

第13章 非圧縮性流れの解法—MAC型解法

3次元流れや乱流の計算には、流速と圧力の方程式を解く原始関数法 (primitive function method) が適している。この種の解法の基本は Navier-Stokes(NS) 方程式と連続方程式を解くものである。しかしながら、実際にこれらの方程式を解こうとすると、NS 方程式から流速を求めることはできるが、連続方程式から残りの変数、圧力を求めることができない。このことは、圧縮性流れの解法で連続方程式から密度、運動方程式から流速、エネルギー方程式から例えば岐点内部エネルギーをそれぞれ求めることができるのとは対照的である。このために非圧縮性流れの解法では何らかの戦略が必要になる。この章には最も一般的に用いられてきた MAC 型の解法について述べる。MAC 法では NS 方程式と、連続方程式に代わるものとして圧力方程式が解かれる。他に非圧縮性流れの有力な解法としては、連続方程式に擬似圧縮項を加えた方程式から圧力を計算する擬似圧縮性法がある。ところで NS 方程式と圧力方程式を解く方法では、解が連続の条件を満足しなくなる虞がある。MAC 法ではこの難点を、いわゆる食違い格子を用いた圧力の差分方程式にある付加項を導入することによってみごとに克服している。なおこのアイデアは、曲線座標格子に拡張することはできるが、任意形状要素を用いる有限要素法に拡張することは難しい。そのために有限要素法では、NS 方程式と連続方程式をガレルキン法で同時に解く方法、ペナルティ関数法、ラグランジュ剰数法、ソレノイダル重み関数を用いる方法など各種の解法が考案されてきた。

13.1 MAC 法

Harlow-Welch によって提案された MAC(marker and cell) 法¹、はじめはラグランジュ座標系におけるマーカー粒子の動きを求めると同時にオイラー座標系における格子セルの方程式を解くものであった。その後マーカー粒子は自由表面にのみ置かれるようになり、通常は格子セルの方程式のみが解かれるようになった。ここには格子セルの方程式の解き方について述べる。MAC 法の基礎方程式は次の保存形で書かれた Navier-Stokes 方程式と圧力方程式である。

$$\frac{d\mathbf{u}}{dt} \equiv \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{u}\mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g} \quad (13.1)$$

$$\nabla^2 p = -\nabla \cdot (\nabla \cdot \mathbf{u}\mathbf{u}) + \nabla \cdot \mathbf{g} \quad (13.2)$$

ただしこの式は無次元化されており、 \mathbf{g} は重力加速度などの流体に作用する物体力である。圧力方程式 (13.2) は式 (13.1) の発散を取り、連続の条件

$$\nabla \cdot \mathbf{u} = 0 \quad (13.3)$$

を考慮することによって導かれたものである。MAC 法では定常流れも非定常流れも時間進行法で求められ、式 (13.1) の初期値問題から \mathbf{u} 、式 (13.2) の境界値問題から p が交互に繰り返し計算される。

¹Harlow, F.H. and Welch, J.E., Numerical calculation of time dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids*, Vol.8(1965), 2182-89.

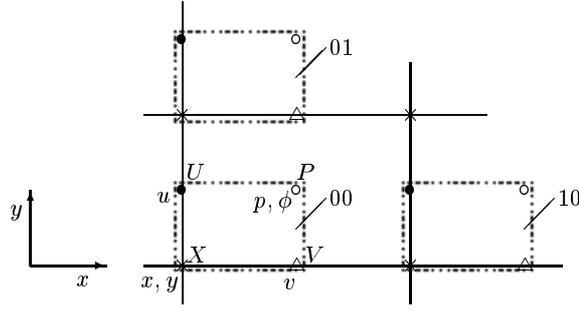


図 13.1: MAC 法の格子

計算には図 13.1 に示す食違い格子 (staggered grid) が用いられる。格子点上に x, y , 格子セルの中心に圧力 p , 辺の中心に流速成分 u, v が定義される。ここでは図に点線で囲んだ 4 点に同一添字を用いることとし、必要に応じ添字 X, P, U, V を付けこれらの点を区別することにする。普通にセル中心に添字 ij , 辺の中心に $i+1/2, j, i, j+1/2, \dots$ を付ける場合, あるいはセル中心点に添字 P , 辺の中心に東西南北にちなんで n, e, w, s , 隣接セル中心に N, E, W, S を付けることもある。式 (13.1) の差分方程式は, 時間微分に Euler 前進差分, 空間微分に 2 次中心差分を用いれば次のようになる。

$$u_{00}^{n+1} = u_{00}^n - \Delta t \left[f_{00} + \frac{p_{00} - p_{-1,0}}{\Delta x} \right]^n, \quad (13.4a)$$

$$v_{00}^{n+1} = v_{00}^n - \Delta t \left[g_{00} + \frac{p_{00} - p_{0,-1}}{\Delta y} \right]^n \quad (13.4b)$$

ただし

$$\begin{aligned} f_{00} &= \frac{(u^2)_{00P} - (u^2)_{-1,0P}}{\Delta x} + \frac{(vu)_{01X} - (vu)_{00X}}{\Delta y} - \nu \Lambda u_{00} - (g_x)_{00}, \\ g_{00} &= \frac{(uv)_{10X} - (uv)_{00X}}{\Delta x} + \frac{(v^2)_{00P} - (v^2)_{0,-1P}}{\Delta y} - \nu \Lambda v_{00} - (g_y)_{00} \\ \Lambda u_{00} &= \frac{u_{-1,0} - 2u_{00} + u_{10}}{\Delta x^2} + \frac{u_{0,-1} - 2u_{00} + u_{01}}{\Delta y^2} \end{aligned} \quad (13.4c)$$

Λ は Laplace 差分演算子である。式 (13.4c) の u^2, vu, uv, v^2 は 2 次中心差分では

$$\begin{aligned} (u^2)_{00P} &= (u_{00} + u_{10})^2/4, & (v^2)_{00P} &= (v_{00} + v_{01})^2/4, \\ (vu)_{00X} &= (uv)_{00X} = (u_{0,-1} + u_{00})(v_{-1,0} + v_{00})/4 \end{aligned}$$

のようになる。なおレイノルズ数が小さく不安定になる場合には, 2 次中心差分の代わりに後で示す上流差分スキームを用いるべきである。

MAC 法の圧力の差分方程式は, 直接 圧力方程式 (13.2) の差分近似を取るのではなく次のように導出される。圧力方程式は Navier-Stokes 方程式の発散を取り連続の条件を考慮することによって導かれたが, 圧力の差分方程式も, Navier-Stokes 差分方程式 (13.4) の発散を数値的に取って, すなわち $\{(13.4a)_{10} - (13.4a)_{00}\}/\Delta x + \{(13.4b)_{01} - (13.4b)_{00}\}/\Delta y$ と置いて導かれる。

$$\begin{aligned} D_{00}^{n+1} &= D_{00}^n - \Delta t \left[\frac{f_{10} - f_{00}}{\Delta x} + \frac{g_{01} - g_{00}}{\Delta y} \right]^n - \Delta t \Lambda p_{00}^n \\ D_{00} &= \frac{u_{10} - u_{00}}{\Delta x} + \frac{v_{01} - v_{00}}{\Delta y} \end{aligned}$$

また連続の条件については、 D_{00}^{n+1} は次に求める流速 $u_{ij}^{n+1}, v_{ij}^{n+1}$ が連続の条件を満足するように 0 とするが、 D_{00}^n は本来は 0 になるべき量であるがすでに求めた流速 u_{ij}^n, v_{ij}^n に対しては一般に 0 にはならないので残しておく。結局圧力の差分方程式は次のようになる。

$$Ap_{00} = -\frac{f_{10}-f_{00}}{\Delta x} - \frac{g_{01}-g_{00}}{\Delta y} + \frac{D_{00}}{\Delta t} \tag{13.5}$$

なお MAC 法の差分方程式はもともとは有限体積法の定式化から導出されたものである。

MAC 法で解を時間ステップ Δt 進める計算手順は次のようになる。

- まず既知の u^n, p^n, f^n, g^n の値を用い式 (13.4) から陽的に u^{n+1} の値を計算する。
- 次にこの u^{n+1} を用い f, g, D の値を更新する。
- それから式 (13.5) の連立 1 次方程式を反復法で解いて p の値を更新する。

次に MAC 法の特徴について述べる。食違い格子は、一見解法を複雑にしているように思われるが、式 (13.4) の圧力項と発散 D の表現を簡単にし、結果として連続の条件 $D_{00}^{n+1} = 0$ を満足する圧力の差分方程式をコンパクトにしている。この圧力の差分方程式は、普通の格子を用いた差分法や四辺形混合要素を用いた有限要素法に見られる spurious 誤差の発生を完全に防いでいる²。圧力の差分方程式 (13.5) と単に圧力方程式 (13.2) を差分近似したものとを比較すれば、前者は後者に付加項 $D_{00}/\Delta t + \nu \Delta D_{00}$ を考慮したものであることが分かる。圧力の差分方程式は通常反復法で解かれ、完全に収束しないうちに次のステップに移る。このときの圧力の差分方程式 (13.5) の残差を r_{00} とすれば、次に求める u^{n+1} に対して連続の条件の誤差は

$$D_{00}^{n+1} = -\Delta t r_{00}$$

のように見積られる。この誤差は良性のもので解が収束すれば 0 に近づく。一方付加項のない圧力の差分方程式の場合には、 u^{n+1} に対して連続の条件の誤差は

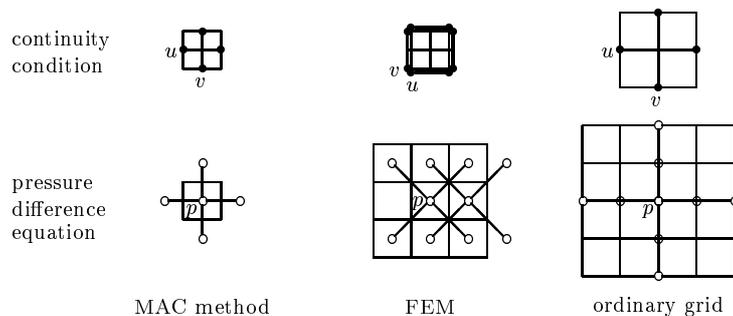
$$D_{00}^{n+1} = (1 + \Delta t \nu \Delta) D_{00}^n - \Delta t r_{00}$$

のようになる。この誤差は集積し計算を不能にする。そのためにこの種の解法ではこの章の冒頭に述べた何らかの戦略が必須になる。

² 下図に食違い格子を用いる MAC 法、各要素の中心に p 、4 頂点に u, v の定義される四辺形混合要素を用いる有限要素法、通常格子を用いる差分法に対し、連続の条件と圧力差分方程式を比較する。正方形要素の場合にはこの FEM の圧力差分方程式は

$$\frac{p_{11}+p_{-1,1}+p_{1,-1}+p_{-1,-1}-4p_{00}}{2h^2} = -\frac{f_{10}-f_{00}+f_{11}-f_{01}}{2h} - \frac{g_{01}-g_{00}+g_{11}-g_{10}}{2h} + \frac{D_{00}}{\Delta t}$$

のようになり、圧力 p_{00} は隣の $p_{10}, p_{-1,0}, p_{01}, p_{0,-1}$ とは無関係になり、斜め隣の p_{11}, \dots と結び付く。この圧力 p_{ij} の方程式は、 $i+j$ が偶数の点のものと奇数の点のものが独立に解かれることになり、そのために圧力の値は checker-board 状に波打つ。長方形や任意四辺形要素でもこのような傾向は強く現れる。この波打ち誤差は spurious 誤差といわれ、波打ちを鎮めるために間に合せの手段として平滑化 (smoothing) が広く行われている。



13.2 SMAC 法

この節の説明の都合上，MAC 法は Navier-Stokes 方程式と圧力の Poisson 方程式

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \Delta t (\nabla \cdot \mathbf{u}\mathbf{u} + \nabla p - \nu \nabla^2 \mathbf{u} - \mathbf{g})^n \quad (13.6a)$$

$$\nabla^2 p = \nabla \cdot \mathbf{u} / \Delta t - \nabla \cdot (\nabla \cdot \mathbf{u}\mathbf{u} - \nu \nabla^2 \mathbf{u} - \mathbf{g}) \quad (13.6b)$$

を用い， \mathbf{u}^n が与えられるときに式 (13.6b) から p^n ，式 (13.6a) から \mathbf{u}^{n+1} を次々に計算する陽解法 (explicit method) と見ることにする。

Amsden-Harlow の SMAC 法 (simplified MAC method) は，Navier-Stokes 方程式 (13.6a) に時間分離法 (time-splitting method) を適用し，圧力そのものではなく圧力の増分 ϕ の Poisson 方程式を解くものである。その基礎方程式は

$$\mathbf{u}^* = \mathbf{u}^n - \Delta t (\nabla \cdot \mathbf{u}\mathbf{u} + \nabla p - \nu \nabla^2 \mathbf{u} - \mathbf{g})^n \quad (13.7a)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi \quad (13.7b)$$

$$\nabla^2 \phi = \nabla \cdot \mathbf{u}^* / \Delta t, \quad p^{n+1} = p^n + \phi \quad (13.7c)$$

となる。計算には MAC 法と同様の食違い格子と差分近似式が用いられ，解を時間ステップ Δt 進める計算手順は次の通りである。

- \mathbf{u}^n が与えられるときに，式 (13.7a) を陽的に解いて中間の流速 \mathbf{u}^* を計算
- 式 (13.7c) を反復法で解いて圧力の増分 ϕ を計算，また p^{n+1} を計算
- 式 (13.7b) から \mathbf{u}^{n+1} を計算

式 (13.7a) と式 (13.7b) を加えれば

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \Delta t \{ (\nabla \cdot \mathbf{u}\mathbf{u} - \nu \nabla^2 \mathbf{u} - \mathbf{g})^n + \nabla p^{n+1} \}$$

となるから，この SMAC 法では，Navier-Stokes 方程式の対流項と粘性項にオイラー前進差分，圧力項にオイラー後退差分を適用していることになる。なお式 (13.7c) は式 (13.7b) の発散を取り，連続の条件 $\nabla \cdot \mathbf{u}^{n+1} = 0$ を課して導出されたものである。SMAC 法は MAC 法の特徴をそのまま受け継いでいる。すなわち食違い格子が用いられ，また圧力の差分方程式には一見 MAC 法のような付加項は見当たらないが， \mathbf{u}^* を介して間接的に考慮されている。定常流れを時間進行法で解く場合に，解が収束すれば

$$(\nabla \cdot \mathbf{u}\mathbf{u} + \nabla p - \nu \nabla^2 \mathbf{u} - \mathbf{g})^n \rightarrow 0; \quad \mathbf{u}^* \rightarrow \mathbf{u}^n, \quad \nabla \cdot \mathbf{u}^* \rightarrow 0, \quad \phi \rightarrow 0, \quad \mathbf{u}^{n+1} \rightarrow \mathbf{u}^*$$

となり，式 (13.7a)–(13.7c) の差分近似式の残差はすべて 0 に近付き，修正値も 0 に近づくようになる。

ここで SMAC 法を用い，よく解かれているバックステップ流路 (backward-facing stepped duct) の流れを計算する。計算領域と格子を図 13.2 に示す。ステップの高さ 1 とし，座標の原点をステップの角に取り，その格子番号を $i, j = 0$ とする。計算格子は等間隔長方形食違い格子 (uniform rectangular staggered grid) で，格子間隔は $\Delta x = 1/3$, $\Delta y = 1/5$ である。このとき上流と下流の境界は $i_0 = -30$, $i_1 = 90$ ，また下流側下方境界と上方境界は $j_0 = -5$, $j_1 = 15$ になる。上流境界に流速 u ，下流境界に静圧 p ，また流れのレイノルズ数 Re の値を与え，定常の層流として計算する。次に FORTRAN 95/90 free source form で書かれたプログラムを示す。



図 13.2: バックステップ流路の計算格子

```

PROGRAM MAIN
!*****
! Flow Problem: Steady 2D incompressible Flow through Backward-Stepped Duct
! Numerical Scheme: SMAC, Rectangular Grid, Central-Differences or Chakravarthy-Osher TVD Scheme
!*****
PARAMETER(i0=-30,i1=90,j0=-5,j1=15)
DIMENSION u(i0:i1,j0:j1),v(i0:i1,j0:j1),uX(i0:i1,j0:j1),vX(i0:i1,j0:j1),p(i0:i1,j0:j1), &
  phi(i0:i1,j0:j1),c(0:4,i0:i1,j0:j1),z(i0:i1,j0:j1),f(i0:i1,j0:j1),locf(2,8),fmax(8), &
  al(-2:8)
COMMON //na,Re,dx,dy,dt
CHARACTER*10 z1,z2(8)
DATA naf,Re,dx,dy,dt/10000, 100., .33333, .2, .04/ !naf=na_max, Re=Reynolds number
OPEN(20,FILE='OUTPUT.dat')
! initial values: parabolic velocity profile
DO j=0,j1-1; y=(j+.5)*dy !pustream side duct
  FORALL(i=i0:0)u(i,j)=(2./3.)*y*(3.-y)
ENDDO
DO j=j0,j1-1; y=(j-j0+.5)*dy; u1=(9./32.)*y*(4.-y) !downstream side duct
  FORALL(i=1:i1)u(i,j)=((90.-i)*u(0,j)+i*u1)/90.
ENDDO
! Recovery pressure of the diverging duct: dp=coef*(1-.75*.75)/2, far-upstream and -downstream
! pressure gradient: -4/3/Re, -9/16/Re, pressure losses of the duct: Dp=(4/3*10+9/16*30)/Re-dp
CALL COEF(c,i0,i1,j0,j1)
na=0; 100 na=na+1 !na=iteration number
CALL COMPZ(u,v,z,i0,i1,j0,j1) !compute zeta
CALL EXPCD(u,v,uX,vX,p,z,i0,i1,j0,j1,locf,fmax) !compute u*,v* by explicit
CALL COMPP(u,v,p,phi,c,i0,i1,j0,j1,resP,locf,fmax) !compute phi,p
CALL COMPU(u,v,phi,i0,i1,j0,j1,locf,fmax) !compute u,v
! Record converging process and decide convergence
IF(na==1)WRITE(20,'(A46)') na resNS resP outflow CPU-time'
IF(MOD(na,20)==0)THEN
  i=i1; resNS=0.
  resNS = AMAX1(fmax(1),-fmax(2),fmax(3),-fmax(4))
  flow = 3.*dy/8.*(3.*u(i,-5)+u(i,-4)+(u(i,10)+3.*u(i,11) & !outlet flow rate
    +3.*u(i,12)+u(i,13))+u(i,13)+3.*u(i,14))
  DO j=-4,8,2; flow = flow+dy/3.*(u(i,j)+4.*u(i,j+1)+u(i,j+2)); ENDDO
  CALL CPU_TIME(sec) !CPU-time
  WRITE(20,'(I5,4F10.5)')na,resNS,resP,flow,sec
  umin=0.; DO j=j0,j1; umin=AMIN1(umin,u(i1,j)); ENDDO !search outlet backward flow
  IF(umin<0.)WRITE(20,'(A60)')'should extend the duct to remove outlet backward flow'
ENDIF
IF(resP>10.) GOTO 110 !divergence
IF(resNS<1.E-4.AND.resP<1.E-4) GOTO 110 !convergence
IF(na<naf) GOTO 100 !iteration
! Output computational results
110 CONTINUE
DO k=1,4

```

```

SELECT CASE(k)
  CASE(1); z1='velocity u'; FORALL(i=i0:i1,j=j0:j1)f(i,j)=uX(i,j)
  CASE(2); z1='velocity v'; FORALL(i=i0:i1,j=j0:j1)f(i,j)=vX(i,j)
  CASE(3); z1=' pressure '; FORALL(i=i0:i1,j=j0:j1)f(i,j)= p(i,j)
  CASE(4); z1='vorticity '; FORALL(i=i0:i1,j=j0:j1)f(i,j)= z(i,j)
ENDSELECT
WRITE(20,'(/1H 10X A7,A10,A7,10X A4,I5/))'***** ',z1,' *****','na =',na
DO ii=i0,i1-40,40
  DO j=j1,j0,-1; WRITE(20,'(I3,41F8.4,I3)')j,(f(i,j),i=ii,ii+40),j; ENDDO
  WRITE(20,'(41I8/))'(i,i=ii,ii+40)
ENDDO
ENDDO
! Compute and output reattachment point(rap)
irap=5; 10 irap=irap+1; IF(u(irap,j0)<0.) GOTO 10 !irap: grid point near rap
ii=-3; 12 ii=ii+1; i=irap+ii
du=u(i,j0); d2u=uX(i,j0+1)-2.*du !du=Delta u, d2u=Delta^2u
d3u=u(i,j0+1)-3.*uX(i,j0+1)+3.*du !d3u=Delta^3u
a=2.; 13 fa = du+(a-1.)/2.*(d2u+(a-2.)/3.*d3u) !fa=f(alpha) by cubic interpolation
IF(ABS(fa)<.0001) GOTO 14
dfa = d2u/2.+(2.*a-3.)/6.*d3u !dfa=f'(alpha)
a=a-fa/dfa; GOTO 13 !by Newton method
14 al(ii)=a; IF(a>0.14) GOTO 12 !al(iii): alpha of u=0
IF(al(ii)<0.01)ii=ii-1 !irap+iii: adjacent grid point to rap
xrap = (irap+ii+al(ii)/(al(ii-1)-al(ii)))*dx !comp rap by linear extrapolation
WRITE(20,'(5X A7,F5.1)')'xrap = ',xrap
! Output locations and values of maximum and minimum of residuals and divergence
DATA z2/'maxdu =','mindu =','maxdv =','mindv =','maxphi =','minphi =','maxdiv =','mindiv ='/
FORALL(i=1:8)locf(1,i)=locf(1,i)-31; FORALL(i=1:8)locf(2,i)=locf(2,i)-6
DO i=1,8; WRITE(20,'(5X A10,2I4,3X E10.3)')z2(i),(locf(k,i),k=1,2),fmax(i); ENDDO
CLOSE(20)
CALL BSDuct_CG(uX,vX,p,z)
STOP
END PROGRAM MAIN

! ***** Compute u* and v* by explicit scheme using central-differences
SUBROUTINE EXPCD(u,v,uX,vX,p,z,i0,i1,j0,j1,locf,fmax)
!PARAMETER(i0=-30,i1=90,j0=-5,j1=15)
DIMENSION u(i0:i1,j0:j1),v(i0:i1,j0:j1),uX(i0:i1,j0:j1),vX(i0:i1,j0:j1),p(i0:i1,j0:j1), &
z(i0:i1,j0:j1),du(i0:i1,j0:j1),dv(i0:i1,j0:j1),w1(-30:90),lo(2),locf(2,8),fmax(8)
COMMON //na,Re,dx,dy,dt
DO i=i0,i1; j00=0; IF(i>0)j00=j0
  FORALL(j=j00+1:j1-1)uX(i,j)=(u(i,j-1)+u(i,j))/2. !u_X(u at point X)
ENDDO
DO j=j0+1,j1-1; i00=0; IF(j>0)i00=i0
  FORALL(i=i00+1:i1-1)vX(i,j)=(v(i-1,j)+v(i,j))/2. !v_X
  i=i1; vX(i,j)=v(i-1,j) !outlet
ENDDO
FORALL(i=i0:i1,j=j0:j1)du(i,j)=0.; FORALL(i=i0:i1,j=j0:j1)dv(i,j)=0.
DO j=j0,j1-1; i00=0; IF(j>=0)i00=i0 !computation of u*
  FORALL(i=i00 :i1-1)w1(i)=(u(i,j)+u(i+1,j))*(u(i,j)+u(i+1,j))/4. !w1=uu at P
  FORALL(i=i00+1:i1-1)du(i,j)=(w1(i)-w1(i-1))/dx
  i=i1; du(i,j)=u(i,j)*(u(i,j)-u(i-1,j))/dx !outlet
DO i=i00+1,i1
  du(i,j)=du(i,j)+(vX(i,j+1)*uX(i,j+1)-vX(i,j)*uX(i,j))/dy &
  +(p(i,j)-p(i-1,j))/dx+(z(i,j+1)-z(i,j))/dy/Re !corrections of NS eq
ENDDO
ENDDO

```

```

DO j=j0+1,j1-1; i00=0; IF(j>0)i00=i0 !computation of v*
  FORALL(i=i00:i1-2)dv(i,j)=(uX(i+1,j)*vX(i+1,j)-uX(i,j)*vX(i,j))/dx
  i=i1-1; dv(i,j)=(uX(i,j)+uX(i+1,j))/2.*(v(i,j)-v(i-1,j))/dx !near outlet
ENDDO
DO i=i0,i1-1; j00=0; IF(i>=0)j00=j0
  FORALL(j=j00:j1-1)w1(j)=(v(i,j)+v(i,j+1))*(v(i,j)+v(i,j+1))/4. !w4=vv at P
  DO j=j00+1,j1-1
    dv(i,j)=dv(i,j)+(w1(j)-w1(j-1))/dy &
    +(p(i,j)-p(i,j-1))/dy-(z(i+1,j)-z(i,j))/dx/Re !corrections of NS eq
  ENDDO
ENDDO
FORALL(i=i0+1:i1,j=j0 :j1-1)u(i,j)=u(i,j)-dt*du(i,j) !u* at U
FORALL(i=i0:i1-1,j=j0+1:j1-1)v(i,j)=v(i,j)-dt*dv(i,j) !v* at V
lo=MAXLOC(du); FORALL(k=1:2)locf(k,1)=lo(k); fmax(1)=MAXVAL(du)
lo=MINLOC(du); FORALL(k=1:2)locf(k,2)=lo(k); fmax(2)=MINVAL(du)
lo=MAXLOC(dv); FORALL(k=1:2)locf(k,3)=lo(k); fmax(3)=MAXVAL(dv)
lo=MINLOC(dv); FORALL(k=1:2)locf(k,4)=lo(k); fmax(4)=MINVAL(dv)
END SUBROUTINE EXPCD

! ***** Set up coefficients of Laplace difference operator
SUBROUTINE COEF(c,i0,i1,j0,j1)
!PARAMETER(i0=-30,i1=90,j0=-5,j1=15)
DIMENSION c(0:4,i0:i1,j0:j1)
COMMON //na,Re,dx,dy,dt
rsdx=1./dx/dx; rsdy=1./dy/dy
DO j=j0,j1-1; i00=0; IF(j>=0)i00=i0
  FORALL(i=i00:i1-1)
    c(1,i,j)=rsdx; c(2,i,j)=rsdy
    c(3,i,j)=rsdy; c(4,i,j)=rsdy; c(0,i,j)=-2.*(rsdx+rsdy)
  ENDFORALL
  i= i00; c(1,i,j)=0.; c(0,i,j)=c(0,i,j)+rsdx !inlet & step, phi_x=0
ENDDO
DO i=i0,i1-1; j=0; IF(i>=0)j=j0
  c(3,i,j)=0.; c(0,i,j)=c(0,i,j)+rsdy !bottoms, phi_y=0
  j=j1-1; c(4,i,j)=0.; c(0,i,j)=c(0,i,j)+rsdy !top, phi_y=0
ENDDO
END SUBROUTINE COEF

! ***** Solve bound value problem of Poisson eqn for pressure inclement phi
SUBROUTINE COMPP(u,v,p,phi,c,i0,i1,j0,j1,resp,locf,fmax)
!PARAMETER(i0=-30,i1=90,j0=-5,j1=15)
DIMENSION u(i0:i1,j0:j1),v(i0:i1,j0:j1),p(i0:i1,j0:j1),phi(i0:i1,j0:j1),c(0:4,i0:i1,j0:j1), &
a(120,3),b(120),rhs(-30:89,-5:14),w(j0:j1),lo(2),locf(2,8),fmax(8)
COMMON //na,Re,dx,dy,dt
n1=120; n2=20; kf=4; al=1.0; bt=0.5 !alpha:accel, beta:damping
FORALL(j=j0:j1-1)p(i1,j)=0. ! Dirichlet cond
! Compute rhs
DO i=i0,i1-1; j00=0; IF(i>=0)j00=j0; DO j=j00,j1-1
  phi(i,j)=0.
  rhs(i,j)=((u(i+1,j)-u(i,j))/dx+(v(i,j+1)-v(i,j))/dy)/dt !rhs
ENDDO; ENDDO
! Solve simultaneous linear eqs by ADI-method
k=0; 40 k=k+1 !i-sweep
DO i=i0,i1-1; im1=MAX(i-1,i0); j00=0; IF(i>=0)j00=j0
  DO j=j00,j1-1; k=j-j00+1 !setup coefs & rhs of linear eqs
    a(k,1)=c(3,i,j); a(k,2)=c(0,i,j)*al; a(k,3)=c(4,i,j)
    b(k)=rhs(i,j)-c(1,i,j)*phi(im1,j)-c(2,i,j)*phi(i+1,j)-(1.-al)*c(0,i,j)*phi(i,j)
  ENDDO
ENDDO

```

```

ENDDO
CALL GAUSS3(a,b,n1,j1-j00) !solve linear eqs
FORALL(j=j00:j1-1)phi(i,j)=b(j-j00+1) !phi at P
ENDDO
k=k+1 !j-sweep
DO j=j0,j1-1; jml=MAX(j-1,j0); i00=0; IF(j>=0)i00=i0
DO i=i00,i1-1; k=i-i00+1 !setup coefs & rhs of linear eqs
a(k,1)=c(1,i,j); a(k,2)=c(0,i,j)*al; a(k,3)=c(2,i,j)
b(k)=rhs(i,j)-c(3,i,j)*phi(i,jml)-c(4,i,j)*phi(i,j+1)-(1.-al)*c(0,i,j)*phi(i,j)
ENDDO
CALL GAUSS3(a,b,n1,i1-i00) !solve linear eqs
FORALL(i=i00:i1-1)phi(i,j)=b(i-i00+1) !phi at P
ENDDO
IF(k<kf) GOTO 40
! Compute residuals of phi equations and determine pressure p
resp=0.
DO i=i0,i1-1; iml=MAX(i-1,i0); j00=0; IF(i>=0)j00=j0
DO j=j00,j1-1; jml=MAX(j-1,j0)
res=c(1,i,j)*phi(iml,j)+c(2,i,j)*phi(i+1,j)+c(3,i,j)*phi(i,jml) &
+c(4,i,j)*phi(i,j+1)+c(0,i,j)*phi(i,j)-rhs(i,j) !residuals
resp=MAX(resp,ABS(res)) !max residual of phi fde
p(i,j)=p(i,j)+bt*phi(i,j) !corrected p
ENDDO; ENDDO
lo=MAXLOC(phi); FORALL(k=1:2)locf(k,5)=lo(k); fmax(5)=MAXVAL(phi)
lo=MINLOC(phi); FORALL(k=1:2)locf(k,6)=lo(k); fmax(6)=MINVAL(phi)
END SUBROUTINE COMPP

! ***** Compute u and v from u* and v*
SUBROUTINE COMPU(u,v,phi,i0,i1,j0,j1,locf,fmax)
DIMENSION u(i0:i1,j0:j1),v(i0:i1,j0:j1),phi(i0:i1,j0:j1),d(i0:i1,j0:j1),lo(2),locf(2,8),fmax(8)
COMMON //na,Re,dx,dy,dt
DO j=j0,j1-1; i00=0; IF(j>=0)i00=i0
FORALL(i=i00+1:i1)u(i,j)=u(i,j)-dt*(phi(i,j)-phi(i-1,j))/dx !u at U
ENDDO
DO j=j0+1,j1-1; i00=0; IF(j>0)i00=i0
FORALL(i=i00:i1-1)v(i,j)=v(i,j)-dt*(phi(i,j)-phi(i,j-1))/dy !v at V
ENDDO
! Compute divergent=u_x+v_y at point P
FORALL(i=i0:i1,j=j0:j1)d(i,j)=0.
DO i=i0,i1-1; j00=0; IF(i>=0)j00=j0
FORALL(j=j00:j1-1)d(i,j)=(u(i+1,j)-u(i,j))/dx+(v(i,j+1)-v(i,j))/dy
ENDDO
lo=MAXLOC(d); FORALL(k=1:2)locf(k,7)=lo(k); fmax(7)=MAXVAL(d)
lo=MINLOC(d); FORALL(k=1:2)locf(k,8)=lo(k); fmax(8)=MINVAL(d)
END SUBROUTINE COMPU

! ***** Compute vorticity
SUBROUTINE COMPZ(u,v,z,i0,i1,j0,j1)
!PARAMETER(i0=-30,i1=90,j0=-5,j1=15)
DIMENSION u(i0:i1,j0:j1),v(i0:i1,j0:j1),z(i0:i1,j0:j1)
COMMON //na,Re,dx,dy,dt
! Compute zeta=v_x-u_y at point X
FORALL(i=i0:i1,j=j0:j1)z(i,j)=0. !zeta=v_x on inlet, outlet, top & bottoms
DO j=j0+1,j1-1; i00=0; IF(j>0)i00=i0
FORALL(i=i00+1:i1-1)z(i,j)=(v(i,j)-v(i-1,j))/dx !zeta=v_x
ENDDO
FORALL(j=j0+1:0)z(0,j)=(3.*v(0,j)-v(1,j)/3.)/dx !step

```

```

DO j=j0+1,j1-1; i00=1; IF(j>0)i00=i0
  FORALL(i=i00:i1)z(i,j)=z(i,j)-(u(i,j)-u(i,j-1))/dy          !zeta=zeta-u_y
ENDDO
DO i=i0,i1; j=0; IF(i>0)j=j0
  z(i,j)=z(i,j)-(3.*u(i,j)-u(i,j+1)/3.)/dy          !bottom walls containing corner point
  j=j1; z(i,j)=(3.*u(i,j-1)-u(i,j-2)/3.)/dy          !top
ENDDO
END SUBROUTINE COMPZ

! ***** Solve simultaneous linear eqns with tri-diagonal matrix
SUBROUTINE GAUSS3(a,b,n1,n)
DIMENSION a(n1,3),b(n1)
DO k=1,n-1
  b(k)=b(k)/a(k,2);          a(k,3)=a(k,3)/a(k,2)
  b(k+1)=b(k+1)-a(k+1,1)*b(k); a(k+1,2)=a(k+1,2)-a(k+1,1)*a(k,3)
ENDDO
b(n)=b(n)/a(n,2)
DO k=n-1,1,-1; b(k)=b(k)-a(k,3)*b(k+1); ENDDO
END

```

メインプログラムでは、はじめに計算に必要なデータ、初期・境界値が入力される。次に SMAC 法の反復計算がサブルーチンを引用するかたちで行われる。それから計算終了時点で流速 u , v , 静圧 p などの計算結果が出力される。このプログラムでは、上流境界に平行平板間流路の十分発達した層流の流速分布、すなわち放物線速度分布 (平均速度 1) が与えられる。このとき、後に述べる Courant 数 $CFL = u_{\max} \Delta t / \Delta x < 1$ という条件は $\Delta t < \Delta x / u_{\max} = 0.222$ になる。しかしながら実際の計算では時間ステップ Δt は 0.04 程度以下でないと解は収束しない。また Reynolds 数 $Re = 1/\nu$ は、対流項を中心差分で近似するときには、文献等にも見られるように、約 70 以下でないと収束しない。ここでは $\Delta t = 0.04$, $Re = 70$ に選ぶことにする。流れが平均流速でこの流路を通り抜ける時間は約 50 である。したがって最大反復数 n_{af} は少なくとも 1000 のオーダー必要である。 $res_{NS} = res_{NS}$ は Navier-Stokes 差分方程式の最大残差値, $res_p = res_p$ は圧力差分方程式の最大残差値を表すものである。 $flow = flow$ は Simpson の 1/3 と 3/8 公式を用いて計算した出口流量で本来は 3.0 になるべきものである。これらの量は解の収束状況を示す指標として、CPU 時間とともに出力される。収束判定は $res_{NS} < 0.0001$ かつ $res_p < 0.0001$ によって行われる。

再付着距離は u の出力をもとに作図で求めることもできるが、ここではメインプログラムの終わりに計算で求めることにする。再付着距離 x_{rap} の位置は $u_{i_{j_0}}$ の値が負になる領域 $5 < i < i_{rap}$ の下流側にある。まず配列 $a1$ を用意し、各格子線 $x = const.$ 上で $u = 0$ になる点を求める。下壁面からの無次元距離 $\alpha = y / (\Delta y / 2)$ を導入すれば、この点の位置 α は次の 3 次方程式で表される。

$$u(\alpha) = u_0 + \alpha \left[\Delta u_0 + \frac{1}{2}(\alpha-1) \left(\Delta^2 u_0 + \frac{1}{3}(\alpha-2) \Delta^3 u_0 \right) \right] = 0$$

ただし $u_0 = 0$, $\Delta u_0 = u_{j_0}$, $\Delta^2 u_0 = u_{j_0+1X} - 2u_{j_0}$, $\Delta^3 u_0 = u_{j_0+1} - 3u_{j_0+1X} + 3u_{j_0}$, 添え字 i は省略。この式は 3 次のニュートン前進補間公式に基づくものである。この式で $\alpha = 0$ は下壁面上で流速 0 を意味し、大括弧の中 = 0, すなわち

$$f(\alpha) \equiv \Delta u_0 + \frac{1}{2}(\alpha-1) \left(\Delta^2 u_0 + \frac{1}{3}(\alpha-2) \Delta^3 u_0 \right) = 0$$

は流れの中のある点で流速 0 を示す。この式の根 α はニュートン法で容易に求めることができる。

$$\alpha^{n+1} = \alpha^n - f(\alpha^n) / f'(\alpha^n)$$

$$f'(\alpha) = \frac{1}{2} \Delta^2 u_0 + \frac{1}{6} (2\alpha - 3) \Delta^3 u_0$$

