Improved algorithms for reaction path following: Higher-order implicit algorithms

Carlos Gonzalez[a)] and H. Bernhard Schlegel[b)]
Department of Chemistry, Wayne State University, Detroit, Michigan 48202

(Received 13 May 1991; accepted 17 June 1991)

Eight new algorithms for reaction path following are presented, ranging from third order to sixth order. Like the second-order algorithm [J. Chem. Phys. **90**, 2154 (1989)] these are implicit methods, i.e., they rely on the tangent (and in some cases the curvature) at the endpoint of the step. The tangent (and the curvature, if needed) are obtained by a constrained optimization using only the gradient. At most, only one Hessian calculation is needed per step along the path. The various methods are applied to the Müller–Brown surface and to a new surface whose reaction path is known analytically to test their ability to follow the reaction path and to reproduce the curvature along the path.

I. INTRODUCTION

In the study of both reaction mechanisms and reaction kinetics, the concept of the reaction path is central.[1] For theoretical studies of mechanisms, it is desirable to follow the lengthy and curved path from the transition state forward to the products and back to the reactants at the lowest computational cost possible. For the calculation of rate constants by variational transition-state theory, accurate values for the curvature and frequencies along the path are needed,[2] preferably with as little computational effort as possible. However, reaction paths are notoriously difficult to follow, especially if the path is curved and twisted or if accurate properties such as curvature and frequencies are needed.[3,4] Numerous algorithms exist for following reaction paths.[3-8] Many of the traditional numerical methods for solving differential equations have difficulties (e.g., severe oscillations) even for simple reactions.[3,4] Many steps with very small step lengths are often needed to map out the path accurately. This is not a problem if an analytic surface is available; however, if the reaction path is determined directly from electronic structure calculations, as in the direct dynamics calculations,[4] this leads to excessively long computational times. Traditional higher-order methods are not necessarily more efficient than lower-order methods.[4]

A variety of specialized algorithms have been developed to overcome some of the difficulties of following reaction paths.[5-8] Ishida, Morokuma, and Komornicki[5] (IMK), with modifications by Gordon *et al.*, added a one-dimensional optimization to the Euler method to step back to the reaction path. Müller and Brown[6] (MB) used an optimization with a constrained step length to find the next point down hill along the path. Page and McIver[7] have integrated a Taylor expansion of the energy surface. Depending on the energy derivatives used and the terms retained in the integration, this yields a sequence of methods of increasing accuracy (QUAD, CUBE, etc.; LQA, CLQA, etc.). In earlier papers,[8] we presented a reaction path following algorithm based on a constrained optimization that is exact when the reaction path is an arc of a circle. This method has been shown to be more efficient than many of the other algorithms for small step sizes and better able to follow curved paths than other algorithms when large step sizes are used. In the limit of small step size, our algorithm is correct to second order, i.e., it yields the correct tangent and curvature vector both along the path and at the transition state. Because of the constrained optimization, our method returns to the vicinity of the true path even under conditions which cause other second order methods to deviate substantially.

Traditional numerical techniques use explicit methods for integrating ordinary differential equations[9] (ODE's), i.e., methods that do not require the value of the function at the endpoint of the step. Reaction path following often gives rise to stiff ordinary differential equations. Implicit methods can be proven to have much greater stability for stiff differential equations.[9] However, implicit methods require the final value of the function at the end of the integration step and, hence, are difficult to implement in a practical manner. Our second-order reaction path following algorithm is the same as the implicit trapezoid method;[9] our contribution is the manner of obtaining the final point needed for this method, i.e., the constrained optimization.

In the present paper, we present a family of third-, fourth-, and higher-order algorithms for reaction path following. Like our second-order algorithm, these are implicit methods and employ a constrained optimization to obtain the next point on the path. Some require no additional information beyond what is needed for the second order algorithm; some require an extra gradient calculation; others need at most only one second derivative calculation for each step along the path. The order of each algorithm is determined by comparison with the Taylor expansion of the reaction path. The new methods are tested on two analytical model potential-energy surfaces and are compared by their ability to follow the path and reproduce the magnitude of the curvature along the path.

II. THEORY

A. Taylor expansion of the reaction path

With the notation of Page and McIver,[7] the reaction path $\mathbf{x}(s)$ can be written as a function of the arc length s,

[a)] Current address: Department of Chemistry, Carnegie Mellon University, Pittsburgh, PA 15213.

[b)] Author to whom correspondence should be addressed.

$$\mathbf{x}(s) = \mathbf{x}(0) + s\mathbf{v}^0(0) + \tfrac{1}{2}s^2\mathbf{v}^1(0) + \tfrac{1}{6}s^3\mathbf{v}^2(0) + \cdots. \quad (1)$$

The tangent vector $\mathbf{v}^0$ and curvature vector $\mathbf{v}^1$ are given by

$$\mathbf{v}^0(s) = \frac{d\mathbf{x}(s)}{ds} = \mathbf{v}^0(0) + s\mathbf{v}^1(0) + \tfrac{1}{2}s^2\mathbf{v}^2(0) + \cdots, \quad (2)$$

$$\mathbf{v}^1(s) = \frac{d\mathbf{v}^0}{ds} = \mathbf{v}^1(0) + s\mathbf{v}^2(0) + \tfrac{1}{2}s^2\mathbf{v}^3(0) + \cdots. \quad (3)$$

If the energy surface is expanded in terms of the gradient $\mathbf{g}$, the Hessian H, and the third derivatives F,

$$\begin{aligned} E(\mathbf{x}) = E(\mathbf{0}) &+ \sum_i g(0)_i[x_i - x(0)_i] \\ &+ \tfrac{1}{2}\sum_i H(0)_{ij}[x_i - x(0)_i][x_j - x(0)_j] \\ &+ \tfrac{1}{6}\sum_i F(0)_{ijk}[x_i - x(0)_i][x_j - x(0)_j] \\ &\times [x_k - x(0)_k]\cdots, \end{aligned} \quad (4)$$

then the tangent and curvature at $\mathbf{x}(0)$ are given by

$$\mathbf{v}^0(0) = -\mathbf{g}(0)/|\mathbf{g}(0)|, \quad (5)$$

$$\mathbf{v}^1(0) = -\{H - [\mathbf{v}^0(0)^t H\mathbf{v}^0(0)]I\}\mathbf{v}^0(0)/|\mathbf{g}(0)|. \quad (6)$$

At the transition state, $\mathbf{g}$ is zero and the tangent vector is the eigenvector of the Hessian that corresponds to the negative eigenvalue. The curvature vector is given by

$$\begin{aligned} \mathbf{v}^1(0) = &-\{H - 2[\mathbf{v}^0(0)^t H\mathbf{v}^0(0)]I\}^{-1} \\ &\times\{F^1 - [\mathbf{v}^0(0)^t F^1\mathbf{v}^0(0)]I\}\mathbf{v}^0(0), \end{aligned} \quad (7)$$

where $F^1_{ij} = \Sigma_k F_{ijk}\mathbf{v}^0(0)_k$.

