Theor Chem Account (2007) 118:837–844
DOI 10.1007/s00214-007-0331-5

REGULAR ARTICLE

Comparison of algorithms for conical intersection optimisation using semiempirical methods

Thomas W. Keal · Axel Koslowski · Walter Thiel

Received: 29 March 2007 / Accepted: 25 April 2007 / Published online: 6 June 2007

Abstract We present a comparison of three previously published algorithms for optimising the minimum energy crossing point between two Born–Oppenheimer electronic states. The algorithms are implemented in a development version of the MNDO electronic structure package for use with semiempirical configuration interaction methods. The penalty function method requires only the energies and gradients of the states involved, whereas the gradient projection and Lagrange–Newton methods also require the calculation of non-adiabatic coupling terms. The performance of the algorithms is measured against a set of well-known small molecule conical intersections. The Lagrange–Newton method is found to be the most efficient, with the projected gradient method also competitive. The penalty function method can only be recommended for situations where non-adiabatic coupling terms cannot be calculated.

Keywords Potential–energy surfaces · Conical intersections · Optimization · Nonadiabatic coupling terms · Semiempirical methods

1 Introduction

The vital role that conical intersections play in many photochemical processes is well-known [1,2]. A conical intersection occurs when the potential energy surfaces of two or more adiabatic electronic states cross. If this region is accessible to a chemical species in an upper state, then an ultrafast radiationless transition to a lower state is possible. This phenomenon has been observed in systems ranging from simple molecules to complex biological systems such as the retinal chromophore in rhodopsin [3].

Conical intersections in polyatomic systems are multidimensional seams rather than single points. The seam can be characterised with a conical intersection optimisation, which is used to find the minimum energy crossing point (MECP) on the seam. This is not as straightforward as a geometry optimisation on a single potential energy surface, because the algorithm used must incorporate the constraint that the states involved have the same energy. A variety of approaches have been taken to accomplish this task [4–12].

A conical intersection optimisation requires a balanced description of the electronic structure of all the states involved. Ab initio multi-reference methods such as CASSCF or CASPT2 are widely used, but these are computationally expensive. Semiempirical configuration interaction methods offer a lower cost alternative, while still taking into account the correlation effects necessary to describe excited states.

In this article, we present our implementation of three previously published conical intersection optimisation algorithms in a development version of the semiempirical MNDO package [13]. Section 2 outlines the theoretical basis of the algorithms and details of their implementation. The algorithms are validated against several well-known small molecule conical intersections in Sect. 3 and their performance is discussed. Section 4 summarises our recommendations for conical intersection optimisation.

Electronic supplementary material The online version of this article (doi:10.1007/s00214-007-0331-5) contains supplementary material, which is available to authorized users.

T. W. Keal · A. Koslowski · W. Thiel (✉)
Max-Planck-Institut für Kohlenforschung,
Kaiser-Wilhelm-Platz 1,
45470 Mülheim an der Ruhr, Germany
e-mail: thiel@mpi-muelheim.mpg.de

2 Theory and methods

The problem of finding a conical intersection between two states is an example of an equality constrained optimisation [14]

$$\begin{aligned} &\text{minimise} && E_J \\ &\text{subject to} && E_J - E_I = 0 \end{aligned} \tag{1}$$

where E_I is the energy of the lower state I and E_J is the energy of the upper state J. In fact, the function to be minimised can be the energy of either of the states or the mean of the two, because all of these quantities have the same value at a point of conical intersection.

The points where the constraint is satisfied form a conical intersection seam (also known as the intersection space). In this work, we consider intersections between two states of the same spin multiplicity, for which the seam is a hyperline of $(N-2)$ dimensions, where N is the number of nuclear coordinates. An MECP is a local minimum on the seam. The degeneracy of the states is lifted by moving in the remaining two directions, which define a plane called the branching space. If the energy of the states is plotted against these two directions, which we denote $\mathbf{g}_{IJ}$ and $\mathbf{h}_{IJ}$, a double cone is formed around the degeneracy, hence the name conical intersection.

The gradient difference vector $\mathbf{g}_{IJ}$, is defined for a configuration interaction (CI) wavefunction as

$$g_{IJ}^{q_\alpha} = \mathbf{C}_I^\dagger \frac{\partial \mathbf{H}}{\partial q_\alpha} \mathbf{C}_I - \mathbf{C}_J^\dagger \frac{\partial \mathbf{H}}{\partial q_\alpha} \mathbf{C}_J \tag{2}$$

where q_α represents an individual nuclear coordinate, $\mathbf{C}_I$ are the CI coefficients of state I, and $\mathbf{H}$ is the CI electronic Hamiltonian matrix. The gradient of the interstate coupling $\mathbf{h}_{IJ}$, is

$$h_{IJ}^{q_\alpha} = \mathbf{C}_I^\dagger \frac{\partial \mathbf{H}}{\partial q_\alpha} \mathbf{C}_J \tag{3}$$

This quantity is closely related to, but not the same as, the non-adiabatic coupling vector [7]. $\mathbf{g}_{IJ}$ and $\mathbf{h}_{IJ}$ are central quantities in two of the algorithms described below.