このようにして $\alpha > 0$ の範囲すなわち剥離泡内で α の値が求められる．再付着距離は $\alpha = 0$ になる x で次式から求められる．

$$x_{rap} = \left(i_{rap} + I + \frac{\alpha I}{\alpha_{I-1} - \alpha_I} \right) dx$$

ただし $i_{rap} + I$ は再付着点の上流側の隣接格子点の番号である．

以下には各サブルーチンを初心者向けにやや詳しく説明する．プログラム中のコメントも参照されたい．プログラムを簡潔にするためにステップの上流部分と下流部分を分けずに同時に処理している．

サブルーチン COMPZ：格子点 X 上の渦度 ζ の値を計算．その計算式は領域の内点では

$$\zeta_{ijX} = (v_x - u_y)_{ijX} = \frac{v_{ij} - v_{i-1,j}}{\Delta x} - \frac{u_{ij} - u_{i,j-1}}{\Delta y} \quad (13.8)$$

また境界点 例えば下方壁面上の点では

$$\zeta_{i0X} = -(u_y)_{i0X} = -\frac{3u_{i0} - u_{i1}/3}{\Delta y} \quad (13.9)$$

この式は滑りなしの条件 $u_{i0X} = 0$ を考慮して導かれたものである．なおこの渦度は Navier-Stokes 方程式の粘性項 $\nu \nabla^2 u = -\nu \zeta_y$, $\nu \nabla^2 v = \nu \zeta_x$ の計算に用いられる．

サブルーチン EXPCD： U 点の u^* と V 点の v^* を式 (13.7a) から計算．配列 $u(i, j)$ は u_{ij}^n , u_{ij}^* または u_{ij}^{n+1} のどれかである．また配列 $du(i, j) = du_{ij}$ は次式の大括弧の中の定常 Navier-Stokes 差分方程式の残差で，その値は解が収束した時点でほぼ 0 になる．式 (13.7a) の差分方程式は対流項を保存形の 2 次中心差分で近似すれば次のようになる．

$$u_{ij}^* = u_{ij}^n - \Delta t \left[\frac{w_{ij}^1 - w_{i-1,j}^1}{\Delta x} + \frac{w_{i,j+1}^2 - w_{ij}^2}{\Delta y} + \frac{p_{ij} - p_{i-1,j}}{\Delta x} + \nu \frac{\zeta_{i,j+1} - \zeta_{ij}}{\Delta y} \right]^n \quad (13.10a)$$

$$v_{ij}^* = v_{ij}^n - \Delta t \left[\frac{w_{i+1,j}^3 - w_{ij}^3}{\Delta x} + \frac{w_{ij}^4 - w_{i,j-1}^4}{\Delta y} + \frac{p_{ij} - p_{i,j-1}}{\Delta y} - \nu \frac{\zeta_{i+1,j} - \zeta_{ij}}{\Delta x} \right]^n \quad (13.10b)$$

ただし $w_{ij}^1 = (u^2)_{ijP}$, $w_{ij}^2 = w_{ij}^3 = (uv)_{ijX}$, $w_{ij}^4 = (v^2)_{ijP}$ である．また下流境界のところでは，反射しないように対流項を 1 次上流差分で近似すれば次のようになる．

$$u_{ij}^* = u_{ij}^n - \Delta t \left[u_{ij} \frac{(u_{ij} - u_{i-1,j})}{\Delta x} + \frac{w_{i,j+1}^2 - w_{ij}^2}{\Delta y} + \frac{p_{ij} - p_{i-1,j}}{\Delta x} + \nu \frac{\zeta_{i,j+1} - \zeta_{ij}}{\Delta y} \right]^n \quad (i = i_1) \quad (13.11a)$$

$$v_{ij}^* = v_{ij}^n - \Delta t \left[u_{ijV} \frac{(v_{ij} - v_{i-1,j})}{\Delta x} + \frac{w_{ij}^4 - w_{i,j-1}^4}{\Delta y} + \frac{p_{ij} - p_{i,j-1}}{\Delta y} - \nu \frac{\zeta_{i+1,j} - \zeta_{ij}}{\Delta x} \right]^n \quad (i = i_1 - 1) \quad (13.11b)$$

上式では下流境界の静圧 $p_{i_1j} = 0$ としている．MAXLOC(du) は du_{ij} の最大値の位置 (i, j) , MAXVAL(du) は du_{ij} の最大値を求める関数で，MINLOC(du) と MINVAL(du) も最小値に関する同様の関数である．

サブルーチン COMPP： ϕ の Poisson 方程式 (13.7c) の境界値問題を解き静圧の修正値 p^{n+1} を求める．この方程式は 2 階の楕円型で，したがって境界条件はすべての境界に関数またはその法線微分の値を与えれば必要十分である．式 (13.7c) の差分方程式は 2 次中心差分を用いれば次のようになる．

$$\frac{\phi_{i-1,j} - 2\phi_{ij} + \phi_{i+1,j}}{\Delta x^2} + \frac{\phi_{i,j-1} - 2\phi_{ij} + \phi_{i,j+1}}{\Delta y^2} = \frac{1}{\Delta t} \left[\frac{u_{i+1,j}^* - u_{ij}^*}{\Delta x} + \frac{v_{i,j-1}^* - v_{ij}^*}{\Delta y} \right] \quad (13.12)$$

またここでは下流境界に関数値 $\phi_{i_1j} = 0$, その他の境界に法線微分値を与えることにする．固定壁境界では式 (13.7b) から $\phi_x = 0$ または $\phi_y = 0$ となる．また上流境界でも $u_{i_0j}^n = u_{i_0j}^* = u_{i_0j}^{n+1}$ であるから $\phi_x = 0$ と

なる． ϕ_{ij} の連立 1 次方程式はここでは ADI 法で解かれる．この連立 1 次方程式の係数，Laplace 差分演算子の係数 c はサブルーチン COEF で設定され，ADI 法における 3 重対角行列の連立 1 次方程式はサブルーチン GAUSS3 を引用し Gauss 消去法で解かれる．配列 a , b , c は 3 重対角行列の連立 1 次方程式の係数，右辺， ϕ_{ij} の連立 1 次方程式の係数の記録に用いられる． $AL = \alpha$ は ADI 法の加速パラメータ， $BT = \beta$ は静圧 p の修正計算における減衰係数である．ADI 法による ϕ の計算は，解があまり収束しない段階で次の時間ステップの計算に移るが，ここでは掃引数 $kf = k_f = 4$ とし x 方向と y 方向を交互に各 2 回掃引している． $p^{n+1} = p^n + \beta \phi$.

サブルーチン COMPU: u^{n+1} と v^{n+1} の修正値を式 (13.7b) から計算．式 (13.7b) の差分方程式は 2 次中心差分を用いれば次のようになる．

$$u_{ij}^{n+1} = u_{ij}^* - \Delta t \frac{\phi_{ij} - \phi_{i-1,j}}{\Delta x}, \quad v_{ij}^{n+1} = v_{ij}^* - \Delta t \frac{\phi_{ij} - \phi_{i,j-1}}{\Delta y} \quad (13.13)$$

下表は解の収束状況を示したものである．表中の res_{NS} は Navier-Stokes 差分方程式 (13.10) と (13.11) の最大残差値， res_P は圧力差分方程式 (13.12) の最大残差値， $\Delta flow = flow-3$ は流速 $u_{i,j}$ から求めた出口流量の誤差である．この表中の数字はすべて 10^5 倍されている．解の収束を $res_{NS} < 0.0001$ かつ $res_P < 0.0001$ から判定することにすれば，収束解は $n = 3039$ で得られることになる．流速は上流側流路中心の 1.5 が最大で，圧力は上流境界の 0.1541 が最大，ステップの角のところの -0.0838 が最小である．なお判定条件を 1 桁厳しくした場合には $n = 4000$ を少し超えるところで収束する．再付着距離 5.4 は変わらない．この計算は $\alpha = 1.0$, $\beta = 0.5$ と置いて得られたものである． $\alpha < 1.0$ 例えば 0.9 にとって ADI 法の計算を加速することは不安定になりできない．またこれらの係数の値を変えても収束にほとんど影響しない．それはこの定常流れの収束が，圧力差分方程式の収束よりも，渦度が流れ場の隅々まで完全に拡散することによって完了するからである．

表 13.1: バックステップ流路流れの解の収束状況 ($Re = 70$, SMAC 法)

n	$\times 10^{-5}$		
	res_{NS}	res_P	$\Delta flow$
100	2585	3176	-1875
1000	513	298	1020
2000	80	15	346
3000	6	13	285
4000	1	1	309

平行平板間流路の圧力降下は上流側が $4/3Re = .01905$ ，下流側が $9/16Re = .00803$ である．計算の結果はそれぞれ .0191, .0069 となる．下流側の数値が小さいのは，下流境界がステップの十分下流ではなく，まだ圧力回復効果が作用しているところにあるからである．すなわち出口の速度分布には流路拡大の効果が残っており，最大流速は 1.152 でその最終値 1.125 に向けてなお減速中である．

13.3 陰解法化と Neumann 安定解析

MAC 法，SMAC 法などの陽解法 (explicit method) ではクーラン数 $C = \max(|u|/\Delta x, |v|/\Delta y)\Delta t$ の制約が厳しく，特に高レイノルズ数流れではこの制約は一層厳しいものになる．つまり時間間隔 Δt を大きく取ることができないので多くの反復計算を要することになる．これに対し陰解法 (implicit method) では，上流化，線形化，因子化を適切に行えば，クーラン数をかなり大きく取ることができる．SMAC 法でも，式 (13.7a)

を陰解法のものに書き換えればクーラン数を大きく取ることができる．ここではまず Navier-Stokes 方程式と同様の性質を持つスカラー量 ϕ の輸送方程式

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \mathbf{f} + d = 0 \quad (13.14)$$

の陰解法化について述べる．

式 (13.14) に梯形則 (trapezoidal law) を適用すれば次式が得られる．

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + (1-\theta)(\nabla \cdot \mathbf{f} + d)^n + \theta(\nabla \cdot \mathbf{f} + d)^{n+1} = 0 \quad (13.15)$$

この式は $\theta = 0$ ならば陽解法， $\theta = 1/2$ ならば Crank-Nicholson 法， $\theta = 1$ ならば完全陰解法になる．また $\theta = 1/2$ のときにのみ時間に関して 2 次精度でほかの θ では 1 次精度である³．式 (13.15) は， $\mathbf{f} = \mathbf{u}\phi$ と置くことができれば線形化 (linearization) することができ，更に $\phi^{n+1} = \phi^n + \Delta\phi^n$ と置けば次の Δ 形陰解法 (delta-form implicit method) の式に書き換えることができる．

$$(1 + \Delta t \theta \nabla \cdot \mathbf{u}) \Delta \phi^n = -\Delta t (\nabla \cdot \mathbf{f} + d)^n \quad (13.16)$$

ここで左辺の演算子 $\nabla \cdot$ は \mathbf{u} だけでなく $\Delta\phi^n$ にも作用していることに注意すべきである．なおここでは d は時間によらないものとし $d^{n+1} = d^n$ とした．定常流れを時間進行法で求める場合には，この式の右辺の括弧の中は解くべき方程式 (13.14) の残差，また $\Delta\phi^n$ は修正値を表す．反復計算で解が収束すればこの残差はゼロに近付き修正値もゼロに近づく．この式の右辺は解の精度に直接関係するので精度良く離散化しなければならないが，左辺の演算子は，解の精度に無関係で，安定化と計算量の軽減のために大胆に近似することが許される．通常左辺には，安定で簡単な 1 次上流差分が用いられ，拡散項 d は主要部のみが考慮され，また近似因子法が適用され 1 次元問題に分解して解かれる．

非定常流れの場合には，時間微分については，2 次精度の Crank-Nicholson 法や後退差分法 (Gear 法)，更には各種の Runge-Kutta 法，Adams-Bashforth 法などが用いられる．また空間微分については，右辺だけでなく左辺の演算子も解の精度に直接影響するので，2 次精度以上のものを用いなければならない．それゆえ式 (13.16) を直接解くことはあまり得策ではない．ここではまず 2 次の Crank-Nicholson 法を基に，各時間ステップごとに Newton 法で反復計算する方法について述べる．式 (13.14) に Crank-Nicholson 法を適用すれば，

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + \frac{1}{2} \{ (\nabla \cdot \mathbf{f} + d)^n + (\nabla \cdot \mathbf{f} + d)^{n+1} \} = 0$$

この式を Newton 法で反復計算するために $\phi^{n+1} \approx \phi^{(m)} = \phi^{(m-1)} + \Delta\phi^{(m)}$ と置けば次式が得られる．

$$\left(1 + \frac{1}{2} \Delta t \nabla \cdot \mathbf{u}\right) \Delta \phi^{(m)} = -(\phi^{(m-1)} - \phi^n) - \frac{1}{2} \Delta t \{ (\nabla \cdot \mathbf{f} + d)^n + (\nabla \cdot \mathbf{f} + d)^{(m-1)} \} \quad (13.17)$$

ただし上添字 (m) は $(n+1)$ 時間ステップの第 m 近似値を表し， $\phi^{(0)} = \phi^n$ ，解が収束すれば $\phi^{n+1} = \phi^{(m)}$ となる．この式から得られる収束解の精度は左辺の演算子には無関係で，左辺は定常流れの場合と同様に大胆に近似することができる．

³ Taylor 展開 $\phi^{n+1} = \phi^n + \Delta t \phi_t^n + (1/2!) \Delta t^2 \phi_{tt}^n + (1/3!) \Delta t^3 \phi_{ttt}^n + \dots$ ， $\mathbf{f}^{n+1} = \mathbf{f}^n + \Delta t (\partial \mathbf{f} / \partial t)^n + (1/2!) \Delta t^2 (\partial^2 \mathbf{f} / \partial t^2)^n + \dots$ を式 (13.15) に代入すれば，次式が得られる．

$$(\phi_t + \nabla \cdot \mathbf{f} + d)^n + \frac{1}{2!} \Delta t \frac{\partial}{\partial t} [\phi_t + 2\theta (\nabla \cdot \mathbf{f} + d)]^n + \frac{1}{3!} \Delta t^2 \frac{\partial^2}{\partial t^2} [\phi_t + 3\theta (\nabla \cdot \mathbf{f} + d)]^n + \dots = 0$$

この式に式 (13.14) の関係を用いれば， $\theta = 1/2$ のときにのみ 2 次精度であることが分かる．

次に式 (13.14) に 2 次後退差分を用いれば,

$$\frac{\phi^{n-1} - 4\phi^n + 3\phi^{n+1}}{2\Delta t} + (\nabla \cdot \mathbf{f} + d)^{n+1} = 0$$

上と同様に各時間ステップごとに Newton 法で反復計算することにすれば, その計算式は次のようになる.

$$\left(1 + \frac{2}{3}\Delta t \nabla \cdot \mathbf{u}\right) \Delta \phi^{(m)} = -\frac{1}{3}(\phi^{n-1} - 4\phi^n + 3\phi^{(m-1)}) - \frac{2}{3}\Delta t (\nabla \cdot \mathbf{f} + d)^{(m-1)} \quad (13.18)$$

この式から得られる収束解の精度も左辺の演算子には無関係で, 左辺は定常流れの場合と同様に大胆に近似することができる.

ここでスカラー輸送方程式 (13.14) の解の性質について簡単に述べる. この方程式は線形化すれば

$$\left(\frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla\right) \phi + d = 0 \quad (13.19)$$

となる. \mathbf{u} は通常流速で, 演算子 $\partial/\partial t + \mathbf{u} \cdot \nabla$ は時空間 xt における流跡線 $dx/dt = \mathbf{u}$ に沿う内微分 (interior derivative) を表している. 式 (13.14) は初期値問題として解かれるが, 式 (13.19) を 1 つの流跡線に沿って時間 t に関して積分すれば

$$\phi(\mathbf{x}, t) = \phi(\mathbf{x}_0, t_0) - \int_{t_0}^t d dt \quad (13.20)$$

となる. ただしこの式の積分路は流跡線に沿って取るものとする. d は一般に拡散項や発生項を意味するが, 輸送方程式が同次で $d = 0$ の場合には, 1 つの流跡線に沿って $\phi(t) = \phi(t_0)$ となり, 式 (13.19) の解は流跡線に沿って増減することなく速度 \mathbf{u} で伝播する波になる. 特に定常の場合には各流跡線に沿って ϕ の値は一定になる. 今この波を局所的に平面波⁴で近似すれば次のように表される.

$$\phi = f(\boldsymbol{\nu} \cdot \mathbf{x} - vt) \quad (13.21)$$

ただし f は波形, $\boldsymbol{\nu} \cdot \mathbf{x} - vt$ は位相 (phase) と呼ばれ, また $\boldsymbol{\nu}$ は波面の単位法線ベクトル, $v = \boldsymbol{\nu} \cdot \mathbf{u}$ は波面の法線速度である. 式 (13.21) は $d = 0$ の同次方程式 (13.14) を満足する. なお一般に, 微分方程式が同次の場合にはこのように基本解が任意関数で与えられる⁵. 輸送方程式が拡散項を含み同次でない $d \neq 0$ の場合には, 解 $\phi(\mathbf{x}, t)$ は, 式 (13.20) から明らかなように, 一つの流跡線に沿って速度 \mathbf{u} で伝播する間に d の値により増減する波になる. しかしその基本的性質は同じといえることができる.

以上見てきた輸送方程式の初期値問題は, 数学的には適正 (proper) で安定に解くことができる. しかしながらこれを離散近似した差分方程式の初期値問題は必ずしも安定に解くことはできず, その差分近似に際しては特別の注意が必要である. ここでは 1 次元の輸送方程式

$$\frac{\partial \phi}{\partial t} + \frac{\partial f}{\partial x} = \left(\frac{\partial}{\partial t} + u \frac{\partial}{\partial x}\right) \phi = 0 \quad (13.22)$$

について考える. この方程式は, xt 空間内に 1 パラメータ族の流跡線 $dx/dt = u$ を持ち⁶, 各流跡線に沿って解 ϕ の値が一定になることを示している. ここで一つの例を挙げれば, ϕ は塩水濃度, 式 (13.22) は管内の塩水の流れを表す式と見ることができる. 管の断面積は変化してもよいが流れは各断面内で一様とする.

⁴ 数学的表現である. 一般に n 次元空間において面は $n-1$ 次元, 線は 1 次元である. なお 2 次元から $n-2$ 次元の部分空間は多様体 (manifold) と呼ばれる.

⁵ 各項の階数が同じ方程式を同次方程式 (homogeneous equation) という. 例えば Laplace 方程式は 2 階の同次方程式である.

⁶ 一般に $F(x, t, \alpha) = 0$ は xt 空間内の 1 つの曲線を表すが, パラメータ α を変化させれば次々に曲線を発生させることができる. このようにして作られる曲線の集合を 1 パラメータ曲線族 (one-parameter family of curves) という. $F(x, t, \alpha, \beta) = 0$ は 2 パラメータ曲線族の式である.

上流端から流入する塩水の濃度が変化すれば濃度分布は非定常になるが，この流れでは生成や拡散がないので，各流体粒子（素片）の濃度 ϕ は速度 u で輸送される間一定に保たれる．

始めに離散近似した方程式の安定性を判別する von Neumann の方法 (von Neumann stability analysis) について述べる．この安定解析では，解 $\phi(x, t)$ の Fourier 成分を

$$\phi_i^n = r^n \exp(i \kappa i \Delta x) \quad (13.23)$$

のように置く．ただし $\phi_i^n = \phi(x_i, t^n)$, $x_i = i \Delta x$, $t^n = n \Delta t$, $i = \sqrt{-1}$ である．この波は波数 κ で，振幅関数 r の大きさが 1 より小ならば減衰し 1 より大ならば増幅する．差分方程式に式 (13.23) を代入すれば， r と $C \equiv u \Delta t / \Delta x$ の関係式が得られる． C は安定性にかかわる重要なパラメータでクーラン数 (Courant number) と呼ばれる．差分方程式が安定であるためには，すべての波数 κ に対し $|r| < 1$ であることが必要で，そのための条件は r と C の関係式から導出される．

ここで一例として，同次の式 (13.22) を Euler 前進法と 1 次上流差分で近似するとき得られる差分方程式

$$\phi_i^{n+1} - \phi_i^n + C^+ (\phi_i^n - \phi_{i-1}^n) + C^- (\phi_{i+1}^n - \phi_i^n) = 0 \quad (13.24)$$

の安定条件を導く．ただし $C^\pm = (C \pm |C|) / 2$ ．この式に式 (13.23) を代入すれば r と C の関係式

$$r - 1 + |C| (1 - \cos \kappa \Delta x) + i C \sin \kappa \Delta x = 0$$

が得られる．この式にその共役複素数を掛け実数化すれば，次の安定解の得られるための条件式が導かれる．

$$r^2 = 1 + 2(C^2 - |C|)(1 - \cos \kappa \Delta x) \leq 1$$

これより， $1 - \cos \kappa \Delta x \geq 0$ であるから，次の Neumann の安定条件が得られる．

$$|C| = |u| \Delta t / \Delta x \leq 1 \quad (13.25)$$

この条件は点 x_i, t^{n+1} を通る流跡線が時間 $t = t^n$ に区間 $[x_{i-1}, x_{i+1}]$ を通らなければならないことを示している．時間間隔 Δt はこの条件を満足する範囲に取られるべきで，この条件が満足されないときには計算が不安定になる虞がある．

次に 1 次元スカラー輸送方程式に梯形則を適用し離散近似したもう少し一般的な差分方程式

$$\phi_i^{n+1} + \lambda \theta (\hat{f}_{i+1/2}^{n+1} - \hat{f}_{i-1/2}^{n+1}) = \phi_i^n - \lambda (1 - \theta) (\hat{f}_{i+1/2}^n - \hat{f}_{i-1/2}^n) \quad (13.26)$$

を取り上げる．ただし $\lambda = \Delta x / \Delta t$ ，また \hat{f} は数値流束 (numerical flux) と呼ばれるもので，上の 1 次上流差分の場合には対称性を持たせれば

$$\hat{f}_{i+1/2} = \frac{1}{2} (f_i + f_{i+1}) - \frac{1}{2} |u_{i+1/2}| \Delta \phi_{i+1/2} \quad (13.27)$$

となる．ただし $\Delta \phi_{i+1/2} = \phi_{i+1} - \phi_i$ ．これは良く知られた Roe スキーム⁷ の数値流束関数である．1 次上流差分の場合の式 (13.26) に式 (13.23) を代入すれば， r と C の関係式

$$r [1 + \theta \{ i C \sin \kappa \Delta x + |C| (1 - \cos \kappa \Delta x) \}] = 1 - (1 - \theta) \{ i C \sin \kappa \Delta x + |C| (1 - \cos \kappa \Delta x) \}$$

⁷Roe, P.L., Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.*, Vol.43, 357-372.

が得られる．この式は

$$r^2 = 1 - \frac{2|C|\{1 - (1 - 2\theta)|C|\}(1 - \cos \kappa \Delta x)}{1 + 2\theta|C|(1 + \theta|C|)(1 - \cos \kappa \Delta x)} \leq 1$$

のように書き換えることができ，右辺第 2 項の分母は常に正であるから，その分子も正でなければならないことになる．これより次の安定条件が導かれる．

$$(1 - 2\theta)|C| \leq 1 \quad (13.28)$$

この条件から，1 次上流差分の場合には， $\theta = 1$ の完全陰解法は Courant 数 $|C|$ によらず安定で，また $\theta = 1/2$ の Crank-Nicholson 法は中立安定で Courant 数をかなり大きく取ることができることになる．

次に 2 次中心差分の場合には数値流束関数は

$$\hat{f}_{i+1/2} = u_{i+1/2}(\phi_i + \phi_{i+1})/2 \quad (13.29)$$

となる．この場合には， r と C の関係式は

$$r(1 + i\theta C \sin \kappa \Delta x) = 1 - i(1 - \theta)C \sin \kappa \Delta x$$

となり，この式は

$$r^2 = 1 - \frac{(2\theta - 1)C^2 \sin^2 \kappa \Delta x}{1 + \theta^2 C^2 \sin^2 \kappa \Delta x} \leq 1$$

のように書き換えられ，これより次の安定条件が導かれる．

$$\theta \geq 1/2 \quad (13.30)$$

この条件から 2 次中心差分の場合には，Courant 数 $|C|$ によらず $\theta = 0$ のオイラー前進法では不安定， $\theta = 1/2$ の Crank-Nicholson 法では中立安定， $\theta = 1$ の完全陰解法では安定である．しかしながら，前述の計算例が示すように，Neumann 安定解析では不安定であるにもかかわらず，高くないレイノルズ数 70，最大 Courant 数 0.24 では安定に解を求めることができた．また後述の例が示すように，Neumann 安定解析では無条件安定であっても実際の計算で不安定になることはよくあることである．

13.4 TVD 安定性と TVD スキーム

TVD 安定性と TVD スキームに関しては，第 8 章と第 16 章に詳しい説明があるので，ここにはその概要を述べるにとどめる．前節に述べたように，非定常 1 次元非圧縮性流れの輸送方程式 (13.22) は， xt 面上に 1 パラメータ族の流跡線を持ち，解 $\phi(x, t)$ は各流跡線に沿って一定値を取る．総変化量 (total variation) TV はこの $\phi(x, t)$ に対して次のように定義される．

$$TV(t) \equiv \int |\phi_x(x, t)| dx \quad (13.31)$$

ただし積分区間は任意の 1 つの流跡線から他の流跡線まで取るものとする．

式 (13.31) の TV は例えば右図の $\phi(t)$ に対しては次のようになる．

$$\begin{aligned} TV &= \int_{x_a}^{x_b} |\phi_x| dx \\ &= (\phi_b - \phi_2) + (\phi_1 - \phi_{min}) + |\phi_{min} - \phi_a| \end{aligned}$$

TV は一般に時間 t の関数であるがこの問題では時間によらず一定になる。 TV はもともと非定常 1 次元圧縮性流れのオイラー方程式に関して定義されたものである。この方程式の場合には、流跡線のほかに 2 つの 1 パラメータ族の特性曲線が存在し、各特性曲線族に対する TV は時間によらず一定になる。しかし積分区間に衝撃波がある場合には時間と共に減少する。すなわち TV は増加することではなく TVD (total variation diminishing) である。 TV の定義式 (13.31) は差分で表せば

$$TV(\phi^n) \equiv \sum_i |\phi_{i+1}^n - \phi_i^n| \quad (13.32)$$

また TVD 条件は

$$TV(\phi^{n+1}) \leq TV(\phi^n) \quad (13.33)$$

となる。TVD スキームは常に TVD 条件 (13.33) を満足する解 ϕ の得られる数値スキームのことである。TVD 条件が満足されるときには、当然、(i) ϕ の新たな極値は発生しない、(ii) ϕ の最小値が減少、最大値が増加することもない。したがって TVD スキームを用いれば解を安定に求めることができる。

始めに 1 次上流差分の TVD 安定性について述べる。1 次元のスカラー輸送方程式 (13.26) に、1 次上流差分の数値流束関数 (13.27) を代入し、その両辺の TV を取れば、3 条件

$$C^+ \geq 0, \quad C^- \leq 0; \quad (1-\theta)|C| \leq 1 \quad (13.34)$$

のもとで、次の関係が成立する。

$$TV(\phi^{n+1}) \leq TV(LHS) = TV(RHS) \leq TV(\phi^n)$$

これより 1 次上流差分は、式 (13.34) のもとで、TVD 条件 (13.33) を満足し、TVD スキームであることが分かる。この場合の TVD 安定条件 (13.34) は、はじめの 2 つは常に満足され、 $(1-\theta)|C| \leq 1$ にのみ留意すれば良いことになる。例えば $\theta = 1/2$ の Crank-Nicholson 法の場合には、Neumann 安定条件ではクーラン数 $|C|$ を自由に選んでも良いということであったが、TVD 安定条件では $|C| \leq 2$ となる。

2 次以上の差分スキームは、Courant 数 $|C|$ を適切に選ぶだけでは不十分で不安定になることもある。以下には 2 次以上のスキームにおける解の不安定性と、それを TVD 安定化する対策について述べる。Chakravarthy-Osher 型スキームは 2 次上流差分と 2 次中心差分を線形結合したもので、その数値流束関数は次のように与えられる⁸。

$$\begin{aligned} \hat{f}_{i+1/2} = & \hat{f}_{i+1/2}^{(1)} + \frac{1}{4}(1-\kappa)\Delta f_{i-1/2}^+ + \frac{1}{4}(1+\kappa)\Delta f_{i+1/2}^+ \\ & - \frac{1}{4}(1-\kappa)\Delta f_{i+3/2}^- - \frac{1}{4}(1+\kappa)\Delta f_{i+1/2}^- \end{aligned} \quad (13.35)$$

この式の右辺第 1 項は 1 次上流差分の数値流束で、第 2 項以降はこれを 2 次にする補正項である。また κ は結合のパラメータで、 $\kappa = -1$ ならば 2 次上流差分、 $\kappa = 1$ ならば 2 次中心差分、 $\kappa = 1/2$ ならば Leonard の QUICK スキーム⁹ の数値流束になる。QUICK スキームの数値流束は上流側にシフトした 3 点を通る 2 次多項式、すなわち $u_{i+1/2} \geq 0$ ならば点 f_{i-1} , f_i , f_{i+1} 、また $u_{i+1/2} < 0$ ならば点 f_i , f_{i+1} , f_{i+2} を通る 2 次 Lagrange 補間多項式の $f(x_{i+1/2})$ である。式 (13.35) の数値流束を用いた差分式の精度は Taylor 展開をもとに評価すれば次のようになる。

$$\frac{1}{\Delta x}(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}) = \left(\frac{\partial f}{\partial x}\right)_i + \frac{1}{12}(3\kappa - 1)\Delta x^2 \left(\frac{\partial^3 f}{\partial x^3}\right)_i + O(\Delta x^3) \quad (13.36)$$

⁸Chakravarthy, S.R. and Osher, S., A new class of high accuracy TVD schemes for hyperbolic conservation laws. *AIAA Paper* 85-0363, 1985.

⁹Leonard, B.P., A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Meth. Appl. Meth. Engng.*, Vol.19(1979),59-98.

最後の 2 項は打ち切り誤差で、これから $\kappa = 1/3$ に取ればスキームの精度が 3 次になることが分かる。また κ が $1/3$ に近い QUICK スキームは 2 次ではあるが、2 次中心差分や 2 次上流差分に比べ精度の高いものであることも分かる。

次にこれらのスキームの TVD 安定化について述べる。式 (13.35) の数値流束は 1 次上流差分の数値流束とこれを 2 次にする補正項からなる。1 次上流差分の部分は TVD 安定であるから、補正項の部分をこれらのスキームが TVD 不安定にならないようにその大きさを制限すれば、スキームは TVD になるはずである。そのために各種の制限関数 (limiter) が用いられてきた。Chakravarthy-Osher TVD スキームでは minmod 制限関数が用いられその数値流束は次のように書かれる。

$$\begin{aligned} \hat{f}_{i+1/2} = & \hat{f}_{i+1/2}^{(1)} + \frac{1}{4}(1-\kappa)\Delta\tilde{f}_{i-1/2}^+ + \frac{1}{4}(1+\kappa)\Delta\tilde{f}_{i+1/2}^+ \\ & - \frac{1}{4}(1-\kappa)\Delta\tilde{f}_{i+3/2}^- - \frac{1}{4}(1+\kappa)\Delta\tilde{f}_{i+1/2}^- \end{aligned} \quad (13.37)$$

ただし

$$\hat{f}_{i+1/2}^{(1)} = \frac{1}{2}(f_i + f_{i+1} - |\Delta f_{i+1/2}|), \quad (13.38a)$$

$$\tilde{f}_{j+1/2}^\pm = \text{minmod}(\Delta f_{j+1/2}^\pm, b\Delta f_{j-1/2}^\pm), \quad (j = i, i+1) \quad (13.38b)$$

$$\tilde{f}_{j+1/2}^\pm = \text{minmod}(\Delta f_{j+1/2}^\pm, b\Delta f_{j+3/2}^\pm), \quad (j = i-1, i) \quad (13.38c)$$

$$\text{minmod}(x, y) = \text{sign}(x) \max[0, \min\{|x|, \text{sign}(x)y\}] \quad (13.38c)$$

また $f_j^\pm = u_{i+1/2}^\pm \phi_j$, ($j = i-1, i, i+1, i+2$), $|\Delta f_{i+1/2}| = |u_{i+1/2}| |\Delta \phi_{i+1/2}|$ である。

ここで式 (13.37) の数値流束の意味を考える。まず制限関数の働かない場合にはこの式は

$$\begin{aligned} \hat{f}_{i+1/2} = & f_i^+ + \frac{1}{4}(1-\kappa)(f_i^+ - f_{i-1}^+) + \frac{1}{4}(1+\kappa)(f_{i+1}^+ - f_i^+) \\ & + f_{i+1}^- - \frac{1}{4}(1-\kappa)(f_{i+2}^- - f_{i+1}^-) - \frac{1}{4}(1+\kappa)(f_{i+1}^- - f_i^-) \end{aligned} \quad (13.39)$$

ただし $f_j^\pm = u_j^\pm \phi_j$, ($j = i-1, i, i+1, i+2$) となる。すなわち一般には、 $\hat{f}_{i+1/2}$ の値は、 u の正負により上下いずれかの式によって、上流側にシフトした 3 点 f_{i-1}, f_i, f_{i+1} または f_i, f_{i+1}, f_{i+2} の値から計算される。その値は u と κ により、図 13.3 に示す範囲内にある。

しかしながら u の符号が $u < 0$, ($x < x_i + \alpha \Delta x$), $u \geq 0$, ($x \geq x_i + \alpha \Delta x$) のように変わる音速点では¹⁰,

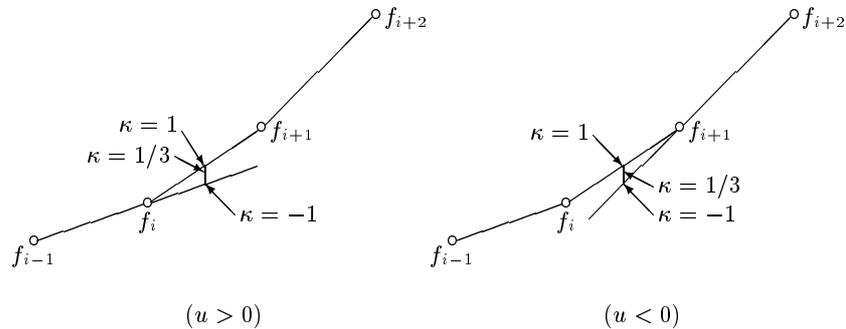


図 13.3: $\hat{f}_{i+1/2}$ の値の範囲

¹⁰ 音速点で流速 u の符号が変わるのではなく、圧縮性 1 次元流れには 3 つの特性曲線族 $dx/dt = u, u \pm c$ が存在し、その 1 つ圧力波の $u-c$ の符号が変わるということである。また一般に多次元流れでは、流速の横成分 v, w の符号が変わるところで同様のことが起きる。

上式は次のようになる．

$$\hat{f}_{i+1/2} = \begin{cases} f_i + (1-\kappa)f_i/4 + (1+\kappa)(f_{i+1}-f_i)/4 & (-1 < \alpha < 0) \\ (1+\kappa)(f_{i+1}+f_i)/4 & (0 \leq \alpha < 1) \\ f_{i+1} + (1-\kappa)f_{i+1}/4 - (1+\kappa)(f_{i+1}-f_i)/4 & (1 \leq \alpha < 2) \end{cases}$$

これらの式は $\kappa = 1$ の場合には 2 次中心差分のままであるが， κ の値が 1 よりも小さくなると近似度は次第に低下する．例えば $x_{i-1} < x < x_i$ に音速点があり $\kappa = 1/3$ の場合には，上式より 1 階微分の差分式は

$$(f_x)_i = \frac{1}{\Delta x}(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}) = \frac{1}{\Delta x} \left(-\frac{1}{3}f_{i-1} + \frac{1}{2}f_i + \frac{1}{3}f_{i+1} \right)$$

となり，誤差評価するまでもなく破綻していることが分かる．念のため Taylor 展開をもとに誤差評価すれば，この差分式は $2(f_x)_i/3 + f_i/(2\Delta x) + \dots$ となり，誤差の大きさは $O(\Delta x^{-1})$ である．2 次以上の高精度差分式を用いた計算結果が音速点の近傍で不自然に波打つことは良く知られている．

同様に u の符号が $u \geq 0$, ($x < x_i + \alpha\Delta x$), $u < 0$, ($x \geq x_i + \alpha\Delta x$) のように変わるところでも

$$\hat{f}_{i+1/2} = \begin{cases} f_{i+1} - (1-\kappa)(f_{i+2}-f_{i+1}+f_{i-1})/4 - (1+\kappa)(f_{i+1}-f_i)/4 & (-1 < \alpha < 0) \\ (1-\kappa)(f_i-f_{i-1}-f_{i+2}+f_{i+1})/4 + (3-\kappa)(f_i+f_{i+1})/4 & (0 \leq \alpha < 1) \\ f_i + (1-\kappa)(f_i-f_{i-1}-f_{i+2})/4 + (1+\kappa)(f_{i+1}-f_i)/4 & (1 \leq \alpha < 2) \end{cases}$$

となり差分式は破綻する．これを回避するために各種の対策が立てられてきた．例えば流束ベクトル分離法では，移行区間を設け，数値流束 \hat{f}^- 100% の状態から， \hat{f}^- の割合を徐々に減らし数値流束 \hat{f}^+ の割合を徐々に増やし， \hat{f}^+ 100% の状態に移行させるというものがある．この種の解法では不自然な波打ちは完全に回避できるが，その数値誤差は一般に小さくない．

