
Estimating the derivative coupling vector through model Hamiltonian parameters

Joshua A Kammeraad and Paul M Zimmerman

Purpose

The purpose of this document is to provide the original medium of the derivation of the derivative coupling in the model used for the accompanying paper as a more detailed, verifiable, and extensible record.

Model Hamiltonian:

```
$Assumptions = {x, y} ∈ Reals && {g, h} > 0;  
Hamiltonian[g_, h_, x_, y_] := {{g * x, h * y}, {h * y, -g * x}}  
MatrixForm[Hamiltonian[g, h, x, y]]
```

$$\begin{pmatrix} g x & h y \\ h y & -g x \end{pmatrix}$$

As has been previously described in Fdez et. al. in 2016 (see ref in main text), adiabatic energies and gradients computed using this model Hamiltonian are invariant with respect to the signs and transposition of \vec{x} and \vec{y} (although the actual values of g , h , x , and y and the mapping between model coordinates and original nuclear coordinates will vary with respect to transposition and sign). Our convention was to choose \vec{x} as the Davidson eigenvector with the largest eigenvalue and \vec{y} as the eigenvector with the second largest eigenvalue.

While the direction and magnitude of the derivative coupling vector are also invariant with respect to sign and transposition of \vec{x} and \vec{y} , the derivative coupling's sign is not. In CASSCF, the sign of the derivative coupling vector is dependent on the choice of sign of the wavefunction. This sign is arbitrary but must be chosen consistently across time in dynamics simulations to correctly model interference and population transfer. In our method, equivalent consistency should be achievable without dependence on the underlying method by consistently choosing the assignment and sign of the Davidson eigenvectors. Making the choice that maximizes corresponding eigenvector overlap between nearby molecular geometries should provide this consistency even when passing through a conical intersection (as long as only two states are involved). For example, in surface hopping overlap could be maximized between adjacent time steps.

We leave enforcement of this consistency for future work so when benchmarking our vectors we choose the sign of the Davidson derivative coupling to match that of the benchmark.

Eigenvectors and eigenvalues of the parameterized model Hamiltonian

The Hamiltonian is diagonalized to determine the normalized eigenvectors and their corresponding eigenvalues as a function of the branching plane molecular coordinates. Each eigenvector represents the coefficients of an adiabat when represented as a linear combination of the 2 diabats. Each eigenvalue represents the energy associated with the corresponding adiabat. Note the distinction between these eigenpairs of the model Hamiltonian and the eigenpairs of the ΔE^2 Hessian of the energies (Hamiltonian eigenvalues) used to fit the model.

Eigenvalues of the Hamiltonian:

```
eigenvalues[g_, h_, x_, y_] := Eigenvalues[Hamiltonian[g, h, x, y]]
eigenvalues[g, h, x, y]
```

$$\{-\sqrt{g^2 x^2 + h^2 y^2}, \sqrt{g^2 x^2 + h^2 y^2}\}$$

Corresponding normalized eigenvectors:

```
normalEigenvecs[g_, h_, x_, y_] :=
  Simplify[Normalize /@ (Eigenvectors[Hamiltonian[g, h, x, y]])]
normalEigenvecs[g, h, x, y]
```

$$\left\{ \left\{ \frac{g x - \sqrt{g^2 x^2 + h^2 y^2}}{h y \sqrt{1 + \frac{(-g x + \sqrt{g^2 x^2 + h^2 y^2})^2}{h^2 y^2}}}, \frac{1}{\sqrt{1 + \frac{(-g x + \sqrt{g^2 x^2 + h^2 y^2})^2}{h^2 y^2}}} \right\}, \right. \\ \left. \left\{ \frac{g x + \sqrt{g^2 x^2 + h^2 y^2}}{h y \sqrt{1 + \frac{(g x + \sqrt{g^2 x^2 + h^2 y^2})^2}{h^2 y^2}}}, \frac{1}{\sqrt{1 + \frac{(g x + \sqrt{g^2 x^2 + h^2 y^2})^2}{h^2 y^2}}} \right\} \right\}$$

