

CFD コード断片集

春日 悠

2016年6月24日

目次

1	はじめに	2
2	1次元熱伝導方程式	2
2.1	1次元熱伝導方程式の離散化	2
2.2	ソース項の線形化	4
2.3	セル界面の熱伝導率の補間	4
2.4	例1	5
2.5	例2	6
2.6	例3	7
3	2次元スカラー輸送方程式	8
3.1	2次元スカラー輸送方程式の離散化	8
3.2	対流項の離散化	10
3.3	例	11
4	2次元の流れ計算	12
4.1	運動方程式	12
4.2	SIMPLE法	13
4.3	例1	14
4.4	例2	18
5	不足緩和	21
5.1	例1	21
5.2	例2	22
6	TDMA	22
6.1	例1	23
6.2	例2	23
6.3	例3	24

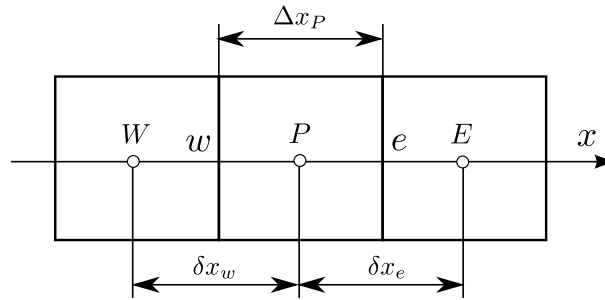


図 1: 1次元セル

6.4 例 4	25
6.5 例 5	25

1 はじめに

数値流体力学 (CFD) のコード断片を収集する。動作の確認は行わない。コードの引用については、わかりやすさのために多少編集している場合がある。

2 1次元熱伝導方程式

2.1 1次元熱伝導方程式の離散化

1次元熱伝導方程式を有限体積法で離散化する。

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right) \quad (1)$$

上式をコントロールボリューム (セル) P において x で積分する。

$$\rho c \frac{\partial T_P}{\partial t} \Delta x_P = \left(\lambda \frac{\partial T}{\partial x} \right)_e - \left(\lambda \frac{\partial T}{\partial x} \right)_w \quad (2)$$

ここで、添字の e, w はセル P のそれぞれ右、左にある隣接セル E, W とセル P との界面位置を表している (図 1)。右辺の各項を次のように離散化する。

$$\begin{aligned} \left(\lambda \frac{\partial T}{\partial x} \right)_e &= \lambda_e \frac{T_E - T_P}{\delta x_e} \\ \left(\lambda \frac{\partial T}{\partial x} \right)_w &= \lambda_w \frac{T_P - T_W}{\delta x_w} \end{aligned} \quad (3)$$

方程式の右辺を離散化すると、次のようになる。

$$- \left(\frac{\lambda_e}{\delta x_e} + \frac{\lambda_w}{\delta x_w} \right) T_P + \frac{\lambda_e}{\delta x_e} T_E + \frac{\lambda_w}{\delta x_w} T_W \quad (4)$$

時間微分項については，たとえば，方程式を次の形で表すとする．

$$\frac{\partial T}{\partial t} = F(T) \quad (5)$$

陽解法で離散化する場合，次式のようになる．

$$\frac{T^1 - T^0}{\Delta t} = F(T^0) \quad (6)$$

ここで，添字の 1, 0 はそれぞれ現在の時刻，前の時刻を表している．陰解法では次のようになる．

$$\frac{T^1 - T^0}{\Delta t} = F(T^1) \quad (7)$$

一般化するために係数 f を導入すると，次式のように表すことができる．

$$\frac{T^1 - T^0}{\Delta t} = fF(T^1) + (1-f)F(T^0) \quad (8)$$

$f = 0$ で陽解法， $f = 1$ で陰解法， $f = 0.5$ ではクランク・ニコルソン法になる．

以上より，もとの方程式を離散化すると，次のようになる．

$$\begin{aligned} \frac{\rho c \Delta x_P}{\Delta t} (T_P - T_P^0) = & f \left[- \left(\frac{\lambda_e}{\delta x_e} + \frac{\lambda_w}{\delta x_w} \right) T_P + \frac{\lambda_e}{\delta x_e} T_E + \frac{\lambda_w}{\delta x_w} T_W \right] \\ & + (1-f) \left[- \left(\frac{\lambda_e}{\delta x_e} + \frac{\lambda_w}{\delta x_w} \right) T_P^0 + \frac{\lambda_e}{\delta x_e} T_E^0 + \frac{\lambda_w}{\delta x_w} T_W^0 \right] \end{aligned} \quad (9)$$

整理すると，次のようになる．

$$\begin{aligned} \left(f \frac{\lambda_e}{\delta x_e} + f \frac{\lambda_w}{\delta x_w} + \frac{\rho c \Delta x_P}{\Delta t} \right) T_P = & f \frac{\lambda_e}{\delta x_e} T_E + f \frac{\lambda_w}{\delta x_w} T_W \\ & + (1-f) \frac{\lambda_e}{\delta x_e} T_E^0 + (1-f) \frac{\lambda_w}{\delta x_w} T_W^0 \\ & + \left[\frac{\rho c \Delta x_P}{\Delta t} - (1-f) \frac{\lambda_e}{\delta x_e} - (1-f) \frac{\lambda_w}{\delta x_w} \right] T_P^0 \end{aligned} \quad (10)$$

これは次式のように表される．

$$a_P T_P = a_E T_E + a_W T_W + b \quad (11)$$

ここで

$$\begin{aligned} a_E &= f \frac{\lambda_e}{\delta x_e} \\ a_W &= f \frac{\lambda_w}{\delta x_w} \\ a_P^0 &= \frac{\rho c \Delta x_P}{\Delta t} \\ a_P &= a_E + a_W + a_P^0 \\ b &= (1-f) \frac{\lambda_e}{\delta x_e} T_E^0 + (1-f) \frac{\lambda_w}{\delta x_w} T_W^0 \\ &+ \left[a_P^0 - (1-f) \frac{\lambda_e}{\delta x_e} - (1-f) \frac{\lambda_w}{\delta x_w} \right] T_P^0 \end{aligned} \quad (12)$$

定常計算の場合，時間微分項がないので，次式ようになる．

$$\left(\frac{\lambda_e}{\delta x_e} + \frac{\lambda_w}{\delta x_w} \right) T_P = \frac{\lambda_e}{\delta x_e} T_E + \frac{\lambda_w}{\delta x_w} T_W \quad (13)$$

