumption of SIMPLE（1980)

6.6.2 SIMPLEC—Partially overcoming 2nd assumption of SIMPLE（1984)

6.6.3 SIMPLEX— Partially overcoming 2nd assumption of SIMPLE （1986)

6.6.4 Comparisons of algorithms

6.6 Developments of SIMPLE algorithm

6.6.1 SIMPLER－Overcoming 1st assumption of SIMPLE（1980）

1.Basic idea

Pressure field is solved from the assumed velocity field, rather than assumed independently.

p ’ is used to correct velocity, but not pressure. The improved pressure is solve from updated velocity field.

2. How to get pressure field from given velocity field

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

Rewritten in terms of v

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

$$u_e = \tilde{u} + (\frac{A_e}{a_e})(p_P - p_E) = \tilde{u} + d_e(p_P - p_E); \quad v_n = \tilde{v}_n + d_n(p_P - p_N)$$

$\tilde{u}_e$ is called pseudo-velocity(假拟速度).

Substituting u_e, v_n into continuum equation and re-arranging:

$$a_P p_P = a_E p_E + a_W p_W + a_N p_N + a_S p_S + b$$

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

Equations for $a_E \sim a_S, a_P$ are the same as that for *p*'.

3. Boundary condition of *p*-equation

The same as for *p*': zero coefficients of boundary neighbor node.

4. Calculation procedure of SIMPLER

(1) Asuuming initial field u^0, v^0, determining coefficient, b and pseudo-velocity $\tilde{u}, \tilde{v}$;

(2) Solving pressure equations, and taking the results as p^*;

(3) Solving discretized momentum equations ,and taking the results as u^*, v^*;

(4) Solving pressure correction equations, yielding p';

(5) Correcting velocity from p' , yielding u', v';

(6) Taking （$u^* + u'$),($v^* + v'$) as the flow solution of the present level and starting iteration for next level.

5. Discussion on SIMPLER algorithm

SIMPLER=SIMPLE REVISED—Patankar

(1) At each iteration level two pressure equations are solved , hence more computational time is needed for each iteration. However, the improved consistency between initial flow and pressure fields makes the total iteration times often shorter.

(2) In SIMPLER no any effort is taken to overcome the 2nd assumption; In addition a new inconsistency is introduced: pressure is always determined from the previous flow field.

6.6.2 SIMPLEC－Partially overcoming the 2nd assumption (1984)

1.Basic idea

In SIMPLE some inconsistency is introduced when neglecting the velocity correction term of neighbour nodes :neglecting $\sum a_{nb}u_{nb}^{'}$ is equivalent to let $a_{nb} \rightarrow 0$,while in the main diagonal term , i.e, in $a_P = \sum a_{nb} - S_P \Delta V$ no any correspondent action is taken.

2. A more consistent treatment

At the two sides of the $u^{'} - p^{'}$ equation

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

subtracting the term $\sum a_{nb}u_e^{'}$ from both sudes yielding:

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

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

It can be expected that: u_e', u_{nb}' are in the same order of magnitude, $\sum a_{nb}(u_{nb}' - u_e')$ is much smaller than other terms at right side, hence effect of neglecting it will be much smaller that that of neglecting $\sum a_{nb}u_{nb}'$ in SIMPLE algorithm.

$$u_e' = \underbrace{\left(\frac{A_e}{a_e - \sum a_{nb}}\right)}_{\boxed{d_e}}(p_P' - p_E') \qquad v_n' = \underbrace{\left(\frac{A_n}{a_n - \sum a_{nb}}\right)}_{\boxed{d_n}}(p_P' - p_N')$$

This is velocity correction equation in SIMPLEC.

3.Calculation procedure of SIMPLEC

The same as SIMPLE with following two different treatments

(1) The *d* term in velocity correction equation is:

$$d_e = \frac{A_e}{a_e - \sum a_{nb}}; \ d_n = \frac{A_n}{a_n - \sum a_{nb}}$$

(2) No underrelaxation for *p*' .

4. The denominator in *d* will never be zero

Because the underrelaxation of flow field is organized into the solution procedure, the coefficient a_e, a_n in the above equations are actually a_e / α_e and a_n / α_n , respectively! Hence $(a_e / \alpha_e - \sum a_{nb}) > 0$

5. Discussion on SIMPLEC algorithm

SIMPLEC=SIMPLE CONSISTENT, van Doormaal, Raithby(1984)

(1) Through simply improving the coefficient d SIMPLEC partially overcomes the 2nd assumption in SIMPLE without introducing additional computational work ;

(2) Algorithm comparison shows that at a finer grid system SIMPLEC is more efficient.

