```
      subroutine gaussfg_init(natom,nbonds,ib,jb,nquant,labels,nlink)

!     callするとき:
!       gaussfg_init(natom,mbona,ix(i18),ix(i20),nquant,labels,nlink)

!     natom           : 全原子数 ( リンク原子含まず )
!     nbonds          : 水素を含まないbondの数
!     ib(*)           : (bondする原子iのインデックス-1)x3
!     jb(*)           : (bondする原子jのインデックス-1)x3
!     nquant          : QM原子数 ( リンク原子含まず )
!     labels(*)       : QM原子のインデックスリスト
!     nlink           : リンク原子の数
!
!     nqm_gfg         - QM原子数
!     nln_gfg         - リンク原子数
!     nqmln_gfg       - QM原子数+リンク原子数
!     iqmln_gfg(*)    - リンクしているQM原子インデックス
!     immln_gfg(*)    - リンクしているMM原子インデックス
!
!     gh_route_gfg    - Gaussian route section
!     gh_molsp_gfg    - Gaussian molecule specification section

      use iflport ! Intel Fortran 7.0
!      use ifport ! Intel Fortran 9.0

      implicit none
      integer natom,nbonds,ib(*),jb(*),nquant,labels(*),nlink

      integer natom_gfg, nqm_gfg, nln_gfg, nqmln_gfg,
     &        iqmln_gfg(1000), immln_gfg(1000)
      character gh_route_gfg*400, gh_molsp_gfg*80

      common /gfg/ natom_gfg, nqm_gfg, nln_gfg, nqmln_gfg,
     &             iqmln_gfg, immln_gfg,
     &             gh_route_gfg, gh_molsp_gfg

      integer i, j, ii, jj, mm
      integer io1, ierr
      parameter (io1=70)

      ! gaussian.headerがないときはリターン
      inquire(file='gaussian.header', exist=ierr)
      if(.not. ierr) return

      ! 定数
      natom_gfg = natom
      nqm_gfg = nquant
      nln_gfg = nlink
      nqmln_gfg = nqm_gfg+nln_gfg

      ! Gaussianヘッダーの読み込み
      open(io1,file='gaussian.header',status='old',action='read')
      read(io1,'(a)') gh_route_gfg ! forceやchargeが入っていてはいけない
      read(io1,'(a)') gh_molsp_gfg
      close(io1)

      ierr = runqq('rm', '-f gaussian-debug.log') ! log

      ! リンク原子に結合しているQM原子,MM原子の取得
      mm = 0
      do i=1, nbonds
        ii = 0
        jj = 0
        do j=1, nquant
          if((ib(i)/3+1)==labels(j)) ii=1
```

```
66            if((jb(i)/3+1)==labels(j)) jj=1
67          enddo
68          if((ii+jj)==1) then
69            if(ii==1) then
70              iqmln_gfg(mm+1) = ib(i)/3+1
71              immln_gfg(mm+1) = jb(i)/3+1
72            else
73              iqmln_gfg(mm+1) = jb(i)/3+1
74              immln_gfg(mm+1) = ib(i)/3+1
75            endif
76            mm = mm + 1
77          endif
78        enddo
79
80        end
81
82  ! ----------------------------------------------------------------------
83
84        SUBROUTINE gaussfg(XPARAM,INT,ESCF,FULSCF,GRAD,LGRAD,
85       .                   coord,nquant,crdsmm,chrgs,iread,dxyzqm,
86       .                   dxyzcl,icalcf,iqmres,mark,jqatms)
87  c
88  c     Routine has been modified for qm-mm coupling and includes COSMO.
89  c
90  c     Variables for qm-mm:
91  c
92  c     coord(3,numatm) - Cartesian coordinates of qm atoms.  This was
93  c                       originally a MOPAC array defined locally in
94  c                       COMPFG and filled by GMETRY.  Now Cartesian
95  c                       coordinates are passed directly.
96  c     nquant          - Number of qm atoms.
97  c     crdsmm(*)       - Main AMBER array containing coordinates for
98  c                       all atoms (used here for mm atoms).
99  c     chrgs(*)        - Atomic charges for mm atoms.
100 c     iread           - Flag for reading in density matrix 0=no, 1=yes.
101 c     dxyzqm(3,*)     - Quantum mechanical derivatives from qm-mm
102 c                       interactions.
103 c     dxyzcl(3,*)     - Classical derivatives from qm-mm interactions.
104 c     icalcf          - Flag for calculating atomic forces 0=yes, 1=no.
