Marcus Theory with Gaussian and ADF — a Tutorial

Svante Hedström*, Benjamin Rudshteyn, Subhajyoti Chaudhuri, and Victor S. Batista

Yale University, Department of Chemistry
225 Prospect Street, New Haven, CT 06520

svante.hedstrom@yale.edu

Erwin with his psi can do
  calculations quite a few.
  but one thing has not been seen
  just what does psi really mean.
  -- Walter Hückel, Trans. by Felix Bloch

Current version written March 2016, based on many previous versions.

Contents

1. Scope and requirements

This tutorial will show you how to compute the classical Marcus Theory rate[1-2] as well as the more advanced Marcus-Jortner-Levich rate[3] using *ab initio* DFT calculations for the biphenyl (donor) - benzoquinonyl (acceptor) dyad, one of the dyads (#**7**) used experimentally to confirm the existence of the Marcus Inverted Region.[4-5] First, we will use the *Gaussian* software package[6] to obtain free energies (ΔG) and reorganization energies (λ) (total, internal, and solvent) necessary to compute Marcus theory. Second, we will use the *ADF* software package[7-11] to compute the couplings between unoccupied orbitals ($|H_{AD}|$). Third, we will use the "*Dushin*" code to compute the "Huang-Rhys" factors.[12] Fourth, we will use a home-made Fortran code that utilizes Monte Carlo to compute an importance-sampled rate using the Marcus-Jortner-Levich expression. The science underlying Marcus Theory and Marcus-Jortner-Levich theory are explained more thoroughly in the publication from our group.[13] The purpose of this tutorial is just to enable you to do the relevant calculations described there.

This tutorial assumes you are familiar with basic Unix commands such as *grep* as well as using the *Gaussian* software package to run geometry optimizations and frequencies as well as generate orbital isosurfaces (cubes) on a supercomputing cluster. The tutorial will bring you up to speed with using *ADF* and compiling a program with the Fortran compiler from Intel called *ifort*. Furthermore, it is assumed that GaussView is installed either locally or on the supercomputer, and a Fortran compiler are installed on the supercomputer. The Fortran compiler *ifort* and GaussView exist as modules on Yale's supercomputers Omega and Grace. To use GaussView on the supercomputer, make sure to ssh with the -Y flag to permit graphical applications running over X-server. Additional instructions are available at
http://research.computing.yale.edu/support/hpc/user-guide/running-gui-cluster-x11-forwarding for either PC or Mac.

There are many files required to follow this tutorial, they can be found at
https://www.dropbox.com/home/Tutorials/ET-calcs/mastertutorial ask any Batista group member to add you to the Tutorials Dropbox folder if you don't have access yet.

2. List of Files Required to Start the Tutorial

Table X0. Files are provided in the Dropbox link https://www.dropbox.com/home/Tutorials/ET-calcs/mastertutorial.

Filename	**Purpose**
BiPh-benzoquinonyl.xyz	Geometry of the dyad.
dushin-Svante.zip	Dushin code files.
frag1.pbs	Submission script for ADF frag1 job (see Table X5).
frag2.pbs	Submission script for ADF frag2 job (see Table X5).
transferint.pbs	Submission script for ADF transferint job (see Table X5).
writeAnADF_version2.sh	Script that takes in the donor and acceptor fragment geometries to set up the ADF input file fast.
param.dat	Parameters used for Monte Carlo code
dushin.dat	Sample input file used for Dushin code

3. List of Files Generated During the Tutorial as well as constants

Table X1. Calculations needed for ΔG for either Marcus Theory or Marcus-Jortner-Levich Theory, from *Gaussian*.

Filename	**Charge Multiplicity**	**Purpose**	**Level of Theory**
BiPh-benzoquinonyl.log	0 1	Geometry optimization of the whole dyad.	b3lyp/6-31g(d)
BiPh-gs.log	0 1	Optimization/frequency on the donor cut to include only the extent of the LUMO+1 (isopropyl groups).	b3lyp/6-31+(2d,p)
benzoquinonyl-gs.log	0 1	Optimization/frequency on the acceptor cut to include only the extent of the LUMO (isopropyl groups).	b3lyp/6-31+(2d,p)
BiPh-anion.log	-1 2	Optimization/frequency on the donor cut to include only the extent of the LUMO+1 (isopropyl groups).	b3lyp/6-31+(2d,p)
benzoquinonyl-anion.log	-1 2	Optimization/frequency on the acceptor cut to include only the extent of the LUMO (isopropyl groups).	b3lyp/6-31+(2d,p)

Table X2. Calculations needed for λ_T (total) for either Marcus Theory or for Marcus-Jortner-Levich Theory, based on geometries from Table X1, from *Gaussian*.

Filename	**Charge**	**Purpose**	**Level of Theory**

	Multiplicity		
lambdatotal/anion-from-neutral/ benzoquinonyl-anion-from-neutral.log	-1 2	Single-point anion calculation on neutral geometry.	b3lyp/6-31+(2d,p)
lambdatotal/neutral-from-anion/ benzoquinonyl-neutral-from-anion.log	0 1	Single-point neutral calculation on anion geometry.	b3lyp/6-31+(2d,p)
lambdatotal/anion-from-neutral/ BiPh-anion-from-neutral.log	-1 2	Single-point anion calculation on neutral geometry.	b3lyp/6-31+(2d,p)
lambdatotal/neutral-from-anion/ benzoquinonyl-neutral-from-anion.log	0 1	Single-point anion calculation on neutral geometry.	b3lyp/6-31+(2d,p)

Table X3. Calculations needed for λ_V (internal, which uses different SCRF keywords) for Marcus Theory, based on geometries from Table X1, from *Gaussian*.

Filename	Charge Multiplicity	Purpose	Level of Theory
lambdainternal/anion-from-neutral/ benzoquinonyl-anion-from-neutral.log	-1 2	Single-point anion calculation on neutral geometry.	b3lyp/6-31+(2d,p)
lambdainternal/neutral-from-anion/ benzoquinonyl-neutral-from-anion.log	0 1	Single-point neutral calculation on anion geometry.	b3lyp/6-31+(2d,p)
lambainternal/anion-from-neutral/ BiPh-anion-from-neutral.log	-1 2	Single-point anion calculation on neutral geometry.	b3lyp/6-31+(2d,p)
lambainternal/anion-from-neutral benzoquinonyl-neutral-from-anion.log	0 1	Single-point anion calculation on neutral geometry.	b3lyp/6-31+(2d,p)

Table X5. Calculations needed for the Boltzmann-averaged coupling ($|H_{AD}|$).

Filename	Charge	Purpose	Level of

	Multiplicity		Theory
scan/B-Sp-benzoquinonyl_scan_24_15.log	0 1	Scan of the ground state in 24 steps of 15°.	b3lyp/ 6-31g(d)
scan/step#/frag1	0 1	single-point calculation on the biphenyl donor cut in the middle of the steroid spacer – needed at every scan step #	b3lyp/ tzvp
scan/step#/frag2	0 1	single-point calculation on the benzoquinonyl acceptor cut in the middle of the steroid spacer – needed at every scan step	b3lyp/ tzvp
scan/step#/transferint	0 1	frag1 and frag2 in the same file to evaluate the charge-transfer integral – needed at every scan step	b3lyp/ zvp

Table X6. Calculations, needed for the cavity volumes that go into distance correction to λ as well as the input for the Dushin code, from *Gaussian*.

Filename	**Charge Multiplicity**	**Purpose**	**Level of Theory**
forDushin/BiPh-gs.log	0 1	Optimization/frequency on the donor cut to exclude the steroid completely.	b3lyp/6-31+(2d,p)
forDushin/benzoquinonyl-gs.log	0 1	Optimization/frequency on the acceptor cut to exclude the steroid completely.	b3lyp/6-31+(2d,p)
forDushin/BiPh-anion.log	-1 2	Optimization/frequency on the donor cut to exclude the steroid completely.	b3lyp/6-31+(2d,p)
forDushin/benzoquinonyl-anion.log	-1 2	Optimization/frequency on the acceptor cut to exclude the steroid completely.	b3lyp/6-31+(2d,p)

Table X7. List of relevant constants.

Filename	**Value**	**Unit**
Temperature (T)	296.00	K
Boltzmann Constant (k_B)	8.61733E-05	eV/K
1 Hartree =	27.2114	eV
Planck's Constant ($\hbar = h / (2\pi)$)	6.58212E-16	eV*s

3. Geometry Optimizations and Frequencies for Free Energy

The file BiPh-benzoquinonyl.xyz contains the geometry of the full dyad with B3LYP/6-31G(d) (same as 6-31G*). It must be first relaxed to a minimum. Use this kind of input file (three dots indicate the rest of the coordinates):

```
%nprocshared=12

%mem=59GB

%chk=BiPh-benzoquinonyl.chk

#p opt b3lyp/6-31g(d) scrf=(smd,solvent=thf)


Optimization of the full dyad


0 1

 C              -4.02694300    1.59526100    0.51043100

.

.

.
```

The optimized structure is given in Figure 1. The SCF energy you get should be close to -1585.71942283 Hartrees.

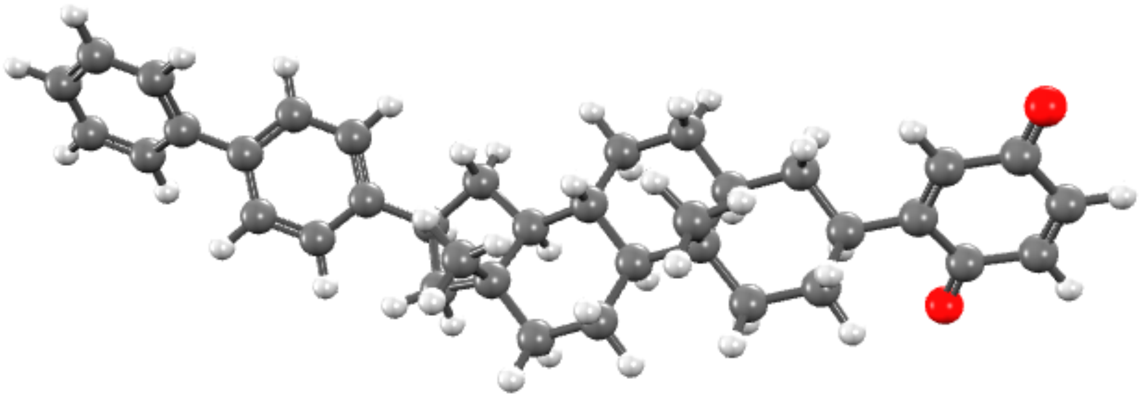


Figure 1. Optimized structure of the full dyad.

The output optimized structure must then be cut into isolated fragments containing part of the steroid linker, as dictated by the extent of delocalization of the orbitals of interest that represent isolated electron density. For this particular case, it is the LUMO+1 of the biphenyl group (donor) and the LUMO of the benzoquinonyl (acceptor), as illustrated by Figure 2. Note that the acceptor has the frontier orbital of lower energy.

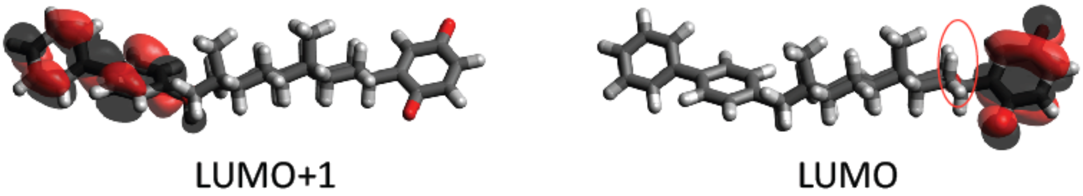


Figure 2. The frontier unoccupied orbitals (isovalue = 0.02) of the dyad with the isopropyl group to remain circled.

