c \*\*\*\* Monte Carlo calc to make mi=8 images of 2VB1 that reproduce site occupancies \*\*\*\*

implicit real\*8 (a-h,o-z)

include 'readhin.cmn'

include 'config.cmn'

real\*8 rcut

real\*8 k1,k2,k3,k4,k5

real\*8 eps(mtype),sig(mtype)

real\*8 frtype(2,mgroup)

logical accept,again,im8incl(0:26)

integer natom1,nmol1

integer i,k,j,ktot

integer ntype(mpos,mgroup)

integer iingroup

character\*2 datatype(mtype)

character\*80 row

character\*1 ihinA

character\*12 statname

real\*8 statacc, staten, stattot

real\*8 statgroup(mpos,mgroup),statwater(mgroup)

real\*8 statgracc(4,mgroup),stat1,stat2,stat3

real\*8 histo(2200,4),nhisto,hsave(2200)

integer ihistres(1000),histres(2,1000)

integer iedor1,iedor2,igroupres1,igroupres2

integer igroupedo1,igroupedo2,igroupno31,igroupno32

integer igrouphoh1,igrouphoh2,kn,ihin

integer ic,nimage,imol,isave

integer initr1,initr2,iwatr1,iwatr2

integer iedo1,iedo2,init1,init2,iwat1,iwat2,ipos

integer im,ia,ib,i0,imol0

real\*8 temp,mi,qr,dx,dy,dz

character\*9 watertest

character\*11 hinname

character\*12 fname(3)

data fname /'2VB1AAAc.hin','2VB1BBBc.hin','2VB1AACc.hin'/

resgroup= -1

if (nargs().lt.2) then

write (6,\*) 'ENTER A LETTER FROM A - Z'

read (5,'(a)') ihinA

else

call getarg (1,ihinA)

end if

im8incl= .false.

im8incl(13)= .true.

if (ihinA.eq.'A') qr= rand(0)

if (ihinA.eq.'B') then

do i=1,2

qr=rand(0)

end do

end if

if (ihinA.eq.'C') then

do i=1,20

qr=rand(0)

end do

end if

if (ihinA.eq.'D') then

do i=1,25

qr=rand(0)

end do

end if

if (ihinA.eq.'E') then

do i=1,5

qr=rand(0)

end do

end if

if (ihinA.eq.'F') then

do i=1,8

qr=rand(0)

end do

end if

if (ihinA.eq.'G') then

do i=1,50

qr=rand(0)

end do

end if

if (ihinA.eq.'H') then

do i=1,39

qr=rand(0)

end do

end if

if (ihinA.eq.'G') then

do i=1,61

qr=rand(0)

end do

end if

if (ihinA.eq.'I') then

do i=1,71

qr= rand(0)

end do

end if

c \*\*\*\*\*\*\*\*\*\*\*\*\* restart from dump file \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

open (1,file='config.dump',status='old',form='unformatted',err=30)

read (1) xg,yg,zg,xoff,yoff,zoff,eps2,sig2,frtype,e,enew

read (1) nberint,intlist,ntype,npos,group,config,iswater

read (1) rcut,natom1,nmol1,initr1,initr2,iwatr1,iwatr2

read (1) iedor1,iedor2,iattype,resgroup,grouptype

read (1) ngroup,ningroup,resgroup,igroupres1,igroupres2

read (1) igroupedo1,igroupedo2,igroupno31,igroupno32

read (1) igrouphoh1,igrouphoh2,configres

read (1) temp,kn,ihin,nrest,nrest1,ngroup1,resnamet

read (1) iarest,jarest,molind,molend

close (1)

if (.not. readhin ('config.hin',0) ) stop 'restart hin file not found'

goto 31

30 continue

c \*\*\*\*\*\*\*\*\*\*\*\* else start from the beginning \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

c \*\*\*\* read and process the coords for configs A B and C \*\*\*\*

do ic= 1,3

nmol= 0

natom= 0

if (.not. readhin (fname(ic),0)) stop 'file not found'

write (6,\*) ic,' axis lens:',alen,blen,clen

nimage= 1

if (ic.eq.2) then

c \*\*\*\* add space for 9 missing atoms in TYR 20 - A of config B and delete two HXT \*\*\*\*

c \*\*\*\* missing atoms are CD1 CD2 CE1 CE2 CZ OH and hydrogens HD1 HD2 HE1 HE2 and HH \*\*\*\*

resname(20,imol)(1:3)= 'TYR'

i= iares(20,1) - 1

call hdel (i+11,2,1)

call hadd (i+6,6,1)

call hadd (i+16,5,1)

c \*\*\*\* copy over coords of str B \*\*\*\*

do i= iares(20,1),jares(20,1)

xg(i,2)= x(i)

yg(i,2)= y(i)

zg(i,2)= z(i)

end do

c \*\*\*\* copy back A str \*\*\*\*

call hcopy (isave,iares(20,1),nares(20,1))

c \*\*\*\* copy back new coords, use CG locn as shit for missing atoms \*\*\*\*

i= iares(20,1) + 5

dx= xg(i,2) - xg(i,1)

dy= yg(i,2) - yg(i,1)

dz= zg(i,2) - zg(i,1)

do i= iares(20,1),jares(20,1)

if (xg(i,2).eq.-1000.) then

x(i)= xg(i,1) + dx

y(i)= yg(i,1) + dy

z(i)= zg(i,1) + dz

else

x(i)= xg(i,2)

y(i)= yg(i,2)

z(i)= zg(i,2)

end if

end do

end if

c \*\*\*\*\*\*\*\*\*\*\*\*\* find EDOs \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

iedo1= 0

do imol= 1,nmol

if (resname(1,imol)(1:3).eq.'EDO') then

c \*\*\*\* mol numbers of first and last 1,2-ethanediol \*\*\*\*\*

if (iedo1.eq.0) iedo1= imol

iedo2= imol

end if

end do

write (6,\*) '1,2-ethanediols =', iedo2-iedo1+1,iedo1,iedo2

c \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* find NO3s \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

init1= 0

do imol= 1,nmol

if (resname(1,imol)(1:3).eq.'NO3') then

c \*\*\*\* mol numbers of first and last nitrates \*\*\*\*

if (init1.eq.0) init1= imol

init2= imol

end if

end do

write (6,\*) 'nitrates =',init2-init1+1,init1,init2

c \*\*\*\* find waters, add space for hydrogens \*\*\*\*

iwat1= 0

do imol= 1,nmol

if (resname(1,imol)(1:3).eq.'Hoh') then

resname(1,imol)(1:3)= 'HOH'

c \*\*\*\* mol numbers of first and last waters \*\*\*\*

if (iwat1.eq.0) iwat1= imol

iwat2= imol

c \*\*\*\* add space for 2 atoms at end of res \*\*\*\*

j= jares(1,imol)

call hadd (j,2,1)

nat(j)= 8

atname(j)= ' O'

ats(j)= 'O '

attype(j)= 'OW'

q(j)= -0.834

flag(j)= ' h'

ncon(j)= 2

icon(1,j)= 2

icon(2,j)= 3

scon(1,j)= 's'

scon(2,j)= 's'

nat(j+1)= 1

atname(j+1)= ' 1H'

ats(j+1)= 'H '

attype(j+1)= 'HW'

q(j+1)= 0.417

flag(j+1)= ' -'

ncon(j+1)= 1

icon(1,j+1)= 1

scon(1,j+1)= 's'

nat(j+2)= 1

atname(j+2)= ' 2H'

ats(j+2)= 'H '

attype(j+2)= 'HW'

q(j+2)= 0.417

flag(j+2)= ' -'

ncon(j+2)= 1

icon(1,j+2)= 1

scon(1,j+2)= 's'

end if

end do

write (6,\*) 'waters =',iwat2-iwat1+1,iwat1,iwat2

write (6,\*) 'Finished mol ',fname(ic),' natom=',natom

c \*\*\*\* copy into main coord storage for each posibility \*\*\*\*

do i= 1,natom

xg(i,ic)= x(i)

yg(i,ic)= y(i)

zg(i,ic)= z(i)

do j= 1,6

anisog(j,i,ic)= aniso(j,i)

end do

end do

if (ic.eq.1) then

c \*\*\*\* for str A, copy TYR 20 - A so as to overwrite errors for str B \*\*\*\*

isave= natom+1

call hcopy (iares(20,1),isave,nares(20,1))

end if

end do

c \*\*\*\*\*\*\*\*\*\*\*\* create 72 alternative posibilities for each water \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

c \*\*\*\*\*\*\*\* read water orientations \*\*\*\*\*\*\*

open (1,file='makeh2o.out',status='old')

read (1,\*) ((water(i,j),i=1,6),j=1,mpos)

close (1)

write (6,\*) '72 reference water orientatioons read from makeh2o.out'

do imol= iwat1,iwat2

c \*\*\*\* set std water config for normal waters \*\*\*\*

i= iares(1,imol)

write (6,\*) 'setting water ',imol,i

do ipos= 1,72

xg(i,ipos)= xg(i,1)

yg(i,ipos)= yg(i,1)

zg(i,ipos)= zg(i,1)

xg(i+1,ipos)= water(1,ipos) + xg(i,ipos)

yg(i+1,ipos)= water(2,ipos) + yg(i,ipos)

zg(i+1,ipos)= water(3,ipos) + zg(i,ipos)

xg(i+2,ipos)= water(4,ipos) + xg(i,ipos)

yg(i+2,ipos)= water(5,ipos) + yg(i,ipos)

zg(i+2,ipos)= water(6,ipos) + zg(i,ipos)

end do

x(i+1)= xg(i+1,1)

y(i+1)= yg(i+1,1)

z(i+1)= zg(i+1,1)

x(i+2)= xg(i+2,1)

y(i+2)= yg(i+2,1)

z(i+2)= zg(i+2,1)

end do

call writehin ('jnk.hin',0)

c \*\*\*\*\*\*\*\*\*\*\* create 7 copies \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

write (6,\*) 'axis lens:',alen,blen,clen

write (6,\*) boxvecs

open (24,file='imagespecs-2VB1.dat')

write (24,'(a)') header(6)(1:len\_trim(header(6)))

write (24,'(a,3f9.3)') '; image 1 0 0 0 1 0 0 0 1', 0.,0.,0.

write (header(6),'(a,6f9.3)') '; box', alen\*2,blen\*2,clen\*2,

$ alpha,beta,gamma

natom1= natom

nmol1= nmol

mi= 8

if (natom1\*mi.gt.m) stop 'dimension m too small'

if (nmol1\*mi.gt.mmol) stop 'dimension mmol too small'

im= 1

do ia= 0,1

do ib= 0,1

do ic= 0,1

if (ia.ne.0 .or. ib.ne.0 .or. ic.ne.0) then

im= im + 1

i0= (im-1)\*natom1

imol0= (im-1)\*nmol1

do imol= 1,nmol1

call hcopymol (imol,imol0+imol)

molind(imol+imol0)= molind(imol)+i0

molend(imol+imol0)= molend(imol)+i0

do ires= 1,nres(imol + imol0)

kk= len\_trim (resname(ires,imol))

resname(ires,imol+imol0)(kk:kk)=

$ char (ichar ('A') + im - 1)

end do

end do

dx= boxvecs(1)\*ia + boxvecs(4)\*ib + boxvecs(7)\*ic

dy= boxvecs(5)\*ib + boxvecs(8)\*ic

dz= boxvecs(9)\*ic

write (6,'(a,i3,3f10.3)') 'expn image',im,dx,dy,dz

write (24,'(a,3f9.3)') '; image 1 0 0 0 1 0 0 0 1', dx,dy,dz

do i= 1,natom1

x(i+i0)= x(i) + dx

y(i+i0)= y(i) + dy

z(i+i0)= z(i) + dz

do ipos= 1,mpos

xg(i+i0,ipos)= xg(i,ipos) + dx

yg(i+i0,ipos)= yg(i,ipos) + dy

zg(i+i0,ipos)= zg(i,ipos) + dz

do j= 1,6

anisog(j,i+i0,ipos)= anisog(j,i,ipos)

end do

end do

end do

end if

end do

end do

end do

close (24)

c \*\*\*\* double cell vectors \*\*\*\*

alen= alen \* 2.0

blen= blen \* 2.0

clen= clen \* 2.0

do i= 1,9

boxvecs(i)= boxvecs(i) \* 2.0

boxvecsinv(i)= boxvecsinv(i) / 2.0

end do

c \*\*\*\* find closeby images to 8-image set \*\*\*\*

write (6,\*) 'closest images to the 8-image set',natom

do i= 1,natom

do j= 1,i-1

dx= x(j) - x(i)

dy= y(j) - y(i)

dz= z(j) - z(i)

a= boxvecsinv(1)\*dx + boxvecsinv(4)\*dy + boxvecsinv(7)\*dz

b= boxvecsinv(5)\*dy + boxvecsinv(8)\*dz

c= boxvecsinv(9)\*dz

da= 0.0

if (a.gt. 0.5) da= -1.0

if (a.lt.-0.5) da= 1.0

db= 0.0

if (b.gt. 0.5) db= -1.0

if (b.lt.-0.5) db= 1.0

dc= 0.0

if (c.gt. 0.5) dc= -1.0

if (c.lt.-0.5) dc= 1.0

sx= boxvecs(1)\*da + boxvecs(4)\*db + boxvecs(7)\*dc

sy= boxvecs(5)\*db + boxvecs(8)\*dc

sz= boxvecs(9)\*dc

dx= dx + sx

dy= dy + sy

dz= dz + sz

if ((da.ne.0.0 .or. db.ne.0.0 .or. dc.ne.0.0) .and. dx\*\*2+dy\*\*2+dz\*\*2.lt.64.0) then

index= nint ((da+1) + (db+1)\*3 + (dc+1)\*9 )

write (6,\*) da,db,dc,index

if (.not.im8incl(index)) then

im8incl(index)= .true.

nhead= nhead + 1

if (nhead.gt.mh) stop 'header too big'

write (header(nhead),'(a,3f9.3)') '; image 1 0 0 0 1 0 0 0 1',sx,sy,sz

write (6,'(i4,a,a50)') index,' ',header(nhead)

end if

end if

end do

end do

call writehin ('config.hin',0)

c \*\*\*\* convert all indices from res/mol to absolute res number \*\*\*\*

nrest= 0

do imol= 1,nmol

do ires= 1,nres(imol)

nrest= nrest + 1

if (nrest.gt.mrest) stop 'mrest too small'

iarest(nrest)= iares(ires,imol)

jarest(nrest)= jares(ires,imol)

narest(nrest)= nares(ires,imol)

resnamet(nrest)= resname(ires,imol)

if (imol.eq.iedo1) iedor1= nrest

if (imol.eq.iedo2) iedor2= nrest

if (imol.eq.init1) initr1= nrest

if (imol.eq.init2) initr2= nrest

if (imol.eq.iwat1) iwatr1= nrest

if (imol.eq.iwat2) iwatr2= nrest

end do

end do

write (6,\*) 'total nber of residues in expanded system=',nrest

nrest1= nrest/mi

write (6,\*) 'new res indices for nitrates, 1,2-ethanediol and waters='

$ ,initr1,initr2,iedor1,iedor2,iwatr1,iwatr2

c \*\*\*\*\*\*\* find the lattice displ vectors for ints between nearest images of residues \*\*\*\*

c \*\*\*\*\*\*\* and included interaction list for each residue \*\*\*\*

nberint= 0

rcut= 5.0

write (6,\*) 'cuttoff distance for interactions=',rcut

rcut2= rcut\*\*2

do ires= 1,nrest

i1= iarest(ires)

do jres= 1,nrest

j1= iarest(jres)

dx= x(j1) - x(i1)

dy= y(j1) - y(i1)

dz= z(j1) - z(i1)

a= boxvecsinv(1)\*dx + boxvecsinv(4)\*dy + boxvecsinv(7)\*dz

b= boxvecsinv(5)\*dy + boxvecsinv(8)\*dz

c= boxvecsinv(9)\*dz

da= 0.0

if (a.gt. 0.5) da= -1.0

if (a.lt.-0.5) da= 1.0

db= 0.0

if (b.gt. 0.5) db= -1.0

if (b.lt.-0.5) db= 1.0

dc= 0.0

if (c.gt. 0.5) dc= -1.0

if (c.lt.-0.5) dc= 1.0

xoff(ires,jres)= boxvecs(1)\*da + boxvecs(4)\*db + boxvecs(7)\*dc

yoff(ires,jres)= boxvecs(5)\*db + boxvecs(8)\*dc

zoff(ires,jres)= boxvecs(9)\*dc

C if (ires.eq.nrest) write (6,'(2i5,3f8.2,6f6.2,3f8.2)') ires,jres,dx,dy,dz,a,b,c,da,db,dc,

