CHAPTER

17

Molecular structure optimizations with Gaussian process regression

*Roland Lindh** *and Ignacio Fdez. Galván*

Department of Chemistry—BMC, Uppsala University, Uppsala, Sweden

***Corresponding author: E-mail address: roland.lindh@kemi.uu.se**

Abstract

Molecular structure optimization is one of the most common tasks performed in computational chemistry. Many applications require locating special points in a potential energy surface: minima, saddle points, and others. Given that the calculation of energies and gradients with accurate quantum chemical methods is quite computationally demanding, one is often interested in finding these special points with a minimal number of energy evaluations. During the last decades, optimization methods and strategies based on a second-order expansion of the potential energy surface have been developed and perfectioned, reaching a high level of efficiency and robustness. These "conventional" methods are briefly described in this chapter. More recently, alternative models and methods applying machine learning techniques (and most significantly Gaussian process regression) are being proposed and developed, and already show superior characteristics with respect to the established methods. These new approaches are discussed, in particular the restricted variance optimization method is described in some detail. Practical examples include optimization of stable structures and transition states.

Keywords: Geometry optimization, Gaussian process regression, Constrained optimization, Transition state, Potential energy surface, Surrogate model

Introduction

The Born-Oppenheimer approximation is a fundamental concept in computational chemistry. On the one hand, it introduces a semiseparation of coordinates—electronic and nuclear coordinates—facilitating efficiency in solving the Schrödinger equation as two separate problems, on the other hand, it introduces the notion of the potential energy surface (PES). The latter forms the pedestal of the way in which chemists think about chemical reactivity—a molecule's properties and reactivity are closely associated with the molecular structure and the

https://doi.org/10.1016/B978-0-323-90049-2.00017-2

corresponding local topology of the PES. This is true for both ground-state and excited-state chemistry. In that sense it is instrumental, to theoretical simulations of equilibrium and reaction properties—either thermally or photo-activated—to possess computational tools for an efficient exploration of PES. Since the development of analytical gradients in computational chemistry, an ongoing parallel development of methods to locate equilibrium and transition state structures, or to explore particular parts of the PES through constrained optimizations, as in reaction paths, has been in progress. These developments have almost uniquely been restricted to second-order methods utilizing approximate Hessians—a type of quasi-Newton approach. In that respect researchers have tried to optimize the selection of coordinates, explored different types of second-order optimization methods, proposed various measures to produce accurate approximate Hessians, or to design Hessian update methods appropriate for various types of explorations of the PES. It should, however, be noted that these methods are nowadays very mature and significant new developments are not frequently reported. That is, until recently, machine learning (ML) techniques started to be used as an alternative to the traditional second-order methods as a surrogate model. The ML methods have several features which will trump any second-order method, for example, they can include several stationary points at the same time, the methods do not require any Hessian-update procedure, the surrogate model will converge to the parent model as the number of iterations increases, it will include anharmonic characteristics, and it will provide analytic estimates of the expected discrepancy between the parent and surrogate model at any point in the geometry space of any given molecule, as well as directly model numerical noise in the input data via hyperparameters. ML optimizations, however, are in their infancy and have several issues that need to be ironed out, for example, the selection of coordinates, the optimization of the associated hyperparameters of the ML procedure, and the use of the dispersion estimate, to mention a few. Having said that it is fair to say that recent developments nowadays describe robust and stable ML techniques performing in parity with or most often superior to conventional optimization methods.

Hence, we believe that the field is mature enough to deserve a subsummary of the ongoing activities and present it in such a way that the matter will be exposed to more researchers and students in the field of computational chemistry. Considering that the ML methods have just recently been incorporated to the arsenal of tools used by theoretical chemistry, we hope that a review like this will also inspire other researchers to explore the potentials of ML in other aspects of computer simulations of the chemical reactivity. Thus, in this chapter, we present a general review of the state of the art of the field and describe new recent developments. Toward the end of the review we give some detailed case studies employing the so-called restricted variance optimization (RVO) approach in order to demonstrate the flexibility and power of the integration of ML technique in as mundane tasks as molecular structure optimizations and PES explorations.

Methods

This section is subdivided into three parts. The first part is to set the stage for a comparison between established methods and ML-inspired approaches. That is, it will describe standard methods for molecular structure optimizations as gradually developed over the last 50 years.

This will also introduce some concepts which are of essence in the development of ML-supported procedures. The presentation will be a brief yet sufficiently detailed review of conventional techniques, which are used in association with optimizations of equilibrium and transition state structures, reaction paths, and combined with geometrical constraints. For a perhaps more verbose and complete presentation of the subject we direct the eager reader to the excellent review by Schlegel [1]. In the second part, we discuss pure ML methods which will directly produce molecular structures, without any reference to a parent model running in parallel to support the surrogate model with data. Finally, in the third section, we review the current developments of the incorporation of ML methods in molecular geometry optimizations. This section will be ended by a more detailed description of the implementation of a Gaussian process regression (GPR) method in terms of RVO [2, 3]. As the presentation moves along, we will contrast the performance details of the various methods against each other.

Established methods for molecular optimizations

Given a PES, we have a more or less complicated function which yields an energy as a function of molecular structure—the parent model. The standard molecular structure optimization methods, designed by computational chemists, have been developed in the light of the fact that in conventional ab initio calculations energy and gradient calculations are of comparable computational expense while higher derivatives are out of reach for frequent evaluation and use in iterative procedures. In that respect, the optimization paradigms implemented are, by and large, those that use the analytic values of the energy and gradient of the parent model. This would, however, imply strict first-order optimization methods. Here, though, the general convergence rate is rather poor. Higher-order methods would make much more sense in improved performance and general robustness of the procedure. However, the application of analytic Hessian (second derivatives) is, in general, out of the question. Hence the compromise: computational chemists have generally based their optimization methods on procedures on analytic values of the energies and gradients, and approximative estimates of higher derivatives. In particular, the use of second-order surrogate models in connection with Hessian update methods—the quasi-Newton method [4, 5]—is today the established optimization strategy. In this context we will touch on some of the standards that have been developed over the years in the next section. Especially, we will discuss various versions of the quasi-Newton procedure, step restrictions, the selection of coordinates, procedures to generate the estimates of the Hessian, Hessian-update methods, and techniques for constrained optimizations. At the very end of this section, we will also briefly discuss the geometry optimization using direct inversion in the iterative subspace (GDIIS) technique.

