Appendix 1 (Drifter Subroutine)

```
SUBROUTINE DRIFTER
C VERSION(06/21/99)
C
$ INCLUDE comdeck
C
C——————————————————————————————————————
C THIS SUBROUTINE INTEGRATES PARTICLES FORWARD TO THE
C PRESENT INTERNAL TIME STEP
C
C P here is dimensioned as particle number, parameter
C where the first par. is I from the grid corner
C second par. is J from the grid corner
C third par. is K from the level bottom
C fourth par. is DX of the particle
C fifth par. is DY of the particle
C sixth par. is DZ of the particle
C seventh par. is age of the particle
C eighth par. is density of the particle
C
C DX,DY,DZ are all 0 at the start of a grid (their origin)
C and 1 at the boundary to the next grid, or level
```

```
C————————————————————————————————
C
DIMENSION BUP(NUMP),BUP1(NUMP),BUP2(NUMP)
.,TUP(NUMP),TUP1(NUMP),TUP2(NUMP),UP(NUMP)
.,BVP(NUMP),BVP1(NUMP),BVP2(NUMP)
.,TVP(NUMP),TVP1(NUMP),TVP2(NUMP),VP(NUMP)
.,BWP(NUMP),BWP1(NUMP),BWP2(NUMP)
.,TWP(NUMP),TWP1(NUMP),TWP2(NUMP),WP(NUMP)
.,PH2(NUMP),PH1(NUMP),PH3(NUMP)
.,ut(im,jm,kb),vt(im,jm,kb),wt(im,jm,kb)
REAL MOVX,MOVY,MOVZ,clev,dispx,dispy,dispz
INTEGER LK,IP,JP,ZKP,PK,IPF,JPF,ZKPF,SEQ,IPR
INTEGER pong,stick,ob,pstart
IUTR=105
IUTRS=104
IURES=110
OPEN(IUTR,FILE='track.dat',access='append')
OPEN(IUTRS,FILE='track.stats',access='append')
OPEN(IURES,FILE='track.res',access='append')
c the number of time steps between outputs IPR
IPR=1440
```

```
c switch to elastically scatter off rigid boundaries
PONG=1
c switch to make particles stick to rigid boundaries (go aground)
stick=0
c switch to make particles stick to open boundaries
ob=1
c floating/sinking rate (relative density) parameter
clev=1.0
c switch to start moving particles
pstart=0
c constant eddy diffusion for random walk
dispx=0.005
dispy=0.005
dispz=0.0
do 3 i=1,im
do 3 j=1,jm
do 3 k=1,kb
c dummy u,v,w for sub
ut(i,j,k)=u(i,j,k)*dum(i,j)
vt(i,j,k)=v(i,j,k)*dvm(i,j)
wt(i,j,k)=w(i,j,k)
```

```
3 continue
DO 1 LK=1,NUMP
c initialize
BUP(LK)=0.0
BUP1(LK)=0.0
BUP2(LK)=0.0
TUP(LK)=0.0
TUP1(LK)=0.0
TUP2(LK)=0.0
UP(LK)=0.0
BVP(LK)=0.0
BVP1(LK)=0.0
BVP2(LK)=0.0
TVP(LK)=0.0
TVP1(LK)=0.0
TVP2(LK)=0.0
VP(LK)=0.0
BWP(LK)=0.0
BWP1(LK)=0.0
```

```
BWP2(LK)=0.0
TWP(LK)=0.0
TWP1(LK)=0.0
TWP2(LK)=0.0
WP(LK)=0.0
PH1(LK)=0.0
PH2(LK)=0.0
PH3(LK)=0.0
1 CONTINUE
MOVX=0.0
MOVY=0.0
MOVZ=0.0
C print *, 'start of drifter',nofi
IF (NOFI .EQ. 0) THEN
c initialize particles positions