105 c     iqmres(*)       - Atom-based interaction list.
106 c     mark(*)         - Center of geometry atom marker list.
107 c     jqatms(*)       - Atom numbers for qm atoms (based on AMBER scheme).
108 c
109 c
110 c                -- S. Dixon (8/11/94)
111 c
112       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
113 *COMDECK SIZES
114 ************************************************************************
115 *   THIS FILE CONTAINS ALL THE ARRAY SIZES FOR USE IN MOPAC.
116 *
117 *     THERE ARE ONLY 5 PARAMETERS THAT THE PROGRAMMER NEED SET:
118 *     MAXHEV = MAXIMUM NUMBER OF HEAVY ATOMS (HEAVY: NON-HYDROGEN ATOMS)
119 *     MAXLIT = MAXIMUM NUMBER OF HYDROGEN ATOMS.
120 *     MAXTIM = DEFAULT TIME FOR A JOB. (SECONDS)
121 *     MAXDMP = DEFAULT TIME FOR AUTOMATIC RESTART FILE GENERATION (SECS)
122 *     ISYBYL = 1 IF MOPAC IS TO BE USED IN THE SYBYL PACKAGE, =0 OTHERWISE
123 *     SEE ALSO NMECI, NPULAY AND MESP AT THE END OF THIS FILE
124 *
125       PARAMETER (MAXHEV=40, MAXLIT=40)
126       PARAMETER (MAXTIM=3600, MAXDMP=3600)
127       PARAMETER (ISYBYL=1)
128 *
129 ************************************************************************
130 *
```

```
131   *   THE FOLLOWING CODE DOES NOT NEED TO BE ALTERED BY THE PROGRAMMER
132   *
133   ************************************************************************
134   *
135   *    ALL OTHER PARAMETERS ARE DERIVED FUNCTIONS OF THESE TWO PARAMETERS
136   *
137   *      NAME                    DEFINITION
138   *     NUMATM          MAXIMUM NUMBER OF ATOMS ALLOWED.
139   *     MAXORB          MAXIMUM NUMBER OF ORBITALS ALLOWED.
140   *     MAXPAR          MAXIMUM NUMBER OF PARAMETERS FOR OPTIMISATION.
141   *     N2ELEC          MAXIMUM NUMBER OF TWO ELECTRON INTEGRALS ALLOWED.
142   *     MPACK           AREA OF LOWER HALF TRIANGLE OF DENSITY MATRIX.
143   *     MORB2           SQUARE OF THE MAXIMUM NUMBER OF ORBITALS ALLOWED.
144   *     MAXHES          AREA OF HESSIAN MATRIX
145   *     MAXALL          LARGER THAN MAXORB OR MAXPAR.
146   ************************************************************************
147         PARAMETER (VERSON=7.00D0)
148         PARAMETER (NUMATM=MAXHEV+MAXLIT)
149         PARAMETER (MAXORB=4*MAXHEV+MAXLIT)
150         PARAMETER (MAXPAR=3*NUMATM)
151         PARAMETER (MAXBIG=MAXORB*MAXORB*2)
152         PARAMETER (N2ELEC=(50*MAXHEV*(MAXHEV-1)+10*MAXHEV*MAXLIT
153        +                     +(MAXLIT*(MAXLIT-1))/2))
154         PARAMETER (MAXHES=(MAXPAR*(MAXPAR+1))/2,MORB2=MAXORB**2)
155         PARAMETER (MPACK=(MAXORB*(MAXORB+1))/2)
156         PARAMETER (MAXPR=6*MAXORB)
157         PARAMETER (MAXALL=4*MAXHEV+3*MAXLIT)
158          PARAMETER (NMECI=11,  NPULAY=MPACK, MMCI=60)
159         PARAMETER (MESP=10)
160         PARAMETER (LENABC=600)
161         PARAMETER (LENAB2=LENABC*(LENABC+5))
162         PARAMETER (NPPA = 1082, MAXNSS = 500)
163         PARAMETER (MAXDEN=10*MAXHEV+MAXLIT)
164   ************************************************************************
165   *DECK MOPAC
166         dimension crdsmm(*),chrgs(*),dxyzqm(3,*),dxyzcl(3,*),iqmres(*),