Based on the isosurfaces shown in Figure 2, the molecule may be cut into two parts, a donor (biphenyl, left) and an acceptor (benzoquinonyl unit, right), as shown in Figure 3. However, the isodensity extends a bit onto the steroid linker, therefore the cut should be made along bonds 1-2, 3-4, 9-11, and 8-10. After cutting, the valence of the dangling carbon atoms should be filled by hydrogens, resulting in the structure shown in Figure 4.

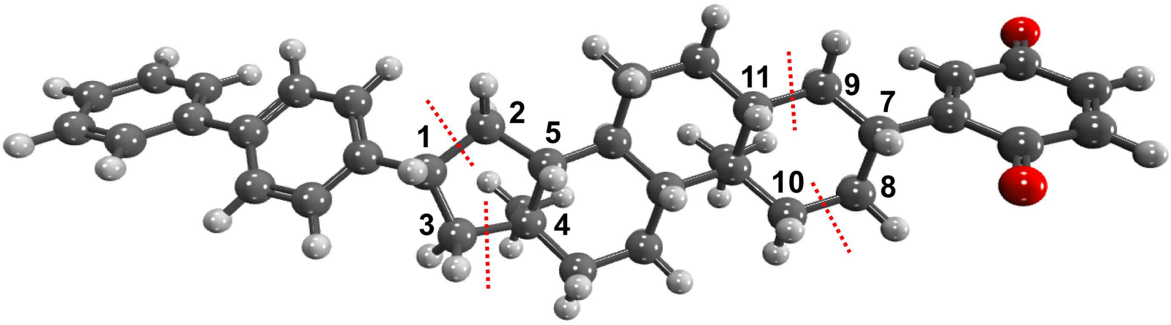


Figure 3. The structure of the whole dyad indicating where the molecule should be cut to generate the fragments that can have isolated charge with particular carbon atoms labeled arbitrarily.

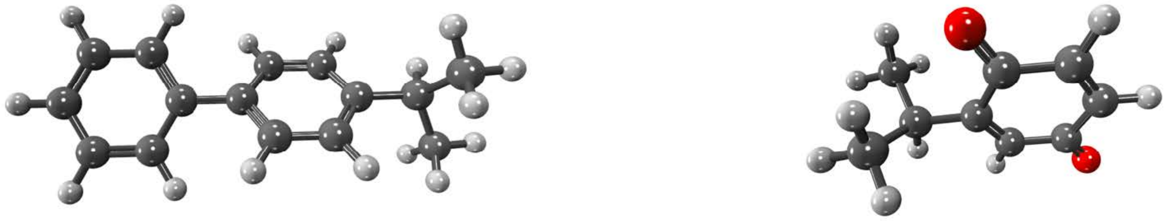


Figure 4. The structure of the dyad fragments of the donor (biphenyl, left) and the acceptor (benzoquinonyl unit, right).

These two dyad fragments, must now be optimized and their frequencies must be calculated, both with either a neutral/singlet and a anionic/doublet electronic state as given below (remember to include at least 2 blank lines at the end of every input file):

A. Donor, ground state of the neutral/singlet.

```
%chk=biph-gs_neutral.chk
%mem=59gb
%nprocshared=12
#p opt freq b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf,Read) nosymm temperature=296

donor ground state of the neutral/singlet

0 1
 C              -2.01831800   -0.00753700    0.00016300
 C              -2.78925900    1.13213800    0.28633000
.
.
.

NonEq=write
```

B. Donor, ground state of the anion/doublet.

```
%chk=biph-gs_anion.chk
%mem=59gb
```

```
%nprocshared=12
#p opt freq b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf,Read) nosymm temperature=296

donor ground state of the anion/doublet

-1 2
 C              -2.01831800   -0.00753700    0.00016300
 C              -2.78925900    1.13213800    0.28633000
.
.
.

NonEq=write
```

C. Acceptor, ground state of the neutral/singlet.

```
%chk=benzoquinonyl-gs_neutral.chk
%mem=59gb
%nprocshared=12
#p opt freq b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf,Read) nosymm temperature=296

acceptor ground state of the neutral/singlet

0 1
 C              -2.35784600    0.81103300   -0.00010100
 C              -1.31684800    1.65469300   -0.00025700.
.
.
.

NonEq=write
```

D. Acceptor, ground state of the anion/doublet.

```
%chk=benzoquinonyl-gs_anion.chk
%mem=59gb
%nprocshared=12
#p opt freq b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf,Read) nosymm temperature=296

acceptor ground state of the anion/doublet
```

```
-1 2
 C              -2.35784600    0.81103300   -0.00010100
 C              -1.31684800    1.65469300   -0.00025700.
.
.
.

NonEq=write
```

Note that these calculations are using a better basis set (6-31+(2d,p)) since the individual calculations are now smaller. Also, note that the command `temperature=296` modifies the output of the frequency calculation to give free energies at 296 K, the temperature of the electron transfer experiments (the temperature does not affect the optimization, since the potential energy surface is not affected). Additionally, these calculations have the command `scrf=(smd,solvent=thf,Read)` to put in the dielectric continuum for THF as well as read in an additional command given after a blank line after the geometry specification. The additional command is `NonEq=write` which saves the state of the continuum to the checkpoint file for calculations of the total reorganization energy later in this tutorial.

The free energies from these calculations should give results similar to these (in Hartrees):

```
A. biph-gs_neutral.log: Sum of electronic and thermal Free Energies=       -581.090755
B. biph-gs_anion.log: Sum of electronic and thermal Free Energies=       -581.144873
C. benzoquinonyl-gs_neutral.log: Sum of electronic and thermal Free Energies=       -499.329756
D. benzoquinonyl-gs_anion.log: Sum of electronic and thermal Free Energies=       -499.461972
```

Therefore the free energy ends up being (with the B & C being the reactants and the A & D being the products of the electron transfer):

$$\Delta G = \left[G(D) + G(A) - \left(G(B) + G(C)\right)\right] * 27.2107\frac{eV}{H} = -2.12540\ eV$$

Compare this value to the experimental value of -2.1 eV.[5]

4. Not-so-Brief Overview of Marcus Theory and Calculating Reorganization Energies

As alluded to in the previous section, the reaction of interest is (D = Donor; A = Acceptor):

$$D^- + A \rightarrow A^- + D$$

The previous section calculated the ΔG:

$$\Delta G = [E_D + E_{A^-} - (E_{D^-} + E_A)]$$

On Figure 5, this value is

$$\Delta G = [E_2 - E_1]$$

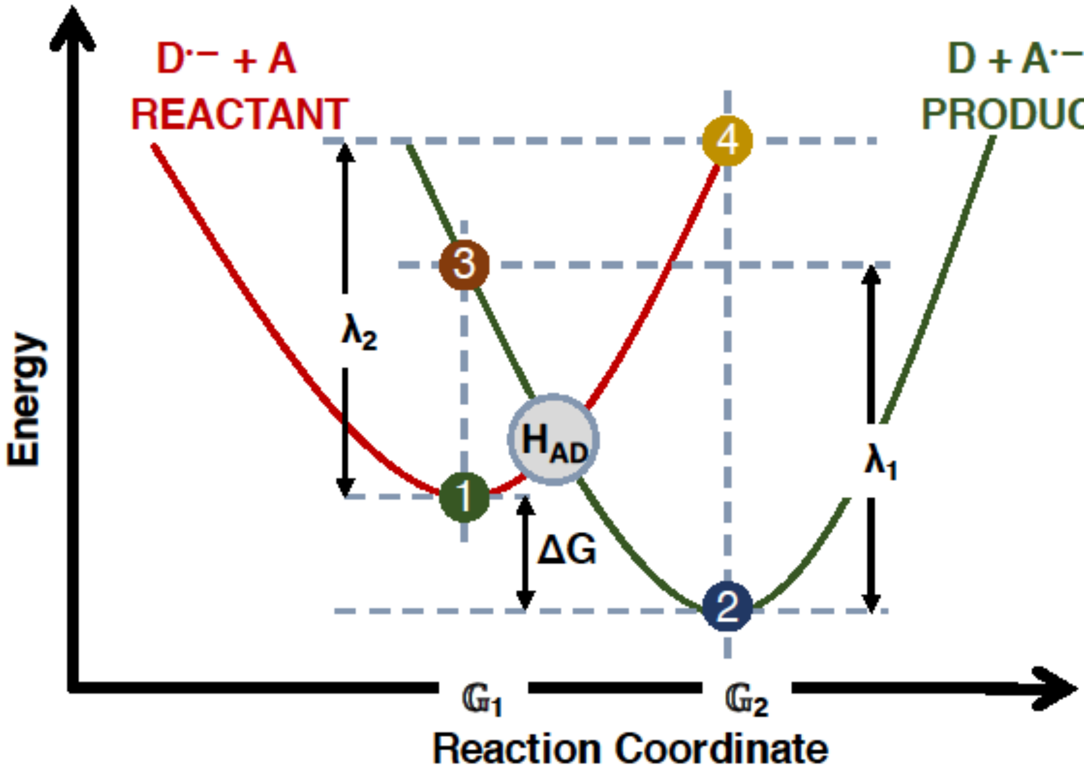


Figure 5. Computation scheme for Marcus parameters.

This free energy of electron transfer ΔG is used in the semiclassical, simplest form of Marcus Theory:

$$k_{ET,\mathrm{Marcus}} = \frac{2\pi|H_{AD}|^2}{\hbar\sqrt{4\pi\lambda k_b T}} \exp\left(-\frac{(\Delta G^0 + \lambda)^2}{4\lambda k_b \mathrm{T}}\right)$$

where k_B is the Boltzmann constant (Table X7), H_{AD} is the electronic coupling between donor and acceptor and λ is the reorganization energy, which is related to the curvature of the parabola. There are two kinds of λ's, one on the reactant potential energy surface (λ_2) and one on the product potential energy surface (λ_1). The two are to be averaged together:

$$\lambda = \frac{\lambda_1 + \lambda_2}{2}$$

The λ to be used here is actually called the *total* reorganization energy (λ_T) and includes the contribution of the donor and the acceptor as well as the solvent.

To calculate this λ_T, we must take the anion doublet donor and do a single point calculation changing the charge & multiplicity to neutral singlet acceptor (to move that point from the reactant to the product parabola). Parallel calculations are done to

- change anion donor to neutral
- change neutral acceptor to anion
- change anion acceptor to neutral

`NonEq=write` and `NonEq=read` keywords are used to fix the (implicit) solvation shell to the initial state. This means that the reorganization energy has contributions from both the change in geometry of the molecule and the solvent i.e. we get the total reorganization energy.

Example input files are shown below to get points 3 and 4 in Figure 5. Note the use of `%oldchk` which will generate a copy of the .chk file. Put them in a different folder, called lambdatotal.

