Supporting Information

Computational analysis of two-state reactivity in *β*-hydride elimination mechanisms of Fe(II) and Co(II)-alkyl complexes supported by *β*-diketiminate ligand

Toshiyuki Takayanagi*, Kohei Saito, Haruya Suzuki, Yuya Watabe, Takashi Fujihara

Department of Chemistry, Saitama University, Shimo-Okubo 255, Sakura–ku, Saitama City, Saitama 338–8570, Japan

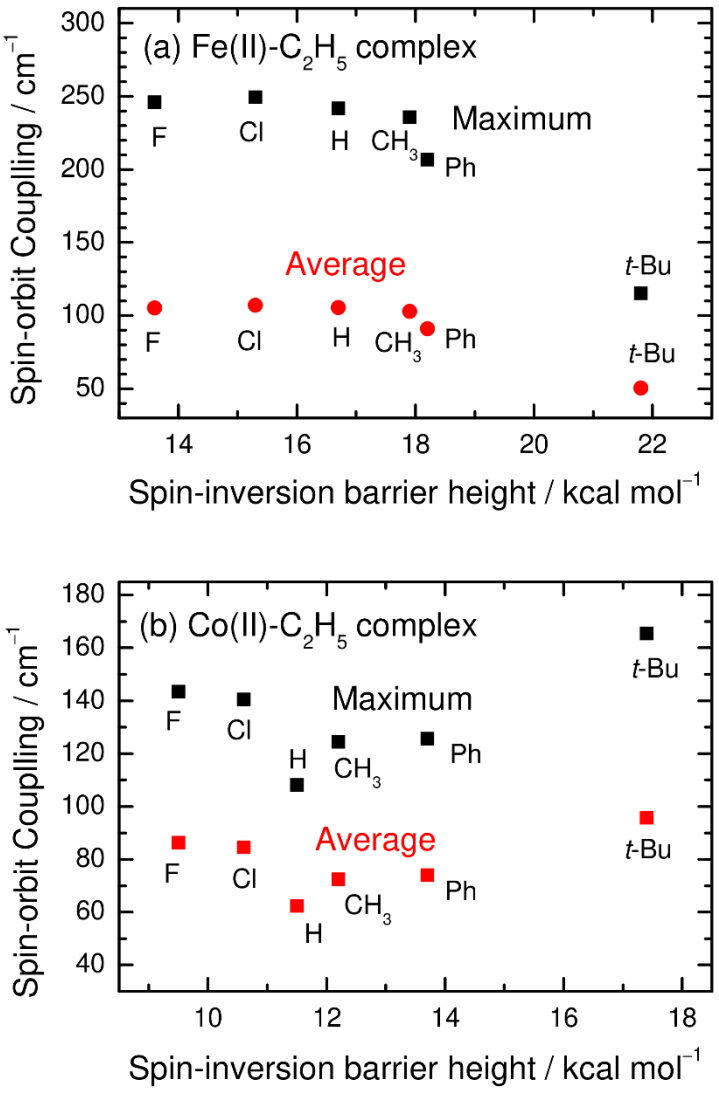

Figure S1

Maximum and averaged spin-orbit couplings as a function of the spin-inversion free barrier height for (a) LFe-C_2H_5 complex and (b) LCo-C_2H_5-complex supported by substituted *β*–diketiminate ligand (R = F, Cl, H, CH_3, Ph, *t*-Bu). The spin-orbit coupling matrix was calculated using the MolSOC code, where the B3LYP-D3/def2-SVPP level results were employed.

Table S1. Spin-orbit coupling matrix calculated at the spin-inversion transition state.