Quasi-Newton methods

As mentioned earlier, molecular geometry optimizations employ a surrogate model based on a second-order expansion—a quadratic model, traditionally in the form of a truncated Taylor expansion

$$E(\boldsymbol{q}) = E(\boldsymbol{q}_0) + \boldsymbol{g}(\boldsymbol{q}_0)^{\mathrm{t}}\Delta\boldsymbol{q} + \frac{1}{2}\Delta\boldsymbol{q}^{\mathrm{t}}\boldsymbol{H}(\boldsymbol{q}_0)\Delta\boldsymbol{q} \tag{1}$$

where $\boldsymbol{q}$ denotes an arbitrary molecular structure expressed in some coordinates, $\boldsymbol{g}(\boldsymbol{q}_0)$ and $\boldsymbol{H}(\boldsymbol{q}_0)$ are the first derivative (in the form of a column vector) and the approximative second derivative (in the form of a matrix), respectively, with respect to the coordinates of the molecular structure evaluated at an arbitrary reference structure, $\boldsymbol{q}_0$, and $\Delta\boldsymbol{q} = \boldsymbol{q} - \boldsymbol{q}_0$. This surrogate model is a pragmatic choice which will, more or less, coerce optimizations to convergence. The model is accurate close to the point of expansion; however, it has a number of limitations as follows: (i) it cannot describe anharmonic characteristics of the PES, (ii) it cannot describe several stationary points simultaneously, (iii) it will not converge toward the parent model as the number of data points (iterations) increases, and (iv) it will not give an analytic estimate of the range within which the model is accurate given an acceptable allowed error.

The Taylor-like quadratic surrogate model has been used with some success over the years; however, an alternative quadratic approach, the rational function and rational function optimization (RFO) [6], has demonstrated superior performance. Here the surrogate model is a rational function with quadratic polynomials as numerator and denominator,

$$E(\boldsymbol{q}) = E(\boldsymbol{q}_0) + \frac{1}{2}\frac{(1\ \Delta\ \boldsymbol{q}^{\mathrm{t}})\begin{pmatrix} 1 & \boldsymbol{g}^{\mathrm{t}} \\ \boldsymbol{g} & \boldsymbol{H} \end{pmatrix}\begin{pmatrix} 1 \\ \Delta\boldsymbol{q} \end{pmatrix}}{(1\ \Delta\ \boldsymbol{q}^{\mathrm{t}})\begin{pmatrix} 1 & \boldsymbol{0}^{\mathrm{t}} \\ \boldsymbol{0} & \boldsymbol{S} \end{pmatrix}\begin{pmatrix} 1 \\ \Delta\boldsymbol{q} \end{pmatrix}} \tag{2}$$

where the symmetric matrix $\boldsymbol{S}$ is usually set to the unit matrix $\boldsymbol{I}$. Note that the matrix product on the numerator, together with the $\frac{1}{2}$ factor, is nothing more than a rewrite of the second and third terms of Eq. (1), and that the denominator becomes $(1 + |\Delta\boldsymbol{q}|^2)$ when $\boldsymbol{S} = \boldsymbol{I}$. The success of the method is attributed to the denominator that brings in a semianharmonic characteristic in the model—the potential does not increase indefinitely as a pure harmonic potential, but is tapered to a finite value.

Both of these surrogate models have been implemented not only for standard equilibrium optimizations but also for optimizations of transition states (TS) and constrained optimizations. In the case of transition state optimizations two modifications have been presented, the decoupled optimization and the coupled optimization. In the decoupled optimization the combined minimization in $3N - 7(6)$ directions (N being the number of atoms) is decoupled from the maximization in the direction of the reaction coordinate, partitioned rational function optimization (P-RFO) [7]. An alternative is to use the so-called image function technique of Smith [8], which changes the sign of the components of the gradient and Hessian in the direction of the reaction coordinate. This was first used by Helgaker [9] in combination with RFO, denoted as image rational function optimization (I-RFO) and demonstrated that a coupled approach, which is strictly a minimization in all degrees of freedom, is a very efficient method for determining the location of TS. The current status of TS optimizations is that performance is closely connected to the qualitative accuracy of the Hessian—Is the separation of the reaction coordinate and the complementary coordinate space accurate enough?

Step restriction

The very fact that the quadratic surrogate model is an approximation, for which the disagreement with the parent model presumably increases with the distance to the reference structure of the molecule, means that some care has to be taken in order to avoid a large step

which most likely corresponds to a molecular structure for which the surrogate model produces poor predictions of the energy. In this respect, a safety measure has been introduced—the step restriction procedure. This approach simply says that predicted corrections of a molecular structure should have a length, $l(\boldsymbol{q})$ (usually evaluated in terms of the Cartesian coordinates), that is not longer than a given threshold value, τ. Older implementations of this strategy simply scaled the suggested step such that the length of the step is not longer than the length threshold. However, a more prudent approach is that, in the case the predicted step is too long, a constrained optimization is conducted such that an optimal structure with a step length of precisely τ is obtained [10]. This has been implemented for both standard second-order Newton-Raphson (NRO) and rational function optimizations, giving rise to restricted-step versions of both methods—RS-NRO and RS-RFO [11, 12]. While in the former case the step restriction is implemented as a constrained optimization with Lagrange multipliers,

$$L(\boldsymbol{q}) = E(\boldsymbol{q}_0)+\boldsymbol{g}(\boldsymbol{q}_0)^{\mathrm{t}}\Delta\boldsymbol{q} + \frac{1}{2}\Delta\boldsymbol{q}^{\mathrm{t}}\boldsymbol{H}(\boldsymbol{q}_0)\Delta\boldsymbol{q} + \lambda(l(\boldsymbol{q}) - \tau) \tag{3}$$