B. Second-order algorithm

In previous work,[8] we developed a reaction path algorithm that is formally second order and is able to follow an arc of a circle exactly. As shown in Fig. 1(a), the algorithm consists of two steps: (a) a move of $\frac{1}{2}\sigma$ from $\mathbf{x}(0)$ along $\mathbf{v}(0)$ to $\mathbf{x}^*$ (no energy or gradient calculations are carried out at this pivot point) and (b) a move of $\frac{1}{2}\sigma$ from $\mathbf{x}^*$ and optimize $\mathbf{x}(s)$ under the constraint $|\mathbf{x}(s) - \mathbf{x}^*| = \frac{1}{2}\sigma$. When the optimization has converged, the gradient perpendicular to $\mathbf{x}(s) - \mathbf{x}^*$ must be zero, i.e., $\mathbf{g}(s) = \text{const.}[\mathbf{x}(s) - \mathbf{x}^*]$. Since $\mathbf{v}^0(s) = -\mathbf{g}(s)/|\mathbf{g}(s)|$ and $\mathbf{v}^0(s)$ is a unit vector, the displacement from the pivot point can also be written as

$$\mathbf{x}(s) - \mathbf{x}^* = \tfrac{1}{2}\sigma\mathbf{v}^0(s) \quad (8)$$

[recall that $|\mathbf{x}(s) - \mathbf{x}^*| = \frac{1}{2}\sigma$]. The second-order algorithm can then be summarized as

$$\mathbf{x}(s) = \mathbf{x}(0) + \tfrac{1}{2}\sigma\mathbf{v}^0(0) + \tfrac{1}{2}\sigma\mathbf{v}^0(s), \quad (9)$$

where $\mathbf{v}^0(s)$ is obtained by optimization of $\mathbf{x}(s)$ such that $|\mathbf{x}(s) - \mathbf{x}^*| = \frac{1}{2}\sigma$. Since $\frac{1}{2}\sigma\mathbf{v}^0(0)$ and $\frac{1}{2}\sigma\mathbf{v}^0(s)$ form an isosceles triangle, the path between $\mathbf{x}(0)$ and $\mathbf{x}(s)$ with tangents $\mathbf{v}^0(0)$ and $\mathbf{v}^0(\mathrm{s})$ is an arc of a circle, by construction. This method is, in fact, the implicit trapezoid method or the implicit second order Runge–Kutta method of integrating ordinary differential equations.[9] Normally, implicit methods such as this are not practical because they require the value of the function at the endpoint [in this case $\mathbf{v}^0(s)$]. However, the constrained optimization makes it possible to find $\mathbf{x}(s)$ and $\mathbf{v}^0(s)$ without resorting to approximations.

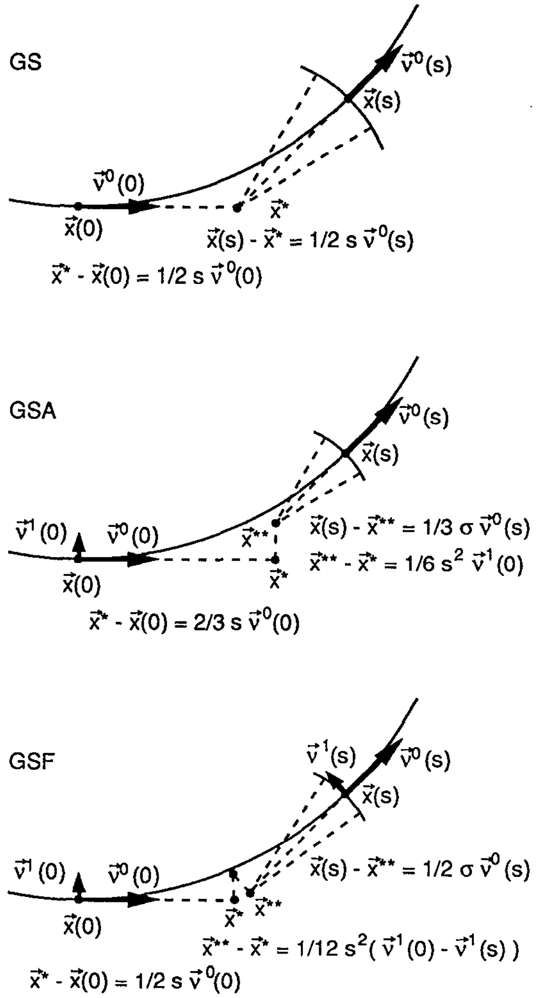
GS
$\vec{v}^0(s)$
$\vec{x}(s)$
$\vec{v}^0(0)$
$\vec{x}(0)$
$\vec{x}^*$
$\vec{x}(s) - \vec{x}^* = 1/2\ s\ \vec{v}^0(s)$
$\vec{x}^* - \vec{x}(0) = 1/2\ s\ \vec{v}^0(0)$
GSA
$\vec{v}^0(s)$
$\vec{x}(s)$
$\vec{v}^1(0)$
$\vec{v}^0(0)$
$\vec{x}^{**}$
$\vec{x}(s) - \vec{x}^{**} = 1/3\ \sigma\ \vec{v}^0(s)$
$\vec{x}^{**} - \vec{x}^* = 1/6\ s^2\ \vec{v}^1(0)$
$\vec{x}(0)$
$\vec{x}^*$
$\vec{x}^* - \vec{x}(0) = 2/3\ s\ \vec{v}^0(0)$
GSF
$\vec{v}^1(s)$
$\vec{v}^0(s)$
$\vec{x}(s)$
$\vec{v}^1(0)$
$\vec{v}^0(0)$
$\vec{x}(s) - \vec{x}^{**} = 1/2\ \sigma\ \vec{v}^0(s)$
$\vec{x}(0)$
$\vec{x}^*$
$\vec{x}^{**}$
$\vec{x}^{**} - \vec{x}^* = 1/12\ s^2(\vec{v}^1(0) - \vec{v}^1(s))$
$\vec{x}^* - \vec{x}(0) = 1/2\ s\ \vec{v}^0(0)$

FIG. 1. Schematics for the second-order algorithm, third-order method *A*, and fourth-order method *F*.

The step size σ is related to the arc length s by

$$s = 2\sigma\theta\cot\theta/2, \quad (10)$$

where $\mathbf{v}^0(0)\cdot v^0(s) = \cos\theta$.