C $ xoff(ires,jres),yoff(ires,jres),zoff(ires,jres)

c \*\*\*\* find shortest int distance between all atoms in residues \*\*\*\*

if (ires.lt.jres) then

do i= iarest(ires),jarest(ires)

do j= iarest(jres),jarest(jres)

dx= x(j) - x(i) + xoff(ires,jres)

dy= y(j) - y(i) + yoff(ires,jres)

dz= z(j) - z(i) + zoff(ires,jres)

r2= dx\*\*2 + dy\*\*2 + dz\*\*2

C write (6,\*) i,j,r2

if (r2.lt.1.0) stop 'short distance error'

if (r2.lt.rcut2) then

c \*\*\*\* add to interaction lists \*\*\*\*

nberint(ires)= nberint(ires) + 1

if (nberint(ires).gt.mint) stop 'mint too small'

intlist(nberint(ires),ires)= jres

nberint(jres)= nberint(jres) + 1

if (nberint(jres).gt.mint) stop 'mint too small'

intlist(nberint(jres),jres)= ires

C write (6,'(a,2i5,4f8.3,2i4)') 'int found',ires,jres,dx,dy,dz,r2,nberint(ires),nberint(jres)

goto 20

end if

end do

end do

20 continue

end if

end do

end do

maxnberint= 0

do ires= 1,nrest

write (6,\*) resnamet(ires),ires,' nber interacting residues='

$ ,nberint(ires)

maxnberint= max (maxnberint,nberint(ires))

end do

write (6,\*) 'maximum nber of interactions of a

$ residue with others=',maxnberint

c \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Read in Lennard-Jones parameters \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

continue

open (71,file='LJ.dat',status='old')

ntypelj=0

21 continue

read (71,'(a)',end=22) row

ntypelj=ntypelj+1

len=len\_trim(row)

datatype(ntypelj)=row(1:2)

read(row(4:16),\*) sig(ntypelj),eps(ntypelj)

goto 21

close(71)

22 continue

do i=1,ntypelj

do j=1,ntypelj

eps2(i,j)=sqrt(eps(i)\*eps(j))

sig2(i,j)=(sig(i)+sig(j))

end do

end do

do i=1,natom

do j=1,ntypelj

if (attype(i).eq.datatype(j)) then

iattype(i)=j

write(6,'(a7,a2,3x,a12,i5)') 'attype:',attype(i),

$ 'atom number:',i

end if

end do

end do

c \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Read in groups \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

open (72,file='groups.dat',status='old')

23 continue

read (72,'(a)',end=24) row

len=len\_trim(row)

read (row(1:6),\*) igroup,iingroup

ngroup=igroup

grouptype(igroup)='RES'

read (row(7:9),\*) npos(igroup)

do i=1,npos(igroup)-1

read (row(11+(i-1)\*6:15+(i-1)\*6),\*) frtype(i,igroup)

end do

ningroup(igroup)=iingroup

do i=1,iingroup

read (72,'(a)') row

len=len\_trim(row)

read (row(4:7),\*) nbres

group(i,igroup)= nbres

if (row(1:3).eq.'HOH') grouptype(igroup)='HOH'

if (row(1:3).eq.'NO3') grouptype(igroup)='NO3'

if (row(1:3).eq.'EDO') grouptype(igroup)='EDO'

if (grouptype(igroup).eq.'RES') resgroup(nbres) = igroup

if (grouptype(igroup).eq.'RES') write (6,'(a,2i8)') ' RESGROUP',nbres,igroup

end do

goto 23

c123456789012345678901234567890123456789012345678901234567890

cRES

c 21 1 3 0.33 0.47

cSER 85

24 continue

close(72)

c \*\*\*\* Correction of residue numbers of EDO, NO3, HOH \*\*\*\*\*

iiedor=iedor1

iinitr=initr1

iiwatr=iwatr1

do igroup=1,ngroup

c \*\*\*\*\* 1,2-Ethanediols \*\*\*\*\*\*\*\*\*\*

if (grouptype(igroup).eq.'EDO') then

group(1,igroup)=iiedor

resgroup(iiedor)=igroup

iiedor=iiedor+1

end if

c \*\*\*\*\* Nitrates \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if (grouptype(igroup).eq.'NO3') then

group(1,igroup)=iinitr

resgroup(iinitr)=igroup

iinitr=iinitr+1

end if

c \*\*\*\*\* and Waters \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if (grouptype(igroup).eq.'HOH') then

group(1,igroup)=iiwatr

resgroup(iiwatr)=igroup

iiwatr=iiwatr+1

end if

end do

c \*\*\*\*\*\*\* Group numbers \*\*\*\*\*\*

do igroup=1,ngroup

if (grouptype(igroup).eq.'RES') igroupres2=igroup

if (grouptype(igroup).eq.'EDO') igroupedo2=igroup

if (grouptype(igroup).eq.'NO3') igroupno32=igroup

if (grouptype(igroup).eq.'HOH') igrouphoh2=igroup

end do

igroupres1= 1

igroupedo1= igroupres2+1

igroupno31= igroupedo2+1

igrouphoh1= igroupno32+1

write (6,\*) 'first group with residues:',igroupres1

write (6,\*) 'last group with residues:',igroupres2

write (6,\*) 'first group with 1,2-ethanediol:',igroupedo1

write (6,\*) 'last group with 1,2-ethanediol:',igroupedo2

write (6,\*) 'first group with nitrate:',igroupno31

write (6,\*) 'last group with nitrate:',igroupno32

write (6,\*) 'first group with water:',igrouphoh1

write (6,\*) 'last group with water:',igrouphoh2

write (6,\*) 'total number of groups in each copy:',ngroup

c \*\*\*\*\*\*\* Generate 7 copies of the groups \*\*\*\*\*\*\*\*\*

if(ngroup\*8.gt.mgroup) stop 'MGROUP'

ngroup1=ngroup

ngroup=8\*ngroup

do i=2,8

do igroup=1,ngroup1

grouptype(igroup+(i-1)\*ngroup1)=grouptype(igroup)

npos(igroup+(i-1)\*ngroup1)=npos(igroup)

ningroup(igroup+(i-1)\*ngroup1)=ningroup(igroup)

do j=1,npos(igroup)-1

frtype(j,igroup+(i-1)\*ngroup1)= frtype(j,igroup)

end do

do j=1,ningroup(igroup)

group(j,igroup+(i-1)\*ngroup1)= group(j,igroup)+

$ (i-1)\*nrest1

end do

end do

end do

do i= 2,8

do igroup=1,ngroup1

do iingroup=1,ningroup(igroup)

resgroup(group(iingroup,igroup)+(i-1)\*nrest1)=

$ igroup+(i-1)\*ngroup1

end do

end do

end do

c write (6,'(a7)') 'GROUPS:'

c do igroup=1,ngroup

c write (6,'(a17,i5,a1)') 'Residues in group',igroup,':'

c do iingroup=1,ningroup(igroup)

c write (6,'(a8,i5,4x,a13,i5)') 'residue:',group(iingroup

c $ ,igroup)

c end do

c end do

c \*\*\*\*\*\*\*\*\*\* Generate intial configuration \*\*\*\*\*\*\*\*\*\*\*\*\*\*

config= 0

iswater=.false.

do igroup1 = 1, ngroup1

do ipos = 1, npos(igroup1) - 1

nbrconf = nint (frtype(ipos,igroup1)\*8)

jkm=9

if (npos(igroup1).eq.3) then

do i= 1,8

if (config(igroup1+(i-1)\*ngroup1).ne.0) jkm=jkm-1

end do

end if

do i = 1, nbrconf

ichange = ceiling (rand(0)\*(jkm-i))

nkm=0

do im = 1,8

c \*\*\*\*\*\*\*\*\*\*\* RES \*\*\*\*\*\*\*\*\*\*

if (grouptype(igroup1).eq.'RES') then

if (config(igroup1+(im-1)\*ngroup1).eq.0) nkm=nkm+1

if (config(igroup1+(im-1)\*ngroup1).eq.0 .and.

$ ichange.eq.nkm) then

config(igroup1+(im-1)\*ngroup1)= ipos

end if

c \*\*\*\*\*\*\*\*\*\*\* EDO \*\*\*\*\*\*\*\*\*\*\*

else if (grouptype(igroup1).eq.'EDO') then

if (config(igroup1+(im-1)\*ngroup1).eq.0) nkm=nkm+1

if (config(igroup1+(im-1)\*ngroup1).eq.0 .and.

$ ichange.eq.nkm) then

config(igroup1+(im-1)\*ngroup1)= ipos

end if

c \*\*\*\*\*\*\*\*\*\*\* NO3 \*\*\*\*\*\*\*\*\*\*\*\*

else if (grouptype(igroup1).eq.'NO3') then

if (config(igroup1+(im-1)\*ngroup1).eq.0) nkm=nkm+1

if (config(igroup1+(im-1)\*ngroup1).eq.0 .and.

$ ichange.eq.nkm) then

config(igroup1+(im-1)\*ngroup1)= ipos

end if

c \*\*\*\*\*\*\*\*\*\*\* HOH \*\*\*\*\*\*\*\*\*\*\*\*\*

else if (grouptype(igroup1).eq.'HOH') then

config(igroup1+(im-1)\*ngroup1) = ceiling (rand(0)\*72)

if (.not.iswater(igroup1+(im-1)\*ngroup1)) nkm= nkm+1

if (.not.iswater(igroup1+(im-1)\*ngroup1) .and.

$ ichange.eq.nkm) then

iswater (igroup1+(im-1)\*ngroup1)= .true.

end if

end if

end do

end do

end do

end do

do igroup = 1,ngroup

if (grouptype(igroup).eq.'RES' .and. config(igroup).eq.0) then

config(igroup)= npos(igroup)

else if (grouptype(igroup).eq.'EDO' .and. config(igroup).eq.0) then

config(igroup)= 4

else if (grouptype(igroup).eq.'NO3' .and. config(igroup).eq.0) then

config(igroup)= 4

end if

end do

c \*\*\*\*\*\*\*\*\*\*\*\*\* Config for each residue \*\*\*\*\*\*\*\*\*\*\*\*\*

do irest=1,nrest

configres(irest)= 1

end do

do igroup=1,ngroup

do iingroup=1,ningroup(igroup)

configres(group(iingroup,igroup))= config(igroup)

end do

end do

c do irest=1,nrest

c if (grouptype(resgroup(irest)).eq.'NO3') then

c if (configres(irest).eq.1) then

c write (6,'(a12,2x,a11)') resnamet(irest),'nitrate'

c else if (configres(irest).eq.4) then

c write (6,'(a12,2x,a11)') resnamet(irest),'not nitrate'

c end if

c end if

c end do

c do irest=1,nrest

c if (iswater(resgroup(irest))) then

c watertest=' WATER'

c else

c watertest='NOT WATER'

c end if

c if (config(resgroup(irest)).ne.0) then

c write (6,'(i2,a5,2x,a9)') config(resgroup(irest))

c $ ,grouptype(resgroup(irest)),watertest

c end if

c end do

c \*\*\*\*\*\*\*\*\*\*\*\* Calculate energy \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

do irest=1,nrest

do jrest=1,nrest

e(irest,jrest)=0.e0

end do

end do

do irest=1,nrest

ipos= configres(irest)

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

jpos= configres(jrest)

if (irest.lt.jrest) then

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (resgroup(irest).ne.-1) then

if (grouptype(resgroup(irest)).eq.'HOH' .and.

$ .not. iswater(resgroup(irest))) then

e(irest,jrest)= 0.e0

goto 301

end if

if (grouptype(resgroup(irest)).eq.'EDO' .or.

$ grouptype(resgroup(irest)).eq.'NO3') then

if (ipos.eq.4) then

e(irest,jrest)= 0.e0

goto 301

end if

end if

end if

if (resgroup(jrest).ne.-1) then

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

e(irest,jrest)= 0.e0

goto 301

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

e(irest,jrest)= 0.e0

goto 301

end if

end if

end if

call energy (irest,ipos,jrest,jpos)

e(irest,jrest)= enew(irest,jrest)

301 continue

e(jrest,irest)=e(irest,jrest)

end if

end do

end do

c do irest= 1,nrest

c do jrest= 1,nrest

c if (e(irest,jrest).ne.0) write (6,\*) 'ENERGY:',e(irest,jrest)

c $ ,irest,resnamet(irest)(1:3),jrest,resnamet(jrest)(1:3)

c end do

c end do

c \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Main MC computation loop \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ihin=10

kn= 1

31 continue

staten = 0.d0

stattot= 0.d0

statgroup= 0.d0

statgracc= 0.d0

statacc= 0.d0

c \*\*\*\*\*\*\*\*\*\*\*\*\*\* Statistics \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

write(statname,'(a6,a,a5)') 'config',ihinA,'.stat'

open (11, file=statname, status='unknown')

write (11,\*) 'INITIAL CONFIG:'

do igroup1 = 1, ngroup1

stat1 = 0.0

stat2 = 0.0

stat3 = 0.0

do i=1,8

igroup= igroup1 + (i-1)\*ngroup1

if (grouptype(igroup).eq.'RES'.and. npos(igroup).eq.2) then

if (config(igroup).eq.1) stat1 = stat1 + 1.0

if (config(igroup).eq.2) stat2 = stat2 + 1.0

else if (grouptype(igroup).eq.'RES' .and.

$ npos(igroup).eq.3) then

if (config(igroup).eq.1) stat1 = stat1 + 1.0

if (config(igroup).eq.2) stat2 = stat2 + 1.0

if (config(igroup).eq.3) stat3 = stat3 + 1.0

else if (grouptype(igroup).eq.'NO3') then

if (config(igroup).eq.1) stat1 = stat1 + 1.0

if (config(igroup).eq.4) stat2 = stat2 + 1.0

else if (grouptype(igroup).eq.'EDO') then

if (config(igroup).eq.1) stat1 = stat1 + 1.0

if (config(igroup).eq.4) stat2 = stat2 + 1.0

else if (grouptype(igroup).eq.'HOH') then

if (iswater(igroup)) stat1 = stat1 + 1.0

if (.not. iswater(igroup)) stat2 = stat2 + 1.0

end if

end do

write (11,'(i4,2x,a)') igroup1,grouptype(igroup1)

write (11,700) stat1/8.0,frtype(1,igroup1)

if (npos(igroup1).eq.3) then

write (11,700) stat2/8.0, frtype(2,igroup1)

write (11,700) stat3/8.0, 1.0-(frtype(1,igroup1) +

$ frtype(2,igroup))

else

write (11,700) stat2/8.0, 1.0- frtype(1,igroup1)

end if

write (11,\*)

end do

write (11,\*)

write (11,\*) '------------------------------------------------------'

write (11,\*)

write (11,\*) 'INITIAL MOLECULES:'

do igroup1= 1, ngroup1

if (grouptype(igroup1).ne.'HOH') then

write (11,'(8(i2),i4)') (config(igroup1+(i-1)\*ngroup1),i=1,8)

$ ,igroup1

else

write (11,\*) (iswater(igroup1+(i-1)\*ngroup1),i=1,8),igroup1

end if

end do

c do igroup=1,ngroup

c if (grouptype(igroup).eq.'HOH' .and. .not. iswater(igroup)) then

c write (11,'(i5,2x,a)') igroup,'not water'

c else

c write (11,'(i5,2x,a,2x,i3)') igroup,'config',config(igroup)

c end if

c end do

c write (11,\*)

c \*\*\*\*\*\*\*\*\*\*\* Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ktot= 100000000

c ktot=100

k1= 35.0

k2= 3.0

k3= 9.0

k4= 170.0

k5= 170.0\*10.0

kstart = kn

do k= 1, ktot

stattot = stattot+1.d0

if (k .le. 2\*ktot/10) temp = 4000.e0+273.15e0

if (k .gt. 2\*ktot/10 .and. k.le.4\*ktot/10) temp=3900.e0+273.15e0

if (k .gt. 4\*ktot/10 .and. k.le.6\*ktot/10) temp=3800.e0+273.15e0

if (k .gt. 6\*ktot/10 .and. k.le.8\*ktot/10) temp=3500.e0+273.15e0

if (k .gt. 8\*ktot/10 .and. k.le.9\*ktot/10) temp=0.00001e0

if (k .gt. 9\*ktot/10) temp = 0.00001e0

accept = .false.