Chakravarthy-Osher スキームの流束の式 (13.39) は，その右辺第 1 行または第 2 行それぞれが一種の補間式になっている． u の符号が変わるところでこれらの補間式が適当にミックスされたものは，上記のように補間機能を果たさない．補間機能を保つためには，2 つの補間式のいずれか一方を選べば良く， $u_{i+1/2}$ の符号によって選べば次式が得られる．

$$\begin{aligned} \hat{f}_{i+1/2} = \text{sign}(u_{i+1/2}^+) & \left\{ f_i + \frac{1}{4}(1-\kappa)(f_i-f_{i-1}) + \frac{1}{4}(1+\kappa)(f_{i+1}-f_i) \right\} \\ & - \text{sign}(u_{i+1/2}^-) \left\{ f_{i+1} - \frac{1}{4}(1-\kappa)(f_{i+2}-f_{i+1}) - \frac{1}{4}(1+\kappa)(f_{i+1}-f_i) \right\} \end{aligned} \quad (13.40)$$

u の符号が変わるところでこの式の精度を調べる． $u_{i-1/2} < 0$, $u_{i+1/2} > 0$ の場合には，この流束を用いた 1 階微分の差分式は $(f_{i+1}-f_{i-1})/(2\Delta x) = (f_x)_i + O(\Delta x^2)$ となる．また $u_{i-1/2} > 0$, $u_{i+1/2} < 0$ の場合には，1 階微分の差分式は $(2-\kappa)(f_{i+1}-f_{i-1})/2\Delta x - (1-\kappa)(f_{i+2}-f_{i-2})/4\Delta x = (f_x)_i + O(\Delta x^2)$ となる．いずれの場合にも κ によらず 2 次精度が保証される．式 (13.40) は，式 (13.39) で $f_j^\pm = |\text{sign}(u_{i+1/2}^\pm)|f_j$, ($j = i-1, i, i+1, i+2$) と置いたものと同じである．式 (13.39) で $f_j^\pm = u_{i+1/2}^\pm \phi_j$, ($j = i-1, i, i+1, i+2$) と置けば，もとの式と全く同じではないが， u の符号が変わるところでも κ によらず 2 次精度の式が得られる．この Chakravarthy-Osher スキームの流束の式は $\kappa = 1/3$ のときにも 2 次精度であるが，実際の計算の結果は 3 次精度の式を用いたものと違わない．それは u が準線形方程式 (13.22) の最高階数の微分ではなくその係数であることを考えれば当然といえよう．

次に Chakravarthy-Osher TVD スキームの流束の式 (13.37) における制限関数の働きについて考える．minmod 制限関数は式 (13.38c) で定義され，プログラムもこの式をもとに作られるが，分かり易く書き直せば次のようになる．

$$\text{minmod}(x, y) = \begin{cases} x & (x, y \text{ 同符号}, |x| \leq |y|) \\ y & (x, y \text{ 同符号}, |x| > |y|) \\ 0 & (x, y \text{ 異符号}) \end{cases}$$

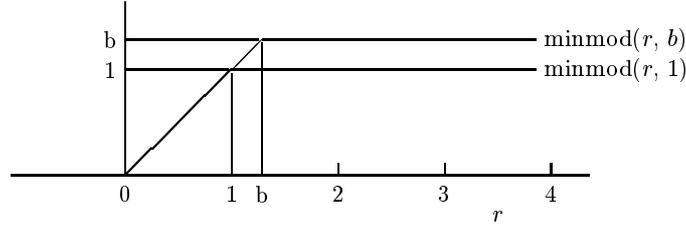


図 13.4: minmod 制限関数

また $y/x = r$ と置けば

$$\min\text{mod}(r, 1) = \begin{cases} 1 & (r \geq 1) \\ r & (0 < r < 1) \\ 0 & (r \leq 0) \end{cases}$$

のようになる．この $\min\text{mod}(r, 1)$ の値は図 13.4 に示すようになる．これより式 (13.38b) の意味するものは，当該勾配 $\Delta f_{j+1/2}$ の大きさが隣の勾配の大きさを b 倍したものよりも小さいときにはそのまま，大きいときには $b\Delta f_{j-1/2}$ に置き換え，また隣の勾配の傾きが逆のときには 0 に置くというものである．このように勾配の大きさを制限するものを勾配制限関数 (slope limiter) という． b は 1 よりも大きい正数であるが，その値を大きく取れば制限関数が働き解の精度の低下する範囲は当然狭くなる．

次に 1 次元のスカラー輸送方程式 (13.26) に，Chakravarthy-Osher TVD スキームの数値流束関数 (13.37) を代入した式から求められる TVD 安定条件を結果のみ示す．

$$\tilde{C}^+ \geq 0, \quad \tilde{C}^- \leq 0; \quad (1-\theta)|\tilde{C}| \leq 1 \quad (13.41)$$

ただし

$$\tilde{C}^\pm = \tilde{u}^\pm \Delta t / \Delta x, \quad (13.42a)$$

$$|\tilde{C}| = \tilde{C}^+ - \tilde{C}^-, \quad (13.42b)$$

$$\begin{aligned} \tilde{u}_{i\mp 1/2}^\pm \equiv u_{i\mp 1/2}^\pm \left[1 + \frac{1-\kappa}{4} \left\{ \frac{u_{i\pm 1/2}^\pm}{u_{i\mp 1/2}^\pm} \min\text{mod}(1, br_{i\pm 1/2}^\mp) - \min\text{mod}(r_{i\mp 3/2}^\pm, b) \right\} \right. \\ \left. + \frac{1+\kappa}{4} \left\{ \frac{u_{i\pm 1/2}^\pm}{u_{i\mp 1/2}^\pm} \min\text{mod}(r_{i\pm 1/2}^\mp, b) - \min\text{mod}(1, br_{i\mp 3/2}^\pm) \right\} \right] \end{aligned} \quad (13.42c)$$

また $r_{j+1/2}^\pm = \Delta\phi_{j+1/2} / \Delta\phi_{j+1/2\pm 1}$ である．これらの 3 条件の第 1 式と第 2 式より次の条件が導かれる．

$$1 \leq b \leq (3-\kappa)/(1-\kappa) \quad (13.43)$$

また第 3 式より次の条件が導かれる．

$$(1-\theta)b_1|C| \leq 1, \quad b_1 = \frac{1}{4}\{5-\kappa+(1+\kappa)b\} \quad (13.44)$$

ただし $|C| = (\Delta t / \Delta x) \max(|u_{i-1/2}|, |u_{i+1/2}|)$ である．Chakravarthy-Osher TVD スキームが文字通りの TVD スキームであるためには，式 (13.38b) の b が式 (13.43) を満足し，また Courant 数 C すなわち時間間隔 Δt が式 (13.44) を満足するように選ばなければならない．

広く使われている 3 次上流差分 ($\kappa = 1/3$) の場合には $b \leq 4$ になる．その上限 $b = 4$ に取れば，陽解法 ($\theta = 0$) では $|C| \leq 0.4$ ，Crank-Nicholson 法 ($\theta = 1/2$) では $|C| \leq 0.8$ になり，完全陰解法 ($\theta = 1$) では $|C|$ の上限はない．また Crank-Nicholson 法で $|C| = 1$ に取れば， $b \leq 2.5$ になる．3 次上流差分の場合に， $b = 4$ に取れば，式 (13.37) の数値流束は次のようになる．

$$\hat{f}_{i+1/2} = \begin{cases} f_i + \frac{1}{6} \min\text{mod}(\Delta f_{i-1/2}, 4\Delta f_{i+1/2}) + \frac{1}{3} \min\text{mod}(\Delta f_{i+1/2}, 4\Delta f_{i-1/2}) & (u \geq 0) \\ f_{i+1} - \frac{1}{6} \min\text{mod}(\Delta f_{i+3/2}, 4\Delta f_{i+1/2}) - \frac{1}{3} \min\text{mod}(\Delta f_{i+1/2}, 4\Delta f_{i+3/2}) & (u < 0) \end{cases} \quad (13.45)$$

次にこの式の制限関数の働きを図を使って説明する．ここでは $u > 0$ ， $\Delta f > 0$ の場合を取上げるがほかの場合も制限関数の働きは同じである．この場合には式 (13.45) の第 1 式は次のようになる．

$$\hat{f}_{i+1/2} = f_i + \left\{ \frac{1}{6} \min\text{mod}(r, 4) + \frac{1}{3} \min\text{mod}(1, 4r) \right\} \Delta f_{i+1/2} \quad (r = \Delta f_{i-1/2} / \Delta f_{i+1/2}) \quad (13.46a)$$

$$= f_i + \left\{ \frac{1}{6} \min\text{mod}(1, 4r) + \frac{1}{3} \min\text{mod}(r, 4) \right\} \Delta f_{i-1/2} \quad (r = \Delta f_{i+1/2} / \Delta f_{i-1/2}) \quad (13.46b)$$

図 13.5 の左図は， f_i, f_{i+1} を固定し f_{i-1} を変えた場合の $\hat{f}_{i+1/2}$ の値を上式から計算したものである． $r > 4$ では第 1 の $\min\text{mod}$ 関数が働き， $\hat{f}_{i+1/2}$ の値が f_{i+1} の値を超えないよう制限される．同様に右図は， f_{i-1}, f_i を固定し f_{i+1} を変えた場合の $\hat{f}_{i+1/2}$ の値を下式から計算したもので， $r < 1/4$ で第 1 の $\min\text{mod}$ 関数の働き， $\hat{f}_{i+1/2}$ の値が f_{i+1} の値を超えないように制限される．なおこれらの式の第 2 の $\min\text{mod}$ 関数はクーラン数の上限に関わるものである．

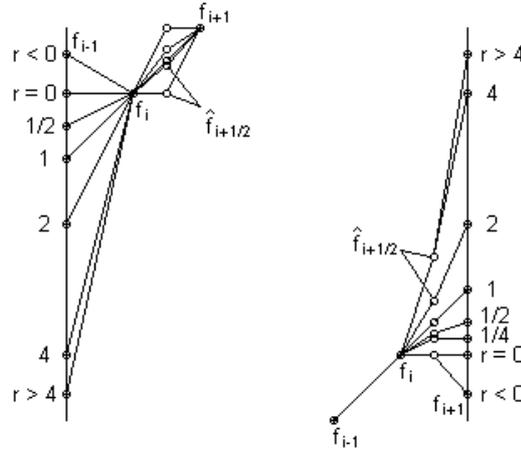


図 13.5: Chakravarthy-Osher TVD スキームの数値流束

ここで多少補足すれば「TVD スキームを用いた場合に最小値が減少，最大値が増加しない」というのは，1 次元の同次方程式 (13.22) の場合には輸送量 ϕ に対して，また 1 次元圧縮性流れの場合には 3 つの特性族の各特性に沿って伝播するエントロピーや 2 つのリーマン不変量に対して言っているのである．これらの不変量は，拡散項や発生項のある非同次の場合にも，拡散や発生の効果により増減し「最小値が減少，最大値が増加しない」ということにはならない．また多次元の場合には，拡散や発生項がなくても，流れの x 成分には y, z 成分が発生項の役割をするので同じことが言える．輸送方程式の実質微分の差分近似に TVD スキームを用いれば，拡散項や発生項の有無にかかわらず輸送量は良く保存され解は安定化する．TVD スキームを用いたために，本来最大値の増加する物理量が最大値が制限され増加しなくなるというようなことは有り得ないことである．

13.5 SMAC Δ形陰解法

SMAC Δ形陰解法は，SMAC 法の第 1 式 (13.7a) を式 (13.16) によって陰解法の式に書換えたもので，その全体の式は次のようになる．

$$\{1 + \Delta t \theta (\nabla \cdot \mathbf{u}^n - \nu \nabla^2)\} \Delta \mathbf{u}^* = rhs^n, \quad \mathbf{u}^* = \mathbf{u}^n + \Delta \mathbf{u}^* \quad (13.47a)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi \quad (13.47b)$$

$$\nabla^2 \phi^n = \nabla \cdot \mathbf{u}^* / \Delta t, \quad p^{n+1} = p^n + \phi \quad (13.47c)$$

ただし $rhs = -\Delta t (\nabla \cdot \mathbf{u} \mathbf{u} + \nabla p - \nu \nabla^2 \mathbf{u} - \mathbf{g})$ である．式 (13.47a) の左辺の演算子 $\nabla \cdot$ は $\Delta \mathbf{u}^*$ にも作用していることに注意すべきである． $\theta = 1/2$ ならば時間に関し 2 次精度の Crank-Nicholson 法， $\theta = 1$ ならば 1 次精度のオイラー陰解法になる．なお式 (13.47b) と (13.47c) は SMAC 法のものと同じで，またこれらの式は SMAC 法と同じ手順で解かれる．Δ形陰解法の利点については 13.3 節を参照されたい．

次に非定常流れの SMAC Δ形陰解法の式は式 (13.17) を用い次のように書かれる．

$$\{1 + \Delta t \theta (\nabla \cdot \mathbf{u}^n - \nu \nabla^2)\} \Delta \mathbf{u}^{*(m)} = -(\mathbf{u}^{(m-1)} - \mathbf{u}^n) + \frac{1}{2} (rhs^n + rhs^{(m-1)}),$$

$$\mathbf{u}^{*(m)} = \mathbf{u}^{(m-1)} + \Delta \mathbf{u}^{*(m)} \quad (13.48a)$$

$$\mathbf{u}^{(m)} = \mathbf{u}^{*(m)} - \frac{1}{2} \Delta t \nabla \phi^{(m)} \quad (13.48b)$$

$$\nabla^2 \phi^{(m)} = \frac{2}{\Delta t} \nabla \cdot \mathbf{u}^{*(m)}, \quad p^{(m)} = p^{(m-1)} + \phi^{(m)} \quad (13.48c)$$

この解法では非定常流れが各時間ステップごとに反復計算によって求められる．その予測値は $\mathbf{u}^{(0)} = \mathbf{u}^n$ ， $rhs^{(0)} = rhs^n$ ，または時間に関する外挿によって求められる．

SMAC Δ形陰解法の式 (13.47a) または (13.48a) は近似因子法を適用することによって効率よく解くことができる．近似因子 (approximate-factorization, AF) 法は，旧ソ連の Yanenko によって考案された多次元問題を 1 次元問題に帰着させて解く部分段階法のひとつで，Holt(1971) 監修の訳本によって西側に紹介された¹¹．一般に Δ形陰解法の式は，2 次元の場合に，対流項に 1 次上流差分，拡散項に 2 次中心差分を用いれば形式的に次のように書くことができる．

$$(1 + c_1^+ \nabla_x + c_1^- \Delta_x + c_2^+ \nabla_y + c_2^- \Delta_y) u = rhs \quad (13.49)$$

ただし c_1^+ ， c_1^- ， c_2^+ ， c_2^- は Courant 数に拡散係数を考慮したもの， ∇_x ， ∇_y は後退差分， Δ_x ， Δ_y は前進差分の演算子で，例えば $c_1^+ \nabla_x u = \theta (C^+ + \nu / \Delta x^2) (u_{00} - u_{-1,0})$ である．この式は差分式で表せば次のようになる．

$$c_0 u_{00} - c_1^+ u_{-1,0} + c_1^- u_{10} - c_2^+ u_{0,-1} + c_2^- u_{01} = rhs \quad (13.50)$$

ただし $c_0 = (1 + |c_1| + |c_2|)$ である．式 (13.49) は従来の近似因子法では，

$$(1 + c_1^+ \nabla_x + c_1^- \Delta_x) (1 + c_2^+ \nabla_y + c_2^- \Delta_y) u = rhs \quad (13.51)$$

のように因子化され，次式により 2 段階に分けて解かれる．

$$(1 + |c_1|) \bar{u}_{00} - c_1^+ \bar{u}_{-1,0} + c_1^- \bar{u}_{10} = rhs \quad (13.52a)$$

$$(1 + |c_2|) u_{00} - c_2^+ u_{0,-1} + c_2^- u_{01} = \bar{u}_{00} \quad (13.52b)$$

¹¹Yanenko, N.N., On the implicit difference computing methods for solving multidimensional heat conduction equations (Russian), *Izv. Uchebn. Zaved., Matematika*, 4(1963), 148-157. Yanenko, N.N., *The Method of Fractional Steps.*, 1971, Springer-Verlag.

この因子化に伴う誤差は，式 (13.52a) に式 (13.52b) を代入し因子化前の式 (13.50) と比較することによって次のように求められる．

$$\begin{aligned} & -c_2^+(-c_1^+u_{-1,-1} + |c_1|u_{0,-1} + c_1^-u_{1,-1}) + |c_2|(-c_1^+u_{-1,0} + |c_1|u_{00} + c_1^-u_{10}) \\ & + c_2^-(-c_1^+u_{-1,1} + |c_1|u_{01} + c_1^-u_{11}) \end{aligned}$$

この誤差は $c_1 \approx 0$ または $c_2 \approx 0$ または $u(x, y)$ が 1 次的に変化するところでは無視できるが，一般には Courant 数の 2 乗の大きさになることが分かる．このため時間間隔 Δt を大きく取ると，解の収束はかえって悪くなり，さらに発散する虞もでてくる．この因子化の誤差を小さく抑えるために，Jameson らは LU-SGS 法 (lower-upper symmetric-Gauss-Seidel method)¹² を提案している．この方法では式 (13.49) は

$$(1 + c_1^+ \nabla_x + c_2^+ \nabla_y) c_0^{-1} (1 + c_1^- \Delta_x + c_2^- \Delta_y) u = rhs$$

のように因子化され，次式により 2 段階に分けて解かれる．

$$c_0 \bar{u}_{00} - c_1^+ \bar{u}_{-1,0} - c_2^+ \bar{u}_{0,-1} = rhs \quad (13.53a)$$

$$c_0 u_{00} + c_1^- u_{10} + c_2^- u_{01} = c_0 \bar{u}_{00} \quad (13.53b)$$

この因子化の誤差は，式 (13.53a) に式 (13.53b) を代入し因子化前の式 (13.50) と比較することによって次のように求められる．

$$-c_0^{-1} \{ c_1^+ c_2^- u_{-1,1} + (c_1^+ c_1^- + c_2^+ c_2^-) u_{00} + c_2^+ c_1^- u_{1,-1} \}$$

この誤差はたかだか Courant 数の大きさであることが分かる．

式 (13.50) に対する近似因子法の式は $c_1, c_2 \geq 0$ とすれば，一般に次のように書くことができる．

$$d_1 \bar{u}_{00} - d_2 \bar{u}_{-1,0} = d_0 rhs, \quad d_3 u_{00} - d_4 u_{0,-1} = \bar{u}_{00}$$

これら 2 つの式を 1 つに纏めた式と もとの式 (13.50) を比較すれば，次の 4 つの条件が得られる．

$$d_1 d_3 = d_0 c_0, \quad d_2 d_3 = d_0 c_1, \quad d_1 d_4 = d_0 c_2, \quad d_2 d_4 = 0$$

d_i はこれらすべての条件を満足させることはできないが，平均的に妥当なものとして次のように決めることができる．

$$d_1 = d_3 = d_0 = c_0, \quad d_2 = c_1, \quad d_4 = c_2$$

これより， $c_1, c_2 < 0$ の場合も含めて，筆者らの考案した次の改良型近似因子法 (IAF, improved approximate-factorization) の式を導出できる．

$$c_0 \bar{u}_{00} - c_1^+ \bar{u}_{-1,0} + c_1^- \bar{u}_{10} = rhs \quad (13.54a)$$

$$c_0 u_{00} - c_2^+ u_{0,-1} + c_2^- u_{01} = c_0 \bar{u}_{00} \quad (13.54b)$$

この因子化に伴う誤差は

$$c_0^{-1} (c_1^+ c_2^+ u_{-1,-1} - c_1^- c_2^+ u_{1,-1} - c_1^+ c_2^- u_{-1,1} + c_1^- c_2^- u_{11})$$

¹² Jameson, A. and Yoon, S., Lower-upper implicit schemes with multiple grids for the Euler equations, *AIAA J.*, **25**(1987), 929-935. Yoon, S. and Kwak, D., Implicit Navier-Stokes solver for three-dimensional compressible flows, *AIAA J.*, **30**(1992), 2653-59.

で Courant 数の大きさになる .

3次元の場合には Δ形陰解法の式は

$$(1+c_1^+\nabla_x+c_1^-\Delta_x+c_2^+\nabla_y+c_2^-\Delta_y+c_3^+\nabla_z+c_3^-\Delta_z)u = rhs \quad (13.55)$$

となる . 従来の近似因子法の式は

$$(1+|c_1|)\bar{u}_{000}-c_1^+\bar{u}_{-1,0,0}+c_1^-\bar{u}_{100} = rhs \quad (13.56a)$$

$$(1+|c_2|)\bar{u}_{000}-c_2^+\bar{u}_{0,-1,0}+c_2^-\bar{u}_{010} = \bar{u}_{000} \quad (13.56b)$$

$$(1+|c_3|)u_{000}-c_3^+u_{0,0,-1}+c_3^-u_{001} = \bar{u}_{000} \quad (13.56c)$$

LU-SGS の式は

$$c_0\bar{u}_{000}-c_1^+\bar{u}_{-1,0,0}-c_2^+\bar{u}_{0,-1,0}-c_3^+\bar{u}_{0,0,-1} = rhs \quad (13.57a)$$

$$c_0u_{000}+c_1^-u_{100}+c_2^-u_{010}+c_3^-u_{001} = c_0\bar{u}_{000} \quad (13.57b)$$

また改良型近似因子法の式は

$$c_0\bar{u}_{000}-c_1^+\bar{u}_{-1,0,0}+c_1^-\bar{u}_{100} = rhs \quad (13.58a)$$

$$c_0\bar{u}_{000}-c_2^+\bar{u}_{0,-1,0}+c_2^-\bar{u}_{010} = c_0\bar{u}_{000} \quad (13.58b)$$

$$c_0u_{000}-c_3^+u_{0,0,-1}+c_3^-u_{001} = c_0\bar{u}_{000} \quad (13.58c)$$

ただし $c_0 = 1+|c_1|+|c_2|+|c_3|$ である . 近似因子法または改良型近似因子法の各式は 3重対角行列の連立 1次方程式で , Gauss 消去法で容易に解くことができる . ベクトルコンピュータでは , この計算は並べて係数を与え , 並べて解くことによって並列処理すべきである . また LU-SGS 法では , 2次元の場合には $i+j$ 一定線上の点を並列処理すべきで , 式 (13.53a) で $i+j$ が増加する向きに , 式 (13.53b) で $i+j$ が減少する向きに交互に掃引が行われる . また 3次元の場合には $i+j+k$ 一定面上の点を並列処理することになりかなり煩わしい . これらの方法の因子化の誤差は次の表のようになる .

表 13.2: 因子化に伴う誤差の大きさ

	2次元	3次元
近似因子法	c^2	c^3
LU-SGS 法	c	c
改良型近似因子法	c	c

ここで再び 13.2 節に示したバックステップ流路を通る定常 2次元流れを SMAC Δ形陰解法で解くことにする . そのプログラムは , u^* のサブルーチンを Δ形陰解法のものに換えたほかは前のものと同じである . ただし陰解法では時間ステップを大きく取ることができるので $\Delta t = 0.2$ とし , また安定性も良いのでサブルーチン COMPP でダンピングをはずし $\beta = 1.0$ とし , 漸近定常解が速く得られるようにしている . 以下には Δ形陰解法のサブルーチン IMPCO のみ示すことにする .

```
! ***** Compute u* and v* by delta-form implicit method using Chakravarthy-Osher TVD scheme
SUBROUTINE IMPCO(u,v,uX,vX,p,z,i0,i1,j0,j1,locf,fmax)
DIMENSION u(i0:i1,j0:j1),v(i0:i1,j0:j1),uX(i0:i1,j0:j1),vX(i0:i1,j0:j1),p(i0:i1,j0:j1), &
z(i0:i1,j0:j1),hf(i0:i1),ru(i0:i1,j0:j1),rv(i0:i1,j0:j1),rhs(2,i0:i1,j0:j1), &
a(120,120,3),b(120,120),c(0:4,i0:i1,j0:j1),w1(i0:i1),w11(i0:i1),w2(j0:j1), &
```

```

      w21(j0:j1),lo(2),locf(2,8),fmax(8)
COMMON //na,Re,dx,dy,dt
theta=1.
DO i=i0,i1; j00=0; IF(i>0)j00=j0
  FORALL(j=j00:j1)w2(j)=u(i,j)
  CALL INTUX(w2,w21,j0,j1,j00)
  FORALL(j=j00:j1)uX(i,j)=w21(j)
ENDDO
DO j=j0+1,j1-1; i00=0.; IF(j>0)i00=i0
  FORALL(i=i00:i1)w1(i)=v(i,j)
  CALL INTVX(w1,w11,i0,i1,i00)
  FORALL(i=i00:i1)vX(i,j)=w11(i)
ENDDO
! Compute convection term using Chakravarthy-Osher TVD scheme
DO j=j0,j1-1; i00=0; IF(j>=0)i00=i0                                !hf=uu_P(uu at point P)
  FORALL(i=i0:i1)w1(i)=u(i,j)
  CALL INTUP(w1,w11,i0,i1,i00)                                      !w11=u_P
  i=i0 ; hf(i)=w11(i)*w11(i)                                       !near inlet
  i=0 ; uP=(12.*u(i+1,j)-u(i+2,j))/32.; hf(i)=uP*uP                !near step, u_0j=(u_x)_0j=0
  i=i1-1; hf(i)=w11(i)*w11(i)                                       !near outlet
  i=i1 ; uP=(3.*u(i,j)-u(i-1,j))/2.; hf(i)=uP*uP                  !nearby outside of outlet
  cycle_1: DO i=i00+1,i1-1; uP=w11(i)
    m=1; IF(uP>0.)m=-1; ip=i+m
    IF(i==i1-1.AND.m==1)CYCLE cycle_1
    du=u(i+1,j)-u(i,j); du1=u(ip+1,j)-u(ip,j)
    hf(i)=uP*uP+ABS(uP)*(-du/2.+AMINMOD(du1,4.*du)/6.+AMINMOD(du,4.*du1)/3.)
  ENDDO cycle_1
  FORALL(i=i00+1:i1)ru(i,j)=(hf(i)-hf(i-1))/dx                      !residuals of NSu eqn
ENDDO
DO i=i0+1,i1; j00=0; IF(i>0)j00=j0                                !hf=vu at X
  j=j00 ; hf(j)=0.                                                 !bottom
  j=j00+1; hf(j)=vX(i,j)*(3.*u(i,j-1)+2.*u(i,j)-u(i,j+1))/5.      !bottom wall neighboring point
  j=j1-1 ; hf(j)=vX(i,j)*(3.*u(i,j)+2.*u(i,j-1)-u(i,j-2))/5.    !neighbor top
  j=j1 ; hf(j)=0.                                                 !top
  cycle_2: DO j=j00+1,j1-1
    uX0=uX(i,j); vX0=vX(i,j)
    m=1; IF(vX0>0.)m=-1; jp=j+m
    IF(j==j00+1.AND.m==1 .OR. j==j1-1.AND.m==1)CYCLE cycle_2
    du=u(i,j)-u(i,j-1); du1=u(i,jp)-u(i,jp-1)
    hf(j)=vX0*uX0+ABS(vX0)*(-du/2.+AMINMOD(du1,4.*du)/6.+AMINMOD(du,4.*du1)/3.)
  ENDDO cycle_2
  FORALL(j=j00:j1-1)ru(i,j)=ru(i,j)+(hf(j+1)-hf(j))/dy            !residuals of NSu eqn
ENDDO
DO j=j0+1,j1-1; i00=0; IF(j>0)i00=i0                                !hf=uv at X
  i=i00; hf(i)=0.                                                 !inlet and step
  i=i0+1; hf(i)=uX(i,j)*(v(i-1,j)+v(i,j))/2.                       !neighbor inlet
  i=1; hf(i)=uX(i,j)*(3.*v(i-1,j)+2.*v(i,j)-v(i+1,j))/5.         !neighbor step
  i=i1-1; hf(i)=uX(i,j)*(-v(i-2,j)+6.*v(i-1,j)+3.*v(i,j))/8.    !neighbor outlet
  i=i1 ; hf(i)=uX(i,j)*(-v(i-2,j)+3.*v(i-1,j))/2.                 !outlet
  cycle_3: DO i=i00+2,i1-1; uX0=uX(i,j); vX0=vX(i,j)
    m=1; IF(uX0>0.)m=-1; ip=i+m
    IF(i==i1-1.AND.m==1)CYCLE cycle_3
    dv=v(i,j)-v(i-1,j); dv1=v(ip,j)-v(ip-1,j)
    hf(i)=uX0*vX0+ABS(uX0)*(-dv/2.+AMINMOD(dv1,4.*dv)/6.+AMINMOD(dv,4.*dv1)/3.)
  ENDDO cycle_3
  FORALL(i=i00:i1-1)rv(i,j)=(hf(i+1)-hf(i))/dx                      !residuals of NSv eqn
ENDDO

```

```

DO i=i0,i1-1; j00=0; IF(i>=0)j00=j0                                !hf=vv at P
  FORALL(j=j0:j1)w2(j)=v(i,j)
  CALL INTVP(w2,w21,j0,j1,j00)                                    !w21=v_P
  j=j00 ; hf(j)=w21(j)*w21(j)                                    !near bottom
  j=j1-1; hf(j)=w21(j)*w21(j)                                    !near top
  DO j=j00+1,j1-2; vP=w21(j)
    m=1; IF(vP>0.)m=-1; jp=j+m
    dv=v(i,j+1)-v(i,j); dv1=v(i,jp+1)-v(i,jp)
    hf(j)=vP*vP+ABS(vP)*(-dv/2.+AMINMOD(dv1,4.*dv)/6.+AMINMOD(dv,4.*dv1)/3.)
  ENDDO
  FORALL(j=j00+1:j1-1)rv(i,j)=rv(i,j)+(hf(j)-hf(j-1))/dy        !residuals of NSv eqn
ENDDO
! Compute pressure and viscous terms, and residuals of NS eqn
DO i=i0+1,i1; j00=0; IF(i>0)j00=j0
  FORALL(j=j00:j1-1)ru(i,j)=ru(i,j)+(p(i,j)-p(i-1,j))/dx+(z(i,j+1)-z(i,j))/dy/Re
ENDDO
DO i=i0,i1-1; j00=0; IF(i>=0)j00=j0
  FORALL(j=j00+1:j1-1)rv(i,j)=rv(i,j)+(p(i,j)-p(i,j-1))/dy-(z(i+1,j)-z(i,j))/dx/Re
ENDDO
! Compute u* by implicit SMAC-scheme
ttx=theta*dt/dx; tty=theta*dt/dy
FORALL(i=i0+1:0,j=j0:-1,l=1:4)c(l,i,j)=0.
FORALL(i=i0+1:0,j=j0:-1) c(0,i,j)=1.
DO i=i0+1,i1; j00=0; IF(i>0)j00=j0
  FORALL(j=j00+1:j1-2)w2(j)=vX(i,j)
  CALL INTVP(w2,w21,j0,j1,j00)                                    !w21=v_U
  DO j=j00,j1-1
    up=(u(i,j)+ABS(u(i,j)))/2.; um=(u(i,j)-ABS(u(i,j)))/2.
    vp=(w21(j)+ABS(w21(j)))/2.; vm=(w21(j)-ABS(w21(j)))/2.
    c(1,i,j)=-ttx*(up+1./dx/Re)                                    !coefs of linear eqs
    c(2,i,j)= ttx*(um-1./dx/Re); IF(i==i1)c(2,i,j)=0.          !outlet
    c(3,i,j)=-tty*(vp+1./dy/Re)
    c(4,i,j)= tty*(vm-1./dy/Re)
    c(0,i,j)=1.-c(1,i,j)-c(2,i,j)-c(3,i,j)-c(4,i,j)
    rhs(1,i,j)=-dt*ru(i,j)                                       !rhs of linear eqs
  ENDDO; ENDDO
! Computation du* by modified AF-method
DO j=j0,j1-1; l=j-j0+1; DO i=i0+1,i1; k=i-i0
  a(l,k,1)=c(1,i,j); a(l,k,2)=c(0,i,j); a(l,k,3)=c(2,i,j)
  b(l,k)=rhs(1,i,j)
ENDDO; ENDDO
CALL GAUSSZ(a,b,120,j1-j0,i1-i0)
FORALL(j=j0:j1-1,i=i0+1:i1)rhs(1,i,j)=b(j-j0+1,i-i0)           !du**
DO j=j0,j1-1; l=j-j0+1; DO i=i0+1,i1; k=i-i0
  a(k,l,1)=c(3,i,j); a(k,l,2)=c(0,i,j); a(k,l,3)=c(4,i,j)
  b(k,l)=c(0,i,j)*rhs(1,i,j)
ENDDO; ENDDO
CALL GAUSSZ(a,b,120,i1-i0,j1-j0)
FORALL(j=j0:j1-1,i=i0+1:i1)rhs(1,i,j)=b(i-i0,j-j0+1)         !du*
DO j=j0,j1-1; i00=0; IF(j>=0)i00=i0
  FORALL(i=i00+1:i1)u(i,j)=u(i,j)+rhs(1,i,j)                   !u* at U
ENDDO
! Compute v* by implicit SMAC-scheme
FORALL(i=i0:-1,j=j0+1:0,l=1:4)c(l,i,j)=0.
FORALL(i=i0:-1,j=j0+1:0) c(0,i,j)=1.
DO j=j0+1,j1-1; i00=0; IF(j>0)i00=i0
  FORALL(i=i00:i1-1)w1(i)=uX(i,j)

```

```

CALL INTUP(w1,w11,i0,i1,i00) !w11=u_V
i=0; IF(j<=0)w11(i)=(12.*uX(i+1,j)-uX(i+2,j))/32. !near step
DO i=i00,i1-1
  up=(w11(i)+ABS(w11(i)))/2.; um=(w11(i)-ABS(w11(i)))/2.
  vp=(v(i,j)+ABS(v(i,j)))/2.; vm=(v(i,j)-ABS(v(i,j)))/2.
  c(1,i,j)=-ttx*(up+1./dx/Re) !coefs of linear eqs
  c(2,i,j)= ttx*(um-1./dx/Re); IF(i==i1-1)c(2,i,j)=0. !near outlet
  c(3,i,j)=-tty*(vp+1./dy/Re)
  c(4,i,j)= tty*(vm-1./dy/Re)
  c(0,i,j)=1.-c(1,i,j)-c(2,i,j)-c(3,i,j)-c(4,i,j)
  rhs(2,i,j)=-dt*rv(i,j) !rhs of linear eqs
ENDDO; ENDDO
! Computation dv* by modified AF-method
DO j=j0+1,j1-1; l=j-j0; DO i=i0,i1-1; k=i-i0+1
  a(1,k,1)=c(1,i,j); a(1,k,2)=c(0,i,j); a(1,k,3)=c(2,i,j)
  b(1,k)=rhs(2,i,j)
ENDDO; ENDDO
CALL GAUSSZ(a,b,120,j1-j0-1,i1-i0)
FORALL(j=j0+1:j1-1,i=i0:i1-1)rhs(2,i,j)=b(j-j0,i-i0+1) !dv**
DO j=j0+1,j1-1; l=j-j0; DO i=i0,i1-1; k=i-i0+1
  a(k,l,1)=c(3,i,j); a(k,l,2)=c(0,i,j); a(k,l,3)=c(4,i,j)
  b(k,l)=c(0,i,j)*rhs(2,i,j)
ENDDO; ENDDO
CALL GAUSSZ(a,b,120,i1-i0,j1-j0-1)
FORALL(j=j0+1:j1-1,i=i0:i1-1)rhs(2,i,j)=b(i-i0+1,j-j0) !dv*
DO j=j0+1,j1-1; i00=0; IF(j>0)i00=i0
  FORALL(i=i00:i1-1)v(i,j)=v(i,j)+rhs(2,i,j) !v* at V
ENDDO
lo=MAXLOC(uq); FORALL(k=1:2)locf(k,1)=lo(k); fmax(1)=MAXVAL(uq)
lo=MINLOC(uq); FORALL(k=1:2)locf(k,2)=lo(k); fmax(2)=MINVAL(uq)
lo=MAXLOC(vq); FORALL(k=1:2)locf(k,3)=lo(k); fmax(3)=MAXVAL(vq)
lo=MINLOC(vq); FORALL(k=1:2)locf(k,4)=lo(k); fmax(4)=MINVAL(vq)
END SUBROUTINE IMPCO

! ***** MINMOD function
FUNCTION AMINMOD(a,b)
s=SIGN(1.,a); AMINMOD=s*MAX(0.,MIN(ABS(a),s*b))
END FUNCTION

! ***** Interpolate values at P from values at U
SUBROUTINE INTUP(u,u1,n0,n1,n00)
DIMENSION u(n0:n1),u1(n0:n1)
FORALL(i=n00+1:n1-2)u1(i)=(-u(i-1)+9.*(u(i)+u(i+1))-u(i+2))/16.
i=n00; u1(i)=(3.*u(i)+6.*u(i+1)-u(i+2))/8.
i=n1-1; u1(i)=(3.*u(i+1)+6.*u(i)-u(i-1))/8.
END SUBROUTINE INTUP

! ***** Interpolate values at P from values at V
SUBROUTINE INTVP(u,u1,n0,n1,n00)
DIMENSION u(n0:n1),u1(n0:n1)
FORALL(j=n00+1:n1-2)u1(j)=(-u(j-1)+9.*(u(j)+u(j+1))-u(j+2))/16.
j=n00; u1(j)=(12.*u(j+1)-u(j+2))/32.
j=n1-1; u1(j)=(12.*u(j)-u(j-1))/32.
END SUBROUTINE INTVP

! ***** Interpolate values at X from values at U
SUBROUTINE INTUX(u,u1,n0,n1,n00)
DIMENSION u(n0:n1),u1(n0:n1)
FORALL(j=n00+2:n1-2)u1(j)=(-u(j-2)+9.*(u(j-1)+u(j))-u(j+1))/16.
j=n00 ; u1(j)=0.