Point 3:

```
%oldchk=../biph-gs_anion.chk
%chk=biph-neutral-from-anion.chk
%mem=59gb
%nprocshared=12
#p b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf,ExternalIteration,Read) geom=check nosymm

donor on neutral surface in the anionic geometry

0 1

NonEq=read
```

and

```
%oldchk=../benzoquinonyl-gs_neutral.chk
%chk=benzoquinonyl-anion-from-neutral.chk
%mem=59gb
%nprocshared=12
#p b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf,ExternalIteration,Read) geom=check nosymm

acceptor on anionic surface in the neutral geometry

-1 2

NonEq=read
```

Point 4:

```
%oldchk=../biph-gs_neutral.chk
%chk=biph-anion-from-neutral.chk
%mem=59gb
%nprocshared=12
#p b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf,ExternalIteration,Read) geom=check nosymm

donor on anionic surface in the neutral geometry

-1 2

NonEq=read
```

and

```
%oldchk=../benzoquinonyl-gs_anion.chk
%chk=benzoquinonyl-neutral-from-anion.chk
%mem=40gb
%nprocshared=8
#p b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf,ExternalIteration,Read) geom=check nosymm

acceptor on neutral surface in the anionic geometry

0 1

NonEq=read
```

Table 1 gives the results you can expect to get. Please note that for reorganization energies, SCF energies, rather than free energies are used.

Table 1. SCF Energies (for the total reorganization energy calculation) from the neutral or

anionic part (the energy for the charge state that comes after “from” in the name is from the calculations in the previous section.

Filename	**anion SCF**	**Number**	**neutral SCF**	**Number**
benzoquinonyl-anion-from-neutral.log	-499.5507263	3	-499.4625761	1
benzoquinonyl-neutral-from-anion.log	-499.5924479	2	-499.4446715	4
biph-anion-from-neutral.log	-581.3204536	4	-581.3147696	2
biph-neutral-from-anion.log	-581.3629364	1	-581.3006836	3

Table 2. Total reorganization energies.

Variable	**Reorganization / eV = total lambda**	**Comment**
λ_2	1.6432	Points 4-1
λ_1	1.5186	Points 3-2
Average:	1.5809	

In principle, the λ_T from Table 2 can be used in the Marcus Theory expression, but it makes the approximation that the donor and acceptor are infinitely separated, which is not a good approximation necessarily if the parts are covalently linked as they are here. Therefore, the following equation is used as part of the routine to correct the energy, where a is the solvent sphere shell radius (D = donor, A = acceptor) and R is the donor to acceptor distance:

$$\lambda_S = \lambda_{S,\infty}\frac{\lambda_{S,R}^{\text{Marcus}}}{\lambda_{S,\infty}^{\text{Marcus}}} = \lambda_{S,\infty}\frac{\left(\frac{1}{2a_D}+\frac{1}{2a_A}-\frac{1}{R}\right)}{\left(\frac{1}{2a_D}+\frac{1}{2a_A}\right)}$$

$\lambda_{S,\infty}$ is the so-called solvent reorganization energy at the infinite distance approximation. λ_S, the solvent reorganization energy without the infinite distance approximation is related to λ_T in the following way:

$$\lambda_s = \lambda_T - \lambda_V$$

where λ_V is the so-called internal reorganization energy.

To apply these equations, first the internal reorganization energy must be calculated. Conveniently, the input files are the same as for the total reorganization energy, except the additional SCRF keywords must be modified as below. Put them in a different folder, called lambdainternal.

Point 3:

```
%oldchk=../biph-gs_anion.chk
%chk=biph-neutral-from-anion.chk
%mem=59gb
%nprocshared=12
#p b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf) geom=check nosymm

donor on neutral surface in the anionic geometry

0 1
```

and

```
%oldchk=../benzoquinonyl-gs_neutral.chk
%chk=benzoquinonyl-anion-from-neutral.chk
%mem=59gb
%nprocshared=12
#p b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf) geom=check nosymm

acceptor on anionic surface in the neutral geometry

-1 2
```

Point 4:

```
%oldchk=../biph-gs_neutral.chk
%chk=biph-anion-from-neutral.chk
%mem=59gb
%nprocshared=12
#p b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf) geom=check nosymm

donor on anionic surface in the neutral geometry

-1 2
```

and

```
%oldchk=../benzoquinonyl-gs_anion.chk
```

```
%chk=benzoquinonyl-neutral-from-anion.chk

%mem=59gb

%nprocshared=12

#p b3lyp/6-31+(2d,p) scrf=(smd,solvent=thf) geom=check nosymm


acceptor on neutral surface in the anionic geometry


0 1
```

Table 3. SCF Energies (for the internal reorganization energy calculation) from the neutral or anionic part (the energy for the charge state that comes after "from" in the name is from the calculations in the previous section.

Filename	anion SCF	Number	neutral SCF	Number
benzoquinonyl-anion-from-neutral.log	-499.5829556	3	-499.4625761	1
benzoquinonyl-neutral-from-anion.log	-499.5924479	2	-499.4530703	4
biph-anion-from-neutral.log	-581.3519352	4	-581.3147696	2
biph-neutral-from-anion.log	-581.3629364	1	-581.3068569	3

Table 4. Internal reorganization energies.

Variable	Reorganization / eV = total lambda	Comment
λ_2	0.5580	Points 4-1
λ_1	0.4736	Points 3-2
Average:	0.5158	

Now we can calculate $\lambda_{S,\infty}$, which is

$$\lambda_{S,\infty} = \lambda_{T,\infty} - \lambda_V = 1.5809\ eV - 0.5158\ eV = 1.0651\ eV$$

Now to calculate the distance correction

$$\frac{\left(\frac{1}{2a_D} + \frac{1}{2a_A} - \frac{1}{R}\right)}{\left(\frac{1}{2a_D} + \frac{1}{2a_A}\right)}$$

we need to determine those a's!

Unfortunately, this requires yet another round of calculations on the same molecules, but with so-called "truncated" spacers, as shown in Figure 6, completely missing the steroid linker. Note that for these calculations, the use of nosymm is omitted, so that symmetry is imposted in the calculation (to allow biphenyl to be planar as necessary). Note that for the Dushin code later, the functional (B3LYP) must be capitalized and the beginning part (#P) must have a capitalized "P". Put them in a folder called "forDushin", the significance of which will be noted later.

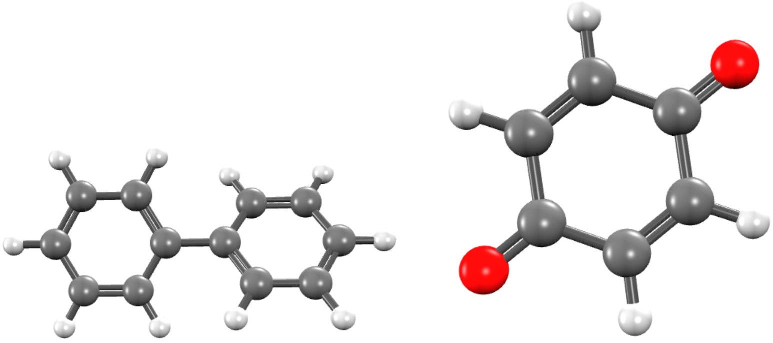


Figure 6. Truncated spacer donor (left) and acceptor (right).

```
%nprocshared=12

%mem=59GB

%chk=BiPh_anion_opt_nosymm.chk

#P opt freq uB3LYP/6-31+(2d,p) scrf=(smd,solvent=thf)


opt freq of biphenyl donor anion


-1 2

 C                 2.01827500    0.00221900    0.00580400

 C                 2.78799800    1.15601900   -0.22073900

.

.

.
```

and

```
%nprocshared=12
%mem=59GB
%chk=BiPh_gs_opt_nosymm.chk
#p opt freq ub3lyp/6-31+(2d,p) scrf=(smd,solvent=thf) nosymm geom=connectivity

opt freq of biphenyl donor ground state neutral

0 1
 C                1.99481900    0.00304600    0.00622400
 C                2.82871300    1.16819800    0.15607900
```

and

```
%nprocshared=12
%mem=59GB
%chk=benzoquinonyl_anion_opt_nosymm.chk
#p opt freq ub3lyp/6-31+(2d,p) scrf=(smd,solvent=thf) nosymm geom=connectivity

opt freq for Dushin of benzoquinonyl acceptor anion

-1 2
 C               -2.35005300    0.80923800   -0.00008600
 C               -1.31227300    1.66041300   -0.00020800
```

and

```
%nprocshared=12
%mem=59GB
%chk=benzoquinonyl_gs_opt_nosymm.chk
#p opt freq ub3lyp/6-31+(2d,p) scrf=(smd,solvent=thf) nosymm geom=connectivity

opt freq for Dushin of benzoquinonyl ground state neutral

0 1
 C               -2.33358500    0.76530000   -0.00007200
 C               -1.27259200    1.63560000   -0.00021100
```

Once confirmed not to have any imaginary frequencies, these calculations will yield the The value retrieved by the following command:

```
grep 'GePol: Cavity volume                           =' *log
```

appears as something like:

```
GePol: Cavity volume                           =    150.952 Ang**3
```

Where the number is the cavity volume in units of Å^3. These volumes are given in Table 5 and are converted to radii in Table 6 using the equation for the volume of a sphere:

$$V = \frac{4}{3}\pi r^3 \therefore r = \frac{3}{4\pi}\sqrt[3]{V}$$

Table 5. Cavity volume of the molecules in Figure 6 in units of Å^3.

Biphenyl	Benzoquninonyl
173.0370	150.4220

Table 6. The cavity radius of sphere (Å) of the molecules (neutral) in Figure 6.

a_D	a_A
3.4569	3.2992

We also need the center-to-center separation of the donor and acceptor in the actual dyad, shown in Figure 7 and Table 7. This separation is the average between the distance between the farthest H's and the closest C's (not including the spacer) in the structure of the whole dyad.

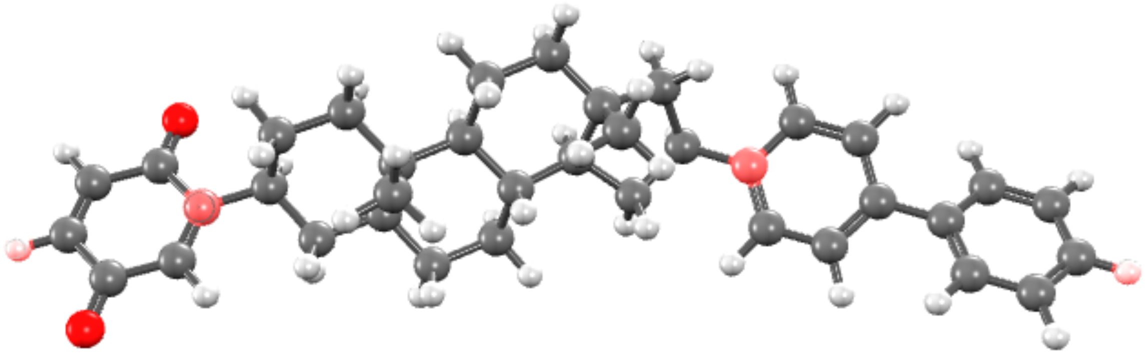


Figure 7. Optimized structure of the full dyad with the farthest H's (white) and the closest C's (black) labeled.

Table 7.

R (farthest H) / Å	R (closest C) / Å	R (Center to Center Distance) / Å
23.6232	11.9010	17.7621

Using the equation earlier for correcting the reorganization energy of the solvent, we get a λ_S=0.8099, which yields a $\lambda_T = \lambda_S + \lambda_V$ = 0.8099 eV + 0.5158 eV = 1.3257 eV.

5. Coupling with ADF

Before commencing the coupling calculation, a scan of the acceptor-spacer dihedral angle is required (put it in a subfolder named scan). Figure 8 depicts which dihedral angle will be scanned by 24 steps of 15°.