Fe-C_2H_5 complex supported by substituted β-deketiminate (R = F)

ms' | *ms*	2	1	0	–1	–2
1	13.5 + 66.4*i*	–213.0	–5.5 + 27.1*i*	0	0
0	0	9.6 + 46.9*i*	–245.9	–9.6 + 46.9*i*	0
–1	0	0	5.5 + 27.1*i*	–213.0	–13.5 + 66.4*i*

Fe-C_2H_5 complex supported by substituted β-deketiminate (R = Cl)

ms' | *ms*	2	1	0	–1	–2
1	–13.3 – 69.2*i*	216.1	5.4 – 28.2*i*	0	0
0	0	–9.4 – 48.9*i*	249.5	9.4 – 48.9*i*	0
–1	0	0	–5.4 – 28.2*i*	216.1	13.3 – 69.2*i*

Fe-C_2H_5 complex supported by substituted β-deketiminate (R = H)

ms' | *ms*	2	1	0	–1	–2
1	–0.8 – 79.8*i*	209.3	0.3 – 32.6*i*	0	0
0	0	–0.6 – 56.4*i*	241.7	0.6 – 56.4*i*	0
–1	0	0	–0.3 – 32.6*i*	209.3	0.8 – 79.8*i*

Fe-C_2H_5 complex supported by substituted β-deketiminate (R = CH_3)

ms' | *ms*	2	1	0	–1	–2
1	4.3 + 77.4*i*	–204.2	–1.7 + 31.6*i*	0	0
0	0	3.0 + 54.7*i*	–235.8	–3.0 + 54.7*i*	0
–1	0	0	1.7 + 31.6*i*	–204.2	–4.3 + 77.4*i*

Fe-C_2H_5 complex supported by substituted β-deketiminate (R = Ph)

ms' | *ms*	2	1	0	–1	–2
1	29.1 + 66.0*i*	–179.0	–11.9 + 26.9*i*	0	0
0	0	20.6 + 46.7*i*	–206.7	–20.6 + 46.7*i*	0
–1	0	0	11.9 + 26.9*i*	–179.0	–29.1 + 66.0*i*

Fe-C_2H_5 complex supported by substituted β-deketiminate (R = *t*-Bu)

ms' | *ms*	2	1	0	–1	–2
1	–32.4 – 21.5*i*	99.9	13.2 – 8.8*i*	0	0
0	0	–22.9 – 15.2*i*	115.4	22.9 – 15.2*i*	0
–1	0	0	–13.2 – 8.8*i*	99.9	32.4 – 21.5*i*

Co-C_2H_5 complex supported by substituted β-deketiminate (R = F)

ms' | *ms*	3/2	1/2	–1/2	–3/2
1/2	42.4 – 137.0*i*	48.4	–24.5 – 79.1*i*	0
–1/2	0	24.5 – 79.1*i*	48.4	–42.4 – 137.0*i*

Co-C_2H_5 complex supported by substituted β-deketiminate (R = Cl)

ms' | *ms*	3/2	1/2	–1/2	–3/2
1/2	39.4 – 134.8*i*	48.3	–22.7 – 77.8*i*	0
–1/2	0	22.7 – 77.8*i*	48.3	–39.4 – 134.8*i*

Co-C_2H_5 complex supported by substituted β-deketiminate (R = H)

ms' | *ms*	3/2	1/2	–1/2	–3/2
1/2	85.9 – 65.6*i*	–0.04	49.6 – 37.9*i*	0
–1/2	0	–49.6 – 37.9*i*	–0.04	–85.9 – 65.6*i*

Co-C_2H_5 complex supported by substituted β-deketiminate (R = CH_3)

ms' | *ms*	3/2	1/2	−1/2	−3/2
1/2	−51.2 + 113.4*i*	18.3	29.6 + 65.5*i*	0
−1/2	0	−29.6 + 65.5*i*	18.3	51.2 + 113.4*i*

Co-C_2H_5 complex supported by substituted β-deketiminate (R = Ph)

ms' | *ms*	3/2	1/2	−1/2	−3/2
1/2	64.1 − 108.2*i*	−28.3	−37.0 − 62.5*i*	0
−1/2	0	37.0 − 62.5*i*	−28.3	−64.1 − 108.2*i*

Co-C_2H_5 complex supported by substituted β-deketiminate (R = *t*-Bu)

ms' | *ms*	3/2	1/2	−1/2	−3/2
1/2	31.3 − 162.4*i*	12.3	−18.1 − 93.8*i*	0
−1/2	0	18.1 − 93.8*i*	12.3	−31.3 − 162.4*i*