seedx=-1
seedy=-2
seedz=-3
lk=0
DO 5 i=1,im
DO 5 j=1,jm
```

```
do 5 k=1,10
if (fsm(i,j) .eq. 1.0) then
c print *,fsm(i,j)
if (lk .le. nump) then
P(LK,1)=float(i)
P(LK,2)=float(j)
P(LK,3)=float(k)
P(LK,4)=0.5
C0.0+mod(lk,48)/48.0
c0.5
P(LK,5)=0.5
C0.0+mod(lk/48,48)/48.0
c0.5
P(LK,6)=.2
P(LK,7)=1.0
P(LK,8)=0.0
P(LK,9)=0.0
P(LK,10)=393000.0
lk=lk+1
endif
```

```
endif
5 CONTINUE
if (lk .lt. nump) then
do 7 llk=lk,nump
P(lLK,1)=28
P(lLK,2)=39
P(lLK,3)=2.0
P(lLK,4)=0.0
P(lLK,5)=.7
P(lLK,6)=.2
P(lLK,7)=1.0
P(lLK,8)=0.0
P(lLK,9)=0.0
P(lLK,10)=393000.0
7 continue
endif
NOFI=1
write(IUTRS,*) NUMP
write(IUTRS,*) PAR
write(IUTRS,*) INT
write(IUTRS,*) DTI
```

```
write(IUTRS,*) IPR
ENDIF
C
C For every particle main loop
SEQ=INT
DO 10 LK=1,NUMP
C check for start time
IF (ifix(p(lk,10)) .le. INT) then
p(lk,9)=1.0
pstart=1
endif
C set old i,j,k
IP=IFIX(P(LK,1))
JP=IFIX(P(LK,2))
ZKP=IFIX(P(LK,3))
C linerly interp h1
IF (P(LK,5) .GT. 0.5) THEN
ph1(LK)=H1(IP,JP)+(P(LK,5)-0.5)
&*(H1(IP,JP+1)-H1(IP,JP))
ELSE
```

```
ph1(LK)=H1(IP,JP-1)+(P(LK,5)+0.5)
&*(H1(IP,JP)-H1(IP,JP-1))
ENDIF
C linerly interp h2
IF (P(LK,4) .GT. 0.5) THEN
ph2(LK)=H2(IP,JP)+(P(LK,4)-0.5)
&*(H2(IP+1,JP)-H2(IP,JP))
ELSE
ph2(LK)=H2(IP-1,JP)+(P(LK,4)+0.5)
&*(H2(IP,JP)-H2(IP-1,JP))
ENDIF
C linerly interp sigma
ph3(LK)=DZ(ZKP)
&+2.0*(P(LK,6)*DZ(ZKP)+DZ(ZKP-1))
&*(DZ(ZKP)-DZ(ZKP-1))/(DZ(ZKP)+DZ(ZKP-1))
C linerly interp U
C if dy is further than half way through the grid
C then the closest neighbors are IP+1,JP IP+1,JP+1
C IP,JP IP,JP+1
IF (P(LK,5) .GT. 0.5) THEN
C if dz is lt .5 the lower level is ZKP+1 and the upper is ZKP
```

```
C and the particle is in the ZKP level
IF (P(LK,6) .LT. 0.5) THEN
C calculate the lower two linear interp U's
bup1(LK)=UT(IP+1,JP,ZKP+1)+(P(LK,5)-0.5)
&*(UT(IP+1,JP+1,ZKP+1)-UT(IP+1,JP,ZKP+1))
bup2(LK)=UT(IP,JP,ZKP+1)+(P(LK,5)-0.5)
&*(UT(IP,JP+1,ZKP+1)-UT(IP,JP,ZKP+1))
C combine by linearly interp those two u's
bup(LK)=bup2(LK) + P(LK,4)
&*(bup1(LK)-bup2(LK))
C do the same for the level above
tup1(LK)=UT(IP+1,JP,ZKP)+(P(LK,5)-0.5)
&*(UT(IP+1,JP+1,ZKP)-UT(IP+1,JP,ZKP))
tup2(LK)=UT(IP,JP,ZKP)+(P(LK,5)-0.5)
&*(UT(IP,JP+1,ZKP)-UT(IP,JP,ZKP))
tup(LK)=tup2(LK) + P(LK,4)*(tup1(LK)-tup2(LK))
C combine the upper and lower by linear interp
up(LK)=bup(LK)+(2.0*P(LK,6)*DZ(ZKP)+DZ(ZKP+1))
&*(tup(LK)-bup(LK))/(DZ(ZKP)+DZ(ZKP+1))
C if dz is ge .5 the lower grid is ZKP and the upper grid is ZKP-1