enold = 0.e0

ennew = 0.e0

im= ceiling (rand(0)\*8)

irn= ceiling (rand(0)\*(k1+k2+k3+k4+k5))

nkm=0

c \*\*\*\*\*\*\* RES \*\*\*\*\*\*\*\*\*

if (irn.le.k1) then

igroup1 = ceiling (rand(0)\*(igroupres2 - igroupres1 + 1))+

$ igroupres1 - 1

statgracc (2,igroup1)= statgracc (2,igroup1) + 1.d0

igroup= igroup1+(im-1)\*ngroup1

if (npos(igroup).eq.2) then

if (config(igroup).eq.1) then

igroupnew= 2

jkm = ceiling (rand(0)\*nint ((1-frtype(1,igroup1))\*8))

do km= 1,8

if (config(igroup1+(km-1)\*ngroup1).eq.2) nkm= nkm + 1

if (config(igroup1+(km-1)\*ngroup1).eq.2

$ .and. nkm.eq.jkm) then

kgroupnew= config(igroup)

kgroup = igroup1 + (km-1)\*ngroup1

end if

end do

else if (config(igroup).eq.2) then

igroupnew= 1

jkm = ceiling (rand(0)\*nint(frtype(1,igroup1)\*8))

do km= 1,8

if (config(igroup1+(km-1)\*ngroup1).eq.1) nkm= nkm + 1

if (config(igroup1+(km-1)\*ngroup1).eq.1

$ .and. nkm.eq.jkm) then

kgroupnew= config(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

end if

else if (npos(igroup).eq.3) then

if (config(igroup).eq.1) then

inew = ceiling (rand(0)\*2)

if (inew.eq.1) igroupnew= 2

if (inew.eq.2) igroupnew= 3

if (igroupnew.eq.2) then

jkm= ceiling (rand(0)\*nint(frtype(2,igroup1)\*8))

do km= 1,8

if (config(igroup1+(km-1)\*ngroup1).eq.2) nkm= nkm+1

if (config(igroup1+(km-1)\*ngroup1).eq.2

$ .and. nkm.eq.jkm) then

kgroupnew= config(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

else if (igroupnew.eq.3) then

jkm= ceiling (rand(0)\*(8-nint (frtype(1,igroup1)\*8)-

$ nint (frtype(2,igroup1)\*8)))

do km=1,8

if (config(igroup1+(km-1)\*ngroup1).eq.3) nkm= nkm + 1

if (config(igroup1+(km-1)\*ngroup1).eq.3

$ .and. jkm.eq.nkm) then

kgroupnew= config(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

end if

else if (config(igroup).eq.2) then

inew= ceiling (rand(0)\*2)

if (inew.eq.1) igroupnew= 1

if (inew.eq.2) igroupnew= 3

if (igroupnew.eq.1) then

jkm= ceiling (rand(0)\*nint(frtype(1,igroup1)\*8))

do km= 1,8

if (config(igroup1+(km-1)\*ngroup1).eq.1) nkm= nkm + 1

if (config(igroup1+(km-1)\*ngroup1).eq.1

$ .and. jkm.eq.nkm) then

kgroupnew= config(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

else if (igroupnew.eq.3) then

jkm= ceiling (rand(0)\*(8-nint(frtype(1,igroup1)\*8)

$ -nint(frtype(2,igroup1)\*8)))

do km= 1,8

if (config(igroup1+(km-1)\*ngroup1).eq.3) nkm= nkm + 1

if (config(igroup1+(km-1)\*ngroup1).eq.3

$ .and. jkm.eq.nkm) then

kgroupnew= config(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

end if

else

inew= ceiling (rand(0)\*2)

if (inew.eq.1) igroupnew= 1

if (inew.eq.2) igroupnew= 2

if (igroupnew.eq.1) then

jkm= ceiling (rand(0)\*nint (frtype(1,igroup1)\*8))

do km= 1,8

if (config(igroup1+(km-1)\*ngroup1).eq.1) nkm= nkm + 1

if (config(igroup1+(km-1)\*ngroup1).eq.1

$ .and. jkm.eq.nkm) then

kgroupnew= config(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

else if (igroupnew.eq.2) then

jkm= ceiling (rand(0)\*nint(frtype(2,igroup1)\*8))

do km= 1,8

if (config(igroup1+(km-1)\*ngroup1).eq.2) nkm= nkm + 1

if (config(igroup1+(km-1)\*ngroup1).eq.2

$ .and. jkm.eq.nkm) then

kgroupnew= config(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

end if

end if

end if

c write(6,\*) 'CONF:',config(igroup),config(kgroup)

c if (config(igroup).eq.igroupnew) then

c write (6,\*) grouptype(igroup),' ','config',config(igroup),'igroup',

c $ igroup,'igroupnew', igroupnew,'im',im,frtype(1,igroup)

c $ ,igroup-(im-1)\*ngroup1,

c goto 32

c end if

c \*\*\* Calculating the new energies and test if accept or reject the step \*\*\*

c \*\*\*\*\*\*\*\*\* igroup \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

write (6,\*) 'IGROUP',igroup,igroupnew,ngriup,ngroup1

do iingroup=1,ningroup(igroup)

irest= group(iingroup,igroup)

ipos= igroupnew

do iint=1,nberint(irest)

jrest=intlist(iint,irest)

if (resgroup(jrest).eq.igroup) then

jpos= igroupnew

const= 0.5e0

else

jpos= configres(jrest)

write (6,'(a,5i8)') 'JPOS=',jpos,jrest,ipos,irest,iint

const= 1.0e0

end if

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (resgroup(jrest).ne.-1) then

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(irest,jrest)= 0.e0

goto 302

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(irest,jrest)= 0.e0

goto 302

end if

end if

end if

call energy (irest,ipos,jrest,jpos)

302 continue

c write (6,\*) 'RES:',resnamet(irest),resnamet(jrest),

c $ configres(jrest),'e:',e(irest,jrest),'enew',enew(irest,jrest)

c if (grouptype(resgroup(jrest)).eq.'HOH') then

c write (6,\*) iswater(resgroup(jrest))

c end if

enold= enold + const\*e(irest,jrest)

ennew= ennew + const\*enew(irest,jrest)

end do

end do

c \*\*\*\*\*\*\* kgroup \*\*\*\*\*\*\*\*

do kingroup=1,ningroup(kgroup)

krest= group(kingroup,kgroup)

kpos= kgroupnew

do kint=1,nberint(krest)

jrest=intlist(kint,krest)

if (resgroup(jrest).eq.kgroup) then

jpos= kgroupnew

const= 0.5e0

else

jpos= configres(jrest)

const= 1.0e0

end if

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (resgroup(jrest).ne.-1) then

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(krest,jrest)= 0.e0

goto 3022

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(krest,jrest)= 0.e0

goto 3022

end if

end if

end if

call energy (krest,kpos,jrest,jpos)

3022 continue

c write (6,\*) 'RES:',resnamet(irest),resnamet(jrest),

c $ configres(jrest),'e:',e(irest,jrest),'enew',enew(irest,jrest)

c if (grouptype(resgroup(jrest)).eq.'HOH') then

c write (6,\*) iswater(resgroup(jrest))

c end if

enold= enold + const\*e(krest,jrest)

ennew= ennew + const\*enew(krest,jrest)

end do

end do

if (enold.ge.ennew) accept= .true.

if (enold.lt.ennew) then

if (rand(0).le.exp((enold-ennew)/(Rgas\*temp))) accept= .true.

end if

if (accept) then

statgracc(1,igroup1)= statgracc(1,igroup1) + 1.d0

config(igroup)= igroupnew

config(kgroup)= kgroupnew

statacc = statacc + 1.d0

do iingroup=1,ningroup(igroup)

irest=group(iingroup,igroup)

configres(irest)= igroupnew

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

e(irest,jrest)= enew(irest,jrest)

e(jrest,irest)= e(irest,jrest)

end do

end do

do kingroup=1,ningroup(kgroup)

krest=group(kingroup,kgroup)

configres(krest)= kgroupnew

do kint= 1,nberint(krest)

jrest= intlist(kint,krest)

e(krest,jrest)= enew(krest,jrest)

e(jrest,krest)= e(krest,jrest)

end do

end do

end if

c \*\*\*\*\*\*\* EDO \*\*\*\*\*\*\*\*\*

else if (irn.gt.k1 .and. irn.le.(k1+k2)) then

igroup1 = ceiling (rand(0)\*(igroupedo2 - igroupedo1 + 1))+

$ igroupedo1 - 1

igroup= igroup1+(im-1)\*ngroup1

if (config(igroup1+(im-1)\*ngroup1).eq.1) then

igroupnew= 4

if (nint ((1-frtype(1,igroup1))\*8).eq.0) goto 32

jkm = ceiling (rand(0)\*nint ((1-frtype(1,igroup1))\*8))

do km= 1,8

if (config(igroup1+(km-1)\*ngroup1).eq.4) nkm= nkm + 1

if (config(igroup1+(km-1)\*ngroup1).eq.4

$ .and. jkm.eq.nkm) then

kgroupnew= config(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

else if (config(igroup1+(im-1)\*ngroup1).eq.4) then

igroupnew= 1

jkm = ceiling (rand(0)\*nint(frtype(1,igroup1)\*8))

do km= 1,8

if (config(igroup1+(km-1)\*ngroup1).eq.1) nkm= nkm + 1

if (config(igroup1+(km-1)\*ngroup1).eq.1

$ .and. jkm.eq.nkm) then

kgroupnew= config(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

end if

c if (config(igroup).eq.igroupnew) then

c write (6,\*) grouptype(igroup),' ','config',config(igroup),'igroup',

c $ igroup,'igroupnew', igroupnew,'im',im,frtype(1,igroup)

c $ ,igroup-(im-1)\*ngroup1

c goto 32

c end if

c \*\*\* Calculating the new energies and test if accept or reject the step \*\*\*

c \*\*\*\*\* igroup \*\*\*\*\*\*\*\*\*

do iingroup=1,ningroup(igroup)

irest= group(iingroup,igroup)

ipos= igroupnew

do iint=1,nberint(irest)

jrest=intlist(iint,irest)

jpos = configres(jrest)

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (resgroup(jrest).ne.-1) then

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(irest,jrest)= 0.e0

goto 303

end if

if (resgroup(jrest).eq.kgroup) then

enew(irest,jrest)= 0.e0

goto 303

end if

if (ipos.eq.4) then

enew(irest,jrest)= 0.e0

goto 303

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(irest,jrest)= 0.e0

goto 303

end if

end if

end if

call energy (irest,ipos,jrest,jpos)

303 continue

c write (6,\*) 'EDO:',resnamet(irest),resnamet(jrest),

c $ configres(jrest),'enew:',enew(irest,jrest),'e:',e(irest,jrest)

c if (grouptype(resgroup(jrest)).eq.'HOH') then

c write (6,\*) iswater(resgroup(jrest))

c end if

enold= enold + e(irest,jrest)

ennew= ennew + enew(irest,jrest)

end do

end do

c \*\*\*\*\*\*\*\*\* kgroup \*\*\*\*\*\*\*\*\*\*\*

do kingroup=1,ningroup(kgroup)

krest= group(kingroup,kgroup)

kpos= kgroupnew

do kint=1,nberint(krest)

jrest=intlist(kint,krest)

jpos = configres(jrest)

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (resgroup(jrest).ne.-1) then

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(krest,jrest)= 0.e0

goto 3032

end if

if (resgroup(jrest).eq.igroup) then

enew(krest,jrest)= 0.e0

goto 3032

end if

if (kpos.eq.4) then

enew(krest,jrest)= 0.e0

goto 3032

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(krest,jrest)= 0.e0

goto 3032

end if

end if

end if

call energy (krest,kpos,jrest,jpos)

3032 continue

c write (6,\*) 'EDO:',resnamet(irest),resnamet(jrest),

c $ configres(jrest),'enew:',enew(irest,jrest),'e:',e(irest,jrest)

c if (grouptype(resgroup(jrest)).eq.'HOH') then

c write (6,\*) iswater(resgroup(jrest))

c end if

enold= enold + e(krest,jrest)

ennew= ennew + enew(krest,jrest)

end do

end do

c \*\*\*\*\*\*\* Test the new energy \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if (enold.ge.ennew) accept= .true.

if (enold.lt.ennew) then

if (rand(0).le.exp((enold-ennew)/(Rgas\*temp))) accept= .true.

end if

statgracc(2,igroup1)= statgracc(2,igroup1) + 1.d0

if (accept) then

statgracc(1,igroup1)= statgracc(1,igroup1) + 1.d0

config(igroup)= igroupnew

config(kgroup)= kgroupnew

statacc = statacc + 1.d0

do iingroup=1,ningroup(igroup)

irest=group(iingroup,igroup)

configres(irest)= igroupnew

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

e(irest,jrest)= enew(irest,jrest)

e(jrest,irest)= e(irest,jrest)

end do

end do

do kingroup=1,ningroup(kgroup)

krest=group(kingroup,kgroup)

configres(krest)= kgroupnew

do kint= 1,nberint(krest)

jrest= intlist(kint,krest)

e(krest,jrest)= enew(krest,jrest)

e(jrest,krest)= e(krest,jrest)

end do

end do

end if

c \*\*\*\*\*\*\* NO3 \*\*\*\*\*\*

else if (irn.gt.(k1+k2) .and. irn.le.(k1+k2+k3)) then

igroup1 = ceiling (rand(0)\*(igroupno32 - igroupno31 + 1))+

$ igroupno31 - 1

igroup= igroup1+(im-1)\*ngroup1

if (config(igroup1+(im-1)\*ngroup1).eq.1) then

igroupnew= 4

if (nint((1-frtype(1,igroup1))\*8).eq.0) goto 32

jkm = ceiling (rand(0)\*nint ((1-frtype(1,igroup1))\*8))

do km= 1,8

if (config(igroup1+(km-1)\*ngroup1).eq.4) nkm= nkm + 1

if (config(igroup1+(km-1)\*ngroup1).eq.4

$ .and. jkm.eq.nkm) then

kgroupnew= config(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

else if (config(igroup1+(im-1)\*ngroup1).eq.4) then

igroupnew= 1

jkm = ceiling (rand(0)\*nint(frtype(1,igroup1)\*8))

do km= 1,8

if (config(igroup1+(km-1)\*ngroup1).eq.1) nkm= nkm + 1

if (config(igroup1+(km-1)\*ngroup1).eq.1

$ .and. jkm.eq.nkm) then

kgroupnew= config(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

end if

c if (config(igroup).eq.igroupnew) then

c write (6,\*) grouptype(igroup),' ','config',config(igroup),'igroup',

c $ igroup,'igroupnew', igroupnew,'im',im,frtype(1,igroup)

c $ ,igroup-(im-1)\*ngroup1

c goto 32

c end if

c \*\*\* Calculating the new energies and test if accept or reject the step \*\*\*

c \*\*\*\*\*\* igroup \*\*\*\*\*\*\*

do iingroup=1,ningroup(igroup)

irest= group(iingroup,igroup)

ipos= igroupnew

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

if (resgroup(jrest).eq.kgroup) then

jpos= kgroupnew

else

jpos= configres(jrest)

end if

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (resgroup(jrest).ne.-1) then

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(irest,jrest)= 0.e0

goto 304

end if

if (resgroup(jrest).eq.kgroup) then

enew(irest,jrest)= 0.e0

goto 304

end if

if (ipos.eq.4) then

enew(irest,jrest)= 0.e0

goto 304

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(irest,jrest)= 0.e0

goto 304

end if

end if

end if

call energy (irest,ipos,jrest,jpos)

