Numerical Heat Transfer

(数值传热学)

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

Instructor Tao, Wen-Quan

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

2024-October-22

6.4 Approximations made 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 the coefficients of the discretized momentum equations;
2. Assuming an initial pressure field, p^*;
3. Solving the discretized momentum equation based on p^*, obtaining u^*, v^*;
4. Solving pressure correction equation, obtaining p';
5. Revising pressure and velocities by p': $p = p^* + \alpha_p p'$

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

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 flow field at present iteration level.

In the following discussion focus will be paid on the solution of flow field, and Step 6a will be ignored （忽略）. 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 the velocity correction effects of the 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 (方案) were proposed to improve (overcome) 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) relationship between velocity and pressure;

2.Overestimating (夸大)the effects of pressure correction of the 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 the 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(Text book)]

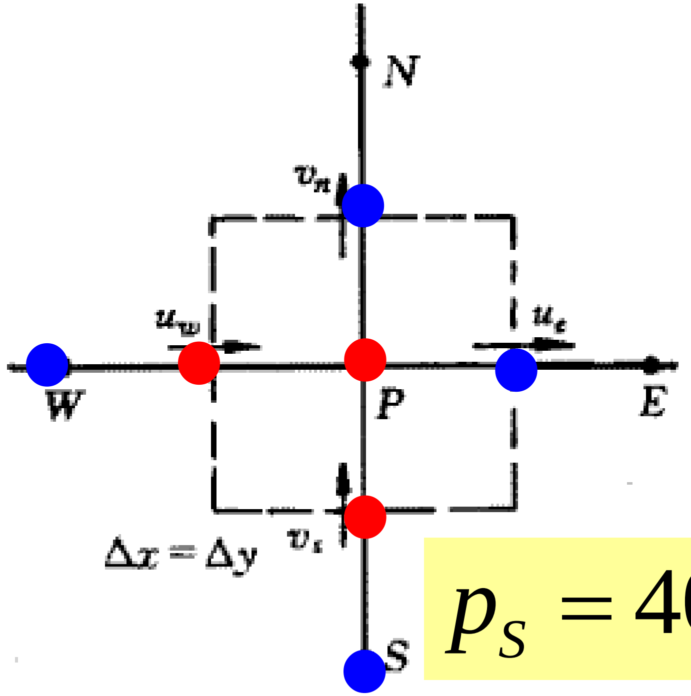
Known:

$$p_W, p_S, u_e, v_n$$

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

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

Find: p_P, u_w, v_s



$p_S = 40$

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

$$u_w = 0.7(p_W - p_P) \quad v_s = 0.6\,(p_S - p_P)$$

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

$$u_w = 0 + 0.7(p_W - p_P)$$

$$\overline{u_w^*} = 0$$

$$d_w = 0.7$$

Similarly, $d_s = 0.6$

With this understanding , the problem can be solved according to the solution steps of SIMPLE algorithm. See Textbook.

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 field 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 two simplified approximations affect the computational results (solution accuracy) ?

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（随着迭代的进行）；

(2) The term $\sum u_{nb}^{'}$ in $u_e^{'}$ will gradually approach zero (趋近于0) with the proceeding of iteration if it converges!

2. What type does the boundary condition of the pressure correction equation belong to？ Can it uniquely determine the solution?

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

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

－Gresho’s 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!

$$\vec{U} \bullet \nabla \vec{U} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \vec{U}$$

$$\nabla \bullet \vec{U} = 0$$

No slip on the boundary

This formulation can uniquely define the flow field.

Thinking question: Why for temperature not?

(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 (右端项) of the ABEqs. should be zero.

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

Mathematically →

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

Right term

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

→ Mass conservation of the entire domain should be satisfied.

