

```

real*8 xx
real*4 xn(0:500),u(0:500),b(0:500)
real*4 ran1(0:500),ran2(0:500),emid(0:500)
real*4 g(0:500),beta(0:500),amu(0:500)
real*4 sizeb(0:500),popb(0:500),un(0:500)
real*8 anmon(500),ansde(500),asmon(500),assde(500)
integer nmon(500),nsmon(500),nsde(500),nssde(500)
integer nd(500)
c This program solves a size-structured population problem.
c This program works best.
c First, a second Monte Carlo method is used.
c Second, an SPDE is solved.
c If iplot=1, a Matlab program is produced.
c Try time=.5, nt=200, nrunt=500, nnnn=400, npop=200, xx=971652.
  iplot=0
  xx=971652.
  time=.5
  xmax=1.0
  xmin=0.0
  pi=3.141592654
  nt=1000
  ant=nt
  dt=time/ant
  nrunt=500
  nnnn=250
  npop=200
  if(iplot.eq.1) goto 890
  write(6,100) nt,nnnn
  write(6,100) nrunt,nrunt,time,xmin,xmax
890 continue

c This part uses Monte Carlo.
  nm=0
  nv=0
  sm=0.0
  sv=0.0
  do 500 nrun=1,nrunt
    nn=npop
    ann=nn
c mm is the number of size classes
c un(j) is number between xn(j-1) and x(j)
    mm=100

```

```

    amm=mm
    hx=(xmax-xmin)/amm
    do 10 j=0,mm
    aj=j
    xn(j)= aj*hx+xmin
10  continue
    do 701 j=1,mm
    un(j)=calcu(xn(j-1),xn(j))
701 continue
    un(0)=0.0
    s=0.0
    do 109 i=1,mm
    s=s+un(i)/amm
109 continue
c   write(6,199) s
199 format(5x,e12.5)

    do 50 i=1,nt
    do 20 j=1,mm
    if(j.gt.1) goto 201
    s=0.0
    do 202 jj=1,mm
    beta(jj)=betacalc((xn(jj-1)+xn(jj))*0.5)
    s=s+beta(jj)*un(jj)*dt
202 continue
    call randun(xx,rand1,rand2)
    if(rand1.lt.s) un(1)=un(1)+1.0
201 continue
    call randun(xx,rand1,rand2)
    p=s
    death=amucalc(xn(j),p)
    hlp=death*dt*un(j)
    if(hlp.lt.0.0) hlp=0.0
    if(rand1.lt.hlp) un(j)=un(j)-1.0
    growth=gcalc(xn(j))
    hlp=growth*dt*un(j-1)/hx
    if(hlp.lt.0.0) hlp=0.0
    if(rand2.lt.hlp) un(j)=un(j)+1.0
    if(rand2.lt.hlp) un(j-1)=un(j-1)-1.0
20  continue
    k=0
50  continue

```

```

s1=0.0
s2=0.0
s3=0.0
s4=0.0
do 30 j=1,mm
s1=s1+un(j)
s2=s2+un(j)*un(j)
30 continue
do 301 j=1,mm
hlp=(xn(j-1)+xn(j))/2.0
s3=s3+un(j)*hlp
s4=s4+un(j)*un(j)*hlp*hlp
301 continue

s3=s3/s1
s4=s4/s1 -s3*s3
anmon(nrun)=s1
asmon(nrun)=s3
sm=sm+s3
sv=sv+s3*s3
nm=nm+s1
nv=nv+s1*s1
if(iplot.eq.1) goto 910
c write(6,100) nrunt,nt,time
c write(6,100) nrunt,nt,s1,s2,s3,s4
910 continue
100 format(2x,i5,2x,i5,4(2x,f12.5))
500 continue
anrun=nrunt
anm=nm
sm=sm/anrun
sv=sv/anrun-sm*sm
sv=sqrt(sv)
anm=anm/anrun
anv=nv
anv=sqrt(abs(anv/anrun-anm*anm))
if(iplot.eq.1) goto 909
write(6,100) nrunt,nrun,anm,anv,sm,sv
909 continue

nt=400