これは次のように表される．

$$a_P T_P = a_E T_E + a_W T_W \quad (14)$$

ここで

$$\begin{aligned} a_E &= \frac{\lambda_e}{\delta x_e} \\ a_W &= \frac{\lambda_w}{\delta x_w} \\ a_P &= a_E + a_W \end{aligned} \quad (15)$$

2.2 ソース項の線形化

ソース項 (この場合は発熱) がある場合は右辺に足せばよいが，ソース項が変数 (ここでは温度) に依存している場合は，安定化のために線形化が必要である．たとえば，ソース項 S が次式のように表されるとする．

$$S = S_p T + S_u \quad (16)$$

S_p は温度に依存していてもよいが，負の値である必要がある．たとえば定常計算の場合，ソース項を含んだ離散化式は次式で表される．

$$(a_E + a_W - S_p) T_P = a_E T_E + a_W T_W + S_u \quad (17)$$

代数方程式が対角優位であるためには， S_p の値は負でなければならない．

2.3 セル界面の熱伝導率の補間

セル界面の熱伝導率は，両側セルの熱伝導率の調和平均で求める．

$$\frac{\delta x_e}{\lambda_e} = \frac{\delta x_{eP}}{\lambda_P} + \frac{\delta x_{eE}}{\lambda_E} \quad (18)$$

ここで， δx_{eP} はセル界面 e とセル P 中心との距離， δx_{eE} はセル界面 e とセル E 中心との距離で， $\delta x_e = \delta x_{eP} + \delta x_{eE}$ である．変形すると次式になる．

$$\frac{\lambda_e}{\delta x_e} = \frac{\lambda_P \lambda_E}{\lambda_P \delta x_{eE} + \lambda_E \delta x_{eP}} \quad (19)$$

調和平均を用いないと，たとえば単純に足して 2 で割るような計算 (相加平均) の場合， $\lambda_E = 0$ でも熱流束が 0 にならない．

2.4 例1

EXA1 [2] よりコードを示す．代数方程式は次式で表されている．

$$a_P T_P = a_E T_E + a_W T_W + b \quad (20)$$

時間微分項の離散化には陰解法 ($f = 1$) が使われている．

方程式の係数の設定部分は以下の通りである．

```
NIM1 = NI - 1
DO 30 I = 2, NIM1
  AW(I) = ALAMD/DG(I - 1)
  AE(I) = ALAMD/DG(I)
  AO(I) = (DX(I)*RHO*CP)/DELT
  B(I) = AO(I)*TO(I)
  AP(I) = AE(I) + AW(I) + AO(I)
30 CONTINUE
```

境界条件は以下の通り．

```
AW(1) = 0.
AE(1) = 0.
AO(1) = 1.
TO(1) = TH
B(1) = AO(1)*TO(1)
AP(1) = 1.

AW(NI) = 0.
AE(NI) = 0.
AO(NI) = 1.
B(NI) = AO(NI)*TL
AP(NI) = 1.
```

計算の進行部分は次の通りである．

```
TIME = 0.
999 CONTINUE
TIME = TIME + DELT

...

IF(TIME .LT. TIMEND) GO TO 999
```

2.5 例2

EXA2 [2] よりコードを示す．本コードの問題は定常計算であり，また線形化されたソース項が含まれている．代数方程式は次式で表されている．

$$a_P T_P = a_E T_E + a_W T_W + S_u \quad (21)$$

```
DO 10 I = 2, NI - 1
  VOL = R(I)*SEW(I)
  AREAE = RW(I+1)
  AREAW = RW(I)
  ALDE = 1./((SPW(I+1)/ALAMD(I+1) + SEP(I)/ALAMD(I))
  AE(I) = ALDE*AREAE
  ALDW = 1./((SPW(I)/ALAMD(I) + SEP(I-1)/ALAMD(I-1))
  AW(I) = ALDW*AREAW
  IF(I .LE. NI1) THEN
    SOURCE = QS*VOL
  ELSE
    SOURCE = 0.
  ENDIF
  SU(I) = SOURCE
  SP(I) = 0.
10 CONTINUE
```

```
TMONIN = QS*R1*R1*0.5
RESORT = 0.

DO 20 I = 2, NI - 1
  AP(I) = AE(I) + AW(I) - SP(I)
  RESOR = AE(I)*T(I+1) + AW(I)*T(I-1)
&      - AP(I)*T(I) + SU(I)
  RESORT = RESORT + ABS(RESOR)
20 CONTINUE

RESORT = RESORT/TMONIN
```

本コードの問題では物性および境界条件が温度に依存しているため，反復計算をしている．そのため，残差 (RESORT) を計算している．残差は発熱量を基準にしている．計算の停止条件は以下のようにになっている．一定の反復計算回数に達するか，残差が小さくなるか，発散を検出すると止まるようになっている．

```
200 ITERT = ITERT + 1
...
IF((ITERT .GT. NITERT) .OR. (RESORT .GT. 1E3))
```

```

& GO TO 270
  IF (RESORT .LT. SORMAX) GO TO 271
  GO TO 200

```

2.6 例3

heat.for [3] よりコードを示す．代数方程式は次式で表されている．

$$a_P T_P = a_E T_E + a_W T_W + S_u \quad (22)$$

ここでは，もとの方程式は無次元化されている．

境界条件は以下の通り．

```

TEMP(1) = 0.
TEMP(NI) = 0.
AP(1) = 1.
AE(1) = 0.
SU(1) = TEMP(1)
AP(NI) = 1.
AW(NI) = 0.
SU(NI) = TEMP(NI)

```

方程式の係数の設定部分は以下の通りである．

```

NIM1 = NI - 1
F1 = 1. - F

DO 100 I = 2, NIM1
  AEOR = 1./DXEP(I)
  AWOR = 1./DXPW(I)
  AP0 = SEW(I)/DTIME
  AP(I) = F*AEOR + F*AWOR + AP0
  AE(I) = AEOR*F
  AW(I) = AWOR*F
  SU(I) = AEOR*F1*TEMP(I+1) + AWOR*F1*TEMP(I-1)
&      + (AP0 - F1*AEOR - F1*AWOR)*TEMP(I)
  SP(I) = 0.
100 CONTINUE