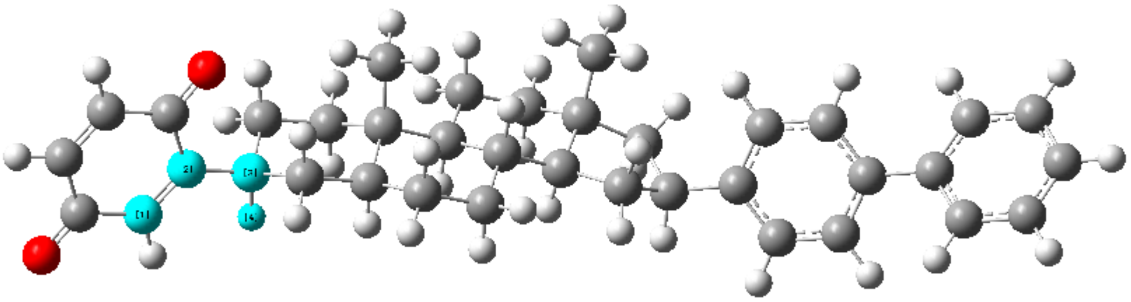


Figure 8. Dihedral angle defined by the blue four atoms used in the scan. The atoms have the labels 39, 13, 71, and 73, from right to left.

```
%nprocshared=12
%mem=59GB
%chk=B-Sp-benzoquinonyl_scan_24_15.chk
#p opt=modredundant b3lyp/6-31g(d) scrf=(smd,solvent=thf) geom=connectivity

B-Sp-benzoquinonyl-631gd-HF scanning 24 steps of 15 degrees

0 1
 C                -4.02694300    1.59526100    0.51043100
 C                -2.95026800    0.47882500    0.56297500
.
.
.

D 39 13 71 73 S 24 15.000000
```

The resulting scan should look like the plot in Figure 9. The data needed to generate the scan energies can be easily generated by opening the log file in Gaussview 5.0 and clicking Results > Scan > right click on y axis > Properties > Y-axis Units > eV > Current > OK > right click on graph > Save Data. This procedure will generate an x,y data file as usual.

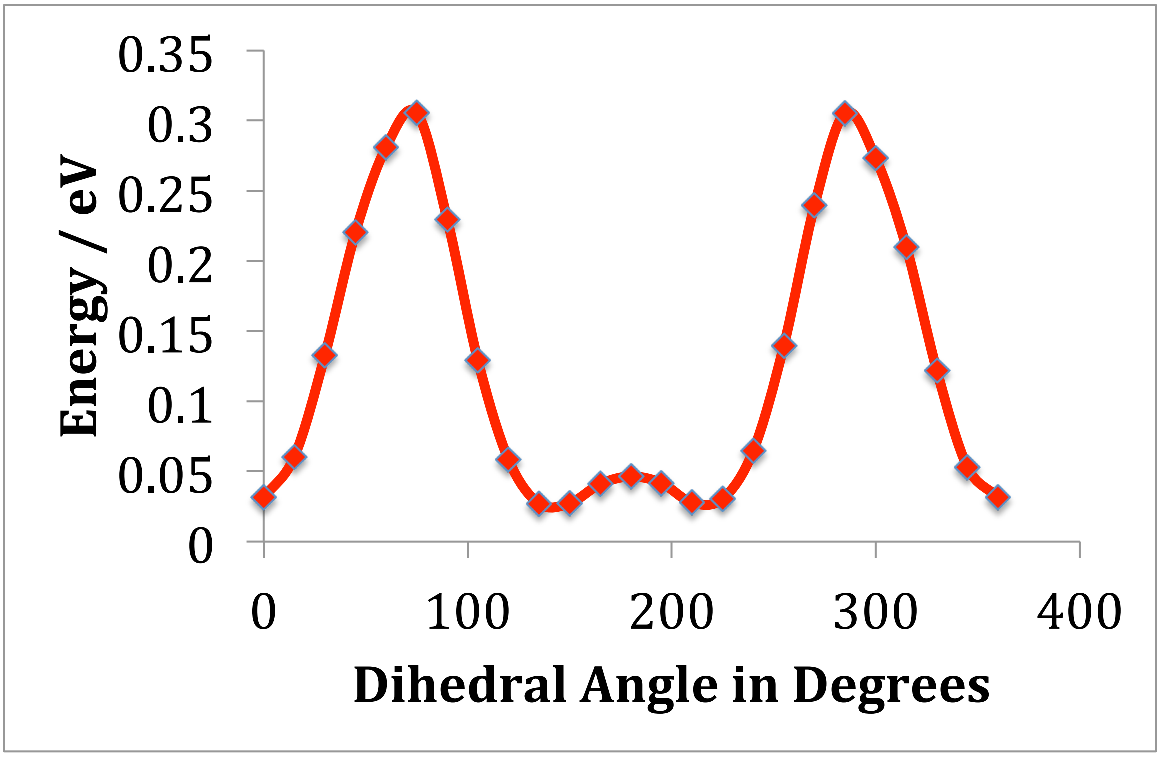


Figure 9. The resulting potential energy surface defined by the dihedral angle scan.

In the subfolder, run the following command to generate a subfolder for each geometry of the scan:

```
for f in `seq 0 15 345` ; do echo $f ; mkdir $f ; done
```

In each of the resulting folders, save the geometries of the scan through GaussView as .gjf or .com files. Figure 10 shows how these geometries ought to be cut in order to set up the coupling calculations. Figure 11 shows the intermediate geometry-building step where extra C atoms are retained so that they may be turned into H's. The resulting structures are given in Figure 12. The donor is termed "frag1" in this section and the acceptor is termed "frag2" in this section, so each angle subfolder should be a file called frag1.gjf and frag2.gjf. These files should be converted into .xyz file format manually, defined by the number of atoms on top, a blank line, and the xyz coordinates (in Å), e.g. for the hydrogen molecule:

```
2

this is a comment line

H 0.0 0.0 0.0

H 0.0 0.0 0.9
```

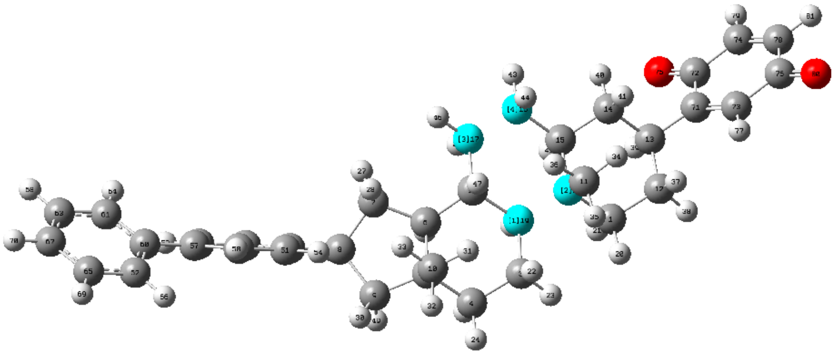


Figure 10. How to cut the molecule in the middle of the spacer in order to set up the coupling calculations. The carbon atoms labeled in blue should be replaced with hydrogens to fill the valence after cutting, rather than adding a brand new connection, in order to preserve the dihedral angle.

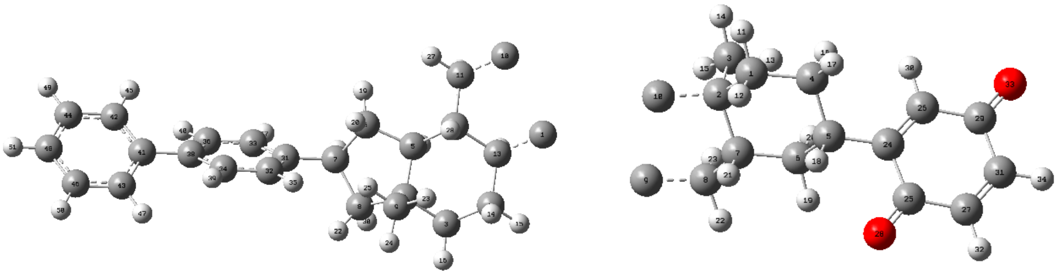


Figure 11. Extraneous carbons indicated by dashed bonds. These must be converted into H's by clicking in GaussView 5.0.

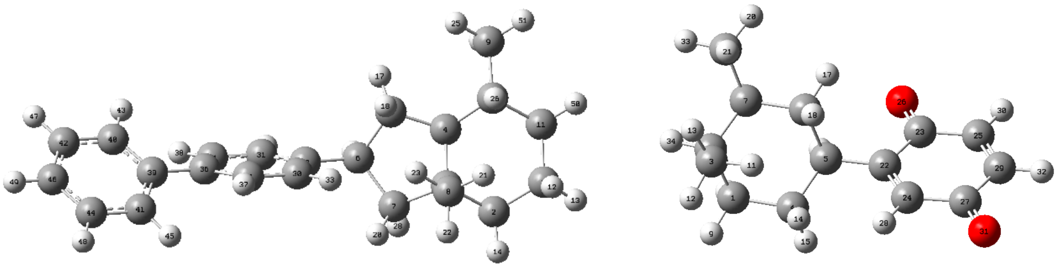


Figure 12. Resulting structures for coupling calculations (frag1 on the left for donor and frag2 on the right for acceptor).

```
bash -x ../../writeAnADF_version2.sh frag1.xyz frag2.xyz
```

This script outputs the input files needed for the ADF calculation.

frag1 looks like the file below (phrases in parentheses do not actually occur in the file).

frag2 looks the same but with “DM2” on top.

```
TITLE DM1 fragment (comment line)
ATOMS (xyz atom specification)
 C              -1.09290900   2.57878000  -0.54644600
 C               0.29514800   2.77511700  -1.19327100
.
.
.
END

INTEGRATION 5.0 5.0 5.0 (size of Voronoi integration grid)

CHARGE 0  (neutral charge on system)

XC  (what functional are you using?)
 Hybrid B3LYP
END

BASIS (what basis set are you using?)
 type TZ2P
END

SYMMETRY NOSYM (no symmetry allowed)

END INPUT (denotes the end of the input file)
```

The transferint input file is harder to set up. The script handles it, but it is basically like frag1 and frag2, but with the coordinates put together and with extra keywords. See the file below (again, phrases in parentheses do not actually occur in the file).

```
TITLE DIMER
ATOMS
  C              -1.09290900   2.57878000  -0.54644600 f=DM1 (coordinates from frag1)
  C               0.29514800   2.77511700  -1.19327100 f=DM1
.
.
.
  C              -4.04232000   1.69159200  -0.67984300 f=DM2 (coordinates from frag2)
  C              -2.98064200   0.88880600   0.12048000 f=DM2
.
.
.
END
```

```
INTEGRATION 5.0 5.0 5.0

CHARGE 0

XC
 Hybrid B3LYP
END

BASIS
 TYPE TZ2P
END

FRAGMENTS (new keywords that refer to binary files created after running frag1 & frag2 sucessfully)
 DM1 DM1.tape21
 DM2 DM2.tape21
END

TRANSFERINTEGRALS (calculates the transfer integral)

SYMMETRY NOSYM

END INPUT
```

The submission script (Yale’s Omega) for ADF looks like the following (frag1.pbs, provided; change `firstname.lastname@yale.edu` to your email to get a notification when the job is done):

```
#!/bin/bash -l
#PBS -N frag1
#PBS -l walltime=24:00:00
#PBS -l nodes=2:ppn=12
#PBS -l mem=60gb
#PBS -q esi
#PBS -m ae -M firstname.lastname@yale.edu

module purge
module load Applications/ADF/2012.01
mkdir -p $PBS_O_WORKDIR/frag1_temp
export SCM_TMPDIR=$PBS_O_WORKDIR/frag1_temp
export SCM_IOBUFFERSIZE=256

cd $SCM_TMPDIR
```

```
$ADFBIN/adf < $PBS_O_WORKDIR/frag1 > $PBS_O_WORKDIR/frag1.out
```

The same kind of file is appropriate for frag2. However, the transferint.pbs looks different, because it handles the moving around of the frag1 and frag2 binary files into the temporary folder for the transferint calculation.

```
#!/bin/bash -l
#PBS -N transferint
#PBS -l walltime=24:00:00
#PBS -l nodes=2:ppn=12
#PBS -l mem=60gb
#PBS -q esi
#PBS -m ae -M firstname.lastname@yale.edu

module purge
module load Applications/ADF/2012.01
mkdir -p $PBS_O_WORKDIR/transferint_temp
export SCM_TMPDIR=$PBS_O_WORKDIR/transferint_temp
export SCM_IOBUFFERSIZE=256

cd $SCM_TMPDIR
cp ../frag1_temp/TAPE21 DM1.tape21
cp ../frag2_temp/TAPE21 DM2.tape21

$ADFBIN/adf < $PBS_O_WORKDIR/transferint > $PBS_O_WORKDIR/transferint.out
```

Upload these kinds of files for every scan step folder to the cluster. First, the frag1 and frag2 calculations must be run. ONLY then may the transferint calculation be run. The relevant output will be in the resulting transferint.out file.