```

```

        ant=nt
        dt=time/ant
        nrunt=500
        nnnn=400
c This part solves an SPDE.
        nn=nnnn
        p=npop
        ann=nn
        hx=(xmax-xmin)/ann
        do 60 j=0,nn
        aj=j
        xn(j)=aj*hx+xmin
60      continue
        do 65 j=0,nn
        g(j)=gcalc(xn(j))
        beta(j)=betacalc(xn(j))
        amu(j)=amucalc(xn(j),p)
65      continue
        nrun=nrunt
        anrun=nrun
        st1=0.0
        st2=0.0
        tes=0.0
        tvs=0.0
        do 95 npaths=1,nrun
        do 70 j=1,nn
        u(j)=calcu(xn(j-1),xn(j))
70      continue
        do 75 i=1,nt
        do 74 k=1,nn
        call random(xx,rand1,rand2)
c          rand1=0.0
c          rand2=0.0
        ran1(k)=rand1
        ran2(k)=rand2
74      continue
        call random(xx,rand1,rand2)
c          rand1=0.0
        s=0.0
        do 77 k=1,nn
        s=s+(beta(k)*u(k))*dt
77      continue

```

```

u(0) = hx*(s+sqrt(abs(s))*rand1)/(dt*g(0))
do 78 k=1,nn
s=u(k)-dt*(g(k)*u(k)-g(k-1)*u(k-1))/hx
hlp=(amu(k)*u(k))*dt
s=s-hlp+sqrt(abs(hlp))*ran1(k)
hlp1=sqrt(abs(g(k)*u(k)*dt/hx))
hlp2=sqrt(abs(g(k-1)*u(k-1)*dt/hx))
s=s-hlp1*ran2(k)+hlp2*ran2(k-1)
b(k)=s
78 continue
do 79 k=1,nn
79 u(k)=b(k)
75 continue
s1=0.0
s2=0.0
s3=0.0
s4=0.0
do 91 j=0,nn
91 s1=s1+u(j)
do 92 j=0,nn
s2=s2+u(j)*xn(j)
s3=s3+u(j)*xn(j)*u(j)*xn(j)
92 continue
s2=s2/s1
s3=s3/s1-s2*s2
tes=tes+s1/anrun
tvs=tvs+s1*s1/anrun
st1=st1+s2/anrun
st2=st2+s2*s2/anrun
ansde(npaths)=s1
assde(npaths)=s2
asmon(nrun)=nn
if(iplot.eq.1) goto 901
c write(6,100) nn,nt,time,s1,s2,s3
c write(6,100) nn,nrun,time,st1,st2
901 continue
95 continue
tvs=sqrt(abs(tvs-tes*tes))
st2=sqrt(abs(st2-st1*st1))
if(iplot.eq.1) goto 667
write(6,100) nn,nrun,tes,tvs,st1,st2
667 continue

```

```

s1=0.0
s2=0.0
s3=0.0
np=10
ns=10
anp=np
ans=ns
pmid=330.
smid=.25
hp=100./anp
hs=.1/ans
pran=hp*anp
sran=hs*ans
do 911 j=0,np
aj=j
popb(j)=aj*hp+pmid-pran/2.0
911 continue
do 912 j=0,ns
aj=j
sizeb(j)= aj*hs+smid-sran/2.0
912 continue
do 914 i=1,nrunt
do 915 j=1,np
hlp=anmon(i)
if(popb(j-1).le.hlp.and.popb(j).gt.hlp) nmon(j)=nmon(j)+1
hlp=ansde(i)
if(popb(j-1).le.hlp.and.popb(j).gt.hlp) nsde(j)=nsde(j)+1
915 continue
914 continue
do 96 i=1,nrunt
do 97 j=1,ns
hlp=asmon(i)
if(sizeb(j-1).le.hlp.and.sizeb(j).gt.hlp) nsmon(j)=nsmon(j)+1
hlp=assde(i)
if(sizeb(j-1).le.hlp.and.sizeb(j).gt.hlp) nssde(j)=nssde(j)+1
97 continue
96 continue
if(iplot.ne.1) goto 908
write(6,170)
170 format(' pop=[')
do 171 j=1,np