```

計算の進行部分は次の通りである．

```

10 NITER = NITER + 1
  TIME = TIME + DTIME
  CALL CALCOE

```

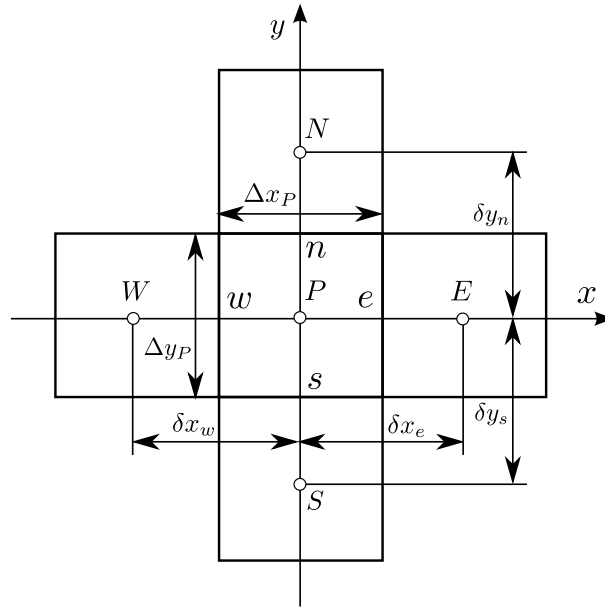


図 2: 2次元セル

```
CALL TDMA(NI, TEMP)
IF (TIME .LT. TMAX) GO TO 10
```

方程式の係数を計算し，TDMA で方程式を解いている．

3 2次元スカラー輸送方程式

3.1 2次元スカラー輸送方程式の離散化

次式の2次元スカラー輸送方程式を考える．

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x} \left(u\phi - \Gamma \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(v\phi - \Gamma \frac{\partial \phi}{\partial y} \right) = S \quad (23)$$

方程式を有限体積法で離散化する．コントロールボリューム (セル) の配置は図 2 のようになっているものとして，セル P において方程式の各項を時間および x, y で積分すると，以下ようになる．

$$\frac{1}{\Delta t} \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial \phi}{\partial t} dx dy dt = \frac{\Delta x_P \Delta y_P}{\Delta t} (\phi_P - \phi_P^0) \quad (24)$$

$$\begin{aligned} \frac{1}{\Delta t} \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial}{\partial x} \left(u\phi - \Gamma \frac{\partial \phi}{\partial x} \right) dx dy dt = \\ \Delta y_P \left[u_e \phi_e - \left(\Gamma \frac{\partial \phi}{\partial x} \right)_e \right] - \Delta y_P \left[u_w \phi_w - \left(\Gamma \frac{\partial \phi}{\partial x} \right)_w \right] \end{aligned} \quad (25)$$

$$\frac{1}{\Delta t} \int_t^{t+\Delta t} \int_s^n \int_w^e \frac{\partial}{\partial y} \left(v\phi - \Gamma \frac{\partial \phi}{\partial y} \right) dx dy dt = \Delta x_P \left[v_n \phi_n - \left(\Gamma \frac{\partial \phi}{\partial y} \right)_n \right] - \Delta x_P \left[v_s \phi_s - \left(\Gamma \frac{\partial \phi}{\partial y} \right)_s \right] \quad (26)$$

$$\frac{1}{\Delta t} \int_t^{t+\Delta t} \int_s^n \int_w^e S dx dy dt = (S_P \phi_P + S_C) \Delta x_P \Delta y_P \quad (27)$$

ここで、添字 e, w, n, s はセル P と隣接セルとの界面位置を意味している。
セル界面位置の ϕ は、次のように補間する。

$$\begin{aligned} \phi_e &= f_e \phi_E + (1 - f_e) \phi_P \\ \phi_w &= f_w \phi_W + (1 - f_w) \phi_P \\ \phi_n &= f_n \phi_N + (1 - f_n) \phi_P \\ \phi_s &= f_s \phi_S + (1 - f_s) \phi_P \end{aligned} \quad (28)$$

ここで、 $f_e = \delta x_{eP} / \delta x_e$ などである。また、セル界面位置の拡散項は、次のように離散化する。

$$\begin{aligned} \left(\Gamma \frac{\partial \phi}{\partial x} \right)_e &= \Gamma_e \frac{\phi_E - \phi_P}{\delta x_e} \\ \left(\Gamma \frac{\partial \phi}{\partial x} \right)_w &= \Gamma_w \frac{\phi_P - \phi_W}{\delta x_w} \\ \left(\Gamma \frac{\partial \phi}{\partial y} \right)_n &= \Gamma_n \frac{\phi_N - \phi_P}{\delta y_n} \\ \left(\Gamma \frac{\partial \phi}{\partial y} \right)_s &= \Gamma_s \frac{\phi_P - \phi_S}{\delta y_s} \end{aligned} \quad (29)$$

ここではスタッガード格子を想定し、界面位置の速度成分 u, v については直接計算されるものとする。

以上を整理すると、次式を得る。

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + b \quad (30)$$

ここで

$$\begin{aligned}
a_E &= -f_e F_e + \frac{\Gamma_e \Delta y_P}{\delta x_e} \\
a_W &= f_w F_w + \frac{\Gamma_w \Delta y_P}{\delta x_w} \\
a_N &= -f_n F_n + \frac{\Gamma_n \Delta x_P}{\delta y_n} \\
a_S &= f_s F_s + \frac{\Gamma_s \Delta x_P}{\delta y_s} \\
F_e &= u_e \Delta y_P \\
F_w &= u_w \Delta y_P \\
F_n &= v_n \Delta x_P \\
F_s &= v_s \Delta x_P \\
a_P &= a_E + a_W + a_N + a_S \\
&\quad + F_e - F_w + F_n - F_s \\
&\quad + \frac{\Delta x_P \Delta y_P}{\Delta t} - S_P \Delta x_P \Delta y_P \\
b &= \frac{\Delta x_P \Delta y_P}{\Delta t} \phi_P^0 + S_c \Delta x_P \Delta y_P
\end{aligned} \tag{31}$$

係数 a_P 中の $F_e - F_w + F_n - F_s$ は、非圧縮性流体の場合は質量保存から本来 0 になるべき項であるが、計算上はそうなるとは限らないので、値が正である場合は残しておいた方が計算の安定化のためにはよい。