In the MNDO package, the CI wavefunctions are calculated from a semiempirical Hamiltonian using the GUGA-CI formalism [15]. The quantity $\mathbf{h}_{IJ}$ is calculated using a previously implemented semi-analytic gradient routine [16], but with transition density matrices replacing standard density matrices according to the formulation of Lengsfield and Yarkony [17].

2.1 Penalty function method

The method of Ciminelli et al. [10] is a penalty function method, in which the constraint is enforced by adding a term to the objective function that increases as the energy difference between the states increases. No knowledge of $\mathbf{h}_{IJ}$ is required. The form given in Ref. [10] is

$$f(\mathbf{R}) = \frac{E_I + E_J}{2} + c_1 c_2^2 \ln\left[1 + \left(\frac{E_J - E_I}{c_2}\right)^2\right]. \tag{4}$$

The first term minimises the average of the lower and upper state energies and the second term is the penalty function, which minimises the energy difference. The constant c_1 determines how much weight should be given to these two goals, while c_2 controls how quickly the conical intersection seam is approached. The optimisation will not converge exactly to a true constrained minimum unless $c_1 \to \infty$, but highly distorted geometries on the seam can be reached if c_1 is too large, leading to optimisation failure. The recommended values from Ref. [10] are $c_1 = 5$ (kcal mol^{-1})$^{-1}$ and $c_2 = 5$ kcal mol^{-1}.

The penalty function method is implemented in MNDO using its standard quasi-Newton geometry optimisation routines, but with the 'energy' replaced by Eq. 4 and the 'gradient' replaced by the gradient of Eq. 4. The initial Hessian is approximated as a diagonal matrix by finite difference of the gradient and updated using the BFGS approximation [14]. Two minor modifications were required relating to tests for resetting the Hessian. By default the Hessian is reset when the objective function drops by more than 10 kcal mol^{-1}. As the value of Eq. 4 may change quite rapidly (at least as long as the energy difference between states is significant), the threshold was raised to 100 kcal mol^{-1}. The Hessian is also normally reset when the gradient vector and the line search direction become too close to being orthogonal. However, this test had to be removed as it hindered convergence towards the end of the optimisation. Together these modifications greatly improved the performance of the penalty function method.

2.2 Gradient projection method

The method of Bearpark et al. [8] is a gradient projection method. To enforce the constraint of Eq. 1, the energy difference $E_I - E_J$ is minimised in the branching space, while the upper state energy E_J is minimised in the intersection space to find the MECP. This separation between the two objectives of the optimisation avoids the problems associated with the penalty function method.

Following Ref. [18], we define the gradient that minimises the energy difference as

$$\mathbf{f}_1 = 2(E_I - E_J)\frac{\mathbf{g}_{IJ}}{|\mathbf{g}_{IJ}|}. \tag{5}$$

To minimise the upper state energy in the intersection space, the upper state gradient must be projected into it. The intersection space is the orthogonal complement to the $\mathbf{g}_{IJ}$, $\mathbf{h}_{IJ}$

plane, so the projection matrix $\mathbf{P}$ is defined as

$$\mathbf{P} = \mathbf{I} - \tilde{\mathbf{g}}_{IJ}\tilde{\mathbf{g}}_{IJ}^{\dagger} - \tilde{\mathbf{h}}_{IJ}\tilde{\mathbf{h}}_{IJ}^{\dagger} \tag{6}$$

where $\tilde{\mathbf{g}}_{IJ}$ and $\tilde{\mathbf{h}}_{IJ}$ represent the branching space vectors after orthonormalisation. The upper state gradient projection is then

$$\mathbf{f}_2 = \mathbf{P}\frac{\partial E_J}{\partial \mathbf{q}}. \tag{7}$$

The gradient to be minimised is the linear combination

$$\mathbf{g} = c_3\left[c_4\mathbf{f}_1 + (1 - c_4)\mathbf{f}_2\right] \tag{8}$$

where $c_3 > 0$ and $0 < c_4 \leq 1$. The value of c_3 scales the entire gradient, which is an indirect means of controlling the step size calculated by the quasi-Newton procedure. This means that $\mathbf{g}$ cannot be used as a measure of convergence as the threshold will vary depending on the choice of c_3. Instead, we follow Ref. [18] and test for convergence against the projected gradient $\mathbf{f}_2$. The value of c_4 determines the relative weight given to the two components of the gradient. As the value of both components should be zero at the MECP, the true minimum should be found no matter what value of c_4 is chosen (unlike the penalty function method). However, the value chosen may slow or hinder convergence.

