

Conical Intersection Optimization Based on a Double Newton–Raphson Algorithm Using Composed Steps

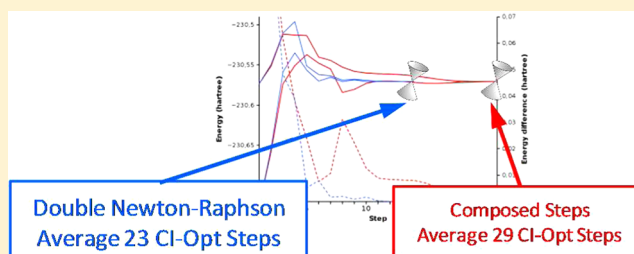
Sergi Ruiz-Barragan,[†] Michael A. Robb,[‡] and Lluís Blancafort^{*,†}

[†]Institut de Química Computacional and Department de Química, University of Girona, 17071 Girona, Spain

[‡]Department of Chemistry, Imperial College, London SW7 2AZ, United Kingdom

S Supporting Information

ABSTRACT: An algorithm for conical intersection optimization based on a double Newton–Raphson step (DNR) has been implemented and tested in 11 cases using CASSCF as the electronic structure method. The optimization is carried out in redundant coordinates, and the steps are the sum of two independent Newton–Raphson steps. The first step is carried out to reach the energy degeneracy and uses the gradient of the energy difference between the crossing states and the so-called branching space Hessian. The second step minimizes the energy in the intersection space and uses the projected excited state gradient and the intersection space Hessian. The branching and intersection space Hessians are obtained with a Broyden–Fletcher–Goldfarb–Shanno update from the gradient difference and projected excited state gradients, respectively. In some cases, mixing of the quasi-degenerate states near the seam causes changes in the direction of the gradient difference vector and induces a loss of the degeneracy. This behavior is avoided switching to a composed step (CS) algorithm [Sicilia et al. *J. Chem. Theory Comput.* **2008**, *4*, 27], i.e., a hybrid DNR-CS implementation. Compared to the composed gradient (CG) [Bearpark et al. *Chem. Phys. Lett.* **1994**, *223*, 269] and hybrid CG-CS algorithms, the DNR-CS algorithm reaches the MECI in 30% and 15% less steps, respectively. The improvement occurs mostly because the approach to the seam is more efficient, and a degeneracy threshold of 0.001 hartree is reached at lower energies than in the CG and CG-CS cases.



INTRODUCTION

Conical intersections (CIs) are very important in molecular photochemistry and photophysics.¹ They provide efficient funnels for the decay from the excited state to the ground state, or for transitions between different excited states, and therefore they play a key role in ultrafast excited state processes. They account for excited state processes of molecules in the gas phase and in solution and also of chromophores embedded in biological systems such as DNA, rhodopsin, the photoactive yellow protein, or others (see refs 2 and 3, and references therein).

In computational photochemistry, the first approach to study a photochemical process is usually the computation of critical points on the surface.⁴ In this approach, the calculation of minimum energy conical intersection (MECI) structures is particularly relevant. However, locating these points can be difficult because they often have counterintuitive structures that are difficult to predict *a priori*. Therefore, efficient algorithms to optimize these points are important. Such algorithms are also necessary for their further use in different contexts, such as the implementation in QM/MM formalisms. Although the first CI optimization algorithms were developed about two decades ago,⁵ efforts to develop new algorithms are ongoing and are directed along several directions. The first one is the search for more efficient optimization algorithms, which is the main aim of this paper.⁶ Another line is the development of approaches that do not require the interstate coupling vector (one of the two directions

that breaks the degeneracy at the intersection, see details below), which is difficult to calculate with many electronic structure methods.⁷ Finally, several methods have been developed that include environmental effects, mostly based on hybrid schemes.⁸

CIs are crossings of potential energy surfaces of the same multiplicity that form an $(N - 2)$ -dimensional subspace, usually called intersection space or seam of intersection. Here, N is the number of degrees of freedom of the surface, which is equal to $3n - 6$, where n is the number of atoms. The degeneracy is lifted in first order along the remaining two directions of the surface, which form the so-called branching space. Computationally, the problem of locating the MECI is an energy minimization subject to the constraint of energy degeneracy between the states. This is equivalent to an optimization in the $(N - 2)$ dimensional subspace. This problem can be tackled with three general methods: penalty function methods, projected gradient methods, and Lagrange–Newton ones. These methods have been compared previously.¹⁰ In this paper, we focus on projected gradient methods, and we present an improved optimization scheme of this type.

The CI minimization conditions can be described as

$$\Delta E = 0 \quad (1)$$

Received: December 3, 2012

Published: January 15, 2013



$$\mathbf{g}_{\text{IS}} = 0 \quad (2)$$

where ΔE is the energy difference between the two states and \mathbf{g}_{IS} is the so-called intersection space gradient, i.e., the projection of the gradient in the intersection space. The main idea of the projected gradient methods is to carry out a Newton–Raphson optimization driven by \mathbf{g}_{IS} . This gradient is obtained by projecting out the branching space vectors, which are given by eqs 3 and 4:

$$\mathbf{x}_1 = \frac{\partial(E_2 - E_1)}{\partial \mathbf{q}} \quad (3)$$

$$\mathbf{x}_2 = \frac{\partial(\mathbf{C}_1 | \hat{H} | \mathbf{C}_2)}{\partial \mathbf{q}} \quad (4)$$

\mathbf{x}_1 is the gradient difference vector, and \mathbf{x}_2 is the interstate coupling vector. Both vectors can be obtained routinely at the complete active space self-consistent field (CASSCF) level of theory.¹¹ E_i is the adiabatic energy of state i , H is the electronic part of the Hamiltonian, and \mathbf{C}_1 and \mathbf{C}_2 are the CI eigenvectors of the MC-SCF problem. The resulting projector is

$$\mathbf{P} = \mathbf{I} - \hat{\mathbf{x}}_1 \hat{\mathbf{x}}_1^T - \hat{\mathbf{x}}_2 \hat{\mathbf{x}}_2^T \quad (5)$$

We follow the usual notation, where $\hat{\mathbf{x}}$ stands for the normalized vector \mathbf{x} . \mathbf{x}_1 and \mathbf{x}_2 are orthogonalized for the projection. The intersection space gradient \mathbf{g}_{IS} is obtained applying the projector to the gradient of either of the degenerate states (both projections give the same result):

$$\mathbf{g}_{\text{IS}} = \mathbf{P} \frac{\partial E_2}{\partial \mathbf{q}} = \mathbf{P} \frac{\partial E_1}{\partial \mathbf{q}} \quad (6)$$

The main idea behind projection methods is that once the energy degeneracy condition is fulfilled, optimization steps along \mathbf{x}_1 and \mathbf{x}_2 should be avoided because they may cause a loss of the energy degeneracy. This is taken care of (to first order) by construction, since the intersection space gradient of eq 6 is orthogonal to \mathbf{x}_1 and \mathbf{x}_2 .