```

Appendix 1 (Continued)

```
C the particle is still in level ZKP

ELSE

C calculate the lower two linear interp U's

bup1(LK)=UT(IP+1,JP,ZKP)+(P(LK,5)-0.5)

&*(UT(IP+1,JP+1,ZKP)-UT(IP+1,JP,ZKP))

bup2(LK)=UT(IP,JP,ZKP)+(P(LK,5)-0.5)

&*(UT(IP,JP+1,ZKP)-UT(IP,JP,ZKP))

C combine by linearly interp those two u's

bup(LK)=bup2(LK) + P(LK,4)

&*(bup1(LK)-bup2(LK))

C do the same for the level above

tup1(LK)=UT(IP+1,JP,ZKP-1)+(P(LK,5)-0.5)

&*(UT(IP+1,JP+1,ZKP-1)-UT(IP+1,JP,ZKP-1))

tup2(LK)=UT(IP,JP,ZKP-1)+(P(LK,5)-0.5)

&*(UT(IP,JP+1,ZKP-1)-UT(IP,JP,ZKP-1))

tup(LK)=tup2(LK) + P(LK,4)*(tup1(LK)-tup2(LK))

C combine the upper and lower via linear interp.

up(LK)=bup(LK)+(2.0*(P(LK,6)-0.5)*DZ(ZKP)+DZ(ZKP-1))

&*(tup(LK)-bup(LK))/(DZ(ZKP)+DZ(ZKP-1))

ENDIF

C if dy is less than or equal to half the grid dist
```

```
C then the closet neighbors are JP-1
ELSE
C if dz is lt .5 the lower grid is ZKP+1 and the upper is ZKP
C and the particle is in the ZKP level
IF (P(LK,6) .LT. 0.5) THEN
C calculate the lower two linear interp U's
bup1(LK)=UT(IP+1,JP-1,ZKP+1)+(P(LK,5)+0.5)
&*(UT(IP+1,JP,ZKP+1)-UT(IP+1,JP-1,ZKP+1))
bup2(LK)=UT(IP,JP-1,ZKP+1)+(P(LK,5)+0.5)
&*(UT(IP,JP,ZKP+1)-UT(IP,JP-1,ZKP+1))
C combine by linearly interp those two u's
bup(LK)=bup2(LK) + P(LK,4)
&*(bup1(LK)-bup2(LK))
C do the same for the level above
tup1(LK)=UT(IP+1,JP-1,ZKP)+(P(LK,5)+0.5)
&*(UT(IP+1,JP,ZKP)-UT(IP+1,JP-1,ZKP))
tup2(LK)=UT(IP,JP-1,ZKP)+(P(LK,5)+0.5)
&*(UT(IP,JP,ZKP)-UT(IP,JP-1,ZKP))
tup(LK)=tup2(LK) + P(LK,4)*(tup1(LK)-tup2(LK))
C combine the upper and lower by linear interp
```

```
up(LK)=bup(LK)+(2.0*P(LK,6)*DZ(ZKP)+DZ(ZKP+1))
&*(tup(LK)-bup(LK))/(DZ(ZKP)+DZ(ZKP+1))
C if dz is ge .5 the lower grid is ZKP and the upper grid is ZKP-1
C the particle is still in level ZKP
ELSE
C calculate the lower two linear interp U's
bup1(LK)=UT(IP+1,JP-1,ZKP)+(P(LK,5)+0.5)
&*(UT(IP+1,JP,ZKP)-UT(IP+1,JP-1,ZKP))
bup2(LK)=UT(IP,JP-1,ZKP)+(P(LK,5)+0.5)
&*(UT(IP,JP,ZKP)-UT(IP,JP-1,ZKP))
C combine by linearly interp those two u's
bup(LK)=bup2(LK) + P(LK,4)
&*(bup1(LK)-bup2(LK))
C do the same for the level above
tup1(LK)=UT(IP+1,JP-1,ZKP-1)+(P(LK,5)+0.5)
&*(UT(IP+1,JP,ZKP-1)-UT(IP+1,JP-1,ZKP-1))
tup2(LK)=UT(IP,JP-1,ZKP-1)+(P(LK,5)+0.5)
&*(UT(IP,JP,ZKP-1)-UT(IP,JP-1,ZKP-1))
tup(LK)=tup2(LK) + P(LK,4)*(tup1(LK)-tup2(LK))
C combine the upper and lower via linear interp.