```

```

171 write(6,179) (popb(j-1)+popb(j))/2.0
174 format(2x,i11)
    write(6,176)
176 format(' ];')
    write(6,172)
172 format(' nmon=[')
    do 173 i=1,np
173 write(6,174) nmon(i)
    write(6,176)
    write(6,182)
182 format(' nsde=[')
    do 183 i=1,np
183 write(6,174) nsde(i)
179 format(2x,f13.6)
    write(6,176)
    write(6,270)
270 format(' size=[')
    do 271 j=1,ns
271 write(6,179) (sizeb(j-1)+sizeb(j))/2.0
    write(6,176)
    write(6,272)
272 format(' nsmon=[')
    do 273 i=1,ns
273 write(6,174) nsmon(i)
    write(6,176)
    write(6,282)
282 format(' nssde=[')
    do 283 i=1,ns
283 write(6,174) nssde(i)
    write(6,176)
908 continue
    stop
    end

function gcalc(x)
c    gcalc=.5*(1.0-x)
    gcalc=1.0
    if(x.lt.0.2) gcalc=x/.2
    if(x.gt.0.8) gcalc=(1.0-x)/.2
    if(gcalc.lt.0.0001) gcalc=.0001
    return
end

```

```

function betacalc(x)
betacalc=1.5
betacalc=0.0
return
end

```

```

function amucalc(x,p)
amucalc=.5
amucalc=0.0
return
end

```

```

function calcboundx(ann,x0)
n=10
x1=x0+.5/ann
do 205 i=1,n
hlp=(fx(x1)+4.0*fx((x1+x0)/2.0)+fx(x0))*(x1-x0)/6.0
205 x1=x1-(hlp-1.0)/fx(x1)
calcboundx=x1
return
end

```

```

function calcu(x0,x1)
hlp=(fx(x1)+4.0*fx((x1+x0)/2.0)+fx(x0))*(x1-x0)/6.0
calcu=hlp
return
end

```

```

function fx(x)
anpop=200.0
fx=anpop*exp(-(x-.5)**2/.05)/.395712
c   fx=600.0*(1-x)**2
return
end

```

```

subroutine random(xx,rand1,rand2)
real*8 xx,a,b,d,rng(2)
a=16807.
ib=2147483647
b=ib

```

```

do 55 i=1,2
id=a*xx/b
d=id
xx=a*xx-d*b
55 rng(i)=xx/b
pi=3.141592654
u1=rng(1)
u2=rng(2)
hlp=sqrt(-2.0*log(u1))
rand1=hlp*cos(pi*2.0*u2)
rand2=hlp*sin(pi*2.0*u2)
return
end

```

```

subroutine randun(xx,rand1,rand2)
real*8 xx,a,b,d,rng(2)
a=16807.
ib=2147483647
b=ib
do 55 i=1,2
id=a*xx/b
d=id
xx=a*xx-d*b
55 rng(i)=xx/b
rand1=rng(1)
rand2=rng(2)
return
end

```

c-----

```

real*8 xx
real*4 xn(0:2000),u0(0:300,0:300),b(0:2000),yn(0:2000)
real*4 u(0:300),un(0:300,0:300)
real*4 ynm(0:2000),xnm(0:2000),bfrac(0:2000)
real*4 g0(0:300,0:300),amu0(0:300,0:300)
real*4 ran1(0:300,0:300),ran2(0:300,0:300),emid(0:2000)
real*4 g(0:2000),beta(0:2000),amu(0:2000),aasde(2000)
real*4 sizeb(0:300),popb(0:300),aamon(2000)
real*4 anmon(2000),ansde(2000),asmon(2000),assde(2000)
real*4 ageb(0:300)