167        .          mark(*),jqatms(*)
168         DIMENSION XPARAM(MAXPAR),GRAD(MAXPAR)
169         LOGICAL LGRAD, FULSCF
170         COMMON/KILLBTL/KILLIT
171         COMMON /GEOVAR/ NVAR,LOC(2,MAXPAR),IDUMY,DUMY(MAXPAR)
172         COMMON /GEOSYM/ NDEP,LOCPAR(MAXPAR),IDEPFN(MAXPAR),LOCDEP(MAXPAR)
173         COMMON /GEOM  / GEO(3,NUMATM)
174         COMMON /ATHEAT/ ATHEAT
175         COMMON /WMATRX/ WJ(N2ELEC), WK(N2ELEC)
176         COMMON /ENUCLR/ ENUCLR
177         COMMON /NATYPE/ NZTYPE(107),MTYPE(30),LTYPE
178         COMMON /ELECT / ELECT
179         PARAMETER (MDUMY=MAXPAR**2-MPACK)
180         COMMON /SCRACH/ RXYZ(MPACK), XDUMY(MDUMY)
181         COMMON /HMATRX/ H(MPACK)
182         COMMON /GEOKST/ NATOMS,LABELS(NUMATM),
183        1                NA(NUMATM), NB(NUMATM), NC(NUMATM)
184         COMMON /ERRFN  / ERRFN(MAXPAR), AICORR(MAXPAR)
185         COMMON /VECTOR/ C(MORB2),EIGS(MAXORB),CBETA(MORB2),EIGB(MAXORB)
186         COMMON /LAST  / LAST
187         COMMON /NUMCAL/ NUMCAL
188         COMMON /SCFTYP/ EMIN, LIMSCF
189         COMMON /MOLMEC/ HTYPE(4),NHCO(4,20),NNHCO,ITYPE
190        1       /MOLKST/ NUMAT,NAT(NUMATM),NFIRST(NUMATM),NMIDLE(NUMATM),
191        2                NLAST(NUMATM), NORBS, NELECS,NALPHA,NBETA,
192        3                NCLOSE,NOPEN,NDUMY,FRACT
193   C COSMO change
194         LOGICAL ISEPS, USEPS , UPDA, CFIRST
195         COMMON /ISEPS/  ISEPS, USEPS, UPDA
```

```
196         DATA CFIRST /.TRUE./
197   C end of COSMO change
198   C***********************************************************************
199   C
200   C   COMPFG CALCULATES (A) THE HEAT OF FORMATION OF THE SYSTEM, AND
201   C                       (B) THE GRADIENTS, IF LGRAD IS .TRUE.
202   C
203   C   ON INPUT  XPARAM = ARRAY OF PARAMETERS TO BE USED IN INTERNAL COORDS
204   C             LGRAD  = .TRUE. IF GRADIENTS ARE NEEDED, .FALSE. OTHERWISE
205   C             INT    = .TRUE. IF HEAT OF FORMATION IS TO BE CALCULATED
206   C             FULSCF = .TRUE. IF FULL SCF TO BE DONE, .FALSE. OTHERWISE.
207   C
208   C   ON OUTPUT ESCF  = HEAT OF FORMATION.
209   C             GRAD   = ARRAY OF GRADIENTS, IF LGRAD = .TRUE.
210   C
211   C***********************************************************************
212         COMMON /KEYWRD/KEYWRD
213         COMMON/ANALZZ/ICALX1,ICALX2,ICALX3,ICALCN
214         CHARACTER*241 KEYWRD
215         LOGICAL DEBUG, INT, PRINT, ANALYT, LARGE, USEDCI,
216        1FORCE, TIMES, AIDER
217         DIMENSION COORD(3,NUMATM), W(N2ELEC), DEGREE(3), XPAREF(MAXPAR)
218        1,DELTAP(NMECI**2) ,DELTA(NMECI*MAXORB)
219         SAVE DEGREE, PRINT, DEBUG, aider,analyt,large,usedci,
220        1 force,times
221         EQUIVALENCE (W,WJ)
222   C     DATA ICALCN /0/
223         logical first_call
224         data first_call /.true./
225         save first_call
226
227         !wpp
228         !上の部分について不要な部分多数あり
229         ! メモ：
230         ! * gaussgfのnquantにはlink原子の数がすでに含まれている
231         ! * jqatms(i)は 原子iがQMのとき、その原子番号
232         !                    Linkのとき、-1
233         !                      MMのとき、0  を返す
234         !