up(LK)=bup(LK)+(2.0*(P(LK,6)-0.5)*DZ(ZKP)+DZ(ZKP-1))
```

```
&*(tup(LK)-bup(LK))/(DZ(ZKP)+DZ(ZKP-1))
ENDIF
ENDIF
C linerly interp V
C if dx is further than half way through the grid
C then the closest neighbors are IP+1,JP IP+1,JP+1
C IP,JP IP,JP+1
IF (P(LK,4) .GT. 0.5) THEN
C if dz is lt .5 the lower grid is ZKP+1 and the upper is ZKP
C and the particle is in the ZKP level
IF (P(LK,6) .LT. 0.5) THEN
C calculate the lower two linear interp U's
bvp1(LK)=VT(IP+1,JP,ZKP+1)+(P(LK,4)-0.5)
&*(VT(IP+1,JP+1,ZKP+1)-VT(IP+1,JP,ZKP+1))
bvp2(LK)=VT(IP,JP,ZKP+1)+(P(LK,4)-0.5)
&*(VT(IP,JP+1,ZKP+1)-VT(IP,JP,ZKP+1))
C combine by linearly interp those two u's
bvp(LK)=bvp2(LK) + P(LK,5)
&*(bvp1(LK)-bvp2(LK))
C do the same for the level above
```

```
tvp1(LK)=VT(IP+1,JP,ZKP)+(P(LK,4)-0.5)
&*(VT(IP+1,JP+1,ZKP)-VT(IP+1,JP,ZKP))
tvp2(LK)=VT(IP,JP,ZKP)+(P(LK,4)-0.5)
&*(VT(IP,JP+1,ZKP)-VT(IP,JP,ZKP))
tvp(LK)=tvp2(LK) + P(LK,5)*(tvp1(LK)-tvp2(LK))
C combine the upper and lower by linear interp
vp(LK)=bvp(LK)+(2.0*P(LK,6)*DZ(ZKP)+DZ(ZKP+1))
&*(tvp(LK)-bvp(LK))/(DZ(ZKP)+DZ(ZKP+1))
C if dz is ge .5 the lower grid is ZKP and the upper grid is ZKP-1
C the particle is still in level ZKP
ELSE
C calculate the lower two linear interp U's
bvp1(LK)=VT(IP+1,JP,ZKP)+(P(LK,4)-0.5)
&*(VT(IP+1,JP+1,ZKP)-VT(IP+1,JP,ZKP))
bvp2(LK)=VT(IP,JP,ZKP)+(P(LK,4)-0.5)
&*(VT(IP,JP+1,ZKP)-VT(IP,JP,ZKP))
C combine by linearly interp those two u's
bvp(LK)=bvp2(LK) + P(LK,5)
&*(bvp1(LK)-bvp2(LK))
C do the same for the level above
tvp1(LK)=VT(IP+1,JP,ZKP-1)+(P(LK,4)-0.5)
```

```
&*(VT(IP+1,JP+1,ZKP-1)-VT(IP+1,JP,ZKP-1))
tvp2(LK)=VT(IP,JP,ZKP-1)+(P(LK,4)-0.5)
&*(VT(IP,JP+1,ZKP-1)-VT(IP,JP,ZKP-1))
tvp(LK)=tvp2(LK) + P(LK,5)*(tvp1(LK)-tvp2(LK))
C combine the upper and lower via linear interp.