(3) The inconsistency of initial fields assumption still exists in SIMPLEC, though somwhat alleviated.

6.6.3 SIMPLEX algorithm

1. Basic idea of SIMPLEX (1986, Raithby)

The essential step in SIMPLEC is the improvement of d:

$$d_e = \frac{A_e}{a_e - \sum a_{nb}};\ d_n = \frac{A_n}{a_n - \sum a_{nb}}$$

Extending this idea: If a set of algebraic equation of d can be formed which can take the effects of neighboring nodes into consideration, the iteration may be speeded up

2. Derivation of d-equation

Taking following expression in SIMPLE

$$u_e^{'} = d_e(p_P^{'} - p_E^{'}) = \underline{d_e \Delta p_e^{'}}$$

Introducing: $u'_{nb} = d_{nb} \Delta p'_{nb}$

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

Yielding $a_e d_e \Delta p'_e = \sum a_{nb} d_{nb} \Delta p'_{nb} + A_e \Delta p'_e$

Assuming that $\Delta p'_e = \Delta p'_{nb}$ **A new assumption!**

Then: $a_e d_e \Delta p'_e = \sum a_{nb} d_{nb} \Delta p'_{nb} + A_e \Delta p'_e$ $\longrightarrow$

$$\boxed{a_e d_e = \sum a_{nb} d_{nb} + A_e}$$

ABEqs. for d !

From known coefficients of momentum equations d can be solved.

No neighboring nodes were neglected but a new assumption was introduced $\Delta p'_e = \Delta p'_{nb}$

Boundary condition for d： Zero coefficients of BNNs.

3. Calculation procedure of SIMPLEX

BNNs: boundary neighboring nodes

(1) Assuming initial u^0, v^0, calculating coefficients and , b

(2) Assuming pressure field p^*;

(3) Solving discretized momentum equations, yielding u^*, v^*;

(4) Solving *d* equations， and pressure correction equations, yielding p';

(5) Correcting velocity from p'， yielding u', v';

(6) Taking （u^*+u'),(v^*+v') ,(p^*+p') as the solutions of the present level and starting the iteration for the next level （p' is not under-relaxed.）。