304 continue

c write (6,\*) 'NO3:',resnamet(irest),resnamet(jrest),

c $ configres(jrest),'enew:',enew(irest,jrest),'e:',e(irest,jrest)

c if (grouptype(resgroup(jrest)).eq.'HOH') then

c write (6,\*) iswater(resgroup(jrest))

c end if

enold= enold + e(irest,jrest)

ennew= ennew + enew(irest,jrest)

end do

end do

c \*\*\*\*\* kgroup \*\*\*\*\*\*

do kingroup=1,ningroup(kgroup)

krest= group(kingroup,kgroup)

kpos= kgroupnew

do kint= 1,nberint(krest)

jrest= intlist(kint,krest)

if (resgroup(jrest).eq.igroup) then

jpos= igroupnew

else

jpos= configres(jrest)

end if

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (resgroup(jrest).ne.-1) then

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(krest,jrest)= 0.e0

goto 305

end if

if (resgroup(jrest).eq.igroup) then

enew(krest,jrest)=0.e0

goto 305

end if

if (kpos.eq.4) then

enew(krest,jrest)= 0.e0

goto 305

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(krest,jrest)= 0.e0

goto 305

end if

end if

end if

call energy (krest,kpos,jrest,jpos)

305 continue

c write (6,\*) 'NO3:',resnamet(krest),resnamet(jrest),

c $ configres(jrest),'enew:',enew(krest,jrest),'e:',e(krest,jrest)

c if (grouptype(resgroup(jrest)).eq.'HOH') then

c write (6,\*) iswater(resgroup(jrest))

c end if

enold= enold + e(krest,jrest)

ennew= ennew + enew(krest,jrest)

end do

end do

c \*\*\*\*\*\*\*\*\*\* TEST IF ACCEPT OR REJECT STEP \*\*\*\*\*\*\*\*\*\*\*

if (enold.ge.ennew) accept= .true.

if (enold.lt.ennew) then

if (rand(0).le.exp((enold-ennew)/(Rgas\*temp))) accept= .true.

end if

statgracc(2,igroup1)= statgracc(2,igroup1) + 1.d0

if (accept) then

statgracc(1,igroup1)= statgracc(1,igroup1) + 1.d0

config(igroup)= igroupnew

config(kgroup)= kgroupnew

statacc = statacc +1.d0

do iingroup=1,ningroup(igroup)

irest=group(iingroup,igroup)

configres(irest)= igroupnew

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

e(irest,jrest)= enew(irest,jrest)

e(jrest,irest)= e(irest,jrest)

end do

end do

do kingroup= 1, ningroup(kgroup)

krest= group(kingroup,kgroup)

configres(krest)= kgroupnew

do kint= 1, nberint(krest)

jrest= intlist(kint,krest)

e(krest,jrest)= enew(krest,jrest)

e(jrest,krest)= e(krest,jrest)

end do

end do

end if

c \*\*\*\*\*\*\* Water \*\*\*\*\*\*

else if (irn.gt.(k1+k2+k3) .and. irn.le.(k1+k2+k3+k4)) then

igroup1 = ceiling (rand(0)\*(igrouphoh2 - igrouphoh1 + 1))+

$ igrouphoh1 - 1

igroup= igroup1+(im-1)\*ngroup1

if (iswater(igroup)) then

iswaternew= .false.

if (nint ((1-frtype(1,igroup1))\*8).eq.0) goto 32

jkm = ceiling (rand(0)\*nint ((1-frtype(1,igroup1))\*8))

do km= 1,8

if (.not.(iswater(igroup1+(km-1)\*ngroup1))) nkm= nkm + 1

if (.not.(iswater(igroup1+(km-1)\*ngroup1))

$ .and. jkm.eq.nkm) then

kwaternew= iswater(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

else if (.not.(iswater(igroup1+(im-1)\*ngroup1))) then

iswaternew= .true.

jkm = ceiling (rand(0)\*nint(frtype(1,igroup1)\*8))

do km= 1,8

if (iswater(igroup1+(km-1)\*ngroup1)) nkm= nkm + 1

if (iswater(igroup1+(km-1)\*ngroup1)

$ .and. jkm.eq.nkm) then

kwaternew= iswater(igroup)

kgroup= igroup1 + (km-1)\*ngroup1

end if

end do

end if

c \*\*\* Calculating the new energies and test if accept or reject the step \*\*\*

c \*\*\*\* igroup \*\*\*\*\*

do iingroup=1,ningroup(igroup)

irest= group(iingroup,igroup)

ipos= configres(irest)

do iint=1,nberint(irest)

jrest=intlist(iint,irest)

jpos= configres(jrest)

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (.not.iswaternew) then

enew(irest,jrest)= 0.e0

goto 306

end if

if (resgroup(jrest).eq.kgroup) then

enew(irest,jrest)= 0.e0

goto 306

end if

if (resgroup(jrest).ne.-1) then

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(irest,jrest)= 0.e0

goto 306

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(irest,jrest)= 0.e0

goto 306

end if

end if

end if

call energy (irest,ipos,jrest,jpos)

306 continue

c write (6,'(a,a,2x,a,i5,2x,2(2x,a,2x,f11.7))') 'HOH:'

c $ ,resnamet(irest),resnamet(jrest),

c $ configres(jrest),'enew:',enew(irest,jrest),'e:',e(irest,jrest)

c write (6,\*) 'iswaternew',iswaternew,' ','iswater'

c $ ,iswater(resgroup(irest))

c if (grouptype(resgroup(jrest)).eq.'HOH') then

c write (6,\*) 'iswater(resgroup(jrest))',iswater(resgroup(jrest))

c end if

enold= enold + e(irest,jrest)

ennew= ennew + enew(irest,jrest)

end do

end do

c \*\*\*\*\* kgroup \*\*\*\*\*

do kingroup=1,ningroup(kgroup)

krest= group(kingroup,kgroup)

kpos= configres(krest)

do kint=1,nberint(krest)

jrest=intlist(kint,krest)

jpos= configres(jrest)

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (.not.kwaternew) then

enew(krest,jrest)= 0.e0

goto 3062

end if

if (resgroup(jrest).eq.igroup) then

enew(krest,jrest)= 0.e0

goto 3062

end if

if (resgroup(jrest).ne.-1) then

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(krest,jrest)= 0.e0

goto 3062

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(krest,jrest)= 0.e0

goto 3062

end if

end if

end if

call energy (krest,kpos,jrest,jpos)

3062 continue

enold= enold + e(krest,jrest)

ennew= ennew + enew(krest,jrest)

end do

end do

if (enold.ge.ennew) accept= .true.

if (enold.lt.ennew) then

if (rand(0).le.exp((enold-ennew)/(Rgas\*temp))) accept= .true.

end if

statgracc(2,igroup1)=statgracc(2,igroup1) + 1.d0

if (accept) then

statgracc(1,igroup1)= statgracc(1,igroup1) + 1.d0

iswater(igroup)= iswaternew

iswater(kgroup)= kwaternew

statacc = statacc + 1.d0

do iingroup=1,ningroup(igroup)

irest=group(iingroup,igroup)

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

e(irest,jrest)= enew(irest,jrest)

e(jrest,irest)= e(irest,jrest)

end do

end do

do kingroup=1,ningroup(kgroup)

krest=group(kingroup,kgroup)

do kint= 1,nberint(krest)

jrest= intlist(kint,krest)

e(krest,jrest)= enew(krest,jrest)

e(jrest,krest)= e(krest,jrest)

end do

end do

end if

c \*\*\*\*\*\*\* Water config \*\*\*\*\*\*

else

nbr= 0

do igroup2= igrouphoh1+(im-1)\*ngroup1,igrouphoh2+

$ (im-1)\*ngroup1

if (iswater(igroup2)) nbr=nbr+1

end do

jgroup= ceiling (rand(0)\*nbr)

nbr= 0

do igroup2= igrouphoh1+(im-1)\*ngroup1,igrouphoh2+(im-1)\*ngroup1

if (iswater(igroup2)) nbr=nbr+1

if (iswater(igroup2) .and. nbr.eq.jgroup) then

igroup=igroup2

igroupnew = ceiling (rand(0)\*72)

end if

end do

c \*\*\*\*\* and rotation about x-, y- or z-axis \*\*\*\*\*\*\*\*\*\*

statgracc(4,igroup-(im-1)\*ngroup1)=

$ statgracc(4,igroup-(im-1)\*ngroup1) + 1.d0

irest= group(1,igroup)

iaxis= ceiling (rand(0)\*3)

theta= (rand(0) - 0.5e0)\*3.14159265e0/6.0e0

dx1= xg(iarest(irest)+1,igroupnew)-xg(iarest(irest),igroupnew)

dy1= yg(iarest(irest)+1,igroupnew)-yg(iarest(irest),igroupnew)

dz1= zg(iarest(irest)+1,igroupnew)-zg(iarest(irest),igroupnew)

dx2= xg(iarest(irest)+2,igroupnew)-xg(iarest(irest),igroupnew)

dy2= yg(iarest(irest)+2,igroupnew)-yg(iarest(irest),igroupnew)

dz2= zg(iarest(irest)+2,igroupnew)-zg(iarest(irest),igroupnew)

xnew(iarest(irest))= xg(iarest(irest),igroupnew)

ynew(iarest(irest))= yg(iarest(irest),igroupnew)

znew(iarest(irest))= zg(iarest(irest),igroupnew)

if (iaxis.eq.1) then

xnew(iarest(irest)+1)= xg(iarest(irest)+1,igroupnew)

ynew(iarest(irest)+1)= dy1 \* cos (theta) - dz1 \* sin (theta)

$ + yg(iarest(irest),igroupnew)

znew(iarest(irest)+1)= dy1 \* sin (theta) + dz1 \* cos (theta)

$ + zg(iarest(irest),igroupnew)

xnew(iarest(irest)+2)= xg(iarest(irest)+2,igroupnew)

ynew(iarest(irest)+2)= dy2 \* cos (theta) - dz2 \* sin (theta)

$ + yg(iarest(irest),igroupnew)

znew(iarest(irest)+2)= dy2 \* sin (theta) + dz2 \* cos (theta)

$ + zg(iarest(irest),igroupnew)

else if (iaxis.eq.2) then

xnew(iarest(irest)+1)= dx1 \* cos (theta) + dz1 \* sin (theta)

$ + xg(iarest(irest),igroupnew)

ynew(iarest(irest)+1)= yg(iarest(irest)+1,igroupnew)

znew(iarest(irest)+1)= -dx1 \* sin (theta) + dz1 \*cos (theta)

$ + zg(iarest(irest),igroupnew)

xnew(iarest(irest)+2)= dx2 \* cos (theta) + dz2 \* sin (theta)

$ + xg(iarest(irest),igroupnew)

ynew(iarest(irest)+2)= yg(iarest(irest)+2,igroupnew)

znew(iarest(irest)+2)= -dx2 \* sin (theta) + dz2 \*cos (theta)

$ + zg(iarest(irest),igroupnew)

else

xnew(iarest(irest)+1)= dx1 \* cos (theta) - dy1 \* sin (theta)

$ + xg(iarest(irest),igroupnew)

ynew(iarest(irest)+1)= dx1 \* sin (theta) + dy1 \* cos (theta)

$ + yg(iarest(irest),igroupnew)

znew(iarest(irest)+1)= zg(iarest(irest)+1,igroupnew)

xnew(iarest(irest)+2)= dx2 \* cos (theta) - dy2 \* sin (theta)

$ + xg(iarest(irest),igroupnew)

ynew(iarest(irest)+2)= dx2 \* sin (theta) + dy2 \* cos (theta)

$ + yg(iarest(irest),igroupnew)

znew(iarest(irest)+2)= zg(iarest(irest)+2,igroupnew)

end if

c write (6,\*) 'COORDS:'

c write (6,\*) xnew(iarest(irest)+1),xg(iarest(irest)+1,igroupnew)

c write (6,\*) ynew(iarest(irest)+1),yg(iarest(irest)+1,igroupnew)

c write (6,\*) znew(iarest(irest)+1),zg(iarest(irest)+1,igroupnew)

c write (6,\*) xnew(iarest(irest)+2),xg(iarest(irest)+2,igroupnew)

c write (6,\*) ynew(iarest(irest)+2),yg(iarest(irest)+2,igroupnew)

c write (6,\*) znew(iarest(irest)+2),zg(iarest(irest)+2,igroupnew)

c \*\*\* Calculating the new energies and test if accept or reject the step \*\*\*

do iingroup=1,ningroup(igroup)

irest= group(iingroup,igroup)

ipos= igroupnew

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

jpos= configres(jrest)

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (grouptype(resgroup(irest)).eq.'HOH' .and.

$ .not. iswater(resgroup(irest))) then

enew(irest,jrest)= 0.e0

goto 307

end if

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(irest,jrest)= 0.e0

goto 307

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(irest,jrest)= 0.e0

goto 307

end if

end if

etot=0.e0

do i= iarest(irest),jarest(irest)

do j= iarest(jrest),jarest(jrest)

dx= xnew(i)-xg(j,jpos)+xoff(jrest,irest)

dy= ynew(i)-yg(j,jpos)+yoff(jrest,irest)

dz= znew(i)-zg(j,jpos)+zoff(jrest,irest)

r2= dx\*\*2+dy\*\*2+dz\*\*2

r= sqrt(r2)

sigma2=(sig2(iattype(i),iattype(j)))\*\*2

r2s= r2/sigma2

r6= r2s\*r2s\*r2s

r12= r6\*r6

elj= eps2(iattype(i),iattype(j))\*(1.e0/r12-2.e0/r6)

ecoul=627.5095\*0.5291772086\*q(i)\*q(j)/r

etot=etot+elj+ecoul

end do

end do

enew(irest,jrest)= etot

307 continue

enold= enold + e(irest,jrest)

ennew= ennew + enew(irest,jrest)

end do

end do

if (enold.ge.ennew) accept= .true.

if (enold.lt.ennew) then

if (rand(0).le.exp ((enold-ennew)/(Rgas\*temp))) then

accept= .true.

end if

end if

if (accept) then

statgracc(3,igroup-(im-1)\*ngroup1)=

$ statgracc(3,igroup-(im-1)\*ngroup1) + 1.d0

config(igroup)= igroupnew

statacc= statacc + 1.d0

do iingroup=1,ningroup(igroup)

irest=group(iingroup,igroup)

configres(irest)= igroupnew

do ipos=1,72

do i=iarest(irest)+1,jarest(irest)

if (iaxis.eq.1) then

dyi= yg(i,ipos)-yg(iarest(irest),ipos)

dzi= zg(i,ipos)-zg(iarest(irest),ipos)

yg(i,ipos)= dyi \* cos (theta) - dzi \* sin (theta) +

$ yg(iarest(irest),ipos)

zg(i,ipos)= dyi \* sin (theta) + dzi \* cos (theta) +

$ zg(iarest(irest),ipos)

else if (iaxis.eq.2) then

dxi= xg(i,ipos)-xg(iarest(irest),ipos)

dzi= zg(i,ipos)-zg(iarest(irest),ipos)

xg(i,ipos)= dxi \* cos (theta) + dzi \* sin (theta) +

$ xg(iarest(irest),ipos)

zg(i,ipos)= - dxi \* sin (theta) + dzi \* cos (theta) +

$ zg(iarest(irest),ipos)

else

dxi= xg(i,ipos)-xg(iarest(irest),ipos)

dyi= yg(i,ipos)-yg(iarest(irest),ipos)

xg(i,ipos)= dxi \* cos (theta) - dyi \* sin (theta) +

$ xg(iarest(irest),ipos)

yg(i,ipos)= dxi \* sin (theta) + dyi \* cos (theta) +

$ yg(iarest(irest),ipos)

end if

end do

end do

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

e(irest,jrest)= enew(irest,jrest)

e(jrest,irest)= e(irest,jrest)

end do

end do

end if

end if

staten= staten + 1.d0

32 continue

c if (accept) watertest='T'

c if (.not.accept) watertest='F'

c write (6,\*) 'MONTE-CARLO:'

c write (6,'(a5,f9.3,2x,a5,f9.3,2x,a4,i5,2x,a3,2x,i2,2x,i2)')

c $ 'E\_new',ennew,'E\_old',

c $ enold,grouptype(igroup),igroup

c $ ,watertest

c $ ,igroupnew,config(igroup)

c write(6,\*)

c \*\*\*\*\*\*\*\* Statistics \*\*\*\*\*\*\*\*\*\*\*\*\*\*

if (k.eq.2\*ktot/10 .or. k.eq.4\*ktot/10 .or.

$ k.eq.6\*ktot/10 .or. k.eq.8\*ktot/10 .or.

