The Abdus Salam
International Centre
for Theoretical Physics

United Nations
Educational, Scientific and
Cultural Organization

IAEA
International Atomic Energy Agency

2549-5

ADGLASS Winter School on Advanced Molecular Dynamics Simulations

18 - 21 February 2013

langevin and langevin_meta

Alessandro Laio
International School for Advanced Studies (SISSA)
Italy

```
program langevin
implicit none
real*8 D,dt,s,dW,fs,kT,eps,fb
integer it,NT,idum
integer ig,NG,iu
real*8, allocatable ::  VG(:)
real*8 smin,ds,smax,Pbias_of_s,err_F,dVg_ds,sg,ev
integer :: tau_G=500 !frequency of Gaussian deposition
real*8 :: w=0.04 !height of the Gaussians
real*8 :: delta_s=0.06 !width of the Gaussians


  D=1.d-4    ! diffusion coefficient
  dt=1.d0    !time step
  idum=-345595 !seed for the random number
  s=-1.d0     !initial condition
  eps=1.d-3   !eps for computing the numerical derivative
  kT=1.d0   !temperature
  NT=5000000
  NG=100
  smin=-2.5
  smax=2.5
  ds=(smax-smin)/dble(NG+1)
  allocate(VG(NG))

  iu=1000
  call system("rm -f fort.*")
  do it=1,NT
    fs=-(V(s+eps)-V(s-eps))/2/eps      !force from the true potential
    ig=floor((s-smin)/ds)+1
    if(ig<1 .or. ig>NG)then
      write(6,*)s
      stop 's out of range'
    endif
    dVg_ds=(VG(ig+1)-VG(ig))/ds

    !integration of the Langevin equation with the ito role.
    dW=sqrt(dt)*gasdev(idum)        !this is a Wiener process
    s=s+D*dt*(fs-dVg_ds)/kT+sqrt(2.*D)*dW
    if(mod(it,tau_G)==0)then
      ds=(smax-smin)/dble(NG+1)
      do ig=1,NG
        sg=smin+dble(ig-1)*ds
        ev=(s-sg)**2/2/delta_s**2
        if(ev>6.)cycle
        VG(ig)=VG(ig)+w*exp(-ev)
      enddo
      write(6,*)it,s
    endif
    if(mod(it,NT/30)==0  .or. it==1)then
      iu=iu+1
      do ig=1,NG
        sg=smin+dble(ig-1)*ds
        write(iu,*)sg,VG(ig),V(sg)
      enddo
      call flush(iu)
    endif

  enddo
contains
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! this is the "true" potential
    real*8 function V(s)
    implicit none
```

```
    real*8 s
    V=8.d0*(s**4-2.d0*s**2+s/4.d0)
    if(s<-2.5)V=V+100.*(s+2.5)**2
    if(s>2.5)V=V+100.*(s-2.5)**2
    end function V
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      real*8 FUNCTION ran1(idum)
      implicit none
      INTEGER idum,IA,IM,IQ,IR,NTAB,NDIV
      real*8 AM,EPS,RNMX
      PARAMETER (IA=16807,IM=2147483647,AM=1./IM,IQ=127773,IR=2836, &
     NTAB=32,NDIV=1+(IM-1)/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
      INTEGER j,k,iv(NTAB),iy
      SAVE iv,iy
      DATA iv /NTAB*0/, iy /0/
      if (idum.le.0.or.iy.eq.0) then
        idum=max(-idum,1)
        do 11 j=NTAB+8,1,-1
          k=idum/IQ
          idum=IA*(idum-k*IQ)-IR*k
          if (idum.lt.0) idum=idum+IM
          if (j.le.NTAB) iv(j)=idum
11      continue
        iy=iv(1)
      endif
      k=idum/IQ
      idum=IA*(idum-k*IQ)-IR*k
      if (idum.lt.0) idum=idum+IM
      j=1+iy/NDIV
      iy=iv(j)
      iv(j)=idum
      ran1=min(AM*iy,RNMX)
      return
      END function ran1
      REAL*8 FUNCTION gasdev(idum)

      implicit none
      INTEGER idum
      INTEGER iset
      real*8 fac,gset,rsq,v1,v2
      SAVE iset,gset
      DATA iset/0/
      if (iset.eq.0) then
1       v1=2.*ran1(idum)-1.
        v2=2.*ran1(idum)-1.
        rsq=v1**2+v2**2
        if(rsq.ge.1..or.rsq.eq.0.)goto 1
        fac=sqrt(-2.*log(rsq)/rsq)