```

```

j=n00+1; u1(j)=(3.*u(j-1)+2.*u(j)-u(j+1)/5.)/4.
j=n1-1 ; u1(j)=(3.*u(j)+2.*u(j-1)-u(j-2)/5.)/4.
j=n1   ; u1(j)=0.
END SUBROUTINE INTUX
! ***** Interpolate values at X from values at V
SUBROUTINE INTVX(u,u1,n0,n1,n00)
DIMENSION u(n0:n1),u1(n0:n1)
FORALL (i=n00+2:n1-2)u1(i)=(-u(i-2)+9.*(u(i-1)+u(i))-u(i+1))/16.
i=n00   ; u1(i)=0.
i=n00+1; u1(i)=(3.*u(i-1)+2.*u(i)-u(i+1)/5.)/4.
i=n1-1 ; u1(i)=(3.*u(i)+6.*u(i-1)-u(i-2))/8.
i=n1   ; u1(i)=(3.*u(i-1)-u(i-2))/2.
END SUBROUTINE INTVX
! ***** Solve simultaneous linear eqns with tri-diagonal matrix by Gauss elimination
SUBROUTINE GAUSSZ(a,b,m1,m,n)
DIMENSION a(m1,m1,3),b(m1,m1)
DO k=1,n-1; DO l=1,m
  b(1,k)=b(1,k)/a(1,k,2); a(1,k,3)=a(1,k,3)/a(1,k,2)
  b(1,k+1) =b(1,k+1) -a(1,k+1,1)*b(1,k)
  a(1,k+1,2)=a(1,k+1,2)-a(1,k+1,1)*a(1,k,3)
ENDDO; ENDDO
FORALL (l=1:m)b(1,n)=b(1,n)/a(1,n,2)
DO k=n-1,1,-1
  FORALL (l=1:m)b(1,k)=b(1,k)-a(1,k,3)*b(1,k+1)
ENDDO
END SUBROUTINE GAUSSZ

```

このサブルーチンでは，SMAC Δ形陰解法の式 (13.47a) の差分方程式が

$$c_{ij}^1 u_{i-1,j} + c_{ij}^2 u_{i+1,j} + c_{ij}^3 u_{i,j-1} + c_{ij}^4 u_{i,j+1} + c_{ij}^0 u_{i,j} = rhs_{ij}^1 \quad (13.59a)$$

$$c_{ij}^1 v_{i-1,j} + c_{ij}^2 v_{i+1,j} + c_{ij}^3 v_{i,j-1} + c_{ij}^4 v_{i,j+1} + c_{ij}^0 v_{i,j} = rhs_{ij}^2 \quad (13.59b)$$

$$\begin{aligned}
rhs_{ij}^1 &= -\Delta t \left(\frac{\widehat{uu}_{ij} - \widehat{uu}_{i-1,j}}{\Delta x} + \frac{\widehat{vu}_{ij} - \widehat{vu}_{i,j-1}}{\Delta y} + \frac{p_{ij} - p_{i-1,j}}{\Delta x} + \nu \frac{\zeta_{i,j+1} - \zeta_{ij}}{\Delta y} \right), \\
rhs_{ij}^2 &= -\Delta t \left(\frac{\widehat{vv}_{i+1,j} - \widehat{vv}_{ij}}{\Delta x} + \frac{\widehat{vv}_{ij} - \widehat{vv}_{i,j-1}}{\Delta y} + \frac{p_{ij} - p_{i,j-1}}{\Delta y} - \nu \frac{\zeta_{i+1,j} - \zeta_{ij}}{\Delta x} \right)
\end{aligned} \quad (13.59c)$$

のように書かれ，左辺の演算子には 1 次上流差分，右辺の対流項には Chakravarthy-Osher TVD スキームが用いられる．すなわち

$$\begin{aligned}
c_{ij}^1 &= \frac{\Delta t \theta}{\Delta x} \left(-u_{ij}^+ - \frac{\nu}{\Delta x} \right), & c_{ij}^2 &= \frac{\Delta t \theta}{\Delta x} \left(u_{ij}^- - \frac{\nu}{\Delta x} \right), \\
c_{ij}^3 &= \frac{\Delta t \theta}{\Delta y} \left(-v_{ij}^+ - \frac{\nu}{\Delta y} \right), & c_{ij}^4 &= \frac{\Delta t \theta}{\Delta y} \left(v_{ij}^- - \frac{\nu}{\Delta y} \right), & c_{ij}^0 &= 1 - c_{ij}^1 - c_{ij}^2 - c_{ij}^3 - c_{ij}^4
\end{aligned} \quad (13.60a)$$

$$\begin{aligned}
\widehat{uu}_{ij} &= (u_{ijP})^2 + |u_{ijP}| \left\{ -\frac{1}{2} \Delta_x u_{ijP} + \frac{1}{6} \min\text{mod}(\Delta_x u_{i-I,jP}, 4\Delta_x u_{ijP}) \right. \\
&\quad \left. + \frac{1}{3} \min\text{mod}(\Delta_x u_{ijP}, 4\Delta_x u_{i-I,jP}) \right\}, \\
\widehat{vv}_{ij} &= v_{ijX} u_{ijX} + |v_{ijX}| \left\{ -\frac{1}{2} \Delta_y u_{ijX} + \frac{1}{6} \min\text{mod}(\Delta_y u_{i,j-JX}, 4\Delta_y u_{ijX}) \right. \\
&\quad \left. + \frac{1}{3} \min\text{mod}(\Delta_y u_{ijX}, 4\Delta_y u_{i,j-JX}) \right\}, \quad \dots\dots
\end{aligned} \quad (13.60b)$$

ただし $u_{ij}^{\pm}, v_{ij}^{\pm}$ は式 (13.59a) では U 点, (13.59b) では V 点の値が取られる. また \widehat{u}_{ij}, \dots は Chakravarthy-Osher TVD スキームの数値流束で, $\widehat{u}_{ij}, \widehat{v}_{ij}$ は P 点上, $\widehat{u}_{ij}, \widehat{v}_{ij}$ は X 点上に定義される. $I = 1 (u_{ijP} \leq 0), = -1 (u_{ijP} > 0), J = 1 (v_{ijX} \leq 0), = -1 (v_{ijX} > 0)$ である.

このサブルーチンでは, 右辺の対流項, 圧力項, 粘性項が計算され, 連立 1 次方程式の係数と右辺の値が設定され, 改良型近似因子法で u^*, v^* の値が求められる. 対流項の計算では境界条件を考慮する必要がある. ステップ ($x = 0$) のところの境界条件は $u_{0j} = v_{0j} = (u_x)_{0j} = 0$ で, 3 次式を考えれば $u_{0jP} = (12u_{1j} - u_{2j})/32$, $v_{1jX} = (3v_{0j} + 2v_{1j} - v_{2j})/5/4$ となる. 下方と上方の境界に関しても同様である. 改良型近似因子法は, ダクトの欠けている部分に 1 次式 $1 \cdot u_{ij} = 0$ を補い, 長方形領域に対して適用される. 3 重対角行列の連立 1 次方程式はサブルーチン GAUSSZ を用いて解かれる. 計算のベクトル化率を上げるために, 次元の 1 つ多い配列 a, b を用い, 連立 1 次方程式の係数の設定と Gauss 消去法の計算はすべて並列に行われる.

解の収束状況を表 13.3 に示す. レイノルズ数 $Re = 70$, 時間間隔 $\Delta t = 0.2$, COMPP の $\beta = 1.0$. また $res_{NS}, res_P, \Delta flow$ の定義は表 13.1 と同じで, これらの値はすべて 10^5 倍されている. 解の収束判定条件は前と同じで, 収束解は $n = 706$ で得られる. 流速の最大値はダクト入口中心の 1.5 で, 圧力の最大値はダクト入口の 0.1547, 最小値はステップ角近傍の -0.0840 , 再付着距離は 5.7 の近くにある. 表 13.4 は時間間隔 Δt を変えた場合の解の収束に要する反復数 n を陽解法のものも含め比較したものである. この表から Δ 形陰解法の反復数は陽解法に比べ 1/4 以下になることが分かる. なお CPU 時間は陽解法に較べそれほど減らない. Δ 形陰解法の利点は CPU 時間よりもむしろ高レイノルズ数まで計算できることである. 表 13.6 は高レイノルズ数の計算結果をまとめたものである. 更に高レイノルズ数では剥離域が出口に達するため収束解は得られない.

表 13.3: バックステップダクト流れの解の収束状況 ($Re = 70$, SMAC Δ 形陰解法)

n	$\times 10^{-5}$		
	res_{NS}	res_P	$\Delta flow$
250	508	397	66
500	62	24	243
750	6	7	362
1000	1	2	293

表 13.4: バックステップダクト流れの収束状況の比較 ($Re = 70$)

	Δt	C_{max}	n	p_{max}	p_{min}
陽 解 法	0.04	0.18	3 034	0.1539	-0.0840
Δ 形陰解法	0.1	0.45	1 171	0.1541	-0.0849
	0.2	0.9	706	0.1546	-0.0841
	0.4	1.8	655	0.1544	-0.0849
	0.5	2.25	700	0.1547	-0.0846
	0.6	2.7	1 106	0.1540	-0.0849
	0.8	3.6		収束解得られず	

表 13.5: バックステップダクトの高レイノルズ数流れ

Re	n	再付着距離	α	β	備考
200	1100	11.7	1.0	1.0	
300	1884	15.3	1.0	0.5	
400	2796	18.4	1.5	0.5	
500	7854	20.8	1.5	0.5	計算途中に出口逆流

13.6 直角座標系から一般曲線座標系への変換

直角座標 (cartesian coordinates) を x , 一般曲線座標 (general curvilinear coordinates) を ξ で表す¹³ . これらの座標間の変換の測度 (metric) x_ξ, \dots と ξ_x, \dots の間には次の関係が成立する¹⁴ .

$$\begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix} \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13.61)$$

また変換のヤコビアン J は

$$J = \begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{vmatrix} \quad (13.62)$$

となる . これらは 2 次元の場合には次のようになる .

$$x_\xi = J\eta_y, \quad x_\eta = -J\xi_y, \quad y_\xi = -J\eta_x, \quad y_\eta = J\xi_x, \quad J = x_\xi y_\eta - x_\eta y_\xi \quad (13.63)$$

x に関する微分を ξ に関する微分に書き換えれば次のようになる¹⁵ .

$$\frac{\partial}{\partial x_i} = \frac{\partial \xi_j}{\partial x_i} \frac{\partial}{\partial \xi_j}, \quad \nabla = (\nabla \xi_j) \frac{\partial}{\partial \xi_j} \quad (13.64)$$

具体的には , 例えば

$$\frac{\partial}{\partial x} = \frac{1}{J} \begin{vmatrix} y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \\ \partial/\partial \xi & \partial/\partial \eta & \partial/\partial \zeta \end{vmatrix}, \quad 2 \text{次元では} \quad \frac{\partial}{\partial x} = \frac{1}{J} \left(y_\eta \frac{\partial}{\partial \xi} - y_\xi \frac{\partial}{\partial \eta} \right)$$

また一般に

$$\frac{\partial}{\partial \xi_j} J \frac{\partial \xi_j}{\partial x_i} = 0, \quad \frac{\partial}{\partial \xi_j} J \nabla \xi_j = 0$$

であるから次の関係が導かれる .

$$J \frac{\partial \phi}{\partial x_i} = J \frac{\partial \xi_j}{\partial x_i} \frac{\partial \phi}{\partial \xi_j} = \frac{\partial}{\partial \xi_j} \left(J \frac{\partial \xi_j}{\partial x_i} \phi \right) \quad (13.65a)$$

$$J \nabla \phi = \frac{\partial}{\partial \xi_j} J (\nabla \xi_j) \phi \quad (13.65b)$$

$$J \nabla^2 = \frac{\partial}{\partial \xi_j} \left(J g^{jk} \frac{\partial}{\partial \xi_k} \right) \quad (13.65c)$$

ただし (g^{ij}) は測度テンソルで , その成分は次のように定義される .

$$g^{ij} = \nabla \xi_i \cdot \nabla \xi_j = \frac{\partial \xi_i}{\partial x_k} \frac{\partial \xi_j}{\partial x_k} \quad (13.66)$$

¹³ここで一般は直交に対し直交+非直交の意である . 以前には非直交であることを強調し非直交曲線座標格子 (non-orthogonal curvilinear coordinate grid) などと言われた .

¹⁴例えば $x = x(\xi, \eta, \zeta) = x(\xi(x, y, z), \eta(x, y, z), \zeta(x, y, z))$ であるから , この式を y で微分すれば $0 = \xi_y x_\xi + \eta_y x_\eta + \zeta_y x_\zeta$ なる関係が得られ , 式 (??) はこれらの関係を纏めたものである .

¹⁵ここでは Einstein 総和規約 (Einstein convention) が用いられる . すなわち同じ添字が繰り返されるときには $a_i b_i = a_1 b_1 + a_2 b_2 + a_3 b_3$ のように解釈する .

流線に沿う微分は次のように書き換えられる .

$$\mathbf{u} \cdot \nabla = u_i \frac{\partial}{\partial x_i} = u_i \frac{\partial \xi_j}{\partial x_i} \frac{\partial}{\partial \xi_j} = U_j \frac{\partial}{\partial \xi_j} = \mathbf{U} \cdot \tilde{\nabla} \quad (13.67)$$

ただし U_j は反変速度 (contravariant velocity) の ξ_j 方向成分である . 反変速度 U は ξ 空間の流速で¹⁶ , 物理空間の流速 \mathbf{u} との間には次の関係がある .

$$U_j = \frac{\partial \xi_j}{\partial x_i} u_i, \quad u_i = \frac{\partial x_i}{\partial \xi_j} U_j \quad (13.68)$$

なおこのような関係を知らずに , 基礎方程式の x_i の微分を , 式 (13.64) によって ξ_j の微分に書き換えるとたいへん長い式になり収拾がつかなくなる .

流れの渦度 $\zeta = \nabla \times \mathbf{u}$ の x_i 方向成分は

$$\zeta_i = \epsilon_{ijk} \frac{\partial u_k}{\partial x_j} \quad (13.69)$$

ただし (ϵ_{ijk}) は Eddington の ϵ で

$$\epsilon_{ijk} = \begin{cases} 1 & ((ijk) = (123), (231), \text{または } (312)) \\ -1 & ((ijk) = (321), (213), \text{または } (132)) \\ 0 & (\text{その他の場合}) \end{cases}$$

例えば , $\zeta_1 = \partial u_3 / \partial x_2 - \partial u_2 / \partial x_3$ である . 任意のベクトル \mathbf{a} に対し次の関係が成立する .

$$\nabla \xi_l \cdot \nabla \times \mathbf{a} = \nabla \xi_l \cdot \nabla \xi_m \times \frac{\partial \mathbf{a}}{\partial \xi_m} = \frac{1}{J} \epsilon_{lmn} \frac{\partial x_i}{\partial \xi_n} \frac{\partial a_i}{\partial \xi_m} = \frac{1}{J} \epsilon_{lmn} \frac{\partial}{\partial \xi_m} \left(\frac{\partial \mathbf{x}}{\partial \xi_n} \cdot \mathbf{a} \right) \quad (13.70)$$

なおこの式の導出には次式が用いられた .

$$x_\xi = J \begin{vmatrix} \eta_y & \eta_z \\ \zeta_z & \zeta_z \end{vmatrix}, \dots, \quad \epsilon_{lmn} \frac{\partial^2 \mathbf{x}}{\partial \xi_m \partial \xi_n} = 0 \quad (13.71)$$

式 (13.70) で , 特に $\mathbf{a} = \mathbf{u}$ ならば . 次の反変渦度と反変速度の関係が得られる .

$$Z_l = \frac{1}{J} \epsilon_{lmn} \frac{\partial}{\partial \xi_m} (g_{nj} U_j) \quad (13.72)$$

ただし (g_{ij}) は測度テンソルで , $(g_{ij})^{-1} = (g^{ij})$, その成分は次式で定義される .

$$g_{ij} = \frac{\partial \mathbf{x}}{\partial \xi_i} \cdot \frac{\partial \mathbf{x}}{\partial \xi_j} = \frac{\partial x_k}{\partial \xi_i} \frac{\partial x_k}{\partial \xi_j} \quad (13.73)$$

また Z_l は反変渦度の ξ_l 方向成分で , 反変渦度 Z と物理空間の渦度 ζ の間には次の関係がある .

$$Z_j = \frac{\partial \xi_j}{\partial x_i} \zeta_i, \quad \zeta_i = \frac{\partial x_i}{\partial \xi_j} Z_j \quad (13.74)$$

¹⁶ 反変速度が写像空間の流速であることは次のように説明できる . ある時間に点 \mathbf{x} にある流体粒子は単位時間後に点 $\mathbf{x} + \mathbf{u}$ に移動する . 点 \mathbf{x} の曲線座標を ξ とすれば点 $\mathbf{x} + \mathbf{u}$ の曲線座標は $\xi + \mathbf{u} \cdot \nabla \xi$ となる . したがって反変速度 $U \equiv \mathbf{u} \cdot \nabla \xi$ は写像空間で流体粒子が単位時間に移動する距離すなわち流速であることが分かる .

13.7 曲線座標格子の SMAC Δ形陰解法

物理空間の曲線座標格子を写像空間の立方体格子に写像する．立方体格子の稜の長さ $\Delta\xi, \Delta\eta, \Delta\zeta$ は通常 1 に取られるが，以下の説明では当面 省略せずに残しておくことにする．直角座標格子の MAC 型解法を曲線座標格子の解法に拡張する際には，その特徴を損なわないようにしなければならない．そのために，各セル面の中心に妥当な流速成分として写像空間の流速である反変速度 \tilde{U} の成分またはこれに相当のものが定義される．既存の解法では，反変速度成分，体積流束，反変物理速度成分，共変物理速度成分が用いられているが，ここでは体積流束 (volume flux) $\tilde{U} \equiv J\mathbf{U}$ を用いることにする．

連続方程式は式 (13.65) を用いれば次のようになる．

$$J \frac{\partial u_i}{\partial x_i} = \frac{\partial}{\partial \xi_j} \left(J \frac{\partial \xi_j}{\partial x_i} u_i \right) = \frac{\partial}{\partial \xi_j} \tilde{U}_j = 0 \quad (13.75)$$

図 13.6 は ξ 空間内の 1 つの立方体格子セルを示したものである．式 (13.75) は，この立方体格子セルにわたって積分すれば

$$(\tilde{U}_{100} - \tilde{U}_{000})\Delta\eta\Delta\zeta + (\tilde{V}_{010} - \tilde{V}_{000})\Delta\zeta\Delta\xi + (\tilde{W}_{001} - \tilde{W}_{000})\Delta\xi\Delta\eta = 0 \quad (13.76)$$

となる．式 (13.76) の中で，

$$\begin{aligned} \tilde{U}_{000}\Delta\eta\Delta\zeta &= \left(J \frac{\partial \xi}{\partial x_i} u_i \right)_{000U} \Delta\eta\Delta\zeta = \Delta\eta\Delta\zeta \begin{vmatrix} u & x_\eta & x_\zeta \\ v & y_\eta & y_\zeta \\ w & z_\eta & z_\zeta \end{vmatrix}_{000U} \\ &\approx \mathbf{u}_{000U} \cdot (\mathbf{x}_{010} - \mathbf{x}_{000}) \times (\mathbf{x}_{001} - \mathbf{x}_{000}) \end{aligned}$$

ここに $(\mathbf{x}_{010} - \mathbf{x}_{000}) \times (\mathbf{x}_{001} - \mathbf{x}_{000})$ は \mathbf{x} 空間内の対応する六面体格子セルの， $\xi = \xi_0$ 面の面積ベクトルであるから， $\tilde{U}_{000}\Delta\eta\Delta\zeta$ はこの面を通る流量である．したがって式 (13.76) はこの六面体セルの 6 つの面から出入りする流量の和が 0 になるという意味の式である．写像空間内の流れに垂直に取られた単位面積当たりの物理空間の体積流量は体積流束 (volume flux) と呼ばれ， $\tilde{U}, \tilde{V}, \tilde{W}$ はこの体積流束の ξ, η, ζ 方向成分で，また $\Delta\xi = \Delta\eta = \Delta\zeta = 1$ ならば対応する六面体セルの面を通る流量になる．なお上式の導出には式 (??) から得られる次の関係が用いられた．

$$\xi_x = \frac{1}{J} \begin{vmatrix} y_\eta & y_\zeta \\ z_\eta & z_\zeta \end{vmatrix}, \dots$$

次に体積流束の運動方程式について述べる．体積流束成分 \tilde{U}_i は流速 \mathbf{u} の左から $J\nabla\xi_i$ を演算したものであるから， \tilde{U}_i の運動方程式は運動方程式 $d\mathbf{u}/dt \equiv \partial\mathbf{u}/\partial t + \nabla \cdot \mathbf{u}\mathbf{u} = -\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{g}$ の左から $J\nabla\xi_i$ を演算

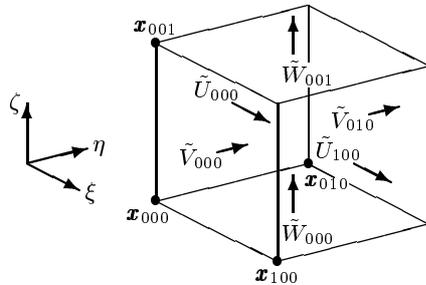


図 13.6: ξ 空間内の立方体格子セル

することによって導くことができる．その対流項は次のようになる¹⁷．

$$\begin{aligned} C_l &= J \nabla \xi_l \cdot (\nabla \cdot \mathbf{u}) = \nabla \xi_l \cdot \frac{\partial}{\partial \xi_i} \tilde{U}_i \mathbf{u} = \frac{\partial}{\partial \xi_i} \tilde{U}_i U_l - \mathbf{u} \cdot \tilde{U}_i \frac{\partial}{\partial \xi_i} \nabla \xi_l \\ &= \frac{\partial}{\partial \xi_i} \tilde{U}_i U_l + \{i^l_j\} \tilde{U}_i U_j \end{aligned} \quad (13.77)$$

ただし $\{i^l_j\}$ は Christoffel の記号である．

$$\frac{\partial^2 \mathbf{x}}{\partial \xi_i \partial \xi_j} = \{i^l_j\} \frac{\partial \mathbf{x}}{\partial \xi_l}, \quad \{i^l_j\} = \frac{1}{2} g^{lk} \left(\frac{\partial g_{jk}}{\partial \xi_i} + \frac{\partial g_{ik}}{\partial \xi_j} - \frac{\partial g_{ij}}{\partial \xi_k} \right) \quad (13.78)$$

なお式 (13.77) の導出には次の関係が用いられた¹⁸．

$$\frac{\partial \mathbf{x}}{\partial \xi_j} \cdot \frac{\partial}{\partial \xi_i} \nabla \xi_l = -\nabla \xi_l \cdot \frac{\partial^2 \mathbf{x}}{\partial \xi_i \partial \xi_j} = -\nabla \xi_l \cdot \{i^k_j\} \frac{\partial \mathbf{x}}{\partial \xi_k} = -\{i^l_j\}$$

粘性拡散項については 14.4 節に詳しく述べる．結局，体積流束成分 \tilde{U}_l の運動方程式は次のようになる．

$$\frac{\partial}{\partial t} \tilde{U}_l = -\frac{\partial}{\partial \xi_i} \tilde{U}_i U_l - \{i^l_j\} \tilde{U}_i U_j - \tilde{g}^{li} \frac{\partial p}{\partial \xi_i} - \nu \epsilon_{lij} \frac{\partial}{\partial \xi_i} (\tilde{g}_{jk} \tilde{Z}_k) + J \frac{\partial \xi_l}{\partial x_k} g_k \quad (13.79)$$

ただし $\tilde{U}_i = J U_i$ ， $\tilde{g}^{ij} = J g^{ij}$ ， $\tilde{g}_{ij} = g_{ij}/J$ ， $\tilde{Z} = J Z$ である．この式の右辺第 2 項は，対流項を保存形で表したために出てきた付加項で，その値は滑らかな格子では通常小さく等間隔の直交格子では 0 になる．

13.2 節の SMAC 法の基礎方程式は一般曲線座標系の SMAC 法の式に書き換えれば次のようになる．

$$\tilde{U}_l^* = \tilde{U}_l^n - \Delta t \left(C_l + \tilde{g}^{li} \frac{\partial p}{\partial \xi_i} - D_l \right)^n \quad (l = 1, 2, 3) \quad (13.80a)$$

$$\tilde{U}_l^{n+1} = \tilde{U}_l^* - \Delta t \tilde{g}^{li} \frac{\partial \phi}{\partial \xi_i} \quad (13.80b)$$

$$\frac{\partial}{\partial \xi_l} \left(\tilde{g}^{li} \frac{\partial \phi}{\partial \xi_i} \right) = \frac{1}{\Delta t} \frac{\partial}{\partial \xi_l} \tilde{U}_l^*, \quad p^{n+1} = p^n + \phi \quad (13.80c)$$

SMAC 法の陰解法化は，前にも述べたように，NS 方程式の前段の式 (13.80a) だけを陰解法化すれば達成でき，それによって時間間隔 Δt を大きく取ることが可能になる．一般曲線座標系の SMAC Δ 形陰解法の式は次のようになる．

$$\left\{ J + \Delta t \theta \left(\tilde{U}_i \frac{\partial}{\partial \xi_i} - \mathfrak{D}_l \right) \right\} \Delta U_l^* = rhs_l^n, \quad (l = 1, 2, 3) \quad (13.81a)$$

$$\begin{aligned} rhs_l &= -\Delta t \left(C_l + \tilde{g}^{li} \frac{\partial p}{\partial \xi_i} - D_l \right), \quad \tilde{U}_l^* = \tilde{U}_l^n + J \Delta U_l^* \\ \tilde{U}_l^{n+1} &= \tilde{U}_l^* - \Delta t \tilde{g}^{li} \frac{\partial \phi}{\partial \xi_i} \end{aligned} \quad (13.81b)$$

$$\frac{\partial}{\partial \xi_l} \left(\tilde{g}^{li} \frac{\partial \phi}{\partial \xi_i} \right) = \frac{1}{\Delta t} \frac{\partial}{\partial \xi_l} \tilde{U}_l^*, \quad p^{n+1} = p^n + \phi \quad (13.81c)$$

定常流れの計算では，解が収束すれば $\tilde{U}_l^* = \tilde{U}_l^n$ であるから，式 (13.81a) の右辺 rhs は残差， ΔU_l^* は修正値を表している．解が収束すればこの残差はゼロに近づき，修正値もゼロに近づく．したがってこの式の

¹⁷ 第 2 式から第 3 式へは式 (13.65) と (13.68)，第 3 式へは $\tilde{U}_i U_l = \nabla \xi_l \cdot (\tilde{U}_i \mathbf{u})$ の微分が用いられた．

¹⁸ 第 1 式から第 2 式へは $\frac{\partial}{\partial \xi_i} \left(\frac{\partial \mathbf{x}}{\partial \xi_j} \cdot \nabla \xi_l \right) = \frac{\partial}{\partial \xi_i} \delta_{jl} = 0$ なる関係，第 3 式へは上の Christoffel 記号の式，それから第 4 式へは $\nabla \xi_l \cdot (\partial \mathbf{x} / \partial \xi_k) = \delta_{kl}$ を用いている．

右辺は，解の精度に直接関係するので精度良く離散化しなければならないが，左辺の演算子は大胆に近似することができ，1次上流差分が用いられ，拡散項は主要部のみ考慮され，また近似因子法が適用される． \mathcal{D}_l は拡散項の近似演算子で無視することもできるが，岐点の近傍の流れではこの項の働きが大きいのでここではその主要部のみ残すことにすれば，一般に $g_{ii} > g_{ij}$ ($i \neq j$) であるから次のようになる．

$$\mathcal{D}_{1\cdot} = \nu_{\text{eff}} \left\{ \frac{\partial}{\partial \xi} \left(\tilde{g}_{22} \tilde{g}_{33} \frac{\partial}{\partial \xi} J \cdot \right) + \frac{\partial}{\partial \eta} \left(\tilde{g}_{33} \frac{\partial}{\partial \eta} g_{11\cdot} \right) + \frac{\partial}{\partial \zeta} \left(\tilde{g}_{22} \frac{\partial}{\partial \zeta} g_{11\cdot} \right) \right\} \quad (13.82)$$

また2次元の場合には

$$\mathcal{D}_{1\cdot} = \nu_{\text{eff}} \left\{ \frac{\partial}{\partial \xi} \left(\tilde{g}_{22} \frac{\partial}{\partial \xi} \cdot \right) + \frac{\partial^2}{\partial \eta^2} (\tilde{g}_{11\cdot}) \right\}$$

非定常流れの計算では， $\theta = 1/2$ の Crank-Nicholson 法を用い，各時間ステップごとに Newton 反復法で反復計算を行うことにする．この非定常流れの SMAC Δ 形陰解法の式は次のようになる．

$$\left\{ J + \Delta t \theta \left(\tilde{U}_l \frac{\partial}{\partial \xi_i} - \tilde{\mathcal{D}}_l \right) \right\} \Delta U_l^{*(m)} = -(\tilde{U}_l^{(m-1)} - \tilde{U}_l^n) + \frac{1}{2} (rhs_l^n + rhs_l^{(m-1)}), \quad (l = 1, 2, 3) \quad (13.83a)$$

$$\tilde{U}_l^{*(m)} = \tilde{U}_l^{(m-1)} + J \Delta U_l^{*(m)}$$

$$\tilde{U}_l^{(m)} = \tilde{U}_l^{*(m)} - \frac{1}{2} \Delta t \tilde{g}^{li} \frac{\partial \phi^{(m)}}{\partial \xi_i} \quad (13.83b)$$

$$\frac{\partial}{\partial \xi_i} \left(\tilde{g}^{li} \frac{\partial \phi^{(m)}}{\partial \xi_i} \right) = \frac{2}{\Delta t} \frac{\partial}{\partial \xi_i} \tilde{U}_l^{*(m)}, \quad p^{(m)} = p^{(m-1)} + \phi^{(m)} \quad (13.83c)$$

上添字 (m) は時間ステップ $(n+1)$ における第 m 近似値を意味する． $\tilde{U}_l^{(0)} = \tilde{U}_l^n$ ， $p^{(0)} = p^n$ と置いて計算を始め，解が収束すれば $\tilde{U}_l^{(m)} = \tilde{U}_l^{n+1}$ ， $p^{(m)} = p^{n+1}$ となる．一般に定常流れが収束するまでには数千回の反復が必要であるが，非定常流れの各時間ステップの解は3~5回の反復で十分である．定常流れの収束までに多くの反復を要するのは拡散効果によるもので，一方1時間ステップに拡散効果の実質及び範囲は限られるので1時間ステップ当たりの反復回数は少いで済むことになる．

13.8 バックステップ流路流れのプログラム

前章と同じバックステップ流路を通る流れを，長方形格子の代わりに曲線座標格子を用い SMAC Δ 形陰解法で解く1つのプログラムを示す．ただしレイノルズ数がある程度高くなっても剥離域が下流境界に達しないように下流側流路はかなり延長されている．このプログラムは，長方形格子の SMAC Δ 形陰解法のプログラムとだいたい同じであるがかなり長大になるので，始めに図 13.7 のフローチャートを使って計算手順の概要を説明する．このプログラムではメインプログラムで Mode を指定することにより，定常または非定常流れ，低レイノルズ数または高レイノルズ数流れの都合4つの場合を選択できる．定常流れの初期値はこのプログラムの中で与えられるが，非定常流れの初期値は別途計算したものを読込ませなければならない．低レイノルズ数流れは対流項を2次中心差分で計算するもの，高レイノルズ数流れは3次の Chakravarthy-Osher TVD スキームで計算するものである．

このフローチャートによれば，始めに GRID で格子形成，METRIC で後の計算に必要な測度，ヤコビアン，測度テンソル，また COEF で圧力差分方程式の係数の設定が行われ，それから定常流れの予測値の計算または非定常流れの初期値の読込みが行われる．それから左側の反復計算に移るが， n は時間ステップ， m は非定常流れの各時間ステップにおける反復数である．各反復計算では式 (13.81) または (13.83) により COMP1

で \tilde{U}^* , COMPP で ϕ と p , COMPU2 で \tilde{U} , また COMPUZ で ζ の計算が行われる. $n20$ 回ごとに NS 差分方程式の最大残差値と圧力差分方程式の最大残差値を打出し, また非定常流れでは CG に必要なデータをファイルに書込む. この反復計算は解が発散したときには直ちに中止し, 解が収束または規定の反復数に達した時に次のステップまたは計算を終了する. 終了直前には計算の最終結果をファイルに書込み CG で流れの可視化を行う.