```

```

integer nmon(2000),nsmon(2000),nsde(2000),nssde(2000)
integer nd(2000),namon(2000),nasde(2000)
c This program solves an age- and size-structured population problem.
c First, a Monte Carlo method is used.
c Second, an SPDE is solved.
c If iplot=1, a Matlab program is produced.
c Try time=.5, nt=4000, nrunt=500, nt=400, npop=400.

```

```

iplot=0
xx=23256.
time=1.0
xmax=1.0
xmin=0.0
ymin=0.0
ymax=1.0
nt=4000
ant=nt
dt=time/ant
nrunt=500
npop=400
anpop=npop
if(iplot.eq.1) goto 890
write(6,100) nt
write(6,100) nrunt,npop,time,xmin,xmax
890 continue

```

```

c This part uses Monte Carlo.

```

```

nm=0
nv=0
sm=0.0
sv=0.0
sms=0.0
svs=0.0
sma=0.0
sva=0.0
do 500 nrun=1,nrunt
nn=npop
ann=nn
c xn is size and yn is age
mms=10
amms=mms
mma=20

```

```

        amma=mma
c mms is number of size classes and mma is number of age classes
        hx=(xmax-xmin)/amms
        hy=(ymax-ymin)/amma
        do 10 j=0,mms
            aj=j
            xn(j)= aj*hx+xmin
            xnm(j+1)=xn(j)+hx/2.0
10      continue
            do 108 k=0,mma
                ak=k
                yn(k)=ak*hy+ymin
                ynm(k+1)=yn(k)+hy/2.0
108    continue
            do 11 j=1,mms
                do 11 k=1,mma
                    un(j,k)=calcu(xn(j-1),xn(j),yn(k-1),yn(k))
                    un(0,k)=0.0
                    un(j,0)=0.0
11      continue
            do 107 j=1,mms
                bfrac(j)=calcb(xn(j-1),xn(j))
107    continue

        do 50 i=1,nt
            s=0.0
            do 202 jj=1,mms
                do 202 kk=1,mma
                    hlp=betacalc(xnm(jj),ynm(kk))
                    s=s+un(jj,kk)*hlp*dt
202    continue
            call randun(xx,rand1,rand2)
            if(rand1.gt.s) goto 203
            do 204 jj=1,mms
204    un(jj,1)=un(jj,1)+bfrac(jj)
203    continue

        do 20 j=1,mms
            do 20 k=1,mma
                call randun(xx,rand1,rand2)
                death=amucalc(xnm(j),ynm(k))
                hlp=death*dt*un(j,k)

```

```

        if(rand1.lt.hlp) un(j,k)=un(j,k)-1.0
        call randun(xx,rand1,rand2)
        growth1=gcalc(xnm(j-1),ynm(k-1))
        hlp1=growth1*un(j-1,k)*dt/hx
        if(rand1.lt.hlp1) un(j-1,k)=un(j-1,k)-1.0
        if(rand1.lt.hlp1) un(j,k)=un(j,k)+1.0

c Here, age increases deterministically by dt.
        un(j,k)=un(j,k)*(1.0-dt/hy)+dt*un(j,k-1)/hy
        if(un(j,k).lt.0.0) un(j,k)=0.0
20    continue
50    continue
        s1=0.0
        s2=0.0
        s3=0.0
        s4=0.0
        do 30 j=1,mms
        do 30 k=1,mma
        s1=s1+un(j,k)
        s2=s2+un(j,k)*xnm(j)
        s4=s4+un(j,k)*ynm(k)
30    continue
        s2=s2/s1
        s4=s4/s1
        anmon(nrun)=s1
        asmon(nrun)=s2
        aamon(nrun)=s4
        sm=sm+s1
        sv=sv+s1*s1
        sms=sms+s2
        svs=svs+s2*s2
        sma=sma+s4
        sva=sva+s4*s4
        nm=nm+nn
        nv=nv+nn*nn
        if(iplot.eq.1) goto 910
c        write(6,100) nrun,nt,time
c        write(6,100) nrun,nn,s1,s2
c        write(6,100) nrun,nn,s4
910   continue
100   format(2x,i5,2x,i5,4(2x,f12.5))