One limitation of the gradient projection method is that the dimensions of the two components of the gradient do not agree, and therefore there is no corresponding objective function to minimise [19]. This means that techniques which rely on a well-defined objective function, such as a line search or dynamic trust radius, cannot be used. Unfortunately, the full quasi-Newton step can usually also not be used (until the system is close to the constrained minimum), because the quadratic approximation may not hold over the full distance of the step and the approximate Hessian will be of relatively poor quality (if the initial Hessian is a unit matrix).

Ref [18] overcomes this problem by using a combination of the scale factor $c_3 = 0.01$ and a static trust radius of 0.1 Å. A weighting of $c_4 = 0.01$ was used, which puts most of the emphasis on minimising $\mathbf{f}_2$. In Ref [19], the Newton step is simply scaled by a factor of 0.15. In our experience both of these approaches are problematic. In the latter case, a simple scale factor on the Newton step slows convergence near the constrained minimum, where the full Newton step could safely be taken. This does not happen if the gradient is scaled instead, because the Hessian is updated consistently with the scaled gradient. In the former case, the static trust radius is helpful, but the chosen scale factors do not perform well. In particular, the optimisation would often fail to converge to a constrained minimum due to the low value of c_4.

In the standard MNDO quasi-Newton routine, the initial inverse Hessian matrix $\mathbf{B}^{-1}$ is not a unit matrix, but is approximated as a diagonal matrix using a finite difference of the gradient

$$B^{-1}_{\alpha\alpha} = \frac{\Delta q_\alpha}{\Delta g_\alpha}. \tag{9}$$

The Hessian is therefore inversely proportional to the size of the gradient. This means that any initial scaling of the gradient using c_3 is effectively cancelled out by the Hessian. Fortunately, the improved quality of the Hessian means that this scaling is not necessary anyway, so we use a nominal value of $c_3 = 1.0$. Again the Hessian is updated using the BFGS formula.

The final parameter c_4 is more important because the weighting between the two components can strongly affect the rate of convergence. If the gradient is too strongly weighted to minimising the energy difference, the geometry can become highly distorted and the local minimum can become unreachable (or the orbital tracking procedure can fail; see Sect. 3). If it is too strongly weighted to minimising the excited state energy, the conical intersection may never be reached. In practice a value of $c_4 = 0.9$ minimises convergence problems. We did not choose to scale the Newton step for the reasons outlined above, but a static trust radius of 0.1 Å was applied.

2.3 Lagrange–Newton method

The method of Manaa and Yarkony [7] is a Lagrange–Newton method. In this method at least two constraints are used. The two mandatory constraints specify that at the point of conical intersection,

$$E_I - E_J = 0 \tag{10}$$

$$H_{IJ} \equiv \mathbf{C}_I^{\dagger}\mathbf{H}\mathbf{C}_J = 0. \tag{11}$$

The derivatives of these constraints correspond to $\mathbf{g}_{IJ}$ and $\mathbf{h}_{IJ}$, respectively. The constraints are enforced by associating each with a Lagrange multiplier and finding the stationary point of a Lagrangian function. In the original scheme, the Lagrangian was

$$L_{IJ}(\mathbf{q}, \boldsymbol{\xi}, \boldsymbol{\lambda}) = E_I + \xi_1(E_I - E_J) + \xi_2 H_{IJ} + \sum_{i=1}^{M}\lambda_i K_i \tag{12}$$

The first term minimises the lower state energy, while the second and third terms enforce the constraints of Eqs. 10 and 11. The final term in the Lagrangian is optional. It represents M explicit geometrical constraints (for example, on bond lengths or angles). These constraints have been implemented in the MNDO program but are not considered in this work as the other two algorithms do not support extra constraints.

By analogy to the Newton–Raphson method for unconstrained optimisation, the Lagrangian function is expanded to

second order, giving

$$\begin{bmatrix} \nabla\nabla L_{IJ} & \mathbf{g}_{IJ} & \mathbf{h}_{IJ} & \mathbf{k} \\ \mathbf{g}_{IJ}^{\dagger} & 0 & 0 & \mathbf{0} \\ \mathbf{h}_{IJ}^{\dagger} & 0 & 0 & \mathbf{0} \\ \mathbf{k}^{\dagger} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta\mathbf{q} \\ \delta\xi_1 \\ \delta\xi_2 \\ \delta\boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \nabla L_{IJ} \\ E_I - E_J \\ 0 \\ \mathbf{K} \end{bmatrix} \tag{13}$$

with the gradient of the Lagrangian

$$\nabla L_{IJ} = \mathbf{g}_I + \xi_1 \mathbf{g}_{IJ} + \xi_2 \mathbf{h}_{IJ} + \sum_{i=1}^{M} \lambda_i \mathbf{k}_i \tag{14}$$

where the geometrical constraint gradients $\mathbf{k}_i$ are defined as in Ref. [7]. The equations are then solved for $\delta\mathbf{q}$, $\delta\xi_1$, $\delta\xi_2$ and $\delta\boldsymbol{\lambda}$ and iterated until convergence (tested against the norm of the right hand side of Eq. 13).