$ k.eq.9\*ktot/10) then

write (11,'(a,2x,f6.0,a)') 'after temp:',temp,'K'

write (11,'(a,f11.0)') 'NUMBER OF CYCLES:',stattot

write (11,'(a,f10.0)') 'Number of energy tests:', staten

write (11,'(a,f10.0)') 'Number of accepted moves:', statacc

write (11,'(a,2(f7.5,a,2x))') 'Accepted moves:',statacc/stattot,

$ '(tot)', statacc/staten, '(energy tests)'

write (11,\*)

write (11,\*) '---------------------------------------------------'

write (11,\*)

end if

c \*\*\*\* Save results in 2VB1i.hin and config.dump \*\*\*\*\*\*

if ((mod (k,ktot/10) .eq. 0)) then

do irest= 1,nrest

do i= iarest(irest), jarest(irest)

ipos= configres(irest)

x(i)= xg(i,ipos)

y(i)= yg(i,ipos)

z(i)= zg(i,ipos)

do j= 1,6

aniso(j,i)= anisog(j,i,ipos)

end do

if (grouptype(resgroup(irest)).eq.'HOH' .and.

$ (.not.iswater(resgroup(irest)))) then

x(i)= 0.000

y(i)= 0.000

z(i)= 0.000

end if

if (grouptype(resgroup(irest)).eq.'NO3' .and.

$ config(resgroup(irest)).eq.4) then

x(i)= 0.000

y(i)= 0.000

z(i)= 0.000

end if

if (grouptype(resgroup(irest)).eq.'EDO' .and.

$ config(resgroup(irest)).eq.4) then

x(i)= 0.000

y(i)= 0.000

z(i)= 0.000

end if

end do

end do

goto 111

kn= k + 1

ihin=ihin+1

write (hinname,'(a5,i2,a4)') '2VB1i',ihin,'.hin'

call writehin (hinname,0)

open (1,file='config.dump',status='unknown',form='unformatted')

write (1) xg,yg,zg,xoff,yoff,zoff,eps2,sig2,frtype,e,enew

write (1) nberint,intlist,ntype,npos,group,config,iswater

write (1) rcut,natom1,nmol1,initr1,initr2,iwatr1,iwatr2

write (1) iedor1,iedor2,iattype,resgroup,grouptype

write (1) ngroup,ningroup,resgroup,igroupres1,igroupres2

write (1) igroupedo1,igroupedo2,igroupno31,igroupno32

write (1) igrouphoh1,igrouphoh2,configres

write (1) temp,kn,ihin,nrest,nrest1,ngroup1,resnamet

write (1) iarest,jarest,molind,molend

close (1)

end if

111 continue

c \*\*\*\*\* Update statistics \*\*\*\*\*\*

do igroup= 1, ngroup

ipos = config(igroup)

if (grouptype(igroup).eq.'HOH') then

if (.not. iswater(igroup)) then

goto 400

else

statwater(igroup) = statwater(igroup) + 1.d0

end if

end if

statgroup(ipos,igroup) = statgroup(ipos,igroup) + 1.d0

400 continue

end do

end do

enold=0.e0

do irest=1,nrest

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

if (irest.lt.jrest) then

enold= enold + e(irest,jrest)

end if

end do

end do

write (11,\*) 'SUM OF INTERRESIDUE ENERGIES BEFORE OPT:',enold

c \*\*\*\*\* res, edo, no3 \*\*\*\*\*\*\*\*\*\*\*

ien= -1

500 continue

ien= ien + 1

again=.false.

nhisto= 0

do i=1,2200

do j=1,4

histo(i,j)=0.0

end do

end do

do igroup1= 1, igroupno32

do im= 1,8

igroup= igroup1 +(im-1)\*ngroup1

do km= 1,8

if (im.lt.km) then

enold= 0.e0

ennew= 0.e0

kgroup= igroup1 +(km-1)\*ngroup1

if (config(igroup).ne.config(kgroup)) then

igroupnew= config(kgroup)

kgroupnew= config(igroup)

c \*\*\*\*\*\*\*\*\* igroup \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

do iingroup=1,ningroup(igroup)

irest= group(iingroup,igroup)

ipos= igroupnew

do iint=1,nberint(irest)

jrest=intlist(iint,irest)

if (resgroup(jrest).eq.igroup) then

jpos= igroupnew

const= 0.5e0

else if (resgroup(jrest).eq.kgroup) then

jpos= kgroupnew

const= 1.0e0

else

jpos= configres(jrest)

const= 1.0e0

end if

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(irest,jrest)= 0.e0

goto 4021

end if

if (grouptype(resgroup(irest)).eq.'EDO' .or.

$ grouptype(resgroup(irest)).eq.'NO3') then

if (ipos.eq.4) then

enew(irest,jrest)= 0.e0

goto 4021

end if

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(irest,jrest)= 0.e0

goto 4021

end if

end if

call energy (irest,ipos,jrest,jpos)

4021 continue

enold= enold + const\*e(irest,jrest)

ennew= ennew + const\*enew(irest,jrest)

end do

end do

c \*\*\*\*\*\*\* kgroup \*\*\*\*\*\*\*\*

do kingroup=1,ningroup(kgroup)

krest= group(kingroup,kgroup)

kpos= kgroupnew

do kint=1,nberint(krest)

jrest=intlist(kint,krest)

if (resgroup(jrest).eq.kgroup) then

jpos= kgroupnew

const= 0.5e0

else if (resgroup(jrest).eq.igroup) then

jpos= igroupnew

const= 1.0e0

else

jpos= configres(jrest)

const= 1.0e0

end if

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(krest,jrest)= 0.e0

goto 4022

end if

if (grouptype(resgroup(krest)).eq.'EDO' .or.

$ grouptype(resgroup(krest)).eq.'NO3') then

if (kpos.eq.4) then

enew(krest,jrest)= 0.e0

goto 4022

end if

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(krest,jrest)= 0.e0

goto 4022

end if

end if

call energy (krest,kpos,jrest,jpos)

4022 continue

enold= enold + const\*e(krest,jrest)

ennew= ennew + const\*enew(krest,jrest)

end do

end do

nhisto=nhisto + 1

histo(nhisto,1)= ennew-enold

histo(nhisto,2)= igroup1

histo(nhisto,3)= im

histo(nhisto,4)= km

if (ennew.lt.enold) then

again = .true.

config(igroup)= igroupnew

config(kgroup)= kgroupnew

do iingroup=1,ningroup(igroup)

irest=group(iingroup,igroup)

configres(irest)= igroupnew

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

e(irest,jrest)= enew(irest,jrest)

e(jrest,irest)= e(irest,jrest)

end do

end do

do kingroup=1,ningroup(kgroup)

krest=group(kingroup,kgroup)

configres(krest)= kgroupnew

do kint= 1,nberint(krest)

jrest= intlist(kint,krest)

e(krest,jrest)= enew(krest,jrest)

e(jrest,krest)= e(krest,jrest)

end do

end do

end if

end if

end if

end do

end do

end do

c \*\*\*\*\*\*\*\*\* and water \*\*\*\*\*\*\*\*\*\*\*\*

do igroup1= igrouphoh1, igrouphoh2

do im= 1,8

igroup= igroup1 +(im-1)\*ngroup1

do km = 1,8

enold= 0.e0

ennew= 0.e0

if (im.lt.km) then

kgroup= igroup1 + (km-1)\*ngroup1

if ((iswater(igroup).and. .not.iswater(kgroup))

$ .or. (.not.iswater(igroup).and.iswater(kgroup))) then

iswaternew= iswater(kgroup)

kwaternew= iswater(igroup)

c \*\*\*\*\*\*\*\*\* igroup \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

do iingroup=1,ningroup(igroup)

irest= group(iingroup,igroup)

ipos = configres(irest)

do iint=1,nberint(irest)

jrest=intlist(iint,irest)

jpos = configres(jrest)

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (.not.iswaternew) then

enew(irest,jrest)= 0.e0

goto 4031

end if

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(irest,jrest)= 0.e0

goto 4031

end if

if (grouptype(resgroup(irest)).eq.'EDO' .or.

$ grouptype(resgroup(irest)).eq.'NO3') then

if (ipos.eq.4) then

enew(irest,jrest)= 0.e0

goto 4031

end if

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(irest,jrest)= 0.e0

goto 4031

end if

end if

call energy (irest,ipos,jrest,jpos)

4031 continue

enold= enold + e(irest,jrest)

ennew= ennew + enew(irest,jrest)

end do

end do

c \*\*\*\*\*\*\* kgroup \*\*\*\*\*\*\*\*

do kingroup=1,ningroup(kgroup)

krest= group(kingroup,kgroup)

kpos= configres(krest)

do kint=1,nberint(krest)

jrest=intlist(kint,krest)

jpos= configres(jrest)

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (.not.kwaternew) then

enew(krest,jrest)= 0.e0

goto 4032

end if

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(krest,jrest)= 0.e0

goto 4032

end if

if (grouptype(resgroup(krest)).eq.'EDO' .or.

$ grouptype(resgroup(krest)).eq.'NO3') then

if (kpos.eq.4) then

enew(krest,jrest)= 0.e0

goto 4032

end if

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(krest,jrest)= 0.e0

goto 4032

end if

end if

call energy (krest,kpos,jrest,jpos)

4032 continue

enold= enold + e(krest,jrest)

ennew= ennew + enew(krest,jrest)

end do

end do

nhisto=nhisto+1

histo(nhisto,1)= ennew - enold

histo(nhisto,2)= igroup1

histo(nhisto,3)= im

histo(nhisto,4)= km

if (ennew.lt.enold) then

again = .true.

iswater(igroup)= iswaternew

iswater(kgroup)= kwaternew

do iingroup=1,ningroup(igroup)

irest= group(iingroup,igroup)

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

e(irest,jrest)= enew(irest,jrest)

e(jrest,irest)= e(irest,jrest)

end do

end do

do kingroup=1,ningroup(kgroup)

krest=group(kingroup,kgroup)

do kint= 1,nberint(krest)

jrest= intlist(kint,krest)

e(krest,jrest)= enew(krest,jrest)

e(jrest,krest)= e(krest,jrest)

end do

end do

end if

end if

end if

end do

end do

end do

if (again) goto 500

c \*\*\*\*\* Add a water to each nitrate hole \*\*\*\*

write (statname,'(a6,a,a5)') 'config',ihinA,'.hist'

open (12, file=statname, status='unknown')

c call watermc (1000000)

c \*\*\*\*\* Write the initial structure in a .hin file \*\*\*\*\*\*\*\*\*\*\*\*

do imol=8\*nmol1, 1, -1

if (resname(1,imol)(1:3).eq.'NO3' .or.

$ resname(1,imol)(1:3).eq.'EDO' .or.

$ resname(1,imol)(1:3).eq.'HOH') then

j1 = molind(imol)

j2 = molend(imol)

if (x(j1) .eq. 0 .and. y(j1) .eq. 0

$ .and. z(j1) .eq. 0 .and. x(j2) .eq. 0

$ .and. y(j2) .eq. 0 .and. z(j2) .eq. 0) then

call hdelmol (imol)

end if

end if

end do

write (hinname,'(a5,a,a4)') '2VB1i',ihinA,'.hin'

call writehin (hinname,0)

c \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Statistics \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

write (11,'(a,f11.0)') 'TOTAL NUMBER OF CYCLES:',stattot

write (11,'(a,f10.0)') 'Number of energy tests:', staten

write (11,'(a,f10.0)') 'Number of accepted moves:', statacc

write (11,'(a,2(f7.5,a,2x))') 'Accepted moves:',statacc/stattot, '(tot)',

$ statacc/staten, '(energy tests)'

write (11,\*)

do igroup1= igroupres1, igroupres2

statgracc(4,1)= statgracc(4,1) + statgracc(2,igroup1)

statgracc(3,1)= statgracc(3,1) + statgracc(1,igroup1)

end do

do igroup1= igroupedo1, igroupedo2

statgracc(4,2)= statgracc(4,2) + statgracc(2,igroup1)

statgracc(3,2)= statgracc(3,2) + statgracc(1,igroup1)

end do

do igroup1= igroupno31, igroupno32

statgracc(4,3)= statgracc(4,3) + statgracc(2,igroup1)

statgracc(3,3)= statgracc(3,3) + statgracc(1,igroup1)

end do

do igroup1= igrouphoh1, igrouphoh2

statgracc(4,4)= statgracc(4,4) + statgracc(2,igroup1)

statgracc(3,4)= statgracc(3,4) + statgracc(1,igroup1)

end do

do igroup1= igrouphoh1,igrouphoh2

statgracc(4,5)= statgracc(4,5) + statgracc(4,igroup1)

statgracc(3,5)= statgracc(3,5) + statgracc(3,igroup1)

end do

write (11,\*) '---------------------------------------------------'

write (11,\*)

write (11,\*) 'ACCEPTANCE RATE'

write (11,\*)

write (11,'(a,f6.4)') 'RES:', statgracc(3,1)/statgracc(4,1)

write (11,'(a,f6.4)') 'EDO:', statgracc(3,2)/statgracc(4,2)

write (11,'(a,f6.4)') 'NO3:', statgracc(3,3)/statgracc(4,3)

write (11,'(a,f6.4)') 'HOH:', statgracc(3,4)/statgracc(4,4)

write (11,'(a,f6.4)') 'HOH CONFIG:', statgracc(3,5)/statgracc(4,5)

write (11,\*)

do igroup1= 1, ngroup1

if (nint ((1-frtype(1,igroup1))\*8).ne.0) then

write (11,'(i3,2x,a)') igroup1,grouptype(igroup1)

write (11,'(f6.4)') statgracc(1,igroup1)/statgracc(2,igroup1)

write (11,\*)

end if

end do

write (11,\*)

write (11,\*) 'WATER CONFIG'

do igroup= igrouphoh1,igrouphoh2

write (11,'(a)') resnamet(group(1,igroup))

write (11,'(f6.4)') statgracc(3,igroup)/statgracc(4,igroup)

write (11,\*)

end do

write (11,\*) '---------------------------------------------------'

write (11,\*)

write (11,\*) 'END CONFIG:'

do igroup1 = 1, ngroup1

stat1 = 0.0

stat2 = 0.0

stat3 = 0.0

do i=1,8

igroup= igroup1 + (i-1)\*ngroup1

if (grouptype(igroup).eq.'RES'.and. npos(igroup).eq.2) then

if (config(igroup).eq.1) stat1 = stat1 + 1.0

if (config(igroup).eq.2) stat2 = stat2 + 1.0

else if (grouptype(igroup).eq.'RES' .and.