The method can be generalized by writing it as a 2 bar linkage,

$$\mathbf{x}(s) = \mathbf{x}(0) + \sigma_1\mathbf{p}_1 + \sigma_2\mathbf{p}_2. \quad (11)$$

The vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ must be tangent to the path at $\mathbf{x}(0)$ and $\mathbf{x}(s)$, respectively. Thus

$$\mathbf{p}_1 = \mathbf{v}^0(0), \quad \mathbf{p}_2 = \mathbf{v}^0(s). \quad (12)$$

As discussed earlier, $\mathbf{p}_2$ is obtained by a constrained optimization of $\mathbf{x}(s)$ such that $|\mathbf{x}(s) - \mathbf{x}^*| = \sigma_2$. Insertion of Eq. (2) for $\mathbf{v}^0(s)$ and substitution of $\mathbf{p}_1$ and $\mathbf{p}_2$ into Eq. (11) yields

$$\mathbf{x}(s) = \mathbf{x}(0) + (\sigma_1 + \sigma_2)\mathbf{v}^0(0) + \sigma_2 s\mathbf{v}^1(0) + \cdots. \quad (13)$$

Comparison with the first- and second-order terms Eq. (1) gives

$$\sigma_1 = \tfrac{1}{2}s, \quad \sigma_2 = \tfrac{1}{2}s. \quad (14)$$

Therefore, the method is correct to second order as well as being exact for a circle. With other choices for σ_1 and σ_2, the method can be made exact for any planar curve.

C. Third-order algorithms

If a second-order implicit method can be constructed from a 2 bar linkage, then perhaps a third-order method can be built from a 3 bar linkage,

$$\mathbf{x}(s) = \mathbf{x}(0) + \sigma_1 \mathbf{p}_1 + \sigma_2 \mathbf{p}_2 + \sigma_3 \mathbf{p}_3. \tag{15}$$

As in the second-order algorithm, the requirement that the path be tangent to the linkage at $\mathbf{x}(0)$ and $\mathbf{x}(s)$ specifies the first and last bars in the linkage,

$$\mathbf{p}_1 = \mathbf{v}^0(0), \quad \mathbf{p}_3 = \mathbf{v}^0(s). \tag{16}$$

Like the second-order algorithm, $\mathbf{p}_3$ is obtained by a constrained optimization of $\mathbf{x}(s)$ such that $|\mathbf{x}(s) - [\mathbf{x}(0) + \sigma_1 \mathbf{p}_1 + \sigma_2 \mathbf{p}_2]| = \sigma_3$. A number of related methods can be devised depending on the choice of $\mathbf{p}_2$.

Method A: $p_2 = v^1(0)$

For some purposes, such as variational transition-state-theory calculations, the Hessian is needed at each point on the path. The curvature vector can then be computed from the Hessian via Eq. (6). If the Hessian is not available, then the curvature vector can be computed by numerical differentiation with only one additional gradient calculation,

$$\mathbf{v}^1(0) = [\mathbf{v}^0(\Delta) - \mathbf{v}^0(0)]/\Delta, \quad \Delta \ll s. \tag{17}$$

Substitution of $\mathbf{v}^0(0)$, $\mathbf{v}^1(0)$, and $\mathbf{v}^0(s)$ [from Eq. (2)] for $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ in Eq. (15) gives

$$\mathbf{x}(s) = \mathbf{x}(0) + \sigma_1 \mathbf{v}^0(0) + \sigma_2 \mathbf{v}^1(0) + \sigma_3 \left[\mathbf{v}^0(0) + s\mathbf{v}^1(0) + \tfrac{1}{2}s^2\mathbf{v}^2(0) + \tfrac{1}{6}s^3\mathbf{v}^3(0) + \cdots\right]. \tag{18}$$

Comparing order by order with Eq. (1) and requiring terms in $\mathbf{v}^0(0)$, $\mathbf{v}^1(0)$, and $\mathbf{v}^2(0)$ to be correct yields

$$\sigma_1 = \tfrac{2}{3}s, \quad \sigma_2 = \tfrac{1}{6}s^2, \quad \sigma_3 = \tfrac{1}{3}s. \tag{19}$$

The error in the fourth-order term is $\frac{1}{72}s^4\mathbf{v}^3(0)$. As indicated by Eq. (7), the curvature vector at the transition state depends on the third derivatives; however, the necessary information can be calculated by finite difference of the gradients,

$$F^1\mathbf{v}^0(0) = \{\mathbf{g}[\mathbf{x}_{ts} + s'\mathbf{v}^0(0)] + \mathbf{g}[\mathbf{x}_{ts} - s'\mathbf{v}^0(0)]\}/s'^2 \quad (s' \ll s). \tag{20}$$

Alternatively, the second-order algorithm can be used to obtain the first step away from the transition state.

Method B: $p_2 = v^0(-s)$

A second possibility for $\mathbf{p}_2$ is $\mathbf{v}^0(-s)$, the tangent vector from the previous optimized point on the path. Substitution of $\mathbf{v}^0(0)$, $\mathbf{v}^0(-s)$ (from Eq. (2)] and $\mathbf{v}^0(s)$ [from Eq. (2)] for $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ in Eq. (15) gives

$$\begin{aligned}\mathbf{x}(s) = \mathbf{x}(0) &+ \sigma_1 \mathbf{v}^0(0) \\ &+ \sigma_2 \left[\mathbf{v}^0(0) - s\mathbf{v}^1(0) + \tfrac{1}{2}s^2\mathbf{v}^2(0) - \tfrac{1}{6}s^3\mathbf{v}^3(0) + \cdots\right] \\ &+ \sigma_3 \left[\mathbf{v}^0(0) + s\mathbf{v}^1(0) + \tfrac{1}{2}s^2\mathbf{v}^2(0) + \tfrac{1}{6}s^3\mathbf{v}^3(0) + \cdots\right].\end{aligned} \tag{21}$$

From comparison with Eq. (1) the values for the σ's are

$$\sigma_1 = \tfrac{2}{3}s, \quad \sigma_2 = -\tfrac{1}{12}s, \quad \sigma_3 = \tfrac{5}{12}s. \tag{22}$$

The error in the fourth-order term is $\frac{1}{36}s^4\mathbf{v}^3(0)$. If a previous point on the path has already been computed, this method requires no additional energy or gradient computations beyond those needed for the second-order algorithm. For the first point on the path, the second-order algorithm can be used to step away from the transition state.