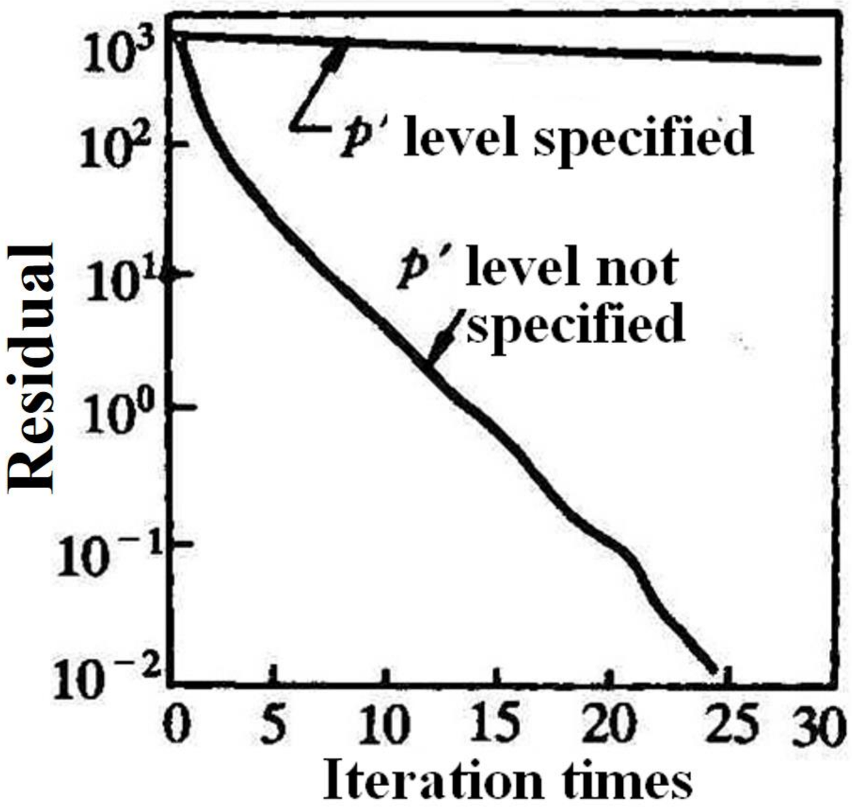
Thus the requirement of mass conservation at each iteration level corresponds to the execution of Neumann boundary condition for the satisfaction of compatibility requirement!

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 for non-linearity

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

Iteration process is generally expressed as:

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

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

a_P of new eq.

b of new eq.

The obtained numerical results have already been underrelaxed!

Discussion: Can the direct underrelaxation be used for velocity?

$$u = u^* + \alpha_u u'$$

No！ No! 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

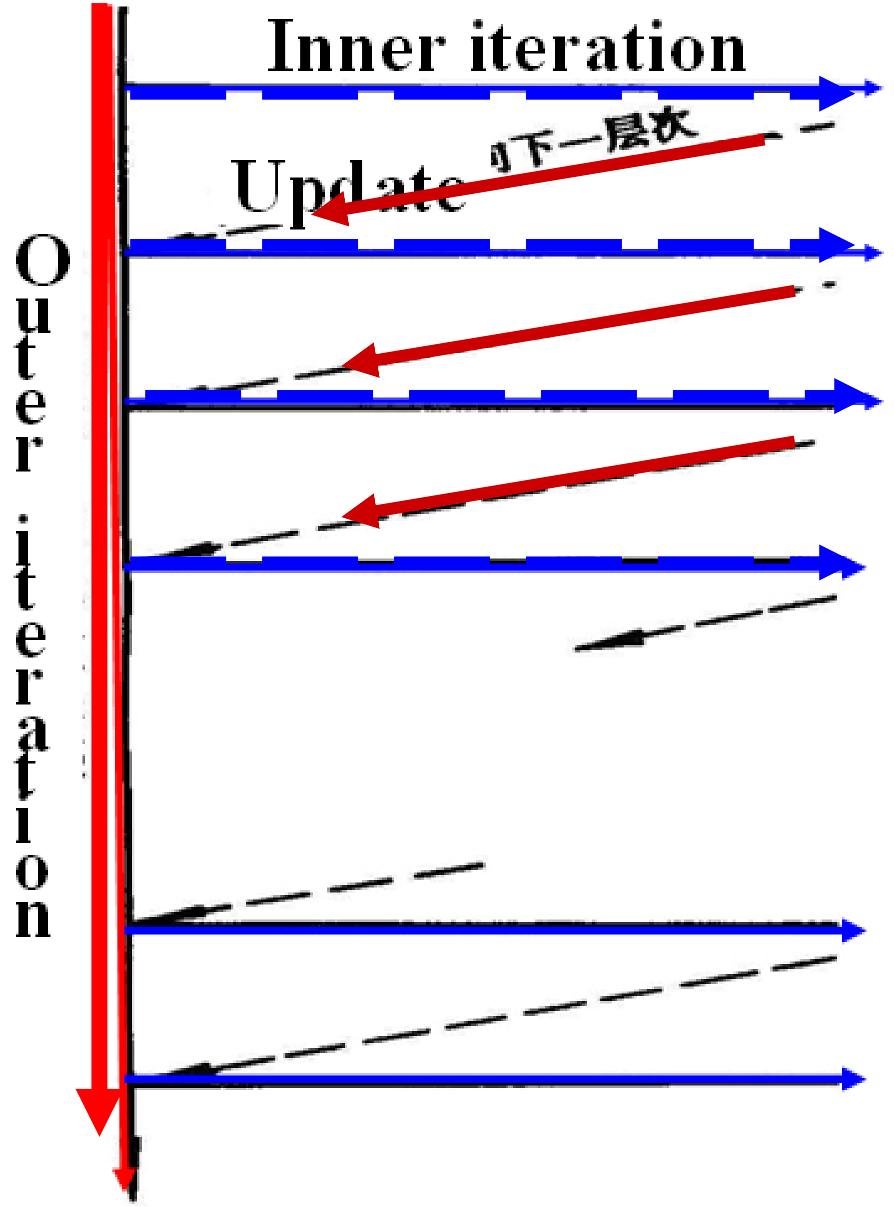
(1) Iteration for solving ABEqs.— Inner iteration

This is the solution procedure for ABEqs. with specified coefficients and source term. Discussed in Chapter 5

(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 of flow field solution

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.