Projected gradient methods are more efficient than penalty function ones because the projected gradient helps to maintain the energy degeneracy during the optimization. Their practical advantage with respect to Lagrange–Newton methods is that they can be easily implemented into available optimization routines. Here, we focus on the combination of projected gradient algorithms with a Newton–Raphson optimization in redundant coordinates¹² and Broyden–Fletcher–Goldfarb–Shanno (BFGS) update of the Hessian.¹³ One important issue is how the energy degeneracy condition is implemented, and this is the main point of our paper. The first possibility is to work with a composed gradient (CG) \mathbf{g}_c of the form^{5b}

$$\mathbf{g}_c = \mathbf{g}_{\text{IS}} + \mathbf{f} \quad (7)$$

where

$$\mathbf{f} = 2\Delta E \hat{\mathbf{x}}_1 \quad (8)$$

In eq 7, \mathbf{g}_{IS} and \mathbf{f} are orthogonal, and \mathbf{g}_c only becomes zero when both components are zero, i.e., when the two minimization conditions of eqs 1 and 2 are fulfilled. When the degeneracy is reached, the \mathbf{f} component is zero, and steps along the branching space, which might lift the degeneracy, should be avoided. However, the CG algorithm has the drawback that the gradient is not well suited for the Hessian update, since it is made of components in the branching space and the inter-

section space. The branching space components act on the updated Hessian. In the Newton–Raphson step, this can lead to displacements in the branching space even when \mathbf{f} is zero, lifting the degeneracy.

To circumvent this problem, we have recently introduced a method which uses a composed step (CS) instead of a CG.^{6c} To understand the performance of the different algorithms, it is useful to think of the optimization as a process made of two main phases: in the first phase the energy gap is large, and the approach to the energy degeneracy dominates the optimization. The optimization in the intersection space is carried out in the second phase, once the degeneracy is reached. As we explain more in detail below, in the CS algorithm the Newton–Raphson step is only carried out in the intersection space, and the energy degeneracy condition is approached with a linear step along the gradient difference vector \mathbf{x}_1 . This improves the optimization in the intersection space, but the approach to the intersection space is more efficient in the CG case. For that reason, we proposed a hybrid CG-CS algorithm which starts with the CG method and switches to the CS one when the energy degeneracy is reached.^{6c}

In the present paper, we introduce a new projection based algorithm with a CS where the approach to the seam is improved. The so-called double Newton–Raphson (DNR) algorithm does two Newton–Raphson steps in every optimization cycle. Thus, compared to the CS algorithm the linear step along \mathbf{x}_1 is replaced with a quadratic step. The three methods (CG, hybrid CG-CS, and DNR) are compared for several cases, and the results show that the optimization is improved with a hybrid DNR-CS algorithm.

Theoretical Basis. The CS Algorithm and Its Implementation. The main idea of the CS algorithm is to carry out displacements composed of two independent steps, each one following the conditions of eqs 1 and 2. We follow the derivation of ref 6c, which is related to the derivations presented in refs 5c and 6b. To reach the energy degeneracy from a starting point \mathbf{q}_0 , the energy difference is expanded in a Taylor series to first order:

$$\Delta E(\mathbf{q}_0 + \Delta \mathbf{q}^{\text{BS}}) = \Delta E^0 + \Delta \mathbf{q}_{\text{BS}}^T \mathbf{x}_1 \quad (9)$$

where ΔE^0 is the energy gap at \mathbf{q}_0 . The displacement that leads to energy degeneracy is parallel to \mathbf{x}_1 and is called the branching space displacement:

$$\Delta \mathbf{q}_{\text{BS}} = -\frac{\Delta E^0}{\delta \kappa^0} \hat{\mathbf{x}}_1 \quad (10)$$

$\delta \kappa^0$ is the length of \mathbf{x}_1 at \mathbf{q}_0 . The energy minimization in the intersection space is achieved with a Newton–Raphson procedure using the intersection space gradient:

$$\Delta \mathbf{q}_{\text{IS}} = -(\tilde{\mathbf{H}}_{\text{IS}})^{-1} \mathbf{g}_{\text{IS}} \quad (11)$$

In eq 11, $\tilde{\mathbf{H}}_{\text{IS}}$ is derived from the so-called intersection space Hessian, \mathbf{H}_{IS} . The intersection space Hessian is the second derivative matrix of the seam energy.¹⁴ Here, it is obtained from \mathbf{g}_{IS} with the BFGS update (see details in the Implementation section). The optimization step is the sum of the intersection and branching space steps:

$$\Delta \mathbf{q} = \Delta \mathbf{q}_{\text{BS}} + \Delta \mathbf{q}_{\text{IS}} \quad (12)$$

Once the degeneracy is reached, the convergence with the CS algorithm is improved with respect to the CG one. However the CS has a poor performance when the seam is approached because the steps are forced to go along the \mathbf{x}_1 direction. As a result, the initial part of the optimization, which is dominated

by the approach to the seam, reaches the degeneracy in more steps and leads to regions of the seam with higher energy. To solve this problem, in our previous work we implemented a hybrid algorithm that combines the best of the two algorithms: The optimization starts with the CG algorithm until the energy degeneracy is reached, i.e., when the energy gap between the two states falls below a threshold of 0.005 hartree. After the threshold, the optimization switches to the CS procedure to optimize the energy in the intersection space. We refer to this implementation as the hybrid CG-CS one.

The DNR Algorithm. The main motivation of the DNR algorithm is to improve the approach to the seam. The idea to do this is borrowed from the CG algorithm, where the energy degeneracy condition (eq 1) is reached *via* the gradient component \mathbf{f} (eq 8). In its turn, \mathbf{f} is derived from the condition of minimizing $(\Delta E)^2$. Thus, we can express the DNR algorithm as the minimization of two functions, E_{IS} and $(\Delta E)^2$. Each function is optimized with a Newton–Raphson step, and the optimization step is the sum of the two. Therefore, the working equations for the intersection space step and the total optimization step are the same ones as in the CS case (eqs 11 and 12). The branching space step is given by

$$\Delta \mathbf{q}_{\text{BS}} = (\mathbf{H}_{\text{BS}})^{-1} \mathbf{g}_{\text{BS}} \quad (13)$$

and \mathbf{g}_{BS} is equal to \mathbf{f} in eq 8:

$$\mathbf{g}_{\text{BS}} = \frac{1}{\delta \kappa^0} \frac{\partial (\Delta E^2)}{\partial \mathbf{q}} = 2 \Delta E \hat{\mathbf{x}}_1 \quad (14)$$

Similar to the intersection space Hessian, the branching space Hessian \mathbf{H}_{BS} is obtained from \mathbf{g}_{BS} with the BGFS update. Strictly speaking, the gradient for minimization of $(\Delta E)^2$ is $2 \Delta E \hat{\mathbf{x}}_1$. However, in eq 14 we follow the original derivation of the CG algorithm, and we divide the gradient by the length of \mathbf{x}_1 , which greatly improves the convergence.