Analytic difference gradient and derivative coupling vectors in the model

Energy difference:

```
diffEnergy[g_, h_, x_, y_] =  
  eigenvalues[g, h, x, y][[2]] - eigenvalues[g, h, x, y][[1]]
```

$$2 \sqrt{g^2 x^2 + h^2 y^2}$$

Difference gradient:

The difference gradient can be evaluated in the model by analytic differentiation of the difference in the energies (eigenvalues) of the model Hamiltonian with respect to molecular coordinates.

```
differenceGradient[g_, h_, x_, y_] = D[diffEnergy[g, h, x, y], {{x, y}}]
```

$$\left\{ \frac{2 g^2 x}{\sqrt{g^2 x^2 + h^2 y^2}}, \frac{2 h^2 y}{\sqrt{g^2 x^2 + h^2 y^2}} \right\}$$

Derivative coupling:

There is no derivative coupling in the diabatic basis so the derivative coupling can be evaluated in the model using derivatives with respect to molecular coordinates of the coefficients of the adiabats when represented in the diabatic basis. Equivalent equations are in methods section of main text.

```
derivativeCoupling[g_, h_, x_, y_] =  
  Simplify[(normalEigenvecs[g, h, x, y][[1]][[1]] *  
    D[normalEigenvecs[g, h, x, y][[2]][[1]], {{x, y}}] +  
    normalEigenvecs[g, h, x, y][[1]][[2]] *  
    D[normalEigenvecs[g, h, x, y][[2]][[2]], {{x, y}}]) *  
    (*had trouble getting Mathematica to assume g and h are positive*)  
    Sign[h] * y / Abs[y]
```

$$\left\{ -\frac{g h y}{2 (g^2 x^2 + h^2 y^2)}, \frac{g h x}{2 g^2 x^2 + 2 h^2 y^2} \right\}$$

Angle between the difference gradient and derivative coupling:

```
angle[g_, h_, x_, y_] =  
  FullSimplify[VectorAngle[differenceGradient[g, h, x, y] * Sign[g] * Sign[h],  
    derivativeCoupling[g, h, x, y]]]
```

$$\text{ArcCos}\left[\frac{(-g^2 + h^2) x y}{\sqrt{(x^2 + y^2) (g^4 x^2 + h^4 y^2)}}\right]$$

Plot of difference gradient + derivative coupling in the branching plane

To visualize the difference gradient and derivative coupling, sample values $g=2$ and $h=1$ were chosen. The corresponding analytic difference gradient and derivative coupling in the model are printed for reference.

differenceGradient[2, 1, x, y]

$$\left\{ \frac{8x}{\sqrt{4x^2 + y^2}}, \frac{2y}{\sqrt{4x^2 + y^2}} \right\}$$

derivativeCoupling[2, 1, x, y]

$$\left\{ -\frac{y}{4x^2 + y^2}, \frac{2x}{8x^2 + 2y^2} \right\}$$

```
vecPlot = VectorPlot[
  {differenceGradient[2, 1, x, y], derivativeCoupling[2, 1, x, y]}, {x, -1, 1},
  {y, -1, 1}, LabelStyle -> {Black, FontFamily -> "Georgia", FontSize -> 20},
  ImageSize -> 500, PlotRange -> All, PlotLabel ->
  "Branching plane vector behavior\nnear a conical intersection (top view)",
  PlotLegends -> Placed[{"Difference gradient", "Derivative coupling"}, Below],
  FrameLabel -> {x, y}, VectorPoints -> 5, VectorScale ->
  {0.15, Scaled[0.75], If[Abs[#1] < 0.01 && Abs[#2] < 0.01, None, #5^-.5] &}};
vecPlot = Show[vecPlot /. Arrow[{p_, q_}] -> Arrow[{p + (q - p) / 2, q + (q - p) / 2}]]
```