```

```

500 continue
    anrun=nrunt
    anm=nm
    sm=sm/anrun
    sv=sqrt(sv/anrun-sm*sm)
    sms=sms/anrun
    svs=sqrt(svs/anrun-sms*sms)
    sma=sma/anrun
    sva=sqrt(sva/anrun-sma*sma)
    anm=anm/anrun
    anv=nv
    anv=sqrt(anv/anrun-anm*anm)
    if(iplot.eq.1) goto 909
    write(6,100) nt,nrunt,anm,anv,sm,sv
    write(6,100) mms,mma,sms,svs
    write(6,100) nrunt,npop,sma,sva
909 continue

```

c This part solves an age- and size-structured population problem.

c If idet=1, a deterministic method is used.

```

xx=22656.
idet=0
nrun=nrunt
if(idet.eq.1) nrun=1
anrun=nrun

nt=400
kk=40
jj=20

p=anpop
xmax=1.0
xmin=0.0
ymax=1.0
ymin=0.0
ant=nt
dt=time/ant
ajj=jj
akk=kk
hx=(xmax-xmin)/ajj
hy=(ymax-ymin)/akk

```

```

do 601 j=0,jj
aj=j
xn(j)=aj*hx+xmin
xnm(j)=xn(j)+hx/2.0
601 continue
do 602 k=0,kk
ak=k
yn(k)=ak*hy+ymin
ynm(k)=yn(k)+hy/2.0
602 continue
do 651 j=0,jj
do 651 k=0,kk
g0(j,k)=gcalc(xn(j),yn(k))
beta(j)=3.0*fb(xn(j))
amu0(j,k)=amucalc(xn(j),yn(k))
651 continue
st1=0.0
st2=0.0
tes=0.0
tvs=0.0
st3=0.0
st4=0.0
tea=0.0
tva=0.0
do 95 npaths=1,nrun
do 701 j=0,jj
do 701 k=0,kk
u0(j,k)=f(xn(j),yn(k))*hx*hy
701 continue
do 751 i=1,nt
do 74 j=0,jj
do 74 k=0,kk
call random(xx,rand1,rand2)
ran1(j,k)=rand1
ran2(j,k)=rand2
74 continue
s=0.0
do 771 j=0,jj
do 771 k=0,kk
s=s+u0(j,k)
771 continue
hlp7=1.0