Implementation. The DNR algorithm has been implemented in the Gaussian Development Version program.¹⁵ Several practical issues deserve comment. The first one is related with the use of redundant coordinates for optimization, which requires a transformation of the gradient from Cartesian to redundant coordinates. If the transformation to redundant coordinates is done on \mathbf{g}_{IS} , after the projection step (eq 6), the orthogonality between the gradient vectors \mathbf{g}_{IS} and \mathbf{x}_1 is lost during the transformation. Thus, the transformation of a gradient from Cartesian to redundant coordinates (\mathbf{x}^{cart} to \mathbf{x}^{red}) is given by eq 15:^{12,16}

$$\mathbf{x}^{\text{red}} = \mathbf{G}^{-1} \mathbf{B} \mathbf{x}^{\text{cart}} \quad (15)$$

where \mathbf{B} is the Wilson matrix¹⁷ and \mathbf{G}^{-1} is the generalized inverse of $\mathbf{G} = \mathbf{B} \mathbf{B}^T$. Before the transformation, $\mathbf{g}_{\text{IS}}^{\text{cart}}$ and $\mathbf{x}_1^{\text{cart}}$ (both in Cartesians) are orthogonal. After the transformation to redundant coordinates, the scalar product is

$$(\mathbf{g}_{\text{IS}}^{\text{red}})^T \mathbf{x}_1^{\text{red}} = (\mathbf{g}_{\text{IS}}^{\text{cart}})^T \mathbf{B}^T (\mathbf{G}^{-1})^T \mathbf{G}^{-1} \mathbf{B} \mathbf{x}_1^{\text{cart}} \quad (16)$$

For the orthogonality to be retained in redundant coordinates, the product $\mathbf{B}^T (\mathbf{G}^{-1})^T \mathbf{G}^{-1} \mathbf{B}$ should be the unity matrix, which is not the case. The loss of orthogonality of \mathbf{g}_{IS} with respect to the branching space vectors is similar to what happens when geometry constraints are applied to the optimization gradient in the CG case.¹⁸ Similar to this case, it is better to transform \mathbf{g}_{IS} , \mathbf{x}_1 , and \mathbf{x}_2 to redundant coordinates before the projection and do the projection in redundant coordinates. This procedure, which is different from the current implementation of the CG

algorithm in Gaussian,¹⁹ has been adopted here for the three investigated algorithms. The back transformation from redundant to Cartesian coordinates does not suffer from this problem because there one wants to maintain orthogonality between the $\Delta \mathbf{q}_{\text{IS}}$ displacement and the \mathbf{x}_1 and \mathbf{x}_2 gradient vectors. In this case, the back transformation for the gradient vectors is

$$\mathbf{x}^{\text{cart}} = \mathbf{B}^T \mathbf{x}^{\text{red}} \quad (17)$$

In its turn, the back transformation for the intersection space displacement can be approximated as^{12,16}

$$\Delta \mathbf{q}^{\text{cart}} = \mathbf{B}^T \mathbf{G}^{-1} \Delta \mathbf{q}^{\text{red}} \quad (18)$$

Thus, the scalar products between $\Delta \mathbf{q}_{\text{IS}}$ and \mathbf{x}_1 and \mathbf{x}_2 , respectively, are not affected by the transformation, and the orthogonality is retained.

The second issue is related with the construction of $\tilde{\mathbf{H}}_{\text{IS}}$ (eq 11) from \mathbf{H}_{IS} . Thus, \mathbf{H}_{IS} can be defined as the derivative of the intersection space gradient:

$$\mathbf{H}_{\text{IS}} = \frac{\partial}{\partial \mathbf{q}} \mathbf{g}_{\text{IS}} \quad (19)$$

\mathbf{H}_{IS} is obtained from \mathbf{g}_{IS} following the BGFS update.^{6c} To avoid $\Delta \mathbf{q}_{\text{IS}}$ components along \mathbf{x}_1 and \mathbf{x}_2 , the intersection space Hessian should have zero eigenvalues along these directions. However, this is not necessarily the case because the directions of these vectors change during the optimization, and \mathbf{H}_{IS} is obtained from an update. This can be corrected multiplying \mathbf{H}_{IS} from the left and right by \mathbf{P} . The resulting matrix has two zero eigenvalues corresponding to the directions of \mathbf{x}_1 and \mathbf{x}_2 . To avoid problems with the inversion in the Newton–Raphson step (eq 11), these eigenvalues are increased artificially to a high value, following the idea of ref 12. Thus, the final matrix used for the Newton–Raphson step is

$$\tilde{\mathbf{H}}_{\text{IS}} = \mathbf{P} \mathbf{H}_{\text{IS}} \mathbf{P} + (\mathbf{I} - \mathbf{P}) \mathbf{A} (\mathbf{I} - \mathbf{P}) \quad (20)$$

\mathbf{A} is a diagonal matrix whose elements are set to a large constant (5000 hartree-bohr⁻² in the present case). Note that before the step in redundant coordinates is calculated (eq 11), $\tilde{\mathbf{H}}_{\text{IS}}$ from eq 20 is further modified by the optimization routine implemented in Gaussian; i.e., displacements in the redundant part of the coordinate space are avoided as indicated in ref 12. The construction of the intersection space step (eq 11) is similar to what is proposed in eq 13 of ref 5c, although the Hessian is different. In ref 5c, the Newton–Raphson procedure is carried out with the so-called reduced Hessian matrix, which has dimension $N - 2$; here, we use redundant coordinates, and instead of reducing the dimension of the matrix, we find it more convenient to use $\tilde{\mathbf{H}}_{\text{IS}}$ and treat the zero eigenvalues along \mathbf{x}_1 and \mathbf{x}_2 analogously to those along the redundant part of the coordinate space.

The branching space Hessian has also some characteristics that have to be treated with care. First, in any molecule there are many coordinates that do not affect the energy difference. This results in a large number of near zero eigenvalues in \mathbf{H}_{BS} , which in turn can cause problematic $\Delta \mathbf{q}_{\text{BS}}$ steps, namely displacements along coordinates that do not affect the energy gap. To avoid this problem, we have followed the idea of eq 20 and increased artificially all eigenvalues of \mathbf{H}_{BS} smaller than a threshold of 2×10^{-4} hartree-bohr⁻². Another problem is caused by the tendency of the electronic states to mix near the degeneracy. The mixture of the states interchanges the GD and IC vectors \mathbf{x}_1 and \mathbf{x}_2 , causing arbitrary changes in the direction

of \mathbf{x}_1 . These changes affect \mathbf{g}_{BS} (see eq 14), and more importantly, they affect \mathbf{H}_{BS} through the update formula. In turn, this results in poor $\Delta\mathbf{q}_{BS}$ steps and a loss of the degeneracy near the seam. Moreover, this effect extends to further optimization steps because it affects \mathbf{H}_{BS} . A way to mitigate this problem is to switch from the DNR to the CS procedure, which is equivalent to changing from a quadratic to a linear branching space optimization step. We have followed a practical approach, based on energy criteria, to detect this problem and carry out the switch. Thus, the change comes into effect when there is a sudden rise in the energy gap near the seam during the optimization, i.e., when the energy gap lies below a threshold of 0.005 hartree, and it increases by 0.010 hartree or more from one step to the next one. We refer to this as the hybrid DNR-CS implementation.

The algorithms have been combined with the usual optimization features in Gaussian.¹⁹ The initial Hessian for \mathbf{H}_{BS} and \mathbf{H}_{IS} is the parametrized, diagonal Hessian in redundant coordinates used for default optimizations, which gives satisfactory results. The convergence is checked against the forces in \mathbf{g}_{IS} and the total displacement $\Delta\mathbf{q}$.