3.2 対流項の離散化

上記で導いた離散化方程式は、界面の ϕ の値をセル値から線形に補間しているため、対流項に中心差分を用いていることになり、レイノルズ数が高くなると計算が破綻する。それを避けるため、界面の ϕ に上流のセル値をとる風上法という方法が用いられる。風上法を用いた場合、係数はつぎのようになる。

$$\begin{aligned}
a_E &= \max(-f_e F_e, 0) + \frac{\Gamma_e \Delta y_P}{\delta x_e} \\
a_W &= \max(f_w F_w, 0) + \frac{\Gamma_w \Delta y_P}{\delta x_w} \\
a_N &= \max(-f_n F_n, 0) + \frac{\Gamma_n \Delta x_P}{\delta y_n} \\
a_S &= \max(f_s F_s, 0) + \frac{\Gamma_s \Delta x_P}{\delta y_s}
\end{aligned} \tag{32}$$

ただし、風上法は数値拡散が大きい。計算が破綻するのは拡散に比べて対流が卓越する場合なので、対流が拡散と同レベルである場合には中心差分を用い、そうでなければ

ば風上法を用いるという方法も考えられる．この方法はハイブリッド法と呼ばれる．ハイブリッド法を用いるには，中心差分の係数をつぎのように修正すればよい．

$$\begin{aligned}
 a_E &= \max(a_{E_r}, -F_e, 0) \\
 a_W &= \max(a_{W_r}, F_w, 0) \\
 a_N &= \max(a_{N_r}, -F_n, 0) \\
 a_S &= \max(a_{S_r}, F_s, 0)
 \end{aligned}
 \tag{33}$$

ただ，ハイブリッド法は中心差分と風上法をとつぜん切り替える．Patanker [1] はこれに代わる方法として，べき乗法を推奨している．

3.3 例

SUNSET [2] よりコードを示す．

```

DO 1100 I = 2, IEND
  VS = V(I, 1)
  GAMS = 0.5*(GAM(I, 1) + GAM(I, 2))
  FS(I) = VS*DX(I)
  DS = GAMS*DX(I)/DELY(1)
  A = 0.5*FS(I) + DS
  ASS(I) = AMAX1(A, FS(I), 0.)
1100 CONTINUE

DO 1600 J = 2, JEND
  UW = U(1, J)
  GAMW = 0.5*(GAM(1, J) + GAM(2, J))
  FW = UW*DY(J)
  DW = GAMW*DY(J)/DELX(1)
  A = 0.5*FW + DW
  AWW = AMAX1(A, FW, 0.)

DO 1600 I = 2, IEND
  AW(I, J) = AWW
  AS(I, J) = ASS(I)

  UE = U(I, J)
  GAME = (1. - FACX(I))*GAM(I, J)
&      + FACX(I)*GAM(I+1, J)

  VN = V(I, J)
  GAMN = (1. - FACY(J))*GAM(I, J)
&      + FACY(J)*GAM(I, J+1)

  FE = UE*DY(J)
  FN = VN*DX(I)

  DE = GAME*DY(J)/DELX(I)

```

```

A = -FACX(I)*FE + DE
AE(I, J) = AMAX1(A, -FE, 0.)

A = (1. - FACX(I))*FE + DE
AWW = AMAX1(A, FE, 0.)

DN = GAMN*DX(I)/DELY(J)
A = -FACY(J)*FN + DN
AN(I, J) = AMAX1(A, -FN, 0.)

A = (1. - FACY(J))*FN + DN
ASS(I) = AMAX1(A, FN, 0.)

RES = FE - FW + FN - FS(I)
SP(I, J) = -AMAX1(0., RES)

FW = FE
1600 FS(I) = FN

```

一見、セルごとに係数を計算しているように見えるが、 w, s 界面はループの1つ前に計算されたものを用いていて、 e, n 界面の係数を計算するときと一緒に隣接セルの w, s 界面の係数を計算している。計算領域の外にも1セルあることが想定されていることに注意する。

残りの処理は次の通りである。

```

J0 = (IPHI - 1)*JDIM
DO 6100 J = 2, JEND
  JJ = J0 + J
  DO 6100 I = 2, IEND
    AP(I, J) = AE(I, J) + AW(I, J) + AN(I, J)
    &      + AS(I, J) - SP(I, J)
    DVDT = DX(I)*DY(J)/DTIME
    AP(I, J) = AP(I, J) + DVDT
    SC(I, J) = SC(I, J) + DVDT*PHIOLD(I, JJ)
    ...

```

このあと、方程式はI方向とJ方向でそれぞれ TDMA により解かれる。

4 2次元の流れ計算

4.1 運動方程式

運動方程式は、スカラー輸送方程式と同様に離散化できる。ただし、ここではスタッガード格子を用いるため、速度はスカラーセルの界面で定義される。運動方程式の離

散化式は次式で表される．

$$a_P u_P = a_E u_E + a_W u_W + a_N u_N + a_S u_S + \Delta y (p_w - p_e) + b \quad (34)$$

$$a_P v_P = a_E v_E + a_W v_W + a_N v_N + a_S v_S + \Delta x (p_s - p_n) + b \quad (35)$$

ここで，添字 P や E などは，注目している速度のセルの相対位置を表しており，スカラーセルから見たら u_P , v_P は u_e や v_n などに当たる．

4.2 SIMPLE 法

圧力の情報は陽には表されず，連続の式から導く必要がある．その方法のひとつに，SIMPLE 法がある．SIMPLE 法では速度と圧力の補正値を求め，それぞれを修正していく．

速度と圧力が，それぞれ次式のように予測値と補正値に分けられるとする．

$$u = u^* + u' \quad (36)$$

$$v = v^* + v' \quad (37)$$

$$p = p^* + p' \quad (38)$$

ここで u^* は u の予測値， u' は u の補正値である．ある時点での速度の予測値は，連続の式を満たすとは限らないが，運動方程式は満足する．

$$a_P u_P^* = a_E u_E^* + a_W u_W^* + a_N u_N^* + a_S u_S^* + \Delta y (p_w - p_e) + b \quad (39)$$

これを最終的な速度の運動方程式から引くと，速度の補正値についての方程式が得られる．

$$a_P u_P' = a_E u_E' + a_W u_W' + a_N u_N' + a_S u_S' + \Delta y (p_w' - p_e') \quad (40)$$