Method C: $p_2 = x(0) - x(-s)$

A third possibility for $\mathbf{p}_2$ is the displacement vector from the previous optimized point on the path, $\mathbf{x}(0) - \mathbf{x}(-s)$. Substitution of $\mathbf{v}^0(0)$, $\mathbf{x}(-s)$ [from Eq. (2)] and $\mathbf{v}^0(s)$ into Eq. (15) gives

$$\begin{aligned}\mathbf{x}(s) = \mathbf{x}(0) &+ \sigma_1 \mathbf{v}^0(0) \\ &+ \sigma_2 \left[s\mathbf{v}^0(0) - \tfrac{1}{2}s^2\mathbf{v}^1(0) + \tfrac{1}{6}s^3\mathbf{v}^2(0) - \tfrac{1}{24}s^4\mathbf{v}^3(0) + \cdots\right] \\ &+ \sigma_3 \left[\mathbf{v}^0(0) + s\mathbf{v}^1(0) + \tfrac{1}{2}s^2\mathbf{v}^2(0) + \tfrac{1}{6}s^3\mathbf{v}^3(0) + \cdots\right].\end{aligned} \tag{23}$$

From comparison with Eq. (1) the values for the σ's are

$$\sigma_1 = \tfrac{4}{5}s, \quad \sigma_2 = -\tfrac{1}{5}, \quad \sigma_3 = \tfrac{2}{5}s. \tag{24}$$

The error in the fourth-order term is $\frac{17}{120}s^4\mathbf{v}^3(0)$. As in method *B*, the Hessian is not needed and no additional energy or gradient computations are required beyond those for the second-order algorithm.

D. Fourth-order algorithms

Method D

A fourth-order method can be constructed by combining methods *B* and *C*, i.e., by using the tangent at the previous point as well as the displacement to the previous point:

$$\mathbf{x}(s) = \mathbf{x}(0) + \sigma_0[\mathbf{x}(0) - \mathbf{x}(-s)] + \sigma_1 \mathbf{v}^0(-s) + \sigma_2 \mathbf{v}^0(0) + \sigma_3 \mathbf{v}^0(s). \tag{25}$$

Substitution and expansion gives

$$\begin{aligned}\mathbf{x}(s) = \mathbf{x}(0) &+ \sigma_0\left[s\mathbf{v}^0(0) - \tfrac{1}{2}s^2\mathbf{v}^1(0) + \tfrac{1}{6}s^3\mathbf{v}^2(0) - \tfrac{1}{24}s^4\mathbf{v}^3(0) + \tfrac{1}{120}s^5\mathbf{v}^4(0) + \cdots\right] \\ &+ \sigma_1\left[\mathbf{v}^0(0) - s\mathbf{v}^1(0) + \tfrac{1}{2}s^2\mathbf{v}^2(0) - \tfrac{1}{6}s^3\mathbf{v}^3(0) + \tfrac{1}{24}s^4\mathbf{v}^4(0) + \cdots\right] + \sigma_2 \mathbf{v}^0(0) \\ &+ \sigma_3\left[\mathbf{v}^0(0) + s\mathbf{v}^1(0) + \tfrac{1}{2}s^2\mathbf{v}^2(0) + \tfrac{1}{6}s^3\mathbf{v}^3(0) + \tfrac{1}{24}s^4\mathbf{v}^4(0) + \cdots\right].\end{aligned} \tag{26}$$

Comparison with Eq. (1) leads to the following values for the σ's:

$$\sigma_0 = -1, \quad \sigma_1 = \tfrac{1}{3}s, \quad \sigma_2 = \tfrac{4}{3}s, \quad \sigma_3 = \tfrac{1}{3}s. \tag{27}$$

The error in the fifth-order term is $\frac{7}{720}s^5\mathbf{v}^4(0)$. Equation (27) can be rewritten in a more symmetrically form by stepping from $\mathbf{x}(-s)$ rather than $\mathbf{x}(0)$,

$$\mathbf{x}(s) = \mathbf{x}(-s) + \tfrac{1}{3}s\mathbf{v}^0(-s) + \tfrac{4}{3}s\mathbf{v}^0(0) + \tfrac{1}{3}s\mathbf{v}^0(s). \tag{28}$$

In this form, it is clear that method D is actually Simpson's rule for integration (it is also the implicit version of the modified midpoint method with extrapolation). Even though this is a fourth-order algorithm, the Hessian is not needed and no additional energy or gradient computations are needed beyond those required for the second-order algorithm or the third order methods B and C.

Method E

Another fourth-order method can be constructed by using the curvature at the current point and the tangents at $\mathbf{x}(-s)$, $\mathbf{x}(0)$ and $\mathbf{x}(s)$,

$$\mathbf{x}(s) = \mathbf{x}(0) + \sigma_1\mathbf{v}^0(-s) + \sigma_2\mathbf{v}^0(0) + \sigma_3\mathbf{v}^1(0) + \sigma_4\mathbf{v}^0(s). \tag{29}$$

Substitution for $\mathbf{v}^0$ and $\mathbf{v}^1$, and comparison with Eq. (1) gives

$$\sigma_1 = \tfrac{1}{24}s, \quad \sigma_2 = \tfrac{16}{24}s, \quad \sigma_3 = \tfrac{6}{24}s^2, \quad \sigma_4 = \tfrac{7}{24}s. \tag{30}$$

The error in fifth-order term is $\frac{1}{72}s^5\mathbf{v}^4(0)$.

Method F

One of the strengths of the second-order method is that it uses information only from the beginning and end points of the current step, and does not rely on information from previous steps. As illustrated in Fig. 1(c), an analogous fourth-order method can be constructed from $\mathbf{v}^0(0)$, $\mathbf{v}^1(0)$, $\mathbf{v}^0(s)$, and $\mathbf{v}^1(s)$,

$$\mathbf{x}(s) = \mathbf{x}(0) + \sigma_1\mathbf{v}^0(0) + \sigma_2\mathbf{v}^1(0) + \sigma_3\mathbf{v}^0(s) + \sigma_4\mathbf{v}^1(s). \tag{31}$$

Substitution for $\mathbf{v}^0$ and $\mathbf{v}^1$, and comparison with Eq. (1) gives

$$\sigma_1 = \sigma_3 = \tfrac{1}{2}s, \quad \sigma_2 = -\sigma_4 = \tfrac{1}{12}s^2. \tag{32}$$

The error in the fifth-order term is $\frac{1}{144}s^5\mathbf{v}^4(0)$. Gear[9] has discussed this method as a member of a class of A-stable methods. Method F can also be obtained as the average of a forward step from $\mathbf{x}(0)$ to $\mathbf{x}(s)$ and a backward step from $\mathbf{x}(s)$ to $\mathbf{x}(0)$ using method A. Implementation of method F poses some problems, since $\mathbf{v}^1(s)$ must be calculated iteratively and the Hessian is needed to calculate $\mathbf{v}^1(s)$. To avoid repeated (expensive) calculations of the Hessian, the Hessian can be assumed to be constant in a small region near $\mathbf{x}(s)$ or could be updated by any of the techniques used by quasi-Newton optimization algorithms. Method A can be used to get an approximate value for $\mathbf{x}(s)$; the Hessian is calculated once at this point and Eq. (31) is solved iteratively using this fixed Hessian and Eq. (6) to calculate $\mathbf{v}^1(s)$ [$\mathbf{v}^0(s)$ is still calculated by a constrained optimization]. If $\mathbf{v}^1$ is available at the transition state, method F can be used to take the first step away from the transition state; otherwise, the second-order algorithm can be used to take the first step.