the RS-RFO implementation is a bit more elaborate. Here the goal is achieved by multiplying the $\boldsymbol{S}$ matrix, in Eq. (2), with a factor α,

$$E(\boldsymbol{q}) = E(\boldsymbol{q}_0)+\frac{1}{2}\frac{(1\ \Delta\ \boldsymbol{q}^{\mathrm{t}})\begin{pmatrix}1 & \boldsymbol{g}^{\mathrm{t}}\\ \boldsymbol{g} & \boldsymbol{H}\end{pmatrix}\begin{pmatrix}1\\ \Delta\boldsymbol{q}\end{pmatrix}}{(1\ \Delta\ \boldsymbol{q}^{\mathrm{t}})\begin{pmatrix}1 & \boldsymbol{0}^{\mathrm{t}}\\ \boldsymbol{0} & \alpha\boldsymbol{S}\end{pmatrix}\begin{pmatrix}1\\ \Delta\boldsymbol{q}\end{pmatrix}} \tag{4}$$

and varying this factor in a controlled fashion until the desired step length of τ is obtained [12]. The usefulness of this approach rests, of course, on the selection of the step restriction threshold, τ. It is clear that this value has to be selected in an ad hoc manner, since we do not have any knowledge in advance on the error of the surrogate model compared to the parent model. That is, at best we can provide a reasonable guess of what the value should be, based on the discrepancy between the parent and the surrogate model in the past iterations. Note, there is no explicit control or estimate of the validity of the model for any new structure.

Approximate Hessian

The quasi-Newton method relies on an approximate Hessian in combination with Hessian-update methods. This approximate Hessian can be computed either using standard molecular mechanics or semiempirical methods, or with some (possibly simplified) ab initio approach [13–15]. For example, a Hessian derived from approximations to the SCF method has been suggested [16]. However, a favorite estimation procedure for the generation of an approximate Hessian is the Hessian model function (HMF) by Lindh et al. [17]. The approach relies on very few parameters and is for that very reason rather trivial to implement from scratch. Moreover, it is applicable to any kind of covalent type of bonding situation.

It is fair to say that some of these approximations are very crude. However, with the combination of Hessian-update methods, it is sufficient that the approximate Hessian generates an accurate enough separation of soft and stiff modes. After that the Hessian-updated quasi-Newton method will, after very few extra iterations, reach the same final structure as if an analytic Hessian had been used. In this respect, from a CPU time perspective, a few extra

iterations of computing the energy and gradients will be significantly more efficient than computing the analytic Hessian.

Hessian-update methods

Hessian-update methods are procedures to correct an approximate Hessian such that there is a consistency between the analytic gradient information at selected iterations and the approximate Hessian, under the assumption that the potential is quadratic—which it is not in general but to some extent close to the equilibrium structure. The updated approximate Hessian—evaluated at iteration $i + 1$, is expressed as the sum of the previous approximate Hessian and a correction matrix.

$$\boldsymbol{H}_{i+1} = \boldsymbol{H}_i + \boldsymbol{H}_{\text{corr}} \tag{5}$$

The so-called quasi-Newton condition is based on the difference between two gradients—derived from the second-order Taylor expansion of the energy. That is,

$$\boldsymbol{g}(\boldsymbol{q}_i) = \boldsymbol{g}(\boldsymbol{q}_0) + \boldsymbol{H} \cdot (\boldsymbol{q}_i - \boldsymbol{q}_0) \tag{6}$$

which will for two subsequent iterations generate the condition

$$\boldsymbol{g}(\boldsymbol{q}_i) - \boldsymbol{g}(\boldsymbol{q}_{i+1}) = \boldsymbol{H} \cdot (\boldsymbol{q}_i - \boldsymbol{q}_{i+1}) \tag{7}$$

The update procedure proposes a correction matrix, $\boldsymbol{H}_{\text{corr}}$, such that the condition is fulfilled. The quasi-Newton condition effectively includes n independent equations (one for each internal coordinate), while the correction matrix—a symmetric matrix—has $n(n + 1)/2$ degrees of freedom. Hence, the quasi-Newton condition does not define a unique update. Rather, additional constraints, for example, that the norm of the correction should be as small as possible, or that the Hessian index (the number of negative eigenvalues) should or need not be preserved during the update procedure, will define unique update methods. Consistent with this, two classes of rank-2 update procedures have been developed, those to be used for optimizations of equilibrium structures (which ensure, or at least preserve positive definiteness in the approximate Hessian) and those for finding transition state geometries (which allow changes in the Hessian character). In the first class, the most popular update is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update [18–21], where the update is expressed as

$$\boldsymbol{H}_{i+1} = \boldsymbol{H}_i + \frac{\boldsymbol{y}_i \boldsymbol{y}_i^{\text{t}}}{\boldsymbol{y}_i^{\text{t}} \Delta \boldsymbol{q}_i} - \frac{\boldsymbol{H}_i \Delta \boldsymbol{q}_i (\boldsymbol{H}_i \Delta \boldsymbol{q}_i)^{\text{t}}}{\Delta \boldsymbol{q}_i^{\text{t}} \boldsymbol{H}_i \Delta \boldsymbol{q}_i} \tag{8}$$

where $\boldsymbol{y}_i = \boldsymbol{g}(\boldsymbol{q}_{i+1}) - \boldsymbol{g}(\boldsymbol{q}_i)$ and $\Delta \boldsymbol{q}_i = \boldsymbol{q}_{i+1} - \boldsymbol{q}_i$. A similarly devised update—the Murtagh-Sargent-Powell (MSP) update—has been constructed to be optimal for transition state optimizations [22, 23]. These update methods have been modified over the years as, for example, described by Bofill [24]. Finally, these update methods are normally applied to the approximate Hessian using the data from the last 5–20 iterations—TS optimizations tend to require more updates.

