

```

C-----
C 立方体周りの三次元ポテンシャル流れ計算プログラム
C-----

parameter (if=21, jf=21, kf=21)
dimension p (if, jf, kf), u (if, jf, kf), v (if, jf, kf)
dimension w (if, jf, kf)

C
data omega/1.5/
data ia, ib/8, 14/
data ja, jb/8, 14/
data ka, kb/8, 14/

C----- 格子間隔設定 -----
dx=0.1
dy=0.1
dz=0.1
dx2=dx*dx
dy2=dy*dy
dz2=dz*dz
c0=2.0/dx2+2.0/dy2+2.0/dz2

C----- 配列初期設定 -----
do k=1, kf
do j=1, jf
do i=1, if
u (i, j, k)=0.0
v (i, j, k)=0.0
w (i, j, k)=0.0
p (i, j, k)=0.0
enddo
enddo
enddo

C----- 入口, 出口のスカラーポテンシャルを設定 -----
do k=1, kf
do j=1, jf
p (1, j, k)=0.0
p (if, j, k)=1.0
enddo
enddo

C----- 反復計算ルーチン -----
na=0
100 na=na+1
rmax=0.0

C
do ii=2, if-1
i=ii
if (mod (na, 2).eq.0) i=if+1-i
do jj=1, jf
j=jj
if (mod (na, 2).eq.0) j=jf+1-j
do kk=1, kf
k=kk
if (mod (na, 2).eq.0) k=kf+1-k

C----- 立方体内部および表面を検出 -----
if (i.gt.ia.and.i.lt.ib.and.
& j.gt.ja.and.j.lt.jb.and.
& k.gt.ka.and.k.lt.kb) go to 10

C

```

```

im1=i-1
ip1=i+1
jm1=j-1
jp1=j+1
km1=k-1
kp1=k+1
C----- 外部境界における境界条件を処理 -----
  if (j. eq. 1) jm1=2
  if (j. eq. jf) jp1=jf-1
  if (k. eq. 1) km1=2
  if (k. eq. kf) kp1=kf-1
C----- 立方体表面における境界条件を処理 -----
  if (j. ge. ja. and. j. le. jb. and.
&    k. ge. ka. and. k. le. kb) then
  if (i. eq. ib) im1=im1+2
  if (i. eq. ia) ip1=ip1-2
  endif
C
  if (i. ge. ia. and. i. le. ib. and.
&    k. ge. ka. and. k. le. kb) then
  if (j. eq. jb) jm1=jm1+2
  if (j. eq. ja) jp1=jp1-2
  endif
C
  if (i. ge. ia. and. i. le. ib. and.
&    j. ge. ja. and. j. le. jb) then
  if (k. eq. kb) km1=km1+2
  if (k. eq. ka) kp1=kp1-2
  endif
C----- SOR法による計算 -----
  r= (p(im1, j, k)-2.0*p(i, j, k)+p(ip1, j, k))/dx2
&   +(p(i, jm1, k)-2.0*p(i, j, k)+p(i, jp1, k))/dy2
&   +(p(i, j, km1)-2.0*p(i, j, k)+p(i, j, kp1))/dz2
  p(i, j, k)=p(i, j, k) + omega*r/c0
  rmax=amax1(rmax, abs(r))
10 continue
  enddo
  enddo
  enddo
C
  if (na. eq. 1) rmax1=rmax
  rmax=rmax/rmax1
  write(6, 20) na, rmax
20 format(5x, 'na=', i3, 5x, 'rmax=', f10. 7)
C----- 収束判定 -----
  if (na. lt. 1000 .and. rmax. gt. 1.0D-5) go to 100
C----- 流速を計算 -----
  do i=1, if
  do j=1, jf
  do k=1, kf
C
  if (i. gt. ia. and. i. lt. ib. and.
&    j. gt. ja. and. j. lt. jb. and.
&    k. gt. ka. and. k. lt. kb) go to 30
C
  dx1=dx*2.0D0

```

```

dy1=dy*2.0D0
dz1=dz*2.0D0
C
  im1=i-1
  ip1=i+1
  jm1=j-1
  jp1=j+1
  km1=k-1
  kp1=k+1
C
  if (i.eq.1) then
    im1=1
    dx1=dx
  endif
  if (i.eq.if) then
    ip1=if
    dx1=dx
  endif
  if (j.ge.ja.and.j.le.jb.and.
&   k.ge.ka.and.k.le.kb) then
    if (i.eq.ib) then
      im1=im1+1
      dx1=dx

    endif
    if (i.eq.ia) then
      ip1=ip1-1
      dx1=dx
    endif
  endif
C
  if (j.eq.1) then
    jm1=1
    dy1=dy
  endif
  if (j.eq.jf) then
    jp1=jf
    dy1=dy
  endif
  if (i.ge.ia.and.i.le.ib.and.
&   k.ge.ka.and.k.le.kb) then
    if (j.eq.jb) then
      jm1=jm1+1
      dy1=dy
    endif
    if (j.eq.ja) then
      jp1=jp1-1
      dy1=dy
    endif
  endif
C
  if (k.eq.1) then
    km1=1
    dz1=dz
  endif
  if (k.eq.kf) then

```

```

    kp1=kf
    dz1=dz
    endif
    if (i.ge.ia.and.i.le.ib.and.
&    j.ge.ja.and.j.le.jb) then
    if (k.eq.kb) then
    km1=km1+1
    dz1=dz
    endif
    if (k.eq.ka) then
    kp1=kp1-1
    dz1=dz
    endif
    endif
C
    u(i,j,k)=(p(ip1,j,k)-p(im1,j,k))/dx1
    v(i,j,k)=(p(i,jp1,k)-p(i,jm1,k))/dy1
    w(i,j,k)=(p(i,j,kp1)-p(i,j,km1))/dz1
30 continue
    enddo
    enddo
    enddo
C----- 計算データの後処理 -----
    call graphic
C
    stop
    end

```