E. Higher-order algorithms

Method G

If the Hessian is available at each optimized point on the reaction path, it is possible to construct a fifth-order algorithm that takes advantage of all the available information. The new point on the reaction path is written as the current position and a linear combination of the tangent and curvature vectors at the current and previous points and $\mathbf{v}^0(s)$,

$$\mathbf{x}(s) = \mathbf{x}(0) + \sigma_1\mathbf{v}^0(-s) + \sigma_2\mathbf{v}^1(-s) + \sigma_3\mathbf{v}^0(0) + \sigma_4\mathbf{v}^1(0) + \sigma_5\mathbf{v}^0(s). \tag{33}$$

Equations (2) and (3) are used to expand $\mathbf{v}^0(\pm \mathrm{s})$ and $\mathbf{v}^1(\pm \mathrm{s})$,

$$\begin{aligned}\mathbf{x}(s) = \mathbf{x}(0) &+ \sigma_1[\nu^0(0) - s\nu^1(0) + \tfrac{1}{2}s^2\nu^2(0) - \tfrac{1}{6}s^3\nu^3(0) + \tfrac{1}{24}s^4\nu^4(0) + \cdots]\\ &+ \sigma_2[\nu^1(0) - s\nu^2(0) + \tfrac{1}{2}s^2\nu^3(0) - \tfrac{1}{6}s^3\nu^4(0) + \cdots]\\ &+ \sigma_3\nu^0(0) + \sigma_4\nu^1(0) + \sigma_5[\nu^0(0) + s\nu^1(0) + \tfrac{1}{2}s^2\nu^2(0) + \tfrac{1}{6}s^3\nu^3(0) + \tfrac{1}{24}s^4\nu^4(0) + \cdots].\end{aligned} \tag{34}$$

Comparing with Eq. (1) and requiring terms in $\mathbf{v}^0, \mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3$, and $\mathbf{v}^4$ to be correct gives

$$\sigma_1 = \tfrac{25}{120}s, \quad \sigma_2 = \tfrac{8}{120}s^2, \quad \sigma_3 = \tfrac{64}{120}s, \quad \sigma_4 = \tfrac{46}{120}s^2, \quad \sigma_5 = \tfrac{31}{120}s. \tag{35}$$

Method H

A sixth-order method can be constructed by combining methods F and G, i.e., using the tangent and curvature from $\mathbf{x}(-s)$, $\mathbf{x}(0)$, and $\mathbf{x}(s)$,

$$\mathbf{x}(s) = \mathbf{x}(0) + \sigma_1\mathbf{v}^0(-s) + \sigma_2\mathbf{v}^1(-s) + \sigma_3\mathbf{v}^0(0) + \sigma_4\mathbf{v}^1(0) + \sigma_5\mathbf{v}^0(s) + \sigma_6\mathbf{v}^1(s). \tag{36}$$

Substitution for $\mathbf{v}^0$ and $\mathbf{v}^1$, and comparison with Eq. (1) gives

$$\sigma_1 = \tfrac{11}{240}s, \quad \sigma_2 = \tfrac{3}{240}s^2, \quad \sigma_3 = \tfrac{128}{240}s, \quad \sigma_4 = \tfrac{25}{240}s^2, \quad \sigma_5 = \tfrac{101}{240}s, \quad \sigma_6 = -\tfrac{13}{240}s^2. \tag{37}$$

Some caution is needed with both of these higher-order methods. The curvature is much more difficult to compute accurately than the position and tangent of the reaction path. Therefore methods G and H may be less stable than the third- and fourth-order methods discussed earlier.

III. APPLICATIONS

The Müller–Brown surface has been used in a number of previous studies to test reaction path following algorithms:

$$\begin{aligned} E(x,y) &= \sum_{i=1}^{4} A_i \exp\left[a_i(x-x_i^0)^2 \right. \\ &\quad \left. + b_i(x-x_i^0)(y-y_i^0) + c_i(y-y_i^0)^2\right], \\ A &= (-200,\ -100,\ -170,\ 15), \\ x^0 &= (1.0,\ 0.0,\ -0.5,\ -1.0), \\ y^0 &= (0.0,\ 0.5,\ 1.5,\ 1.0), \\ a &= (-1.0,\ -1.0,\ -6.5,\ 0.7), \\ b &= (0.0,\ 0.0,\ 11.0,\ 0.6), \\ c &= (-10.0,\ -10.0,\ -6.5,\ 0.7). \end{aligned} \tag{38}$$

Although this is only a two-dimensional surface, the curved reaction path is difficult to follow accurately with a large step size, and can serve as a suitable test of the methods discussed earlier. The true path on the Müller–Brown surface was computed by the Euler method with a small step size (0.01). Except for gross inadequacies in an algorithm, path following ability is not a sensitive indicator of the accuracy of an algorithm. A much more demanding test is the ability of an algorithm to yield a path with the correct curvature. All of the methods were tested using a step size of 0.15. The necessary numerical and symbolic manipulations were carried out using Mathematica.[10]

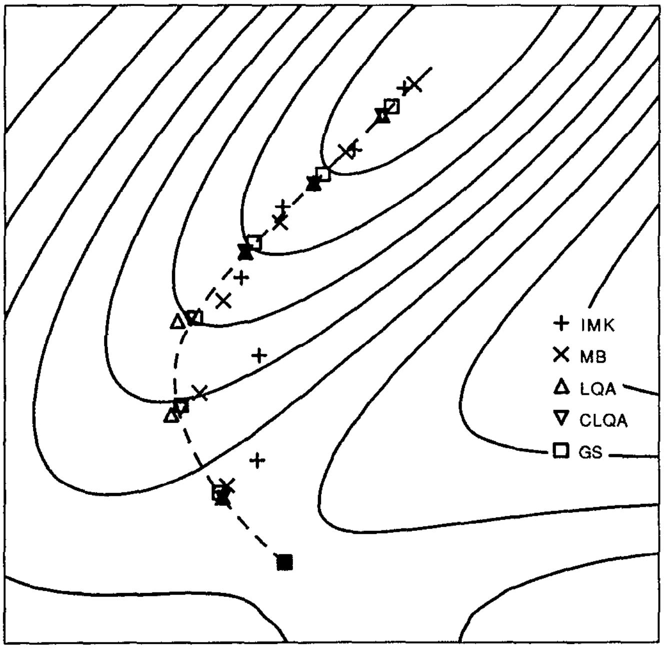

FIG. 2. Performance of some low-order reaction path following algorithms on the Müller-Brown surface (+, IMK; ×, MB; Δ, LQA; ∇, CLQA; □, GS).