This original method suffered from the problem that some components of the Lagrangian gradient ($\mathbf{g}_I$, $\mathbf{g}_{IJ}$, and $\mathbf{h}_{IJ}$) are not slowly varying near the conical intersection seam. This prevented an accurate calculation of the Hessian using standard finite difference techniques, or a reliable update using BFGS or other quasi-Newton methods. This problem was later solved by the introduction of 'extrapolatable functions' [20]. First, instead of minimising the lower state energy, the mean energy of both states is minimised, giving the modified Lagrangian

$$L_{IJ} = \frac{E_I + E_J}{2} + \xi_1 (E_I - E_J) + \xi_2 H_{IJ} + \sum_{i=1}^{M} \lambda_i K_i \tag{15}$$

with the gradient

$$\nabla L_{IJ} = \frac{\mathbf{g}_I + \mathbf{g}_J}{2} + \xi_1 \mathbf{g}_{IJ} + \xi_2 \mathbf{h}_{IJ} + \sum_{i=1}^{M} \lambda_i \mathbf{k}_i \tag{16}$$

Moreover, at a point on the seam (where $E_I - E_J = 0$), $\mathbf{g}_{IJ}$ and $\mathbf{h}_{IJ}$ lose their meaning as independent variables. This is because the degenerate wave functions of states I and J are defined only up to a rotation among themselves. $\mathbf{g}_{IJ}$ and $\mathbf{h}_{IJ}$ can therefore also be defined only up to a rotation, as

$$\begin{bmatrix} \frac{1}{2}\mathbf{g}_{IJ,\theta} \\ \mathbf{h}_{IJ,\theta} \end{bmatrix} = \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{bmatrix} \begin{bmatrix} \frac{1}{2}\mathbf{g}_{IJ} \\ \mathbf{h}_{IJ} \end{bmatrix} \tag{17}$$

As the seam is approached, the first two constraint terms in Eq. 13 become equivalent because the energy difference on the right hand side tends to zero. This means that any $\mathbf{g}_{IJ,\theta}$, $\mathbf{h}_{IJ,\theta}$ pair can be used in Eq. 16. In particular, the vectors can be orthogonalised, which gives a new pair of vectors which are slowly varying in the region of a conical intersection. The value of θ for which $\frac{1}{2}\mathbf{g}_{IJ,\theta}$ and $\mathbf{h}_{IJ,\theta}$ are orthogonal is given by [20]

$$\tan 4\theta = \frac{\mathbf{h}_{IJ} \cdot \mathbf{g}_{IJ}}{(\mathbf{h}_{IJ} \cdot \mathbf{h}_{IJ}) - \frac{1}{4}(\mathbf{g}_{IJ} \cdot \mathbf{g}_{IJ})} \tag{18}$$

Eqs. 17 and 18 are used to generate the orthogonalised vectors $\bar{\mathbf{g}}_{IJ}$ and $\bar{\mathbf{h}}_{IJ}$, which replace $\mathbf{g}_{IJ}$ and $\mathbf{h}_{IJ}$ in Eq. 16 when the optimiser reaches a point sufficiently close to the seam. In our implementation, the orthogonalisation procedure is switched on when the magnitude of the energy difference is below a threshold of $t_1 = 10^{-4}$ kcal mol^{-1}.

The orthogonalised vectors are unique up to transpositions and sign changes (which can occur as more than one value of θ satisfies Eq. 18). However, the vectors are only slowly varying if transpositions and sign changes are identified and reversed if necessary. In the MNDO implementation these possibilities are tested with an overlap criterion between optimisation cycles. The residual energy difference on the right hand side of Eq. 13 must also be taken into account. Convergence is hindered unless the residual energy difference is also transposed and has its sign changed along with the vectors. Furthermore, the energy difference term should be halved if it is transposed (this reflects the fact that $\frac{1}{2}\mathbf{g}_{IJ}$ is used in Eq. 17).

The use of extrapolatable functions ensures that the Hessian of the Lagrangian ($\nabla\nabla L_{IJ}$) can be updated by quasi-Newton methods, which use a finite difference of the gradient between two geometry steps. Before the optimiser reaches the threshold t_1, the gradient difference and interstate coupling gradient terms are excluded from this calculation (because they are not orthogonalised and therefore not slowly varying). The gradient difference used is then

$$\delta\nabla L_{IJ} = \delta\left(\frac{\mathbf{g}_I + \mathbf{g}_J}{2}\right) + \sum_{i=1}^{M} \lambda_i \delta\mathbf{k}_i. \tag{19}$$

This finite difference is sufficiently accurate to find the conical intersection seam, but when the threshold t_1 is reached, a more accurate form is available

$$\delta\nabla L_{IJ} = \delta\left(\frac{\mathbf{g}_I + \mathbf{g}_J}{2}\right) + \xi_1 \delta\bar{\mathbf{g}}_{IJ} + \xi_2 \delta\bar{\mathbf{h}}_{IJ} + \sum_{i=1}^{M} \lambda_i \delta\mathbf{k}_i. \tag{20}$$

Note that the Lagrange multipliers in Eqs. 19 and 20 are those of the current iteration.