        gset=v1*fac
        gasdev=v2*fac
        iset=1
      else
        gasdev=gset
        iset=0
      endif
      return
      END FUNCTION gasdev
end program langevin
```

```
program langevin
implicit none
real*8 D,dt,s,dW,fs,kT,eps,fb
integer it,NT,idum
integer ig,NG
integer, allocatable ::  n(:)
real*8, allocatable ::  sav(:)
real*8 smin,ds,smax,Pbias_of_s,err_F

  D=1.d-4    ! diffusion coefficient
  dt=1.d0    !time step
  idum=-345595 !seed for the random number
  s=-1.d0     !initial condition
  eps=1.d-3   !eps for computing the numerical derivative
  kT=1.d0   !temperature
  NT=700000
  !for the histogram:
  open(10,file="HIST",status='unknown')
  NG=100
  smin=-2.5
  smax=2.5
  ds=(smax-smin)/dble(NG+1)
  allocate(n(NG),sav(NG))
  n(:)=0
  sav(:)=0.d0

  do it=1,NT
    fs=-(V(s+eps)-V(s-eps))/2/eps      !force from the true potential
    fb=-(Vb(s+eps)-Vb(s-eps))/2/eps    !force from the bias potential
    !integration of the Langevin equation with the ito role.
    dW=sqrt(dt)*gasdev(idum)        !this is a Wiener process
    s=s+D*dt*(fs+fb)/kT+sqrt(2.*D)*dW
    if(mod(it,100)==0)then
      ig=int((s-smin)/ds)
      if(ig<1)ig=1
      if(ig>NG)ig=NG
      n(ig)=n(ig)+1
      sav(ig)=sav(ig)+s
      write(6,*)s
    endif
  enddo
  do ig=1,NG
    if(n(ig)<1)cycle
    sav(ig)=sav(ig)/(dble(n(ig)))
    Pbias_of_s=dble(n(ig))/dble(NT)
    err_F=sqrt(kT**2/dble(NT)/ds/Pbias_of_s)
    write(10,'(10f20.12)')sav(ig),Pbias_of_s,err_F,Vb(sav(ig))
  enddo
contains
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! this is the "true" potential
    real*8 function V(s)
    implicit none
    real*8 s
    V=8.d0*(s**4-2.d0*s**2+s/4.d0)
    if(s<-2.5)V=V+100.*(s+2.5)**2
    if(s>2.5)V=V+100.*(s-2.5)**2
    end function V
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! the bias potential is defined here
    real*8 function Vb(s)
    implicit none
    real*8 s
    Vb=9*exp(-(s+1.)**2/0.5**2)+5*exp(-(s-1.)**2/0.5**2)
```

```
    end function Vb
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      real*8 FUNCTION ran1(idum)
      implicit none
      INTEGER idum,IA,IM,IQ,IR,NTAB,NDIV
      real*8 AM,EPS,RNMX
      PARAMETER (IA=16807,IM=2147483647,AM=1./IM,IQ=127773,IR=2836, &
     NTAB=32,NDIV=1+(IM-1)/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
      INTEGER j,k,iv(NTAB),iy
      SAVE iv,iy
      DATA iv /NTAB*0/, iy /0/
      if (idum.le.0.or.iy.eq.0) then
        idum=max(-idum,1)
        do 11 j=NTAB+8,1,-1
          k=idum/IQ
          idum=IA*(idum-k*IQ)-IR*k
          if (idum.lt.0) idum=idum+IM
          if (j.le.NTAB) iv(j)=idum
11      continue
        iy=iv(1)
      endif
      k=idum/IQ
      idum=IA*(idum-k*IQ)-IR*k
      if (idum.lt.0) idum=idum+IM
      j=1+iy/NDIV
      iy=iv(j)
      iv(j)=idum
      ran1=min(AM*iy,RNMX)
      return
      END function ran1
      REAL*8 FUNCTION gasdev(idum)

      implicit none
      INTEGER idum
      INTEGER iset
      real*8 fac,gset,rsq,v1,v2
      SAVE iset,gset
      DATA iset/0/
      if (iset.eq.0) then
1       v1=2.*ran1(idum)-1.
        v2=2.*ran1(idum)-1.
        rsq=v1**2+v2**2
        if(rsq.ge.1..or.rsq.eq.0.)goto 1
        fac=sqrt(-2.*log(rsq)/rsq)
        gset=v1*fac
        gasdev=v2*fac
        iset=1
      else
        gasdev=gset
        iset=0
      endif
      return
      END FUNCTION gasdev
end program langevin
```