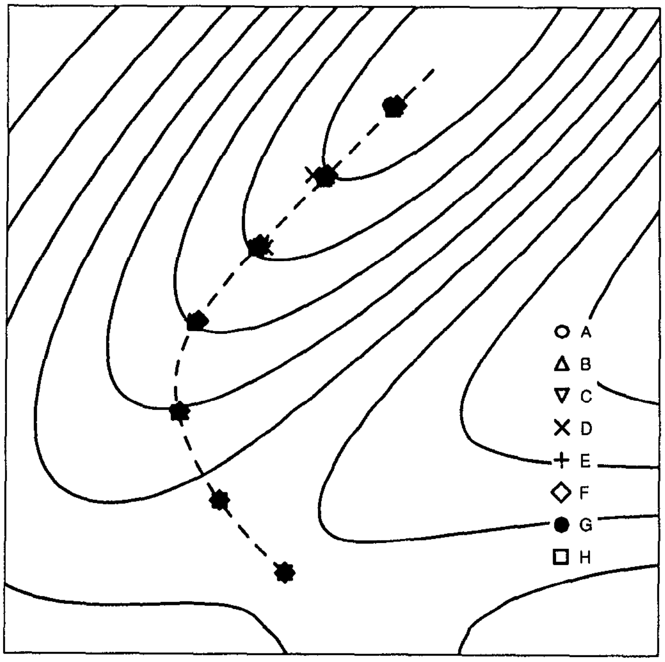

FIG. 3. Performance of some higher-order reaction path following algorithms on the Müller–Brown surface (○, method A; Δ, method B; ∇, method C; ×, method D; +, method E; ◇, method F; ●, method G; □, method H). On the resolution of this plot, all methods except D are equally good at reproducing the position of points the reaction path.

Figure 2 compares the path following ability of the low-order methods. The IMK method[5] has serious difficulties when step sizes as large as 0.15 are used. The Müller–Brown method[6] cuts the corner. The LQA method[7] is slightly wide of the true path; the CLQA method[7] is closer to the path (both were started on the true path at $s = 0.15$ rather than at the transition state). Our second-order method[8] (which

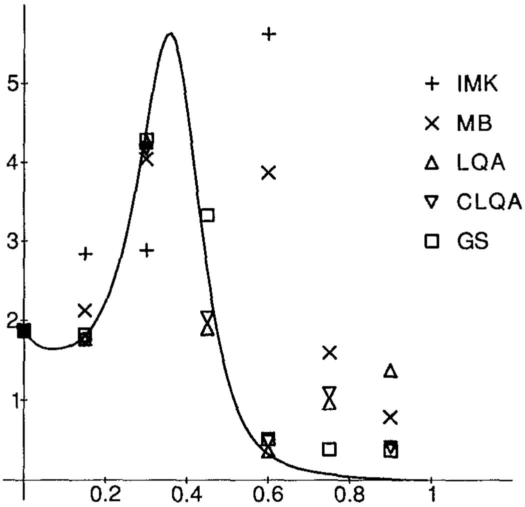

FIG. 4. Magnitude of the curvature along the reaction path on the Müller–Brown surface computed by lower-order methods (+, IMK; ×, MB; Δ, LQA; ∇, CLQA; □, GS).

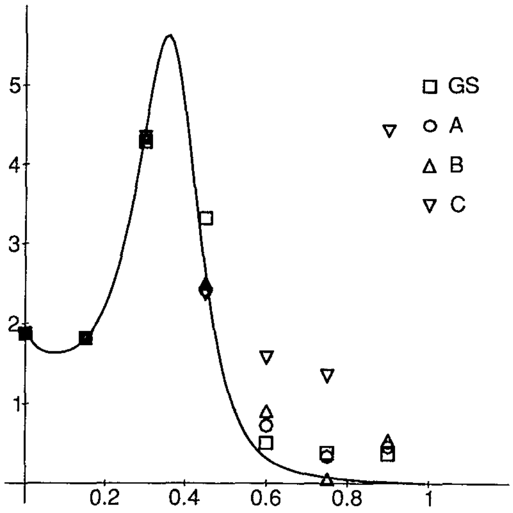

FIG. 5. Magnitude of the curvature along the reaction path on the Müller–Brown surface computed by second- and third-order methods (□, second-order method; ○, method *A*; △, method *B*; ▽, method *C*).

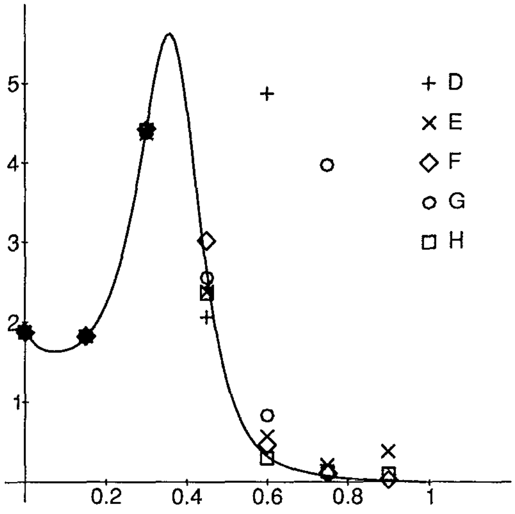

FIG. 6. Magnitude of the curvature along the reaction path on the Müller–Brown surface computed by fourth- and higher-order methods (+, method *D*; ×, method *E*; ◇, method *F*; ○, method *G*; □, method *H*).

needs only gradients) follows the path as closely as the CLQA method (which requires the Hessian and estimates of components of the cubic force constants).

The third-, fourth-, and higher-order algorithms proposed in the present paper are compared in Fig. 3. The second-order method was used for the first step away from the transition state for methods *A–E* and *G*; method *F* was used to start method *H*. Except for method *D*, all methods appear to be equally good on the scale of the figure.