vp(LK)=bvp(LK)+(2.0*(P(LK,6)-0.5)*DZ(ZKP)+DZ(ZKP-1))
&*(tvp(LK)-bvp(LK))/(DZ(ZKP)+DZ(ZKP-1))
ENDIF
C if dy is less than or equal to half the grid dist
C then the closet neighbors are IP,JP IP-1,JP
C IP,JP+1 IP-1,JP+1
ELSE
C if dz is lt .5 the lower grid is ZKP+1 and the upper is ZKP
C and the particle is in the ZKP level
IF (P(LK,6) .LT. 0.5) THEN
C calculate the lower two linear interp U's
bvp1(LK)=VT(IP,JP,ZKP+1)+(P(LK,5)+0.5)
&*(VT(IP-1,JP,ZKP+1)-VT(IP,JP,ZKP+1))
bvp2(LK)=VT(IP,JP+1,ZKP+1)+(P(LK,5)+0.5)
&*(VT(IP-1,JP+1,ZKP+1)-VT(IP,JP+1,ZKP+1))
```

```
C combine by linearly interp those two u's
bvp(LK)=bvp2(LK) + P(LK,4)
&*(bvp1(LK)-bvp2(LK))
C do the same for the level above
tvp1(LK)=VT(IP,JP,ZKP)+(P(LK,5)+0.5)
&*(VT(IP-1,JP,ZKP)-VT(IP,JP,ZKP))
tvp2(LK)=VT(IP,JP+1,ZKP)+(P(LK,5)+0.5)
&*(VT(IP-1,JP+1,ZKP)-VT(IP,JP+1,ZKP))
tvp(LK)=tvp2(LK) + P(LK,4)*(tvp1(LK)-tvp2(LK))
C combine the upper and lower by linear interp
vp(LK)=bvp(LK)+(2.0*P(LK,6)*DZ(ZKP)+DZ(ZKP+1))
&*(tvp(LK)-bvp(LK))/(DZ(ZKP)+DZ(ZKP+1))
C if dz is ge .5 the lower grid is ZKP and the upper grid is ZKP-1
C the particle is still in level ZKP
ELSE
C calculate the lower two linear interp U's
bvp1(LK)=VT(IP,JP,ZKP)+(P(LK,5)+0.5)
&*(VT(IP-1,JP,ZKP)-VT(IP,JP,ZKP))
bvp2(LK)=VT(IP,JP+1,ZKP)+(P(LK,5)+0.5)
&*(VT(IP-1,JP+1,ZKP)-VT(IP,JP+1,ZKP))
C combine by linearly interp those two u's
```

```
bvp(LK)=bvp2(LK) + P(LK,4)
&*(bvp1(LK)-bvp2(LK))
C do the same for the level above
tvp1(LK)=VT(IP,JP,ZKP-1)+(P(LK,5)+0.5)
&*(VT(IP-1,JP,ZKP-1)-VT(IP,JP,ZKP-1))
tvp2(LK)=VT(IP,JP+1,ZKP-1)+(P(LK,5)+0.5)
&*(VT(IP-1,JP+1,ZKP-1)-VT(IP,JP+1,ZKP-1))
tvp(LK)=tvp2(LK) + P(LK,4)*(tvp1(LK)-tvp2(LK))
C combine the upper and lower via linear interp.