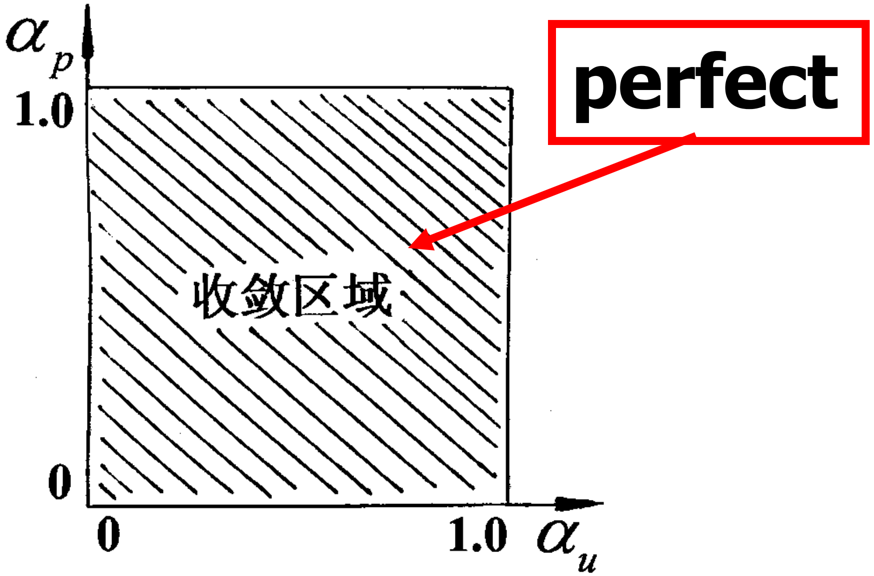
6.6.4 Comparisons of algorithms

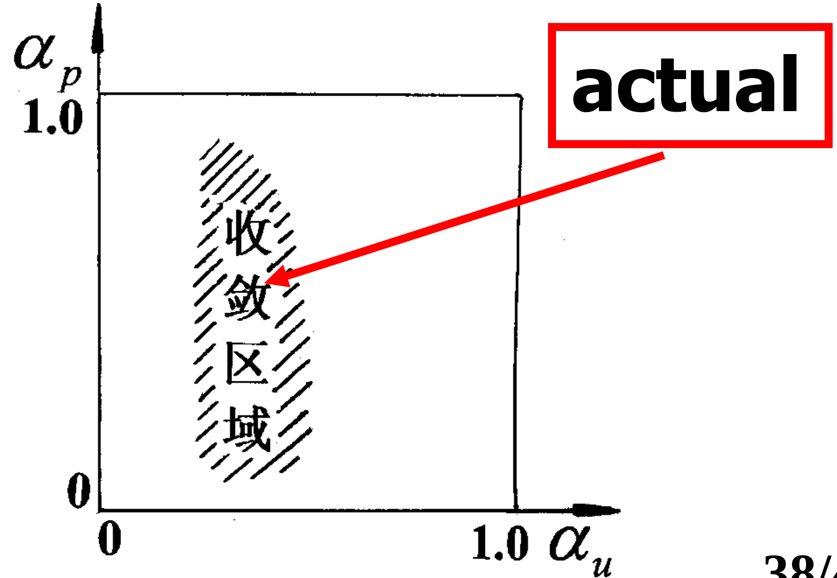
1. Comparison contents

Convergence rate，and robustness (健壮性，鲁棒性)

2. Comparison methods

(1) Adopting graph of $\alpha_u \sim \alpha_p$ **—heavy computational work**





(2) Adopting time step multiple (时步倍率)～iteration time graph

The time step multiple , E, is defined as：

$$\boxed{E = \frac{\alpha}{1-\alpha}}$$

$$\alpha : 0 \to 1$$

$$E : 0 \to \infty$$

It greatly extends the variation range of under-relaxation treatment.

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95
E	0.111	0.25	0.428	0.66	1	1.5	2.33	4	9	19

$$\alpha = 0.999 \quad E = 999$$

3. Comparison conditions

For comparison results being meaningful, it should be conducted under following conditions:
(1)The same grid system; (2) The same convergence criteria; (3)The same discretization scheme; (4)The same solution method for the ABEqs.; (5)The same underrelaxation factors; (6)The same initial fields

4. Remarks

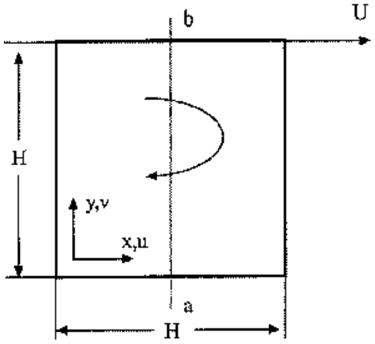
In the comparison of algorithm, the solution and its order of accuracy are the same for all compared algorithms, i.e., different algorithm should have the same numerical results. Algorithm comparison only relates to convergence speed and robustness.

And the comparison of schemes relates to numerical accuracy and computational time. Roughly speaking:：

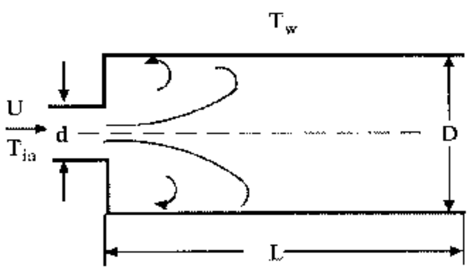
"Algorithm relates to convergence rate, and scheme to solution accuracy".