******* Fortran source code for mixed-spin Hamiltonian calculation (mixspin.f) *************

******* Used as Gaussian external program **

```
      program mixspin
      implicit none
      integer    mem, np, spin1, spin2, ieig, nstate
      double precision Hso
      character*50 method
      common /gauss/method
      namelist /input/method, mem, np, spin1, spin2, Hso, ieig, nstate

c**** Please add # character in Method ******

      open(unit=55,file='G09method.inf',form='formatted')
      read(55,input)
      close(unit=55)

      call gaussinout( mem, np, spin1, spin2, Hso, nstate, ieig )

      stop
      end

c**********************************************************************
c*       Control Gaussian input and output
c**********************************************************************

      subroutine gaussinout( mem, np, spin1, spin2, HSO, nstate, ieig )
      implicit none
      integer mem, np, spin1, spin2, nstate, ieig
      integer nmax, natom, i, ii, j, is, ic, ispin
      parameter(nmax=100)
      double precision xcart(0:3*nmax), v1, v2, v, v12, Ap, rtp, HSO
      double precision dv1(0:3*nmax), dv2(0:3*nmax), dv(0:3*nmax)
      double precision a1(0:9*nmax*nmax)
      double precision a2(0:9*nmax*nmax)
      double precision    a(0:9*nmax*nmax)
      double precision x(nmax), y(nmax), z(nmax)
      double precision ffx(3*nmax*3*nmax)
      integer ielm(nmax), i1, j1, k1, j2, k2
      character*80 inp_file, out_file

      call getarg(2,inp_file)
      call getarg(3,out_file)

      open(11,file=trim(inp_file))
      open(12,file=trim(out_file))

      read(11,*) natom, is, ic, ispin
c     write(6,*) natom, is, ic, ispin
      do i=1, natom
         read(11,*) ielm(i), x(i), y(i), z(i), v
```

```
c              write(6,'(i3,3f10.6)') ielm(i), x(i), y(i), z(i)
          enddo
          close(11)

          do i=0, natom-1
               xcart(3*i+0) = x(i+1)*0.529177d0
               xcart(3*i+1) = y(i+1)*0.529177d0
               xcart(3*i+2) = z(i+1)*0.529177d0
          enddo

c**** Perform Gaussian Calculations *********************************

          ispin = spin1
          call g09call(mem,np,is,natom,ic,ispin,ielm,xcart,v1,dv1,a1)
c           write(6,'(f20.10)') v1

          if (nstate.ne.1) then
               ispin = spin2
               call g09call(mem,np,is,natom,ic,ispin,ielm,xcart,v2,dv2,a2)
c              write(6,'(f20.10)') v2
          endif

c**** V12 is the spin-orbit coupling parameter *******

          if (nstate.eq.2) then
               v12 = HSO*2.859d0/1000.0d0/627.51d0
               rtp = (v1-v2)**2+4.0d0*v12**2
               Ap = (v1-v2)/sqrt(rtp)

               if (ieig.eq.1) then
                    v = ((v1+v2)-sqrt(rtp))/2.0d0
               else
                    v = ((v1+v2)+sqrt(rtp))/2.0d0
               endif

               write(12,'(4D20.12)') v, 0.0, 0.0, 0.0

               if (is.eq.0) goto 9000

c**** Write First Derivatives (is = 1) ******************************

               do i=0, 3*natom-1
                    if (ieig.eq.1) then
                         dv(i)=(1.0d0-Ap)/2.0d0*dv1(i)+(1.0d0+Ap)/2.0d0*dv2(i)
                    else
                         dv(i)=(1.0d0+Ap)/2.0d0*dv1(i)+(1.0d0-Ap)/2.0d0*dv2(i)
                    endif
               enddo

               do i=0, natom-1
```

```
            write(12,'(3D20.12)') dv(3*i+0), dv(3*i+1), dv(3*i+2)
        enddo

        if (is.eq.1) goto 9000

c**** Write Second Derivatives (Hessian) (is = 2) ********************