vp(LK)=bvp(LK)+(2.0*(P(LK,6)-0.5)*DZ(ZKP)+DZ(ZKP-1))
&*(tvp(LK)-bvp(LK))/(DZ(ZKP)+DZ(ZKP-1))
ENDIF
ENDIF
C linerly interp W
IF (P(LK,4) .GE. 0.5) THEN
IF (P(LK,5) .GE. 0.5) THEN
bwp1(LK)=WT(IP+1,JP,ZKP)+(P(LK,5)-0.5)
&*(WT(IP+1,JP+1,ZKP)-WT(IP+1,JP,ZKP))
bwp2(LK)=WT(IP,JP,ZKP)+(P(LK,5)-0.5)
&*(WT(IP,JP+1,ZKP)-WT(IP,JP,ZKP))
```

```
bwp(LK)=bwp2(LK) + P(LK,4)
&*(bwp1(LK)-bwp2(LK))
twp1(LK)=WT(IP+1,JP,ZKP-1)+(P(LK,5)-0.5)
&*(WT(IP+1,JP+1,ZKP-1)-WT(IP+1,JP,ZKP-1))
twp2(LK)=WT(IP,JP,ZKP-1)+(P(LK,5)-0.5)
&*(WT(IP,JP+1,ZKP-1)-WT(IP,JP,ZKP-1))
twp(LK)=twp2(LK) + P(LK,4)*(twp1(LK)-twp2(LK))
wp(LK)=bwp(LK)+(P(LK,6))*(twp(LK)-bwp(LK))
ELSE
bwp1(LK)=WT(IP+1,JP-1,ZKP)+(P(LK,5))
&*(WT(IP+1,JP,ZKP)-WT(IP+1,JP-1,ZKP))
bwp2(LK)=WT(IP,JP-1,ZKP)+(P(LK,5))
&*(WT(IP,JP,ZKP)-WT(IP,JP-1,ZKP))
bwp(LK)=bwp2(LK) + P(LK,4)
&*(bwp1(LK)-bwp2(LK))
twp1(LK)=WT(IP+1,JP-1,ZKP-1)+(P(LK,5))
&*(WT(IP+1,JP,ZKP-1)-WT(IP+1,JP-1,ZKP-1))
twp2(LK)=WT(IP,JP-1,ZKP-1)+(P(LK,5))
&*(WT(IP,JP,ZKP-1)-WT(IP,JP-1,ZKP-1))
twp(LK)=twp2(LK) + P(LK,4)*(twp1(LK)-twp2(LK))
wp(LK)=bwp(LK)+(P(LK,6))*(twp(LK)-bwp(LK))
```

```
ENDIF
ELSE
IF (P(LK,5) .GE. 0.5) THEN
bwp1(LK)=WT(IP,JP,ZKP)+(P(LK,5)-0.5)
&*(WT(IP,JP+1,ZKP)-WT(IP,JP,ZKP))
bwp2(LK)=WT(IP-1,JP,ZKP)+(P(LK,5)-0.5)
&*(WT(IP-1,JP+1,ZKP)-WT(IP-1,JP,ZKP))
bwp(LK)=bwp2(LK) + P(LK,4)
&*(bwp1(LK)-bwp2(LK))
twp1(LK)=WT(IP,JP,ZKP-1)+(P(LK,5)-0.5)
&*(WT(IP,JP+1,ZKP-1)-WT(IP,JP,ZKP-1))
twp2(LK)=WT(IP-1,JP,ZKP-1)+(P(LK,5)-0.5)
&*(WT(IP-1,JP+1,ZKP-1)-WT(IP-1,JP,ZKP-1))
twp(LK)=twp2(LK) + P(LK,4)*(twp1(LK)-twp2(LK))
wp(LK)=bwp(LK)+(P(LK,6))*(twp(LK)-bwp(LK))
ELSE
bwp1(LK)=WT(IP,JP-1,ZKP)+(P(LK,5))
&*(WT(IP,JP,ZKP)-WT(IP,JP-1,ZKP))
bwp2(LK)=WT(IP-1,JP-1,ZKP)+(P(LK,5))
&*(WT(IP-1,JP,ZKP)-WT(IP-1,JP-1,ZKP))
```

```
bwp(LK)=bwp2(LK) + P(LK,4)
&*(bwp1(LK)-bwp2(LK))
twp1(LK)=WT(IP,JP-1,ZKP-1)+(P(LK,5))
&*(WT(IP,JP,ZKP-1)-WT(IP,JP-1,ZKP-1))