補正値は繰り返し計算の中で最終的に 0 になるので，上式の右辺の速度補正の項を無視してしまう．

$$a_P u_P' = \Delta y (p_w' - p_e') \quad (41)$$

これより

$$u_P' = d_x (p_w' - p_e') \quad (42)$$

ここで

$$d_x = \frac{\Delta y}{a_P} \quad (43)$$

v の運動方程式も同様にして

$$v_P' = d_y (p_s' - p_n') \quad (44)$$

ここで

$$d_y = \frac{\Delta x}{a_P} \quad (45)$$

圧力に関する方程式を導くために、連続の式を離散化する。ここでは非圧縮性流体を考える。

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (46)$$

上式を有限体積法で離散化すると、次式を得る。

$$\Delta y(u_e - u_w) + \Delta x(v_n - v_s) = 0 \quad (47)$$

つまり

$$F_e - F_w + F_n - F_s = 0 \quad (48)$$

式 (42)、式 (44) を用いる。位置の記号は速度定義位置の相対位置であることに注意して、セル界面の速度は次のように表される。

$$\begin{aligned} u_e &= d_{ye}(p'_P - p'_E) \\ u_w &= d_{yw}(p'_W - p'_P) \\ u_n &= d_{xn}(p'_P - p'_N) \\ u_s &= d_{xs}(p'_S - p'_P) \end{aligned} \quad (49)$$

これらを用いると、連続の式の離散化式は次式のように表される。

$$a_P p'_P = a_E p'_E + a_W p'_W + a_N p'_N + a_S p'_S + b \quad (50)$$

ここで

$$\begin{aligned} a_E &= \Delta y d_{ye} \\ a_W &= \Delta y d_{yw} \\ a_N &= \Delta x d_{xn} \\ a_S &= \Delta x d_{xs} \\ a_P &= a_E + a_W + a_N + a_S \\ b &= F_e^* - F_w^* + F_n^* - F_s^* \end{aligned} \quad (51)$$

圧力補正值 p' の境界条件は、圧力が規定されている境界は $p' = 0$ とする。速度が規定されている境界は $u' = 0$ 、 $v' = 0$ なので、式 (42)、式 (44) より p' の法線方向の勾配を 0 とすればよい。

4.3 例 1

SUNSET [2] よりコードを示す。 u の方程式の係数の計算は次の通りである。

```

DO 1100 I = 2, IEND
  VS = 0.5*(V(I, 1) + V(I+1, 1))
  GAMS = 0.25*(GAM(I, 1) + GAM(I+1, 1)
&      + GAM(I, 2) + GAM(I+1, 2))
  FS(I) = VS*DX(I)
  DS = GAMS*DX(I)/DELY(1)
  A = 0.5*FS(I) + DS
  ASS(I) = AMAX1(A, FS(I), 0.)
1100 CONTINUE

DO 1600 J = 2, JEND
  UW = 0.5*(U(1, J) + U(2, J))
  GAMW = GAM(2, J)
  FW = UW*DY(J)
  DW = GAMW*DY(J)/DELX(1)
  A = 0.5*FW + DW
  AWW = AMAX1(A, FW, 0.)

DO 1600 I = 2, IEND
  AW(I, J) = AWW
  AS(I, J) = ASS(I)

  UE = 0.5*((U(I, J) + U(I+1, J))
  GAME = GAM(I+1, J)
  VN = 0.5*(V(I, J) + V(I+1, J))
  GAMN = 0.5*(1. - FACY(J))*(GAM(I, J) + GAM(I+1, J))
&      + 0.5*FACY(J)*(GAM(I, J+1) + GAM(I+1, J+1))

  FE = UE*DY(J)
  FN = VN*DX(I)

  DE = GAME*DY(J)/DELX(I)
  A = -FACX(I)*FE + DE
  AE(I, J) = AMAX1(A, -FE, 0.)

  A = (1. - FACX(I))*FE + DE
  AWW = AMAX1(A, FE, 0.)

  DN = GAMN*DX(I)/DELY(J)
  A = -FACY(J)*FN + DN
  AN(I, J) = AMAX1(A, -FN, 0.)

  A = (1. - FACY(J))*FN + DN
  ASS(I) = AMAX1(A, FN, 0.)

  SC(I, J) = DY(J)*(P(I, J) - P(I+1, J))
  RES = FE - FW + FN - FS(I)
  SP(I, J) = -AMAX1(0., RES)

  FW = FE
1600  FS(I) = FN

```

ここで，方程式は無次元化されており，拡散係数 (= 粘性係数) はレイノルズ数 Re の逆数であらされる．あとの処理はスカラー輸送方程式と同様である．

v の方程式については次のとおりである．

```

DO 1100 I = 2, IEND
  VS = 0.5*(V(I, 1) + V(I, 2))
  GAMS = GAM(I, 2)
  FS(I) = VS*DX(I)
  DS = GAMS*DX(I)/DELY(1)
  A = 0.5*FS(I) + DS
  ASS(I) = AMAX1(A, FS(I), 0.)
1100 CONTINUE

DO 1600 J = 2, JEND
  UW = 0.5*(U(1, J) + U(1, J+1))
  GAMW = 0.25*((GAM(1, J) + GAM(1, J+1)
&      + GAM(2, J) + GAM(2, J+1))
  FW = UW*DY(J)
  DW = GAMW*DY(J)/DELX(1)
  A = 0.5*FW + DW
  AWW = AMAX1(A, FW, 0.)

DO 1600 I = 2, IEND
  AW(I, J) = AWW
  AS(I, J) = ASS(I)

  UE = 0.5*((U(I, J) + U(I, J+1))
  GAME = 0.5*(1. - FACX(I))*(GAM(I, J) + GAM(I, J+1))
&      + 0.5*FACX(J)*(GAM(I+1, J) + GAM(I+1, J+1))
  VN = 0.5*(V(I, J) + V(I, J+1))
  GAMN = GAM(I, J+1)

  FE = UE*DY(J)
  FN = VN*DX(I)

  DE = GAME*DY(J)/DELX(I)
  A = -FACX(I)*FE + DE
  AE(I, J) = AMAX1(A, -FE, 0.)

  A = (1. - FACX(I))*FE + DE
  AWW = AMAX1(A, FE, 0.)

  DN = GAMN*DX(I)/DELY(J)
  A = -FACY(J)*FN + DN
  AN(I, J) = AMAX1(A, -FN, 0.)

  A = (1. - FACY(J))*FN + DN
  ASS(I) = AMAX1(A, FN, 0.)

  SC(I, J) = DX(J)*(P(I, J) - P(I, J+1))
  RES = FE - FW + FN - FS(I)
  SP(I, J) = -AMAX1(0., RES)