COMPUTATIONAL DETAILS

The three algorithms have been tested on a test set of 11 cases. The structures are shown in Figure 1, and the Cartesian coordinates

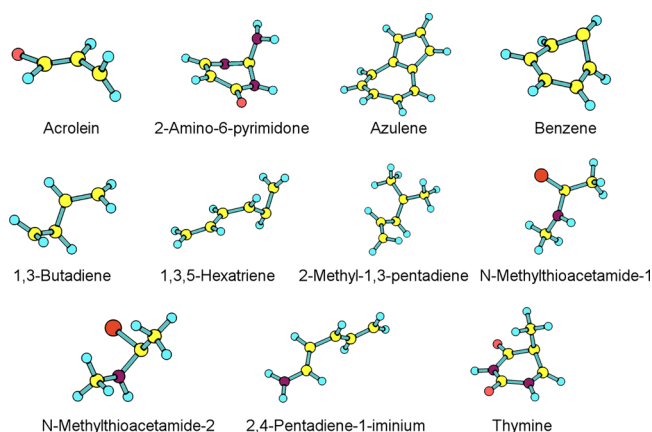


Figure 1. MECI test set used in this work.

are given in the Supporting Information. Four molecules (azulene, benzene, butadiene, and the penta-2,4-dien-1-iminium cation) were previously used in a test set for the comparison of penalty function, projection-based, and Lagrange–Newton MECI optimization algorithms;¹⁰ our test set also includes acrolein and hexatriene, a nucleobase (thymine), and the guanine analogue 2-amino-6-pyrimidone. We also have included three more challenging cases, a floppy butadiene derivative with two additional methyl groups (4-methyl-1,3-pentadiene) and two local MECI minima of *N*-methylthioacetamide which could not be fully optimized in a preceding study with the CG algorithm.²⁰ Most optimizations are started near the excited state minimum, releasing any symmetry constraints. In the cases where several local MECI minima are known on the seam, the molecule has been distorted to drive the optimization toward the global minimum. The coordinates of the starting points are also included in the Supporting Information.

The calculations have been carried out at the state-averaged CASSCF/6-31G** level of theory, and the active spaces are detailed in the Supporting Information. In all optimizations, except for azulene, the gradients include the coupled-perturbed

multiconfiguration self-consistent correction,^{11b} which is only available for active spaces up to eight orbitals. The active space for azulene has 10 active orbitals, and in this case we have used the approximate gradients without the correction.

RESULTS AND DISCUSSION

The results of the MECI searches with the three algorithms are presented in Table 1. We start discussing two representative

Table 1. Results of MECI Searches with the CG, CG-CS, and DNR Algorithms, Including the Distance from the Starting Point to the MECI, the Number of Optimization Cycles, and the Energy Gap at the Optimized Point

test molecule	distance [Å] ^a	method	cycles	ΔE [au] ^b
acrolein	1.70	CG	100 ^c	
		CG-CS	49	0.0000154
		DNR-CS	31	0.0000002
2-amino-6-pyrimidone	1.50	CG	15	0.0000963
		CG-CS	15	0.0000102
		DNR-CS	18	0.0000029
azulene	1.41	CG	15	0.0001397
		CG-CS	16	0.0000465
		DNR-CS	21 (48) ^d	0.0000532
benzene	1.51	CG	21	0.0000331
		CG-CS	21	0.0000002
		DNR-CS	14	0.0000106
1,3-butadiene	2.27 (2.56) ^d	CG	35	0.0001370
		CG-CS	35	0.0000002
		DNR-CS	33 (35) ^e	0.0000001
1,3,5-hexatriene	2.64 (4.21) ^f	CG	33	0.0001113
		CG-CS	29	0.0000003
		DNR-CS	29 (46) ^e	0.0000026
4-methyl-1,3-pentadiene	2.49	CG	40	0.0000705
		CG-CS	37	0.0000002
		DNR-CS	26	0.0000014
<i>N</i> -methylthioacetamide-1 ^g	2.23	CG	100 ^c	
		CG-CS	20	0.0000794
		DNR-CS	20	0.0000175
<i>N</i> -methylthioacetamide-2 ^h	2.36	CG	24	0.0001237
		CG-CS	15	0.0000004
		DNR-CS	17	0.0000078
2,4-pentadiene-1-iminium	2.97	CG	49	0.0000218
		CG-CS	66	0.0000025
		DNR-CS	28	0.0000027
thymine	2.03	CG	15	0.0000886
		CG-CS	15	0.0000006
		DNR-CS	15	0.0000097
average	2.10	CG	41	0.0000913
		CG-CS	29	0.0000142
		DNR-CS	23	0.0000099

^aDistance in Å from the starting point to the converged MECI geometry, calculated in Cartesian coordinates. ^bEnergy difference between the ground and excited states at the end of the optimization.

^cNo convergence reached after the maximum number of 100 optimization steps. ^dIn parentheses, distance to the local MECI obtained in the CG and CG-CS optimizations, see text. ^eIn parentheses, number of steps with the pure DNR algorithm. ^fIn parentheses, distance to the local MECI obtained in the CG optimizations, see text. ^gCI-A1 in ref 20.

^hCI-A2 in ref 20.

cases, for which the energy profiles during the optimizations are shown in Figures 2 and 3. In these plots, the energies of S_1

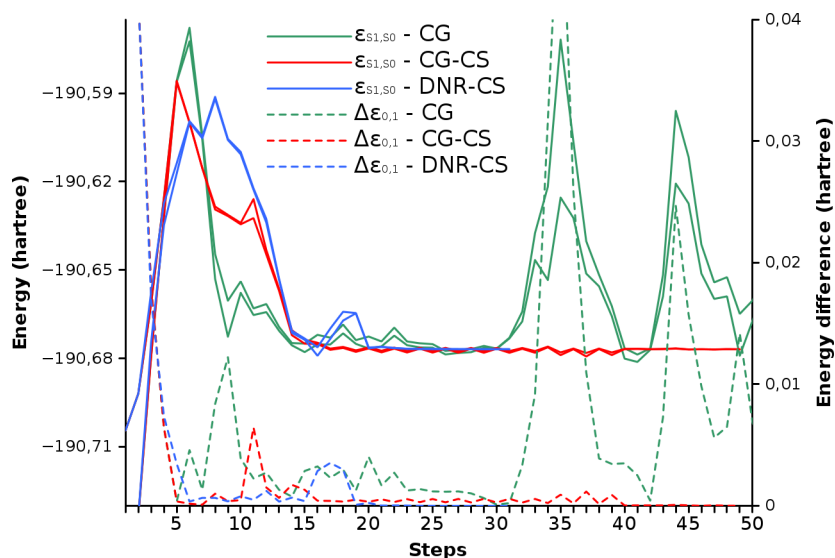


Figure 2. Course of MECI searches with the CG, CG-CS, and DNR-CS algorithms for acrolein (green, red, and blue lines, respectively). Full lines, S_1 and S_0 energies (left y axis); hashed lines, S_1-S_0 energy difference (right y axis). Energies in hartree.

