

# roadmap for gromacs/gaussian interface code

## modified links (/home/gromacs/devel/g03)

- I301

the self-energy of the point-charges is ignored

- I701

gradients on point charges are computed

- I510

CI vectors of  $S_i$  and  $S_{i+1}$  are written on read-write file

- I9999

dumps the following information onto the punch file (fort.7)

1. total QM energy
2. energy difference between  $S_i$  and  $S_{i+1}$
3. gradients of QM atoms
4. gradients of MM atoms
5. CI vectors of  $S_i$  and  $S_{i+1}$

# roadmap for gromacs/gaussian interface code

modified subroutines in I301

- solncr

modified

$$\text{SNR} = \text{SNR1} + \text{SNR2} + \text{SNR3}$$

into

$$\text{SNR} = \text{SNR2}$$

now only the QM-MM interaction is computed

# roadmap for gromacs/gaussian interface code

## modified subroutines in I510

- sref (and ldrive)

added to the declaration section

```
integer states
```

added at the end

```
states = 2  
write (iout,*)'nsec before writing the CI vecs is ', nsec  
call conddf(666,nsec*states+states)  
do i=1,states  
    call fileio(1,666,nsec,c(1,i),0)  
enddo
```

```
C  
C to get also the final eigenvalues punched.  
C
```

```
call fileio(1,666,states,E,0)
```

this stores both CI vectors and energies onto the read-write file

# roadmap for gromacs/gaussian interface code

## modified subroutines in I701

- d1e

added to the declaration section a new bucket for the forces on the point charges

```
parameter (.., IRwFCh=888)
```

added to the background charges recovery section (approx. line 500)

```
If(LRwSol.gt.0.and..not.SCIPCM) then  
< .. snip .. >  
    ifch = IV  
    IV = IV + 3*NumChg*NC  
    Call AClear(NumChg*3*NC,V(Ifch))  
    CALL ConDDF(IRwFCh,NumChg*3*NC)  
else  
    NumChg = 0  
    Ifch=IV  
endif  
MDV1 = NGot - IV + 1
```

continued on next page

# roadmap for gromacs/gaussian interface code

## modified subroutines in I701

- d1e

added an extra argument to all Oneeli() function calls

```
Oneeli(...,V(Ifch))
```

added an extra argument to all frcNCN() function calls

```
frcNCN(...,V(Ifch))
```

added a call to fileio to write the bucket V(Ifch) to the read-write file (approx. line 630)

```
Call FileIO(1,-IRwFCh,NumChg*3*NC,V(Ifch),0)
```

# roadmap for gromacs/gaussian interface code

## modified subroutines in I701

- oneeli

added an extra argument to Oneeli() for the forces on the point charges

```
subroutine Oneeli(...,FCh)
```

added to the declaration section

```
Real*8 FCh(*)
```

added an extra argument to all PrsmSu() function calls

```
PrsmSu(...,FCh)
```

# roadmap for gromacs/gaussian interface code

## modified subroutines in I701

- prmsu

added an extra argument to PrsmSu() for the forces on the point charges

```
subroutine PrsmSu(...,FCh)
```

added to the declaration section

```
Real*8 FCh(*)
```

added an extra argument to all Pirsm() function calls

```
Pirsm(...,FCh)
```

# roadmap for gromacs/gaussian interface code

## modified subroutines in I701

- prism

added an extra argument to Prism() for the forces on the point charges

```
subroutine Prism(...,FCh)
```

added to the declaration section

```
Real*8 FCh(*)
```

added an extra argument to all PrmDig() function calls

```
PrmDig(...,FCh)
```

# roadmap for gromacs/gaussian interface code

## modified subroutines in I701

- prmdig

added an extra argument to PrmDig() for the forces on the point charges

```
subroutine PrmDig(...,FCh)
```

added to the declaration section

```
Real*8 FCh(*)
```

added an extra argument to all Cnt061() function calls

```
Cnt061(...,FCh)
```

# roadmap for gromacs/gaussian interface code

## modified subroutines in I701

- cnt061

added extra argument to cnt061 for the forces on the point charges

```
subroutine cnt061(.., FCh)
```

added to the declaration section

```
Real*8 FCh(3,nGrid,NMatD)
```

added code to store the forces on the point charges

```
FCh(1,(iStart+i-1),IM) = FCh(1,(iStart+i-1),IM)
$                       + FxA + FxB
FCh(2,(iStart+i-1),IM) = FCh(2,(iStart+i-1),IM)
$                       + FyA + FyB
FCh(3,(iStart+i-1),IM) = FCh(3,(iStart+i-1),IM)
$                       + FzA + FzB
```