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

Zero if converged

Residual may be negative

Resume to (回复到)original dimension

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

(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 under-relaxation factor α, the smaller the value of ε should be!

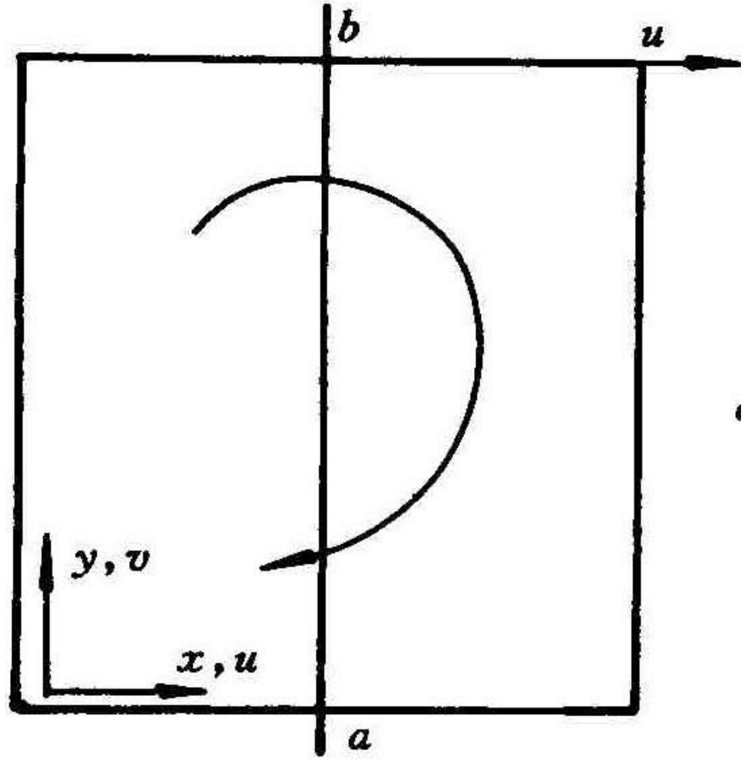
(2) Specify thresholds for SSUM and RMAX, respectively:

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

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 an 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

Resuming to original dimension

dimensionless

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

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 u →

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

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

u_e is called pseudo-velocity(假拟速度).

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

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

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

Expressions 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) Assuming 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) Making correction of velocity from p', yielding u', v'; but not for pressure;

(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 (1980)

(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 sides 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 than that of neglecting $\sum a_{nb}u_{nb}'$ in SIMPLE algorithm.

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

$$d_e \qquad\qquad d_n$$

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 2^{nd} assumption in SIMPLE without introducing additional computational work；

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

(3) The inconsistency of initial fields assumption still exists in SIMPLEC.

X.L. Liu, W.Q. Tao and Y.L. He. A simple method for improving the SIMPLER algorithm for numerical simulations of incompressible fluid flow and heat transfer problems. Engineering Computations, 2005, 22: 921-939

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 the neighboring nodes into consideration, the iteration may be speeded up

2. Derivation of *d*-equation

Taking following expression in SIMPLE **a**lgorithm (算法)

$$u_e' = d_e(p_P' - p_E') = \underline{d_e \Delta p}_e'$$

as a general expression, then

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

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

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

Algebraic equations 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 the boundary-neighboring nodes.

3. Calculation procedure of SIMPLEX

(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.) .