In Ref. [20], the Hessian update algorithm of Murtagh and Sargent [21] is recommended. This algorithm has been implemented, but in practice the BFGS procedure was found to be more efficient. The BFGS approximation always gives a positive definite $\nabla\nabla L_{IJ}$, but the overall Hessian has one negative eigenvalue for each Lagrange multiplier.

In some cases, particularly when the conical intersection seam is found quickly but the geometry is far from the MECP, the energy difference may rise again to the point where Eq. 17 is no longer a reasonable approximation and the orthogonalisation procedure should be turned off again. We defined this threshold as $t_2 = 1$ kcal mol^{-1}.

Unlike the penalty function and gradient projection methods, the Lagrange–Newton method cannot be trivially incorporated into an existing optimisation routine. The MNDO implementation is based on an existing Newton–Raphson routine, but with modifications resulting from the form of Eq. 13 and the determination of the Hessian as outlined above. Unlike the other methods, an inverse Hessian cannot be used because of the need to explicitly insert values for the rows and columns corresponding to the constraints in Eq. 13. The computational cost of solving Eq. 13 scales as N^3, compared to N^2 for the equivalent step with an inverse Hessian. The initial Hessian is diagonal with empirical values corresponding to bond lengths, angles and dihedral angles. In our calculations, a static trust region of 0.1 Å is used.

3 Results and discussion

The three algorithms were validated using a set of 12 well-known conical intersections from ten molecules: the aliphatic systems ethylene [22], butadiene [18,23], ketene [24] and diazomethane [25,26]; the aromatic systems benzene [8,27], azulene [28] and indacene [29]; and three small retinal protonated Schiff base models: methaniminium [30], penta-3,5-dieniminium [26,31] and 4-*cis*-γ-methylnona-2,4,6,8-tetraeniminium [32]. In all cases the conical intersections are between the ground state and the first excited singlet state. In the case of butadiene three minima on the conical intersection seam were considered (*s*-cisoid, central, and *s*-transoid). The conical intersection geometries are illustrated in Fig. 1.

All calculations were carried out in internal coordinates using the OM2 semiempirical Hamiltonian [33,34] and GUGA configuration interaction. The SCF calculations were performed in a restricted open-shell Hartree–Fock (ROHF) formalism corresponding to a singlet excited state to avoid convergence problems associated with restricted Hartree–Fock (RHF) when there are two solutions very close in energy (as is the case near a point of conical intersection). In each case three reference configurations were used for the configuration interaction procedure, corresponding to the closed shell ground state and to single and double HOMO–LUMO excitations. All single and double excitations within the active space from these references were included in the calculations.

The active space was chosen in accordance with previous studies, which for most of the molecules was the π-system (or those orbitals with the most π character when at a distorted geometry). Ethylene and the methaniminium cation also include in the active space all occupied orbitals corresponding to b_2 character at the ground state minimum, and the active spaces for ketene and diazomethane consisted of two orbitals of b_2 character and three (π) orbitals of b_1 character at the ground state minimum. The orbitals were identified at distorted geometries by inspection. The size of the active spaces along with ROHF occupations are given in Table 1.

The orbitals in the active space were tracked by calculating the dot product between the orbital coefficients of the