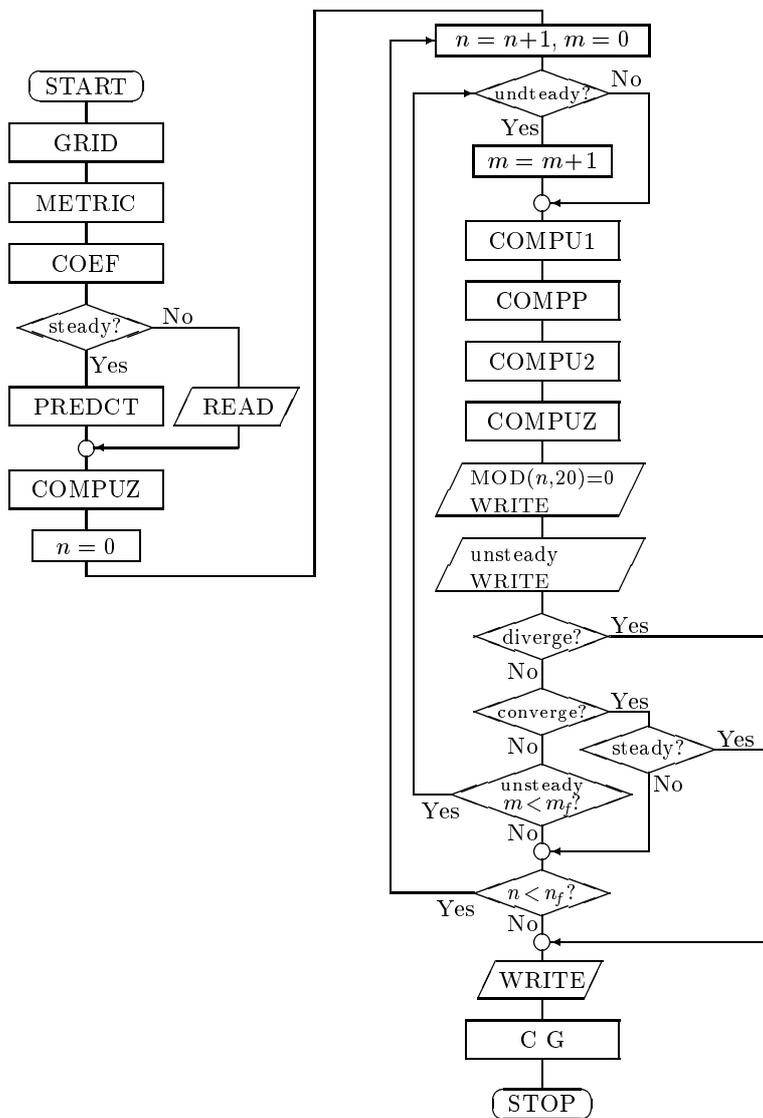


図 13.7: SMAC Δ形陰解法のフローチャート

以下にバックステップ流路を通る定常流れのプログラムを示す.

```

PROGRAM MAIN
! *****
! Flow Problem: 2D Incompressible Flow through Backward Facing Stepped Duct
! Numerical Method: General Curvilinear Coordinate Grid, SMAC Delta-Form Implicit Method,
!   Chakravarthy-Osher TVD Scheme
! *****
    
```

```

PARAMETER (if=140,jf=25)
DIMENSION x(2,0:if,0:jf),UJ(2,0:if,0:jf),u(2,0:if,0:jf),p(0:if,0:jf),phi(0:if,0:jf), &
          z(0:if,0:jf),UJX(2,0:if,0:jf),f(0:if,0:jf),co(0:if,0:jf,-1:1,-1:1),locf(2,8),fmax(8)
COMMON //n,Re,dt,i0,j0,lowRe,steady,unsteady
COMMON /COMPUZ1/UJX
LOGICAL lowRe,steady,unsteady
CHARACTER*4 z1,z2; CHARACTER*10 z3(8)
DATA nf,mf,Re,dt,Mode/5000,5,1000.,1.,2/          !nf=max n, mf=max(m), Re=Reynolds number
i0=if-1; j0=jf-1
! Mode=1: for low Re steady flow
! Mode=2: for high Re steady flow
! Mode=3: for low Re unsteady flow
! Mode=4: for high Re unsteady flow
steady = (Mode==1 .OR. Mode==2)
unsteady = (Mode==3 .OR. Mode==4)
lowRe = (Mode==1 .OR. Mode==3)
CALL GRID(x,if,jf)                                !grid generation
CALL METRIC(x)                                    !metrics, metric tensors
CALL COEF(co)                                     !coefficients of press-fde
IF(steady)CALL PREDCT(x,UJ,u,p)                   !predict JU and p
IF(unsteady)READ(10)x,UJ,u,p                      !unsteady: initial data
CALL COMPUZ(UJ,u,z,locf,fmax,CFLmax)              !velocities vorticity & divergence
OPEN(20,FILE='OUTPUT.dat')
WRITE(20,60); IF(unsteady)WRITE(20,61)
n=0; 100 n=n+1                                     !n: step
m=0; 101 IF(unsteady)m=m+1                         !m: approx
CALL COMPU1(m,UJ,u,p,z,locf,fmax)                 !compute JU*
CALL COMPP (UJ,p,phi,co,resP,locf,fmax)           !compute phi and p
CALL COMPU2(UJ,phi)                               !compute JU
CALL COMPUZ(UJ,u,z,locf,fmax,CFLmax)              !velocities vorticity & divergence
! Decide convergence and output computational results
IF(MOD(n,20)==0 .AND. (steady.OR.m==1))THEN
  i=if; resNS=0.
  resNS=AMAX1(fmax(1),-fmax(2),fmax(3),-fmax(4))
  flow=3./8.*(3.*UJ(1,i,0)+UJ(1,i,1)+UJ(1,i,23)+3.*UJ(1,i,24))      !outlet flow rate
  DO j=1,23,2; flow=flow+(UJ(1,i,j)+4.*UJ(1,i,j+1)+UJ(1,i,j+2))/3.; ENDDO
  CALL CPU_TIME(sec)
  WRITE(20,62)n,resNS,resP,CFLmax,flow,sec          !print residuals
ENDIF
60 FORMAT(/1H 3X 'n', 6X 'resNS', 6X 'resP', 7X 'CFLmax', 4X 'outflow', 5X 'CPU-time'/)
61 FORMAT(1H 5X 'm', 6X 'resNS', 6X 'resP'/)
62 FORMAT(1H I5, 2F11.5, 2X F9.3, F11.5, 2X F9.2)
IF(resNS>2. .OR. resP>2.)                          GOTO 110          !diverge
IF(resNS<.0001 .AND. resP<.0001)THEN              !converge
  IF(steady)                                        GOTO 110
  IF(unsteady)                                     GOTO 111
ENDIF
IF(unsteady.AND.m<mf)                              GOTO 101          !unsteady:(m)-approx
111 IF(n<nf)                                        GOTO 100          !n-step
110 CONTINUE
DO k=1,6
  SELECT CASE(k)
    CASE(1); FORALL(i=0:if,j=0:jf)f(i,j)=x(1,i,j) ; z1=" x"; z2=" "; mz= 0
    CASE(2); FORALL(i=0:if,j=0:jf)f(i,j)=x(2,i,j) ; z1=" y"; z2=" "; mz= 0
    CASE(3); FORALL(i=0:if,j=0:jf)f(i,j)=u(1,i,j) ; z1=" u"; z2=" "; mz= 0
    CASE(4); FORALL(i=0:if,j=0:jf)f(i,j)=u(2,i,j) ; z1=" v"; z2=" "; mz= 0
    CASE(5); FORALL(i=0:if,j=0:jf)f(i,j)=p(i,j) ; z1=" p"; z2="*10 "; mz= 1
  
```

```

CASE(6); FORALL(i=0:if,j=0:jf)f(i,j)=z(i,j) ; z1="zeta"; z2="/10 "; mz=-1
ENDSELECT !mz:scale factor
CALL WRTF(f,z1,z2,mz,n) !print out computational results
ENDDO
CALL REATTACH(x,UJ,UJX,if,jf,xrap,k) !compute reattachment point xrap
WRITE(20,'(5X A7,F5.1,3X I4)')'xrap = ',xrap,k !print out xrap
! Output locations and values of maximum and minimum of residuals and divergence
DATA z3/'maxUq = ','minUq = ','maxVq = ','minVq = ','maxphi = ','minphi = ','maxdiv = ','mindiv = '/
FORALL(k=1:2,i=1:8)locf(k,i)=locf(k,i)-1
DO i=1,8; WRITE(20,'(5X A10,2I4,3X E10.3)')z3(i),(locf(k,i),k=1,2),fmax(i); ENDDO
CALL StepDF_CG(x,u,p,z,if,jf) !computer graphics for steady
CLOSE(20)
END PROGRAM MAIN

! ***** Generate curvilinear coordinate grid by analytical method
SUBROUTINE GRID(x,if,jf) !if=140,jf=25
! Generate by sloving boundary value problem of elliptic pde
DIMENSION x(2,0:if,0:jf),x0(2,0:if,0:jf),xxi(4,0:if,0:jf),aJ(0:if,0:jf),g(3,0:if,0:jf), &
P(0:if,0:jf),Q(0:if,0:jf),f(0:if,0:jf)
CHARACTER*4 z1
DATA dxi,det,al1/3*1./
nf=200; i1=35; i2=40; i0=if-1; j0=jf-1
OPEN(20,FILE='OUTPUT.dat')
! ***** Give values of grid coordinates on boundaries by geometrical series
! a+ar+ar^2+...+ar^(n-1)=a(r^n-1)/(r-1)
x(1,i1,0)=0.; x(2,i1,0)=0.
DO i=i1-1,0,-1
x(1,i,0)=x(1,i+1,0)-.09975*1.055**(i1-i-1); x(2,i,0)=0. !bottom wall
ENDDO
DO i=i1+1,i2
x(1,i,0)=0.; x(2,i,0)=x(2,i-1,0)-.10045*1.35**(i-i1-1) !step wall
ENDDO
DO i=i2+1,70
x(1,i,0)=x(1,i-1,0)+.2*1.055**(i-i2-1) !bottom wall
ENDDO
FORALL(i=71:if)x(1,i,0)=x(1,70,0)+.8232*FLOAT(i-70) !bottom wall
FORALL(i=i2:if)x(2,i,0)=-1.
FORALL(i=0:if)x(2,i,jf)=3. !top wall
! ***** Give predict values of x on top bound
FORALL(i=0:i1) x(1,i,jf)=x(1,i,0)
FORALL(i=i1+1:i2)x(1,i,jf)=x(1,i1,jf)-x(2,i,0)
FORALL(i=i2+1:if)x(1,i,jf)=x(1,i2,jf)+(x(1,if,0)-1.)/x(1,if,0)*x(1,i,0)
! ***** Determine starting values of x,y by interpolation
DO i=0,if; DO j=1,j0
et=j/FLOAT(jf); alp=(-et*et+1.6*et+.4)*et
! The cubic polynomial is formed to satisfy the conds of f(0)=0, f(1)=1, f'(0)=.4, f'(1)=.6
! for finer grid near walls. But it is impossible sufficiently to control grid spaces by it.
x(1,i,j)=(1.-alp)*x(1,i,0)+alp*x(1,i,jf)
x(2,i,j)=(1.-alp)*x(2,i,0)+alp*x(2,i,jf)
ENDDO; ENDDO
n=0; 100 n=n+1; IF(n==10)al1=1.4
! ***** Compute x_xi y_xi x_eta y_eta
DO l=1,2
FORALL(i=1:i0,j=0:jf)xxi(l ,i,j)=(x(1,i+1,j)-x(1,i-1,j))/2./dxi !x_xi, y_xi
FORALL(i=0:if,j=1:j0)xxi(l+2,i,j)=(x(1,i,j+1)-x(1,i,j-1))/2./det !x_eta, y_eta
FORALL(j=0:jf)xxi(l ,0,j)=-(3.*x(1,0,j)-4.*x(1,1,j)+x(1,2,j))/2./dxi
FORALL(j=0:jf)xxi(l ,if,j)=(3.*x(1,if,j)-4.*x(1,i0,j)+x(1,i0-1,j))/2./dxi

```

```

FORALL(i=0:if)xxi(1+2,i, 0)=-(3.*x(1,i, 0)-4.*x(1,i, 1)+x(1,i, 2))/2./det
FORALL(i=0:if)xxi(1+2,i,jf)= (3.*x(1,i,jf)-4.*x(1,i,j0)+x(1,i,j0-1))/2./det
ENDDO
! ***** Compute J J^2; xi_x eta_x xi_y eta_y; g_11 g_12 g_22
FORALL(i=0:if,j=0:jf)
  aJ(i,j)=xxi(1,i,j)*xxi(4,i,j)-xxi(3,i,j)*xxi(2,i,j) !J
  g(1,i,j)=(xxi(1,i,j)*xxi(1,i,j)+xxi(2,i,j)*xxi(2,i,j))/aJ(i,j) !g_11/J
  g(2,i,j)=(xxi(1,i,j)*xxi(3,i,j)+xxi(2,i,j)*xxi(4,i,j))/aJ(i,j) !g_12/J
  g(3,i,j)=(xxi(3,i,j)*xxi(3,i,j)+xxi(4,i,j)*xxi(4,i,j))/aJ(i,j) !g_22/J
ENDFORALL
! ***** Compute control functions P and Q
! P: control i1-line and i2-line, Q: control grid spacing of j-direction
FORALL(i=0:if,j=0:jf)P(i,j)=0.
FORALL(i=0:if,j=0:jf)Q(i,j)=0.
FORALL(i=1:i1-1,j=1:jf)P(i,j)=2./(x(1,i+1,0)-x(1,i-1,0)) & !xi_xx
  *(1./(x(1,i+1,0)-x(1,i,0))-1./(x(1,i,0)-x(1,i-1,0)))
FORALL(i=i1+1:i2-1,j=1:15)P(i,j)=(15-j)/15.*2./(x(2,i+1,0)-x(2,i-1,0)) & !c*xi_yy
  *(1./(x(2,i+1,0)-x(2,i,0))-1./(x(2,i,0)-x(2,i-1,0)))
FORALL(i=i2+1:i0,j=1:10) P(i,j)=(10-j)/10.*2./(x(1,i+1,0)-x(1,i-1,0)) & !c*xi_xx
  *(1./(x(1,i+1,0)-x(1,i,0))-1./(x(1,i,0)-x(1,i-1,0)))
FORALL(j=1:jf)P(i1,j)=(P(i1-1,j)+P(i1+1,j))/2.
FORALL(j=1:jf)P(i2,j)=(P(i2-1,j)+P(i2+1,j))/2.
DO i=0,if; DO j=1,jf
  coef=25./(x(2,i,jf)-x(2,i,0))/(x(2,i,jf)-x(2,i,0)) !coef=(d eta/d te)*(d ty/dy)^2
  ty=(x(2,i,j)-x(2,i,0))/(x(2,i,jf)-x(2,i,0)) !ty=(y-y_0)/(y_f-y_0)
! Q(i,j)=coef*(6.*ty-4.) !te(ty)=ty^3-2ty^2+2ty te'(0,.5,1)=(12/6 4.5/6 6/6)
  Q(i,j)=coef*(8.*ty-5.) !te(ty)=(8ty^3-15ty^2+13ty)/6 ty' =(13/6 4/6 7/6)
! Q(i,j)=coef*(10.*ty-6.) !te(ty)=(5ty^3-9ty^2+7ty)/3 te' =(14/6 3.5/6 8/6)
! Q(i,j)=coef*(12.*ty-7.) !te(ty)=(4ty^3-7ty^2+5ty)/2 te' =(15/6 3/6 9/6)
! Q(i,j)=coef*(18.*ty-10.) !te(ty)=3ty^3-5ty^2+3ty te' =(18/6 1.5/6 12/6)
! for all te(ty), te(0, .5, 1.0)=(0, .625, 1.0),
! Q=d^2eta/dy^2=(d ty/dy)^2*te''(ty)*(d eta/d te)=coef*te''(ty)
! Lower Q's correspond to grids whose spaces are larger difference.
  IF(i>i1.AND.i<i1+10.AND.j<10) & !modify grid spaces near corner
    Q(i,j)=Q(i,j)-coef*12.*(1.-FLOAT(j)/10.)*(1.-ABS(i1+5-i)/5.)
ENDDO; ENDDO
! ***** Solve difference eqs of L(x)=-J(x_xiP+x_etaQ), L(y)=-J(y_xiP+y_etaQ) by SOR
FORALL(l=1:2,i=0:if,j=0:jf)x0(l,i,j)=x(1,i,j)
nn=0; 110 nn=nn+1
DO 10 l=1,2; DO 10 i=1,i0; DO 10 j=1,j0
  w=(g(3,i,j)*(x(1,i-1,j)+x(1,i+1,j)) &
    -g(2,i,j)*(x(1,i-1,j-1)-x(1,i-1,j+1)-x(1,i+1,j-1)+x(1,i+1,j+1))/2. &
    +g(1,i,j)*(x(1,i,j-1)+x(1,i,j+1)) &
    +aJ(i,j)*(xxi(1,i,j)*P(i,j)+xxi(1+2,i,j)*Q(i,j)))/2./(g(1,i,j)+g(3,i,j))
  10 x(1,i,j)=x(1,i,j)+al1*(w-x(1,i,j))
j=jf; DO 11 i=1,i0 !top wall
  w=(g(3,i,j)*(x(1,i-1,j)+x(1,i+1,j))+g(1,i,j)*2.*x(1,i,j-1) &
    +aJ(i,j)*(xxi(1,i,j)*P(i,j)+xxi(3,i,j)*Q(i,j)))/2./(g(1,i,j)+g(3,i,j))
  11 x(1,i,j)=x(1,i,j)+al1*(w-x(1,i,j))
i=0; DO 12 j=1,j0 !inlet bound
  w=(g(3,i,j)*2.*x(2,1,j)+g(1,i,j)*(x(2,i,j-1)+x(2,i,j+1)) &
    +aJ(i,j)*(xxi(2,i,j)*P(i,j)+xxi(4,i,j)*Q(i,j)))/2./(g(1,i,j)+g(3,i,j))
  12 x(2,i,j)=x(2,i,j)+al1*(w-x(2,i,j))
i=if; DO 13 j=1,j0 !outlet bound
  w=(g(3,i,j)*2.*x(2,i-1,j)+g(1,i,j)*(x(2,i,j-1)+x(2,i,j+1)) &
    +aJ(i,j)*(xxi(2,i,j)*P(i,j)+xxi(4,i,j)*Q(i,j)))/2./(g(1,i,j)+g(3,i,j))
  13 x(2,i,j)=x(2,i,j)+al1*(w-x(2,i,j))

```

```

IF(nn<10)                                GOTO 110
! ***** Decide convergence of x,y
adx=0.
DO i=0,if; DO j=1,j0
  adx=AMAX1(adx,ABS(x(1,i,j)-x0(1,i,j))+ABS(x(2,i,j)-x0(2,i,j)))  !|dx|+|dy|
ENDDO; ENDDO
IF(n==1)WRITE(20,60)
IF(MOD(n,10)==0)WRITE(20,61) n,adx
60 FORMAT(/1H 3X 'n', 7X 'adx'/)
61 FORMAT(1H I5, F11.5)
IF(n<nf.AND.adx>1.E-4.AND.adx<5.)      GOTO 100
ENDSUBROUTINE GRID

! ***** Computation of metrics and metric tensors
SUBROUTINE METRIC(x)
PARAMETER(if=140,jf=25,if1=280,jf1=50)
DIMENSION x(2,0:if,0:jf),x1(2,0:if1,0:jf1),xxi(4,0:if1,0:jf1),aJ(0:if1,0:jf1), &
  gU(3,0:if,0:jf),gV(3,0:if,0:jf),xix(4,0:if1,0:jf1),dxix(0:if,0:jf,2,4), &
  w1(0:if),w2(0:jf),w11(0:if1),w12(0:if1),w21(0:jf1),w22(0:jf1)
COMMON //na,Re,dt,i0,j0,lowRe,steady,unsteady
COMMON /METRIC1/x1 /METRIC2/aJ /METRIC3/gU,gV /METRIC4/xix /METRIC5/dxix
! ***** Determine x1
DO l=1,2; DO j=0,jf
  FORALL(i=0:if)w1(i)=x(1,i,j); CALL INTP(w1,w11,if)
  FORALL(i=0:if1)x1(1,i,2*j)=w11(i)
ENDDO; ENDDO
x1(1,69,0)=(-x(1,33,0)+6.*x(1,34,0)+3.*x(1,35,0))/8.; x1(2,69,0)=0.
x1(1,71,0)=0.; x1(2,71,0)=(3.*x(2,35,0)+6.*x(2,36,0)-x(2,37,0))/8.
x1(1,79,0)=0.; x1(2,79,0)=(-x(2,38,0)+6.*x(2,39,0)+3.*x(2,40,0))/8.
x1(1,81,0)=(3.*x(1,40,0)+6.*x(1,41,0)-x(1,42,0))/8.; x1(2,81,0)=-1.
DO l=1,2; DO i=0,if1
  FORALL(j=0:jf)w2(j)=x1(1,i,2*j); CALL INTP(w2,w21,jf)
  FORALL(j=0:jf1)x1(1,i,j)=w21(j)
ENDDO; ENDDO
! ***** Determine xxi, J
DO l=1,2; DO j=0,jf1
  FORALL(i=0:if1)w11(i)=x1(1,i,j); CALL DIFX(w11,w12,if1,.5)
  FORALL(i=0:if1)xxi(1,i,j)=w12(i)                                !x_xi y_xi at point X
ENDDO; ENDDO
DO l=1,2; DO i=0,if1
  FORALL(j=0:jf1)w21(j)=x1(1,i,j); CALL DIFX(w21,w22,jf1,.5)
  FORALL(j=0:jf1)xxi(1+2,i,j)=w22(j)                            !x_eta y_eta at X
ENDDO; ENDDO
FORALL(i=0:if1,j=0:jf1)aJ(i,j)=xxi(1,i,j)*xxi(4,i,j)-xxi(3,i,j)*xxi(2,i,j)  !Jacobian J
! ***** Determine g_11/J g_12/J g_22/J at U and V
DO i=0,if; ii=2*i; DO j=0,j0; jj=2*j+1
  gU(1,i,j)=(xxi(1,ii,jj)*xxi(1,ii,jj)+xxi(2,ii,jj)*xxi(2,ii,jj))/aJ(ii,jj) !g_11/J at U
  gU(2,i,j)=(xxi(1,ii,jj)*xxi(3,ii,jj)+xxi(2,ii,jj)*xxi(4,ii,jj))/aJ(ii,jj) !g_12/J at U
  gU(3,i,j)=(xxi(3,ii,jj)*xxi(3,ii,jj)+xxi(4,ii,jj)*xxi(4,ii,jj))/aJ(ii,jj) !g_22/J at U
ENDDO; ENDDO
DO i=0,i0; ii=2*i+1; DO j=0,jf; jj=2*j
  gV(1,i,j)=(xxi(1,ii,jj)*xxi(1,ii,jj)+xxi(2,ii,jj)*xxi(2,ii,jj))/aJ(ii,jj) !g_11/J at V
  gV(2,i,j)=(xxi(1,ii,jj)*xxi(3,ii,jj)+xxi(2,ii,jj)*xxi(4,ii,jj))/aJ(ii,jj) !g_12/J at V
  gV(3,i,j)=(xxi(3,ii,jj)*xxi(3,ii,jj)+xxi(4,ii,jj)*xxi(4,ii,jj))/aJ(ii,jj) !g_22/J at V
ENDDO; ENDDO
! ***** Determine xix
DO i=0,if1; DO j=0,jf1

```

```

xix(1,i,j)= xxi(4,i,j)/aJ(i,j); xix(2,i,j)=-xxi(3,i,j)/aJ(i,j)      !xi_x, xi_y at X
xix(3,i,j)=-xxi(2,i,j)/aJ(i,j); xix(4,i,j)= xxi(1,i,j)/aJ(i,j)      !eta_x, eta_y at X
ENDDO; ENDDO
! ***** Determine (xi_x)_xi
DO j=0,j0; jj=2*j+1
DO i=1,i0; ii=2*i
dxix(i,j,1,1)=xix(1,ii+1,jj)-xix(1,ii-1,jj)      !(xi_x)_xi at U
dxix(i,j,1,3)=xix(2,ii+1,jj)-xix(2,ii-1,jj)      !(xi_y)_xi at U
ENDDO
dxix(if,j,1,1)=3.*xix(1,if1,jj)-4.*xix(1,if1-1,jj)+xix(1,if1-2,jj)
dxix(if,j,1,3)=3.*xix(2,if1,jj)-4.*xix(2,if1-1,jj)+xix(2,if1-2,jj)
DO i=1,if; ii=2*i
dxix(i,j,1,2)=xix(1,ii,jj+1)-xix(1,ii,jj-1)      !(xi_x)_eta at U
dxix(i,j,1,4)=xix(2,ii,jj+1)-xix(2,ii,jj-1)      !(xi_y)_eta at U
ENDDO; ENDDO
DO i=0,i0; ii=2*i+1; DO j=1,j0; jj=2*j
dxix(i,j,2,1)=xix(3,ii+1,jj)-xix(3,ii-1,jj)      !(eta_x)_xi at V
dxix(i,j,2,3)=xix(4,ii+1,jj)-xix(4,ii-1,jj)      !(eta_y)_xi at V
dxix(i,j,2,2)=xix(3,ii,jj+1)-xix(3,ii,jj-1)      !(eat_x)_eta at V
dxix(i,j,2,4)=xix(4,ii,jj+1)-xix(4,ii,jj-1)      !(eta_y)_eta at V
ENDDO; ENDDO
ENDSUBROUTINE METRIC

! ***** Prediction of volume flux and pressure
SUBROUTINE PREDCT(x,UJ,u,p)
PARAMETER(if=140,jf=25,if1=280,jf1=50)
DIMENSION x(2,0:if,0:jf),UJ(2,0:if,0:jf),u(2,0:if,0:jf),p(0:if,0:jf),x1(2,0:if1,0:jf1), &
aJ(0:if1,0:jf1),xix(4,0:if1,0:jf1)
COMMON //na,Re,dt,i0,j0,lowRe,steady,unsteady
COMMON /METRIC1/x1 /METRIC2/aJ /METRIC4/xix
! ***** Set up starting values of u, U and p
DO j=0,jf; ty=x1(2,0,2*j)/x1(2,0,jf1)
u(1,0,j)=6.*ty*(1.-ty)      !mean velocity at inlet=1
ENDDO
DO j=0,j0; ty=x1(2,0,2*j+1)/x1(2,0,jf1)
UJ(1,0,j)=aJ(0,2*j+1)*xix(1,0,2*j+1)*6.*ty*(1.-ty)
FORALL(i=1:if)UJ(1,i,j)=UJ(1,0,j)
ENDDO
Dp=1./2.-.75*.75/2.      !recovery pressure in step duct
p0=(4./3.*10.+9./16.*70.9)/Re-Dp      !inlet pressure
FORALL(i=0:35,j=0:j0)p(i,j)=p0+4./3.*(x(1,0,j)-x1(1,2*i+1,2*j+1))/Re
FORALL(i=100:i0,j=0:j0)p(i,j)=9./16.*(x(1,if,j)-x1(1,2*i+1,2*j+1))/Re
FORALL(i=36:99,j=0:j0)p(i,j)=(p(100,j)*(x1(1,2*i+1,2*j+1)-x1(1,71,2*j+1)) &
+p(35,j)*(x1(1,201,2*j+1)-x1(1,2*i+1,2*j+1)))/(x1(1,201,2*j+1)-x1(1,71,2*j+1))
ENDSUBROUTINE PREDCT

! ***** Compute JU^* and JV^* by delta-form implicit method using Chakravarthy-Osher
! type TVD scheme
SUBROUTINE COMPU1(ma,UJ,u,p,z,locf,fmax)
PARAMETER(if=140,jf=25,if1=280,jf1=50)
DIMENSION UJ(2,0:if,0:jf),Uq(2,0:if,0:jf),u(2,0:if,0:jf),p(0:if,0:jf),z(0:if,0:jf), &
aJ(0:if1,0:jf1),gU(3,0:if,0:jf),gV(3,0:if,0:jf),xix(4,0:if1,0:jf1),dxix(0:if,0:jf,2,4), &
UJX(2,0:if,0:jf),hf(0:if),rhs(2,0:if,0:jf),rhs0(2,0:if,0:jf),UJ0(2,0:if,0:jf), &
a(140,140,3),b(140,140),c(0:4,0:if,0:jf),f(0:if,0:jf),w(0:if),w1(0:if),w11(0:if1), &
w12(0:if1),w13(0:if1),w14(0:if),w2(0:jf),w21(0:jf1),w22(0:jf1),w23(0:jf1),w24(0:jf), &
lo(2),locf(2,8),fmax(8)
COMMON //na,Re,dt,i0,j0,lowRe,steady,unsteady

```

```

COMMON /METRIC2/aJ /METRIC3/gU,gV /METRIC4/xix /METRIC5/dxix /COMPUZ1/UJX
LOGICAL lowRe,steady,unsteady
DATA theta,beta/1.,.5/                                !theta:trapezoidal, beta:damping
FORALL(1=1:2,i=0:if,j=0:jf)rhs(1,i,j)=0.
IF(steady.AND.lowRe)THEN                               !for training
! ***** Compute convection term using central-differences
DO j=0,j0
  FORALL(i=0:if)w1(i)=UJ(1,i,j); CALL INTP(w1,w11,if)   !w11=JU_P(JU at poin P)
  FORALL(i=0:i0)hf(i)=w11(2*i+1)*w11(2*i+1)/aJ(2*i+1,2*j+1) !hf=JU_P
  FORALL(i=1:i0)Uq(1,i,j)=hf(i)-hf(i-1)              !JUU_xi
ENDDO
DO i=1,i0
  FORALL(j=0:jf)hf(j)=UJX(2,i,j)*UJX(1,i,j)/aJ(2*i,2*j) !hf=JVU_X
  FORALL(j=0:j0)Uq(1,i,j)=Uq(1,i,j)+hf(j+1)-hf(j)     ! +JVU_eta
ENDDO
DO j=1,j0
  FORALL(i=1:i0)hf(i)=UJX(1,i,j)*UJX(2,i,j)/aJ(2*i,2*j) !hf=JUV_X
  FORALL(i=1:i0-1)Uq(2,i,j)=hf(i+1)-hf(i)             ! JUV_xi
ENDDO
DO i=1,i0-1
  FORALL(j=0:jf)w2(j)=UJ(2,i,j); CALL INTP(w2,w21,jf)   !w21=JV_P
  FORALL(j=0:j0)hf(j)=w21(2*j+1)*w21(2*j+1)/aJ(2*i+1,2*j+1) !hf=JVV_P
  FORALL(j=1:j0)Uq(2,i,j)=Uq(2,i,j)+hf(j)-hf(j-1)     ! +JVV_eta
ENDDO
ELSE
! ***** Compute convection term using Chakravarthy-Osher TVD scheme
DO j=0,j0
  FORALL(i=0:if)w(i)=UJ(1,i,j)/aJ(2*i,2*j+1)           !w =U
  FORALL(i=0:if)w1(i)=UJ(1,i,j); CALL INTP(w1,w11,if)   !w11=JU_P
  i=0 ; hf(i)=w11(2*i+1)*(w(i)+w(i+1))/2.             !near inlet
  i=if-1; hf(i)=w11(2*i+1)*(w(i)+w(i+1))/2.          !near outlet
  cycle_1: DO i=1,i0; UJP=w11(2*i+1)
    m=1; IF(UJP>0.)m=-1; ip=i+m
    IF(i==i0.AND.m==1)CYCLE cycle_1
    UP=(w(i+1)+w(i))/2.; dU=w(i+1)-w(i); dU1=w(ip+1)-w(ip)
    hf(i)=UJP*UP+ABS(UJP)*(-dU/2.+AMINMOD(dU1,4.*dU)/6.+AMINMOD(dU,4.*dU1)/3.) !hf=JU_P
  ENDDO cycle_1
  FORALL(i=1:i0)Uq(1,i,j)=hf(i)-hf(i-1)              ! JUU_xi
ENDDO
! ***** Correct convection term just behind corner
FORALL(i=35:37)w(i)=UJ(1,i,0)*UJ(1,i,0)/aJ(2*i,1)
dum=w(36)-w(35); du=w(37)-w(36)
IF(UJ(1,36,0)<=0.)THEN; Uq(1,36,0)=AMINMOD((dum/du+1.)/2.,2.)*du
ELSE; Uq(1,36,0)=AMINMOD((du/dum+1.)/2.,2.)*dum; ENDF
DO i=1,i0
  FORALL(j=0:j0)w(j)=UJ(1,i,j)/aJ(2*i,2*j+1)           !w=U
  j=0 ; hf(j)=0.                                       !bottom
  j=1 ; hf(j)=UJX(2,i,j)*(3.*w(j-1)+2.*w(j)-w(j+1))/5. !neighbor of bottom wall
  j=j0; hf(j)=UJX(2,i,j)*(3.*w(j)+2.*w(j-1)-w(j-2))/5. !neighbor top
  j=jf; hf(j)=0.                                       !top
  cycle_2: DO j=1,j0; VJX=UJX(2,i,j)
    m=1; IF(VJX>0.)m=-1; jp=j+m
    IF(j==1.AND.m==-1 .OR. j==j0.AND.m==1)CYCLE cycle_2
    UX=(w(j-1)+w(j))/2.; dU=w(j)-w(j-1); dU1=w(jp)-w(jp-1)
    hf(j)=VJX*UX+ABS(VJX)*(-dU/2.+AMINMOD(dU1,4.*dU)/6.+AMINMOD(dU,4.*dU1)/3.) !hf=JVU_X
  ENDDO cycle_2
  FORALL(j=0:j0)Uq(1,i,j)=Uq(1,i,j)+hf(j+1)-hf(j)     ! +JVU_eta

```

```

ENDDO
DO j=1,j0
  FORALL(i=0:i0)w(i)=UJ(2,i,j)/aJ(2*i+1,2*j)           !w=V
  i=0 ; hf(i)=0.
  i=1 ; hf(i)=UJX(1,1,j)*(w(i-1)+w(i))/2.             !neighbor inlet
  i=i0; hf(i)=UJX(1,i,j)*(-w(i-2)+6.*w(i-1)+3.*w(i))/8. !neighbor outlet
  cycle_3: DO i=2,i0; UJX0=UJX(1,i,j)
    m=1; IF(UJX0>0.)m=-1; ip=i+m
    IF(i==i0.AND.m==1)CYCLE cycle_3
    VX=(w(i-1)+w(i))/2.; dV=w(i)-w(i-1); dV1=w(ip)-w(ip-1)
    hf(i)=UJX0*VX+ABS(UJX0)*(-dV/2.+AMINMOD(dV1,4.*dV)/6.+AMINMOD(dV,4.*dV1)/3.) !hf=JUV_X
  ENDDO cycle_3
  FORALL(i=0:i0-1)Uq(2,i,j)=hf(i+1)-hf(i)             !JUV_xi
ENDDO
DO i=0,i0-1
  FORALL(j=0:jf)w(j)=UJ(2,i,j)/aJ(2*i+1,2*j)         !w =V
  FORALL(j=0:jf)w2(j)=UJ(2,i,j); CALL INTVP(w2,w21,jf) !w21=JV_P
  j=0 ; hf(j)=w21(2*j+1)*(12.*w(1)-w(2))/32.         !near bottom
  j=j0; hf(j)=w21(2*j+1)*(12.*w(j0)-w(j0-1))/32.     !near top
  cycle_4: DO j=0,j0; VJP=w21(2*j+1)
    m=1; IF(VJP>0.)m=-1; jp=j+m
    IF(j==0.AND.m==1 .OR. j==j0.AND.m==1)CYCLE cycle_4
    VP=(w(j)+w(j+1))/2.; dV=w(j+1)-w(j); dV1=w(jp+1)-w(ip)
    hf(j)=VJP*VP+ABS(VJP)*(-dV/2.+AMINMOD(dV1,4.*dV)/6.+AMINMOD(dV,4.*dV1)/3.) !hf=JVV_P
  ENDDO cycle_4
  FORALL(j=1:j0)Uq(2,i,j)=Uq(2,i,j)+hf(j)-hf(j-1)    ! +JVV_eta
ENDDO
ENDIF
! ***** Compute convection term at outlet using upwind-difference scheme
DO j=0,j0; i=if; Uq(1,i,j)=0.
  DO k=1,2
    Uq(1,i,j) = Uq(1,i,j)+xix(k,2*i,2*j+1)           &
      *(UJ(1,i,j)*(u(k,i,j)+u(k,i,j+1)-u(k,i-1,j)-u(k,i-1,j+1))/2. &
      +(UJX(2,i,j)+UJX(2,i,j+1))/2.*(u(k,i,j+1)-u(k,i,j)))
  ENDDO; ENDDO
DO j=1,j0; i=i0; Uq(2,i,j)=0.
  DO k=1,2
    Uq(2,i,j) = Uq(2,i,j)+xix(k+2,2*i+1,2*j)         &
      *((UJX(1,i,j)+UJX(1,i+1,j))/2.*(u(k,i+1,j)-u(k,i-1,j))/2. &
      +UJ(2,i,j)*(u(k,i,j+1)+u(k,i+1,j+1)-u(k,i,j-1)-u(k,i+1,j-1))/4.)
  ENDDO; ENDDO
! ***** Compute additional, pressure and diffusion terms
DO i=1,if
  FORALL(j=0:jf)w2(j)=u(1,i,j); CALL INTP(w2,w21,jf)   !w21=u_U
  FORALL(j=0:jf)w2(j)=u(2,i,j); CALL INTP(w2,w22,jf)   !w22=v_U
  FORALL(j=0:jf)w2(j)=UJX(2,i,j); CALL INTP(w2,w23,jf) !w23=JV_U
  FORALL(j=0:j0)w2(j)=(p(i-1,j)+p(i,j))/2.; CALL DIFP(w2,w24,jf,1.) !w24=(p_eta)_U
  DO j=0,j0; jj=2*j+1
    ad = w21(jj)*(UJ(1,i,j)*dxix(i,j,1,1)+w23(jj)*dxix(i,j,1,2)) & !additional term
      +w22(jj)*(UJ(1,i,j)*dxix(i,j,1,3)+w23(jj)*dxix(i,j,1,4))
    IF(i==if)ad = 0.
    pr = gU(3,i,j)*(p(i,j)-p(i-1,j))-gU(2,i,j)*w24(j) !pressure term
    Uq(1,i,j) = Uq(1,i,j)-ad+pr+(z(i,j+1)-z(i,j))/Re !residuals of NS eqn
    rhs(1,i,j) = -dt*Uq(1,i,j) !rhs of linear eqns
  ENDDO; ENDDO
DO j=1,j0
  FORALL(i=0:if)w1(i)=u(1,i,j); CALL INTP(w1,w11,if)   !w11=u_V

```

```

FORALL(i=0:if)w1(i)=u(2,i,j); CALL INTP(w1,w12,if) !w12=v_V
FORALL(i=0:if)w1(i)=UJX(1,i,j); CALL INTP(w1,w13,if) !w13=JU_V
FORALL(i=0:if)w1(i)=(p(i,j-1)+p(i,j))/2.; CALL DIFP(w1,w14,if,1.) !w14=(p_xi)_V
i=i0; w14(i)=(w1(i+1)-w1(i-1))/2. !outlet
DO i=0,i0; ii=2*i+1
  ad = w11(ii)*(w13(ii)*dxix(i,j,2,1)+UJ(2,i,j)*dxix(i,j,2,2)) & !additional term
      +w12(ii)*(w13(ii)*dxix(i,j,2,3)+UJ(2,i,j)*dxix(i,j,2,4))
  IF(i==i0)ad = 0.
  pr = -gV(2,i,j)*w14(i)+gV(1,i,j)*(p(i,j)-p(i,j-1)) !pressure term
  Uq(2,i,j) = Uq(2,i,j)-ad+pr-(z(i+1,j)-z(i,j))/Re !residuals of NS eqn
  rhs(2,i,j) = -dt*Uq(2,i,j) !rhs of linear eqns
ENDDO; ENDDO
IF(unsteady)THEN !only unsteady
  IF(ma==1)FORALL(l=1:2,i=0:if,j=0:jf) UJ0(1,i,j)= UJ(1,i,j)
  IF(ma==1)FORALL(l=1:2,i=0:if,j=0:jf)rhs0(1,i,j)=rhs(1,i,j)
  FORALL(l=1:2,i=0:if,j=0:jf)rhs(1,i,j)=-(UJ(1,i,j)-UJ0(1,i,j))+(rhs0(1,i,j)+rhs(1,i,j))/2.
ENDIF
! Compute UJ^* by implicit SMAC-scheme
tt=dt*theta
DO i=1,if; DO j=0,j0
  UJp=(UJ(1,i,j)+ABS(UJ(1,i,j)))/2.; UJm=(UJ(1,i,j)-ABS(UJ(1,i,j)))/2.
  VJU=(UJX(2,i,j)+UJX(2,i,j+1))/2.
  VJp=(VJU+ABS(VJU))/2.; VJm=(VJU-ABS(VJU))/2.
  c(1,i,j)=-tt*(UJp+gU(3,i,j))/Re !coefs of linear eqs
  c(2,i,j)= tt*(UJm-gU(3,i,j))/Re; IF(i==if)c(2,i,j)=0.
  c(3,i,j)=-tt*(VJp+gU(1,i,j))/Re
  c(4,i,j)= tt*(VJm-gU(1,i,j))/Re
  c(0,i,j)=aJ(2*i,2*j+1)-c(1,i,j)-c(2,i,j)-c(3,i,j)-c(4,i,j)
ENDDO; ENDDO
!Compute dU^* by modifies AF-method
DO j=0,j0; l=j+1; DO i=1,if
  a(1,i,1)=c(1,i,j)
  a(1,i,2)=c(0,i,j)
  a(1,i,3)=c(2,i,j); b(1,i)=rhs(1,i,j)
ENDDO; ENDDO
CALL GAUSSZ(a,b,140,jf,if)
FORALL(j=0:j0,i=1:if)rhs(1,i,j)=b(j+1,i) !dU^**
DO j=0,j0; l=j+1; DO i=1,if
  a(i,l,1)=c(3,i,j)
  a(i,l,2)=c(0,i,j)
  a(i,l,3)=c(4,i,j); b(i,l)=c(0,i,j)*rhs(1,i,j)
ENDDO; ENDDO
CALL GAUSSZ(a,b,140,if,jf)
FORALL(j=0:j0,i=1:if)rhs(1,i,j)=b(i,j+1) !dU^*
! Compute VJ^* by implicit SMAC-scheme
DO i=0,i0; DO j=1,j0
  UJV=(UJX(1,i,j)+UJX(1,i+1,j))/2.
  UJp=(UJV +ABS(UJV) )/2.; UJm=(UJV -ABS(UJV) )/2.
  VJp=(UJ(2,i,j)+ABS(UJ(2,i,j)))/2.; VJm=(UJ(2,i,j)-ABS(UJ(2,i,j)))/2.
  c(1,i,j)=-tt*(UJp+gV(3,i,j))/Re !coefs of linear eqs
  c(2,i,j)= tt*(UJm-gV(3,i,j))/Re
  c(3,i,j)=-tt*(VJp+gV(1,i,j))/Re
  c(4,i,j)= tt*(VJm-gV(1,i,j))/Re
  c(0,i,j)=aJ(2*i+1,2*j)-c(1,i,j)-c(2,i,j)-c(3,i,j)-c(4,i,j)
ENDDO; ENDDO
!Compute dV^* by modifies AF-method
DO j=1,j0; DO i=0,i0; k=i+1