$ npos(igroup).eq.3) then

if (config(igroup).eq.1) stat1 = stat1 + 1.0

if (config(igroup).eq.2) stat2 = stat2 + 1.0

if (config(igroup).eq.3) stat3 = stat3 + 1.0

else if (grouptype(igroup).eq.'NO3') then

if (config(igroup).eq.1) stat1 = stat1 + 1.0

if (config(igroup).eq.4) stat2 = stat2 + 1.0

else if (grouptype(igroup).eq.'EDO') then

if (config(igroup).eq.1) stat1 = stat1 + 1.0

if (config(igroup).eq.4) stat2 = stat2 + 1.0

else if (grouptype(igroup).eq.'HOH') then

if (iswater(igroup)) stat1 = stat1 + 1.0

if (.not. iswater(igroup)) stat2 = stat2 + 1.0

end if

end do

write (11,'(i4,2x,a)') igroup1,grouptype(igroup1)

write (11,700) stat1/8.0,frtype(1,igroup1)

if (npos(igroup1).eq.3) then

write (11,700) stat2/8.0, frtype(2,igroup1)

write (11,700) stat3/8.0, 1.0-(frtype(1,igroup1) +

$ frtype(2,igroup))

else

write (11,700) stat2/8.0, 1.0- frtype(1,igroup1)

end if

write (11,\*)

700 format ('calc:',f5.3,3x,'exp:',f5.3)

end do

write (11,\*)

write (11,\*) '---------------------------------------------------'

write (11,\*) 'GROUPS:'

do igroup = 1, ngroup

write (11,\*)

write (11,'(a,2x,i5,2x,a,2x,a)') 'group:',igroup,'grouptype:',

$ grouptype(igroup)

do ipos = 1, npos(igroup)-1

if (grouptype(igroup).ne.'HOH') then

if (statgroup(ipos,igroup).ne.0.0) then

write (11,'(a,1x,f5.3,1x,a,1x,f5.3)') 'calc:'

$ ,statgroup(ipos,igroup)/stattot,

$ 'frtype:',frtype(ipos,igroup)

end if

else

write (11,'(a,1x,f5.3,1x,a,1x,f5.3)') 'calc:',

$ statwater(igroup)/stattot,

$ 'frtype:',frtype(ipos,igroup)

do jpos = 1,72

if (statgroup(jpos,igroup).ne.0.0) then

write (11,'(i2,3x,f6.4)') jpos,

$ statgroup(jpos,igroup)/stattot

end if

end do

end if

end do

end do

write (11,\*) '------------------------------------------------------'

write (11,\*)

do igroup=1,ngroup

if (grouptype(igroup).eq.'HOH' .and. .not. iswater(igroup)) then

write (11,'(i5,2x,a)') igroup,'not water'

else

write (11,'(i5,2x,a,2x,i3)') igroup,'config',config(igroup)

end if

end do

write (11,\*)

write (11,\*) '------------------------------------------------------'

write (11,\*)

write (11,\*) 'FINAL MOLECULES:'

do igroup1= 1, ngroup1

if (grouptype(igroup1).ne.'HOH') then

write (11,'(8(i2),i4)') (config(igroup1+(i-1)\*ngroup1),i=1,8)

$ ,igroup1

else

write (11,\*) (iswater(igroup1+(i-1)\*ngroup1),i=1,8),igroup1

end if

end do

enold=0.e0

do irest=1,nrest

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

if (irest.lt.jrest) then

enold= enold + e(irest,jrest)

end if

end do

end do

write (11,\*) 'SUM OF INTERRESIDUE ENERGIES AFTER OPT:',enold

c \*\*\*\*\* TEST IF SYMMETRIC \*\*\*\*\*\*\*

do irest=1,nrest

do jrest=1,nrest

if (e(irest,jrest).ne.e(jrest,irest)) then

write (11,\*) 'E-matrix',e(irest,jrest)-e(jrest,irest)

write (11,\*) grouptype(resgroup(irest)),

$ grouptype(resgroup(jrest))

write (11,\*) e(irest,jrest)

end if

end do

end do

c \*\*\*\* TEST IF ZERO ENERGY \*\*\*\*\*

do irest=1,nrest

igroup=resgroup(irest)

do jrest=1,nrest

if ((grouptype(igroup).eq.'HOH' .and.

$ .not.iswater(igroup)) .or.

$ (grouptype(igroup).eq.'NO3' .and.

$ config(igroup).eq.4)

$ .or.(grouptype(igroup).eq.'EDO' .and. config(igroup).eq.4))

$ then

if (e(irest,jrest).ne.0 .or. e(jrest,irest).ne.0) then

write (11,\*) 'E-matrix',grouptype(igroup),e(irest,jrest),

$ e(jrest,irest)

end if

end if

end do

end do

write (11,\*) 'NUMBER OF OPT CYCLES:',ien

alow= 0.e0

do i = 1, nhisto

if (histo(i,1).lt.alow) then

alow= histo(i,1)

igrouplow= histo(i,2)

imlow= histo(i,3)

kmlow= histo(i,4)

end if

end do

write (11,\*) 'LOWEST ENERGY PERMUT:'

write (11,\*) alow,igrouplow,imlow,kmlow

c \*\*\*\*\*\*\* Histogram \*\*\*\*\*\*\*

write (11,\*)

write (11,'(a,2x,f5.0)') 'NUMBER OF PERMUTATIONS',nhisto

do j= 1,nhisto

ihistres(j)= 0

end do

do j= 1,1000

histres(1,j)= (j-1)\*2- 1000

histres(2,j)= j\*2 - 1000

end do

do i= 1,nhisto

entest= histo(i,1)

do j=1,1000

if (entest .gt. histres(1,j)

$ .and. entest .le. histres(2,j)) then

ihistres(j)= ihistres(j) + 1

end if

end do

end do

write (11,\*) 'RESULTS:'

do j=1,1000

write (11,\*) (histres(i,j), i=1,2),ihistres(j)

end do

write (11,\*) 'EACH PERMUTATION:'

do j=1, nhisto

write (11,\*) (histo(j,i), i=1,4)

end do

write (11,\*) 'LESS THAN 4kcal/mol:'

do j=1, nhisto

if (histo(j,1).lt.4) then

write (11,\*) (histo(j,i), i=1,4)

end if

end do

write (11,\*) 'GREATER THAN 40kcal/mol:'

do j=1, nhisto

if (histo(j,1).gt.40) then

write (11,\*) (histo(j,i), i=1,4)

end if

end do

write (12,\*) 'ENERGY:'

do j=1,1000

write (12,\*) histres(1,j)

end do

write (12,\*) 'NUMBER:'

do j=1,1000

write (12,\*) ihistres(j)

end do

close(12)

close(11)

end

c \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

subroutine energy (irest,ipos,jrest,jpos)

implicit real\*8 (a-h,o-z)

include 'readhin.cmn'

include 'config.cmn'

etot=0.e0

do i=iarest(irest),jarest(irest)

do j=iarest(jrest),jarest(jrest)

dx=xg(i,ipos)-xg(j,jpos)+xoff(jrest,irest)

dy=yg(i,ipos)-yg(j,jpos)+yoff(jrest,irest)

dz=zg(i,ipos)-zg(j,jpos)+zoff(jrest,irest)

r2=dx\*\*2+dy\*\*2+dz\*\*2

r=sqrt(r2)

sigma2=(sig2(iattype(i),iattype(j)))\*\*2

r2s=r2/sigma2

r6=r2s\*r2s\*r2s

r12=r6\*r6

elj=eps2(iattype(i),iattype(j))\*(1.e0/r12-2.e0/r6)

ecoul=627.5095e0\*0.5291772086e0\*q(i)\*q(j)/r

etot=etot+elj+ecoul

end do

end do

enew(irest,jrest)=etot

return

end

c \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

subroutine watermc (ktot)

implicit real\*8 (a-h,o-z)

include 'readhin.cmn'

include 'config.cmn'

logical hblist(mrest,mrest),accept

integer configw(mrest,2),iattypew(m),wlist(mrest),ntype(mrest)

real\*8 ew(mrest,mrest)

real\*8 xw(m,mpos),yw(m,mpos),zw(m,mpos),qw(m)

real\*8 rtest(20,3)

c \*\*\*\*\*\*\* Make a HB-list \*\*\*\*\*\*\*\*\*\*\*

do irest=1,nrest

do jrest=1,nrest

hblist(irest,jrest)=.false.

end do

end do

do irest= 1,nrest

ipos=configres(irest)

do jrest= 1,nrest

if (irest.lt.jrest .and. iswater(resgroup(irest)) .and.

$ iswater(resgroup(jrest))) then

jpos=configres(jrest)

do i= iarest(irest),jarest(irest)

do j= iarest(jrest),jarest(jrest)

if (nat(i).ne.nat(j)) then

dx= xg(j,jpos) - xg(i,ipos) + xoff(irest,jrest)

dy= yg(j,jpos) - yg(i,ipos) + yoff(irest,jrest)

dz= zg(j,jpos) - zg(i,ipos) + zoff(irest,jrest)

r2= dx\*\*2 + dy\*\*2 + dz\*\*2

if (r2.lt.8.0) then

hblist(irest,jrest)=.true.

hblist(jrest,irest)=.true.

end if

end if

end do

end do

end if

end do

end do

c \*\*\*\*\*\*\* Add water if EDO is absent \*\*\*\*\*\*\*

write (11,\*) 'ADD WATER IF EDO NOT PRESENT:'

c \*\*\*\*\*\* Find ethylene glycol which is not present \*\*\*\*

do igroup1= 1 ,ngroup1

if (grouptype(igroup1).eq.'EDO') then

do im= 1,8

igroup= igroup1+(im-1)\*ngroup1

if (config(igroup).eq.4) then

c \*\*\*\*\*\*\*\* Copy energies from e to ew \*\*\*\*\*\*

do irest= 1, nrest

do jrest= 1, nrest

ew(irest,jrest) = e(irest,jrest)

end do

end do

c \*\*\*\*\*\* Copy charges, types and coords \*\*\*\*\*\*\*

do i=1,natom

qw(i) = q(i)

iattypew(i) = iattype(i)

do ipos= 1,72

xw(i,ipos) = xg(i,ipos)

yw(i,ipos) = yg(i,ipos)

zw(i,ipos) = zg(i,ipos)

end do

end do

c \*\*\*\*\* Copy configs for each residue, edo, nitrate and water \*\*\*\*

do irest= 1,nrest

configw(irest,1)= configres(irest)

ntype(irest)=1

if (resgroup(irest).eq.igroup) ntype(irest)=2

end do

c \*\*\*\* Change charges, types, coords of the selected nitrate \*\*\*\*

irest= group(1,igroup)

configw(irest,2)= configres(irest)

qw(iarest(irest))= -0.834

qw(iarest(irest)+1)= 0.417

qw(iarest(irest)+2)= 0.417

qw(iarest(irest)+3)= -0.834

qw(iarest(irest)+4)= 0.417

qw(iarest(irest)+5)= 0.417

iattypew(iarest(irest))= 22

iattypew(iarest(irest)+1)= 15

iattypew(iarest(irest)+2)= 15

iattypew(iarest(irest)+3)= 22

iattypew(iarest(irest)+4)= 15

iattypew(iarest(irest)+5)= 15

do i= iarest(irest)+6,jarest(irest)

qw(i)=0.0

iattypew(i)=0

end do

do i=1,2

do ipos=1,72

xw(iarest(irest)+3\*(i-1),ipos)=

$ xw(iarest(irest)+2\*(i-1)+1,1)

yw(iarest(irest)+3\*(i-1),ipos)=

$ yw(iarest(irest)+2\*(i-1)+1,1)

zw(iarest(irest)+3\*(i-1),ipos)=

$ zw(iarest(irest)+2\*(i-1)+1,1)

end do

end do

do i=1,2

do ipos=1,72

xw(iarest(irest)+3\*(i-1)+1,ipos)= water(1,ipos) +

$ xw(iarest(irest)+3\*(i-1),ipos)

yw(iarest(irest)+3\*(i-1)+1,ipos)= water(2,ipos) +

$ yw(iarest(irest)+3\*(i-1),ipos)

zw(iarest(irest)+3\*(i-1)+1,ipos)= water(3,ipos) +

$ zw(iarest(irest)+3\*(i-1),ipos)

xw(iarest(irest)+3\*(i-1)+2,ipos)= water(4,ipos) +

$ xw(iarest(irest)+3\*(i-1),ipos)

yw(iarest(irest)+3\*(i-1)+2,ipos)= water(5,ipos) +

$ yw(iarest(irest)+3\*(i-1),ipos)

zw(iarest(irest)+3\*(i-1)+2,ipos)= water(6,ipos) +

$ zw(iarest(irest)+3\*(i-1),ipos)

end do

end do

c do i=1,2

c do j=1,2

c do ipos=1,72

c dx = xw(iarest(irest)+3\*(i-1)+j,ipos)

c $ -xw(iarest(irest)+3\*(i-1),ipos)

c dy = yw(iarest(irest)+3\*(i-1)+j,ipos)

c $ -yw(iarest(irest)+3\*(i-1),ipos)

c dz = zw(iarest(irest)+3\*(i-1)+j,ipos)

c $ -zw(iarest(irest)+3\*(i-1),ipos)

c r2= dx\*\*2 + dy\*\*2 +dz\*\*2

c write (11,\*) 'R:', sqrt (r2),ipos

c end do

c end do

c end do

grouptype(igroup)='HOH'

iswater(igroup)=.true.

c \*\*\*\*\*\* Select water mols in intlist and add hb-bonded water mols \*\*\*\*

wlist=0

nlist=1

wlist(1)= irest

do iint = 1, nberint(irest)

jrest = intlist(iint,irest)

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ iswater(resgroup(jrest))) then

nlist=nlist+1

wlist(nlist)= jrest

end if

end do

ilist=1

1001 continue

irest=wlist(ilist)

do jrest=1,nrest

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ iswater(resgroup(jrest))) then

do i=1,nlist

if (wlist(i).eq.jrest) goto 1000

end do

if (hblist(irest,jrest)) then

nlist= nlist+1

wlist(nlist)=jrest

end if

end if

1000 continue

end do

ilist=ilist+1

if (ilist.le.nlist) goto 1001

C \*\*\*\*\*\*\*\*\*\*\* Water-Monte-Carlo \*\*\*\*\*\*\*\*\*\*\*\*

statwac=0.e0

statw=0.e0

kk=0

iw1= ceiling (rand(0)\*2)

do k=1,ktot

c \*\*\*\*\*\* TEMP \*\*\*\*\*\*

if (k .le. 2\*ktot/10) temp = 4000.e0+273.15e0

if (k .gt. 2\*ktot/10 .and. k.le.4\*ktot/10) temp=3900.e0+273.15e0

if (k .gt. 4\*ktot/10 .and. k.le.6\*ktot/10) temp=3800.e0+273.15e0

if (k .gt. 6\*ktot/10 .and. k.le.8\*ktot/10) temp=3500.e0+273.15e0

if (k .gt. 8\*ktot/10 .and. k.le.9\*ktot/10) temp=0.00001e0

if (k .gt. 9\*ktot/10) temp = 0.00001e0

temp=0.000001e0

c \*\*\*\*\* Statistics \*\*\*\*\*

statw=statw + 1.e0

ennew=0.e0

enold= 0.e0

accept=.false.

c \*\*\*\*\* Generate water and orientation \*\*\*\*\*

iw= 1

irest= wlist (ceiling (rand(0)\*nlist))

if (resgroup(irest).eq.igroup) iw= ceiling (rand(0)\*2)

igroupnew= ceiling (rand(0)\*72)

iaxis= ceiling (rand(0)\*3)

theta= (rand(0) - 0.5e0)\*3.14159265e0/6.0e0

ii=iarest(irest)

if (resgroup(irest).eq.igroup .and. iw.eq.2) then

ii=iarest(irest)+3

end if

if (k.le.720) then

kk= kk+1

igroupnew= kk

iw= iw1

if (kk.eq.72) then

kk=0

if (iw1.eq.1) then

iw1=2

else

iw1=1

end if

end if

end if

dx1= xw(ii+1,igroupnew)-xw(ii,igroupnew)

dy1= yw(ii+1,igroupnew)-yw(ii,igroupnew)

dz1= zw(ii+1,igroupnew)-zw(ii,igroupnew)

dx2= xw(ii+2,igroupnew)-xw(ii,igroupnew)

dy2= yw(ii+2,igroupnew)-yw(ii,igroupnew)

dz2= zw(ii+2,igroupnew)-zw(ii,igroupnew)

xnew(ii)= xw(ii,igroupnew)

ynew(ii)= yw(ii,igroupnew)

znew(ii)= zw(ii,igroupnew)

if (iaxis.eq.1) then

xnew(ii+1)= xw(ii+1,igroupnew)

ynew(ii+1)= dy1 \* cos (theta) - dz1 \* sin (theta)

$ + yw(ii,igroupnew)

znew(ii+1)= dy1 \* sin (theta) + dz1 \* cos (theta)

$ + zw(ii,igroupnew)

xnew(ii+2)= xw(ii+2,igroupnew)

ynew(ii+2)= dy2 \* cos (theta) - dz2 \* sin (theta)

$ + yw(ii,igroupnew)

znew(ii+2)= dy2 \* sin (theta) + dz2 \* cos (theta)

$ + zw(ii,igroupnew)

else if (iaxis.eq.2) then

xnew(ii+1)= dx1 \* cos (theta) + dz1 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+1)= yw(ii+1,igroupnew)

znew(ii+1)= -dx1 \* sin (theta) + dz1 \*cos (theta)

$ + zw(ii,igroupnew)

xnew(ii+2)= dx2 \* cos (theta) + dz2 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+2)= yw(ii+2,igroupnew)

znew(ii+2)= -dx2 \* sin (theta) + dz2 \*cos (theta)

$ + zw(ii,igroupnew)

else

xnew(ii+1)= dx1 \* cos (theta) - dy1 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+1)= dx1 \* sin (theta) + dy1 \* cos (theta)