Given a point on an approximate reaction path, the curvature can be computed using Eq. (6). The calculated magnitudes of the curvature vectors for the low-order algorithms

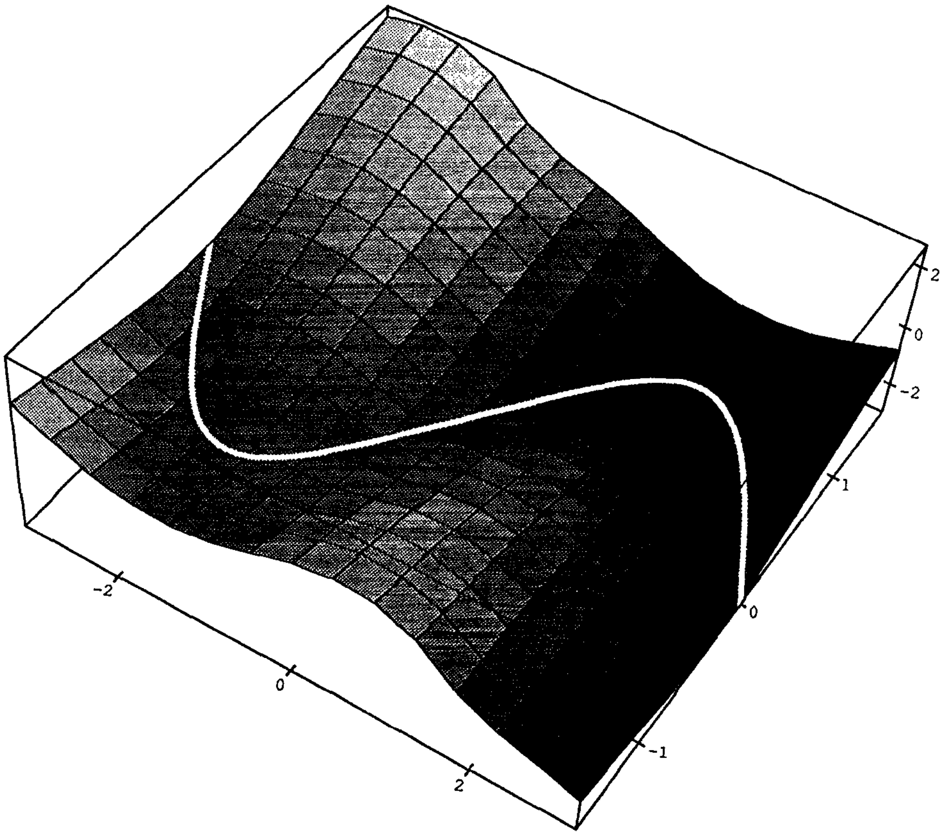

FIG. 7. Model surface with an analytical reaction path given by Eqs. (39) and (40).

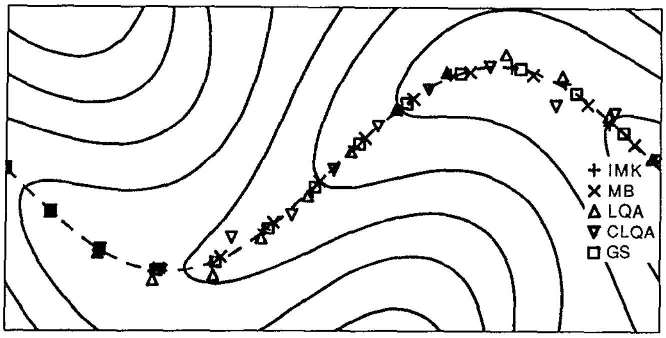

FIG. 8. Performance of some low-order reaction path following algorithms on the surface given by Eq. (39) (+, IMK; ×, MB; Δ, LQA; ∇, CLQA; □, GS).

are shown in Fig. 4 along with the magnitude of the curvature for the true path. The IMK and MB methods perform poorly, as could be anticipated from their strong deviation from the true path. The LQA, CLQA, and the second-order algorithms are equally good prior to the peak in curvature; the CLQA and second-order methods are better than LQA after the peak. All of these methods have some problems in the final straight region.

Figure 5 shows the curvatures computed for the second- and third-order methods. The second-order method and third-order methods *A* and *B* are about equal; method *C* is unsatisfactory. The higher-order methods *D* through *H* are compared in Fig. 6. Methods *D* and *G* are clearly inadequate. Method *E* performs about as well as methods *A* and *B* and the second-order algorithm. Based on their ability to reproduce the curvature toward the end of the reaction path, methods *F* and *H* appear to be the best algorithms.

The second model surface used to test the reaction path following algorithms is given by Eq. (39) and is shown in Fig. 7:

$$E(x,y) = \operatorname{arccot}\left[- e^{y} \cot\left(\frac{x}{2} - \frac{\pi}{4}\right)\right] - 2\exp[- (y - \sin x)^2/2]. \qquad (39)$$

This rather unpleasant looking function[11] has the highly desirable feature of possessing a simple, analytical form for the reaction path and the curvature,

$$y = \sin x, \quad \kappa^2 = \sin^2 x/(1 + \cos^2 x)^3. \qquad (40)$$

By scaling *both* the *x* and *y* coordinates, any desired maximum curvature can be obtained (i.e., $x' = x/b$, $y' = y/b$, $\kappa' = b\kappa$). If necessary, the reaction path can be made more difficult to follow (i.e., the differential equation can be made stiffer) by increasing the exponent in the Gaussian term.

For a stepsize of 0.6, the results for the various reaction path following algorithms on the surface given by Eq. (39) are qualitatively similar to the Müller–Brown surface. As shown in Fig. 8, the low-order techniques all follow the path quite well except for LQA and CLQA. For the higher-order methods presented in Fig. 9, only methods *D* and *G* have difficulty in following the path. The plots of the magnitude

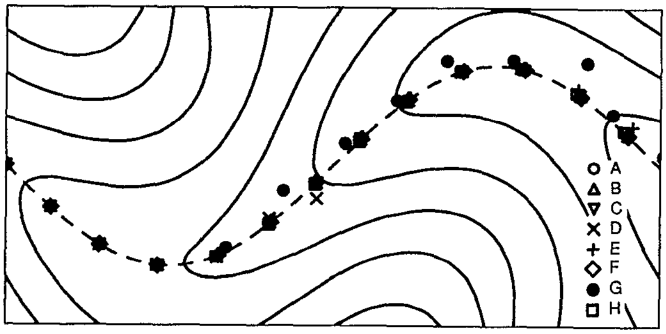

FIG. 9. Performance of some higher-order reaction path following algorithms on the surface given by Eq. (39) (O, method *A*; Δ, method *B*; ∇, method *C*; ×, method *D*; +, method *E*; ◇, method *F*; ●, method *G*; □, method *H*).