```

```

    if(idet.eq.1) hlp7=0.0
    do 772 j=0,jj
    u0(j,kk)=0.0
    call random(xx,rand1,rand2)
    hlp=s*beta(j)*dt*hx
772  u0(j,0)=hlp+hlp7*sqrt(abs(hlp))*rand1
    do 781 j=1,jj
    do 781 k=1,kk
    hlp1=(u0(j,k)-u0(j,k-1))*dt/hy

    if(k.eq.1) hlp1=u0(j,1)*dt/hy-u0(j,0)

    hlp2=(g0(j,k)*u0(j,k)-g0(j-1,k)*u0(j-1,k))*dt/hx
    hlp3=(amu0(j,k)*u0(j,k))*dt
    hlp33=hlp3
    if(hlp33.lt.0.0) hlp33=0.0
    hlp21=(g0(j,k)*u0(j,k))*dt/hx
    hlp22=(g0(j-1,k)*u0(j-1,k))*dt/hx
    if(hlp21.lt.0.0) hlp21=0.0
    if(hlp22.lt.0.0) hlp22=0.0
    hlp4=sqrt(abs(hlp33))*ran1(j,k)
    hlp5=sqrt(abs(hlp21))*ran2(j,k)
    hlp6=-sqrt(abs(hlp22))*ran2(j-1,k)
    hlp8=hlp7*(hlp4+hlp5+hlp6)
    un(j,k)=u0(j,k)-(hlp1+hlp2+hlp3)-hlp8
781  continue

    do 791 j=1,jj
    do 791 k=0,kk
    if(un(j,k).lt.0.0) un(j,k)=0.0
791  u0(j,k)=un(j,k)
751  continue
    s1=0.0
    s2=0.0
    s3=0.0
    s4=0.0
    s5=0.0
    do 831 j=0,jj
    do 831 k=0,kk
    s1=s1+u0(j,k)
    s2=s2+u0(j,k)*xn(j)
    s3=s3+u0(j,k)*yn(k)

```

```

s4=s4+u0(j,k)*xn(j)*xn(j)
s5=s5+u0(j,k)*yn(k)*yn(k)
831 continue
s2=s2/s1
s3=s3/s1
s4=s4/s1-s2*s2
s5=s5/s1-s3*s3
c write(6,101) jj,kk,nt,time,s1,s2,s3
c write(6,101) jj,kk,nt,time,s4,s5
101 format(2x,i4,2x,i4,2x,i4,5(2x,f12.7))
tes=tes+s1/anrun
tvs=tvs+s1*s1/anrun
st1=st1+s2/anrun
st2=st2+s2*s2/anrun
st3=st3+s3/anrun
st4=st4+s3*s3/anrun
ansde(npaths)=s1
assde(npaths)=s2
aasde(npaths)=s3
95 continue
tvs=sqrt(tvs-tes*tes)
st2=sqrt(st2-st1*st1)
st4=sqrt(st4-st3*st3)
if(iplot.eq.1) goto 901
write(6,188) nt,kk,jj,nrun
write(6,109) tes,tvs,st1,st2,st3,st4
188 format(2x,4(i5,2x))
109 format(2x,f10.4,2x,f12.5,2x,4(f9.5,1x))

901 continue
s1=0.0
s2=0.0
s3=0.0
np=10
ns=10
na=10
anp=np
ans=ns
ana=na
pmid=1050.
smid=.50
amid=.265

```

```

hp=600./anp
hs=.04/ans
ha=.04/ana
pran=hp*anp
sran=hs*ans
aran=ha*ana
do 921 j=0,np
aj=j
popb(j)=aj*hp+pmid-pran/2.0
921 continue
do 912 j=0,ns
aj=j
sizeb(j)=aj*hs+smid-sran/2.0
912 continue
do 981 j=0,na
aj=j
ageb(j)=aj*ha+amid-aran/2.0
981 continue
do 914 i=1,nrunt
do 915 j=1,np
hlp=anmon(i)
if(popb(j-1).le.hlp.and.popb(j).gt.hlp) nmon(j)=nmon(j)+1
hlp=ansde(i)
if(popb(j-1).le.hlp.and.popb(j).gt.hlp) nsde(j)=nsde(j)+1
915 continue
914 continue
do 96 i=1,nrunt
do 97 j=1,ns
hlp=asmon(i)
if(sizeb(j-1).le.hlp.and.sizeb(j).gt.hlp) nsmon(j)=nsmon(j)+1
hlp=assde(i)
if(sizeb(j-1).le.hlp.and.sizeb(j).gt.hlp) nssde(j)=nssde(j)+1
97 continue
96 continue
do 961 i=1,nrunt
do 971 j=1,na
hlp=aamon(i)
if(ageb(j-1).le.hlp.and.ageb(j).gt.hlp) namon(j)=namon(j)+1
hlp=aasde(i)
if(ageb(j-1).le.hlp.and.ageb(j).gt.hlp) nasde(j)=nasde(j)+1
971 continue
961 continue