The coupling is to be calculated using the equation below (where $S_{AD}^2 \ll 1$):

$$\mathrm{H}_{AD} = \frac{\mathrm{J}_{AD} - \frac{1}{2}\mathrm{S}_{AD}(e_D + e_A)}{\sqrt{1 - S_{AD}^2}} \approx \mathrm{J}_{AD} - \frac{1}{2}\mathrm{S}_{AD}(e_D + e_A)$$

Since we are interested in the LUMO-LUMO coupling between the fragments, the following Unix commands may be used to quickly obtain the coupling in each folder (the \ indicates that the command should be a one liner):

```
SAD=$(grep 'Overlap integral (electron) LUMO fragment 1 - LUMO fragment 2' transferint.out \
| awk '{print $NF}')
eD=$(grep 'Site energy (electron) LUMO fragment 1 (eV)' transferint.out | awk '{print $NF}')
eA=$(grep 'Site energy (electron) LUMO fragment 2 (eV)' transferint.out | awk '{print $NF}')
JAD=$(grep 'Charge transfer integral (electron) LUMO fragment 1 - LUMO fragment 2 (eV)' \
transferint.out | awk '{print $NF}')

coupling=$(echo  " $Jif - ( 0.5 * $SAD ) * ( $eD + $eA ) " | bc -l  )

echo "LUMO-LUMO" > reportLUMOLUMO.txt

echo "SAD $SAD" >> reportLUMOLUMO.txt

echo "eD $eD" >> reportLUMOLUMO.txt

echo "eA $eA" >> reportLUMOLUMO.txt

echo "JAD $JAD" >> reportLUMOLUMO.txt

echo "Coupling $coupling" >> reportLUMOLUMO.txt

echo '' >> reportLUMOLUMO.txt
```

The resulting `reportLUMOLUMO.txt` file will contain the coupling in eV as calculated by the $coupling variable (compare to equation above). If the coupling of interest were not LUMO-LUMO (or similar thing like HOMO-LUMO or HOMO-HOMO), then it must be retrieved more manually as so (highlights indicate for LUMO-LUMO coupling; note the occupancy of the orbital to help locate it).

For S_{AD} (the two yellow numbers are equivalent; either may be used):

```
===================================
 Overlap matrix in SFO representation
 ===================================


                   Fragment:   1 DM1

                        FO:     58 A        59 A        60 A        61 A        62 A        63 A

                     E(eV):      -6.975      -6.919      -6.072      -0.976      -0.408      -0.278

                       Occ:       2.00        2.00        2.00        0.00        0.00        0.00

      E(eV) Occ     FO          ----------  ----------  ----------  ----------  ----------  ----------

  1 DM1

    -6.975  2.00   58 A           1.00000     0.00000     0.00000     0.00000     0.00000     0.00000

    -6.919  2.00   59 A           0.00000     1.00000    -0.00001     0.00000     0.00000     0.00000

    -6.072  2.00   60 A           0.00000    -0.00001     0.99995     0.00002     0.00000     0.00000

    -0.976  0.00   61 A           0.00000     0.00000     0.00002     0.99999     0.00000     0.00000

    -0.408  0.00   62 A           0.00000     0.00000     0.00000     0.00000     1.00000     0.00000

    -0.278  0.00   63 A           0.00000     0.00000     0.00000     0.00000     0.00000     1.00000

  2 DM2

    -7.861  2.00   41 A          -0.00002     0.00014    -0.00023     0.00006    -0.00004     0.00002

    -7.741  2.00   42 A           0.00003    -0.00001    -0.00114     0.00036    -0.00007     0.00010

    -7.449  2.00   43 A          -0.00004     0.00018     0.00071    -0.00018     0.00007    -0.00003

    -3.645  0.00   44 A           0.00001     0.00000     0.00004     0.00001     0.00001    -0.00002



                   Fragment:   2 DM2

                        FO:      41  A       42  A       43  A       44  A
```

```
                    E(eV):     -7.861       -7.741       -7.449       -3.645
                      Occ:      2.00         2.00         2.00         0.00
         E(eV) Occ    FO             ----------   ----------   ----------   ----------
     1 DM1
        -6.975  2.00   58 A            -0.00002      0.00003     -0.00004      0.00001
        -6.919  2.00   59 A             0.00014     -0.00001      0.00018      0.00000
        -6.072  2.00   60 A            -0.00023     -0.00114      0.00071      0.00004
        -0.976  0.00   61 A             0.00006      0.00036     -0.00018      0.00001
        -0.408  0.00   62 A            -0.00004     -0.00007      0.00007      0.00001
        -0.278  0.00   63 A             0.00002      0.00010     -0.00003     -0.00002
     2 DM2
        -7.861  2.00   41 A             0.99962     -0.00020     -0.00028      0.00008
        -7.741  2.00   42 A            -0.00020      0.99980     -0.00007      0.00005
        -7.449  2.00   43 A            -0.00028     -0.00007      0.99973      0.00005
        -3.645  0.00   44 A             0.00008      0.00005      0.00005      0.99998
```

For S_D (first yellow highlight), S_A (second yellow highlight), and J_{AD} (either green highlight):

```
================================

 Fock matrix in SFO representation  ***  (eV)  ***

 ================================


                    Fragment:   1 DM1

                          FO:     58 A         59 A         60 A         61 A         62 A         63 A

                      E(eV):      -6.975       -6.919       -6.072       -0.976       -0.408       -0.278

                        Occ:       2.00         2.00         2.00         0.00         0.00         0.00

       E(eV) Occ     FO           ----------   ----------   ----------   ----------   ----------   ----------

  1 DM1

     -6.975  2.00   58 A           -7.06917      0.00734      0.00145      0.00003     -0.00261      0.00398

     -6.919  2.00   59 A            0.00734     -7.04883     -0.00465     -0.00215     -0.00595     -0.00584

     -6.072  2.00   60 A            0.00145     -0.00465     -6.20735      0.01827     -0.00207     -0.00313

     -0.976  0.00   61 A            0.00003     -0.00215      0.01827     -1.09187     -0.00175     -0.00254

     -0.408  0.00   62 A           -0.00261     -0.00595     -0.00207     -0.00175     -0.51578     -0.01913

     -0.278  0.00   63 A            0.00398     -0.00584     -0.00313     -0.00254     -0.01913     -0.39458

  2 DM2

     -7.861  2.00   41 A            0.00159     -0.00584     -0.00824      0.00276     -0.00001      0.00002

     -7.741  2.00   42 A           -0.00039     -0.00073      0.01994     -0.00519      0.00038     -0.00070

     -7.449  2.00   43 A            0.00208     -0.00629     -0.02686      0.00731     -0.00084      0.00022

     -3.645  0.00   44 A           -0.00041      0.00095      0.00309     -0.00060     -0.00024     -0.00040
```

```
                     Fragment:   2 DM2
                          FO:     41  A         42  A         43  A         44  A
                       E(eV):     -7.861        -7.741        -7.449        -3.645
                         Occ:      2.00          2.00          2.00          0.00
        E(eV)  Occ     FO              ----------   ----------   ----------   ----------
    1  DM1
      -6.975  2.00    58  A             0.00159     -0.00039      0.00208     -0.00041
      -6.919  2.00    59  A            -0.00584     -0.00073     -0.00629      0.00095
      -6.072  2.00    60  A            -0.00824      0.01994     -0.02686      0.00309
      -0.976  0.00    61  A             0.00276     -0.00519      0.00731     -0.00060
      -0.408  0.00    62  A            -0.00001      0.00038     -0.00084     -0.00024
      -0.278  0.00    63  A             0.00002     -0.00070      0.00022     -0.00040
    2  DM2
      -7.861  2.00    41  A            -7.46981     -0.22792     -0.11608      0.03641
      -7.741  2.00    42  A            -0.22792     -7.39177      0.02399      0.03904
      -7.449  2.00    43  A            -0.11608      0.02399     -6.93549     -0.00026
      -3.645  0.00    44  A             0.03641      0.03904     -0.00026     -3.13594
```

At the end of the day, the coupling as a function of dihedral angle should resemble the plot in Figure 13.

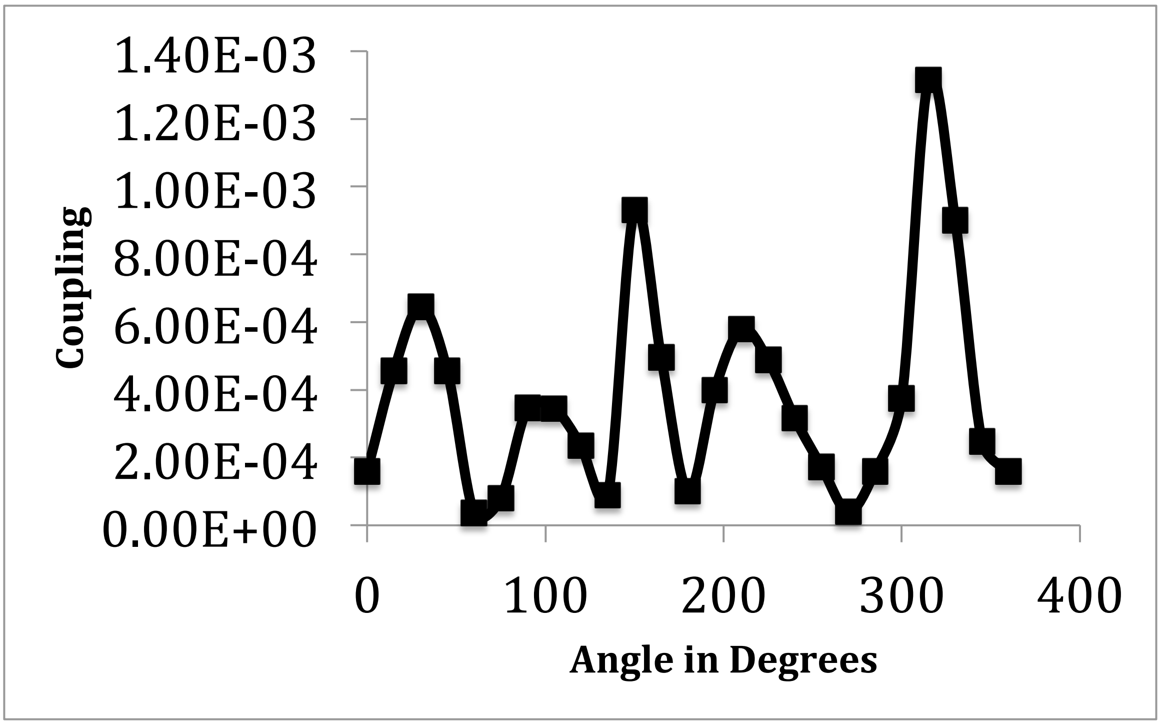


Figure 13. The coupling as a function of the dihedral angle scan.