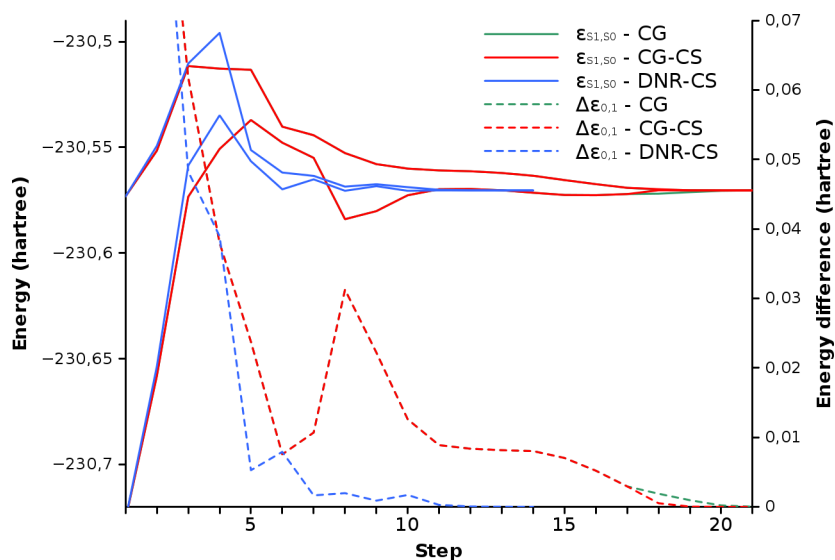


Figure 3. Course of MECI searches with the CG, CG-CS, and DNR-CS algorithms for benzene (green, red, and blue lines, respectively). Full lines, S_1 and S_0 energies (left y axis); hashed lines, S_1-S_0 energy difference (right y axis). Energies in hartree.

and S_0 are presented with full lines, and the energy gap during the optimization with hashed lines (scales on the right and left y axes, respectively). Note that the first phase of the optimization is always identical for the CG and CG-CS algorithms by construction.

The first example (Figure 2) is the optimization of the conical intersection of acrolein. This case exemplifies the loss of degeneracy with the CG algorithm because of the incorrect Hessian update (green curves). In Figure 2, we plot the first 50 steps of the optimization. In the CG case, the degeneracy is reached but lost again in steps 32 and 43 of the run (see the hashed green line, representing the S_1-S_0 energy difference). The rest of the optimization follows a similar course (not shown in Figure 2), and convergence is not reached after the maximum number of 100 optimization steps. In contrast, the CG-CS and DNR algorithms (red and blue curves, respectively) converge after 49 and 31 steps, respectively (see also Table 1). The second example is benzene (Figure 3), where the CG and CG-CS algorithms have a very similar course and

converge in 21 steps (green and red curves, respectively; the two curves are superimposed during most of the run, and the green curve is barely visible). In its turn, the DNR-CS algorithm converges in only 16 steps. In this case, the approach to the seam is more efficient. Thus, the three algorithms approach the seam in a similar number of steps, and the optimizations reach an energy gap of less than 0.01 hartree after 6 steps. However, the DNR-CS algorithm reaches the seam with a smaller absolute energy, and the MECI is reached in a smaller number of total steps.

In Figure 4a and b, we illustrate the effect of state mixing near the seam on optimizations with the DNR algorithm (azulene MECI optimization). In Figure 4a, we compare the optimizations with the pure DNR algorithm and the hybrid DNR-CS implementation (blue and red curves, respectively). In the DNR case, the seam is reached after nine steps, but the degeneracy is lost and recovered repeatedly, as shown by the oscillations of the energy difference (blue hashed line). The oscillations toward and away from the seam are due to repeated changes in the direction of \mathbf{x}_1 ,

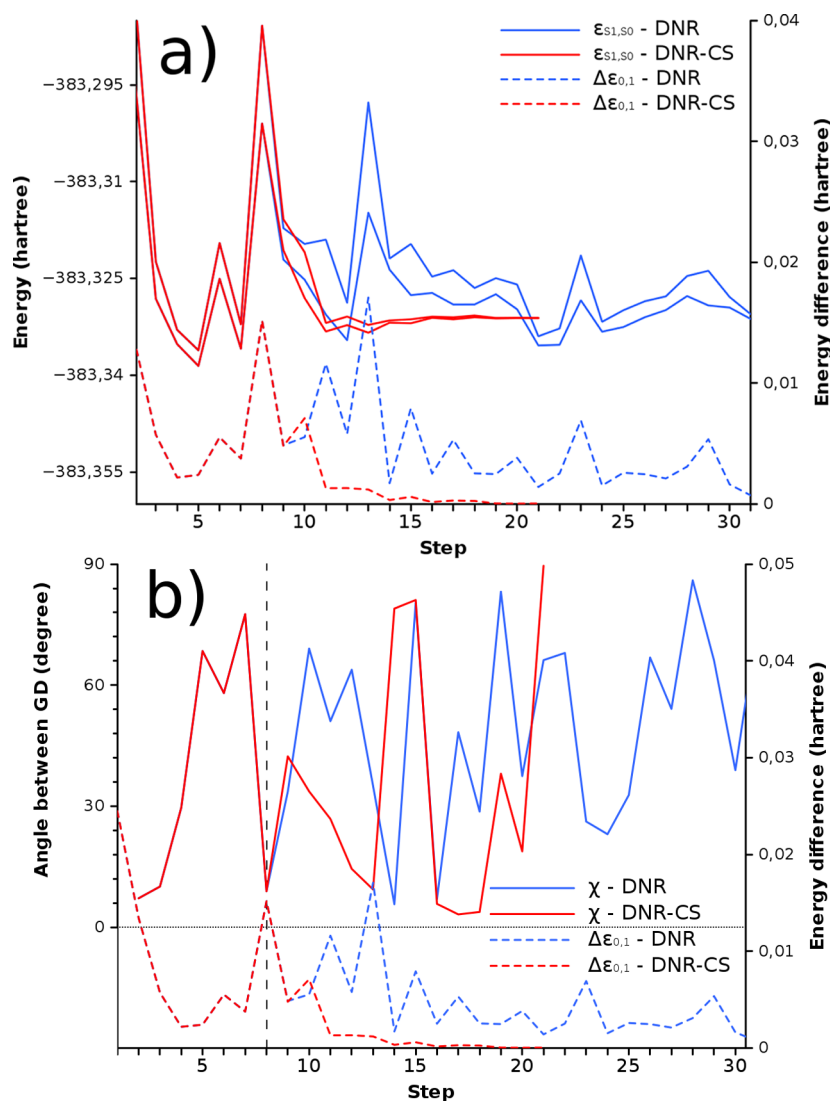


Figure 4. Course of MECI searches with the DNR algorithms for azulene. Blue lines, pure DNR algorithm; red lines, hybrid DNR-CS implementation. (a) Energies and energy difference. Full lines, S_1 and S_0 energies (left y axis); hashed lines, S_1-S_0 energy difference (right y axis). (b) Angle χ between \mathbf{x}_1 in two consecutive steps (left y axis) and S_1-S_0 energy difference (right y axis). Energies in hartree.

due to the mixing of the states near the seam (see the discussion of the previous section). This is shown in Figure 4b, where we show the evolution of the angle χ , formed by \mathbf{x}_1 in two consecutive steps. Large values of this angle correspond to large changes in the direction of \mathbf{x}_1 . This occurs, for example, in step 7 of the DNR optimization and leads to a sudden increase of the energy difference in the following step 8. The oscillations are avoided in the DNR-CS implementation, switching to the CS algorithm when the first increase in the energy gap near the seam is detected (step 9 in this case). This reduces the number of optimization steps from 48 to 21. Note that the changes in the direction of \mathbf{x}_1 become irrelevant when the energy gap is very small, because the branching space steps are multiplied by the energy difference (eqs 13 and 14). The case of hexatriene is similar to that of azulene, since the pure DNR algorithm requires 46 steps to reach the MECI and the hybrid DNR-CS implementation needs only 29 steps. The switch to the CS algorithm has also entered into function in the butadiene case, but here the improvement in the number of steps (33 instead of 35 with the pure DNR algorithm; the change from the DNR to the CS step takes place after step 12) is less significant.