It is worth noting here that in terms of a quasi-Newton approach approximating the parent model, the quasi-Newton condition and the update methods at the best will converge to the analytic Hessian close to a stationary point on the PES. For any other geometry the update is

only effective and will contain higher order contributions—anharmonic contributions—whose effects have not been analyzed. Furthermore, the nature of the update procedure is such that it will only provide a surrogate model which is consistent with the most recent update. For any other structures the analytic gradient of the data set and the ones of the surrogate model will have a mismatch unless displacements and gradients are all orthogonal.

Choice of coordinates

This section deals with the coordinates used in connection with conventional molecular structure optimization. Coordinates typically used with ML approaches will not be discussed here but are rather discussed in the subsequent sections.

The selection of coordinates is instrumental in improving the convergence rate of geometry optimizations. First, one strives toward a coordinate representation in which the Hessian is diagonally dominant—in the right representation the harmonic model will correspond to independent harmonic oscillators. Second, coordinates which in some sense are natural—they try to follow the valleys of the PES as you move in between stationary points—would also have an advantage.

The initial use of Cartesian coordinates was pretty obvious and to some extent represents an unambiguous selection of coordinates. However, it became quickly clear that this was suboptimal for the various reasons discussed earlier—the Hessian is strongly coupled and the Cartesian coordinates describe rectilinear motions, while significant parts of molecular motion are curvilinear. Several improvements have been suggested over the years. The use of the normal modes expressed in terms of Cartesian coordinates will to some extent take care of the coupling (the normal modes are evaluated in most cases by diagonalization of the approximate Hessian). However, it should be noted that a strict quasi-Newton approach is invariant to a linear superposition of the coordinates. In this respect, the use of the normal modes in the basis of Cartesian coordinates will only allow for a trivial elimination of the translational and rotational degrees of freedom [25, 26]. Rather, the use of coordinate that are truly internal (void of translation and rotational components), and that can be of curvilinear nature (as in an angle bend or a torsion), would be preferred. There has been a number of attempts to implement this, such as the so-called Z-matrix method [27] or the approach with natural coordinates [28, 29]. Both of these implementations fulfill the previous criteria by using linear combinations of a set of nonredundant bond lengths, bond angles, and bond torsions, as the internal coordinates are explicitly free of translation or rotation. However, these approaches have a fundamental problem with ring-like systems. Here the selection of nonredundant coordinates could not treat these rings in an equivalent unbiased way—all bonds and the corresponding angles cannot be included, then which ones should be excluded? This problem was subsequently removed by the use of redundant internal coordinates and a general scheme for the elimination of the redundancy [30]. An extension of the use of redundant internal coordinate was later proposed in which the elimination of the redundancy is done so as to optimize the stiffness of the proposed internal coordinates [31].

We finally note that the forward transformation from coordinates and gradients expressed in terms of Cartesian to internal coordinates is trivial, while the back-transformation from internal coordinates to Cartesian coordinates requires an iterative procedure.

Constrained geometry optimization

Constrained optimizations can be used for several reasons, for example, for the purpose of identifying points of intersystem crossing (ISC, as in singlet-triplet transitions) or internal conversion (also called conical intersections) where the PES of states of the same spatial and spin symmetry cross. In these cases, both the energy and the molecular structure are of significance. For other reasons, one might be interested in energies subject to geometrical constraints. One might be interested in finding how the energy varies with respect to the change of a bond distance or the bond angle, or one could be interested in a zero-temperature reaction path of the PES (e.g., exploring the energy profile of a chemical reaction from a reactant to a product structure via a transition state structure).

The technology for constrained optimizations has developed to significant maturity and can be summarized into three levels of sophistication: (i) optimizations with penalty functions [10], (ii) optimizations using the technique of Lagrange multipliers [32–34], and a further development of the latter into (iii) the direct method [35], and (iv) the projected constrained optimization (PCO) [36, 37]. All of these are consistent with fact that some function $\zeta(\boldsymbol{q})$ should have some desired value ζ_0. This could apply to a single or a multitude of constraints, we will here limit the discussion to the case of a single constraint. In the first approach, a simple penalty function, f (a function of $\zeta(\boldsymbol{q}) - \zeta_0$, and with the properties $f(x) \geq 0$ and $f(0) = 0$), is simply added to the energy expression with a penalty factor γ, which determine to what extent we accept an error in relation to the total energy. This new energy expression, let us call it L,

$$L(\boldsymbol{q}) = E(\boldsymbol{q}) + \gamma f(\zeta(\boldsymbol{q}) - \zeta_0) \tag{9}$$

is then minimized with respect to the geometrical parameters, that is, the coordinates for which

$$\nabla_q L(\boldsymbol{q}) = \nabla_q E(\boldsymbol{q}) + \gamma \frac{\partial f}{\partial \zeta} \nabla_q \zeta(\boldsymbol{q}) = 0 \tag{10}$$

is obtained. This is a simple method to implement, however, at convergence the approach will not exactly fulfill the constraint but rather have an error whose magnitude is controlled by the penalty factor. This artifact is alleviated by the use of the technique of Lagrange multipliers,

$$L(\boldsymbol{q}, \lambda) = E(\boldsymbol{q}) + \lambda f(\zeta(\boldsymbol{q}) - \zeta_0) \tag{11}$$

The difference between Eqs. (9) and (11) is that in the latter, the new energy expression—the Lagrangian—is a function of both the molecular structure and the so-called Lagrange multiplier, λ, while in the former γ is a simple parameter with a fixed value. Also, in Eq. (11), the function $f(x)$ need not be positive. The stationary point is found at

$$\nabla_q L(\boldsymbol{q}, \lambda) = \nabla_q E(\boldsymbol{q}) + \lambda \frac{\partial f}{\partial \zeta} \nabla_q \zeta(\boldsymbol{q}) = 0 \tag{12}$$

$$\frac{\partial L(\boldsymbol{q}, \lambda)}{\partial \lambda} = f(\zeta(\boldsymbol{q}) - \zeta_0) = 0 \tag{13}$$