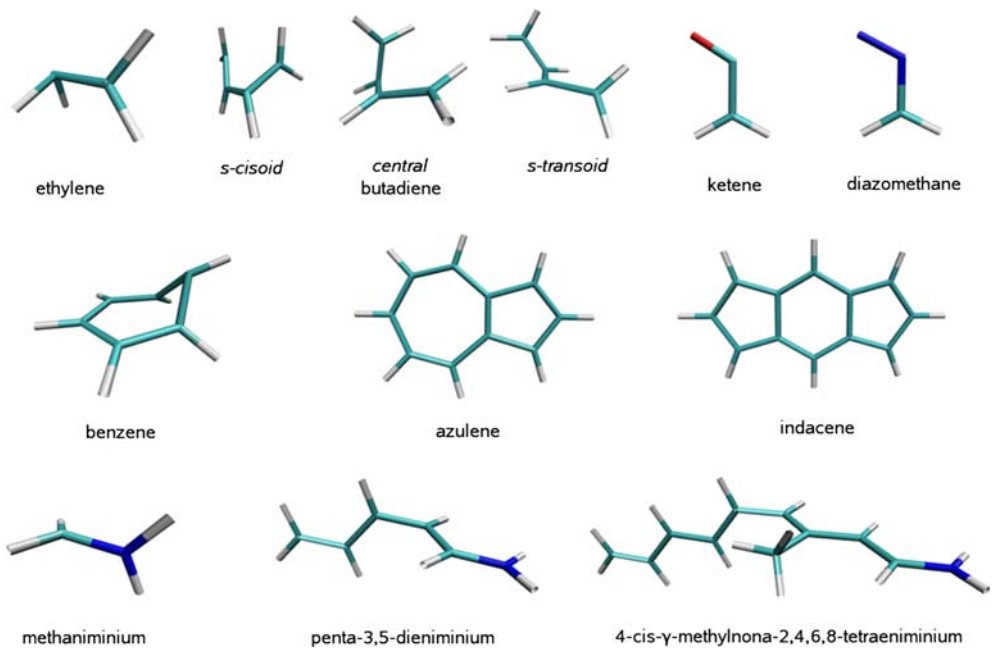

Fig. 1 Geometries of the 12 conical intersections considered in this study. The geometries shown correspond to the minimum energy crossing points found using the Lagrange–Newton algorithm

Table 1 Number of optimisation cycles and computation times for the set of 12 conical intersection optimisations

System			Active	Start		RMSD=0.01 Å[e]		RMSD=0.001 Å[e]		MECP[f]/kcal mol^{-1}		
	Sym[a]	DOF[b]	space[c]	RMSD/Å[d]	Method	Cycles	Time/s	Cycles	Time/s	E_I	E_J	ΔE
Ethylene	C_1	12	4-0′	0.176	PF	26	0.6	35	0.8	121.70922	121.79149	0.08227
(twisted-pyramidal)					GP	22	0.7	34	1.1	121.81800	121.81800	0.00000
					LN	10	0.3	22	0.7	121.81800	121.81800	0.00000
Butadiene												
s-cisoid	C_1	24	3-1′	0.192	PF	86	7.1	116	9.4	121.93681	121.98372	0.04691
					GP	15	1.9	19	2.4	121.98211	121.98211	0.00000
					LN	16	1.9	20	2.4	121.98211	121.98211	0.00000
central	C_1	24	3-1′	0.455	PF	143	11.6	165	13.3	121.69717	121.75675	0.05958
					GP	39	4.6	48	5.6	121.76249	121.76249	0.00000
					LN	26	3.0	33	3.8	121.76249	121.76249	0.00000
s-transoid	C_1	24	3-1′	0.171	PF	65	5.5	76	6.4	119.00126	119.04762	0.04636
					GP	25	3.0	31	3.6	119.04592	119.04593	0.00001
					LN	19	2.2	102	11.5	119.04593	119.04593	0.00000
Ketene	C_s	7	4-1′	0.289	PF	28	0.7	29	0.8	31.27625	31.35757	0.08132
					GP	18	0.5	23	0.7	31.38311	31.38312	0.00001
					LN	12	0.4	16	0.5	31.38311	31.38311	0.00000
Diazomethane	C_s	7	4-1′	0.303	PF	34	0.9	35	1.0	85.60343	85.71396	0.11053
					GP	20	0.7	24	0.8	85.78111	85.78112	0.00001
					LN	14	0.5	18	0.6	85.78112	85.78112	0.00000
Benzene	C_1	30	4-2′	0.133	PF	69	10.7	93	14.1	125.73100	125.76845	0.03745
					GP	17	3.8	32	6.9	125.76374	125.76374	0.00000
					LN	10	2.2	13	2.8	125.76374	125.76374	0.00000
Azulene	C_s	33	6-4′	0.064	PF	37	30.4	56	46.1	100.93902	101.09346	0.15444
					GP	38	34.3	72	62.1	101.25496	101.25496	0.00000
					LN	9	8.5	12	11.0	101.25496	101.25497	0.00001
Indacene	C_s	37	7-5′	0.045	PF	26	50.2	52	100.7	92.42268	92.57986	0.15718
					GP	17	32.4	50	83.5	92.75017	92.75017	0.00000
					LN	10	18.5	32	52.3	92.75017	92.75017	0.00000
Methaniminium	C_1	12	4-0′	0.050	PF	50	1.3	54	1.4	246.62460	246.70463	0.08003
(twisted)					GP	29	1.1	32	1.2	246.72870	246.72870	0.00000
					LN	14	0.5	19	0.7	246.72870	246.72870	0.00000
Penta-3,5-dieniminium	C_1	36	4-2′	0.111	PF	100	21.6	123	26.5	227.45363	227.50335	0.04972
					GP	32	10.3	54	17.0	227.50331	227.50331	0.00000
					LN	32	10.2	72	22.3	227.50331	227.50331	0.00000
Nonatetraeniminium	C_1	69	6-4′	0.420	PF	214	292.7	239	326.5	225.21154	225.22774	0.01620
					GP	135	239.3	151	266.4	225.22228	225.22229	0.00001
					LN	52	90.8	58	101.0	225.22228	225.22228	0.00000