        do i=0, 3*natom-1
        do j=0, 3*natom-1
            if (ieig.eq.1) then
                a(i*3*natom+j) = (1.0d0-Ap)/2.0d0*a1(i*3*natom+j)
     &                          +(1.0d0+Ap)/2.0d0*a2(i*3*natom+j)
     &  +(Ap**2-1.0d0)*(dv1(i)-dv2(i))*(dv1(j)-dv2(j))/2.0d0/sqrt(rtp)
            else
                a(i*3*natom+j) = (1.0d0+Ap)/2.0d0*a1(i*3*natom+j)
     &                          +(1.0d0-Ap)/2.0d0*a2(i*3*natom+j)
     &  -(Ap**2-1.0d0)*(dv1(i)-dv2(i))*(dv1(j)-dv2(j))/2.0d0/sqrt(rtp)
            endif
        enddo
        enddo

        i1=1
        do k1=1, natom
        do j1=1, 3
            do k2=1, natom
            do j2=1, 3
                if(k2*3+j2.le.k1*3+j1) then
                    ffx(i1) = a((3*(k1-1)+j1-1)*3*natom+3*(k2-1)+j2-1)
                    i1=i1+1
                endif
            enddo
            enddo
        enddo
        enddo

        write(12,'(3D20.12)') 0.d0, 0.d0, 0.d0
        write(12,'(3D20.12)') 0.d0, 0.d0, 0.d0

        do i=1, 3*natom
            write(12,'(3D20.12)') 0.d0, 0.d0, 0.d0
        enddo

        do i=1, (3*natom)*(3*natom+1)/2, 3
            write(12,'(3D20.12)') (ffx(j), j=i, i+2)
        enddo

c**** In the case of Nstate = 1 **************************************

      else
            v = v1
```

```
            write(12,'(4D20.12)') v, 0.0, 0.0, 0.0

            if (is.eq.0) goto 9000

c**** Write First Derivatives (is = 1) ******************************

            do i=0, 3*natom-1
                dv(i) = dv1(i)
            enddo

            do i=0, natom-1
                write(12,'(3D20.12)') dv(3*i+0), dv(3*i+1), dv(3*i+2)
            enddo

            if (is.eq.1) goto 9000

c**** Write Second Derivatives (Hessian) (is = 2) ********************

            do i=0, 3*natom-1
            do j=0, 3*natom-1
                a(i*3*natom+j) = a1(i*3*natom+j)
            enddo
            enddo

            i1=1
            do k1=1, natom
            do j1=1, 3
                do k2=1, natom
                do j2=1, 3
                    if(k2*3+j2.le.k1*3+j1) then
                        ffx(i1) = a((3*(k1-1)+j1-1)*3*natom+3*(k2-1)+j2-1)
                        i1=i1+1
                    endif
                enddo
                enddo
            enddo
            enddo

            write(12,'(3D20.12)') 0.d0, 0.d0, 0.d0
            write(12,'(3D20.12)') 0.d0, 0.d0, 0.d0

            do i=1, 3*natom
                write(12,'(3D20.12)') 0.d0, 0.d0, 0.d0
            enddo

            do i=1, (3*natom)*(3*natom+1)/2, 3
                write(12,'(3D20.12)') (ffx(j), j=i, i+2)
            enddo

        endif
```

```
 9000 close(12)

      return
      end

c**********************************************************************
c*     Calculate Potential Energy and Derivatives using G09
c*        Switch = 0    Energy
c*        Switch = 1    Energy + Force
c*        Switch = 2    Energy + Force + Hessian
c**********************************************************************

      subroutine g09call(mem,np,switch,natom,ic,is,iel,x,v,dv,a)
      implicit none
      integer    mem, np
      integer    natom, i, j, k, switch, ic, is, iel(natom)
      double precision    x(0:*), v(0:*), dv(0:*), a(0:*)
      double precision    tmp(0:9*natom*natom)
      character*80    inpch
      character*48    chtmp
      character*50    method
      common /gauss/method

      open(unit=20,file='g09tmp',form='formatted')
      write(20,9000) mem
      write(20,9001) np
      write(20,9002)
      write(20,'(a50)') method