```

```

        if(iplot.ne.1) goto 908
        write(6,170)
170    format(' pop=[')
        do 171 j=1,np
171    write(6,179) (popb(j-1)+popb(j))/2.0
174    format(2x,i11)
        write(6,176)
176    format(' ]:')
        write(6,172)
172    format(' nmon=[')
        do 173 i=1,np
173    write(6,174) nmon(i)
        write(6,176)
        write(6,182)
182    format(' nsde=[')
        do 183 i=1,np
183    write(6,174) nsde(i)
179    format(2x,f13.6)
        write(6,176)
        write(6,270)
270    format(' size=[')
        do 271 j=1,ns
271    write(6,179) (sizeb(j-1)+sizeb(j))/2.0
        write(6,176)
        write(6,272)
272    format(' nsmon=[')
        do 273 i=1,ns
273    write(6,174) nsmon(i)
        write(6,176)
        write(6,282)
282    format(' nssde=[')
        do 283 i=1,ns
283    write(6,174) nssde(i)
        write(6,176)
        write(6,370)
370    format(' age=[')
        do 371 j=1,ns
371    write(6,179) (ageb(j-1)+ageb(j))/2.0
        write(6,176)
        write(6,372)
372    format(' namon=[')

```

```

do 373 i=1,ns
373 write(6,174) namon(i)
write(6,176)
write(6,382)
382 format(' nasde=[')
do 383 i=1,ns
383 write(6,174) nasde(i)
write(6,176)
908 continue

stop
end

function gcalc(x,y)
real*4 x,y
gcalc=.5*(1.0-x)*(1.0-y)*x
return
end

function betacalc(x,y)
real*4 x,y
betacalc=3.0
return
end

function amucalc(x,y)
real*4 x,y
if(y.gt.0.999) y=.999
amucalc=-1.0+2.0/(1.0-y)-(1.0-2.0*x)*(1.0-y)
return
end

function calcul(x1,x2,y1,y2)
xm=(x1+x2)/2.0
ym=(y1+y2)/2.0
hlp1=f(x1,y1)+4.0*f(xm,y1)+f(x2,y1)
hlp2=f(x1,ym)+4.0*f(xm,ym)+f(x2,ym)
hlp3=f(x1,y2)+4.0*f(xm,y2)+f(x2,y2)
calcul=(hlp1+4.0*hlp2+hlp3)*(x2-x1)*(y2-y1)/36.0
return
end

```

```

function calcb(x1,x2)
hlp=fb(x1)+4.0*fb((x1+x2)*.5)+fb(x2)
calcb=hlp*(x2-x1)/6.0
return
end

```

```

function fb(x)
fb=6.0*(1.0-x)*x
return
end

```

```

function f(x,y)
f=400.0*18.0*x*(1.0-x)*(1.0-y)**2
return
end

```

```

subroutine random(xx,rand1,rand2)
real*8 xx,a,b,d,rng(2)
a=16807.
ib=2147483647
b=ib
do 55 i=1,2
id=a*xx/b
d=id
xx=a*xx-d*b
55 rng(i)=xx/b
pi=3.141592654
u1=rng(1)
u2=rng(2)
hlp=sqrt(-2.0*log(u1))
rand1=hlp*cos(pi*2.0*u2)
rand2=hlp*sin(pi*2.0*u2)
return
end

```

```

subroutine randun(xx,rand1,rand2)
real*8 xx,a,b,d,rng(2)
a=16807.
ib=2147483647
b=ib
do 55 i=1,2
id=a*xx/b

```

```
d=id
xx=a*xx-d*b
55  rng(i)=xx/b
    rand1=rng(1)
    rand2=rng(2)
    return
end
```