This implies a minimization in a space of $3N - 6(5) + M$ degrees, where M is the number of constraints, a bit of a contradiction since the subspace in which the minimization is to be conducted has a dimension of $3N - 6(5) - M$. Furthermore, the Lagrangian Hessian has a negative eigenvalue for each constraint resulting in that the optimization is a

minimization–maximization problem. Here the separation into the two different subspaces is based in the eigenvectors of the approximate Hessian—a procedure littered with possible problems. This dilemma was resolved with the direct approach in which the subspaces are separated to first order. Here the optimizations of the constraints are treated separately while the minimization is still done in $3N - 6(5)$ dimensions [35]. While this is a major leap forward there is still the issue that the Hessian can have negative eigenvalues, and that which Hessian-update method to use is not obvious. This was finally resolved by the PCO procedure of Anglada and Bofill [36], which introduces a linear variable transformation to the two different subspaces, $\boldsymbol{x}$ and $\boldsymbol{y}$, for the minimization and fulfillment of the constraints, respectively. First, a second-order equation in $3N - 6(5) - M$ dimensions and a Hessian that is positive definite is defined, and then a first-order equation in M dimensions is defined.

Geometry optimization in the direct inversion of the iterative subspace

An alternative to the quasi-Newton approach is the so-called GDIIS method [38], which deserves to be mentioned here. Although not strictly a quasi-Newton approach, it is based on a harmonic approximation of the PES. While the quasi-Newton approach is only indirectly based on the previous structures and gradients—through the iterative history and the Hessian-update method—the GDIIS approach does explicitly use this information at each iteration. We note that the method is not really a surrogate model, but has the assumption of a harmonic surface in common with the quasi-Newton methods. Hence, it shares most shortcomings with that family of optimization methods. In the GDIIS approach, the coordinates of each structure generated so far are expressed as

$$\boldsymbol{q}_i = \boldsymbol{q}^* + \boldsymbol{e}_i \tag{14}$$

where i is the iteration count, $\boldsymbol{q}^*$ is the unknown equilibrium structure, and $\boldsymbol{e}_i$ is the associated displacement/error vector for the ith iteration. A new coordinate is formed as

$$\boldsymbol{q}_{i+1} = \sum_i c_i \boldsymbol{q}_i = \boldsymbol{q}^* + \sum_i c_i \boldsymbol{e}_i \tag{15}$$

under the constraint that the coefficients add up to unity,

$$\sum_i c_i = 1 \tag{16}$$

A stationary point is found by the condition

$$\sum_i c_i \boldsymbol{e}_i = 0 \tag{17}$$

This would be nice if it were not for the fact that both $\boldsymbol{q}^*$ and the $\boldsymbol{e}_i$s are unknown. Here the approach makes an estimate, assuming that the energy functional is nearly quadratic and the approximate Hessian—improved by Hessian-update methods—is accurate enough, through

$$\boldsymbol{e}_i = -\boldsymbol{H}^{-1}\boldsymbol{g}_i \tag{18}$$

No exact solution can in general be obtained. Instead, optimal coefficients are found through the minimization of the norm of the error vector under the constraint. This optimization problem is solved using a standard Lagrange multiplier approach. This approach has been very

successful, in particular for optimizations starting close to the equilibrium structure—here the error vectors are accurate provided that a high-quality Hessian is used. However, the method has also demonstrated inferior behavior such as converging to nearby stationary points of higher orders or oscillating around inflection points. There has been a number of tricks proposed to minimize the impacts of these flaws [39], at which time it has been argued that the GDIIS approach is on par with quasi-Newton methods as the RFO method with respect to the convergence rate.

Machine learning methods for structure prediction

ML methods have been used for the prediction of structures of chemical systems. Significant contributions have been reported in fields like protein structure prediction [40–42], RNA secondary structure prediction [43], or crystal structure prediction [44, 45]. These are systems characterized by their large number of degrees of freedom, the importance of long-range interactions, and/or the existence of a myriad of local minima in their PES; the ML models typically use a combination of data mining, analogy modeling, heuristics, and refinement with physically based models. However, this kind of application is not the topic of this chapter, and we will therefore not discuss them in detail here. Let us briefly state the reasons for this.

1. We are interested in *general* structure optimization, not limited to a particular class of chemical systems or compounds. ML methods used for structure prediction are typically designed for specific types of systems (e.g., proteins or molecular crystals).
2. The optimization must be *flexible*, allowing for determining the location of different types of critical points, the incorporation of constraints, etc. The methods mentioned earlier only deal with the prediction of stable or the most likely structures.
3. The optimization should be *applied* to arbitrary quantum chemical methods. The problem we address in this chapter is not which electronic structure method can provide more realistic molecular structures, but how to efficiently find the molecular structure that a given electronic structure method predicts. The goal of structure prediction with ML methods is most of the time predicting experimental data.
4. The result must be *accurate*. When minimizing the energy, for instance, it is important that the resulting molecular structure represents a true minimum of the PES (within some operational accuracy). ML methods tend to provide approximations or reasonable guesses for what the minimum structure would be.

The present chapter is concerned with finding minima or other notable points of arbitrary multidimensional functions, with particular attention to the specificities of PESs, and how ML techniques can be of assistance. In this sense, structure prediction methods are only useful—and this is often of no minor importance—as a means to generate an initial or a set of initial structures that can be subsequently optimized to locate the actual structures of interest. For example, these methods can provide a number of likely structures—conformers or isomers—for a molecular system, each of them to be further refined and have its energy (or other properties) evaluated.

In this respect, we can nevertheless highlight some recent works aimed at medium-sized molecular structures, which are the most common target of quantum chemical studies.