Next, we turn our attention to two cases where the three algorithms lead to different local minima on the seam. This is the case for the floppy molecules butadiene (Figure 5) and hexatriene (Figure 6). In both cases, the DNR-CS algorithm finds the global MECI minimum. First, we consider butadiene. Butadiene has several local MECI minima.^{14,21} The global minimum of the intersection space has a *s-transoid* geometry where the $C_1-C_2-C_3-C_4$ dihedral angle φ_1 is 118.5° , and one of the methyl groups is rotated by 106° (value of the $H_5-C_1-C_2-C_3$ angle φ_2 ; Figure 5). The MECI searches start at a *transoid* structure ($\varphi_1 = 180^\circ$) where the two double bonds are twisted from planarity by 90° ($\varphi_2 = 90^\circ$). The hybrid DNR-CS algorithm finds the global minimum in 33 steps, whereas the CG-CS and CG algorithms find a different local minimum, which also has a *transoid* conformation ($\varphi_1 = 113.9^\circ$) but a methylene twist angle $\varphi_2 = 53^\circ$. The local minimum is 0.27 eV higher in energy and has not been described previously. In Figure 5, we plot the changes in φ_2 and the S_1-S_0 energy difference during the optimizations to illustrate how the DNR-CS and CG-CS algorithms approach the seam. In the DNR-CS case (blue curves), the algorithm finds the seam after seven

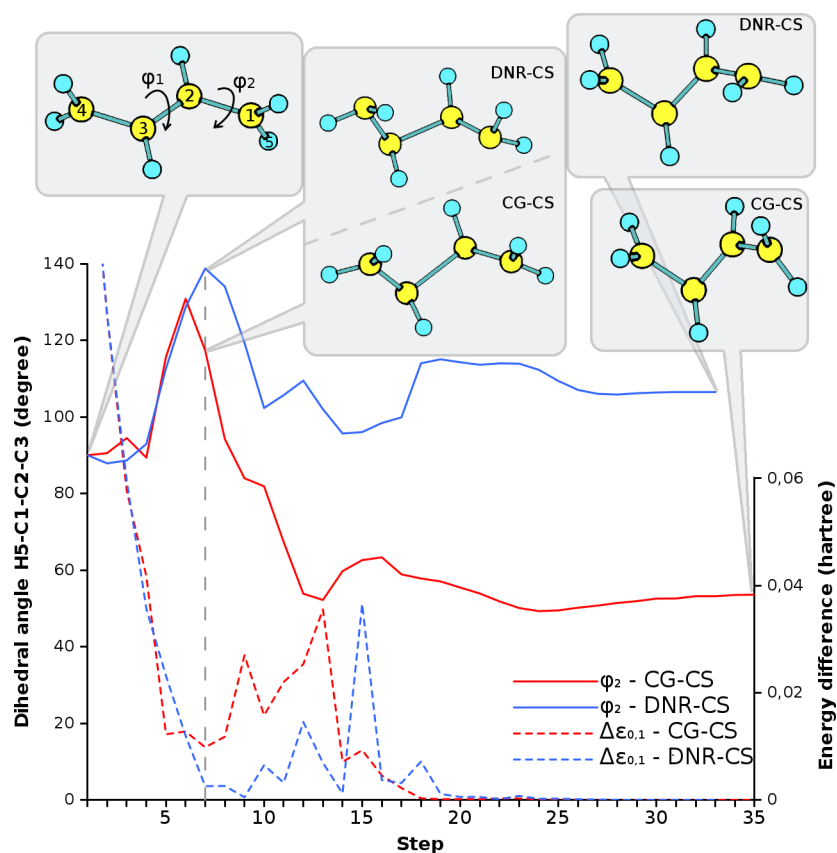


Figure 5. Course of MECI searches with the CG-CS and DNR-CS algorithms for butadiene, leading to different local MECI minima (blue and red lines, respectively). Full lines: $H_5-C_1-C_2-C_3$ dihedral angle ϕ_2 . Dashed lines: S_1-S_0 energy difference.

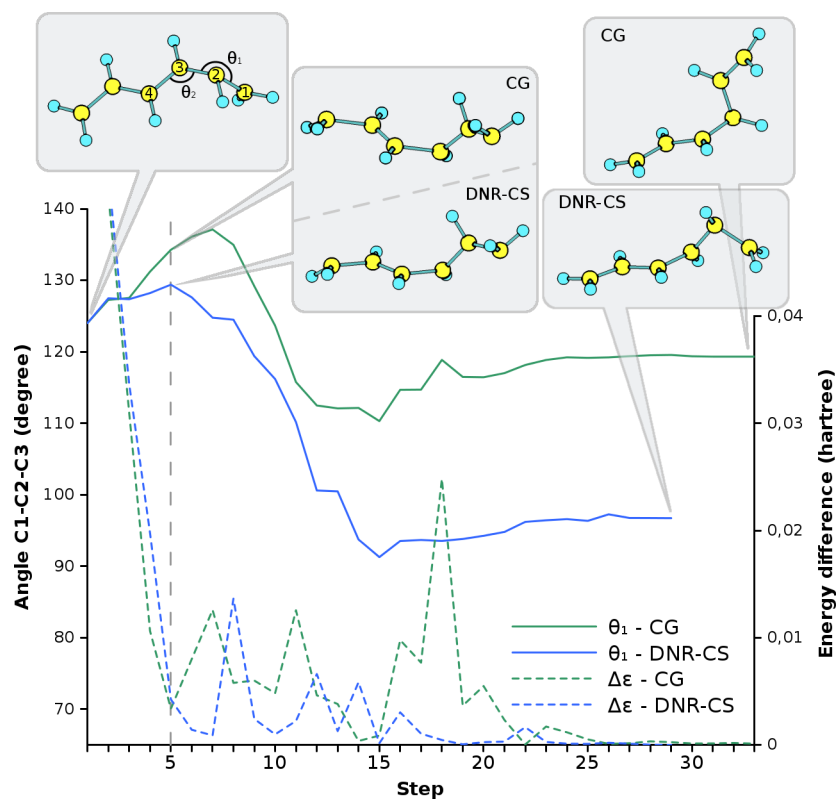


Figure 6. Course of MECI searches with the CG and DNR-CS algorithms for hexadiene, leading to different local MECI minima (green and blue lines, respectively). Full lines: $C_1-C_2-C_3$ bending angle θ_2 . Dashed lines: S_1-S_0 energy difference.

steps, at an angle φ_2 of approximately 130° , and the value of φ_2 decreases in a few more steps until it finds its approximate final value. In contrast to this, the CG and CG-CS (red curves) need more steps to reach the seam and do it at a structure where φ_2 is approximately 60° (step 18).