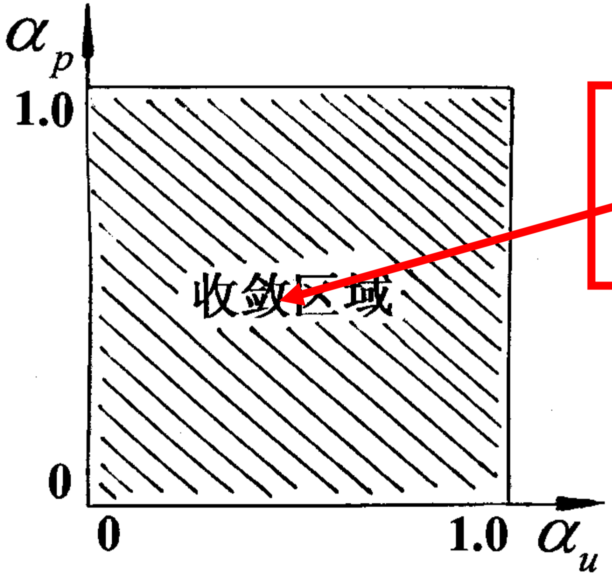
6.6.4 Comparisons of algorithms

1. Comparison contents

Convergence rate，and robu**u**stness **(健壮性，鲁棒性)**

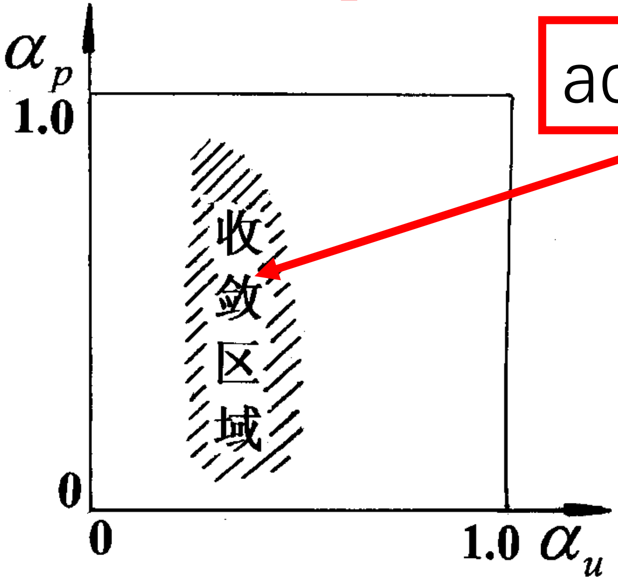
2. Comparison methods

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



Perfect robustness

actual



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

The time step multiple , E, is defined as：

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

$$\alpha=0,\quad E=0$$

$$\alpha\to 1,\ E\to\infty$$

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

$$\alpha=0.999\qquad E=999$$

4. Comparison of four algorithms

Algorithm	1st assumption	2nd assumption
SIMPLE	p^* is independently assumed, improved by a correction term, under-relaxation is needed	$\sum a_{nb}u'_{nb}=0, d_e=A_e/a_e$
SIMPLER	p^* is determined from velocity, pressure correction term is not used to correct pressure	$\sum a_{nb}u'_{nb}=0, d_e=A_e/a_e$
SIMPLEC	p^* is independently assumed, under-relaxation of the pressure correction is not needed.	$\sum a_{nb}(u'_{nb}-u'_e)=0,$ $d_e=A_e/(a_e-\sum a_{nb})$ Partially overcome 2nd assumpt.
SIMPLEX	p^* is independently assumed, under-relaxation of the pressure correction is not needed.	$\Delta p'_e=\Delta p'_{nb}$ d_e, d_n are solved by ABEqs Partially overcome 2nd assumpt.

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

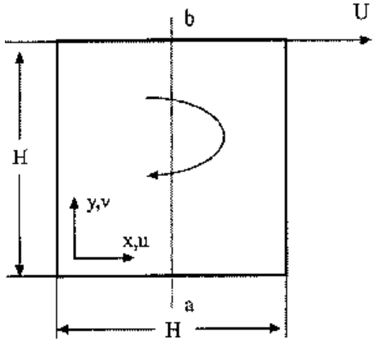
In the comparison of algorithms, the solved results and its order of accuracy are the same for all compared algorithms, i.e., different algorithms should have the same numerical results. Algorithm comparison only relates to convergence speed and robustness.

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

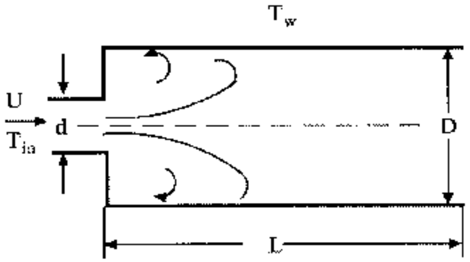
"Algorithm relates to convergence rate, and scheme to solution accuracy".(算法比速度，格式比精度)