235         ! 1 hartree = 627.51d0 kcal/mol
236         ! 1 bohr = 0.52917706d0 angstrom
237
238         integer io1, ierr
239         parameter (io1=70)
240
241         double precision qscale, escale, fscale
242         data qscale /18.2223d0/
243         parameter (escale=627.51d0, fscale=escale/0.52917706d0)
244
245         integer natom_gfg, nqm_gfg, nln_gfg, nqmln_gfg,
246        &        iqmln_gfg(1000), immln_gfg(1000)
247         character gh_route_gfg*400, gh_molsp_gfg*80
248
249         common /gfg/ natom_gfg, nqm_gfg, nln_gfg, nqmln_gfg,
250        &             iqmln_gfg, immln_gfg,
251        &             gh_route_gfg, gh_molsp_gfg
252
253         integer i, j, indxmm, ni_mm, jmm, jcoord
254         logical fcharg, mmflgs(natom_gfg)
255         double precision rx, ry, rz, gx, gy, gz, gmx, gmy, gmz
256         double precision ln_r, one_r, gr3
257
258   !      if(first_call) then
259   !        first_call = .false.
260   !      endif
```

```
261
262         !リンク原子の位置を決める
263         do i=1, nln_gfg
264           do j=1,nqm_gfg
265             if(jqatms(j)==iqmln_gfg(i)) then
266               rx = crdsmm((immln_gfg(i)-1)*3+1) - coord(1,j)
267               ry = crdsmm((immln_gfg(i)-1)*3+2) - coord(2,j)
268               rz = crdsmm((immln_gfg(i)-1)*3+3) - coord(3,j)
269               exit
270             endif
271           enddo
272           ln_r = 1.09d0/sqrt(rx*rx+ry*ry+rz*rz)
273           do j=1,nqm_gfg
274             if(jqatms(j)==iqmln_gfg(i)) then
275               coord(1,nqm_gfg+i) = coord(1,j) + rx*ln_r
276               coord(2,nqm_gfg+i) = coord(2,j) + ry*ln_r
277               coord(3,nqm_gfg+i) = coord(3,j) + rz*ln_r
278               exit
279             endif
280           enddo
281         enddo
282
283         ! QMと相互作用するMM原子を検索
284         fcharg = .false.
285         mmflgs = .false.
286         indxmm = 1
287         do i=1,nqmln_gfg
288           ni_mm = iqmres(indxmm)
289           indxmm = indxmm + 1
290           do j=1,ni_mm
291             jmm = iqmres(indxmm)
292             jcoord = (jmm-1)*3
293             mmflgs(jmm) = .true.
294             indxmm = indxmm + 1
295           enddo
296         enddo
297         if(any(mmflgs == .true.)) fcharg = .true.
298
299         ! Link原子に結合しているMM原子を省く
300         if(fcharg) then
301           do i=1, nln_gfg
302             mmflgs(immln_gfg(i)) = .false.
303           enddo
304         endif
305
306         ! gaussian input fileを作成する
307         open(io1,file='gaussian.in',status='replace',action='write')
308
309         ! gjf:header
310         write(io1,'(a)') '%nosave'
311         write(io1,'(a)') '%rwf=elec.rwf'
312         if(fcharg) then
313           write(io1,'(a)') trim(gh_route_gfg)//' force iop(6/13=1) charge'
314         else
315           write(io1,'(a)') trim(gh_route_gfg)//' force iop(6/13=1)'
316         endif
317         write(io1,'(a)')
318         write(io1,'(a)') 'QM/MM'
319         write(io1,'(a)')
320         write(io1,'(a)') gh_molsp_gfg
321
322         ! gjf:QM
323         !labels,coordは先頭から使うので、i->jqatms(i)としなくてよい
324         do i=1,nqmln_gfg
325           write(io1,'(i,3f)') labels(i),(coord(j,i),j=1,3)
```

```
      enddo
      write(io1,'(a)')

      ! gjf:MM
      if(fcharg) then
        do i=1,natom_gfg
          if(mmflgs(i)) then
            write(io1,'(4f)')
     &        (crdsmm((i-1)*3+j),j=1,3), chrgs(i)/qscale
          endif
        enddo
        write(io1,'(a)')
        do i=1,natom_gfg
          if(mmflgs(i)) then
            write(io1,'(4f)')
     &        (crdsmm((i-1)*3+j),j=1,3)