# roadmap for gromacs/gaussian interface code

## modified subroutines in I701

- frcncn

added extra argument to frcNCN for the forces on the point charges

```
subroutine frcNCN(.., FCh)
```

added to the declaration section

```
Real*8 FCh(3, NumChg, NC)
```

added code to store the forces on the point charges

```
FCh(1, IChg, IC) = FCh(1, IChg, IC) + DX*R3C  
FCh(2, IChg, IC) = FCh(2, IChg, IC) + DY*R3C  
FCh(3, IChg, IC) = FCh(3, IChg, IC) + DZ*R3C
```

# roadmap for gromacs/gaussian interface code

## modified subroutines in I9999

- alldun

added to the declaration section two new buckets

```
parameter (..,IRwCiv=666, IRwFCh=888)
```

added code to retrieve the CI vectors from the read-write file

```
Icivec = IFX + LenFX  
Ncivec = Max(itqry(666),0)
```

added code to retrieve the gradients from the read-write file

```
IFCh = Icivec + Ncivec  
lenfch = Max(itqry(888),0)
```

added code to read the CI vectors and forces into the memory array (V)

```
if(Ncivec.gt.0) Call FILEIO(2,-666,Ncivec,V(Icivec),0)  
if(LenFCh.gt.0) Call FILEIO(2,-888,LenFCh,V(IFCh),0)
```

added four extra argumenst to all PunOut() function calls

```
PunOut(...,V(indgen),V(IFCh),LenFCh,V(Icivec),Ncivec)
```

# roadmap for gromacs/gaussian interface code

modified subroutines in 19999

- punout

added four extra arguments to PunOut()

```
subroutine PunOUT(..., Gen,FCh,LenFCh,CIvec,Ncivec)
```

added to the declaration section

```
Real*8 Gen(*),FCh(*),CIvec(*)
```

modified all decimal symbols in format statements from D to E, such as

```
Format(1X,I3,3D20.10)  →  Format(1X,I3,3E20.10)
```

removed the line

```
Call StrOut(IPunch,ITitle,80,2)
```

continued on next page

# roadmap for gromacs/gaussian interface code

## modified subroutines in I9999

- punout

added code to punch the total energy, energy difference, gradients and CI vectors

```
if(Ncivec.gt.0) then
  E1 = CVec(Ncivec)
  E2 = CVec(Ncivec-1)
  DE = ABS(E1-E2)
Write(IPunch,1042) Gen(32),DE
else
  Write(IPunch,*) Gen(32)
endif
If(LenFCh.gt.0) then
  Write(IPunch,1041) (FCh(I),I=1,LenFCh)
endif
write(IPunch,*) (Ncivec-2)/2
do i=1,Ncivec-2
  write(IPunch,1041) CVec(i)
enddo
If(LenFFX .gt. 0) then .....
```

Gromacs reads in the punch file (fort.7) to retrieve the energy, energy difference, gradients on both QM and MM atoms, and the CI vectors. With the latter and the energy difference gromacs can do a diabatic hop

# roadmap for gromacs/gaussian interface code

modified subroutines in 19999

- archiv

removed the lines

```
If(PunArc) then  
  Write(IOut,1070)  
  Call StrOut(IPunch,BB,NBB,2)  
endIf
```

# roadmap for gromacs/gaussian interface code

## configuring and compiling gromacs

- ideal case

configure creates the Makefiles

```
./configure --with-qmmm-gaussian
```

after configuring, the code can be build

```
make
```

- on IBM power5 machine

configure needs to be invoked as follows

```
./configure --without-fftw --disable-largefile  
--disable-vectorized-sqrt --disable-vector  
--disable-cpu-optimization CC=gcc
```

configuring only needed in case of new hardware, OS, or additional source files

# roadmap for gromacs/gaussian interface code

## diabatic surface hop in gromacs

- gaussian.c (./gmx/src/mdlib/)

simple general *diabatic* surface hopping algorithm

```
int diabatic_hop(t_CIVector current1, t_CIVector previous1,
                t_CIVector current2, t_CIVector previous2, int step)
{
    int
        hop = 0;
    real
        d11=0.0, d12=0.0, d21=0.0, d22=0.0;

    if (step){
        d11 = inproduct(current1.elem,previous1.elem,current1.dim);
        d12 = inproduct(current1.elem,previous2.elem,current1.dim);
        d21 = inproduct(current2.elem,previous1.elem,current1.dim);
        d22 = inproduct(current2.elem,previous2.elem,current1.dim);
    }
    if((fabs(d12)>0.5)&&(fabs(d21)>0.5))
        hop = 1;
    return(hop);
}
```