```



```

      FW = FE
1600   FS(I) = FN

```

p' の方程式については次のとおりである .

```

      DO 6100 J = 2, JEND
        DO 6100 I = I = 2, IEND
          SDX(I, J) = DY(J)/AP(I, J)
6100   SDY(I, J) = DX(I)/AP(I, J)

```

```

      DO 1100 I = 2, IEND
        VS = V(I, 1)
        GAMS = 0.5*(GAM(I, 1) + GAM(I, 2))
        FS(I) = VS*DX(I)
        ASS(I) = DX(I)*SDY(I, 1)
1100 CONTINUE

      DO 1600 J = 2, JEND
        UW = U(1, J)
        GAMW = 0.5*(GAM(1, J) + GAM(2, J))
        FW = UW*DY(J)
        AWW = DY(J)*SDX(1, J)

        DO 1600 I = 2, IEND
          AW(I, J) = AWW
          AS(I, J) = ASS(I)

          FE = UE*DY(J)
          FN = VN*DX(I)

          AE(I, J) = DY(J)*SDX(I, J)
          AN(I, J) = DX(I)*SDY(I, J)

          AWW = AE(I, J)
          ASS(I) = AN(I, J)

          SC(I, J) = FW - FE + FS(I) - FN
          SP(I, J) = ZERO

          FW = FE
1600   FS(I) = FN

```

圧力補正值 p' が求まったら , 速度 u, v と圧力 p を補正する .

```

DO 1100 I = 2, NI
  DO 1100 J = 2, NJ
    U(I, J) = U(I, J)
    &      + SDX(I, J)*(PC(I, J) - PC(I+1, J))
    V(I, J) = V(I, J)
    &      + SDY(I, J)*(PC(I, J) - PC(I, J+1))
1100    P(I, J) = P(I, J) + ALPHA(IP)*PC(I, J)

    PREF = P(IREF, JREF)
    DO 1200 I = 2, NI
      DO 1200 J = 2, NJ
        P(I, J) = P(I, J) - PREF
1200    PC(I, J) = ZERO

```

ここで、圧力の補正には不足緩和が用いられている。また、圧力は参照位置が0になるように調整されている。

計算の進行部分は次の通りである。

```

DO 1400 ITERT = 1, NITERT
  DO 1500 IPHI = 1, 3
1500    CALL SOLPHI
    CALL CORREC
    ...

```

ここでは u, v, p' の方程式がスカラー輸送方程式の形で解かれており、そのあとで u, v, p' がそれぞれ補正されている。

4.4 例2

laminar.for [3] よりコードを示す。本コードは定常計算である。 u の方程式の係数の計算は次の通りである。

```

DO 100 I = 3, NIM1
  DO 100 J = 2, NJM1
    CN = 0.5*DENSIT*(V(I, J+1) + V(I-1, J+1))*SEWU(I)
    CS = 0.5*DENSIT*(V(I, J) + V(I-1, J))*SEWU(I)
    CE = 0.5*DENSIT*(U(I+1, J) + U(I, J))*SNS(J)
    CW = 0.5*DENSIT*(U(I, J) + U(I-1, J))*SNS(J)
    DN = VISCOS*SEWU(I)/DYNP(J)
    DS = VISCOS*SEWU(I)/DYPS(J)
    DE = VISCOS*SNS(J)/DEXPU(I)
    DW = VISCOS*SNS(J)/DXPWU(I)
    AN(I, J) = AMAX1(ABS(0.5*CN), DN) - 0.5*CN
    AS(I, J) = AMAX1(ABS(0.5*CS), DS) + 0.5*CS

```

```

      AE(I, J) = AMAX1(ABS(0.5*CE), DE) - 0.5*CE
      AW(I, J) = AMAX1(ABS(0.5*CW), DW) + 0.5*CW
      DU(I, J) = SNS(J)
      SU(I, J) = DU(I, J)*(P(I-1, J) - P(I, J))
      SP(I, J) = 0.
100 CONTINUE

      RESORU = 0.
      DO 300 I = 3, NIM1,
        DO 300 J = 2, NJM1
          AP(I, J) = AN(I, J) + AS(I, J) + AE(I, J)
&          + AW(I, J) - SP(I, J)
          DU(I, J) = DU(I, J)/AP(I, J)
          RESOR = AN(I, J)*U(I, J+1) + AS(I, J)*U(I, J-1)
&          + AE(I, J)*U(I+1, J) + AW(I, J)*U(I-1, J)
&          - AP(I, J)*U(I, J) + SU(I, J)
          RESORU = RESORU + ABS(RESOR)
300 CONTINUE

```

式の形が異なるが、ハイブリッド法が用いられている。
 v の方程式の係数の計算は次の通りである。

```

      DO 100 I = 2, NIM1
        DO 100 J = 3, NJM1
          CN = 0.5*DENSIT*(V(I, J+1) + V(I, J))*SEW(I)
          CS = 0.5*DENSIT*(V(I, J) + V(I, J-1))*SEW(I)
          CE = 0.5*DENSIT*(U(I+1, J) + U(I+1, J-1))*SNSV(J)
          CW = 0.5*DENSIT*(U(I, J) + U(I, J-1))*SNSV(J)
          DN = VISCOS*SEW(I)/DYNPV(J)
          DS = VISCOS*SEW(I)/DYPSV(J)
          DE = VISCOS*SNSV(J)/DEXP(I)
          DW = VISCOS*SNSV(J)/DXPW(I)
          AN(I, J) = AMAX1(ABS(0.5*CN), DN) - 0.5*CN
          AS(I, J) = AMAX1(ABS(0.5*CS), DS) + 0.5*CS
          AE(I, J) = AMAX1(ABS(0.5*CE), DE) - 0.5*CE
          AW(I, J) = AMAX1(ABS(0.5*CW), DW) + 0.5*CW
          DV(I, J) = SNS(J)
          SU(I, J) = DV(I, J)*(P(I, J-1) - P(I, J))
          SP(I, J) = 0.
100 CONTINUE

      RESORU = 0.
      DO 300 I = 3, NIM1,
        DO 300 J = 2, NJM1
          AP(I, J) = AN(I, J) + AS(I, J) + AE(I, J)
&          + AW(I, J) - SP(I, J)
          DV(I, J) = DV(I, J)/AP(I, J)
          RESOR = AN(I, J)*V(I, J+1) + AS(I, J)*V(I, J-1)
&          + AE(I, J)*V(I+1, J) + AW(I, J)*V(I-1, J)
&          - AP(I, J)*V(I, J) + SU(I, J)
          RESORU = RESORU + ABS(RESOR)
300 CONTINUE