To determine the coupling that goes into the Marcus Theory equations, we need to run a Boltzmann average over this scan using the energies in Figure 9:

$$|\mathrm{H}_{AD}(\mathrm{T})|^2 = \frac{\int |\mathrm{H}_{DA}(\varphi)|^2 \mathrm{e}^{\frac{-\Delta E(\varphi)}{RT}} d\varphi}{\int \mathrm{e}^{\frac{-\Delta E(\varphi)}{RT}} d\varphi} \approx \frac{\sum_{\varphi=0^\circ}^{345^\circ}\left(|\mathrm{H}_{DA}(\varphi)|^2 \mathrm{e}^{\frac{-\Delta E(\varphi)}{RT}}\right)}{\sum_{\varphi=0^\circ}^{345^\circ} \mathrm{e}^{\frac{-\Delta E(\varphi)}{RT}}}$$

$\Delta E(\varphi)$ should be defined relative to the minimum energy of the system (here where the angle is 135°. Table 8 shows you how to calculate $|\mathrm{H}_{DA}(\varphi)|$.

Table 8. Results of the coupling scant to illustrate how to do the Boltzmann averaging and the results to expect. The energy/coupling are in units of eV. The minimum energy on the surface is highlighted in yellow. The bolded result is the final coupling to be used.

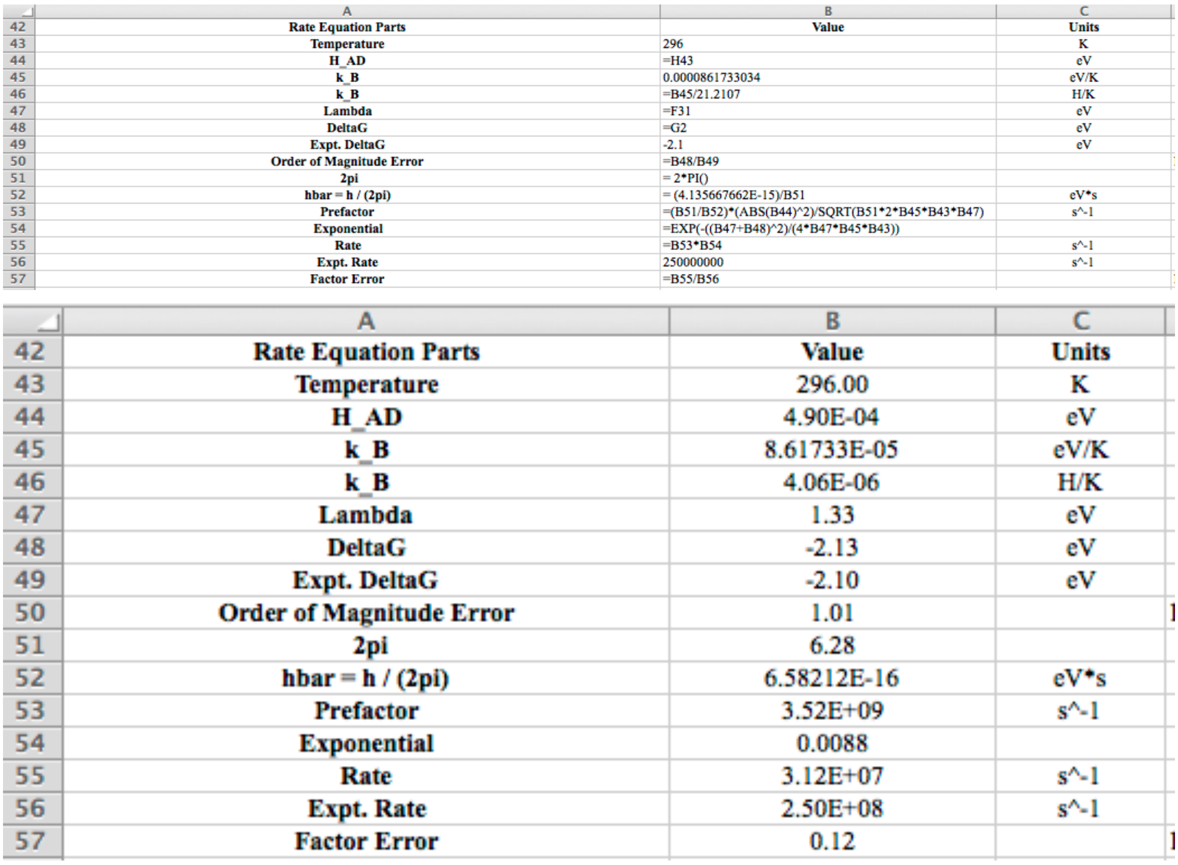
Step	**Dihedral Angle:**	**E**	**Coupling:**	**Coupling Squared** $|\mathrm{H}_{DA}(\varphi)|^2$	**Boltzmann Factor** $\mathrm{e}^{\frac{-\Delta E(\varphi)}{RT}}$	**Coupling Squared * Boltzmann Factor**	**Sqrt(Boltzmann Averaged Coupling)**
1	0.00	0.03	-1.59E-04	2.52E-08	8.36E-01	2.11E-08	**0.000490369**
2	15.00	0.06	-4.57E-04	2.08E-07	2.72E-01	5.67E-08	
3	30.00	0.13	-6.46E-04	4.17E-07	1.60E-02	6.66E-09	
4	45.00	0.22	-4.55E-04	2.07E-07	5.12E-04	1.06E-10	
5	60.00	0.28	-3.62E-05	1.31E-09	4.75E-05	6.23E-14	
6	75.00	0.31	-7.96E-05	6.34E-09	1.82E-05	1.15E-13	
7	90.00	0.23	3.45E-04	1.19E-07	3.56E-04	4.24E-11	
8	105.00	0.13	3.45E-04	1.19E-07	1.83E-02	2.17E-09	
9	120.00	0.06	-2.36E-04	5.56E-08	2.91E-01	1.62E-08	
10	135.00	0.03	8.83E-05	7.80E-09	1.00E+00	7.80E-09	
11	150.00	0.03	-9.31E-04	8.67E-07	9.89E-01	8.58E-07	
12	165.00	0.04	-4.96E-04	2.46E-07	5.79E-01	1.42E-07	
13	180.00	0.05	-1.00E-04	1.00E-08	4.68E-01	4.68E-09	
14	195.00	0.04	4.00E-04	1.60E-07	5.64E-01	9.03E-08	
15	210.00	0.03	-5.79E-04	3.35E-07	9.72E-01	3.26E-07	
16	225.00	0.03	-4.88E-04	2.38E-07	8.72E-01	2.07E-07	
17	240.00	0.06	3.15E-04	9.95E-08	2.28E-01	2.27E-08	
18	255.00	0.14	-1.73E-04	3.00E-08	1.21E-02	3.61E-10	
19	270.00	0.24	-4.06E-05	1.65E-09	2.41E-04	3.99E-13	
20	285.00	0.31	-1.59E-04	2.53E-08	1.82E-05	4.61E-13	
21	300.00	0.27	-3.76E-04	1.41E-07	6.45E-05	9.10E-12	
22	315.00	0.21	-1.32E-03	1.73E-06	7.67E-04	1.33E-09	
23	330.00	0.12	-9.00E-04	8.10E-07	2.45E-02	1.98E-08	
24	345.00	0.05	2.48E-04	6.13E-08	3.62E-01	2.22E-08	

6. Calculating a Marcus Rate

Recall the Marcus Theory Equation (note it is a *prefactor* * *exponential*):

$$k_{ET,\text{Marcus}} = \frac{2\pi|H_{AD}|^2}{\hbar\sqrt{4\pi\lambda k_b T}} \exp\left(-\frac{(\Delta G^0 + \lambda)^2}{4\lambda k_b \text{T}}\right)$$

we now have all the parts needed to calculate a rate:



	A	B	C
42	Rate Equation Parts	Value	Units
43	Temperature	296	K
44	H_AD	=H43	eV
45	k_B	0.0000861733034	eV/K
46	k_B	=B45/21.2107	H/K
47	Lambda	=F31	eV
48	DeltaG	=G2	eV
49	Expt. DeltaG	-2.1	eV
50	Order of Magnitude Error	=B48/B49	
51	2pi	= 2*PI()	
52	hbar = h / (2pi)	= (4.135667662E-15)/B51	eV*s
53	Prefactor	=(B51/B52)*(ABS(B44)^2)/SQRT(B51*2*B45*B43*B47)	s^-1
54	Exponential	=EXP(-((B47+B48)^2)/(4*B47*B45*B43))	
55	Rate	=B53*B54	s^-1
56	Expt. Rate	250000000	s^-1
57	Factor Error	=B55/B56	

	A	B	C
42	Rate Equation Parts	Value	Units
43	Temperature	296.00	K
44	H_AD	4.90E-04	eV
45	k_B	8.61733E-05	eV/K
46	k_B	4.06E-06	H/K
47	Lambda	1.33	eV
48	DeltaG	-2.13	eV
49	Expt. DeltaG	-2.10	eV
50	Order of Magnitude Error	1.01	
51	2pi	6.28	
52	hbar = h / (2pi)	6.58212E-16	eV*s
53	Prefactor	3.52E+09	s^-1
54	Exponential	0.0088	
55	Rate	3.12E+07	s^-1
56	Expt. Rate	2.50E+08	s^-1
57	Factor Error	0.12	

Figure 14. Equations (top) and values (bottom) for calculating the rate according ot Marcus Theory. Note the theory underestimates the rate by a factor of 0.12.