twp2(LK)=WT(IP-1,JP-1,ZKP-1)+(P(LK,5))
&*(WT(IP-1,JP,ZKP-1)-WT(IP-1,JP-1,ZKP-1))
twp(LK)=twp2(LK) + P(LK,4)*(twp1(LK)-twp2(LK))
wp(LK)=bwp(LK)+(P(LK,6))*(twp(LK)-bwp(LK))
ENDIF
ENDIF
C Integrate the vel. and add to old position
c print *,LK,UP(LK),ph1(lk),dti
MOVX=UP(LK)*p(lk,7)*p(lk,9)*DTI/ph1(lk)
c MOVX=MOVX+MOVX*dispx
MOVX=MOVX+(ran(seedx)-0.5)*dispx*p(lk,7)*p(lk,9)
MOVY=VP(LK)*p(lk,7)*p(lk,9)*DTI/ph2(lk)
c MOVY=MOVY+MOVY*dispy
MOVY=MOVY+(ran(seedy)-0.5)*dispy*p(lk,7)*p(lk,9)
MOVZ=WP(LK)*p(lk,7)*p(lk,9)*DTI/ph3(lk)*clev
c MOVZ=MOVZ+MOVZ*dispz
```

```
MOVZ=MOVZ+(ran(seedz)-0.5)*dispz*p(lk,7)*p(lk,9)
c print *,ran(seed),lk
c print *,UP(LK),'after movx'
C Update the I, DX position
P(LK,4)=P(LK,4)+MOVX
c print *,IFIX(p(lk,4)),p(lk,4),p(lk,1)
P(LK,1)=P(LK,1)+IFIX(P(LK,4))
P(LK,4)=P(LK,4)-IFIX(P(LK,4))
c print *,IFIX(p(lk,4)),p(lk,4),p(lk,1)
IF (P(LK,4) .LT. 0.0) THEN
P(LK,4)=1.0+P(LK,4)
P(LK,1)=P(LK,1)-1.0
ENDIF
C Update the J, DY position
P(LK,5)=P(LK,5)+MOVY
P(LK,2)=P(LK,2)+IFIX(P(LK,5))
P(LK,5)=P(LK,5)-IFIX(P(LK,5))
IF (P(LK,5) .LT. 0.0) THEN
P(LK,5)=1.0+P(LK,5)
P(LK,2)=P(LK,2)-1.0
ENDIF
```

```
C Update the K, DZ position
P(LK,6)=P(LK,6)+MOVZ
P(LK,3)=P(LK,3)-IFIX(P(LK,6))
P(LK,6)=P(LK,6)-IFIX(P(LK,6))
IF (P(LK,6) .LT. 0.0) THEN
P(LK,6)=1.0+P(LK,6)
P(LK,3)=P(LK,3)+1.0
ENDIF
IPF=IFIX(P(LK,1))
JPF=IFIX(P(LK,2))
ZKPF=IFIX(P(LK,3))
If (pong .eq. 1) then
C pong x
IF (FSM(IPF,JPF) .EQ. 0.0) THEN
IF (FSM(IPF,JP) .EQ. 0.0) THEN
IPF=IP
P(LK,4)=1.0-P(LK,4)
c print *,'pong x',IP,'ip'
ENDIF
```

```
C pong y
IF (FSM(IP,JPF) .EQ. 0.0) THEN
JPF=JP
P(LK,5)=1.0-P(LK,5)
c print *,'pong y',JP,'jp'
ENDIF
IF ((FSM(IP,JPF) .EQ. 1.0) .AND. (FSM(IPF,JP) .EQ. 1.0)) THEN
IF (MOVY .GE. MOVX) THEN
IPF=IP
P(LK,4)=1.0-P(LK,4)
ELSE
JPF=JP
P(LK,5)=1.0-P(LK,5)
ENDIF
ENDIF
ENDIF
C pong z
IF (ZKPF .GT. 10.0) THEN
ZKPF=ZKP