```

```
300 CONTINUE
```

p' の方程式の係数の計算は次の通りである .

```
RESORM = 0.
DO 100 I = 2, NIM1
  DO 100 J = 2, NJM1
    AN(I, J) = DENSIT*SEW(I)*DV(I, J+1)
    AS(I, J) = DENSIT*SEW(I)*DV(I, J)
    AE(I, J) = DENSIT*SNS(I)*DV(I+1, J)
    AW(I, J) = DENSIT*SNS(I)*DV(I, J)
    CN = DENSIT*V(I, J+1)*SEW(I)
    CS = DENSIT*V(I, J)*SEW(I)
    CE = DENSIT*U(I+1, J)*SNS(J)
    CW = DENSIT*U(I, J)*SNS(J)
    SMP = CN - CS + CE - CW
    SP(I, J) = 0.
    SU(I, J) = -SMP
    RESORM = RESORM + ABS(SMP)
100 CONTINUE

  DO 300 I = 2, NIM1,
    DO 300 J = 2, NJM1
      AP(I, J) = AN(I, J) + AS(I, J) + AE(I, J)
&      + AW(I, J) - SP(I, J)
300 CONTINUE
```

圧力補正值 p' が求まったら , 速度 u, v と圧力 p を補正する .

```
DO 500 I = 2, NIM1
  DO 500 J = 2, NJM1
    IF(I .NE. 2) U(I, J) = U(I, J)
&    + DU(I, J)*(PP(I-1, J) - PP(I, J))
    IF(J .NE. 2) V(I, J) = V(I, J)
&    + DV(I, J)*(PP(I, J-1) - PP(I, J))
500 CONTINUE

  PPREF = PP(IPREF, JPREF)
DO 510 I = 2, NIM1
  DO 510 J = 2, NJM1
    P(I, J) = P(I, J) + URFPP*(PP(I, J) - PPREF)
    PP(I, J) = 0.
510 CONTINUE
```

計算の進行部分は次の通りである .

```

FLOWIN = 0.
XMONIN = 0.
DO 250 J = 2, NJM1
  ARDEN = DENSIT*SNS(J)
  FLOWIN = FLOWIN + ARDEN*UWALL
  XMONIN = XMONIN + ARDEN*UWALL*UWALL
250 CONTINUE

300 CONTINUE
  NITER = NITER + 1
  CALL CALCU
  CALL CALCV
  CALL CALCP
  RESORM = RESORM/FLOWIN
  RESORU = RESORU/XMONIN
  RESORV = RESORV/XMONIN
  SOURCE = AMAX1(RESORM, RESORU, RESORV)
  IF((NITER .EQ. 20 .AND. SOURCE .GT. 1E7*SORMAX)
&    .OR. (NITER .EQ. MAXIT)) THEN
    STOP
  ENDIF
  IF(SOURCE .GT. SORMAX) GO TO 300

```

残差から発散を検出して止めている．本コードはキャビティ流れを想定しており，残差の基準は壁速度 UWALL から計算されている．

5 不足緩和

代数方程式が次式で表されるものとする．

$$a_P\phi_P = a_N\phi_N + a_S\phi_S + d \quad (52)$$

繰り返し計算の不安定性を抑制するために，次のような不足緩和を導入する．

$$\phi_P = \alpha\phi_P^i + (1 - \alpha)\phi_P^{i-1} \quad (53)$$

ここで， α は緩和係数，添字 i は i 回目の繰り返し計算の結果を意味する．方程式の両辺を a_P で割り，上式に代入すると，次式が得られる．

$$\frac{a_P}{\alpha}\phi_P = a_N\phi_N + a_S\phi_S + d + \frac{1 - \alpha}{\alpha}a_P\phi_P^{i-1} \quad (54)$$

5.1 例 1

SUNSET [2] よりコードを示す．

```

7000 AP(I, J) = AP(I, J)/ALPHA(IPHI)
      SC(I, J) = SC(I, J)
      &          + (1. - ALPHA(IPHI))*AP(I, J)*PHI(I, JJ)

```

5.2 例 2

laminar.for [3] よりコードを示す .

```

AP(I, J) = AP(I, J)/URFU
SU(I, J) = SU(I, J) + (1. - URFU)*AP(I, J)*U(I, J)
DU(I, J) = DU(I, J)*URFU

```

DU(I, J) については, SNS(J)/AP(I, J) で計算されるため, AP(I, J) の修正に伴い緩和係数 URFU が掛けられている .

6 TDMA

TDMA (TriDiagonal-Matrix Algorithm) あるいはトーマスアルゴリズムは, 三重対角行列を係数行列にもつ代数方程式を解くための方法である .

n 元の代数方程式が次式のように表されるとする .

$$a_i x_i = b_i x_{i+1} + c_i x_{i-1} + d_i \quad (55)$$

このとき, 下記の手順によって解 x が求まる .