Mansinov et al. [46] proposed a deep neural network that, after being trained on observed (experimental or computed) structures of a class of molecules, could generate a sample of conformations for new molecules. The quality of the resulting structures was competitive, better by some measures, with standard force field optimizations, meaning that they could as confidently be used for further optimization with quantum chemical methods; and the computational cost for generating each structure was significantly lower. The input features for the neural network were both atomic (e.g., element, hybridization, formal charge) and bond (e.g., connectivity, type of bond). Lemm et al. [47] used a similar approach, but based on kernel ridge regression instead of neural networks. The input features were restricted to elements and connectivity, and the structure prediction was extended to TS and crystalline solids. The authors recognized that this kind of approach can be useful for generating initial structures not only for optimization, but also to serve as training sets for other ML models for the prediction of properties from structures. With the specific goal of generating initial structures for TS optimization, Makoś et al. [48] used neural networks to predict TS structures from the Cartesian coordinates of reactants and products. The generated structures, in general, were observed to lead to a successful TS optimization more consistently than the popular quadratic synchronous transit (QST2) [49].

Another way in which ML methods can assist in molecular geometry optimization is by replacing the target PES with some other model that is much cheaper to compute and on which regular optimizations can be performed without particular concerns about the efficiency. This is the strategy commonly used when one employs, for example, molecular mechanics force fields to preoptimize a given molecular structure before proceeding to a more costly optimization on quantum chemical PES. A number of ML models have been proposed to provide a more accurate alternative to force fields, although they must often be trained for a particular system or type of systems [50, 51]. This is, however, again a different problem, and it has more to do with generating a surrogate model for the optimization than with the optimization itself, something that is discussed in more detail below.

Machine learning-based surrogate PES

In this section, we will initially discuss why ML techniques might be of interest to procedures dealing with molecular structure optimizations. This is followed by a brief motivation on which ML method might be the most suitable for this purpose. Finally, we will walk you through the early development of ML methods to generate global surrogate PESs as used in, for example, molecular dynamics—using both neural networks (NN) and kernel methods—to the most recent developments on the use of GPR for local surrogate models used in equilibrium, transition state, and constrained optimizations, and the computation of minimum energy paths. In the subsequent section, we will in some details describe a gradient-enhanced Kriging (GEK) implementation in association with the use of internal coordinates, an HMF, an alternative rationale for selecting the individual characteristic length-scales, and an RVO procedure.

While ML techniques are usually associated with large data sets—as large as several orders of magnitude times the number of the molecular degrees of freedom—and an attempt to find a general solution to a problem, for example, to replace a parent method with a global fitted

surrogate model, it has been demonstrated that for locally accurate surrogate models in association with some ML techniques only a rather limited set of data points—far fewer than the number of degrees of freedom—is required. It is fair to say that while neural networks normally fall in the category of the former, GPR is more attuned to the requirement of the latter [52, 53]. For this particular reason, if an ML method is to be used in association with molecular structure optimization it is natural to examine the development potential of the GPR approach. Furthermore, the nonparametric nature of the GPR has several advantages over RS-QNR and RS-RFO procedures. In comparison, the GPR methods will facilitate a surrogate model that (i) can support the description of more than just one stationary point, (ii) can reproduce the PES at the data points, (iii) will converge to the parent model as the number of data points is increased, (iv) can mimic anharmonic characteristics, and (v) will produce an analytic estimation of the expected dispersion at any point. The second-order quasi-Newton methods cannot meet any of these criteria.

In the search of an ML method that will be sufficiently accurate locally, it is for efficiency considerations instrumental that such a method employs all the information from the parent model as the optimization procedure proceeds. That is, all analytic information has to be employed—in standard ab initio calculations this would translate to use not only the energies but also the gradients. Furthermore, the ML method should to a large extent efficiently use the information it has been provided and not waste it as the iterative procedure proceeds. The GEK [54–56] is such an ML procedure. The method is based on GPR, an extension of the Kriging method [57, 58] in which the fitting is applied to both the function value and the gradient of the parent PES.

What follows is a short review of the scientific studies that have been performed on the subject of the use of ML methods to generate surrogate models, for both a global and a local fit. ML algorithms have been used in computational chemistry, for example, as a mean to speed up molecular dynamics calculations. Here ML methods, especially techniques based on neural networks, were used to generate global surrogate models [52, 59–63]. The use of other ML methods has also been explored, for example, the use of support vector machines to create the transition state surface that separates the reactants from the products [64], and the use of reinforcement learning to eliminate the line-search step in quasi-Newton optimization without any loss of efficiency [65]. Another ML-like approach is the Gaussian approximation potential (GAP) approach [66–68]. This approach, however, depends on the selection of descriptors which can be difficult to generalize for it to be applicable in general [69]. A new direction was presented in a series of papers in 2015–16 [70–73] by Cui and Krems, who were the first to report on the use of GPR as a mean to accurately represent PES for collision dynamics and scattering processes studies. They demonstrated that accurate PESs for the Ar–benzene complex could be obtained by using data from just some 200 trajectories of classical dynamics. Further, they found that the Matérn correlation function [74] gives improved accuracy as compared to standard Gaussian correlation functions. This approach has also been used by others to represent multidimensional reactive PESs [75]. Inspired by the comment of Peterson [52]—"… the training sets are small, so an SVR [(support vector regression)] model may offer superior characteristics to an NN model"—Koistinen et al. were the first to report on the use of GPR for a local fit in association with the computation of minimum energy paths between different arrangement of a heptamer island on a crystal surface [76]. In their study, they report a reduction of actual energy and gradient evaluations by as much as 80%. The