5.Comparison between SIMPLE, SIMPLER, SIMPLEC, SIMPLEX

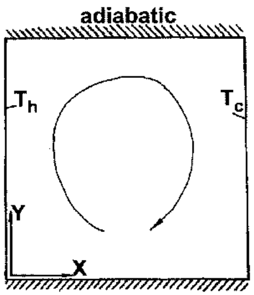
(1) The four problems compared



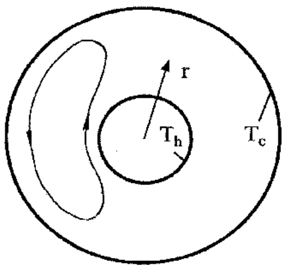
(1)lid-driven cavity flow



(2)flow in a tube with sudden expansion



(3)natural convection in a square cavity



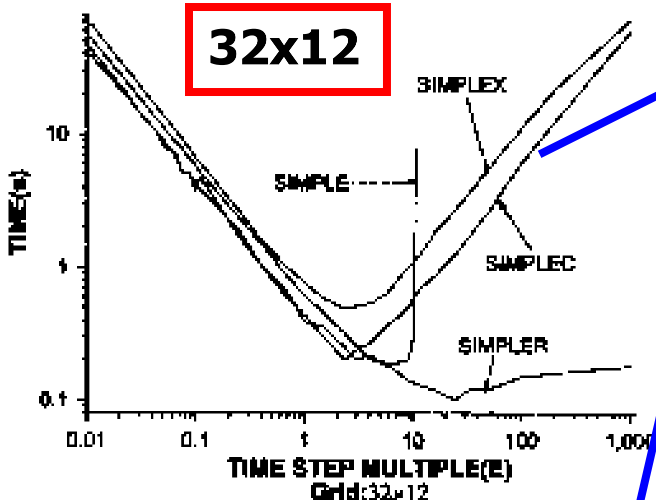
(4)natural convection in a horizontal annular

(2) Comparison results (Example 3)

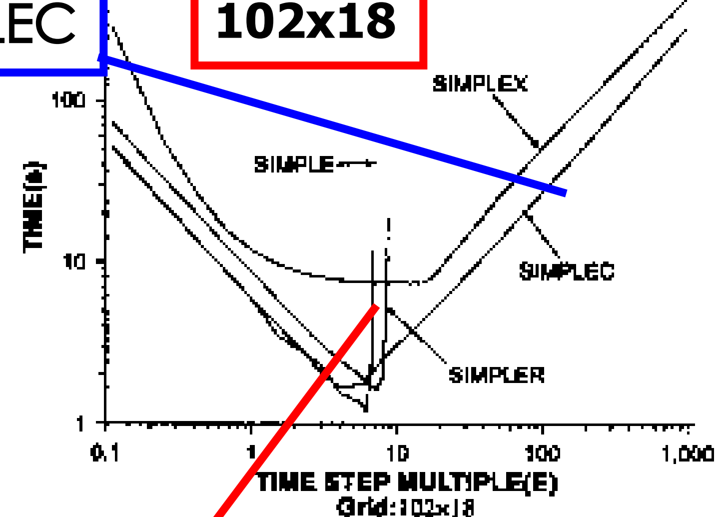
32x12

SIMPLEC

102x18

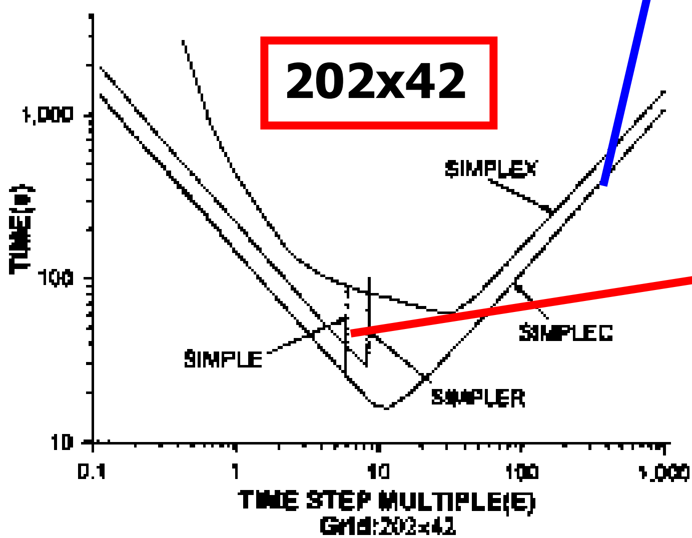


(a) 32 × 12 grid system



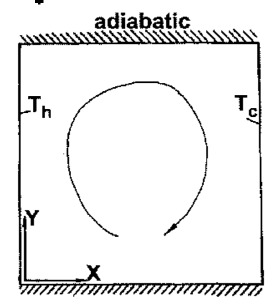
(b) 102 × 18 grid system

202x42



(c) 202 × 42 grid system

SIMPLE
SIMPLER



(3)natural convection in a square cavity

(3) About *d*

$$u' = d(\Delta p') \quad d \uparrow \quad \Delta p' \downarrow$$

Natural convection in a square cavity

	42×42				82×82			
	SIMPLE	SIMPLER	SIMPLEC	SIMPLEX	SIMPLE	SIMPLER	SIMPLEC	SIMPLEX
$d_u(10,6)$	0.5927	0.5927	2.964	2.928	0.2981	0.2981	1.490	1.474
$d_u(20,20)$	0.5960	0.5960	2.980	2.979	0.2975	0.2975	1.488	1.488