          endif
        enddo
        write(io1,'(a)')
      endif

      close(io1)

      ! Gaussianを実行する

      ierr = runqq('cat','gaussian.in >> gaussian-debug.log')
      ierr = runqq('rm','-f elec.rwf')
      ierr = runqq('g03b3','gaussian.in') ! output: elec.rwf
      if(ierr==-1) then
        write(6,*) 'FATAL ERROR IN GAUSSFG - g03 FAILS'
        stop
      endif
      ierr = runqq('rm','-f gdatas.wpp')
      ierr = runqq('./PrtGVal3.exe','elec.rwf') ! output: gdatas.wpp
      if(ierr==-1) then
        write(6,*) 'FATAL ERROR IN GAUSSFG - PrtGval3 FAILS'
        stop
      endif

      ! ファイルからエネルギーと力を読み込む
      open(io1,file='gdatas.wpp',status='old',action='read')

      read(io1,'(a)')
      ! エネルギー
      read(io1,'(3f)') escf, elect, enuclr
!      write(*,*) '!wpp ener:', escf*escale, elect*escale, enuclr*escale
      elect = 0.d0
      enuclr = escf*(escale/23.060362d0)
      escf = (elect+enuclr)*23.060362d0

      if(escf<emin.or.emin==0.d0) emin=escf

      if(icalcf/=0) return !エネルギーだけ計算して力はupdateしない

      read(io1,'(a)')
      ! QM原子とLink原子
      !dxyzqmは先頭から使う
      do i=1,nqmln_gfg
        read(io1,'(3f)') gx, gy, gz
        dxyzqm(1,i) = -gx*fscale
        dxyzqm(2,i) = -gy*fscale
        dxyzqm(3,i) = -gz*fscale
      enddo
      read(io1,'(a)')
      ! MM原子
```

```
391         !dxyzclはそのままの配列
392         if(fcharg) then
393           do i=1,natom_gfg
394             if(mmflgs(i)) then
395               read(io1,'(3f)') gx, gy, gz
396               dxyzcl(1,i) = -gx*fscale
397               dxyzcl(2,i) = -gy*fscale
398               dxyzcl(3,i) = -gz*fscale
399             endif
400           enddo
401         endif
402 
403         close(io1)
404 
405         ! Link原子の力をQM原子,MM原子に分配する
406         do i=1, nln_gfg
407           gx = dxyzqm(1,nqm_gfg+i)
408           gy = dxyzqm(2,nqm_gfg+i)
409           gz = dxyzqm(3,nqm_gfg+i)
410           do j=1,nqm_gfg
411             if(jqatms(j)==iqmln_gfg(i)) then
412               rx = crdsmm((immln_gfg(i)-1)*3+1) - coord(1,j)
413               ry = crdsmm((immln_gfg(i)-1)*3+2) - coord(2,j)
414               rz = crdsmm((immln_gfg(i)-1)*3+3) - coord(3,j)
415               exit
416             endif
417           enddo
418           one_r = 1.d0/sqrt(rx*rx+ry*ry+rz*rz)
419           rx = rx*one_r
420           ry = ry*one_r
421           rz = rz*one_r
422           ln_r = 1.09d0*one_r
423           gr3 = gx*rx+gy*ry+gz*rz
424           gmx = ln_r*(gx-gr3*rx)
425           gmy = ln_r*(gy-gr3*ry)
426           gmz = ln_r*(gz-gr3*rz)
427           dxyzcl(1,immln_gfg(i)) = dxyzcl(1,immln_gfg(i)) + gmx
428           dxyzcl(2,immln_gfg(i)) = dxyzcl(2,immln_gfg(i)) + gmy
429           dxyzcl(3,immln_gfg(i)) = dxyzcl(3,immln_gfg(i)) + gmz
430           do j=1,nqm_gfg
431             if(jqatms(j)==iqmln_gfg(i)) then
432               dxyzqm(1,j) = dxyzqm(1,j) + (gx-gmx)
433               dxyzqm(2,j) = dxyzqm(2,j) + (gy-gmy)
434               dxyzqm(3,j) = dxyzqm(3,j) + (gz-gmz)
435               exit
436             endif
437           enddo
438           dxyzqm(1,nqm_gfg+i) = 0.0d0
439           dxyzqm(2,nqm_gfg+i) = 0.0d0
440           dxyzqm(3,nqm_gfg+i) = 0.0d0
441         enddo
442 
443   !      write(*,'(a)') 'gaussfg: gradient(QM)'
444   !      do i=1,nqmln_gfg
445   !        write(*,'(i5,3(2x,e15.7))') jqatms(i),(dxyzqm(j,i),j=1,3)
446   !      enddo
447   !      write(*,'(a)') 'gaussfg: gradient(MM)'
448   !      do i=1,natom_gfg
449   !        write(*,'(i5,3(2x,e15.7))') i,(dxyzcl(j,i),j=1,3)
450   !      enddo
451 
452    4000 RETURN
453         END
```