For hexatriene (Figure 6), the global MECI is characterized by a “kink” at C_2 (C_1 – C_2 – C_3 bending angle $\theta_1 = 96.7^\circ$), and there is a local MECI with a “kink” at C_3 (C_2 – C_3 – C_4 bending angle $\theta_2 = 98.6^\circ$) that lies 0.03 eV higher.²² The MECI search starts at a structure with a twisted C_1 – C_2 – C_3 moiety (see Figure 6), and the structure found by the DNR-CS and CG-CS algorithms is the global MECI. In contrast to this, the CG algorithm leads to the local minimum on the seam. In this case, the CG and CG-CS algorithms find the seam at a structure with long C–C bonds (see the structures at step 5 in Figure 6). However, the energy optimization with the CG algorithm leads to the local MECI with the kink localized on C_3 (green curves), while that with the CG-CS one finds the global minimum. In comparison, the structure where the DNR-CS algorithm hits the seam has shorter bonds and lower energy, and the energy optimization leads to the global minimum (blue curves).

Overall, the hybrid DNR-CS is the most efficient of the tested algorithms. It reaches the MECI in approximately 15% less steps than the hybrid CG-CS one and 30% less steps than the CG algorithm. The CG-CS and DNR-CS algorithms also give a better convergence with respect to the energy degeneracy. In the CG case, the gradient \mathbf{g}_c near the MECI is dominated by \mathbf{g}_{IS} (eq 7), and the convergence with respect to the energy degeneracy is not strict. However in the CG-CS and DNR-CS algorithms, the energy degeneracy convergence is controlled through the convergence of the displacement size, and this results in smaller energy gaps. More insight into the performance of the different algorithms can be obtained by examining the number of steps required to reach the degeneracy and the energy at the point where the seam is reached. Thus, Table 2 presents the number of steps required to reach a degeneracy threshold of 0.005 and 0.001 hartree, together with the energy of S_1 relative to the MECI at those points. This gives an idea of how close the optimization is to the MECI when it reaches the seam. On average, the DNR-CS algorithm needs a few less steps than the CG and CG-CS algorithms to reach an energy difference smaller than 0.005 hartree (seven steps against 10, respectively). The smaller energy gap of 0.001 hartree is reached in the same number of steps with the DNR-CS and CG-CS algorithms (10 in both cases), but the DNR-CS algorithm reaches the degeneracy closer to the MECI: on average the point of 0.001 hartree degeneracy is reached 0.019 hartree above the MECI with the DNR algorithm, while the CG-CS algorithm reaches such a degeneracy 0.030 hartree above the MECI. Overall, these data illustrate the better efficiency of the hybrid DNR-CS implementation. At the same time, the optimization details show that the course of the optimization can differ widely from case to case: thus, in several cases the 0.001 degeneracy threshold is reached very close to the MECI (for example, for benzene, azulene, or 2-amino-6-pyrimidone), whereas in other cases the seam is reached well above the MECI, such as in the acrolein, hexatriene, and butadiene cases. Moreover, the course of the optimizations also depends on the starting point. Here, most optimizations have been started near the excited state minimum, which usually lies far away from the intersection, to reproduce applications of the algorithms where the MECI structure is not known. In this case, the projected intersection space gradient \mathbf{g}_{IS} at the S_1 minimum is zero, and the first optimization step is defined

Table 2. Number of Steps Required to Reach Degeneracy Thresholds of 0.005 au and 0.001 au and Energy of S_1 at Those Points, Relative to the MECI, for the 11 Test Cases with the CG, CG-CS, and DNR-CS Algorithms

test molecule	method	threshold 0.005 au		threshold 0.001 au	
		steps ^a	E_{rel} ^b [au]	steps ^a	E_{rel} ^b [au]
acrolein	CG	5	0.0911	5	0.0911
	CG-CS	5	0.0911	5	0.0911
	DNR-CS	5	0.0631	6	0.0773
2-amino-6-pyrimidone	CG	5	−0.0064	6	−0.0038
	CG-CS	5	−0.0064	7	−0.0026
	DNR-CS	9	−0.0071	11	−0.0005
azulene	CG	7	0.0012	11	−0.0008
	CG-CS	7	0.0011	8	0.0005
	DNR-CS	4	0.0019	14	0.0004
benzene	CG	17	−0.0012	19	−0.0001
	CG-CS	17	0.0012	18	0.0006
	DNR-CS	7	0.0069	9	0.0029
1,3-butadiene	CG	16	−0.0047	18	−0.0007
	CG-CS	16	−0.0048	18	−0.0008
	DNR-CS	7	−0.1165	9	−0.0584
1,3,5-hexatriene	CG	5	−0.0994	14	−0.0131
	CG-CS	5	−0.1044	6	−0.0751
	DNR-CS	5	−0.0684	7	−0.0501
4-methyl-1,3-pentadiene	CG	12	−0.0203	24	−0.0005
	CG-CS	12	−0.0204	14	−0.0075
	DNR-CS	8	−0.0518	14	−0.0136
N-methylthioacetamide-1 ^c	CG	4	−0.0067	31	−0.0011
	CG-CS	4	−0.0067	6	−0.003
	DNR-CS	5	−0.0243	13	−0.0001
N-methylthioacetamide-2 ^d	CG	5	−0.0143	5	−0.0143
	CG-CS	5	−0.0144	5	−0.0144
	DNR-CS	5	−0.0162	5	−0.0005
2,4-pentadiene-1-iminium	CG	15	−0.1698	45	−0.0003
	CG-CS	15	−0.1699	19	−0.1354
	DNR-CS	13	−0.0631	17	−0.0094
thymine	CG	7	−0.0056	12	−0.0002
	CG-CS	7	−0.0056	8	−0.0014
	DNR-CS	7	−0.0006	8	−0.0002
average	CG	9	0.0383	17	0.0115
	CG-CS	9	0.0387	10	0.0302
	DNR-CS	7	0.0381	10	0.0194

^aOptimization steps required to reach energy degeneracy within the given threshold. ^b S_1 energy at those points, relative to the MECI.

^cCI-A1 in ref 20. ^dCI-A2 in ref 20.

solely by the gradient difference component (eqs 8 and 14). However the optimization leaves the S_1 minimum region quickly, and \mathbf{g}_{IS} increases. Therefore, overall our starting point choice should not bias the optimization course in favor of any of the three algorithms.

CONCLUSIONS

Our comparison of the CG, CG-CS, and DNR-CS algorithms shows that the newly implemented DNR-CS algorithm reduces the number of steps in MECI optimizations substantially. In this algorithm, the optimization step is the sum of an intersection space step that minimizes the energy, and a branching space the leads to the degeneracy. A separate Hessian is updated for every step, with negligible additional computational cost compared to the energy and gradient calculations. Similar to

the CG-CS case, the use of the intersection space Hessian in the DNR algorithm avoids losing the degeneracy during the optimization, which is the main drawback of the CG algorithm. On the other hand, the branching space Hessian improves the approach to the seam compared to the CG and CG-CS cases, by taking second order degeneracy lifting effects into account. Such effects change the direction of the gradient difference vector \mathbf{x}_1 along the surface, since a second order degeneracy lifting component at one point of the surface can contribute at a different point to lift the degeneracy at first order.²³ The branching space Hessian captures these effects.