Natural convection in a square cavity

	42×42				82×82			
	SIMPLE	SIMPLER	SIMPLEC	SIMPLEX	SIMPLE	SIMPLER	SIMPLEC	SIMPLEX
$d_u(12,7)$	1.929	1.930	9.643	9.525	0.9999	0.9999	4.999	4.976
$d_u(22,22)$	1.874	1.873	9.368	9.265	0.9612	0.9612	4.803	4.798

Thus in SIMPLEC， SIMPLEX no underrelaxation is needed for *p* ’.

Zeng M, Tao W Q. A comparison study of the convergence characteristics and robustness for four variants of SIMPLE family at fine grids. Engineering Computations, 2003, 20(3/4):320-341

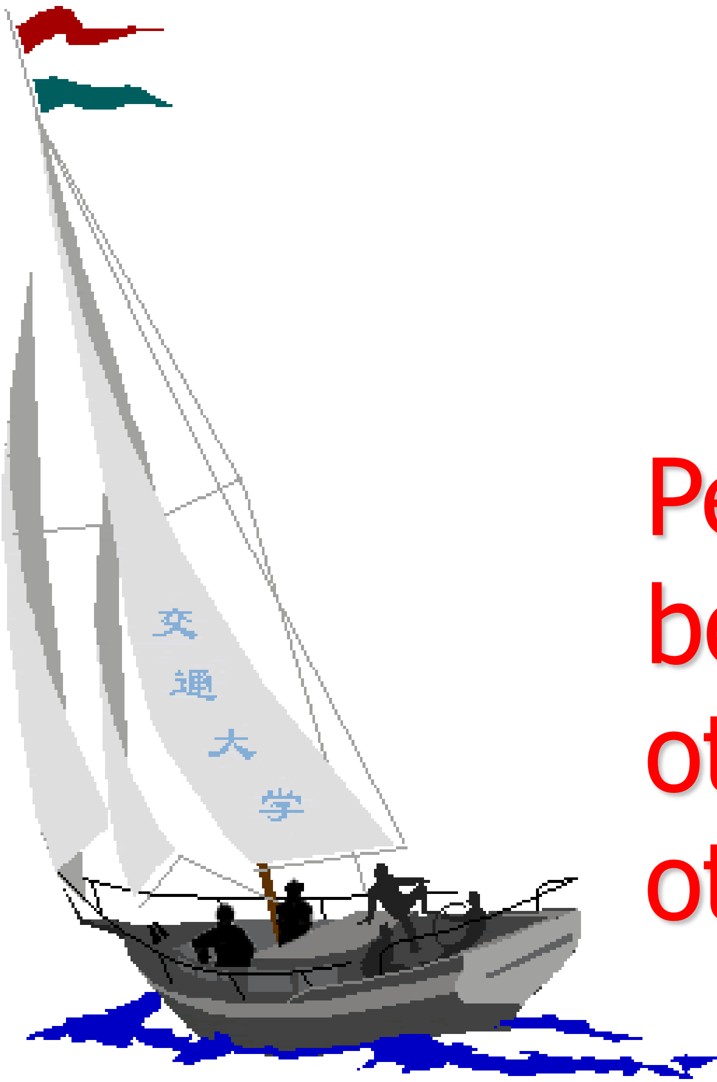
6.6.5 IDEAL algorithms

IDEAL algorithm have completely overcome the two assumptions of SIMPLE alrotithm.

D L Sun, ZG Qu, Y L He ,W Q Tao. An efficient segregated algorithm for incompressible fluid flow and heat transfer problems－IDEAL (inner doubly –iterative efficient algorithm for liked equations) Part I:Mathematical formulation and solution procedure, Numerical Heat Transfer, Part B, 2008, 53(1);1-17

Tao WQ, Qu ZG, He YL , An efficient segregated algorithm for incompressible fluid flow and heat transfer problems－IDEAL (inner doubly –iterative efficient algorithm for liked equations) Part II: Application examples ,Numerical Heat Transfer, Part B, 2008, 53(1);18-38

2-D DEAL code can be found in our website.



同舟共济

渡彼岸！

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