roadmap for gromacs/gaussian interface code

modified links (/gromacs_devel/g01)

- l301

  the self-energy of the point-charges is ignored

- l701

  gradients on point charges are computed

- l510

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

- l9999

  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 l301

- solncr

modified

SNR = SNR1 + SNR2 + SNR3

into

SNR = SNR2

now only the QM-MM interaction is computed

roadmap for gromacs/gaussian interface code

modified subroutines in l510

- 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 l701

- 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)
  Call FileIO(1,-IRwFCh,NumChg*3*NC,V(Ifch),0)
 else
  NumChg = 0
  Ifch=IV
 endIf
 MDV1 = NGot - IV + 1
```

continued on next page

roadmap for gromacs/gaussian interface code

modified subroutines in l701

- d1e

  added an extra argument to all Oneeli() function calls

  ```
  Oneeli(...,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 l701

- 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 l701

- prsmsu

  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

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

  when using OPENMP, add the following statement

  ```
  ThrOK=.FALSE.
  ```

  just before

  ```
  If(ThrOK) then
  ```

  to avoid parallellization of the MM gradients.

roadmap for gromacs/gaussian interface code

modified subroutines in l701

- 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 l701

- 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 l701

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

roadmap for gromacs/gaussian interface code

modified subroutines in l9999

- 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(IFCh),LenFCh,V(Icivec),Ncivec)
```

roadmap for gromacs/gaussian interface code

modified subroutines in l9999

- punout

  added four extra arguments to PunOut()

  subroutine PunOUt(..., 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 l9999

- punout

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

```
if(Ncivec.gt.0) then
   E1 = CIvec(Ncivec)
   E2 = CIvec(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) CIvec(i)
enddo
```

  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 l9999

- archiv

  removed the lines

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

roadmap for gromacs/gaussian interface code

invoking gaussian from gromacs

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

  creating input.com for normal QM/MM computations (standard route)

  void write_gaussian_input(...)

  creating input.com for SH QM/MM computations (non-standard route)

  void write_gaussian_SH_input(...)

  doing the actual call (via system())

  void do_gaussian(...)

  reading fort.7 in normal QM/MM computations (standard route)

  void read_gaussian_input(...)

  reading fort.7 in SH QM/MM computations (non-standard route)

  void read_gaussian_SH_input(...)

roadmap for gromacs/gaussian interface code

diabatic surface hop in gromacs

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

  creating input.com for normal QM/MM computations (standard route)

```
int hop(int step, t_QMrec *qm)
{
  int
    swap = 0;
  real
    d11=0.0,d12=0.0,d21=0.0,d22=0.0;

  /* calculates the inproduct between the current Ci vector and the
   * previous CI vector. A diabatic hop will be made if d12 and d21
   * are much bigger than d11 and d22. In that case hop returns true,
   * otherwise it returns false.
   */
  if(step){ /* only go on if more than one step has been done */
    d11 = inproduct(qm->CIvec1,qm->CIvec1old,qm->CIdim);
    d12 = inproduct(qm->CIvec1,qm->CIvec2old,qm->CIdim);
    d21 = inproduct(qm->CIvec2,qm->CIvec1old,qm->CIdim);
    d22 = inproduct(qm->CIvec2,qm->CIvec2old,qm->CIdim);
  }
 if((fabs(d12)>0.5)&&(fabs(d21)>0.5))
    swap = 1;

  return(swap);
}
```