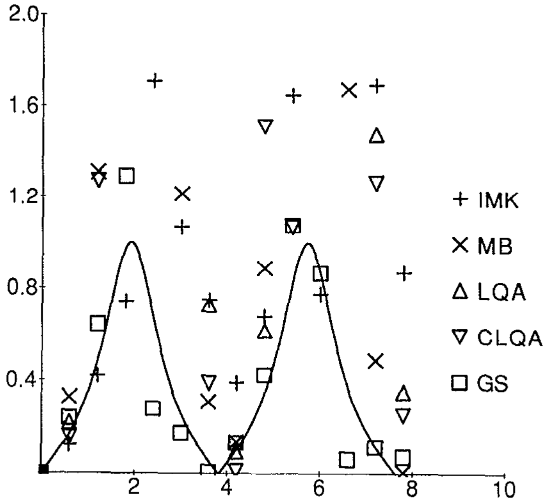

FIG. 10. Magnitude of the curvature along the reaction path on the surface given by Eq. (39) computed by lower-order methods (+, IMK; ×, MB; Δ, LQA; ∇, CLQA; □, GS).

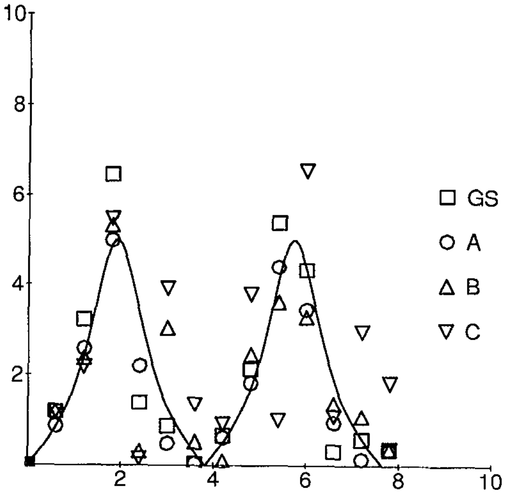

FIG. 11. Magnitude of the curvature along the reaction path on the surface given by Eq. (39) computed by second- and third-order methods (□, second-order method; O, method *A*; Δ, method *B*; ∇, method *C*).

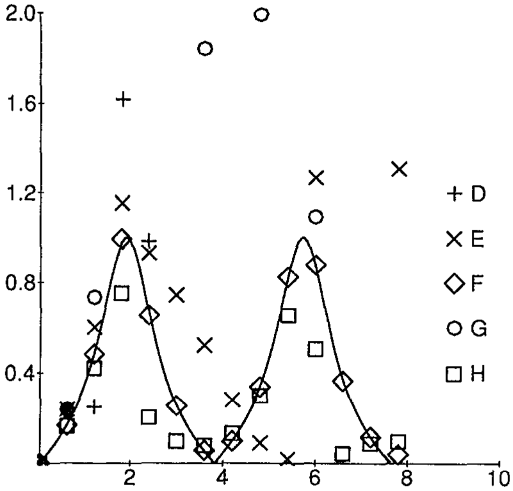

FIG. 12. Magnitude of the curvature along the reaction path on the surface given by Eq. (39) computed by fourth- and higher-order methods (+, method D; ×, method E; ◇, method F; ○, method G; □, method H).

of the curvature vs the path length reveal that this surface is more demanding than the Müller–Brown surface. Even with a stepsize of 0.002, the Euler method gives a maximum in the curvature that is about 2% too low. As illustrated in Fig. 10, the IMK, MB, LQA, and CLQA methods yield a fairly random scatter of values for the curvature. The second-order algorithm is qualitatively correct (i.e., two peaks) but is 40% too high at the first maximum. Figure 11 shows that method A is slightly better than the second-order algorithm and that both are better than methods B and C. Of the higher-order algorithms shown in Fig. 12, D, E, and G are poorer than both method A and the second-order method. Method H is somewhat better; however, the only algorithm that reproduces the curvature satisfactorily at this stepsize is method F. With larger stepsizes (e.g., 0.9) the superiority of method F is even more evident.

IV. CONCLUSIONS

The tests of the proposed path following methods on the model surfaces have once again illustrated that formal order is not necessarily an appropriate measure to judge an algorithm. The Müller–Brown surface illustrates that second-order method is better than methods C (third order), D (fourth order), and G (fifth order). Apparently, using the position of the prior point on the reaction path degrades the performance of reaction path following. The second-order method is comparable to methods B (third order) and E (fourth order); thus, use of the tangent from the prior point appears to be more stable than use the displacement. The third-order method A, which requires one Hessian calculation for each step along the path, is generally similar to or slightly better than the second-order algorithm, which uses only the gradient. The fourth-order method F and its extension, sixth-order method H (both of which require one Hessian calculation per step), seem to be the best at reproducing the curvature for reasonable stepsizes. However, method F is better for the large steps taken on the surface given by Eq. (39). Both methods are *implicit* in the tangent and the curvature. This suggests that using the curvature at the endpoints of the step is the underlying reason for the good performance of methods F and H, rather than the higher order of these methods. Method F has an elegant simplicity and symmetry to it; the fact that no information from prior steps is used means that method F is self-starting and is not susceptible to numerical problems caused by rapid changes in prior steps along the path. Method F, along with a number of the other methods, will be coded into *ab initio* electronic structure codes so that these methods can be tested on higher-dimensional reaction paths.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation Grant No. CHE 90-20398.

[1] K. Fukui, Acc. Chem. Res. **14**, 363 (1981).

[2] D. G. Truhlar and M. S. Gordon, Science **249**, 491 (1990).

[3] B. C Garrett, M. J. Redmon, R. Steckler, D. G. Truhlar, K. K. Baldridge, D. Bartol, M. W. Schmidt, and M. S. Gordon, J. Phys. Chem. **92**, 1476 (1988).

[4] K. K. Baldridge, M. S. Gordon, R. Steckler, and D. G. Truhlar, J. Phys. Chem. **93**, 5107 (1989).

[5] K. Ishida, K. Morokuma, and A. Komornicki, J. Chem. Phys. **66**, 2153 (1977); M. W. Schmidt, M. S. Gordon, and M. Dupuis, J. Am. Chem. Soc. **107**, 2585 (1985).

[6] K. Müller and L. D. Brown, Theor. Chim. Acta **53**, 75 (1979).

[7] M. Page and J. W. McIver, J. Chem. Phys. **88**, 922 (1988); M. Page, C. Doubleday, and J. W. McIver, *ibid.* **93**, 5634 (1990).

[8] C. Gonzalez and H. B. Schlegel, J. Chem. Phys. **90**, 2154 (1989); C. Gonzalez and H. B. Schlegel, J. Phys. Chem. **94**, 5523 (1990).

[9] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1971).

[10] S. Wolfram, *Mathematica* (Addison-Wesley, Reading, Massachusetts, 1988), and associated computer programs.

[11] Some caution is needed in evaluating this function to ensure numerical accuracy [e.g., near $x = (2n + 1)\pi/2$] and continuity (i.e., by adding the appropriate multiples of π to the inverse trigonometric function).