5.Comparison four examples between SIMPLE, SIMPLER, SIMPLEC, SIMPLEX

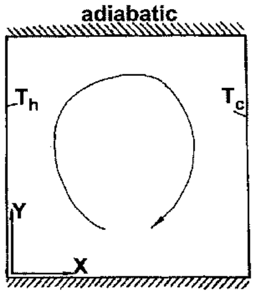
(1) The four problems compared



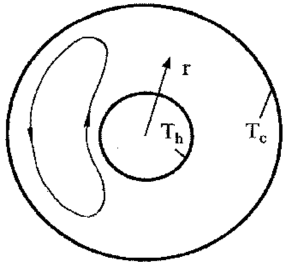
(1)lid-driven cavity flow



(2)flow in a tube with sudden expansion

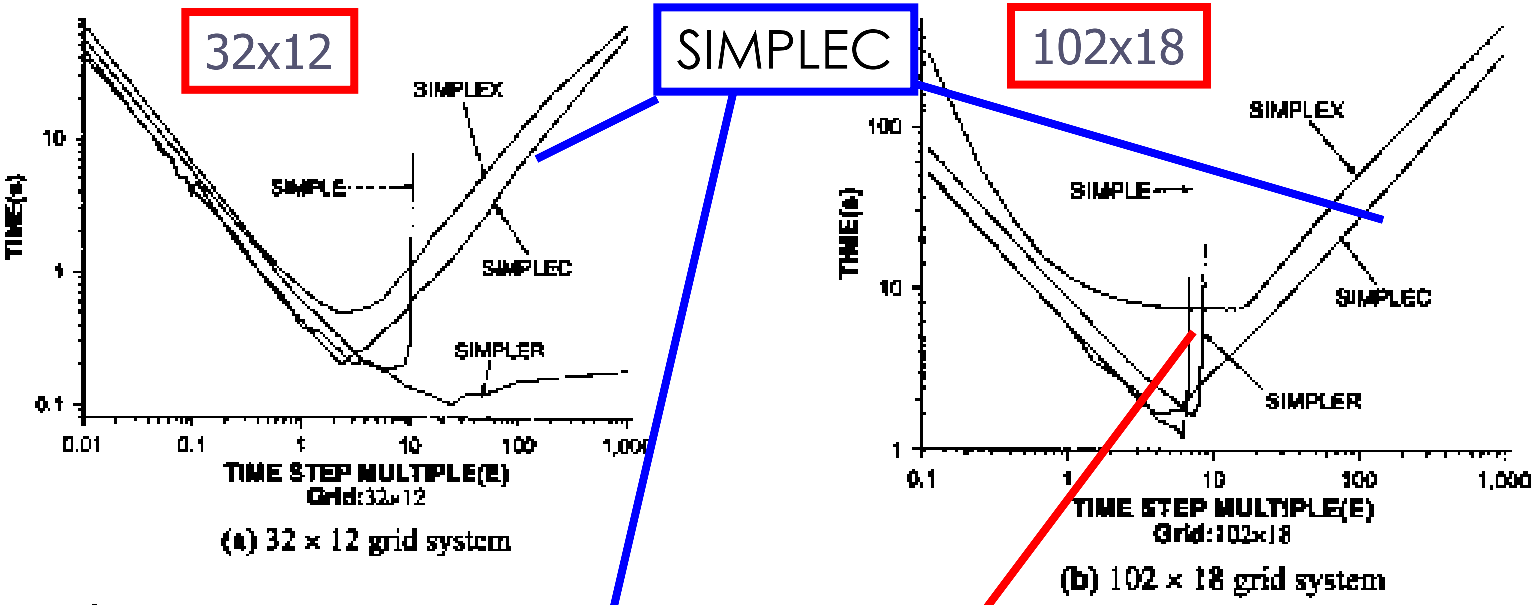


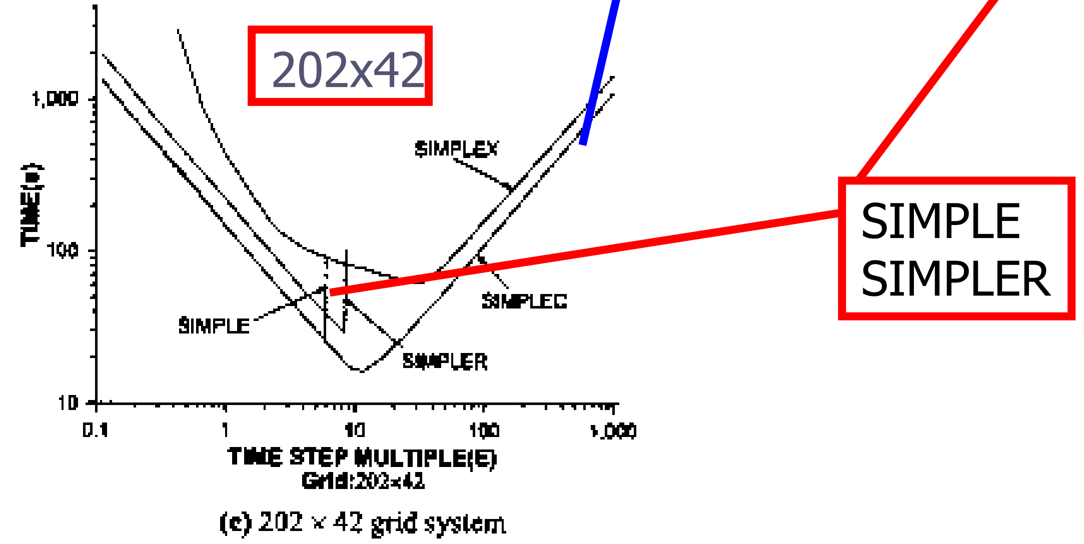
(3)natural convection in a square cavity