P(LK,6)=1.0-P(LK,6)
```

```
c print *,'pong z bottom',ZKP,'zkp'
ENDIF
IF (ZKPF .LT. 1.0) THEN
ZKPF=ZKP
P(LK,6)=1.0-P(LK,6)
c print *,'pong z top',ZKP,'zkp'
ENDIF
endif
If (stick .eq. 1) then
C stick x
IF (FSM(IPF,JP) .EQ. 0.0) THEN
IPF=IP
P(LK,4)=0.0
c print *,'stick x',IP,'ip'
ENDIF
C stick y
IF (FSM(IP,JPF) .EQ. 0.0) THEN
JPF=JP
P(LK,5)=0.0
c print *,'stick y',JP,'jp'
```

```
ENDIF
C stick z
IF (ZKPF .GT. 10.0) THEN
ZKPF=ZKP
P(LK,6)=0.0
c print *,'stick z bottom',ZKP,'zkp'
ENDIF
IF (ZKPF .LT. 1.0) THEN
ZKPF=ZKP
P(LK,6)=0.0
c print *,'stick z top',ZKP,'zkp'
ENDIF
endif
If (ob .eq. 1) then
C open boundary index
IF (JPF .EQ. 20) THEN
IF ((IPF .LE. 19).and.(IPF .GE. 17)) THEN
P(LK,7)=0.0
P(LK,4)=0.0
P(LK,5)=0.0
c print *,'ob ',IPF,JPF
```

Appendix 1 (Continued)

```
ENDIF
ENDIF
IF ((IPF .EQ. 23).and.(JPF .EQ. 19)) THEN
P(LK,7)=0.0
P(LK,4)=0.0
P(LK,5)=0.0
c print *,'ob ',IPF,JPF
ENDIF
IF (IPF .EQ. 29) THEN
IF ((JPF .LE. 18).and.(JPF .GE. 17)) THEN
P(LK,7)=0.0
P(LK,4)=0.0
P(LK,5)=0.0
c print *,'ob ',IPF,JPF
ENDIF
ENDIF
IF (JPF .EQ. 15) THEN
IF ((IPF .LE. 39).and.(IPF .GE. 33)) THEN
P(LK,7)=0.0
P(LK,4)=0.0
```

```
P(LK,5)=0.0
c print *,'ob ',IPF,JPF
ENDIF
ENDIF
endif
P(LK,1)=IPF
P(LK,2)=JPF
P(LK,3)=ZKPF
P(LK,8)=INT
if (pstart .eq. 1) then
res(p(lk,1),p(lk,2))=res(p(lk,1),p(lk,2))+1.0
C IF (MOD(SEQ,IPR).EQ.0) THEN
C WRITE(IUTR,33) P(LK,1),P(LK,2),P(LK,3),P(LK,4),
C &P(LK,5),P(LK,6),P(LK,7),P(LK,8)
c,P(LK,9),P(LK,10)
C 33 format(8E12.5)
c print *,seq,ipr
C ENDIF
ENDIF
10 CONTINUE
```

```
if (pstart .eq. 1) then
IF (MOD(SEQ,IPR).EQ.0) THEN
do 13 j=1,jm
do 13 i=1,im
WRITE(IURES,133) I,J,RES(I,J)
133 format(i3,i3,E15.8)
c print *,seq,ipr
RES(I,J)=0.0
13 continue
ENDIF
ENDIF
c write(*,*) p(22,1),p(20,1),p(22,4),p(20,4)
C write(*,*) ph1(50),ph2(50),ph3(50)
C close(IUTR)
close(IUTRS)
close(IURES)
RETURN
END
```