1. $P_1 \cdots P_{n-1}, Q_1 \cdots Q_n$ を次式から求める .

$$\begin{aligned}
 P_1 &= \frac{b_1}{a_1} \\
 Q_1 &= \frac{d_1}{a_1} \\
 P_i &= \frac{b_i}{a_i - c_i P_{i-1}} \\
 Q_i &= \frac{d_i + c_i Q_{i-1}}{a_i - c_i P_{i-1}}
 \end{aligned} \quad (56)$$

2. 次式より解を求める .

$$\begin{aligned}
 x_n &= Q_n \\
 x_i &= P_i x_{i+1} + Q_i
 \end{aligned} \quad (57)$$

TDMA は、係数行列が三重対角行列になる 1 次元の問題を解くためのものであるが、これを 2 次元の構造格子の問題に用いることもできる。たとえば南北方向に 1 次元的に走査することにして、東西方向の成分はソース項と同じ扱いにすると、次式が得られる。

$$a_P \phi_P = a_N \phi_N + a_S \phi_S + d \quad (58)$$

これに対して TDMA を適用する。南北方向に TDMA を適用し、次いで東西方向にも TDMA を適用するという繰り返し計算を行うことで、解を求めることができる。この方法を線順法という。

6.1 例 1

EXA1 [2] よりコードを示す。代数方程式は次式で表されている。

$$a_P T_P = a_E T_E + a_W T_W + b \quad (59)$$

セルの配置は 図 1 の通りである。

```

NIM1 = NI - 1

P(1) = AE(1)/AP(1)
Q(1) = B(1)/AP(1)

DO 40 I = 2, NI
  P(I) = AE(I)/(AP(I) - AW(I)*P(I-1))
  Q(I) = (B(I) + AW(I)*Q(I - 1))
  &      /(AP(I) - AW(I)*P(I-1))
40 CONTINUE

T(NI) = Q(NI)
DO 50 I = 1, NIM1
  J = NI - I
  T(J) = P(J)*T(J+1) + Q(J)
50 CONTINUE

```

6.2 例 2

EXA2 [2] よりコードを示す。方程式は次式で表されている。

$$a \phi_P = b \phi_E + c \phi_W + d \quad (60)$$

```

SUBROUTINE TDMA(ISTART, IEND, PHI)
COMMON
& /TDMAB/A(20), B(20), C(20), D(20), E(20)

```

```

& /DSIZE/IT

DIMENSION PHI(IT), P(20), Q(20)

DO 10 I = ISTART, IEND
  DUMMY = 1./(A(I) - C(I)*P(I-1))
  P(I) = B(I)*DUMMY
  Q(I) = (D(I) + C(I)*Q(I-1))*DUMMY
10 CONTINUE

DO 20 I = IEND, ISTART, -1
  PHI(I) = P(I)*PHI(I+1) + Q(I)
20 CONTINUE

RETURN
END

```

6.3 例3

SUNSET [2] よりコードを示す．方程式は次式で表されている．

$$a_i \phi_i = b_i \phi_{i+1} + c_i \phi_{i-1} + d_i \quad (61)$$

```

SUBROUTINE THOMAS(N, AI, BI, CI, DI)
DIMENSION AI(N), BI(N), CI(N), DI(N)

NM1 = N - 1

BI(1) = BI(1)/AI(1)
DI(1) = DI(1)/AI(1)

DO 10 I = 2, N
  IM1 = I - 1
  DN = AI(I) - CI(I)*BI(IM1)
  BI(I) = BI(I)/DN
10  DI(I) = (DI(I) + CI(I)*DI(IM1))/DN

DO 20 I = 1, NM1
  NMI = N - I
20  DI(NMI) = BI(NMI)*DI(NMI + 1) + DI(NMI)

RETURN
END

```

ここでは，配列 P_i, Q_i を用意する代わりに配列 b_i, d_i を再利用している．また，解 x_i についても配列 d_i に入れている．

6.4 例 4

heat.for [3] よりコードを示す . 代数方程式は次式で表されている .

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + S_u \quad (62)$$

ただし , 端の点は解かれていない .

```
SUBROUTINE TDMA(NI, PHI)
PARAMETER (NX = 101)
DIMENSION PHI(NI), P(NX), Q(NX)
COMMON
&/COEF/AP(NX), AE(NX), AW(NX), SU(NX), SP(NX)

NIM1 = NI - 1

P(1) = AE(1)/AP(1)
Q(1) = SU(1)/AP(1)
DO 100 I = 2, NI
    TERM = 1./(AP(I) - AW(I)*P(I-1))
    P(I) = AE(I)*TERM
    Q(I) = (SU(I) + AW(I)*Q(I-1))*TERM
100 CONTINUE

PHI(NI) = Q(NI)
DO 110 II = 2, NIM1
    I = NI + 1 - II
    PHI(I) = P(I)*PHI(I+1) + Q(I)
110 CONTINUE
RETURN
END
```

6.5 例 5

laminar.for [3] よりコードを示す . 方程式は次式で表されている .

$$d_i \phi_P = a_i \phi_N + b_i \phi_S + c_i \quad (63)$$

ここでは 2 次元の問題を対象として線順法を用いているが , 南北方向にのみ走査している .

```
SUBROUTINE LISOLV(ISTART, JSTART, NI, NJ, PHI)
PARAMETER (NX=12, NY=12)
DIMENSION PHI(NX, NY), A(NY), B(NY), C(NY), D(NY)
COMMON
&/COEF/AP(NX, NY), AN(NX, NY), AS(NX, NY),
& AE(NX, NY), AW(NX, NY), SU(NX, NY),
```

```

&      SP(NX, NY)

NIM1 = NI - 1
NJM1 = NJ - 1
JSTM1 = JSTART - 1

A(JSTM1) = 0.
DO 100 I = ISTART, NIM1
  C(JSTM1) = PHI(I, JSTM1)
  DO 110 J = JSTART, NJM1
    A(J) = AN(I, J)
    B(J) = AS(I, J)
    C(J) = AE(I, J)*PHI(I+1, J)
&      + AW(I, J)*PHI(I-1, J) + SU(I, J)
    D(J) = AP(I, J)
    TERM = 1./(D(J) - B(J)*A(J-1))
    A(J) = A(J)*TERM
    C(J) = (C(J) + B(J)*C(J-1))*TERM
110  CONTINUE
  DO 120 JJ = JSTART, NJM1
    J = NJ + JSTM1 - JJ
    PHI(I, J) = A(J)*PHI(I, J+1) + C(J)
120  CONTINUE
100 CONTINUE
  RETURN
  END

```

ここでは、配列 P_i , Q_i を用意する代わりに配列 a_i , c_i を再利用している。

参考文献

- [1] スハス V. パタンカー：コンピュータによる熱移動と流れの数値解析，森北出版 (1985)。
- [2] 香月正司，中山顕：熱流動の数値シミュレーション，森北出版 (1990)。
- [3] 荒川忠一：数値流体工学，東京大学出版会 (1994)。