```

```

a(j,k,1)=c(1,i,j)
a(j,k,2)=c(0,i,j)
a(j,k,3)=c(2,i,j); b(j,k)=rhs(2,i,j)
ENDDO; ENDDO
CALL GAUSSZ(a,b,140,j0,if)
FORALL(j=1:j0,i=0:i0)rhs(2,i,j)=b(j,i+1) !dV^**
DO j=1,j0; DO i=0,i0; k=i+1
a(k,j,1)=c(3,i,j)
a(k,j,2)=c(0,i,j)
a(k,j,3)=c(4,i,j); b(k,j)=c(0,i,j)*rhs(2,i,j)
ENDDO; ENDDO
CALL GAUSSZ(a,b,140,if,j0)
FORALL(j=1:j0,i=0:i0)rhs(2,i,j)=b(i+1,j) !dV^*
FORALL(j=0:j0,i=1:if)UJ(1,i,j)=UJ(1,i,j)+beta*aJ(2*i,2*j+1)*rhs(1,i,j) !JU^*
FORALL(j=1:j0,i=0:i0)UJ(2,i,j)=UJ(2,i,j)+beta*aJ(2*i+1,2*j)*rhs(2,i,j) !JV^*
FORALL(i=0:if,j=0:jf)f(i,j)=Uq(1,i,j)
lo=MAXLOC(f); FORALL(k=1:2)locf(k,1)=lo(k); fmax(1)=MAXVAL(f)
lo=MINLOC(f); FORALL(k=1:2)locf(k,2)=lo(k); fmax(2)=MINVAL(f)
FORALL(i=0:if,j=0:jf)f(i,j)=Uq(2,i,j)
lo=MAXLOC(f); FORALL(k=1:2)locf(k,3)=lo(k); fmax(3)=MAXVAL(f)
lo=MINLOC(f); FORALL(k=1:2)locf(k,4)=lo(k); fmax(4)=MINVAL(f)
ENDSUBROUTINE COMPU1

! ***** Set up coefficients of difference equations for phi
SUBROUTINE COEF(co)
PARAMETER(if=140,jf=25,if1=280,jf1=50)
DIMENSION co(0:if,0:jf,-1:1,-1:1),aJ(0:if1,0:jf1),gU(3,0:if,0:jf),gV(3,0:if,0:jf), &
UJX(2,0:if,0:jf),w1(0:if,0:jf),gw(4,0:if,0:jf)
COMMON //na,Re,dt,i0,j0,lowRe,steady,unsteady
COMMON /METRIC2/aJ /METRIC3/gU,gV /COMPUZ1/UJX
! ***** Set up coefficients of difference-equations for phi
DO i=0,if; DO j=0,jf
gw(1,i,j)= dt*gU(3,i,j)
gw(2,i,j)=-dt*gU(2,i,j)/4.
gw(3,i,j)=-dt*gV(2,i,j)/4.
gw(4,i,j)= dt*gV(1,i,j)
ENDDO; ENDDO
FORALL(l=-1:1,m=-1:1,i=0:if,j=0:jf)co(i,j,l,m)=0.
DO i=0,i0; DO j=0,j0 !1st step
co(i,j,1,1)= gw(2,i+1,j)
co(i,j,0,1)= gw(2,i+1,j)
co(i,j,1,0)= gw(1,i+1,j)
co(i,j,0,0)=-gw(1,i+1,j)
co(i,j,1,-1)=-gw(2,i+1,j)
co(i,j,0,-1)=-gw(2,i+1,j)
ENDDO
co(i,0,1,1)=co(i,0,1,1)+3.*gw(2,i+1,0) !bottom
co(i,0,0,1)=co(i,0,0,1)+3.*gw(2,i+1,0)
co(i,0,1,0)=co(i,0,1,0)-3.*gw(2,i+1,0)
co(i,0,0,0)=co(i,0,0,0)-3.*gw(2,i+1,0)
co(i,j0,1,0)=co(i,j0,1,0)+3.*gw(2,i+1,j0) !top
co(i,j0,0,0)=co(i,j0,0,0)+3.*gw(2,i+1,j0)
co(i,j0,1,-1)=co(i,j0,1,-1)-3.*gw(2,i+1,j0)
co(i,j0,0,-1)=co(i,j0,0,-1)-3.*gw(2,i+1,j0)
ENDDO
DO i=1,i0; DO j=0,j0 !2nd step
co(i,j,0,1)=co(i,j,0,1)-gw(2,i,j)

```

```

co(i,j,-1,1)=co(i,j,-1,1)-gw(2,i,j)
co(i,j,0,0)=co(i,j,0,0)-gw(1,i,j)
co(i,j,-1,0)=co(i,j,-1,0)+gw(1,i,j)
co(i,j,0,-1)=co(i,j,0,-1)+gw(2,i,j)
co(i,j,-1,-1)=co(i,j,-1,-1)+gw(2,i,j)
ENDDO
co(i,0,0,1)=co(i,0,0,1)-3.*gw(2,i,0)           !bottom
co(i,0,-1,1)=co(i,0,-1,1)-3.*gw(2,i,0)
co(i,0,0,0)=co(i,0,0,0)+3.*gw(2,i,0)
co(i,0,-1,0)=co(i,0,-1,0)+3.*gw(2,i,0)
co(i,j0,0,0)=co(i,j0,0,0)-3.*gw(2,i,j0)       !top
co(i,j0,-1,0)=co(i,j0,-1,0)-3.*gw(2,i,j0)
co(i,j0,0,-1)=co(i,j0,0,-1)+3.*gw(2,i,j0)
co(i,j0,-1,-1)=co(i,j0,-1,-1)+3.*gw(2,i,j0)
ENDDO
DO j=0,j0-1; DO i=0,i0                           !3rd step
co(i,j,1,1)=co(i,j,1,1)+gw(3,i,j+1)
co(i,j,1,0)=co(i,j,1,0)+gw(3,i,j+1)
co(i,j,0,1)=co(i,j,0,1)+gw(4,i,j+1)
co(i,j,0,0)=co(i,j,0,0)-gw(4,i,j+1)
co(i,j,-1,1)=co(i,j,-1,1)-gw(3,i,j+1)
co(i,j,-1,0)=co(i,j,-1,0)-gw(3,i,j+1)
ENDDO
co(0,j,1,1)=co(0,j,1,1)+3.*gw(3,0,j+1)       !inlet
co(0,j,1,0)=co(0,j,1,0)+3.*gw(3,0,j+1)
co(0,j,0,1)=co(0,j,0,1)-3.*gw(3,0,j+1)
co(0,j,0,0)=co(0,j,0,0)-3.*gw(3,0,j+1)
ENDDO
DO j=1,j0; DO i=0,i0                             !4th step
co(i,j,1,0)=co(i,j,1,0)-gw(3,i,j)
co(i,j,1,-1)=co(i,j,1,-1)-gw(3,i,j)
co(i,j,0,0)=co(i,j,0,0)-gw(4,i,j)
co(i,j,0,-1)=co(i,j,0,-1)+gw(4,i,j)
co(i,j,-1,0)=co(i,j,-1,0)+gw(3,i,j)
co(i,j,-1,-1)=co(i,j,-1,-1)+gw(3,i,j)
ENDDO
co(0,j,1,0)=co(0,j,1,0)-3.*gw(3,0,j)         !inlet
co(0,j,1,-1)=co(0,j,1,-1)-3.*gw(3,0,j)
co(0,j,0,0)=co(0,j,0,0)+3.*gw(3,0,j)
co(0,j,0,-1)=co(0,j,0,-1)+3.*gw(3,0,j)
ENDDO
ENDSUBROUTINE COEF

! ***** Compute static pressure p
SUBROUTINE COMPP(UJ,p,phi,co,resP,locf,fmax)
PARAMETER(if=140,jf=25,ife=70,jfe=13)          !ife=(if+1)/2,jfe=(jf+1)/2
DIMENSION UJ(2,0:if,0:jf),p(0:if,0:jf),phi(0:if,0:jf),co(0:if,0:jf,-1:1,-1:1), &
          rhs(0:if,0:jf),a(ife,0:i0,3),b(ife,0:i0),r(ife),lo(2),locf(2,8),fmax(8)
COMMON //na,Re,dt,i0,j0,lowRe,steady,unsteady
REAL(8) a,b
LOGICAL lowRe,steady,unsteady
! Determine rhs of pressure difference eqn
FORALL(i=0:if,j=0:jf)phi(i,j)=0.
FORALL(i=0:i0,j=0:j0)rhs(i,j)=(UJ(1,i+1,j)-UJ(1,i,j)+UJ(2,i,j+1)-UJ(2,i,j)) !UJ=JU*
! IF(steady.AND.lowRe)THEN
IF(steady)THEN
! ***** Solve pressure difference eqn by SOR method

```

```

alp=1.5; bet=0.5                                !alp:over-relaxation, bet:damping
nf=4
n=0; 100 n=n+1; resP=0.
ib=0; ie=i0; id=1; jb=0; je=j0; jd=1           !odd sweep, throughout region
IF(MOD(n,2)==0)THEN
  ib=i0-1; ie=1; id=-1; jb=j0-1; je=1; jd=-1   !even sweep, inside region
ENDIF
DO i=ib,ie,id; DO j=jb,je,jd
  res=-rhs(i,j)
  DO ip=-1,1; ic=0; IF(ip==-1)ic=3
  DO jp=-1,1; jc=0; IF(jp==-1)jc=3; IF(jp==if)jc=-3
  res=res+co(i,j,ip,jp)*phi(i+ip+ic,j+jp+jc)    !residuals
ENDDO; ENDDO
  phi(i,j)=phi(i,j)-alp*res/co(i,j,0)           !over-relaxation of phi
  resP=AMAX1(resP,ABS(res))                     !max residual
ENDDO; ENDDO
IF(resP>1.E-5 .AND. n<nf)                       GOTO 100    !dicide convergence
ELSE
20 CONTINUE
! ***** Solve pressure difference eqn by Tschebyscheff SLOR method
alp=1.25; bet=0.75
ifo=if-ife; jfo=jf-jfe; nf=4
n=0; 110 n=n+1; resP=0.
! ***** Compute even i-columns
DO j=0,j0
  DO ii=1,ife; i=2*ii-2
    a(ii,j,1)=co(i,j,0,-1); a(ii,j,2)=co(i,j,0, 0)*alp; a(ii,j,3)=co(i,j,0, 1)
    b(ii,j) =rhs(i,j)-(1.-alp)*co(i,j,0,0)*phi(i,j)
    r(ii)   =b(ii,j)
  ENDDO
  DO jp=-1,1; jc=0; IF(jp==-1)jc=3; IF(jp==jf)jc=-3
  DO ii=1,ife; i=2*ii-2
    im1=i-1; IF(im1==-1)im1=2
    cop=co(i,j,-1,jp)*phi(im1,j+jp+jc)+co(i,j,1,jp)*phi(i+1,j+jp+jc)
    b(ii,j)=b(ii,j)-cop
    r(ii)=r(ii)-cop-a(ii,j,jp+2)*phi(i,j+jp+jc)    !residuals
  ENDDO; ENDDO
  DO ii=1,ife
    resP=AMAX1(resP,ABS(r(ii)))                   !max residual
  ENDDO
ENDDO
CALL GAUSSP(a,b,ife,i0,j0,ife)
FORALL(j=0:j0,ii=1:ife)phi(2*ii-2,j)=b(ii,j)     !even column phi
! ***** Compute odd i-columns
DO j=0,j0
  DO ii=1,ifo; i=2*ii-1
    a(ii,j,1)=co(i,j,0,-1); a(ii,j,2)=co(i,j,0, 0)*alp; a(ii,j,3)=co(i,j,0, 1)
    b(ii,j) =rhs(i,j)-(1.-alp)*co(i,j,0,0)*phi(i,j)
    r(ii)   =b(ii,j)
  ENDDO
  DO jp=-1,1; jc=0; IF(jp==-1)jc=3; IF(jp==jf)jc=-3
  DO ii=1,ifo; i=2*ii-1
    cop=co(i,j,-1,jp)*phi(i-1,j+jp+jc)+co(i,j,1,jp)*phi(i+1,j+jp+jc)
    b(ii,j)=b(ii,j)-cop
    r(ii)=r(ii)-cop-a(ii,j,jp+2)*phi(i,j+jp+jc)    !residuals
  ENDDO; ENDDO
  DO ii=1,ifo

```

```

        resP=AMAX1(resP,ABS(r(ii)))           !max residual
    ENDDO
ENDDO
CALL GAUSSP(a,b,ife,i0,j0,ifo)
FORALL(ii=1:ifo,j=0:j0)phi(2*ii-1,j)=b(ii,j)   !odd column phi
! Compute even j-rows
n=n+1; resP=0.
DO i=0,i0
    DO jj=1,jfe; j=2*jj-2
        a(jj,i,1)=co(i,j,-1,0); a(jj,i,2)=co(i,j,0,0)*alp; a(jj,i,3)=co(i,j,1,0)
        b(jj,i) =rhs(i,j)-(1.-alp)*co(i,j,0,0)*phi(i,j)
        r(jj)   =b(jj,i)
    ENDDO
    DO ip=-1,1; ic=0; IF(ip==-1)ic=3
    DO jj=1,jfe; j=2*jj-2
        jm1=j-1; IF(jm1==-1)jm1=2
        jp1=j+1; IF(jp1==j)jp1=j0-2
        cop=co(i,j,ip,-1)*phi(i+ip+ic,jm1)+co(i,j,ip,1)*phi(i+ip+ic,jp1)
        b(jj,i)=b(jj,i)-cop
        r(jj)=r(jj)-cop-a(jj,i,ip+2)*phi(i+ip+ic,j)   !residuals
    ENDDO; ENDDO
    DO jj=1,jfe
        resP=AMAX1(resP,ABS(r(jj)))           !max residual
    ENDDO
ENDDO
CALL GAUSSP(a,b,ife,i0,i0,jfe)
FORALL(i=0:i0,jj=1:jfe)phi(i,2*jj-2)=b(jj,i)   !even row phi
! ***** Compute odd j-rows
DO i=0,i0
    DO jj=1,jfo; j=2*jj-1
        a(jj,i,1)=co(i,j,-1,0); a(jj,i,2)=co(i,j,0,0)*alp; a(jj,i,3)=co(i,j,1,0)
        b(jj,i) =rhs(i,j)-(1.-alp)*co(i,j,0,0)*phi(i,j)
        r(jj)   =b(jj,i)
    ENDDO
    DO ip=-1,1; ic=0; IF(ip==-1)ic=3
    DO jj=1,jfo; j=2*jj-1
        jm1=j-1; IF(jm1==-1)jm1=2
        jp1=j+1; IF(jp1==j)jp1=j0-2
        cop=co(i,j,ip,-1)*phi(i+ip+ic,jm1)+co(i,j,ip,1)*phi(i+ip+ic,jp1)
        b(jj,i)=b(jj,i)-cop
        r(jj)=r(jj)-cop-a(jj,i,ip+2)*phi(i+ip+ic,j)   !residuals
    ENDDO; ENDDO
    DO jj=1,jfo
        resP=AMAX1(resP,ABS(r(jj)))           !max residual
    ENDDO
ENDDO
CALL GAUSSP(a,b,ife,i0,i0,jfo)
FORALL(jj=1:jfe,i=0:i0)phi(i,2*jj-1)=b(jj,i)   !odd row phi
IF(resP>1.E-5 .AND. n<nf) GOTO 110
ENDIF
FORALL(i=0:i0,j=0:j0)p(i,j)=p(i,j)+bet*phi(i,j)   !p=p+bet*phi
lo=MAXLOC(phi); FORALL(k=1:2)locf(k,5)=lo(k); fmax(5)=MAXVAL(phi)
lo=MINLOC(phi); FORALL(k=1:2)locf(k,6)=lo(k); fmax(6)=MINVAL(phi)
ENDSUBROUTINE COMPP

! ***** Compute volume flux JU and JV
SUBROUTINE COMPU2(UJ,phi)

```

```

PARAMETER (if=140, jf=25)
DIMENSION UJ(2,0:if,0:jf), phi(0:if,0:jf), gU(3,0:if,0:jf), gV(3,0:if,0:jf), &
           pV(0:if), dpV(0:if), pU(0:jf), dpU(0:jf)
COMMON //na, Re, dt, i0, j0, lowRe, steady, unsteady
COMMON /METRIC3/gU, gV
FORALL (j=0:j0) phi(if, j) = -phi(i0, j)
DO i=1, if
           ! JU=JU*-dt(g22phi_xi-g12phi_eta)
  FORALL (j=0:j0) pU(j) = (phi(i-1, j) + phi(i, j)) / 2.
  CALL DIFP(pU, dpU, jf, 1.)
  FORALL (j=0:j0) &
    UJ(1, i, j) = UJ(1, i, j) - dt * (gU(3, i, j) * (phi(i, j) - phi(i-1, j)) - gU(2, i, j) * dpU(j))
ENDDO
DO j=1, j0
           ! JV=JV*-dt(-g12phi_xi+g11phi_eta)
  FORALL (i=0:if) pV(i) = (phi(i, j-1) + phi(i, j)) / 2.
  CALL DIFP(pV, dpV, if, 1.)
  i=i0; dpV(i) = (phi(i+1, j) - phi(i-1, j)) / 2.
           ! outlet
  FORALL (i=0:i0) &
    UJ(2, i, j) = UJ(2, i, j) - dt * (-gV(2, i, j) * dpV(i) + gV(1, i, j) * (phi(i, j) - phi(i, j-1)))
ENDDO
ENDSUBROUTINE COMPU2

! ***** Compute velocities u and v and vorticity zeta
SUBROUTINE COMPUZ(uJ, u, z, locf, fmax, CFLmax)
PARAMETER (if=140, jf=25, if1=280, jf1=50)
DIMENSION UJ(2,0:if,0:jf), u(2,0:if,0:jf), z(0:if,0:jf), div(0:if,0:jf), UJX(2,0:if,0:jf), &
           aJ(0:if1,0:jf1), gU(3,0:if,0:jf), gV(3,0:if,0:jf), xix(4,0:if1,0:jf1), w1(0:if), &
           w2(0:jf), w11(0:if1), w12(0:jf1), lo(2), locf(2,8), fmax(8)
COMMON //na, Re, dt, i0, j0, lowRe, steady, unsteady
COMMON /COMPUZ1/UJX /METRIC2/aJ /METRIC3/gU, gV /METRIC4/xix
! ***** Compute velocities at X
DO i=0, if
           ! u= xix4*JUX-xix2*JVX
           ! v=-xix3*JUX+xix1*JVX
  FORALL (j=0:jf) w2(j) = UJ(1, i, j)
  CALL INTUX(w2, w12, jf)
  FORALL (j=0:jf) UJX(1, i, j) = w12(2*j)
ENDDO
DO j=0, jf
  FORALL (i=0:if) w1(i) = UJ(2, i, j)
  CALL INTVX(w1, w11, if)
  FORALL (i=0:if) UJX(2, i, j) = w11(2*i)
ENDDO
FORALL (i=1:if, j=1:j0) u(1, i, j) = xix(4, 2*i, 2*j) * UJX(1, i, j) - xix(2, 2*i, 2*j) * UJX(2, i, j)
FORALL (i=1:if, j=1:j0) u(2, i, j) = -xix(3, 2*i, 2*j) * UJX(1, i, j) + xix(1, 2*i, 2*j) * UJX(2, i, j)
! ***** Compute vorticity zeta at X
FORALL (i=0:if, j=0:jf) z(i, j) = 0.
DO j=1, j0
  FORALL (i=0:if) w1(i) = UJX(1, i, j)
  CALL INTP(w1, w11, if)
           ! w11=JU_V
  FORALL (i=0:i0) w1(i) = gV(2, i, j) * w11(2*i+1) + gV(3, i, j) * UJ(2, i, j)
  FORALL (i=1:i0) z(i, j) = w1(i) - w1(i-1)
           ! *(g21JU+g22JV)_ , xi
  i=0; z(i, j) = -w1(i+2) + 3.*w1(i+1) - 2.*w1(i)
           ! inlet bound
  i=if; z(i, j) = w1(i-3) - 3.*w1(i-2) + 2.*w1(i-1)
           ! outlet bound
ENDDO
DO i=0, if
  FORALL (j=0:jf) w2(j) = UJX(2, i, j)
  CALL INTVP(w2, w12, jf)
           ! w12=JV_U
  FORALL (j=0:j0) w2(j) = gU(1, i, j) * UJ(1, i, j) + gU(2, i, j) * w12(2*j+1)
  FORALL (j=1:j0) z(i, j) = z(i, j) - (w2(j) - w2(j-1))
           ! **-(g11JU+g12JV)_ , eta

```

```

j=0 ; z(i,j)=-(-w2(j+1)/3.+3.*w2(j)); z(40,0)=0.           !bottom wall
j=jf; z(i,j)=- ( w2(j-2)/3.-3.*w2(j-1))                   !top wall
FORALL(j=0:jf)z(i,j)=z(i,j)/aJ(2*i,2*j)                   !zeta=*/J
ENDDO
! ***** Divergence and max CFL number
CFLmax=0.
FORALL(i=0:i0,j=0:j0)div(i,j)=(UJ(1,i+1,j)-UJ(1,i,j) &
+UJ(2,i,j+1)-UJ(2,i,j))/aJ(2*i+1,2*j+1)
lo=MAXLOC(div); FORALL(k=1:2)locf(k,7)=lo(k); fmax(7)=MAXVAL(div)
lo=MINLOC(div); FORALL(k=1:2)locf(k,8)=lo(k); fmax(8)=MINVAL(div)
DO i=0,i0; DO j=0,j0
CFLmax=MAX(CFLmax,ABS(UJ(1,i,j)/aJ(2*i,2*j+1)),ABS(UJ(2,i,j)/aJ(2*i+1,2*j)))
ENDDO; ENDDO
CFLmax=CFLmax*dt                                           !max|CFL|
ENDSUBROUTINE COMPUZ

! ***** Compute location of reattachment point
SUBROUTINE REATTACH(x,UJ,UJX,if,jf,xrap,k)
DIMENSION x(2,0:if,0:jf),UJ(2,0:if,0:jf),UJX(2,0:if,0:jf),al(-3:8)
k=100; irap=45; 10 irap=irap+1; IF(irap>105)RETURN
IF(UJ(1,irap,0)<0.) GOTO 10 !irap: grid point near rap
ii=-4; 12 ii=ii+1; i=irap+ii
du=UJ(1,i,0); d2u=UJX(1,i,1)-2.*du !du=DeltaJU, d2u=Delta^2JU
d3u=UJ(1,i,1)-3.*UJX(1,i,1)+3.*du !d3u=Delta^3JU
a=2.; 13 fa = du+(a-1.)/2.*(d2u+(a-2.)/3.*d3u) !fa=f(alpha) by cubic interpolation
IF(ABS(fa)<.00001) GOTO 14
dfa = d2u/2.+(2.*a-3.)/6.*d3u !dfa=f'(alpha)
a = a-fa/dfa; GOTO 13 !by Newton method
14 al(ii)=a; IF(a>0.14) GOTO 12 !al(ii): alpha of u=0
IF(al(ii)<0.01)ii=ii-1 !irap+ii: adjacent grid point to rap
! using quadratic interpolation formula f(beta)=a0+da*beta+d2a*beta*beta/2.
! get beta from f(beta)=0 by Newton method
a2=al(ii-2); a1=al(ii-1); a0=al(ii)
da=(3.*a0-4.*a1+a2)/2.; d2a=a0-2.*a1+a2
k=0; b=0.; 22 k=k+1; fb = a0+da*b+d2a*b*b/2. !fb=f(beta)
IF(k>10) GOTO 20
IF(ABS(fb)<0.01) GOTO 21
dfb = da+d2a*b !dfb=f'(beta)
b = b-fb/dfb; GOTO 22 !by Newton method
21 xii=x(1,irap+ii,0); xii=x(1,irap+ii+1,0)
20 xrap = xii+b*(xii-xii) !x of reattachment point
ENDSUBROUTINE REATTACH

! ***** minmod limiter
FUNCTION AMINMOD(a,b)
s=SIGN(1.,a); AMINMOD=s*MAX(0.,MIN(ABS(a),s*b))
ENDFUNCTION

! ***** Compute x- or y-differences at point P (pressure)
SUBROUTINE DIFP(u,du,n,h)
DIMENSION u(0:n),du(0:n)
FORALL(i=1:n-2)du(i)=(u(i+1)-u(i-1))/(2.*h)
du(0)=(-3.*u(0)+4.*u(1)-u(2))/(2.*h)
du(n-1)=(u(n-3)-4.*u(n-2)+3.*u(n-1))/(2.*h)
ENDSUBROUTINE DIFP

! ***** Compute x- or y-differences at point X (metrics)

```

```

SUBROUTINE DIFX(u,du,n,h)
DIMENSION u(0:n),du(0:n)
FORALL (i=1:n-1)du(i)=(u(i+1)-u(i-1))/(2.*h)
du(0)=(-3.*u(0)+4.*u(1)-u(2))/(2.*h)
du(n)=(u(n-2)-4.*u(n-1)+3.*u(n))/(2.*h)
ENDSUBROUTINE DIFX

! ***** Solve in parallel, systems of linear eqns with modified tri-diagonal matrix
! by Gaussian eliminaton
SUBROUTINE GAUSSP(a,b,ife,i0,n,if)
DIMENSION a(ife,0:i0,3),b(ife,0:i0)
REAL(8) a,b
FORALL (i=1:if)a(i,0,1)=a(i,0,1)/a(i,0,2)
FORALL (i=1:if)a(i,1,3)=a(i,1,3)-a(i,1,1)*a(i,0,1)
DO k=0,n-1; DO i=1,if
  b(i,k)=b(i,k)/a(i,k,2); a(i,k,3)=a(i,k,3)/a(i,k,2)
  b(i,k+1)=b(i,k+1)-a(i,k+1,1)*b(i,k)
  a(i,k+1,2)=a(i,k+1,2)-a(i,k+1,1)*a(i,k,3)
ENDDO; ENDDO
FORALL (i=1:if)a(i,n,1)= -a(i,n,3)*a(i,n-2,3)
FORALL (i=1:if)a(i,n,2)=a(i,n,2)-a(i,n,1)*a(i,n-1,3)
FORALL (i=1:if)b(i,n)=(b(i,n)-a(i,n,3)*b(i,n-2)-a(i,n,1)*b(i,n-1))/a(i,n,2)
DO k=n-1,0,-1; DO i=1,if
  b(i,k)=b(i,k)-a(i,k,3)*b(i,k+1)
ENDDO; ENDDO
DO i=1,if; b(i,0)=b(i,0)-a(i,0,1)*b(i,2); ENDDO
ENDSUBROUTINE GAUSSP

! ***** Solve in paralell, systems of linear eqns with tri-diagonal matrix
! by Gaussian elimination
SUBROUTINE GAUSSZ(a,b,m1,m,n)
DIMENSION a(m1,m1,3),b(m1,m1)
DO k=1,n-1; DO l=1,m
  b(l,k) =b(l,k) /a(l,k,2)
  a(l,k,3)=a(l,k,3)/a(l,k,2)
  b(l,k+1) =b(l,k+1) -a(l,k+1,1)*b(l,k)
  a(l,k+1,2)=a(l,k+1,2)-a(l,k+1,1)*a(l,k,3)
ENDDO; ENDDO
FORALL (l=1:m)b(l,n)=b(l,n)/a(l,n,2)
DO k=n-1,1,-1
  FORALL (l=1:m)b(l,k)=b(l,k)-a(l,k,3)*b(l,k+1)
ENDDO
ENDSUBROUTINE GAUSSZ

! ***** Interpolate u1 from u (U_P from U)
SUBROUTINE INTP(u,u1,n)
DIMENSION u(0:n),u1(0:2*n)
FORALL (i=0:n)u1(2*i)=u(i)
FORALL (i=1:n-2)u1(2*i+1)=(-u(i-1)+9.*(u(i)+u(i+1))-u(i+2))/16.
i=0 ; u1(2*i+1)=(3.*u(i)+6.*u(i+1)-u(i+2))/8.
i=n-1; u1(2*i+1)=(3.*u(i+1)+6.*u(i)-u(i-1))/8.
ENDSUBROUTINE INTP

! ***** Interpolate U_X from U
SUBROUTINE INTUX(u,u1,jf)
DIMENSION u(0:jf),u1(0:2*jf)
FORALL (j=0:jf-1)u1(2*j+1)=u(j)

```

```

FORALL (j=2:jf-2) u1(2*j) = (-u(j-2) + 9.*(u(j-1)+u(j)) - u(j+1))/16.
j=0 ; u1(2*j) = 0.
j=1 ; u1(2*j) = (3.*u(j-1) + 2.*u(j) - u(j+1))/5./4.
j=jf-1; u1(2*j) = (3.*u(j) + 2.*u(j-1) - u(j+2))/5./4.
j=jf ; u1(2*j) = 0.
ENDSUBROUTINE INTUX

! ***** Interpolate V_P from V
SUBROUTINE INTVP(u,u1,n)
DIMENSION u(0:n),u1(0:2*n)
FORALL (j=0:n) u1(2*j) = u(j)
FORALL (j=1:n-2) u1(2*j+1) = (-u(j-1) + 9.*(u(j)+u(j+1)) - u(j+2))/16.
j=0 ; u1(2*j+1) = (12.*u(j+1) - u(j+2))/32.
j=n-1; u1(2*j+1) = (12.*u(j) - u(j+1))/32.
ENDSUBROUTINE INTVP

! ***** Interpolate V_X from V
SUBROUTINE INTVX(u,u1,if)
DIMENSION u(0:if),u1(0:2*if)
FORALL (i=0:if-1) u1(2*i+1) = u(i)
FORALL (i=2:if-2) u1(2*i) = (-u(i-2) + 9.*(u(i-1)+u(i)) - u(i+1))/16.
i=0 ; u1(2*i) = 0.
i=1 ; u1(2*i) = (3.*u(i-1) + 2.*u(i) - u(i+1))/5./4.
i=if-1; u1(2*i) = (3.*u(i) + 6.*u(i-1) - u(i+2))/8.
i=if ; u1(2*i) = (3.*u(i-1) - u(i+2))/2.
ENDSUBROUTINE INTVX

! ***** Print out computational results in flow field
SUBROUTINE WRTF(f,z1,z2,mz,na)
PARAMETER (if=140,jf=25)
DIMENSION f(0:if,0:jf)
CHARACTER*4 z1,z2,form(7),area(7)
DATA form/"(1H ","2X I","2,2X"," 36("," ", "F8.4",")" ) "/
DATA area/" -3P"," -2P"," -1P"," ", " 1P"," 2P"," 3P"/
form(5) = area(mz+4)
OPEN(20,FILE='OUTPUT.dat')
WRITE(20,60) z1,z2,na
iif = (if-1)/35+1
DO ii=1,iif; is=35*(ii-1); ie=MIN(35*ii,if)
DO j=jf,0,-1; WRITE(20,form) j,(f(i,j),i=is,ie); ENDDO
WRITE(20,61)(i,i=is,ie)
ENDDO
60 FORMAT(/1H 20X " * * * ", 2A4, " * * * ", 10X "na =", I5, /)
61 FORMAT(1H 4X 36I8//)
ENDSUBROUTINE WRTF

! ***** Drawing of Grid and Computational Results
SUBROUTINE StepDF_CG(x,u,p,z,if,jf)
USE DFLIB
DIMENSION x(2,0:if,0:jf),u(2,0:if,0:jf),p(0:if,0:jf),z(0:if,0:jf),pp(7:105,0:jf)
LOGICAL statusmode
TYPE(windowconfig) myscreen
TYPE(wxycoord) wxy
INTEGER(2) status,xwidth,yheight,fontnum,numfonts
INTEGER(4) oldcolor,colors(13)
REAL(4) p0(0:if,0:jf),p1(0:if),p11(0:if),psi(0:if,0:jf),psik(13), &
x1(0:70,0:25),y1(0:70,0:25),f1(0:70,0:25), &

```

```

                x2(0:30,0:25),y2(0:30,0:25),f2(0:30,0:25),ak(13),Dps(13)
REAL(8)      xs,ys,xt,yt
CHARACTER*5  z1(7)
myscreen.numpixels = -1
myscreen.numypixels = -1
myscreen.numtextcols = -1
myscreen.numtextrows = -1
myscreen.numcolors = -1
myscreen.fontsize = -1
myscreen.title = " "C
statusmode = SETWINDOWCONFIG(myscreen)
IF(.NOT. statusmode) statusmode = SETWINDOWCONFIG(myscreen)
statusmode =GETWINDOWCONFIG(myscreen)
xwidth = myscreen.numpixels
yheight = myscreen.numypixels
oldcolor=SETBKCOLORRGB(#FFFFFF)
CALL CLEARSCREEN($GCLEARSCREEN)

! Set first window coordinates
CALL SETVIEWPORT(yheight/5.,INT2(0), yheight,yheight/5.)
status = SETWINDOW(.TRUE., -5.,-2., 11.,2.)
! Draw partial computational grid
oldcolor=SETCOLORRGB(#606060)                                !gray
DO i=0,if; xs=x(1,i,0); ys=x(2,i,0); CALL MOVETO_W(xs,ys,wxy)
  DO j=1,jf; xt=x(1,i,j); yt=x(2,i,j); status = LINETO_W(xt,yt); ENDDO
ENDDO
DO j=0,jf; xs=x(1,0,j); ys=x(2,0,j); CALL MOVETO_W(xs,ys,wxy)
  DO i=1,if; xt=x(1,i,j); yt=x(2,i,j); status = LINETO_W(xt,yt); ENDDO
ENDDO
oldcolor=SETCOLORRGB(#000000)
CALL CHARAC(1.0,-1.8,'COMPUTATIONAL GRID')

! Set second window coordinates
OPEN(20,FILE='OUTPUT.dat')
CALL SETVIEWPORT(yheight/5.,yheight/5., yheight,3.*yheight/5.)
status = SETWINDOW(.TRUE., -6.,-8., 18.,4.)
! Set colors
colors(1) =#E070A0; colors(2) =#FF7070                                !blue
colors(3) =#D0A070; colors(4) =#A0D070; colors(5) =#70FF70        !green
colors(6) =#60FFA0; colors(7) =#50FFD0; colors(8) =#40FFFF        !yellow
colors(9) =#50E0FF; colors(10)=#60C0FF; colors(11)=#7090FF
colors(12)=#7070FF                                                !red
colors(13)=#A070E0
status = REMAPALLPALETTERGB(colors)
! Draw pressure distribution
DO j=0,jf-1                                                        !interpolate p_X
  FORALL(i=0:if-1)p1(i)=p(i,j)
  CALL INTPX(p1,p11,if,if)
  FORALL(i=0:if)p0(i,j)=p11(i)
ENDDO
DO i=0,if
  FORALL(j=0:jf-1)p1(j)=p0(i,j)
  CALL INTPX(p1,p11,if,jf)
  FORALL(j=0:jf)p0(i,j)=p11(j)
ENDDO
DATA Dps/.005, .00667, .0075, .010, .0125, .015, .01667, .02, .025, &
      .03, .03333, .04, .05/
!standard pressure increments