$ + yw(ii,igroupnew)

znew(ii+1)= zw(ii+1,igroupnew)

xnew(ii+2)= dx2 \* cos (theta) - dy2 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+2)= dx2 \* sin (theta) + dy2 \* cos (theta)

$ + yw(ii,igroupnew)

znew(ii+2)= zw(ii+2,igroupnew)

end if

if (irest.eq.wlist(1)) then

if (iw.eq.1) then

ipos=configw(irest,2)

do ii=iarest(irest)+3,iarest(irest)+5

xnew(ii)=xw(ii,ipos)

ynew(ii)=yw(ii,ipos)

znew(ii)=zw(ii,ipos)

end do

else if (iw.eq.2) then

ipos=configw(irest,1)

do ii=iarest(irest),iarest(irest)+2

xnew(ii)=xw(ii,ipos)

ynew(ii)=yw(ii,ipos)

znew(ii)=zw(ii,ipos)

end do

end if

end if

c \*\*\* Calculating the new energies and test if accept or reject the step \*\*\*

c \*\*\*\* Water-water (EDO) interaction \*\*\*\*\*

etot=0.e0

if (resgroup(irest).eq.igroup) then

if (iw.eq.1) jpos= configw(irest,2)

if (iw.eq.2) jpos= configw(irest,1)

do i= 3\*(iw-1)+iarest(irest),3\*(iw-1)+iarest(irest)+2

do j= 3\*(2-iw)+iarest(irest),3\*(2-iw)+iarest(irest)+2

dx= xnew(i)-xw(j,jpos)

dy= ynew(i)-yw(j,jpos)

dz= znew(i)-zw(j,jpos)

r2= dx\*\*2+dy\*\*2+dz\*\*2

r= sqrt(r2)

sigma2=(sig2(iattypew(i),iattypew(j)))\*\*2

r2s= r2/sigma2

r6= r2s\*r2s\*r2s

r12= r6\*r6

elj= eps2(iattypew(i),iattypew(j))\*(1.e0/r12-2.e0/r6)

ecoul=627.5095\*0.5291772086\*qw(i)\*qw(j)/r

if (iattypew(i).eq.0 .or. iattypew(j).eq.0) then

elj=0.e0

ecoul=0.e0

end if

etot=etot+elj+ecoul

end do

end do

enew(irest,irest)= etot

ennew=ennew+enew(irest,irest)

enold=enold+ew(irest,irest)

if (etot.lt.-35.0) then

write (11,\*) etot, resnamet(irest),resnamet(irest)

end if

end if

ipos= igroupnew

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

jpos= configw(jrest,1)

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(irest,jrest)= 0.e0

goto 3073

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(irest,jrest)= 0.e0

goto 3073

end if

end if

iiarest=iarest(irest)

ijarest=jarest(irest)

jiarest=iarest(jrest)

jjarest=jarest(jrest)

if (resgroup(irest).eq.igroup) then

ijarest=iarest(irest)+5

end if

if (resgroup(jrest).eq.igroup) then

jjarest=iarest(jrest)+5

end if

ntest=0

etot=0.e0

do i= iiarest,ijarest

do j= jiarest,jjarest

ntest=ntest+1

if (jrest.eq.wlist(1)) then

if (j.gt.(iarest(jrest)+2)) then

jpos=configw(jrest,2)

else

jpos=configw(jrest,1)

end if

end if

dx= xnew(i)- xw(j,jpos)+ xoff(jrest,irest)

dy= ynew(i)- yw(j,jpos)+ yoff(jrest,irest)

dz= znew(i)- zw(j,jpos)+ zoff(jrest,irest)

r2= dx\*\*2+dy\*\*2+dz\*\*2

r= sqrt(r2)

sigma2=(sig2(iattypew(i),iattypew(j)))\*\*2

r2s= r2/sigma2

r6= r2s\*r2s\*r2s

r12= r6\*r6

if (iattypew(i).eq.22 .and. qw(i).ne.-0.834) then

write (11,\*) 'NOTE OXYGEN CHARGE',qw(i)

end if

if (iattypew(i).eq.15 .and. qw(i).ne.0.417) then

write (11,\*) 'NOTE HYDROGEN CHARGE',qw(i)

end if

elj= eps2(iattypew(i),iattypew(j))\*(1.e0/r12-2.e0/r6)

ecoul=627.5095\*0.5291772086\*qw(i)\*qw(j)/r

if (iattypew(i).eq.0 .or. iattypew(j).eq.0) then

elj=0.e0

ecoul=0.e0

end if

c if ((resnamet(irest)(1:7).eq.'EDO 301' .and.

c $ resnamet(jrest)(1:8).eq.'HOH 2060') .or.

c $ (resnamet(irest)(1:8).eq.'HOH 2060' .and.

c $ resnamet(jrest)(1:7).eq.'EDO 301')) then

c write (11,\*) iattypew(i),iattypew(j),elj, ecoul

c end if

etot=etot+elj+ecoul

rtest(ntest,1)=r

rtest(ntest,2)=ecoul

end do

end do

enew(irest,jrest)= etot

if (etot.lt.-35.0) then

write (11,\*) etot, resnamet(irest),resnamet(jrest)

do i=1,ntest

write (11,\*) rtest(i,1),rtest(i,2)

end do

do i=iiarest,ijarest

write (11,\*) iattypew(i),xnew(i)+xoff(jrest,irest),

$ ynew(i)+yoff(jrest,irest),znew(i)+zoff(jrest,irest)

end do

do i=jiarest,jjarest

write (11,\*) iattypew(i),xw(i,jpos),yw(i,jpos)

$ ,zw(i,jpos)

end do

end if

3073 continue

enold= enold + ew(irest,jrest)

ennew= ennew + enew(irest,jrest)

end do

c \*\*\*\* Test if accept or reject move \*\*\*\*\*

if (enold.ge.ennew) accept=.true.

if (enold.lt.ennew) then

if (rand(0).le. exp ((enold-ennew)/(Rgas\*temp))) then

accept=.true.

end if

end if

if (accept) then

if (irest.eq.wlist(1)) then

configw(irest,iw)= igroupnew

else

configw(irest,1) = igroupnew

end if

statwac=statwac+1.e0

do ipos=1,72

ii=iarest(irest)

if (resgroup(irest).eq.igroup .and. iw.eq.2) then

ii=ii+3

end if

do i=ii+1,ii+2

if (iaxis.eq.1) then

dyi= yw(i,ipos)-yw(ii,ipos)

dzi= zw(i,ipos)-zw(ii,ipos)

yw(i,ipos)= dyi \* cos (theta) - dzi \* sin (theta)+

$ yw(ii,ipos)

zw(i,ipos)= dyi \* sin (theta) + dzi \* cos (theta)+

$ zw(ii,ipos)

else if (iaxis.eq.2) then

dxi= xw(i,ipos)-xw(ii,ipos)

dzi= zw(i,ipos)-zw(ii,ipos)

xw(i,ipos)= dxi \* cos (theta) + dzi \* sin (theta)+

$ xw(ii,ipos)

zw(i,ipos)= - dxi \* sin (theta) + dzi\*cos(theta)+

$ zw(ii,ipos)

else

dxi= xw(i,ipos)-xw(ii,ipos)

dyi= yw(i,ipos)-yw(ii,ipos)

xw(i,ipos)= dxi \* cos (theta) - dyi \* sin (theta)+

$ xw(ii,ipos)

yw(i,ipos)= dxi \* sin (theta) + dyi \* cos (theta)+

$ yw(ii,ipos)

end if

end do

end do

if (irest.eq.wlist(1)) then

ew(irest,irest)=enew(irest,irest)

end if

do iint=1,nberint(irest)

jrest= intlist(iint,irest)

ew(irest,jrest)=enew(irest,jrest)

ew(jrest,irest)=ew(irest,jrest)

end do

end if

end do

c \*\*\*\* Calulate change in total energy \*\*\*\*

write (11,\*)

write (11,\*) '--------------------------------------------'

enold=0.e0

do irest=1,nrest

do jrest=1,nrest

if (irest.le.jrest) then

enold= enold + ew(irest,jrest) - e(irest,jrest)

if (ew(irest,jrest)-e(irest,jrest).lt.-4.0e0 .or.

$ ew(irest,jrest)-e(irest,jrest).gt.4.0e0) then

write (11,\*) resnamet(irest),resnamet(jrest),

$ ew(irest,jrest)-e(irest,jrest)

end if

end if

end do

end do

write (11,\*) igroup1,im,enold

write (12,\*) enold

write (11,'(a,f5.3)') 'Acceptance rate:', statwac/statw

ennew=0.e0

irest=wlist(1)

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

enold=enold - ew(irest,jrest)

ennew=ennew + ew(irest,jrest)

end do

write (11,\*) enold-ew(wlist(1),wlist(1)),

$ ennew+ew(wlist(1),wlist(1)),ew(wlist(1),wlist(1))

c \*\*\*\* Convert pseudo-water to edo (vacuum) \*\*\*\*\*

grouptype(igroup)='EDO'

iswater(igroup)=.false.

end if

end do

end if

end do

c \*\*\*\*\*\*\* Add water if NO3 is absent \*\*\*\*\*\*\*

write (11,\*) 'ADD WATER IF NO3 NOT PRESENT:'

c \*\*\*\*\*\* Find nitrates which is not present \*\*\*\*

do igroup1= 1 ,ngroup1

if (grouptype(igroup1).eq.'NO3') then

do im= 1,8

igroup= igroup1+(im-1)\*ngroup1

if (config(igroup).eq.4) then

c \*\*\*\*\*\*\*\* Copy energies from e to ew \*\*\*\*\*\*

do irest= 1, nrest

do jrest= 1, nrest

ew(irest,jrest) = e(irest,jrest)

end do

end do

c \*\*\*\*\*\* Copy charges, types and coords \*\*\*\*\*\*\*

do i=1,natom

qw(i) = q(i)

iattypew(i) = iattype(i)

do ipos= 1,72

xw(i,ipos) = xg(i,ipos)

yw(i,ipos) = yg(i,ipos)

zw(i,ipos) = zg(i,ipos)

end do

end do

c \*\*\*\*\* Copy configs for each residue, edo, ,nitrate, and water \*\*\*\*

do irest= 1,nrest

configw(irest,1)= configres(irest)

end do

c \*\*\*\* Change charges, types, coords of the selected nitrate \*\*\*\*

irest= group(1,igroup)

qw(iarest(irest))= -0.834

qw(iarest(irest)+1)= 0.417

qw(iarest(irest)+2)= 0.417

qw(iarest(irest)+3)= 0.0

iattypew(iarest(irest))= 22

iattypew(iarest(irest)+1)= 15

iattypew(iarest(irest)+2)= 15

iattypew(iarest(irest)+3)= 0

do ipos=1,72

xw(iarest(irest),ipos)= xw(iarest(irest),1)

yw(iarest(irest),ipos)= yw(iarest(irest),1)

zw(iarest(irest),ipos)= zw(iarest(irest),1)

xw(iarest(irest)+1,ipos)= water(1,ipos) +

$ xw(iarest(irest),ipos)

yw(iarest(irest)+1,ipos)= water(2,ipos) +

$ yw(iarest(irest),ipos)

zw(iarest(irest)+1,ipos)= water(3,ipos) +

$ zw(iarest(irest),ipos)

xw(iarest(irest)+2,ipos)= water(4,ipos) +

$ xw(iarest(irest),ipos)

yw(iarest(irest)+2,ipos)= water(5,ipos) +

$ yw(iarest(irest),ipos)

zw(iarest(irest)+2,ipos)= water(6,ipos) +

$ zw(iarest(irest),ipos)

end do

grouptype(igroup)='HOH'

iswater(igroup)=.true.

c \*\*\*\*\*\* Select water mols in intlist and add hb-bonded water mols \*\*\*\*

wlist=0

nlist=1

wlist(1)= irest

do iint = 1, nberint(irest)

jrest = intlist(iint,irest)

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ iswater(resgroup(jrest))) then

nlist=nlist+1

wlist(nlist)= jrest

end if

end do

ilist=1

1003 continue

irest=wlist(ilist)

do jrest=1,nrest

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ iswater(resgroup(jrest))) then

do i=1,nlist

if (wlist(i).eq.jrest) goto 1002

end do

if (hblist(irest,jrest)) then

nlist= nlist+1

wlist(nlist)=jrest

end if

end if

1002 continue

end do

ilist=ilist+1

if (ilist.le.nlist) goto 1003

C \*\*\*\*\*\*\*\*\*\*\* Water-Monte-Carlo \*\*\*\*\*\*\*\*\*\*\*\*

statwac=0.e0

statw=0.e0

do k=1,ktot

c \*\*\*\*\*\* TEMP \*\*\*\*\*\*

if (k .le. 2\*ktot/10) temp = 4000.e0+273.15e0

if (k .gt. 2\*ktot/10 .and. k.le.4\*ktot/10) temp=3900.e0+273.15e0

if (k .gt. 4\*ktot/10 .and. k.le.6\*ktot/10) temp=3800.e0+273.15e0

if (k .gt. 6\*ktot/10 .and. k.le.8\*ktot/10) temp=3500.e0+273.15e0

if (k .gt. 8\*ktot/10 .and. k.le.9\*ktot/10) temp=0.00001e0

if (k .gt. 9\*ktot/10) temp = 0.00001e0

temp=0.00001e0

c \*\*\*\*\* Statistics \*\*\*\*\*

statw=statw + 1.e0

ennew=0.e0

enold= 0.e0

accept=.false.

c \*\*\*\*\* Generate water and orientation \*\*\*\*\*

irest= wlist (ceiling (rand(0)\*nlist))

igroupnew= ceiling (rand(0)\*72)

iaxis= ceiling (rand(0)\*3)

theta= (rand(0) - 0.5e0)\*3.14159265e0/6.0e0

if (k.le.72) then

theta=0.e0

igroupnew=k

irest=wlist(1)

end if

ii=iarest(irest)

dx1= xw(ii+1,igroupnew)-xw(ii,igroupnew)

dy1= yw(ii+1,igroupnew)-yw(ii,igroupnew)

dz1= zw(ii+1,igroupnew)-zw(ii,igroupnew)

dx2= xw(ii+2,igroupnew)-xw(ii,igroupnew)

dy2= yw(ii+2,igroupnew)-yw(ii,igroupnew)

dz2= zw(ii+2,igroupnew)-zw(ii,igroupnew)

xnew(ii)= xw(ii,igroupnew)

ynew(ii)= yw(ii,igroupnew)

znew(ii)= zw(ii,igroupnew)

if (iaxis.eq.1) then

xnew(ii+1)= xw(ii+1,igroupnew)

ynew(ii+1)= dy1 \* cos (theta) - dz1 \* sin (theta)

$ + yw(ii,igroupnew)

znew(ii+1)= dy1 \* sin (theta) + dz1 \* cos (theta)

$ + zw(ii,igroupnew)

xnew(ii+2)= xw(ii+2,igroupnew)

ynew(ii+2)= dy2 \* cos (theta) - dz2 \* sin (theta)

$ + yw(ii,igroupnew)

znew(ii+2)= dy2 \* sin (theta) + dz2 \* cos (theta)

$ + zw(ii,igroupnew)

else if (iaxis.eq.2) then

xnew(ii+1)= dx1 \* cos (theta) + dz1 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+1)= yw(ii+1,igroupnew)

znew(ii+1)= -dx1 \* sin (theta) + dz1 \*cos (theta)

$ + zw(ii,igroupnew)

xnew(ii+2)= dx2 \* cos (theta) + dz2 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+2)= yw(ii+2,igroupnew)

znew(ii+2)= -dx2 \* sin (theta) + dz2 \*cos (theta)

$ + zw(ii,igroupnew)

else

xnew(ii+1)= dx1 \* cos (theta) - dy1 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+1)= dx1 \* sin (theta) + dy1 \* cos (theta)

$ + yw(ii,igroupnew)

znew(ii+1)= zw(ii+1,igroupnew)

xnew(ii+2)= dx2 \* cos (theta) - dy2 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+2)= dx2 \* sin (theta) + dy2 \* cos (theta)

$ + yw(ii,igroupnew)

znew(ii+2)= zw(ii+2,igroupnew)

end if

c \*\*\* Calculating the new energies and test if accept or reject the step \*\*\*

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

jpos= configw(jrest,1)

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(irest,jrest)= 0.e0

goto 3074

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(irest,jrest)= 0.e0

goto 3074

end if

end if

itest=0

etot=0.e0

iiarest=iarest(irest)

ijarest=jarest(irest)

jiarest=iarest(jrest)

jjarest=jarest(jrest)

do i= iiarest, ijarest

do j= jiarest,jjarest

jpos= configw(jrest,1)

dx= xnew(i)- xw(j,jpos)+xoff(jrest,irest)

dy= ynew(i)- yw(j,jpos)+yoff(jrest,irest)

dz= znew(i)- zw(j,jpos)+zoff(jrest,irest)

r2= dx\*\*2+dy\*\*2+dz\*\*2

rtest(itest+1,3)= yw(j,jpos)

r= sqrt (r2)

sigma2=(sig2(iattypew(i),iattypew(j)))\*\*2

r2s= r2/sigma2

r6= r2s\*r2s\*r2s

r12= r6\*r6

elj= eps2(iattypew(i),iattypew(j))\*(1.e0/r12-2.e0/r6)

rtest(itest+1,1)= eps2(iattypew(i),iattypew(j))

ecoul=627.5095\*0.5291772086\*qw(i)\*qw(j)/r

if (iattypew(i).eq.0 .or. iattypew(j).eq.0) then

elj=0.e0

ecoul=0.e0

end if

itest=itest+1

etot= etot+elj+ecoul

rtest(itest,2)= sig2(iattypew(i),iattypew(j))

end do

end do

enew(irest,jrest)= etot

if (grouptype(resgroup(irest)).eq.'HOH' .and.