7. Running the Dushin Code to get the effective frequency

Using the log files from the calculations done for Figure 6 (including both anions and neutrals), run a separate frequency calculation as so for the example below. After these four jobs are complete, run the following kind of command on the .chk (you may need to load the Gaussian module first) to create a .fchk file with the same name (minus the extension):

```
formchk -3 benzoquinonyl_gs_justfreq.chk
```

```
%nprocshared=12
%mem=59GB
%oldchk=benzoquinonyl_gs.chk
%chk=benzoquinonyl_gs_justfreq.chk
#p freq ub3lyp/6-31+(2d,p) scrf=(smd,solvent=thf) nosymm geom=check guess=read

Frequency calculation for Dushin of benzoquinonyl ground state neutral

0 1
```

Into the same folder on the cluster, upload dushin-Svante.zip and run the following command to open the compressed file (the module provides the ifort compile and yun118 is Svante's XSEDE allocation):

```
unzip dushin-Svante.zip
```

Then the following script (appropriate for XSEDE's Comet)

```
#!/bin/bash
#SBATCH -p shared
#SBATCH -A yun118
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH -t 01:00:00
#SBATCH -J compile_SLURM
#SBATCH -o compile_SLURM.%j.%N.out
#SBATCH -e compile_SLURM.%j.%N.err
#SBATCH --export=ALL
#SBATCH --no-requeue

module load intel/2013_sp1.2.144

echo "compiling dushin.for" > report_compile.txt
ifort -c -g dushin.for
echo "compiling bmatred.for" >> report_compile.txt
ifort -c -O bmatred.for
echo "compiling project0freq.for" >> report_compile.txt
ifort -c -O proj0freq.for
echo "compiling recalc-freq.for" >> report_compile.txt
ifort -c -O recalc-freq.for
echo "compiling subs.for" >> report_compile.txt
ifort -c -g subs.for
echo "compiling subs-dftb.for" >> report_compile.txt
ifort -c -g subs-dftb.for
echo "compiling ddiag.for dmpower.for dmatinv.for" >> report_compile.txt
ifort -c -O ddiag.for dmpower.for dmatinv.for
```

```
echo './dushin line' >> report_compile.txt
ifort -g dushin.o subs.o recalc-freq.o proj0freq.o bmatred.o ddiag.o dmpower.o dmatinv.o
-o ./dushin
echo './displace line' >> report_compile.txt
ifort -g displace.for subs.o recalc-freq.o proj0freq.o bmatred.o ddiag.o dmpower.o dmatinv.o
-o ./displace
#echo './plot-modes line' >> report_compile.txt
#ifort -g plot-modes.for subs.o recalc-freq.o proj0freq.o bmatred.o ddiag.o dmpower.o dmatinv.o
-o ./plot-modes
echo './compare-geom line' >> report_compile.txt
ifort -g compare-geom.for subs.o recalc-freq.o proj0freq.o bmatred.o ddiag.o dmpower.o dmatinv.o
-o ./compare-geom
```

If the compilation was successful, the report_compile.txt file should read:

```
compiling dushin.for
compiling bmatred.for
compiling project0freq.for
compiling recalc-freq.for
compiling subs.for
compiling subs-dftb.for
compiling ddiag.for dmpower.for dmatinv.for
./dushin line
./displace line
./compare-geom line
```

Once compiled, the Dushin input file (dushin.dat) must be prepared (this example is only for the biphenyl donor):

```
2 0  'curvilinear flag and atom order flag'
.

2 1 'gs' 'BiPh_gs_justfreq.log'
0 1 'anion' 'BiPh_anion_opt_nosymm_freq.log'
```

The numbers in this file are not the charge/multiplicity! The first digit (2) indicates that approximate curvilinear is used (0=recti, 1=orth curvi, 2=apprx curvi', ', -1 = skip'). The second digit (0) indicates that the “atom ordering flag” never changes (0= never changes, 1= use the first order thenceforth, 2= deduce it (does read on fail), 3= always read it). The third digit (2) indicates that this is the “initial and primary reference”. The fourth digit (1) indicates that the ‘first order thenceforth' is used. The fifth digit (0) indicates the structure is never used a reference. The sixth digit (1) indicates that the ‘first order thenceforth' is used.

Now run the Dushin code with the following submission script:

```
#!/bin/bash
#SBATCH -p shared
#SBATCH -A yun118
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH -t 01:00:00
#SBATCH -J run_SLURM
#SBATCH -o run_SLURM.%j.%N.out
#SBATCH -e run_SLURM.%j.%N.err
#SBATCH --export=ALL
#SBATCH --no-requeue

module load intel/2013_sp1.2.144
./dushin > dushin.log
```

The important output will be written to dushin.log. After running it, rename the files:

```
mv dushin.log dushin_donor.log
mv dushin.dat dushin_donor.dat
```

Then rerun for the acceptor:

Once compiled, the Dushin input file (dushin.dat) must be prepared (this example is only for the benzoquinonyl acceptor):

```
2 0   'curvilinear flag and atom order flag'
.


2 1 'gs' 'benzoquinonyl_gs_justfreq.log'
0 1 'anion' 'benzoquinonyl_anion_justfreq.log'
```

After running with the same script above, rename the files:

The important output will be written to dushin.log. After running it, rename the files:

```
mv dushin.log dushin_acceptor.log
mv dushin.dat dushin_acceptor.dat
```

The following script will assist you in collecting the frequencies and Q's from Dushin. Note the line with `'{print $10 " " $12}'` is required for systems with high symmetry as Dushin prints an extra column (so now the 10th and 12th columns need to be grabbed).

```
#!/bin/bash
donor=./dushin_donor.log
```

```
acceptor=./dushin_acceptor.log
grep 'lam=' $donor | sed '1,6d' | awk '{print $3 " " $5}' > freq_Qdushin-donor-gs.dat
grep 'lam=' $donor | sed '1,6d' | awk '{print $10 " " $12}' > freq_Qdushin-donor-an.dat
grep 'lam=' $acceptor | sed '1,6d' | awk '{print $3 " " $5}' > freq_Qdushin-acceptor-gs.dat
grep 'lam=' $acceptor | sed '1,6d' | awk '{print $9 " " $11}' > freq_Qdushin-acceptor-an.dat
cat freq_Qdushin-acceptor-an.dat > freq_Qdushin_1.dat
cat freq_Qdushin-donor-gs.dat >> freq_Qdushin_1.dat
cat freq_Qdushin-donor-an.dat > freq_Qdushin_2.dat
cat freq_Qdushin-acceptor-gs.dat >> freq_Qdushin_2.dat
sed -i.bk '/0.000/d' freq_Qdushin_?.dat
```

The resulting `freq_Qdushin_1.dat` and `freq_Qdushin_2.dat` contain an x,y data column of frequencies and Q values (the motion of normal modes) for the anion-acceptor/neutral-donor and the anion-donor/neutral-acceptor systems respectively. The commands above remove the rows with Q values of 0 since they will not contribute to any equations (some of them are deleted by the `sed '1,6d'` which deletes the first 6 lines of the file. Please also manually delete any rows containing a Q>5 as at that point, the harmonic approximations underlying Marcus Theory break down. Paste these data files into Excel.

Calculate a third row:

$$S = \frac{Q^2}{2}$$

and a fourth

$$S * \omega$$

so that the header row for both dat files are given in Table 9 and 10.

Table 9. Results for donor anion and acceptor neutral `freq_Qdushin_2.dat`.

	Freq / cm^{-1}	**Q**	**S**	**Freq * S**
	…	**…**	**…**	**…**
Sum:			4.4951825	3940.406577
w_eff=				876.5843381

Table 10. Results for donor neutral and acceptor anion `freq_Qdushin_1.dat`.

	Freq / cm^{-1}	**Q**	**S**	**Freq * S**
	…	**…**	**…**	**…**
Sum:			3.5187635	3238.331435
w_eff=				920.303804

What these tables give is

$$\omega_{\mathrm{eff}} = \frac{\sum_{i=1}^{N} S_i \omega_{\mathrm{i}}}{\sum_{i=1}^{N} S_i}$$

The actual ω_{eff} to use ends up being the average of the two values from Table 9 and 10: 898.44 cm^{-1}.

The S$_{\mathrm{eff}}$ (Huang-Rhys factor) is (recall λ_V = internal reorganization energy)

$$\mathrm{S}_{\mathrm{eff}} = \frac{\lambda_V}{\hbar\omega_{\mathrm{eff}}}$$

However, rather than use the internal reorganization energy calculated previously, we will use the internal reorganization energy calculated by the Dushin code. Run the following script to collect the values. The above equations come from the literature.[14-15]

```
#!/bin/bash
donor=./dushin_donor.log
acceptor=./dushin_acceptor.log

reorgDonorNeutral=$(grep 'total reorg energy' $donor | awk '{print $6}'  )
reorgDonorAnion=$(grep 'total reorg energy' $donor | awk '{print $7}'  )
reorgAcceptorNeutral=$(grep 'total reorg energy' $acceptor | awk '{print $6}'  )
reorgAcceptorAnion=$(grep 'total reorg energy' $acceptor | awk '{print $7}'  )

echo -e "reorganization energy donor neutral \t $reorgDonorNeutral" > reorg_energies.txt
echo -e "reorganization energy donor anion \t $reorgDonorAnion" >> reorg_energies.txt
echo -e "reorganization energy acceptor neutral \t $reorgAcceptorNeutral" >> reorg_energies.txt
echo -e "reorganization energy acceptor anion \t $reorgAcceptorAnion" >> reorg_energies.txt
```

Using the data in reorg_energies.txt, you can generate Table 11.

Table 11. Reorganization Energy calculation via Dushin with averages calculated accorind go the color code.

From Dushin:	**Reorg Energy / cm^{-1}**	**Reorg Energy / eV**	**Average**	**Sum**
reorganization energy donor neutral	2366	0.29	0.342	0.597
reorganization energy donor anion	3145	0.39		
reorganization energy acceptor neutral	2132	0.26	0.255	
reorganization energy acceptor anion	1982	0.25		

So the $\mathrm{S}_{\mathrm{eff}} = \frac{\lambda_V}{\hbar\omega_{\mathrm{eff}}} = \frac{[0.597\ eV]}{[6.58212*10^{-16} eV*s]*2\pi*[3*\frac{10^{10} cm}{s}]*[898.44 cm^{-1}]} = 5.35$

Which used the fact that the frequency υ in s^{-1} is related to the speed of light in a vacuum $3*10^{10}$ cm/s and the wavenumbers $\tilde{\nu}$ in cm^{-1}.

$$\upsilon = \frac{c}{\lambda} = c\tilde{\nu}$$

8. Marcus-Jortner-Levich (MJL) One Effective Mode Rate

Now we have what we need to calculate this "gnarly" MJL one effective mode rate.

$$k_{\mathrm{ET,MJL,1eff}} = \frac{\pi|H_{AD}|^2}{\hbar\sqrt{\pi\lambda_S k_b T}} \sum_{v=0}^{\infty} e^{-S}\frac{S^v}{v!}\exp\left(-\frac{(\Delta\mathrm{G}^0 + \lambda_S + v\hbar\omega_{\mathrm{eff}})^2}{4\lambda_S \mathrm{k}_B \mathrm{T}}\right)$$

Note that the λ_S, not the λ_T is used. The prefactor is also slightly different. The exponential has an extra term and now there is a lot of stuff related to the vibrational quantum number v, which will be varied from 0 until convergence of the sum is achieved. This equation for each term can be difficult to program, so try doing this in Excel (replace with cell #'s as appropriate) (assumes all energies are in eV and constants have the eV unit within them):

=EXP(-S)*(S^v/FACT(v))*EXP(-((deltaG+lambdas+v*hbar*2*PI()*weff)^2)/(4*lambdas*kB*T))

where FACT = factorial and EXP = exponential.

Table 12 and Figure 15 indicate how fast the summation converges.