implementation was based on the use of Cartesian coordinates and the GPR hyperparameters were reoptimized at every instance a new energy and gradient were added to the data set. In a recent study by the same group, again on the calculations of minimum energy paths, the use of inverse interatomic distances in the covariance function was explored on two previously problematic cases [77]. The use of the inverse interatomic distances resolved the problems and proved to also improve performance on the originally reported benchmark. Another example of a local surrogate model to accelerate optimizations was published by Schmitz and Christiansen, in a work where they used GPR and adaptive delta learning to approximate the gradient of an expensive electronic structure method [78]. In 2018, Denzel and Kästner published two papers in which GPR was used for molecular equilibrium and transition state structure optimizations, respectively [79, 80]. An iterative strategy is applied in which the desired structure is found on the surrogate PES, using some standard second-order optimizer—microiterations—the energy and gradients at this structure is then evaluated on the parent PES—macroiterations—and checked for convergence conditions. If not converged the data from the parent model is used to enhance the surrogate model and the process is repeated. This implementation used Cartesian coordinates, a Matérn covariance function ($\nu = 5/2$), an empirically set universal characteristic length scale of 20 a_0, a prior mean to be 10 E_h higher than any energy in the data set (to guarantee that the energy of the surrogate model is bound), an overshooting approach, and a multilevel GPR approach to reduce the scaling as the iteration count exceeds a threshold value. All this in combination with an L-BFGS optimizer [81] at the level of the microiterations. For the TS optimization case, a P-RFO [7] approach was used. The authors reported substantial reductions in iterations counts. Subsequently, a number of other groups have more or less applied the same recipe with small variations to optimizations of equilibrium structures, TS structures, and minimum energy paths [69, 77, 82–88]. Here, the reports by Koistinen et al. [77, 86] stand out because of the use of inverse interatomic distances in the covariance function, rendering the surrogate model invariant to rotations and translations. In 2020, Meyer and Hauser [89] published an extensive benchmark on the performance of GPR-assisted molecular structure optimization based on different selections of coordinates and two different covariance functions. In their study, convergence rates are reported for Cartesian coordinates, inverse distances, and Z-matrix internal coordinates. The latter two coordinates are explored in four different flavors: fully redundant, delocalized, localized, and reduced redundancy. In terms of the Cartesian coordinates both a squared exponential (Gaussian) and a Matérn covariance function ($\nu = 5/2$) were explored. They concluded that internal coordinates in any form were to be preferred over Cartesian coordinates. Furthermore, they stated that "[t]he lack of heuristics on actual force constants and couplings can only be partially compensated by the choice of a suitable set of internal coordinates, and future undertakings will have to encode this knowledge, e.g., via a pre-informed choice of hyperparameters in their machine learning models." This call is answered by an approach which suggests that the individual characteristic length scales of the internal coordinates be based on the eigenvalues of the initial estimate of the Hessian, in combination with an RVO procedure [2, 3]. Explicit details of this implementation, as used for optimization of equilibrium and transition state structures, and constrained optimizations, as in minimum energy path calculations, are the subject of the next section.

The restricted variance optimization method

This section is devoted to the description of the GEK in association with RVO [2, 3]. We will not at this place reiterate the full set of equations and properties of GPR, in general and GEK, in particular. We assume that the reader is familiar with the subject from the previous Chapters 9 and 10 of this book. However, when significant for the presentation some details of the procedure will be repeated here.

The RVO method, based on GEK, was developed in light of that the standard second-order optimization schemes in general work very well—some aspect of this approach is optimal—and the growing experience on the successful components of an optimal GPR-based procedure. In this respect, the facts that the choice of coordinates makes a significant difference, that the benefit of a good estimate of the Hessian is instrumental, and that a step restriction is an integral part of an efficient implementation were natural constraints to the development. We now start with some of the details of the GEK that we will need in our discussions below. For more details on GPR and how the derivative (gradient) information is included, see the Chapter 9 on kernel methods.

In the Kriging and the GEK the surrogate model—$E^*(\boldsymbol{q})$—will predict the energy as a function of the coordinates (arbitrary coordinates) of the molecular system, $\boldsymbol{q}$.

$$E^*(\boldsymbol{q}) = \mu + \boldsymbol{v}(\boldsymbol{q})^{\mathrm{t}}\boldsymbol{M}^{-1}(\boldsymbol{y} - \boldsymbol{1}\mu) \tag{19}$$

where the first term, μ, is the baseline or trend function, while the second term is the local deviation of the energy around μ [56]. Here, $\boldsymbol{y}$ is the column vector of generalized function values from the source data, and $\boldsymbol{1}$ is a vector with the value of one and zero for elements corresponding to energies and gradients, respectively. The generalized covariance matrix, $\boldsymbol{M}$, is a function of the covariance function, $f(d_{ij})$, where d_{ij} is a scalar generalized distance between the coordinates at sample points i and j, in our case expressed as

$$d_{ij} = d(\boldsymbol{q}_i, \boldsymbol{q}_j) = \sqrt{\sum_{k=1}^{K}\left(\frac{q_{i,k} - q_{j,k}}{l_k}\right)^2} \tag{20}$$

where K is the number of degrees of freedom of the molecular system ($K = 3N - 6$ for a nonlinear system with N nuclei and no external fields), l_k is a scale parameter that influences the width of the covariance function—the characteristic length scale—in the kth dimension. When all l_k are equal, this is simply the (scaled) Euclidean distance. The generalized covariance vector $\boldsymbol{v}(\boldsymbol{q})$ is defined analogously. Finally, the RVO implementation followed the literature and used a Matérn covariance function of order $p = 2$ (i.e., $\nu = (2p + 1)/2 = 5/2$),

$$f_2(d_{ij}) = \left(\frac{5d_{ij}^2}{3} + \sqrt{5}d_{ij} + 1\right)\mathrm{e}^{-\sqrt{5}d_{ij}} \tag{21}$$

For a positive definite $\boldsymbol{M}$ (which is guaranteed by a Matérn covariance function), the expected variance for the prediction is given by [90]

$$s^2(\boldsymbol{q}) = \frac{(\boldsymbol{y} - \boldsymbol{1}\mu)\boldsymbol{M}^{-1}(\boldsymbol{y} - \boldsymbol{1}\mu)}{n}[1 - \boldsymbol{v}(\boldsymbol{q})^{\mathrm{t}}\boldsymbol{M}^{-1}\boldsymbol{v}(\boldsymbol{q})] \tag{22}$$

where the first factor accounts for the variance of the sample points, while the second measures the distance of $\boldsymbol{q}$ to the sample points, and will give zero whenever $\boldsymbol{q} = \boldsymbol{q}_i$. Assuming a Gaussian variance, the actual energy can thus be estimated, with a 95% confidence, to lie in the interval $E^*(\boldsymbol{q}) \pm 1.96\sqrt{s^2(\boldsymbol{q})}$.