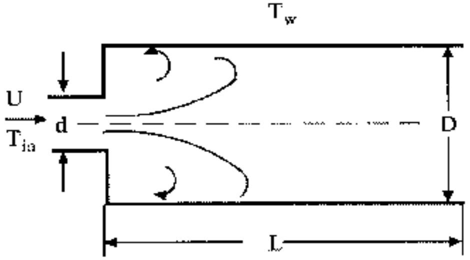


(4)natural convection in a horizontal annular

(2) Comparison results (for Problem 3)







(2)flow in a tube with sudden expansion

(3) About d $u' = d(\Delta p')$ For a certain correction:

$d \uparrow \quad \Delta p' \downarrow$

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 （2008） have completely overcome the two assumptions of SIMPLE algorithm.

1. Sun DL, Qu Z G, He Y L, Tao WQ. An efficient segregated algorithm for incompressible fluid flow and heat transfer problems-IDEAL (Inner doubly iterative efficient algorithm for linked equation) Part I:mathematical formulation and solution. Numerical Heat Transfer, Part B, 2008,53(1);1-17

2. Sun DL, Qu Z G, He Y L, Tao WQ. An efficient segregated algorithm for incompressible fluid flow and heat transfer problems-IDEAL (Inner doubly iterative efficient algorithm for linked equation) Part II:Application examples. Numerical Heat Transfer, Part B, 2008,53(1);18-38

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

Analysis of Homework-3 Errors

Problem 3-1

As shown in the figure , in 1-D steady heat conduction problem, known conditions are: T_1=110, λ =18, S=105, T_f=30, h=40, the units in every term are consistent. Try to determine the values of T_2, T_3; Prove that the solutions meet the overall conservation requirement even though only three nodes are used.

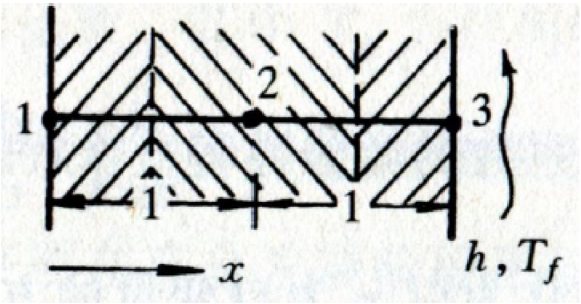
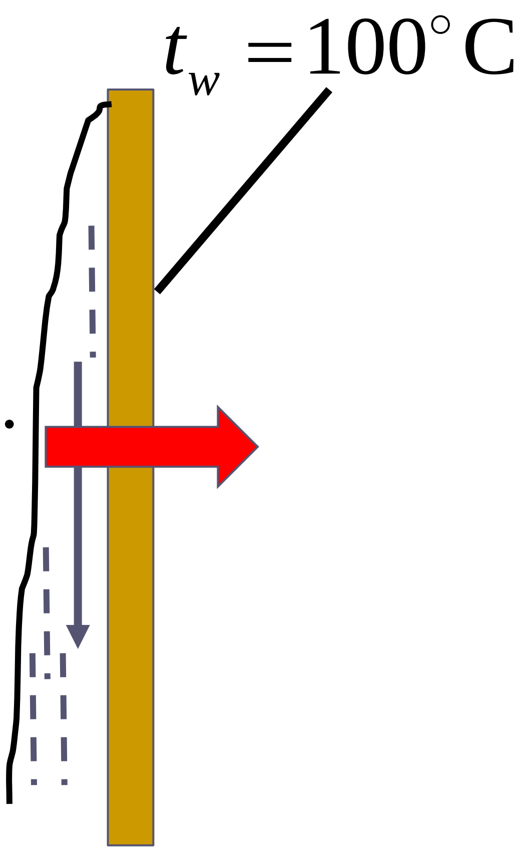