[a] Symmetry at which the optimisation was performed
[b] Number of degrees of freedom
[c] Number of occupied/unoccupied ROHF orbitals
[d] Root mean square deviation (RMSD) of the starting geometry from the final converged reference geometry after alignment
[e] Number of cycles and CPU time to reach the threshold RMSD calculated after alignment to the final converged reference geometry
[f] Heats of formation of the two states and their difference at the optimised minimum energy crossing point (MECP)
PF = penalty function, *GP* = gradient projection, *LN* = Lagrange–Newton
Computation times refer to one Intel Pentium 4-EM64T 3.40 GHz processor

current optimisation cycle with those of the cycle before. If any orbital in the active space could not be assigned with a confidence of over 90% as measured by the dot product, the step was rejected and a new step of half the distance was taken. The rejected steps were not counted in the optimisation cycle count. In general this procedure worked well, but the orbital components can still change by up to 10% per step. As conical intersections are usually found at distorted geometries, the orbitals themselves can change significantly over the course of the optimisation and this can lead to the wrong orbital being tracked, particularly if orbital mixing occurs. It was therefore important to confirm that the active space was still appropriately chosen at the end of the optimisation.

The starting geometries were generated by first optimising the ground state at the same level of theory and then applying an appropriate distortion so that the nearest local minimum would be the intended conical intersection. For example, the starting geometry for the retinal models included an approximately 90° twist in the central double bond. This method also avoided potential problems with orbital tracking when a bond is twisted. However, for two systems this method did not work. For butadiene, simply distorting the dihedral angle of the ground state geometry by an appropriate amount was not sufficient to ensure that the intended one out of the three conical intersections was found. For benzene, an out-of-plane distortion of one carbon atom was not sufficient to prevent an orbital tracking failure when the more heavily distorted conical intersection geometry was approached. In both cases CASSCF optimised conical intersection geometries were chosen as starting geometries [23,27]. Details of the starting and final geometries used are provided in the electronic supplementary material.

The results are presented in Table 1. The algorithms were compared based on how quickly they attained a given level of convergence. Unfortunately, because the gradients used in the optimisation are defined very differently for each method, they could not be used as a measure of convergence. The final heats of formation could also not be used because the penalty function method (with its default parameters) does not converge exactly to the same minimum as the other two methods. Therefore the only criterion that could be used to measure convergence was the geometry itself. The number of optimisation cycles and CPU time were recorded for each algorithm to reach threshold root mean square deviations (RMSDs) of 0.01 and 0.001 Å. For the gradient projection and Lagrange–Newton methods, a cycle corresponds to one geometry step, whereas for the penalty function method one cycle is one line search. The RMSDs were calculated after alignment using the algorithm of Refs. [35,36] with no mass weighting. The reference structure used for the alignment was an aligned average of the final converged structures of the gradient projection and Lagrange–Newton methods, which converged to within 10^{-5} kcal mol^{-1} of the true minimum in all cases.

The precise number of cycles and timings recorded should be treated with caution because differences in the details of the implementations can have a significant effect on performance. Nevertheless, the trend over the whole set of conical intersections is clear. The results indicate that the Lagrange–Newton algorithm is the most efficient overall, followed closely by the gradient projection algorithm, with the penalty function algorithm somewhat less efficient. There are only two exceptions to this general trend, namely the butadiene *s*-transoid and penta-3,5-dieniminium conical intersections. In both cases the Lagrange–Newton method converges to 0.01 Å quickly but takes longer than the other methods to subsequently converge to 0.001 Å. This behaviour can be attributed to the transition to extrapolatable functions, which can occasionally disrupt the optimisation.

The penalty function method does not perform as well as the other two methods, but it does have the advantage of simplicity. In situations where the calculation of non-adiabatic terms is impossible or not implemented, the penalty function approach would be an adequate alternative. We have confirmed that the penalty function method converges towards the true constrained minimum if the optimisation is re-run starting from the previous converged geometry with a higher value of c_1. Therefore if more accurate convergence is required, the c_1 parameter can be sequentially raised over two or more optimisations, although this would obviously require a greater overall number of optimisation cycles and CPU time.

The CPU timings indicate that the additional cost of calculating the gradient of the interstate coupling for the Lagrange–Newton and gradient projection methods is outweighed by the additional cost in optimisation cycles for the penalty function method. The N^3 step of solving Eq. 13 in the Lagrange–Newton method also does not appear to affect the timings significantly, at least for molecules of this size. This may however become important when larger systems are considered.

For all the molecules considered in the test set the approximate location of the conical intersection was known in advance. For applications where this is not the case, a search over a greater area of the potential energy surfaces would be necessary and so the reliability of the orbital tracking procedure would become more important. A general strategy would start with a smaller active space, both to minimise computational cost and because the tracking procedure would then be more reliable. A common approach to locating accessible conical intersections is to follow the minimum energy pathway on the excited state surface from the Franck–Condon point. Conical intersections found in this manner could then be re-optimised with a larger active space from an improved starting geometry.