```

```

FORALL (i=7:105,j=0:jf) pp(i,j)=p0(i,j)
pmax=MAXVAL(pp); pmin=MINVAL(pp)
Dp=(pmax-pmin)/13.
IF(Dp<.004 .OR. Dp>.06) RETURN !no suitable Dps
k=0; 10 k=k+1; IF(Dps(k)<Dp .AND. k<13) GOTO 10 !determine Dps
kkk=k; Dpk=Dps(k); imin=INT(pmin/Dpk); IF(pmin>0.) imin=imin+1 !No of min pressure
ak(12)=Dpk*(FLOAT(imin)+11.5)
11 IF(ak(12)>pmax) THEN; imin=imin-1; ak(12)=ak(12)-Dpk; GOTO 11 !determine imin
ENDIF
FORALL (k=1:12) ak(k)=Dpk*(FLOAT(imin+k)-.5)
FORALL (i=0:70,j=0:jf)
  x1(i,j)=x(1,i+6,j); y1(i,j)=x(2,i+6,j); f1(i,j)=p0(i+6,j); ENDFORALL
CALL DISTRIB(x1,y1,f1,70,jf,ak,colors,1,13)
WRITE(20,'(5X A6,F8.4,5X A6,F8.4)') 'pmax =',pmax,'pmin =',pmin
! Draw velocity vectors
oldcolor=SETCOLORRGB(#000000) !black
amag=.6
DO i=7,75,4; DO j=1,jf-1
  CALL LINE(x(1,i,j),x(2,i,j),x(1,i,j)+amag*u(1,i,j),x(2,i,j)+amag*u(2,i,j))
ENDDO; ENDDO
status = SETWINDOW(.TRUE., 18.,-3., 42.,9.)
! Draw pressure distribution
FORALL (i=0:30,j=0:jf)
  x2(i,j)=x(1,i+75,j); y2(i,j)=x(2,i+75,j); f2(i,j)=p0(i+75,j); ENDFORALL
CALL DISTRIB(x2,y2,f2,30,jf,ak,colors,1,13)
! Draw velocity vectors
oldcolor=SETCOLORRGB(#000000)
DO i=75,99,4; DO j=1,jf-1
  CALL LINE(x(1,i,j),x(2,i,j),x(1,i,j)+amag*u(1,i,j),x(2,i,j)+amag*u(2,i,j))
ENDDO; ENDDO
! Color-pressure relation
CALL COLORVAL(39.0,-0.95,39.3,2.95,colors,13,kkk,Dpk,imin)
oldcolor=SETCOLORRGB(#404040)
CALL CHARAC(23.0,-2.3,'PRESSURE DISTRIBUTION AND VELOCITY VECTORS')

! Set third window coordinates
CALL SETVIEWPORT(yheight/5.,3.*yheight/5., yheight,yheight)
status = SETWINDOW(.TRUE., -6.,-8., 18.,4.)
! Draw vorticity distribution
FORALL (i=0:70,j=0:jf)
  x1(i,j)=x(1,i+6,j); y1(i,j)=x(2,i+6,j); f1(i,j)=z(i+6,j); ENDFORALL
FORALL (k=1:13) ak(k)=-2.6+.4*FLOAT(k) !paint colors(k) between ak(k-1) and ak(k)
CALL DISTRIB(x1,y1,f1,70,jf,ak,colors,1,13)
! Draw streamlines
DO i=0,if
  psi(i,0)=0.
  DO j=1,jf
    psi(i,j)=psi(i,j-1)+(u(1,i,j-1)+u(1,i,j))/2.*(x(2,i,j)-x(2,i,j-1)) &
      -(u(2,i,j-1)+u(2,i,j))/2.*(x(1,i,j)-x(1,i,j-1))
  ENDDO
  dpsi=3.-psi(i,jf)
  FORALL (j=1:jf) psi(i,j)=psi(i,j)+dpsi*FLOAT(j)/jf
ENDDO
FORALL (i=0:70,j=0:jf) f1(i,j)=psi(i+6,j)
oldcolor=SETCOLORRGB(#606060) !gray
FORALL (k=1:13) psik(k)=.25*FLOAT(k-1)
CALL CONTOUR(x1,y1,f1,70,jf,psik,13)

```

```

oldcolor=SETCOLORRGB(#808080)                                !light gray
FORALL(k=1:13)psik(k)=-.01*FLOAT(k)
CALL CONTOUR(x1,y1,f1,70,jf,psik,13)
oldcolor=SETCOLORRGB(#606060)                                !gray
CALL LINE(-6.0,0.0,0.0,0.0) ; CALL LINE(0.0,0.0,0.0,-1.0)
CALL LINE(0.0,-1.0,18.0,-1.0); CALL LINE(-6.0,3.0,18.0,3.0)
status = SETWINDOW(.TRUE., 18.,-3.1, 42.,9)
! Draw vorticity distribution
FORALL(i=0:30,j=0:jf)
  x2(i,j)=x(1,i+75,j); y2(i,j)=x(2,i+75,j); f2(i,j)=z(i+75,j); ENDFORALL
CALL DISTRIB(x2,y2,f2,30,jf,ak,colors,1,13)
! Draw streamlines
FORALL(i=0:30,j=0:jf)f2(i,j)=psi(i+75,j)
oldcolor=SETCOLORRGB(#606060)                                !gray
FORALL(k=1:13)psik(k)=.25*FLOAT(k-1)
CALL CONTOUR(x2,y2,f2,30,jf,psik,13)
oldcolor=SETCOLORRGB(#808080)                                !light gray
FORALL(k=1:13)psik(k)=-.01*FLOAT(k)
CALL CONTOUR(x2,y2,f2,30,jf,psik,13)
oldcolor=SETCOLORRGB(#606060)                                !gray
CALL LINE(18.0,-1.0,42.0,-1.0); CALL LINE(18.0,3.0,42.0,3.0)
! Color-vorticity relation
xs=39.0; xt=39.3
DO k=1,13
  oldcolor=SETCOLORRGB(colors(k))
  ys=-0.95+0.3*FLOAT(k); yt=-0.95+0.3*FLOAT(k-1)
  status=RECTANGLE_W($GFILLINTERIOR,xs,ys,xt,yt)
ENDDO
oldcolor=SETCOLORRGB(#000000)
ys=2.95; yt=-.95; status = RECTANGLE_W($GBORDER,xs,ys,xt,yt)
numfonts=INITIALIZEFONTS()
fontnum =SETFONT('t'Arial'h14w6')
DATA z1/'-2.4 ','-1.6 ','-0.8 ',' 0.0 ',' 0.8 ',' 1.6 ',' 2.4 '/
DO k=1,7; xt=39.4; yt=-0.55+0.6*FLOAT(k-1)
  CALL MOVETO_W(xt,yt,wxy); CALL OUTGTEXT(z1(k))
ENDDO
CALL CHARAC(24.0,-2.3,'VORTICITY DISTRIBUTION AND STREAMLINES')
ENDSUBROUTINE StepDF_CG

! ***** character
SUBROUTINE CHARAC(x,y,A)
USE DFLIB
TYPE (windowconfig) myscreen
TYPE (wxycoord) wxy
INTEGER(2) numfonts,fontnum
REAL(8) xt,yt
CHARACTER*100 A
numfonts=INITIALIZEFONTS()
fontnum=SETFONT("t'Arial'h16w6'")
xt=x; yt=y+0.33; CALL MOVETO_W(xt,yt,wxy); CALL OUTGTEXT(A)
ENDSUBROUTINE CHARAC

! ***** Values of colors
SUBROUTINE COLORVAL(x1,y1,x2,y2,colors,kf,kkk,Dpk,imin)
USE DFLIB
TYPE(wxycoord) wxy
INTEGER*2 status,numfonts,fontnum,intv(13)

```

```

INTEGER*4  oldcolor,colors(kf)
REAL*8     xs,ys,xt,yt
CHARACTER*5 z1(-20:50)
dy=(y2-y1)/FLOAT(kf)
xs=x1; xt=x2
DO k=1,kf
  oldcolor=SETCOLORRGB(colors(k))
  ys=y1+dy*FLOAT(k); yt=y1+dy*FLOAT(k-1)
  status = RECTANGLE_W($GFILLINTERIOR,xs,ys,xt,yt)
ENDDO
oldcolor=SETCOLORRGB(#000000)
ys=y2; yt=y1; status = RECTANGLE_W($GBORDER,xs,ys,xt,yt)
numfonts=INITIALIZEFONTS()
fontnum =SETFONT('t','Arial','h14w6')
DATA z1/'-0.20','-0.19','-0.18','-0.17','-0.16','-0.15','-0.14','-0.13','-0.12','-0.11', &
      '-0.10','-0.09','-0.08','-0.07','-0.06','-0.05','-0.04','-0.03','-0.02','-0.01', &
      '0.00','0.01','0.02','0.03','0.04','0.05','0.06','0.07','0.08','0.09', &
      '0.10','0.11','0.12','0.13','0.14','0.15','0.16','0.17','0.18','0.19', &
      '0.20','0.21','0.22','0.23','0.24','0.25','0.26','0.27','0.28','0.29', &
      '0.30','0.31','0.32','0.33','0.34','0.35','0.36','0.37','0.38','0.39', &
      '0.40','0.41','0.42','0.43','0.44','0.45','0.46','0.47','0.48','0.49', &
      '0.50'/
                                !pressure value of each color band
DATA intv/2, 3, 4, 2, 4, 2, 3, 3*2, 3, 2*2/
                                !text intervals
intvk=intv(kkk)
IF(imin>=0)THEN; itextmin=((imin+intvk-1)/intvk)*intvk
ELSE          ; itextmin=( imin          /intvk)*intvk; ENDIF          !No of bottom text
l=0; 12 l=l+1
xt=x2+0.1; yt=y1+dy*(itextmin-imin+intvk*(l-1))+0.4          !coordinates of character
itext=INT((itextmin+intvk*(l-1))*(Dpk+.0001)*100.)          !No of press character
CALL MOVETO_W(xt,yt,wxy); CALL OUTGTEXT(z1(itext))
IF(itextmin-imin+intvk*l<=12)          GOTO 12
ENDSUBROUTINE COLORVAL

! ***** Draw contours of f with oldcolor
SUBROUTINE CONTOUR(x,y,f0,if,jf,fk,kf)
USE DFLIB
DIMENSION x(0:if,0:jf),y(0:if,0:jf),f0(0:if,0:jf),fk(kf)
TYPE(wxycoord) wxy
INTEGER(2)      status
REAL(4)         f(0:if,0:jf)
REAL(8)         xs,ys
e=.00001
DO k=1,kf
  DO i=0,if; DO j=0,jf
    f(i,j)=f0(i,j)-fk(k)
    IF(ABS(f(i,j))<e) f(i,j)=SIGN(e,f(i,j))
  ENDDO; ENDDO
  DO 10 i=0,if-1; DO 10 j=0,jf-1; imoveto=0          !for all elements
    i1=i; j1=j; x1=x(i1,j1); y1=y(i1,j1); f1=f(i1,j1)
    i2=i+1; j2=j; x2=x(i2,j2); y2=y(i2,j2); f2=f(i2,j2)
    i3=i; j3=j+1; x3=x(i3,j3); y3=y(i3,j3); f3=f(i3,j3)
    i4=i+1; j4=j+1; x4=x(i4,j4); y4=y(i4,j4); f4=f(i4,j4)
    IF(f1*f2<0)THEN
      xs=(f1*x2-f2*x1)/(f1-f2); ys=(f1*y2-f2*y1)/(f1-f2)          !starting point
      CALL MOVETO_W(xs,ys,wxy); imoveto=1
    ENDIF
    IF(f3*f4<0)THEN

```

```

xs=(f3*x4-f4*x3)/(f3-f4); ys=(f3*y4-f4*y3)/(f3-f4)
IF(imoveto==1)THEN
  status = LINETO_W(xs,ys); imoveto=0
ELSE; CALL MOVETO_W(xs,ys,wxy); imoveto=1
ENDIF
ENDIF
IF(f1*f3<0)THEN
  xs=(f1*x3-f3*x1)/(f1-f3); ys=(f1*y3-f3*y1)/(f1-f3)
  IF(imoveto==1)THEN
    status = LINETO_W(xs,ys); imoveto=0; GOTO 10
  ELSE; CALL MOVETO_W(xs,ys,wxy); imoveto=1
  ENDIF
ENDIF
IF(imoveto==1)THEN
  xs=(f2*x4-f4*x2)/(f2-f4); ys=(f2*y4-f4*y2)/(f2-f4)
  status = LINETO_W(xs,ys)
ENDIF
10 CONTINUE
ENDDO
ENDSUBROUTINE CONTOUR

! ***** Draw distribution of f with colors
SUBROUTINE DISTRIB(x,y,f,if,jf,ak,colors,kb,kf)
USE DFLIB
DIMENSION x(0:if,0:jf),y(0:if,0:jf),f(0:if,0:jf),ak(kb:kf),colors(kb:kf)
TYPE(wxycoord) wxy, poly(6)
INTEGER(2) status,m(4)
INTEGER(4) oldcolor,colors
REAL(4) fe(4)
REAL(8) xs,ys,xt,yt,xs0,ys0,xt0,yt0,xs1,ys1,xt1,yt1,wx,wy
jp=jf+1; ak(kf)=1.E10
DO i=0,if-1; DO j=0,jf-1; m1=i*jp+j+1 !for all quadrilateral elements
  m(1)=m1; fe(1)=f(i,j)
  m(2)=m1+jp; fe(2)=f(i+1,j)
  m(3)=m1+1; fe(3)=f(i,j+1)
  m(4)=m1+jp+1; fe(4)=f(i+1,j+1)
! m() and fe() are four node numbers and function values of each element, respectively.
! l1 l2 l3 or l4 is one of four element node numbers 1 2 3 4, and the fe-values increase in
! order. First locate max and min fe-value nodes l1 and l4, then determine middle fe-value
! nodes l2 and l3.
l1=MINLOC(fe,DIM=1); l4=MAXLOC(fe,DIM=1)
sum=FLOAT(10-l1-l4); prd=FLOAT(24/l1/l4); sqr=SQRT(sum*sum-4.*prd)
l2=INT((sum-sqr)/2.+1); l3=INT((sum+sqr)/2.+1)
IF(fe(l2)>fe(l3))THEN; l=l2; l2=l3; l3=l; ENDIF
! a1<=a2<=a3<=a4 are nodal values of function f.
i1=(m(l1)-1)/jp; j1=MOD(m(l1)-1,jp); x1=x(i1,j1); y1=y(i1,j1); a1=f(i1,j1)
i2=(m(l2)-1)/jp; j2=MOD(m(l2)-1,jp); x2=x(i2,j2); y2=y(i2,j2); a2=f(i2,j2)
i3=(m(l3)-1)/jp; j3=MOD(m(l3)-1,jp); x3=x(i3,j3); y3=y(i3,j3); a3=f(i3,j3)
i4=(m(l4)-1)/jp; j4=MOD(m(l4)-1,jp); x4=x(i4,j4); y4=y(i4,j4); a4=f(i4,j4)
k1=kb-1; l1 k1=k1+1; IF(ak(k1)<a1) GOTO 11
k2=kb-1; l2 k2=k2+1; IF(ak(k2)<a2) GOTO 12
k3=kb-1; l3 k3=k3+1; IF(ak(k3)<a3) GOTO 13
k4=kb-1; l4 k4=k4+1; IF(ak(k4)<a4) GOTO 14
! ak(k1)<=ak(k2)<=ak(k3)<=ak(k4) Small range ak(k-1)-ak(k) is painted by colors(k).
! i.e. -infy-ak(kb) by colors(kb), ak(kf-1)-ak(kf):infy by colors(kf).
! k1<=k2<=k3<=k4. ntype shows arrangement of k1 k2 k3 k4,
! ntype=1: k1-k2-k4-k3-k1

```

```

! ntype=2: k1-k2-k3-k4-k1,
! ntype=3: k1-k3-k2-k4-k1(saddle point).
ntype=1
LA = i3==i1.OR.j3==j1; IF(LA==.FALSE.)ntype=2
LB = i2==i1.OR.j2==j1; IF(LB==.FALSE.)ntype=3
! xs ys and xt yt are coordinates of points on edge of quadrilateral element, whose function
! value is a. The coordinates are determined by linear interpolation.
DO 20 k=k1,k4; a=ak(k)
  IF(ntype==3)THEN; nvertx=4 !temporary
    poly(1)%wx=x1; poly(1)%wy=y1
    poly(2)%wx=x3; poly(2)%wy=y3
    poly(3)%wx=x2; poly(3)%wy=y2
    poly(4)%wx=x4; poly(4)%wy=y4;          GOTO 21
  ENDIF
  IF(k==k4)          GOTO 22
  IF(ntype==1)THEN !ntype=1
    IF(k<k3)THEN; xs=(a1-a)/(a1-a3)*x3+(a-a3)/(a1-a3)*x1 !k1<=k<k3
      ys=(a1-a)/(a1-a3)*y3+(a-a3)/(a1-a3)*y1
    ELSE;
      xs=(a3-a)/(a3-a4)*x4+(a-a4)/(a3-a4)*x3 !k3<=k<k4
      ys=(a3-a)/(a3-a4)*y4+(a-a4)/(a3-a4)*y3
    ENDIF
    IF(k<k2)THEN; xt=(a1-a)/(a1-a2)*x2+(a-a2)/(a1-a2)*x1 !k1<=k<k2
      yt=(a1-a)/(a1-a2)*y2+(a-a2)/(a1-a2)*y1
    ELSE;
      xt=(a2-a)/(a2-a4)*x4+(a-a4)/(a2-a4)*x2 !k2<=k<k4
      yt=(a2-a)/(a2-a4)*y4+(a-a4)/(a2-a4)*y2
    ENDIF
  ELSEIF(ntype==2)THEN !ntype=2
    xs=(a1-a)/(a1-a4)*x4+(a-a4)/(a1-a4)*x1 !k1<=k<k4
    ys=(a1-a)/(a1-a4)*y4+(a-a4)/(a1-a4)*y1
    IF(k<k2)THEN; xt=(a1-a)/(a1-a2)*x2+(a-a2)/(a1-a2)*x1 !k1<=k<k2
      yt=(a1-a)/(a1-a2)*y2+(a-a2)/(a1-a2)*y1
    ELSEIF(k<k3)THEN
      xt=(a2-a)/(a2-a3)*x3+(a-a3)/(a2-a3)*x2 !k2<=k<k3
      yt=(a2-a)/(a2-a3)*y3+(a-a3)/(a2-a3)*y2
    ELSE;
      xt=(a3-a)/(a3-a4)*x4+(a-a4)/(a3-a4)*x3 !k3<=k<k4
      yt=(a3-a)/(a3-a4)*y4+(a-a4)/(a3-a4)*y3
    ENDIF
  ELSE;
    xs=(a1-a)/(a1-a4)*x4+(a-a4)/(a1-a4)*x1 !ntype=3
    ys=(a1-a)/(a1-a4)*y4+(a-a4)/(a1-a4)*y1
    IF(k2<k3.AND.k2<=k.AND.k<=k3)THEN
      xs1=(a2-a)/(a2-a3)*x3+(a-a3)/(a2-a3)*x2
      ys1=(a2-a)/(a2-a3)*y3+(a-a3)/(a2-a3)*y2
    ENDIF
    IF(k<k3)THEN; xt=(a1-a)/(a1-a3)*x3+(a-a3)/(a1-a3)*x1
      yt=(a1-a)/(a1-a3)*y3+(a-a3)/(a1-a3)*y1
    ENDIF
    IF(k2<=k)THEN; xt1=(a2-a)/(a2-a4)*x4+(a-a4)/(a2-a4)*x2
      yt1=(a2-a)/(a2-a4)*y4+(a-a4)/(a2-a4)*y2
    ENDIF
  ENDIF
22 CONTINUE
! Divide into some polygons quadrilateral element in accordance with stepped function a(k).
! poly() are coordinates of vertices of polygon, and nvertx is number of the vertices.
! Coloring is performed in order k1...k4 in each element, but the following program written
! in order k1 k4 k1<k<k4.
  IF(k==k1)THEN !k==k1
    poly(1)%wx=x1; poly(1)%wy=y1

```



```

        poly(6)%wx=x2; poly(6)%wy=y2; nvertx=6
    ENDIF; ENDIF
ENDIF
IF(k<k4)THEN
    xs0=xs; ys0=ys; xt0=xt; yt0=yt
ENDIF
21 oldcolor=SETCOLORRGB(colors(k))
    status=POLYGON_W($GFILLINTERIOR,poly,nvertx)
20 CONTINUE
ENDDO; ENDDO
ENDSUBROUTINE DISTRIB

! ***** Interpolate values at U from values at P
SUBROUTINE INTPX(u,u1,if,n)
DIMENSION u(0:if),u1(0:if)
FORALL(i=2:n-2)u1(i)=(-u(i-2)+9.*(u(i-1)+u(i))-u(i+1))/16.
i=0 ; u1(i)=(3.*u(i)-u(i+1))/2.
i=1 ; u1(i)=(3.*u(i-1)+6.*u(i)-u(i+1))/8.
i=n-1; u1(i)=(3.*u(i)+6.*u(i-1)-u(i-2))/8.
i=n ; u1(i)=(3.*u(i-1)-u(i-2))/2.
ENDSUBROUTINE INTPX

! ***** Draw line
SUBROUTINE LINE(x1,y1,x2,y2)
USE DFLIB
TYPE (wxycoord) wxy
INTEGER(2) status
REAL(8) xs,ys,xt,yt
xs=x1; ys=y1; xt=x2; yt=y2; CALL MOVETO_W(xs,ys,wxy); status = LINETO_W(xt,yt)
ENDSUBROUTINE LINE

```

次にこのプログラムのおもなサブルーチンを説明する。サブルーチン GRID は、楕円型微分方程式の境界値問題を解いて曲線座標格子を形成する解析的格子形成法 (analytical grid generation) のプログラムである。計算はすべて写像面上の単位正方形格子に対して行われる。バックステップ流路は写像面上の長方形領域 $0 \leq \xi \leq 140$, $0 \leq \eta \leq 25$ に写像される。格子間隔 $\Delta\xi = \Delta\eta = 1$ 。上流境界 ($x = -10$) の格子点番号は $i = 0$, 下流境界 ($x = 70$) は $i_f = 140$, ステップ ($x = 0$) の角は $i_1 = 35$, 隅は $i_2 = 40$ である。また下方壁面 ($y = 0$ ($x \leq 0$), $x = 0$ ($-1 < y < 0$), $y = -1$ ($x \geq 0$)) の格子点番号は $j = 0$, 上方壁面 ($y = 3$) は $j_f = 25$ である。

曲線座標格子は基本的に次の Poisson 方程式の境界値問題を解くことによって形成される。

$$\xi_{xx} + \xi_{yy} = P, \quad \eta_{xx} + \eta_{yy} = Q \quad (13.84)$$

ただし P, Q は制御関数と呼ばれ主に格子間隔の制御に使われる。しかしながらこの方程式は、独立変数が xy で $\xi\eta$ 面上での計算に適しない。格子形成の計算には上式を書換えた次式が用いられる。

$$L(x) = -J(x_\xi P + x_\eta Q) \quad (13.85a)$$

$$L(y) = -J(y_\xi P + y_\eta Q) \quad (13.85b)$$

ただし L は微分演算子で

$$L = \tilde{g}_{22} \frac{\partial^2}{\partial \xi^2} - 2\tilde{g}_{12} \frac{\partial^2}{\partial \xi \partial \eta} + \tilde{g}_{11} \frac{\partial^2}{\partial \eta^2}$$

$\tilde{g}_{ij} = g_{ij}/J$, ヤコビアン J と測度テンソル g_{ij} は前節で定義されたものである¹⁹ .

式 (13.85) の境界値問題の境界条件は次のように与えられる . 下方境界には境界点の座標が等比級数によって , また上方境界点の $y = 3$, 上流境界点の $x = -10$, 下流境界点の $x = 70$ が与えられる . 上方境界点の x の値は適当に , また内点の xy の値はとりあえず $x_{ij} = (1-\alpha_j)x_{i0} + \alpha_j x_{ij_f}$ によって補間される . ただし $\alpha(\tilde{\eta}) = -\tilde{\eta}^3 + 1.6\tilde{\eta}^2 + 0.4\tilde{\eta}$, $\tilde{\eta} = \eta/\eta_f$ である . なおこの 3 次式は条件 $\alpha(0) = 0$, $\alpha(1) = 1$, $\alpha'(0) = 0.4$, $\alpha'(1) = 0.6$ を満足するように作られたものである .

制御関数 P は , ここでは下方境界上に与えられた格子間隔が領域内部まで適当に浸透するように次式によって与えられる .

$$P_{ij} = P_{i0} \quad \text{or} \quad = \frac{\eta_t - \eta}{\eta_t} P_{i0}$$

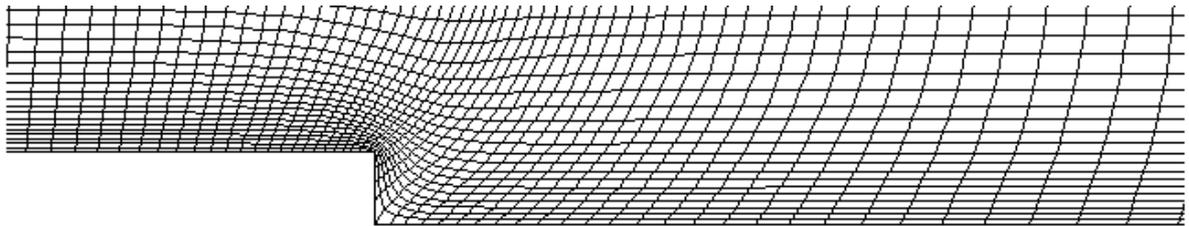
$$P_{i0} = (\xi_{xx})_{i0} = \frac{2}{x_{i+1,0} - x_{i-1,0}} \left(\frac{1}{x_{i+1,0} - x_{i0}} - \frac{1}{x_{i0} - x_{i-1,0}} \right)$$

また制御関数 Q は , j 方向の格子間隔が流路中央で粗く境界とりわけ下方境界で細くなるように次式によって与えられる .

$$Q_{ij} = (\eta_{yy})_{ij} = \eta''(y)_{ij} = \left(\frac{d\tilde{y}}{dy} \right)_{ij}^2 \tilde{\eta}''(\tilde{y})_j \frac{d\eta}{d\tilde{\eta}}$$

$$\left(\frac{d\tilde{y}}{dy} \right)_{ij}^2 \frac{d\eta}{d\tilde{\eta}} = \frac{25}{(y_{ij_f} - y_{i0})^2}, \quad \tilde{\eta}''(\tilde{y}) = 8\tilde{y} - 5$$

ただし下の行の第 1 式は $\tilde{\eta} = \eta/25$, $\tilde{y} = (y - y_0)/(y_{j_f} - y_0)$ なる関係から求められる倍率 , また第 2 式は 3 次式 $\tilde{\eta}(\tilde{y}) = (8\tilde{y}^3 - 15\tilde{y}^2 + 13\tilde{y})/6$, $(0 \leq \tilde{y} \leq 1)$ の 2 階微分である . この 3 次式は条件 $\tilde{\eta}(0) = 0$, $\tilde{\eta}(1) = 1$, $\tilde{\eta}'(0) = 13/6$, $\tilde{\eta}'(1) = 7/6$ を満足するもので , η 方向の格子間隔の分布はこの式によって規定される . なおプログラムには , 格子間隔の差の小さいものから大きいものまでいくつかの式が用意されており , その中から好みのもので選ぶことができる .



COMPUTATIONAL GRID

図 13.8: 一般曲線座標格子

¹⁹ 自明の関係 $x = x(\xi(x, y), \eta(x, y))$ を x で 2 階微分した式と y で 2 階微分した式

$$\xi_x^2 x_{\xi\xi} + 2\xi_x \eta_x x_{\xi\eta} + \eta_x^2 x_{\eta\eta} + \xi_{xx} x_\xi + \eta_{xx} x_\eta = 0,$$

$$\xi_y^2 x_{\xi\xi} + 2\xi_y \eta_y x_{\xi\eta} + \eta_y^2 x_{\eta\eta} + \xi_{yy} x_\xi + \eta_{yy} x_\eta = 0$$

を加え , 式 (13.66) と (13.84) を用いれば次式が得られる .

$$g^{11} x_{\xi\xi} + 2g^{12} x_{\xi\eta} + g^{22} x_{\eta\eta} + x_\xi P + x_\eta Q = 0$$

この式にヤコビアン J をかけ , $Jg^{11} = \tilde{g}_{22}$, $Jg^{12} = -\tilde{g}_{12}$, $Jg^{22} = \tilde{g}_{11}$ なる関係を用いれば , 式 (13.85a) を導出できる . 同様に式 $y = y(\xi(x, y), \eta(x, y))$ から始めれば式 (13.85b) を導出できる .

式 (13.85) の差分方程式は xy の格子点の値 x_{ij} または y_{ij} の連立 1 次方程式で, SOR 法で容易に解くことができる. ただしこれらの差分方程式の係数と右辺には解 xy が含まれるので, これらの係数と右辺の値を更新し SOR 法の計算を続行することになる.

METRIC は, 測度, ヤコビアン, 測度テンソルなどの値を計算するものである. 始めに点 X, P, U, V すなわち $(2i_f+1) \times (2j_f+1)$ 個の点における座標 x, y , 測度 $x_\xi, y_\xi, x_\eta, y_\eta$, ヤコビアン $J = x_\xi y_\eta - x_\eta y_\xi$, 測度 $\xi_x = y_\eta/J, \xi_y = -x_\eta/J, \eta_x = y_\xi/J, \eta_y = x_\xi/J$ の値が求められる. また点 U または V における測度テンソル成分 $\tilde{g}_{11} = (x_\xi^2 + y_\xi^2)/J, \tilde{g}_{12} = (x_\xi x_\eta + y_\xi y_\eta)/J, \tilde{g}_{22} = (x_\eta^2 + y_\eta^2)/J$ が計算され, それぞれ配列 gU または gV に格納される. $\tilde{g}^{11} = \tilde{g}_{22}, \tilde{g}^{12} = -\tilde{g}_{12}, \tilde{g}^{22} = \tilde{g}_{11}$. 最後に点 U の測度の微分 $\partial \xi_x / \partial \xi, \partial \xi_x / \partial \eta, \partial \xi_y / \partial \xi, \partial \xi_y / \partial \eta$ と点 V の $\partial \eta_x / \partial \xi, \partial \eta_x / \partial \eta, \partial \eta_y / \partial \xi, \partial \eta_y / \partial \eta$ が計算され, 配列 $dxix$ に格納される.

PREDCT は, 体積流束 \tilde{U}, \tilde{V} と静圧 p の予測値を与えるものである. まず上流境界の流速が平均流速 1 の放物線分布 $u_{0,j} = 6y_{0,j}(1-y_{0,j}), y_{0,j} = y_{0j}/y_{0j}, v_{0,j} = 0$ によって与えられる. このとき体積流速は $\tilde{U}_{0j} = (J\xi_x u)_{0j}, \tilde{V}_{0j} = 0$ となる. 反変速度が $\xi\eta$ 空間の流速であることを考えれば, 予測値は $\tilde{U}_{ij} = \tilde{U}_{0j}, \tilde{V}_{ij} = \tilde{V}_{0j}$ のように置くことができる. 平行平板間流路の層流では, 粘性による圧力降下は $\Delta p = \nu(\partial^2 u / \partial y^2) \Delta x$ から求められる. ステップの上流側では $\partial^2 u / \partial y^2 = -4/3$, 下流側では $\partial^2 u / \partial y^2 = -9/16$ である. また拡大部の(回復)圧力上昇は $\Delta p = 1/2 - .75^2/2 = 7/32$ となる. したがって下流境界の圧力を 0 とすれば, 上流境界の圧力の予測値は $p_0 = \{-(4/3)x_0 + (9/16)x_{i_f}\}/Re - 7/32$ となる. プログラムでは, p の予測値は j 方向に一様で, 上流側 ($0 \leq i \leq 35$) では $p_{ij} = p_0 + (4/3)(x_{0j} - x_{ij})/Re$, 下流側の ($35 \leq i \leq i_0$) では $p_{ij} = (9/16)(x_{i_f} - x_{ij})/Re$ とし, またステップ下流 ($35 < i < 100$) では $p_{35,j}$ と $p_{100,j}$ から線形補間している. なお拡大部の圧力上昇は, 平均流速の $\bar{u}^2/2$ からではなく $(1/2Y) \int_0^Y [u(y)]^2 dy$ から求めれば, $\Delta p = .6 - .6 \times .75^2 = .2625$ となる.

COMPUZ は点 X の流速 u, v と渦度 ζ を計算するサブルーチンである. これらの量の計算式は式 (13.68) または (13.72) から次のようになる.

$$u = \eta_y \tilde{U} - \xi_y \tilde{V}, \quad v = -\eta_x \tilde{U} + \xi_x \tilde{V} \quad (13.86)$$

$$\zeta = \frac{1}{J} \left\{ \frac{\partial}{\partial \xi} (\tilde{g}_{21} \tilde{U} + \tilde{g}_{22} \tilde{V}) - \frac{\partial}{\partial \eta} (\tilde{g}_{11} \tilde{U} + \tilde{g}_{12} \tilde{V}) \right\} \quad (13.87)$$

なお INTP は格子点 X の値が既知のときに点 U または V の値を補間するサブルーチンで, もちろん点 U または V の値から点 P の値の補間にも使用できる. INTVP は点 V の \tilde{V} の値から点 P の \tilde{V}_P の値を境界条件 $\tilde{V}_{i_0} = (\partial \tilde{V} / \partial \eta)_{i_0} = 0$ を考慮して補間するものである. 同様に INTVX は点 V の \tilde{V} の値から点 X の \tilde{V}_X の値を, INTUX は点 U の \tilde{U} の値から点 X の \tilde{U}_X の値を境界条件を考慮して補間するものである. 等間隔計算点 x_i の関数値 $u_i = u(x_i)$ が与えられるときに中間点 $x_{i+1/2}$ の微分値は周知のように $u'_{i+1/2} = (u_{i+1} - u_i) / \Delta x$ である. また境界 $x_{-1/2}$ のところでは内点の関数値を用い $u'_{-1/2} = (-2u_0 + 3u_1 - u_2) / \Delta x$ となるが, 特に境界値 $u_{-1/2} = 0$ の場合には $u'_{-1/2} = (3u_0 - u_1) / 3 / \Delta x$ となる.

$$\nabla \cdot \mathbf{u} = \frac{1}{J} \left(\frac{\partial}{\partial \xi} \tilde{U} + \frac{\partial}{\partial \eta} \tilde{V} \right)$$

COMPU1 と COMPU2 は, Navier-Stokes 方程式を 2 段階に分けて解く SMAC 法において, それぞれ前段の $\tilde{U}^* = \mathcal{J}U^*$, 後段の $\tilde{U} = \mathcal{J}U$ を計算するサブルーチンである. 前段の COMPU1 から説明する. Δ 形陰解法の式 (13.81a) は, 2 次元の場合には次のようになる.

$$\left\{ J + \Delta t \theta \left(\tilde{U} \frac{\partial}{\partial \xi} + \tilde{V} \frac{\partial}{\partial \eta} \right) - \mathcal{D}_1 \right\} \Delta U_1^* = rhs_1^n,$$

$$\left\{ J + \Delta t \theta \left(\tilde{U} \frac{\partial}{\partial \xi} + \tilde{V} \frac{\partial}{\partial \eta} \right) - \mathcal{D}_2 \right\} \Delta U_1^* = rhs_2^n$$

ただし \mathcal{D}_i は粘性項の主要部で，また右辺 rhs は次のようになる．

$$\begin{aligned} rhs_1 &= -\Delta t \left\{ \frac{\partial}{\partial \xi} \tilde{U}U + \frac{\partial}{\partial \eta} \tilde{V}U - u \left(\tilde{U} \frac{\partial \xi_x}{\partial \xi} + \tilde{V} \frac{\partial \xi_x}{\partial \eta} \right) - v \left(\tilde{U} \frac{\partial \xi_y}{\partial \xi} + \tilde{V} \frac{\partial \xi_y}{\partial \eta} \right) + \tilde{g}^{11} p_\xi + \tilde{g}^{12} p_\eta + \nu \zeta_\eta \right\}, \\ rhs_2 &= -\Delta t \left\{ \frac{\partial}{\partial \xi} \tilde{U}V + \frac{\partial}{\partial \eta} \tilde{V}V - u \left(\tilde{U} \frac{\partial \eta_x}{\partial \xi} + \tilde{V} \frac{\partial \eta_x}{\partial \eta} \right) - v \left(\tilde{U} \frac{\partial \eta_y}{\partial \xi} + \tilde{V} \frac{\partial \eta_y}{\partial \eta} \right) + \tilde{g}^{21} p_\xi + \tilde{g}^{22} p_\eta - \nu \zeta_\xi \right\} \end{aligned}$$

これらの式は左辺の演算子に 1 次上流差分，右辺の対流項に Chakravarthy-Osher TVD スキームを用い差分方程式に書き換えれば次のようになる．

$$\begin{aligned} J\Delta U^* + \Delta t \theta \left\{ \left(\tilde{U}^+ + \frac{\tilde{g}^{22}}{Re} \right) (\Delta U_{ij}^* - \Delta U_{i-1,j}^*) + \left(\tilde{U}^- - \frac{\tilde{g}^{22}}{Re} \right) (\Delta U_{i+1,j}^* - \Delta U_{ij}^*) \right. \\ \left. + \left(\tilde{V}^+ + \frac{\tilde{g}^{11}}{Re} \right) (\Delta U_{ij}^* - \Delta U_{i,j-1}^*) + \left(\tilde{V}^- - \frac{\tilde{g}^{11}}{Re} \right) (\Delta U_{i,j+1}^* - \Delta U_{ij}^*) \right\} = rhs_1 \end{aligned} \quad (13.88a)$$

$$\begin{aligned} J\Delta V^* + \Delta t \theta \left\{ \left(\tilde{U}^+ + \frac{\tilde{g}^{22}}{Re} \right) (\Delta V_{ij}^* - \Delta V_{i-1,j}^*) + \left(\tilde{U}^- - \frac{\tilde{g}^{22}}{Re} \right) (\Delta V_{i+1,j}^* - \Delta V_{ij}^*) \right. \\ \left. + \left(\tilde{V}^+ + \frac{\tilde{g}^{11}}{Re} \right) (\Delta V_{ij}^* - \Delta V_{i,j-1}^*) + \left(\tilde{V}^- - \frac{\tilde{g}^{11}}{Re} \right) (\Delta V_{i,j+1}^* - \Delta V_{ij}^*) \right\} = rhs_2 \end{aligned} \quad (13.88b)$$

$$\begin{aligned} rhs_1 &= -\Delta t \left\{ \widehat{\tilde{U}U}_{ijP} - \widehat{\tilde{U}U}_{i-1,jP} + \widehat{\tilde{V}U}_{i,j+1X} - \widehat{\tilde{V}U}_{ijX} \right. \\ &\quad - u_{ijU} \left(\tilde{U} \frac{\partial \xi_x}{\partial \xi} + \tilde{V} \frac{\partial \xi_x}{\partial \eta} \right)_{ijU} - v_{ijU} \left(\tilde{U} \frac{\partial \xi_y}{\partial \xi} + \tilde{V} \frac{\partial \xi_y}{\partial \eta} \right)_{ijU} \\ &\quad \left. + (\tilde{g}^{22})_{ijU} (p_{ij} - p_{i-1,j}) - (\tilde{g}^{12})_{ijU} \frac{1}{2} (p_{i,j+1U} - p_{i,j-1U}) + \frac{1}{Re} (\zeta_{i,j+1} - \zeta_{ij}) \right\} \end{aligned} \quad (13.88c)$$

$$\begin{aligned} rhs_2 &= -\Delta t \left\{ \widehat{\tilde{U}V}_{i+1,jX} - \widehat{\tilde{U}V}_{i,jX} + \widehat{\tilde{V}V}_{ijP} - \widehat{\tilde{V}V}_{i,j-1P} \right. \\ &\quad - u_{ijV} \left(\tilde{U} \frac{\partial \eta_x}{\partial \xi} + \tilde{V} \frac{\partial \eta_x}{\partial \eta} \right)_{ijV} - v_{ijV} \left(\tilde{U} \frac{\partial \eta_y}{\partial \xi} + \tilde{V} \frac{\partial \eta_y}{\partial \eta} \right)_{ijV} \\ &\quad \left. - (\tilde{g}^{12})_{ijV} \frac{1}{2} (p_{i+1,jV} - p_{i-1,jV}) - (\tilde{g}^{11})_{ijV} (p_{i,j} - p_{i,j-1}) - \frac{1}{Re} (\zeta_{i+1,j} - \zeta_{ij}) \right\} \end{aligned} \quad (13.88d)$$