Figure of Prob. 3-1

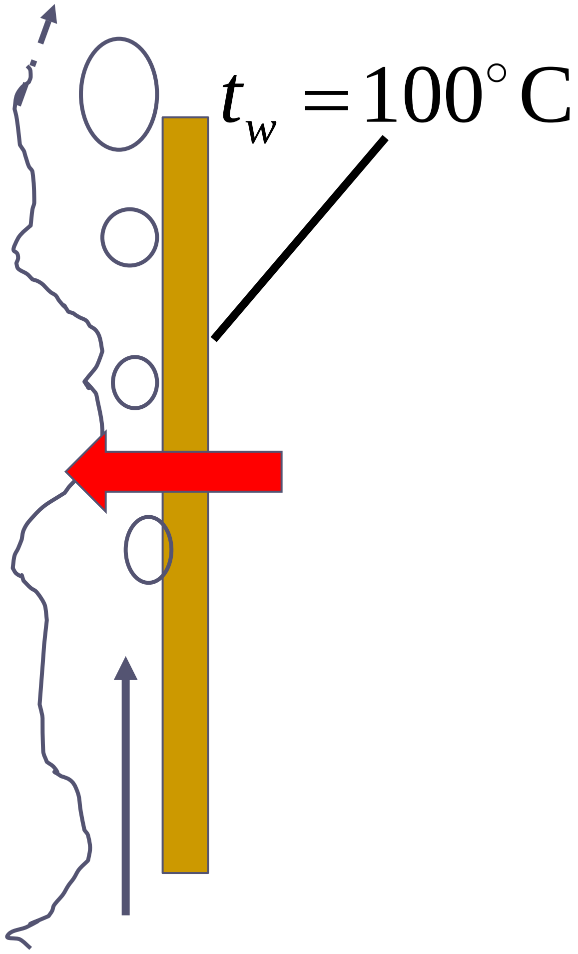
Analysis: For steady-state problem, energy conservation means: Energy obtained =Energy dissipated. Obviously the plate receives energy from its source term and dissipates energy to the surrounding fluid. **However, is there other energy gain or loss?**

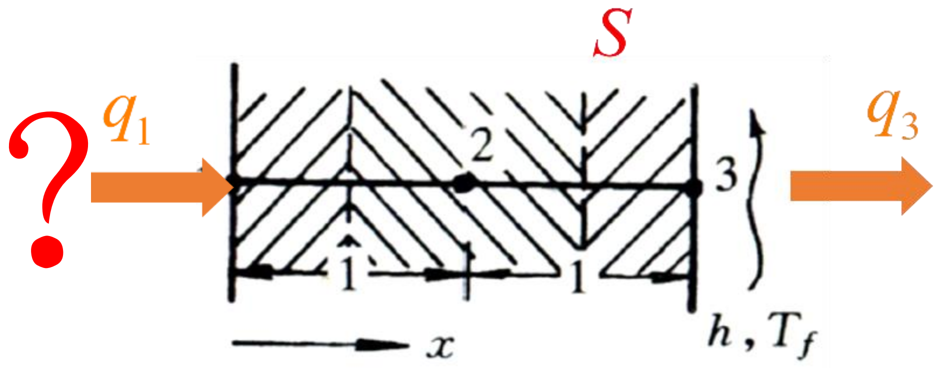
A wall with given temperature may be heated or cooled!

Water vapor condensation. at the wall. The wall is heated!



Water boiling at the wall, The wall is cooled





$$q_3 = h\left(T_3 - T_f\right) = 35\times(36.705 - 20) = 584,675$$

Energy balance for the half CV 1

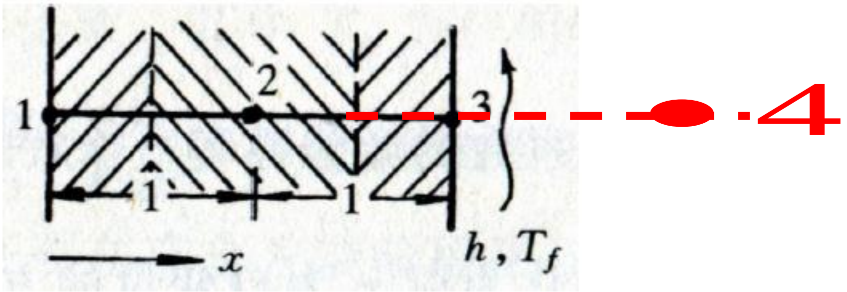
$$q_1 + \lambda \frac{T_2 - T_1}{\delta x} + S\Delta x / 2 = 0 \Rightarrow q_1 = 374.658$$

Overall Energy Balance for the Plate

$$q_1 + 2\delta x S - q_3 = 374.658 + 2\times 105 - 584,675 \cong 0$$

T_4

不能采用“虚拟点法”



采用二阶截差公式计算T_2 $\lambda\dfrac{T_3-2T_2+T_1}{\delta x^2}+S=0,\ \lambda=18, S=105,\ \delta x=1$ (1)