4 Conclusions

Three previously published algorithms for conical intersection optimisation have been implemented in a development version of the MNDO semiempirical electronic structure program. Their performance was assessed over a set of 12 well-known small molecule conical intersections. The Lagrange–Newton method was found to be the most efficient in terms of optimisation cycles and CPU time, with the gradient projection method also competitive. The penalty function method can only be recommended in situations where it is not possible to calculate the non-adiabatic coupling terms. It remains to be seen how the algorithms will perform for conical intersections in larger systems. Future research will consider the implementation of these algorithms in the context of combined quantum mechanics/molecular mechanics calculations.

Acknowledgments We thank Paul Strodel (then University of Paderborn) for initial coding of the penalty function and gradient projection methods. We are grateful for helpful discussions with Marcus Elstner (University of Braunschweig) and Marius Wanko (University of Bremen). This work is supported by the Deutsche Forschungsgemeinschaft (SFB 663).

References

1. Bernardi F, Olivucci M, Robb MA (1996) Chem Soc Rev 25:321–328
2. Domcke W, Yarkony DR, Köppel H (eds) (2004) Conical intersections: electronic structure, dynamics and spectroscopy. Advanced series in physical chemistry, vol 15. World Scientific, Singapore
3. Schoenlein RW, Peteanu LA, Mathies RA, Shank CV (1991) Science 254:412–415
4. Koga N, Morokuma K (1985) Chem Phys Lett 119:371–374
5. Farazdel A, Dupuis M (1991) J Comput Chem 12:276–282
6. Ragazos IN, Robb MA, Bernardi F, Olivucci M (1992) Chem Phys Lett 197:217–223
7. Manaa MR, Yarkony DR (1993) J Chem Phys 99:5251–5256
8. Bearpark MJ, Robb MA, Schlegel HB (1994) Chem Phys Lett 223:269–274
9. Anglada JM, Bofill JM (1997) J Comput Chem 18:992–1003
10. Ciminelli C, Granucci G, Persico M (2004) Chem Eur J 10:2327–2341
11. De Vico L, Olivucci M, Lindh R (2005) J Chem Theory Comput 1:1029–1037
12. Levine BG, Ko C, Quenneville J, Martínez TJ (2006) Mol Phys 104:1039–1051
13. Thiel W (2007) MNDO program, version 6.1, Mülheim
14. Fletcher R (1987) Practical methods of optimization. Wiley, Chichester. 2nd edn.
15. Koslowski A, Beck ME, Thiel W (2003) J Comput Chem 24:714–726
16. Patchkovskii S, Koslowski A, Thiel W (2005) Theor Chem Acc 114:84–89
17. Lengsfield III BH, Yarkony DR (1992) Adv Chem Phys 82:1–71
18. Izzo R, Klessinger M (2000) J Comput Chem 21:52–62
19. Toniolo A, Ben-Nun M, Martínez TJ (2002) J Phys Chem A 106:4679–4689
20. Yarkony DR (2004) J Phys Chem A 108:3200–3205
21. Murtagh BA, Sargent RWH (1970) Comput J 13:185–194
22. Barbatti M, Paier J, Lischka H (2004) J Chem Phys 121:11614–11624
23. Olivucci M, Ragazos IN, Bernardi F, Robb MA (1993) J Am Chem Soc 115:3710–3721
24. Cui Q, Morokuma K (1997) J Chem Phys 107:4951–4959
25. Yamamoto N, Bernardi F, Bottoni A, Olivucci M, Robb MA, Wilsey S (1994) J Am Chem Soc 116:2064–2074
26. Page CS, Olivucci M (2003) J Comput Chem 24:298–309
27. Palmer IJ, Ragazos IN, Bernardi F, Olivucci M, Robb MA (1993) J Am Chem Soc 115:673–682
28. Bearpark MJ, Bernardi F, Clifford S, Olivucci M, Robb MA, Smith BR, Vreven T (1996) J Am Chem Soc 118:169–175
29. Bearpark MJ, Celani P, Jolibois F, Olivucci M, Robb MA, Bernardi F (1999) Mol Phys 96:645–652
30. Barbatti M, Aquino AJA, Lischka H (2006) Mol Phys 104:1053–1060
31. Garavelli M, Celani P, Bernardi F, Robb MA, Olivucci M (1997) J Am Chem Soc 119:6891–6901
32. Garavelli M, Vreven T, Celani P, Bernardi F, Robb MA, Olivucci M (1998) J Am Chem Soc 120:1285–1288
33. Weber W (1996) Ph.D. Thesis, Universität Zürich
34. Weber W, Thiel W (2000) Theor Chem Acc 103:495–506
35. Kabsch W (1976) Acta Cryst A 32:922–923
36. Kabsch W (1978) Acta Cryst A 34:827–828