なおここでは添字 ij^n を一部省略している． $\widehat{\quad}$ の付いている量は数値流束で，このサブルーチンには中心差分と Chakravarthy-Osher TVD スキームの 2 つが組み込まれている．後者の数値流束は

$$\begin{aligned} \widehat{\tilde{U}U}_{ijP} &= \tilde{U}U_{ijP} + |\tilde{U}_{ijP}| \left\{ -\frac{\Delta U_{i+1/2,j}}{2} + \frac{1}{6} \min\text{mod}(\Delta U_{i\pm 1+1/2,j}, 4\Delta U_{i+1/2,j}) \right. \\ &\quad \left. + \frac{1}{3} \min\text{mod}(\Delta U_{i+1/2,j}, 4\Delta U_{i\pm 1+1/2,j}) \right\} \end{aligned} \quad (13.89)$$

$$\Delta U_{i+1/2,j} = U_{i+1,j} - U_{i,j}, \quad \Delta U_{i\pm 1+1/2,j} = \begin{cases} \Delta U_{i-1/2,j} & (\tilde{U}_{ijP} \geq 0) \\ \Delta U_{i+3/2,j} & (\tilde{U}_{ijP} < 0) \end{cases}$$

のようになる．

境界条件は次のように考慮される．流路の入口と出口に逆流はないものとする．境界とその近傍点の数値

流束は次のようになる .

$$\widehat{U}U_{0jP} = \tilde{U}_{0jP} \frac{U_{0j} + U_{1j}}{2}, \quad (13.90a)$$

$$\widehat{V}U_{i0X} = \widehat{V}U_{ijfX} = 0, \quad \widehat{V}U_{i1X} = \tilde{V}_{i1X} \frac{1}{4} \left(3U_{i0} + 2U_{i1} - \frac{U_{i2}}{5} \right), \quad (\tilde{V}_{i1X} > 0) \quad (13.90b)$$

$$\widehat{U}V_{1jX} = \tilde{U}_{1jX} \frac{V_{0j} + V_{1j}}{2}, \quad \widehat{U}V_{i0jX} = \tilde{U}_{i0jX} \frac{1}{8} (-V_{i0-2,j} + 6V_{i0-1,j} + 3V_{i0}), \quad (13.90c)$$

$$\widehat{V}V_{i0P} = \tilde{V}_{i0P} \frac{1}{32} (12V_{i1} - V_{i2}) \quad (\tilde{V}_{i0P} > 0) \quad (13.90d)$$

なお最後の式は壁面上の反射の条件 $\partial V / \partial \eta = 0$ を用いて導かれたものである²⁰

下流境界点の \tilde{U}^* とその隣接点の \tilde{V}^* の式の右辺の対流項は、非保存形の式を 1 次上流差分で近似した安定かつ境界から反射のない次式によって求められる .

$$C_l = J \nabla \xi_l \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) = (\xi_l)_x \left(\tilde{U} \frac{\partial u}{\partial \xi} + \tilde{V} \frac{\partial u}{\partial \eta} \right) + (\xi_l)_y \left(\tilde{U} \frac{\partial v}{\partial \xi} + \tilde{V} \frac{\partial v}{\partial \eta} \right) \quad (l = 1, 2) \quad (13.91a)$$

$l = 1$ の式の $(\partial u / \partial \xi)_{ij}$ と $l = 2$ の式の $(\partial u / \partial \xi)_{i0j}$ は次のように上流差分で近似される . $(\partial v / \partial \xi)_{ij}$, $(\partial v / \partial \xi)_{i0j}$ についても同様である .

$$\left(\frac{\partial u}{\partial \xi} \right)_{ij} = \frac{1}{2} (u_{ij} - u_{i0j} + u_{ij,j+1} - u_{i0,j+1}), \quad \left(\frac{\partial u}{\partial \xi} \right)_{i0j} = \frac{1}{2} (u_{ij} - u_{i0-1,j}) \quad (13.91b)$$

付加項, 圧力項, 粘性項の計算については説明を省略する . この計算には METRIC で計算した $dxix = \partial \xi_x / \partial \xi$, ... と $gU, gV = \tilde{g}_{11}, \dots$ が用いられる .

SMAC Δ 形陰解法の式 (13.88) は 13.5 節に述べた改良型近似因子法を用いて効率的に解かれる . 13.5 節の因子化する前の差分方程式

$$c_0 u_{00} - c_1^+ u_{-1,0} + c_1^- u_{10} - c_2^+ u_{0,-1} + c_2^- u_{01} = rhs$$

と上記の式 (13.88a) を比較すれば、この式の係数 c_1^+ , c_1^- , c_2^+ , c_2^- , c_0 は式 (13.88a) の係数と

$$\begin{aligned} c_1^+ &= -\Delta t \theta (\tilde{U}^+ + \tilde{g}_{22} / Re), & c_1^- &= \Delta t \theta (\tilde{U}^- - \tilde{g}_{22} / Re), \\ c_2^+ &= -\Delta t \theta (\tilde{V}^+ + \tilde{g}_{11} / Re), & c_2^- &= \Delta t \theta (\tilde{V}^- - \tilde{g}_{11} / Re), & c_0 &= J - c_1^+ - c_1^- - c_2^+ - c_2^- \end{aligned}$$

のように対応していることが分かる . これらの係数は配列 c に入れられ、このようにして得られた連立 1 次方程式はサブルーチン GAUSSZ で解かれる . GAUSSZ については 13.5 節の説明参照 .

後段の COMPU2 は式 (13.81b) から \tilde{U}^{n+1} を計算する短いサブルーチンである . 式 (13.81b) の差分方程式は次のようになる .

$$\tilde{U}_{ij}^{n+1} = \tilde{U}_{ij}^* - \Delta t \left\{ (\tilde{g}_{22})_{ijU} (\phi_{ij} - \phi_{i-1,j}) - (\tilde{g}_{12})_{ijU} \frac{\phi_{i,j+1U} - \phi_{i,j-1U}}{2} \right\} \quad (13.92a)$$

$$\tilde{V}_{ij}^{n+1} = \tilde{V}_{ij}^* - \Delta t \left\{ -(\tilde{g}_{12})_{ijV} \frac{\phi_{i+1,jV} - \phi_{i-1,jV}}{2} + (\tilde{g}_{11})_{ijV} (\phi_{ij} - \phi_{i,j-1}) \right\} \quad (13.92b)$$

なお非定常流の場合にも、式 (13.83a) から $\tilde{U}^{*(m)}$ 、式 (13.83b) から $\tilde{V}^{*(m)}$ を全く同じに要領で計算することができる .

²⁰ $y = 0$ 壁面上で流速成分 u, v はそれぞれ $u \sim y, v \sim y^2$ のようにふるまうので、曲線座標格子が壁面上で直交しないしはそれに近いところでこのような取り扱いが妥当と言える .

COEF, COMPP は ϕ の差分方程式の係数を設定し, その連立 1 次方程式を SOR 法または Chebyshev SLOR 法で解いて ϕ の値を求め, 静圧 p の値を更新するサブルーチンである. 解くべき ϕ の微分方程式 (13.81c) または (13.83c) は

$$\frac{\partial}{\partial \xi} \left(\Delta t \tilde{g}_{22} \frac{\partial \phi}{\partial \xi} - \Delta t \tilde{g}_{12} \frac{\partial \phi}{\partial \eta} \right) + \frac{\partial}{\partial \eta} \left(-\Delta t \tilde{g}_{12} \frac{\partial \phi}{\partial \xi} + \Delta t \tilde{g}_{11} \frac{\partial \phi}{\partial \eta} \right) = \frac{\partial \tilde{U}^*}{\partial \xi} + \frac{\partial \tilde{V}^*}{\partial \eta} \quad (13.93)$$

のようなもので, その差分方程式は形式的に次のように表すことができる.

$$\sum_{i'=-1}^1 \sum_{j'=-1}^1 c_{ij i' j'} \phi_{i+i', j+j'} = rhs_{ij} \quad (13.94)$$

より具体的に書けば次のようになる.

$$\begin{aligned} & (g_1)_{i+1,jU} (\phi_{i+1,j} - \phi_{ij}) + (g_2)_{i+1,jU} (\phi_{i+1,j+1} + \phi_{i,j+1} - \phi_{i+1,j-1} - \phi_{i,j-1}) \\ & - (g_1)_{ijU} (\phi_{ij} - \phi_{i-1,j}) - (g_2)_{ijU} (\phi_{i,j+1} + \phi_{i-1,j+1} - \phi_{i,j-1} - \phi_{i-1,j-1}) \\ & + (g_4)_{i,j+1V} (\phi_{i,j+1} - \phi_{ij}) + (g_3)_{i,j+1V} (\phi_{i+1,j+1} + \phi_{i+1,j} - \phi_{i-1,j+1} - \phi_{i-1,j}) \\ & - (g_4)_{ijV} (\phi_{ij} - \phi_{i,j-1}) - (g_3)_{ijV} (\phi_{i+1,j} + \phi_{i+1,j-1} - \phi_{i-1,j} - \phi_{i-1,j-1}) \\ & = \tilde{U}_{i+1,j}^* - \tilde{U}_{ij}^* + \tilde{V}_{i,j+1}^* - \tilde{V}_{ij}^* \end{aligned}$$

ただし $(g_1)_{ijU} = (\Delta t \tilde{g}_{22})_{ijU}$, $(g_2)_{ijU} = -(\Delta t \tilde{g}_{12})_{ijU}/4$, $(g_3)_{ijV} = -(\Delta t \tilde{g}_{12})_{ijV}/4$, $(g_4)_{ijV} = (\Delta t \tilde{g}_{11})_{ijV}$ で, これらの値は配列 gw に格納される. なお局所時間ステップ法では Δt は局所時間を取る.

ϕ の係数 $c_{ij i' j'}$ の値は, 図 13.9 に示すように 4 段階に分けて配列 co に入力される. この図には 2 つの場合, 領域内部の格子セル, 上流境界に隣接する格子セルの場合が取り上げられている. ϕ の 1 階微分は, 一般には中心差分 $(\phi_\xi)_{ij} = (\phi_{i+1,j} - \phi_{i-1,j})/2$ が用いられるが, 上流境界では片側差分 $(\phi_\xi)_{0j} = (-3\phi_{0j} + 4\phi_{1j} - \phi_{2j})/2$ が用いられる. 係数 $c_{0j 4j'}$ の値は空いている $co(0, j, 1, jp)$ に入れられる. 上方と下方の流路壁に隣接する格子セルの場合にも, 上流境界に隣接する格子セルの場合と同様の取り扱いが必要である. なお下流境界には $p_{i,j} = 0$ を与えているので, 下流境界に隣接する格子セルは領域内部のものと同じに取扱える.

ϕ の連立 1 次方程式は, ここでは趣向をかえ SOR 法または Chebyshev SLOR 法で解くことにする. SOR 法では, 奇数回の掃引は点 $ij = 00$ から $i_0 j_0$ まで, また偶数回の掃引は点 $i_0 - 1, j_0 - 1$ から 11 まで逆向きに行われる. 境界点の点緩和は, このように過緩和の効果が損なわれないように行われるべきである. ϕ の差分方程式 (13.94) の残差値は

$$Res_{ij} = \sum_{i'=-1}^1 \sum_{j'=-1}^1 c_{ij i' j'} \phi_{i+i', j+j'} - rhs_{ij}$$

から求められ, 点過緩和の修正計算は次式によって行われる.

$$\phi_{ij}^{(n+1)} = \phi_{ij}^{(n)} - \alpha Res_{ij} / c_{ij00}$$

$\alpha = 1.5$ は過緩和係数である.

非正常流れの場合には, 各時間ステップごとに, ϕ の差分方程式を十分に満足させる必要があり, ここではベクトルコンピュータの性能を十分に引き出せるよう, Chebyshev SLOR 法を用意した. この方法では緩和計算は奇数列 ($i=1,3,5, \dots$) と偶数列 ($i=0,2,4, \dots$) の 2 段階に分け, 各段階の計算では 3 重対角行列

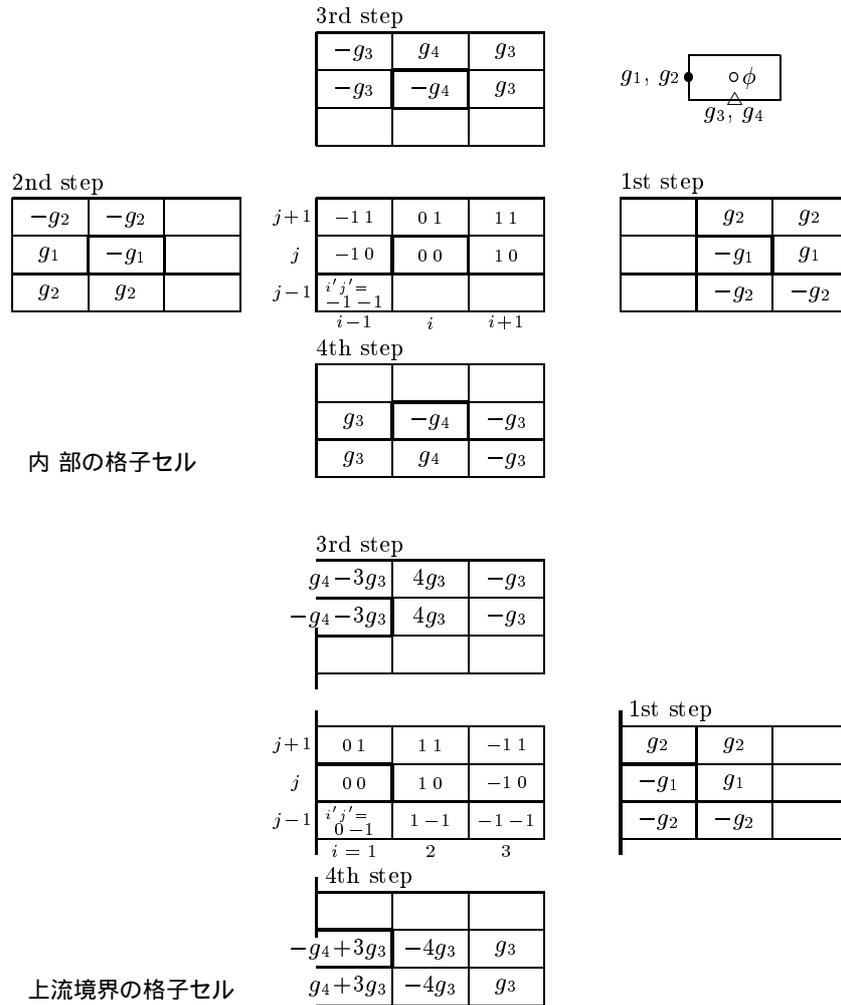
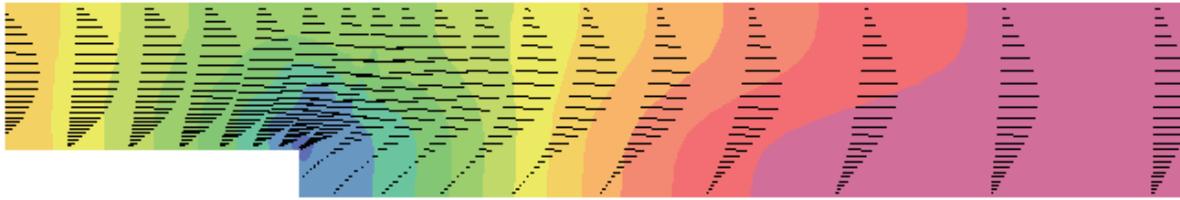


図 13.9: 圧力差分方程式の係数の設定

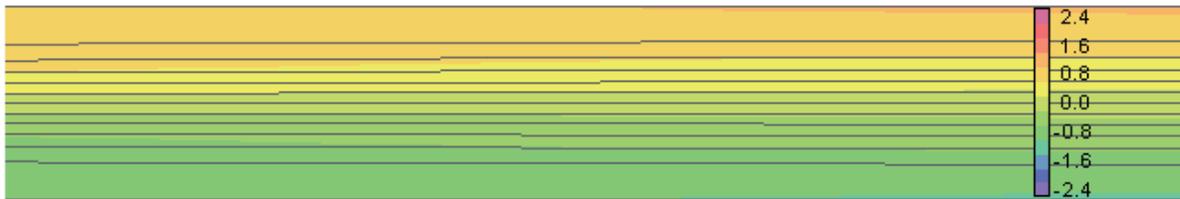
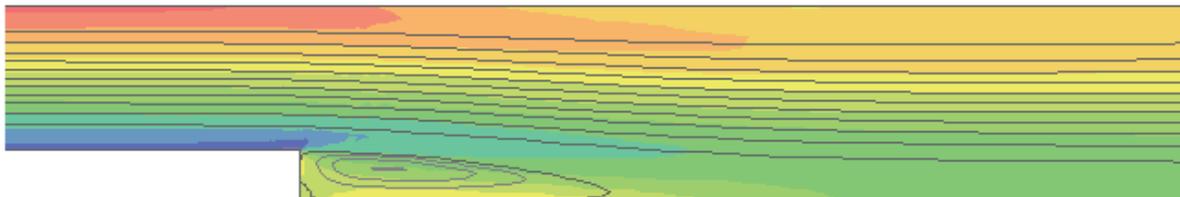
の連立 1 次方程式の係が Gauss 消去法で解かれる．1 つの列の連立 1 次方程式は

$$\begin{aligned}
 & \alpha c_{i0\ 00} \phi_{i0}^{(n+1)} + c_{i0\ 01} \phi_{i1}^{(n+1)} + c_{i0\ 02} \phi_{i2}^{(n+1)} \\
 &= r_{i0} - \sum_{j'=0}^2 (c_{i0\ -1j'} \phi_{i-1,j'}^{(n)} + c_{i0\ 1j'} \phi_{i+1,j'}^{(n)}) - (1-\alpha) c_{i0\ 00} \phi_{i0}^{(n)} \\
 & c_{ij\ 0-1} \phi_{i,j-1}^{(n+1)} + \alpha c_{ij\ 00} \phi_{ij}^{(n+1)} + c_{ij\ 01} \phi_{i,j+1}^{(n+1)} \\
 &= r_{ij} - \sum_{j'=-1}^1 (c_{ij\ -1j'} \phi_{i-1,j+j'}^{(n)} + c_{ij\ 1j'} \phi_{i+1,j+j'}^{(n)}) - (1-\alpha) c_{ij\ 00} \phi_{ij}^{(n)} \quad (j = 1, 2, \dots, j_0) \\
 & c_{ij_0\ 0-2} \phi_{i,j_0-2}^{(n+1)} + c_{ij_0\ 0-1} \phi_{i,j_0-1}^{(n+1)} + \alpha c_{ij_0\ 00} \phi_{i,j_0}^{(n+1)} \\
 &= r_{ij_0} - \sum_{j'=-2}^0 (c_{ij_0\ -1j'} \phi_{i-1,j_0+j'}^{(n)} + c_{ij_0\ 1j'} \phi_{i+1,j_0+j'}^{(n)}) - (1-\alpha) c_{ij_0\ 00} \phi_{i,j_0}^{(n)}
 \end{aligned}$$

のようになる．ただし α は加速のパラメータである．なお上流境界の連立 1 次方程式は境界条件が入り多少異なる



PRESSURE DISTRIBUTION AND VELOCITY VECTORS

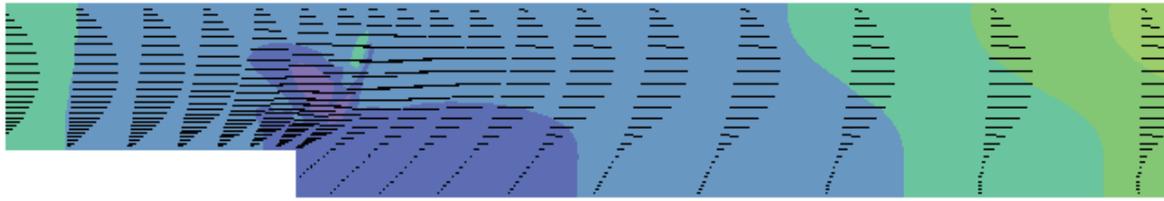


VORTICITY DISTRIBUTION AND STREAMLINES

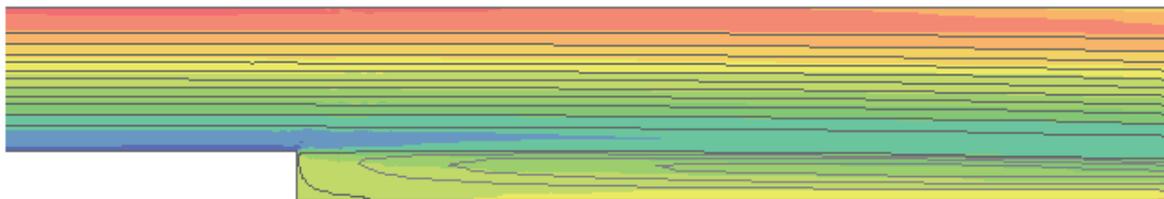
図 13.10: バックステップ流路の流れ ($Re = 100$)

るものになる。係数 $c_{ij'j'}$ の値は一般に配列要素 $co(i, j, ip, jp)$ に入れられるが、 $c_{i0'j'2}$ は $co(i, 0, ip, -1)$ に、 $c_{ij'0'j'-2}$ は $co(i, j, 0, ip, 1)$ に入れられる。

これらの連立 1 次方程式はサブルーチン GAUSSP を引用して解かれる。GAUSSP では奇数列または偶数列の連立 1 次方程式が一まとめに同時に解かれる。また各連立 1 次方程式は、次のように係数行列が変形 3 重対角行列になっている。



PRESSURE DISTRIBUTION AND VELOCITY VECTORS



VORTICITY DISTRIBUTION AND STREAMLINES

図 13.12: バックステップ流路の流れ ($Re = 1000$)

res_{NS} と res_P の大きさが同等になるように決められたものである．なお Chebyshev SLOR 法はベクトル計算機を利用するとき威力を発揮する．

次に計算結果について述べる．ここにはレイノルズ数 100, 500, 1000 の結果を示すが、これらはいずれも $\Delta t = 1.0$, $mode = 2$ と置いて計算したものである．サブルーチン COMPU1 では Chakravarthy-Osher 型の TVD スキームを用い $\theta = 1.0$, $\beta = 0.5$ と置き、またサブルーチン COMPP では SOR 法を用いる場合には $\alpha = 1.5$, $\beta = 0.5$ 、Chebyshev SLOR 法の場合にはレイノルズ数 100 では $\alpha = \beta = 1.0$ 、レイノルズ数 500 と 1000 では $\alpha = 1.25$, $\beta = 0.75$ としている．図 13.10, 13.11, 13.12 は SOR 法の場合の結果である．これらの図には、上から圧力分布と速度ベクトルプロファイル、渦度分布と流線が示されている．計算は $x = -10 \sim 70.91$ の範囲で行われたが、図には全領域ではなく $x = -6 \sim 18$ と $x = 18 \sim 42$ の範囲が示されている．圧力分布は例えば $Re = 100$ の場合には、一つの色の範囲は 0.015 で、 $p = 0.21$ の色は $p = 0.2025 \sim 0.2175$ の領域に塗られている．流線は下壁面で流れ関数 $\psi = 0$ 上壁面で 3 とし、 $\Delta\psi = 0.25$

おきのものが描かれている．また特に剥離域内では $\Delta\psi = 0.01$ おきに描かれている．収束判定は前章同様 $res_{NS} < 0.0001$ かつ $res_P < 0.0001$ で行っている． $Re = 1000$ の場合の解の収束状況を下表に示す．

表 13.6: バックステップ流路流れの解の収束状況 ($Re = 1000$)

n	SOR			Chebyshev SLOR		
	res_{NS}	res_P	$\Delta flow$	res_{NS}	res_P	$\Delta flow$
100	736	436	25693	481	189	29678
1000	41	50	5095	157	79	-6564
2000	12	16	3435	31	25	4669
2340	9	10	2608			
2460				9	10	4402

収束解を得るまでの反復数 n は， Δ 形陰解法の適用によって大幅に減少しているが，時間間隔 Δt やプログラム中の係数 α などにもより，それらの値も適切に設定しないと効率よく収束解を得ることはできない．次表は SOR 法を用いた場合の時間間隔と収束に要する反復数の関係を示しものである．

表 13.7: 時間間隔 Δt と収束反復数 n

Re	Δt	n	x_{rap}	p_{max}	p_{min}
100	0.1	2580	7.2	0.245	0.064
	0.25	1180	7.2	0.240	0.058
	0.63	900	7.1	0.242	0.064
	1.0	1366	7.2	0.242	0.062
	2.5	3562	7.2	0.242	0.062
	4.0	*	7.2	0.243	0.062
500	0.25	**			
	0.4	3100	21.8	-0.017	-0.159
	0.63	1580	21.7	-0.018	-0.159
	1.0	1744	23.5	-0.016	-0.155
	1.6	2713	23.5	-0.016	-0.155
	2.5	4156	23.3	-0.016	-0.155
	4.0	*	23.7	-0.015	-0.151
1000	0.63	**			
	1.0	2348	30.7	-0.038	-0.130
	1.6	2917	38.3	-0.033	-0.129
	2.5	3951	38.8	-0.033	-0.130
	4.0	*	41.1	-0.034	-0.125

表中の*印は $n = 5000$ 回まで計算した未収束の結果，また**印は残差値の状況から収束解が得られないと判断されたことを示す．この表にはまた再付着距離，図示領域の最大最小圧力値も示してある．これらは本来時間間隔 Δt には無関係であるべきものであるが，表に示すように必ずしも無関係とはいえない．次表はこれらの値を更に， $\Delta t = 1.0$ に対し，本章の Chebyshev SLOR 法と前章の長方形格子 (RG) の計算結果と比較したものである．

このことについて多くの読者は収束判定条件があまいのではと思われるかもしれない．しかし結論を先に言えば判定条件は問題ないが，流路の角近傍の精度が不十分ということである．このプログラムでは比較的粗い格子に高次スキームを用い精度を確保しようとしているが，高次スキームも角近傍の極端に歪んだ格子に対しては精度を発揮できない．この曲線座標格子では，上記の判定条件で計算が収束した後に更に計算を継続すると，ある時点で残差が突然増大しその後再付着距離のかなり長い解に収束するかまたは発散するという不都合が生じる．その原因はおそらく $\tilde{U}_{36,0}$ の符号が正から負へ切替わることによる．通常

表 13.8: バックステップ流路流れの再付着距離，最大最小圧力の比較

Re	RG	SOR			Chebyshev SLOR		
	x_{rap}	x_{rap}	p_{max}	p_{min}	x_{rap}	p_{max}	p_{min}
100	7.3	7.2	0.242	0.062	7.1	0.242	0.064
500	20.8	23.5	-0.016	-0.155	22.5	-0.013	-0.143
1000		30.7	-0.038	-0.130	31.4	-0.033	-0.124

点では 13.4 節に詳しく述べたようにこの符号の切替わりにより対流項の値が急変することはないが，角の直後のこの点ではするということである．上記のプログラムでは $\tilde{U}_{36,0}$ の対流項を 2 次の TVD 補間式を微分した式を用いて計算するようにし，とりあえずこの問題に対処している．しかし正攻法はより細かい格子を用いることであろう．

13.9 反変速度を用いた拡散項の簡潔な表示

初めに層流の計算や乱流の直接シミュレーションに用いられる粘性項 $\nu \nabla^2 \mathbf{u}$ を考える．この粘性項に左から $J \nabla \xi_l$ を演算したものは，式 (13.70)–(13.74) を用いれば次のようになる²¹．

$$D_l = J \nabla \xi_l \cdot \nu \nabla^2 \mathbf{u} = -\nu J \nabla \xi_l \cdot \nabla \times \zeta = -\nu \epsilon_{lij} \frac{\partial}{\partial \xi_i} \left(\frac{\partial \mathbf{x}}{\partial \xi_j} \cdot \zeta \right) = -\nu \epsilon_{lij} \frac{\partial}{\partial \xi_i} (\tilde{g}_{jk} \tilde{Z}_k) \quad (13.95)$$

これは 6 項からなる簡潔な式である．渦度 \tilde{Z}_k は式 (13.72) から求められる．すなわち

$$\tilde{Z}_k = \epsilon_{kmn} \frac{\partial}{\partial \xi_m} (\tilde{g}_{ni} \tilde{U}_i)$$

なお反変渦度 Z_k は立方体セルの ξ_k 軸に平行な稜の中心点に定義される．また 2 次元の場合には渦度は紙面に垂直な成分 ζ のみとなり，粘性項は次のようになる．

$$D_1 = -\nu \frac{\partial \zeta}{\partial \eta}, \quad D_2 = \nu \frac{\partial \zeta}{\partial \xi} \quad (13.96)$$

$$\zeta = \frac{1}{J} \left\{ \frac{\partial}{\partial \xi} (\tilde{g}_{21} \tilde{U} + \tilde{g}_{22} \tilde{V}) - \frac{\partial}{\partial \eta} (\tilde{g}_{11} \tilde{U} + \tilde{g}_{12} \tilde{V}) \right\}$$

次に乱流の計算に用いられるより一般的な粘性拡散項を考える．この粘性項は粘性応力テンソル \mathbf{II} を用い一般に次のように表わされる²²．

$$D_l = J \nabla \xi_l \cdot (\nabla \cdot \mathbf{II}) = J \frac{\partial \xi_l}{\partial x_n} \frac{\partial \tau_{mn}}{\partial x_m} = \frac{\partial}{\partial \xi_i} \left(J \frac{\partial \xi_i}{\partial x_m} \frac{\partial \xi_l}{\partial x_n} \tau_{mn} \right) - J \frac{\partial \xi_i}{\partial x_m} \tau_{mn} \frac{\partial}{\partial \xi_i} \frac{\partial \xi_l}{\partial x_n}$$

$$= \frac{\partial}{\partial \xi_i} (JT^{il}) + \{i^l_j\} JT^{ij} \quad (13.97)$$

ただし τ_{mn} は \mathbf{II} の成分， T^{ij} はその反変成分で次のように表される．

$$T^{ij} = \frac{\partial \xi_i}{\partial x_m} \tau_{mn} \frac{\partial \xi_j}{\partial x_n}$$

²¹ 第 2 式から第 3 式へは $\nabla \times (\nabla \times \mathbf{a}) = \nabla (\nabla \cdot \mathbf{a}) - \nabla^2 \mathbf{a}$ なる関係，第 3 式から第 4 式へは式 (13.65) と (13.71)，また第 4 式から第 5 式へは式 (13.73) と (13.74) が用いられた．

²² 式 (13.64)，(13.65)，(13.78) を用いる．

乱流を渦粘性近似を置いて計算する場合には，応力テンソル成分は次式で与えられる．

$$\tau_{mn} = \nu_{\text{eff}} \left(\frac{\partial u_n}{\partial x_m} + \frac{\partial u_m}{\partial x_n} \right) - \frac{2}{3} \delta_{mn} k \quad (13.98)$$

ただし $\nu_{\text{eff}} = \nu + \nu_t$ ， ν_t は渦粘性係数， k は乱れの運動エネルギーである．このとき応力テンソルの反変成分 JT^{ij} は次のようになる．

$$\begin{aligned} JT^{ij} &= J \left[\nu_{\text{eff}} \frac{\partial \xi_i}{\partial x_m} \left\{ \frac{\partial}{\partial x_m} \left(u_n \frac{\partial \xi_j}{\partial x_n} \right) - u_n \frac{\partial^2 \xi_j}{\partial x_n \partial x_m} \right\} \right. \\ &\quad \left. + \nu_{\text{eff}} \frac{\partial \xi_j}{\partial x_n} \left\{ \frac{\partial}{\partial x_n} \left(u_m \frac{\partial \xi_i}{\partial x_m} \right) - u_m \frac{\partial^2 \xi_i}{\partial x_m \partial x_n} \right\} - \frac{2}{3} g^{ij} k \right] \\ &= \nu_{\text{eff}} J \frac{\partial \xi_i}{\partial x_m} \left(\frac{\partial U_j}{\partial x_m} - U_k \frac{\partial}{\partial \xi_k} \frac{\partial \xi_j}{\partial x_m} \right) + \nu_{\text{eff}} J \frac{\partial \xi_j}{\partial x_n} \left(\frac{\partial U_i}{\partial x_n} - U_k \frac{\partial}{\partial \xi_k} \frac{\partial \xi_i}{\partial x_n} \right) - \frac{2}{3} g^{ij} k \\ &\equiv \nu_{\text{eff}} (A^{ij} + A^{ji}) - \frac{2}{3} \tilde{g}^{ij} k \quad (i, j = 1, 2, 3) \end{aligned} \quad (13.99)$$

ここに

$$\begin{aligned} A^{ij} &= J \frac{\partial \xi_i}{\partial x_n} \left(\frac{\partial U_j}{\partial x_n} - U_k \frac{\partial}{\partial \xi_k} \frac{\partial \xi_j}{\partial x_n} \right) \\ &= J \frac{\partial \xi_i}{\partial x_n} \left(\frac{\partial \xi_m}{\partial x_n} \frac{\partial U_j}{\partial \xi_m} + U_k \frac{\partial \xi_m}{\partial x_n} \{ {}^j_m k \} \right) \equiv \frac{\partial \xi_i}{\partial x_n} b_n^j \\ &= \tilde{g}^{im} \left(\frac{\partial U_j}{\partial \xi_m} + \{ {}^j_m k \} U_k \right) \equiv \tilde{g}^{im} a_m^j \end{aligned}$$

拡散項は，渦粘性近似のもとでは，式 (13.97) と式 (13.99) から次のようになる²³．

$$\begin{aligned} D_l &= \frac{\partial}{\partial \xi_i} \left\{ \nu_{\text{eff}} (A^{il} + A^{li}) - \frac{2}{3} \tilde{g}^{il} k \right\} + \{ {}^l_{ij} \} \left\{ \nu_{\text{eff}} (A^{ij} + A^{ji}) - \frac{2}{3} \tilde{g}^{ij} k \right\} \\ &= \nu_{\text{eff}} \frac{\partial}{\partial \xi_i} (A^{il} - A^{li}) + \frac{\partial \nu_{\text{eff}}}{\partial \xi_i} (A^{il} + A^{li}) - \frac{2}{3} \tilde{g}^{li} \frac{\partial k}{\partial \xi_i} \quad (l = 1, 2, 3) \end{aligned} \quad (13.100)$$

ただし

$$\begin{aligned} A^{ij} &= \tilde{g}^{im} a_m^j = \frac{\partial \xi_i}{\partial x_n} b_n^j \quad (i, j = 1, 2, 3) \quad (13.101) \\ a_m^j &= \frac{\partial U_j}{\partial \xi_m} + \{ {}^j_m k \} U_k, \quad b_n^j = \frac{\partial}{\partial \xi_k} \left(\frac{\partial \xi_k}{\partial x_n} \tilde{U}_j - \frac{\partial \xi_j}{\partial x_n} \tilde{U}_k \right) \end{aligned}$$

である²⁴．なお A^{11} ， a_m^1 ， b_n^1 は \tilde{U}_1 と同じ立方体セルの面中心に， A^{12} ， A^{21} は Z_3 と同じ稜の中心に定義される．

²³ 次のように計算する．

$$\{ {}^l_{ij} \} A^{ij} = \frac{\partial \xi_i}{\partial x_n} \{ {}^l_{ij} \} b_n^j = - \left(\frac{\partial}{\partial \xi_j} \frac{\partial \xi_l}{\partial x_n} \right) b_n^j = - \frac{\partial}{\partial \xi_j} A^{lj}, \quad \{ {}^l_{ij} \} \tilde{g}^{ij} = -J \frac{\partial \xi_i}{\partial x_n} \frac{\partial}{\partial \xi_i} \frac{\partial \xi_l}{\partial x_n} = - \frac{\partial}{\partial \xi_i} \tilde{g}^{il}$$

²⁴ $\partial b_n^j / \partial \xi_j = 0$ ，したがって b_n^j は $\xi_j = \text{const.}$ 線に沿って一定になる量である．