使用虚拟点法，假设边界外还有一点4 $\lambda\dfrac{T_4-T_2}{2\delta x}=h\left(T_f-T_3\right), h=35$ (2)

由控制方程可得在3点的离散形式为 $\lambda\dfrac{T_4-2T_3+T_2}{\left(\delta x\right)^2}+S=0$ (3)

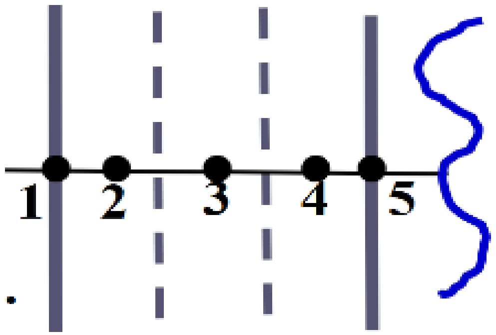
由（2），（3）得：$T_3=T_2+\dfrac{(\delta x)(\Delta x)S}{\lambda}+\dfrac{h\left(T_f-T_3\right)\delta x}{\lambda}, \Delta x=0.5\delta x$ (4)

由（1），（4）解得结果。

Problem 3-2

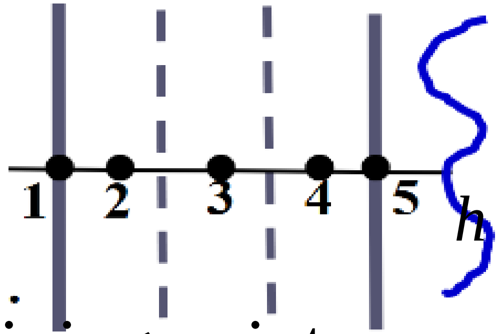
A large plate with thickness of 0.12 m, uniform source $S = 55\times10^3\ \mathrm{W/m^3}$, $\lambda = 25\ \mathrm{Wm^{-1}K^{-1}}$; One of its wall is kept at $T_1 = 80^\circ\mathrm{C}$, while the other wall is cooled by a fluid with $T_f = 25^\circ\mathrm{C}$ and heat transfer coefficient h=55 $\mathrm{Wm^{-2}K^{-1}}$.

Adopt Practice B, divide the plate thickness into three uniform CVs, determine the temperatures of nodes 2,3,4,5. Take 2nd order accuracy discretization for the inner node. Adopt the additional source term method for the right boundary node.



$T_f = 25^\circ\mathrm{C}$,

$h = 55\mathrm{W/(m^2 \bullet K)}$



$$T_f = 25^\circ\mathrm{C},$$

$$h = 55\mathrm{W}/(\mathrm{m}^2 \bullet \mathrm{K})$$

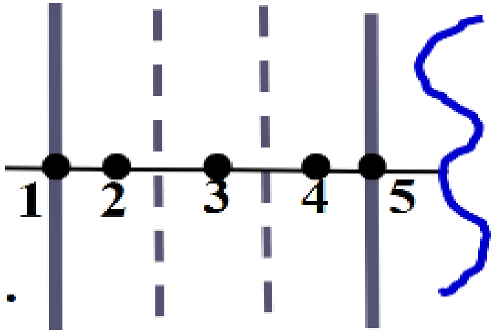
Two major training points:

(1) For nodes 2 and 4 their left and right are not symmetric. How to determine the west and east coefficients:

$$a_E = a_W = \frac{\lambda A}{\delta x}$$

For nodes 2 and 4 λA is the same, and the only difference is the distance between two neighboring points: one is 1/3, the other is 1/6

(2) For nodes 5 it is the boundary node in Practice B, and does not have its volume. Using the ASTM to treat the right convective boundary condition. Three equations of nodes 2,3,4 will compose closed ABEqs.



$T_f = 25^\circ \mathrm{C},$

$h = 55\mathrm{W}/(\mathrm{m}^2 \bullet \mathrm{K})$

1. Write the discretized equations of nodes 2,3,4: for nodes 2 and 4, left and right are not geometric symmetric

2. Adopt the ASTM to treat the right wall boundary condition:

add following two terms for node 4 will make equations for nodes 2,3,4 closed!

$$S_{c,ad} = \frac{\Delta y \bullet T_f}{\Delta V[\frac{1}{h} + \frac{(\delta x)_B}{\lambda_B}]}$$

$$S_{p.ad} = -\frac{\Delta y}{\Delta V \bullet [1/h + (\delta x)_B / \lambda_B]}$$

See page 51 of teaching PPT of Chapter 3(1)

本组网页地址：http://nht.xjtu.edu.cn 欢迎访问！

Teaching PPT will be loaded on ou website

同舟共济

渡彼岸！

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