Finally, the DNR-CS algorithm can be adapted to the transition state and minimum energy path optimizations in the seam described in ref 14. It can be also adapted to a hybrid QM/MM formalism to optimize MECI structures in extended or biological systems and in the condensed phase. This will be the subject of future work.

■ ASSOCIATED CONTENT

■ Supporting Information

Active spaces of the CASSCF calculations and Cartesian coordinates of the MECI. This material is available free of charge via the Internet at <http://pubs.acs.org>.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: lluis.blancafort@udg.edu.

Notes

The authors declare no competing financial interests.

■ ACKNOWLEDGMENTS

This work has been supported by grants CTQ2008-06696 and CTQ2011-26573 from the Spanish Ministerio de Ciencia e Innovación (MICINN) and Ministerio de Economía y Competitividad (MINECO), respectively, SGR0528 from the Catalan Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR), UNGI08-4E-003 from MICINN and the European Fund for Regional Development, and the Xarxa de Referència en Química Teòrica i Computacional de Catalunya from AGAUR. S.R.-B. thanks the MICINN for grant BES-2009-029177.

■ REFERENCES

- (1) (a) Klessinger, M.; Michl, J. *Excited States and Photochemistry of Organic Molecules*; VCH Publishers, Inc.: New York, 1995. (b) Bernardi, F.; Olivucci, M.; Robb, M. A. *Chem. Soc. Rev.* **1996**, 25, 321. (c) Yarkony, D. R. *Rev. Mod. Phys.* **1996**, 68, 985. (d) Domcke, W.; Yarkony, D. R.; Köppel, H. *Conical Intersections: Electronic Structure, Dynamics & Spectroscopy*; World Scientific: Singapore, 2004. (e) Domcke, W.; Yarkony, D. R.; Köppel, H. *Conical Intersections: Theory, Computation and Experiment*; World Scientific: Singapore, 2011.
- (2) Boggio-Pasqua, M.; Burmeister, C. F.; Robb, M. A.; Groenhof, G. *Phys. Chem. Chem. Phys.* **2012**, 14, 7912.
- (3) González, L.; Escudero, D.; Serrano-Andrés, L. *ChemPhysChem* **2012**, 13, 28.
- (4) Blancafort, L.; Ogliaro, F.; Olivucci, M.; Robb, M. A.; Bearpark, M. J.; Sinicropi, A. Computational investigation of photochemical reaction mechanisms. In *Computational Methods in Photochemistry*; Kutateladze, A. G., Ed.; Taylor & Francis: Boca Raton, FL, 2005; Vol. 13, p 31.
- (5) (a) Manaa, M. R.; Yarkony, D. R. *J. Chem. Phys.* **1993**, 99, 5251. (b) Bearpark, M. J.; Robb, M. A.; Schlegel, H. B. *Chem. Phys. Lett.* **1994**, 223, 269. (c) Anglada, J. M.; Bofill, J. M. *J. Comput. Chem.* **1997**, 18, 992.
- (6) (a) Yarkony, D. R. *J. Phys. Chem. A* **2004**, 108, 3200. (b) De Vico, L.; Olivucci, M.; Lindh, R. *J. Chem. Theory Comput.* **2005**, 1, 1029. (c) Sicilia, F.; Blancafort, L.; Bearpark, M. J.; Robb, M. A. *J. Chem. Theory Comput.* **2008**, 4, 257.
- (7) (a) Toniolo, A.; Ben-Nun, M.; Martinez, T. J. *J. Phys. Chem. A* **2002**, 106, 4679. (b) Maeda, S.; Ohno, K.; Morokuma, K. *J. Chem. Theory Comput.* **2010**, 6, 1538. (c) Quapp, W.; Bofill, J. M.; Caballero, M. *Chem. Phys. Lett.* **2012**, 541, 122.
- (8) (a) Muñoz Losa, A.; Martín, M. E.; Galván, I. F.; Aguilar, M. A. *Chem. Phys. Lett.* **2007**, 443, 76. (b) Bearpark, M. J.; Larkin, S. M.; Vreven, T. *J. Phys. Chem. A* **2008**, 112, 7286. (c) Mori, T.; Nakano, K.; Kato, S. *J. Chem. Phys.* **2010**, 133, 064107. (d) Cui, G.; Yang, W. *J. Chem. Phys.* **2011**, 134, 204115.
- (9) Fletcher, R. *Practical Methods of Optimization*; Wiley & Sons: New York, 1981; Vol. 2.
- (10) Keal, T. W.; Koslowski, A.; Thiel, W. *Theor. Chem. Acc.* **2007**, 118, 837.
- (11) (a) Yarkony, D. R. *J. Chem. Phys.* **1990**, 92, 2457. (b) Yamamoto, N.; Vreven, T.; Robb, M. A.; Frisch, M. J.; Schlegel, H. B. *Chem. Phys. Lett.* **1996**, 250, 373.
- (12) Peng, C. Y.; Ayala, P. Y.; Schlegel, H. B.; Frisch, M. J. *J. Comput. Chem.* **1996**, 17, 49.
- (13) Schlegel, H. B. Geometry Optimization: 1. In *Encyclopedia of Computational Chemistry*; Schleyer, P. v. R., Ed.; John Wiley & Sons: Chichester, U.K., 1998; Vol. 2, p 1136.
- (14) Sicilia, F.; Blancafort, L.; Bearpark, M. J.; Robb, M. A. *J. Phys. Chem. A* **2007**, 111, 2182.
- (15) Frisch, 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. J.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Keith, T.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, J. M.; 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, G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Parandekar, P. V.; Mayhall, N. J.; Daniels, A. D.; Farkas, O.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. *Gaussian Development Version*, revision H.13; Gaussian, Inc.: Wallingford, CT, 2010.
- (16) Pulay, P.; Fogarasi, G. *J. Chem. Phys.* **1992**, 96, 2856.
- (17) Wilson, E. B., Jr.; Decius, J. C.; Cross, P. C. *Molecular Vibrations*; McGraw-Hill: New York, 1955.
- (18) Bearpark, M. J.; Blancafort, L.; Paterson, M. J. *Mol. Phys.* **2006**, 104, 1033.
- (19) Frisch, 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. J.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Keith, T.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, J. M.; 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, G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, O.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. *Gaussian 09*, Revision C.01; Gaussian, Inc.: Wallingford, CT, 2010.
- (20) Helbing, J.; Bregy, H.; Bredenbeck, J.; Pfister, R.; Hamm, P.; Huber, R.; Wachtveitl, J.; De Vico, L.; Olivucci, M. *J. Am. Chem. Soc.* **2004**, 126, 8823.

- (21) Olivucci, M.; Ragazos, I. N.; Bernardi, F.; Robb, M. A. *J. Am. Chem. Soc.* **1993**, *115*, 3710.
- (22) Garavelli, M.; Celani, P.; Bernardi, F.; Robb, M. A.; Olivucci, M. *J. Am. Chem. Soc.* **1997**, *119*, 11487.
- (23) Paterson, M. J.; Bearpark, M. J.; Robb, M. A.; Blancafort, L. J. *Chem. Phys.* **2004**, *121*, 11562.