$ grouptype(resgroup(jrest)).eq.'HOH' .and.

$ etot.lt.-6.8) then

write (6,\*) etot,xoff(irest,jrest),

$ yoff(irest,jrest),zoff(irest,jrest)

do iitest=1,itest

write (6,\*) (rtest(iitest,ik),ik=1,3)

end do

do ite= iarest(irest),jarest(irest)

write (6,\*) xnew(ite),ynew(ite),znew(ite)

$ ,qw(ite),iattypew(ite)

end do

do ite= iarest(jrest),jarest(jrest)

write (6,\*) xw(ite,jpos),yw(ite,jpos)

$ ,zw(ite,jpos),qw(ite),

$ iattypew(ite)

end do

end if

3074 continue

enold= enold + ew(irest,jrest)

ennew= ennew + enew(irest,jrest)

end do

c \*\*\*\* Test if accept or reject move \*\*\*\*\*

if (enold.ge.ennew) accept=.true.

if (enold.lt.ennew) then

if (rand(0).le. exp ((enold-ennew)/(Rgas\*temp))) then

accept=.true.

end if

end if

if (accept) then

configw(irest,1)= igroupnew

statwac=statwac+1.e0

do ipos=1,72

do i=iarest(irest)+1,iarest(irest)+2

if (iaxis.eq.1) then

dyi= yw(i,ipos)-yw(iarest(irest),ipos)

dzi= zw(i,ipos)-zw(iarest(irest),ipos)

yw(i,ipos)= dyi \* cos (theta) - dzi \* sin (theta)+

$ yw(iarest(irest),ipos)

zw(i,ipos)= dyi \* sin (theta) + dzi \* cos (theta)+

$ zw(iarest(irest),ipos)

else if (iaxis.eq.2) then

dxi= xw(i,ipos)-xw(iarest(irest),ipos)

dzi= zw(i,ipos)-zw(iarest(irest),ipos)

xw(i,ipos)= dxi \* cos (theta) + dzi \* sin (theta)+

$ xw(iarest(irest),ipos)

zw(i,ipos)= - dxi \* sin (theta) + dzi\*cos(theta)+

$ zw(iarest(irest),ipos)

else

dxi= xw(i,ipos)-xw(iarest(irest),ipos)

dyi= yw(i,ipos)-yw(iarest(irest),ipos)

xw(i,ipos)= dxi \* cos (theta) - dyi \* sin (theta)+

$ xw(iarest(irest),ipos)

yw(i,ipos)= dxi \* sin (theta) + dyi \* cos (theta)+

$ yw(iarest(irest),ipos)

end if

end do

end do

do iint=1,nberint(irest)

jrest= intlist(iint,irest)

ew(irest,jrest)=enew(irest,jrest)

ew(jrest,irest)=ew(irest,jrest)

end do

end if

end do

c \*\*\*\* Calulate change in total energy \*\*\*\*

write (11,\*)

write (11,\*) '--------------------------------------------'

enold=0.e0

do irest=1,nrest

do jrest= 1,nrest

if (irest.lt.jrest) then

enold= enold + ew(irest,jrest) - e(irest,jrest)

if (ew(irest,jrest)-e(irest,jrest).lt.-3.0e0 .or.

$ ew(irest,jrest)-e(irest,jrest).gt.3.0e0) then

write (11,\*) resnamet(irest),resnamet(jrest),

$ ew(irest,jrest)-e(irest,jrest)

end if

end if

end do

end do

write (11,\*) igroup1,im,enold

write (12,\*) enold

write (11,'(a,f5.3)') 'Acceptance rate:', statwac/statw

irest=wlist(1)

ennew=0.e0

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

enold=enold - ew(irest,jrest)

ennew=ennew + ew(irest,jrest)

end do

write (11,\*) enold,ennew

c \*\*\*\* Convert pseudo-water to nitrate (vacuum) \*\*\*\*\*

grouptype(igroup)='NO3'

iswater(igroup)=.false.

end if

end do

end if

end do

c \*\*\*\*\*\*\* Add water if water is absent \*\*\*\*\*\*\*

write (11,\*) 'ADD WATER IF WATER NOT PRESENT:'

c \*\*\*\*\*\* Find water which is not present \*\*\*\*

do igroup1= 1, ngroup1

if (grouptype(igroup1).eq.'HOH') then

do im= 1,8

igroup= igroup1+(im-1)\*ngroup1

if (.not. iswater(igroup)) then

c \*\*\*\*\*\*\*\* Copy energies from e to ew \*\*\*\*\*\*

do irest= 1, nrest

do jrest= 1, nrest

ew(irest,jrest) = e(irest,jrest)

end do

end do

c \*\*\*\*\*\* Copy charges, types and coords \*\*\*\*\*\*\*

do i=1,natom

qw(i) = q(i)

iattypew(i) = iattype(i)

do ipos= 1,72

xw(i,ipos) = xg(i,ipos)

yw(i,ipos) = yg(i,ipos)

zw(i,ipos) = zg(i,ipos)

end do

end do

c \*\*\*\*\* Copy configs for each residue, edo, ,nitrate, and water \*\*\*\*

do irest= 1,nrest

configw(irest,1)= configres(irest)

end do

iswater(igroup)=.true.

c \*\*\*\*\*\* Select water mols in intlist and add hb-bonded water mols \*\*\*\*

irest= group(1,igroup)

wlist=0

nlist=1

wlist(1)= irest

do iint = 1, nberint(irest)

jrest = intlist(iint,irest)

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ iswater(resgroup(jrest))) then

nlist=nlist+1

wlist(nlist)= jrest

end if

end do

ilist=1

1005 continue

irest=wlist(ilist)

do jrest=1,nrest

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ iswater(resgroup(jrest))) then

do i=1,nlist

if (wlist(i).eq.jrest) goto 1006

end do

if (hblist(irest,jrest)) then

nlist= nlist+1

wlist(nlist)=jrest

end if

end if

1006 continue

end do

ilist=ilist+1

if (ilist.le.nlist) goto 1005

C \*\*\*\*\*\*\*\*\*\*\* Water-Monte-Carlo \*\*\*\*\*\*\*\*\*\*\*\*

statwac=0.e0

statw=0.e0

do k=1,ktot

c \*\*\*\*\*\* TEMP \*\*\*\*\*\*

if (k .le. 2\*ktot/10) temp = 298.e0

if (k .gt. 2\*ktot/10 .and. k.le.4\*ktot/10) temp= 298.e0

if (k .gt. 4\*ktot/10 .and. k.le.6\*ktot/10) temp= 298.e0

if (k .gt. 6\*ktot/10 .and. k.le.8\*ktot/10) temp= 298.e0

if (k .gt. 8\*ktot/10 .and. k.le.9\*ktot/10) temp= 298.e0

if (k .gt. 9\*ktot/10) temp = 298.e0

temp= 0.000001e0

c \*\*\*\*\* Statistics \*\*\*\*\*

statw=statw + 1.e0

ennew=0.e0

enold= 0.e0

accept=.false.

c \*\*\*\*\* Generate water and orientation \*\*\*\*\*

irest= wlist (ceiling (rand(0)\*nlist))

igroupnew= ceiling (rand(0)\*72)

iaxis= ceiling (rand(0)\*3)

theta= (rand(0) - 0.5e0)\*3.14159265e0/6.0e0

if (k.le.72) then

theta=0.e0

igroupnew=k

irest=wlist(1)

end if

ii=iarest(irest)

dx1= xw(ii+1,igroupnew)-xw(ii,igroupnew)

dy1= yw(ii+1,igroupnew)-yw(ii,igroupnew)

dz1= zw(ii+1,igroupnew)-zw(ii,igroupnew)

dx2= xw(ii+2,igroupnew)-xw(ii,igroupnew)

dy2= yw(ii+2,igroupnew)-yw(ii,igroupnew)

dz2= zw(ii+2,igroupnew)-zw(ii,igroupnew)

xnew(ii)= xw(ii,igroupnew)

ynew(ii)= yw(ii,igroupnew)

znew(ii)= zw(ii,igroupnew)

if (iaxis.eq.1) then

xnew(ii+1)= xw(ii+1,igroupnew)

ynew(ii+1)= dy1 \* cos (theta) - dz1 \* sin (theta)

$ + yw(ii,igroupnew)

znew(ii+1)= dy1 \* sin (theta) + dz1 \* cos (theta)

$ + zw(ii,igroupnew)

xnew(ii+2)= xw(ii+2,igroupnew)

ynew(ii+2)= dy2 \* cos (theta) - dz2 \* sin (theta)

$ + yw(ii,igroupnew)

znew(ii+2)= dy2 \* sin (theta) + dz2 \* cos (theta)

$ + zw(ii,igroupnew)

else if (iaxis.eq.2) then

xnew(ii+1)= dx1 \* cos (theta) + dz1 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+1)= yw(ii+1,igroupnew)

znew(ii+1)= -dx1 \* sin (theta) + dz1 \*cos (theta)

$ + zw(ii,igroupnew)

xnew(ii+2)= dx2 \* cos (theta) + dz2 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+2)= yw(ii+2,igroupnew)

znew(ii+2)= -dx2 \* sin (theta) + dz2 \*cos (theta)

$ + zw(ii,igroupnew)

else

xnew(ii+1)= dx1 \* cos (theta) - dy1 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+1)= dx1 \* sin (theta) + dy1 \* cos (theta)

$ + yw(ii,igroupnew)

znew(ii+1)= zw(ii+1,igroupnew)

xnew(ii+2)= dx2 \* cos (theta) - dy2 \* sin (theta)

$ + xw(ii,igroupnew)

ynew(ii+2)= dx2 \* sin (theta) + dy2 \* cos (theta)

$ + yw(ii,igroupnew)

znew(ii+2)= zw(ii+2,igroupnew)

end if

c \*\*\* Calculating the new energies and test if accept or reject the step \*\*\*

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

jpos= configw(jrest,1)

c \*\*\*\* Test if HOH, NO3 or EDO is vacuum \*\*\*\*\*

if (resgroup(jrest).ne.-1) then

if (grouptype(resgroup(jrest)).eq.'HOH' .and.

$ .not. iswater(resgroup(jrest))) then

enew(irest,jrest)= 0.e0

goto 3075

end if

if (grouptype(resgroup(jrest)).eq.'EDO' .or.

$ grouptype(resgroup(jrest)).eq.'NO3') then

if (jpos.eq.4) then

enew(irest,jrest)= 0.e0

goto 3075

end if

end if

end if

etot=0.e0

do i= iarest(irest),jarest(irest)

do j= iarest(jrest),jarest(jrest)

dx= xnew(i)-xw(j,jpos)+xoff(jrest,irest)

dy= ynew(i)-yw(j,jpos)+yoff(jrest,irest)

dz= znew(i)-zw(j,jpos)+zoff(jrest,irest)

r2= dx\*\*2+dy\*\*2+dz\*\*2

r= sqrt(r2)

sigma2=(sig2(iattypew(i),iattypew(j)))\*\*2

r2s= r2/sigma2

r6= r2s\*r2s\*r2s

r12= r6\*r6

elj= eps2(iattypew(i),iattypew(j))\*(1.e0/r12-2.e0/r6)

ecoul=627.5095\*0.5291772086\*qw(i)\*qw(j)/r

if (iattypew(i).ne.22 .and. iattypew(i).ne.15) then

write (6,\*) iattypew(i)

end if

if (iattypew(i).eq.0 .or. iattypew(j).eq.0) then

elj=0.e0

ecoul=0.e0

end if

etot=etot+elj+ecoul

end do

end do

enew(irest,jrest)= etot

3075 continue

enold= enold + ew(irest,jrest)

ennew= ennew + enew(irest,jrest)

end do

c \*\*\*\* Test if accept or reject move \*\*\*\*\*

if (enold.ge.ennew) accept=.true.

if (enold.lt.ennew) then

if (rand(0).le. exp ((enold-ennew)/(Rgas\*temp))) then

accept=.true.

end if

end if

if (accept) then

configw(irest,1)= igroupnew

statwac=statwac+1.e0

do ipos=1,72

do i=iarest(irest)+1,iarest(irest)+2

if (iaxis.eq.1) then

dyi= yw(i,ipos)-yw(iarest(irest),ipos)

dzi= zw(i,ipos)-zw(iarest(irest),ipos)

yw(i,ipos)= dyi \* cos (theta) - dzi \* sin (theta)+

$ yw(iarest(irest),ipos)

zw(i,ipos)= dyi \* sin (theta) + dzi \* cos (theta)+

$ zw(iarest(irest),ipos)

else if (iaxis.eq.2) then

dxi= xw(i,ipos)-xw(iarest(irest),ipos)

dzi= zw(i,ipos)-zw(iarest(irest),ipos)

xw(i,ipos)= dxi \* cos (theta) + dzi \* sin (theta)+

$ xw(iarest(irest),ipos)

zw(i,ipos)= - dxi \* sin (theta) + dzi\*cos(theta)+

$ zw(iarest(irest),ipos)

else

dxi= xw(i,ipos)-xw(iarest(irest),ipos)

dyi= yw(i,ipos)-yw(iarest(irest),ipos)

xw(i,ipos)= dxi \* cos (theta) - dyi \* sin (theta)+

$ xw(iarest(irest),ipos)

yw(i,ipos)= dxi \* sin (theta) + dyi \* cos (theta)+

$ yw(iarest(irest),ipos)

end if

end do

end do

do iint=1,nberint(irest)

jrest= intlist(iint,irest)

ew(irest,jrest)=enew(irest,jrest)

ew(jrest,irest)=ew(irest,jrest)

end do

end if

end do

c \*\*\*\* Calulate change in total energy \*\*\*\*

write (11,\*)

write (11,\*) '--------------------------------------------'

enold=0.e0

do irest=1,nrest

do jrest= 1,nrest

if (irest.lt.jrest) then

enold= enold + ew(irest,jrest) - e(irest,jrest)

if (ew(irest,jrest)-e(irest,jrest).lt.-3.0e0 .or.

$ ew(irest,jrest)-e(irest,jrest).gt.3.0e0) then

write (11,\*) resnamet(irest),resnamet(jrest),

$ ew(irest,jrest)-e(irest,jrest),e(irest,jrest)

end if

end if

end do

end do

write (11,\*) igroup1,im,enold

write (12,\*) enold

write (11,'(a,f5.3)') 'Acceptance rate:', statwac/statw

irest=wlist(1)

ennew=0.e0

do iint= 1,nberint(irest)

jrest= intlist(iint,irest)

enold=enold - ew(irest,jrest)

ennew=ennew + ew(irest,jrest)

end do

write (11,\*) enold,ennew

iswater(igroup)=.false.

end if

c \*\*\*\* Convert pseudo-water to water (vacuum) \*\*\*\*\*

end do

end if

end do

return

end