Table 12. Convergence of the summation as the term is calculated for v = 0 to v = 18. Note how the value stops changing around v = 17 and the term peaks around v = 8.

v	**Term**	**Sum**
0	1.79630E-12	1.79630E-12
1	3.14663E-10	3.16460E-10
2	2.03040E-08	2.06205E-08
3	6.43467E-07	6.64088E-07
4	1.12676E-05	1.19317E-05
5	1.16285E-04	1.28217E-04
6	7.36778E-04	8.64995E-04
7	2.94782E-03	3.81282E-03
8	7.60278E-03	1.14156E-02
9	1.28408E-02	2.42564E-02
10	1.43797E-02	3.86361E-02

11	1.07849E-02	4.94211E-02
12	5.46255E-03	5.48836E-02
13	1.88152E-03	5.67651E-02
14	4.43342E-04	5.72085E-02
15	7.18298E-05	5.72803E-02
16	8.03785E-06	5.72883E-02
17	6.23658E-07	5.72890E-02
18	3.36688E-08	5.72890E-02

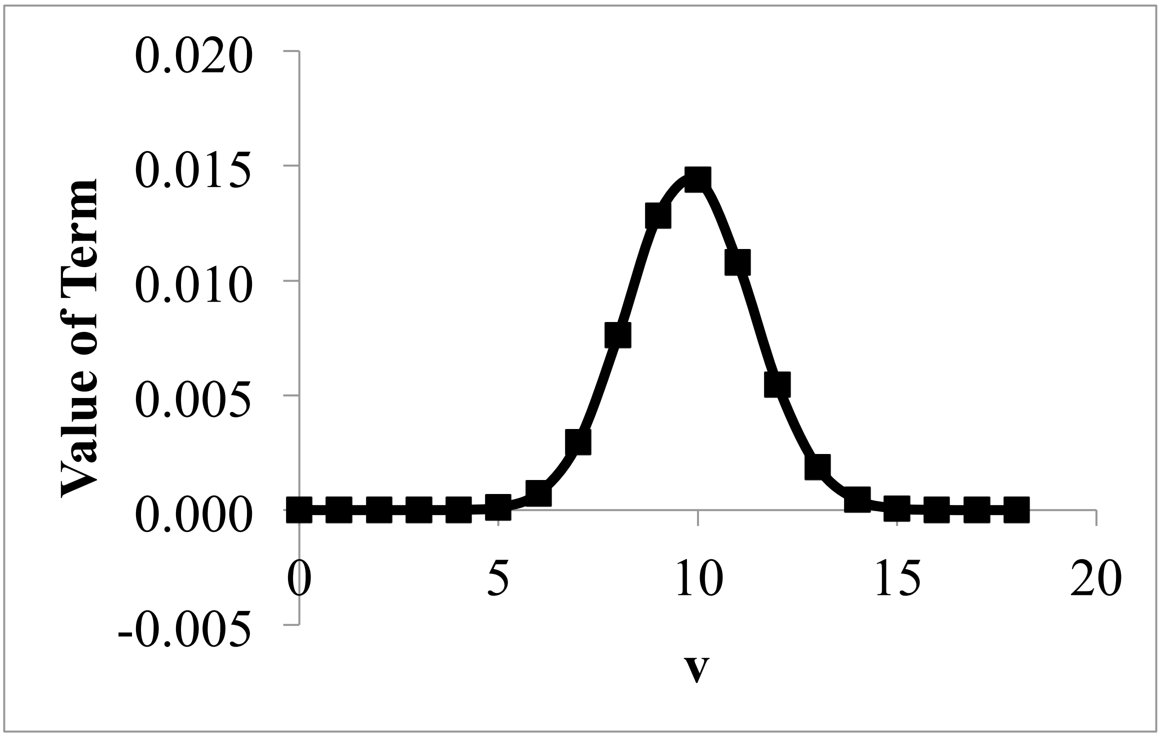


Figure 15. The value of the terms in the summation as a function of v.

Now we can calculate a new $\lambda_{s,\infty}$ with the λ_V from Dushin, which is

$$\lambda_{s,\infty} = \lambda_{T,\infty} - \lambda_V = 1.5809\ eV - 0.5967\ eV = \ 0.9842\ eV$$

which leads to a value of

$$\lambda_s = 0.7971\ \mathrm{eV}$$

We can calculate the prefactor this way in Excel:

```
=(PI()/hbar)*(ABS(H_AD)^2)/SQRT(PI()*lambdas*kB*T)
```

Multiplying this factor together with the converged sum, we get a rate of 2.6 * 10^8 s^{-1}, which matches extremely well with the experimental rate of 2.5 * 10^8 s^{-1}.[5]

9. Marcus-Jortner-Levich (MJL) All Modes Rate

Now we have what we need to calculate this even "gnarlier" MJL all mode rate (see equation below) using the fancy Monte Carlo code called `mck_CleanCommented.f` along with the associated input file param.dat. Every time it is run, preferably please do it as so (on the cluster, where there is gfortran):

```
gfortran -o mck_CleanCommented mck_CleanCommented.f ; ./mck_CleanCommented > mck_report.txt
```

This command will compile the code, run it, and write its output to a file called `mck_report.txt`.

$$k_{\text{ET,MJL,expl}} = \frac{\pi|H_{AD}|^2}{\hbar\sqrt{\pi\lambda_S k_b T}} \exp\left(\sum_{i=1}^{N} -S_i\right) \times \sum_{v_1=0}^{\infty}\cdots\sum_{v_i=0}^{\infty}\cdots\sum_{v_N=0}^{\infty}\left(\prod_{i=1}^{N}\left(\frac{{S_i}^{v_i}}{v_i!}\right) \times \exp\left(-\frac{(\Delta G^0+\lambda_S+\sum_{i=1}^{N}\hbar\omega_i v_i)^2}{4\lambda_S k_B \mathrm{T}}\right)\right)$$

The param.dat file must be edited:

```
      IMPLICIT NONE
      INTEGER nm,nd,np,NFR
      INTEGER*16 NMC
      REAL*8 hbar,Es,beta,dG,VAD
      PARAMETER(nm=11,nd=66)
     PARAMETER (NMC=100000000,np=1,NFR=1,hbar=1.0)
      PARAMETER (beta=1066.807587628)
      PARAMETER (Es=0.7971/27.21139664)
      PARAMETER (dG=-2.12540/27.21139664)
      PARAMETER (VAD=0.0004903692/27.21139664)
!   np: Number of energy points to plot
!   nm: max quantum number nu. nu is integer between 0 and nm
!   nd: # of vibrational modes (dimensions) (the number of lines in the freq_Qdushin.dat file)
!   Es: External reorg. energy, AKA lambda_S, in Hartree, taken from gaussian calcs
!   beta = 1/(kB T) in Hartree, T=296 K as in Closs paper
!   NFR: reading flag (NFR=1 when reading NM frequencies & reduced displ from file
!   freq_Qdushin.dat)
!   dG: delta G for the reaction, taken from Gaussian calcs
!   VAD: coupling in Hartrees, taken from excel and coupling calcs
```

The lines starting with "!" are ignored by the program and just explain what each variable means. Replace the number before the "/27.21139664" with the value in eV ("/27.21139664" is the conversion factor from eV to Hartrees, since the code works in so-called atomic units). The values above are what we get for this system. Recall that

$$\beta = \frac{1}{k_B T} = \frac{1}{8.61722 * 10^{-5}\frac{eV}{K} * \frac{1}{27.2107}\frac{H}{eV} * 298\,K} = 1066.78\,H^{-1}$$

When ready, copy the

```
cp freq_Qdushin_1.dat freq_Qdushin.dat
```

and run the code

```
gfortran -o mck_CleanCommented mck_CleanCommented.f ; ./mck_CleanCommented > mck_report.txt
```

and run the code. You'll get a "histo" and a mck_report.txt file as a result if everything went well.

Rename these files so you don't overwrite them:

```
cp histo histo_1

cp mck_report.txt mck_report_1.txt
```

Repeat the procedure for the second set of data:

```
cp freq_Qdushin_2.dat freq_Qdushin.dat

gfortran -o mck_CleanCommented mck_CleanCommented.f ; ./mck_CleanCommented > mck_report.txt

cp histo histo_2

cp mck_report.txt mck_report_2.txt
```

The output of the files `mck_report_1.txt` and `mck_report_2.txt` should resemble these (The resulting `1` and `2` are for the anion-acceptor/neutral-donor and the anion-donor/neutral-acceptor systems respectively):

1)

```
log(k_MC), k_MC=   7.8627230399445835       72899246.592478096

 norm_sampled=   1.0000000000000000

 norm_analytic=   1.0000000000000002
```

2)

```
log(k_MC), k_MC=   8.2132639417625892       163404473.42020246

 norm_sampled=   1.0000000000000000

 norm_analytic=   1.0000000000000002
```

The norms should be or close to 1 if everything is normally. Above that, the first value is the log (base 10) of the rate and the second value is the rate itself. Taking the average of these two numbers, we get:

log(average) = 8.072440563

average =118151860 = $1.2*10^8$ s^{-1}

which compares also well to the experimental rate 2.5 * 10^8 s^{-1}.[5]

10. References

1. Marcus, R. A., On the Theory of Oxidation‐Reduction Reactions Involving Electron Transfer. I. *J. Chem. Phys.* **1956,** *24* (5), 966-978.

2. Marcus, R. A.; Sutin, N., Electron Transfers in Chemistry and Biology. *Biochim. Biophys. Acta* **1985,** *811* (3), 265-322.

3. Ulstrup, J.; Jortner, J., The Effect of Intramolecular Quantum Modes on Free Energy Relationships for Electron Transfer Reactions. *J. Chem. Phys.* **1975,** *63* (10), 4358-4368.

4. Closs, G. L.; Miller, J. R., Intramolecular Long-Distance Electron Transfer in Organic Molecules. *Science* **1988,** *240* (4851), 440-448.

5. Miller, J.; Calcaterra, L.; Closs, G., Intramolecular Long-Distance Electron Transfer in Radical Anions. The Effects of Free Energy and Solvent on the Reaction Rates. *J. Am. Chem. Soc.* **1984,** *106* (10).

6. Gaussian 09 Revision D.01, F., M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Montgomery, J. A., Jr.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, N. J.; Klene, M.; Knox, J. E.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Zakrzewski, V. G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, Ö.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. Gaussian, Inc., Wallingford CT, 2009.

7. te Velde, G. t.; Bickelhaupt, F. M.; Baerends, E. J.; Fonseca Guerra, C.; van Gisbergen, S. J.; Snijders, J. G.; Ziegler, T., Chemistry with ADF. *J. Comput. Chem.* **2001,** *22* (9), 931-967.

8. *ADF, SCM, Theoretical Chemistry,*, Vrije Universiteit: Amsterdam, 2014.

9. Guerra, C. F.; Snijders, J.; Te Velde, G.; Baerends, E., Towards an Order-N DFT method. *Theor. Chem. Acc.* **1998,** *99* (6), 391-403.

10. Boerrigter, P. M.; Te Velde, G.; Baerends, J. E., Three-Dimensional Numerical Integration for Electronic Structure Calculations. *Int. J. Quantum Chem* **1988,** *33* (2), 87-113.

11. te Velde, G.; Baerends, E. J., Numerical Integration for Polyatomic Systems. *Journal of Computational Physics* **1992,** *99* (1), 84-98.

12. Reimers, J. R., A Practical Method for the Use of Curvilinear Coordinates in Calculations of Normal-Mode-Projected Displacements and Duschinsky Rotation Matrices for Large Molecules. *The Journal of Chemical Physics* **2001,** *115* (20), 9103-9109.

13. Chaudhuri, S.; Hedström, S.; Méndez-Hernández, D. D.; Hendrickson, H. P.; Jung, K. A.; Ho, J.; Batista, V. S., Quantitative First Principles Predictions of Electron Transfer Rates. **2016,** *In Preparation*.

14. Stehr, V.; Fink, R. F.; Engels, B.; Pflaum, J.; Deibel, C., Singlet Exciton Diffusion in Organic Crystals Based on Marcus Transfer Rates. *J. Chem. Theor. Comp.* **2014,** *10* (3), 1242-1255.

15. Stehr, V. Prediction of charge and energy transport in organic crystals with quantum chemical protocols employing the hopping model. Dissertation, Würzburg, Universität Würzburg, 2015, 2015.