      write(20,9003)
      write(20,9009)
      write(20,*) ' '
      write(20,9004)
      write(20,*) ' '
      write(20,9005) ic, is
      do i=0, natom-1
         write(20,9006) iel(i+1), (x(3*i+j), j=0, 2)
      enddo
      write(20,*) ' '
c     write(20,*) ' '
      close(unit=20)

      call system('cat g09tmp basis.inf > g09tmp.inp')
      call system('g09 < g09tmp.inp > g09tmp.out')
      call system('rm -f g09tmp g09tmp.inp g09tmp.out')

      open(unit=20,file='g09tmp',form='formatted')
c      write(20,9007)
      write(20,9000) mem
      write(20,9001) np
```

```
      write(20,9002)
      write(20,'(a50)') method
      write(20,9008)
      if (switch.eq.0) write(20,9009)
      if (switch.eq.1) write(20,9010)
      if (switch.eq.2) write(20,9011)
      write(20,*) ' '
      write(20,9004)
      write(20,*) ' '
      write(20,9005) ic, is
      do i=0, natom-1
            write(20,9006) iel(i+1), (x(3*i+j), j=0, 2)
      enddo
      write(20,*) ' '
      close(unit=20)

 9000 format('%mem=',i2,'GB')
 9001 format('%nproc='i2)
 9002 format('%chk=g09tmp')
 9003 format('pop=none nosymm stable=opt guess=mix')
 9004 format('comment')
 9005 format(2i5)
 9006 format(i3,3f12.6)
 9007 format('--link1--')
 9008 format('pop=none nosymm guess=read')
 9009 format('scf=(xqc,maxcon=128,maxcyc=512,conver=8)
     & fchk')
 9010 format('scf=(xqc,maxcon=128,maxcyc=512,conver=8)
     & fchk=forcecart force')
 9011 format('scf=(xqc,maxcon=128,maxcyc=512,conver=8)
     & fchk=forcecart freq')

c     call system('cat g09tmp.inp')
      call system('cat g09tmp basis.inf > g09tmp.inp')
      call system('g09 < g09tmp.inp > g09tmp.out')
c     call system('cp g09tmp* $QSUB_WORKDIR')
      call system('rm -f g09tmp g09tmp.inp g09tmp.out')

      open(unit=21,file='Test.FChk',form='formatted')
  100 read(21,'(a80)',end=101) inpch
      if (inpch(1:12).eq.'Total Energy') goto 101
      goto 100
  101 backspace(unit=21)
      read(21,9900) chtmp, v(0)
      close(unit=21)
 9900 format(a48,e23.15)
      if (switch.eq.0) goto 9999

      open(unit=22,file='Test.FChk',form='formatted')
  200 read(22,'(a80)',end=201) inpch
```

```
      if (inpch(1:18).eq.'Cartesian Gradient') goto 201
      goto 200
  201 read(22,'(5e16.8)') (dv(i), i=0, 3*natom-1)
      close(unit=22)
      if (switch.eq.1) goto 9999

      open(unit=23,file='Test.FChk',form='formatted')
  300 read(23,'(a80)',end=301) inpch
      if (inpch(1:25).eq.'Cartesian Force Constants') goto 301
      goto 300
  301 j = 3*natom*(3*natom+1)/2
      read(23,'(5e16.8)') (tmp(i), i=0, j-1)
      close(unit=23)
      k = 0
      do i=0, 3*natom-1
      do j=0, i
          a(i*3*natom+j) = tmp(k)
          a(j*3*natom+i) = tmp(k)
          k = k+1
      enddo
      enddo

 9999 call system('rm -f Test.FChk g09tmp.chk')
      return
      end
```

******** Typical input files (Two input files are needed) **

File: G09method.inf

```
&input
  method = '#OPBE/gen int=ultrafine'
  mem = 4             (Memory in GB to be used in Gaussian09)
  np   = 4            (Number of processors to be used in Gaussian09)
  spin1 = 5           (Spin multiplicity for the first state)
  spin2 = 3           (Spin multiplicity for the second state)
  Hso = 100.0         (Constant spin-orbit parameter in cm–1)
  nstate = 2          (2: Two potential energy surfaces,   1: One potential energy surface)
  ieig = 1            (Choice of the lower (1) or upper (2) eigenstate of   2 × 2 Hamiltonian matrix)
 &end
```

File: Basis.inf

```
C N H Fe 0
def2svpp
****
```