The most demanding component in this surrogate model is $\boldsymbol{M}^{-1}$. In practice, instead of computing the inverse of the matrix, it is more efficient to directly compute the product of the inverse and a vector—either $\boldsymbol{\omega}(\boldsymbol{q})^{\mathrm{T}} = \boldsymbol{v}(\boldsymbol{q})^{\mathrm{T}}\boldsymbol{M}^{-1}$ or $\boldsymbol{w} = \boldsymbol{M}^{-1}(\boldsymbol{y} - \boldsymbol{1}\mu)$—and Eq. (19) then becomes

$$E^*(\boldsymbol{q}) = \mu + \sum_i \omega_i(\boldsymbol{q})(y_i - \delta_i\mu) \tag{23}$$

or

$$E^*(\boldsymbol{q}) = \mu + \sum_i w_i v_i(\boldsymbol{q}) \tag{24}$$

where δ_i is an element of $\boldsymbol{1}$, that is, its 1 for elements corresponding to energies and 0 for elements corresponding to gradients. Eq. (24) is particularly attractive, because it is simply a linear combination of "basis functions" $v_i(\boldsymbol{q})$ that depend only on the chosen covariance function and the coordinates of the sample points, $\boldsymbol{q}_i$:

$$v_i(\boldsymbol{q}) = f(d(\boldsymbol{q}_i, \boldsymbol{q})) \tag{25}$$

In the case of GEK, $\boldsymbol{v}(\boldsymbol{q})$ contains not only the covariance function, but also its derivatives with respect to each degree of freedom, and so for n sample points and m dimensions

$$E^*(\boldsymbol{q}) = \mu + \sum_i^n w_i f(d(\boldsymbol{q}_i, \boldsymbol{q})) + \sum_i^n \sum_k^m w_j \frac{\partial f(d(\boldsymbol{q}_i, \boldsymbol{q}))}{\partial q_k} \tag{26}$$

where j simply indexes all the m derivatives for each of the n sample point in some convenient consecutive order. Since $\boldsymbol{w}$ does not depend on the coordinate where the energy is evaluated, the derivatives of $E^*(\boldsymbol{q})$—the gradient and Hessian of the surrogate model—are easily evaluated from the first, second, and third derivatives of the covariance function f:

$$\frac{\partial E^*(\boldsymbol{q})}{\partial q_k} = \sum_i w_i \frac{\partial v_i(\boldsymbol{q})}{\partial q_k} \tag{27}$$

$$\frac{\partial^2 E^*(\boldsymbol{q})}{\partial q_k \partial q_l} = \sum_i w_i \frac{\partial^2 v_i(\boldsymbol{q})}{\partial q_k \partial q_l} \tag{28}$$

Let us add some words on the scaling aspects of GEK. The covariance matrix, $\boldsymbol{M}$, has a dimension of $d_M = n(1 + m)$—for each point in the data set, n of them, there are $(1 + m)$ data items (one energy and the m components of the gradient), where m is the dimensionality of the surrogate model. Since the GEK will involve solving a system of equations, effectively inverting $\boldsymbol{M}$, the effective scaling of the procedure of the approach will scale to the order of d_M^3. With this in mind one has to be very mindful on when to use this approach. Normally the GEK is appropriate in cases with ab initio studies of small to modestly sized systems, however, for extremely large molecular systems in association with molecular mechanics the tradeoff is questionable.

We will now in sequence address and explain some details of the RVO implementation, which is a multilayer algorithm—it microiterates on the surrogate model and macroiterates on the parent model—mimicking an RS-RFO procedure (see Fig. 1). This will be followed by an assessment of the method in comparison with a state-of-the-art implementation of the RS-RFO method.

Hessian approximation

An integral part of the RVO supported by GEK is the use of an estimated Hessian. Much of what follows has its origin in the quality of such a guess. To address this need the HMF of Lindh et al. [17] was selected. This procedure will be used to produce a new estimate each time a microiteration is done.

Coordinates

The selection of coordinates is very important to the potential success of the surrogate model. One important aspect is that the surrogate model should be invariant to the rotations and translations. Hence, internal coordinates are mandatory. Here the space of coordinates is selected to be spanned by the nonredundant force-constant-weighted (FCW) internal coordinates [31] of Lindh et al. The final selection of redundant coordinates, however, are the eigenvectors of the HMF Hessian expressed in terms of FCW internal coordinates. The eigenvalues of the Hessian will be a crucial part of setting reasonable values of the characteristic length scales and the value of the trend functions—the hyperparameters.

The trend function

The selection of the trend function, μ, as a constant allows us to assist the surrogate model with an important property—the surrogate model should be bound. So possibly any value larger than any of the energies in the data set would do. However, we want a soft transition to regions of coordinate space that have not been explored. Hence, a too large value would hamper convergence and make the procedure too slow. Actually, we would like to explore the unknown in a more controlled fashion and not have a too hard boundness interfere with such a procedure. After some experimental calculations, an empirical value of μ was set to be 10.0 E_h above the highest energy in the data set.

The characteristic length scales

The success of standard second-order optimization schemes is that the force constants are very reasonable. This quickly guides the optimization toward convergence. This could possibly be faked by extending the GEK to also include Hessian information, but would probably be of no help since the Hessian is just an approximation. Is there any other way we could get the GEK to mimic that it has a Hessian that is similar to the one used in the quasi-Newton optimization? Yes, we can play a trick with the hyperparameters. In general, this is not analytically possible. However, for a single data point (in our case for the last computed structure) the surrogate model Hessian is diagonal, with elements given by

$$H(\boldsymbol{q}_i)_{kk} = (\mu - E(\boldsymbol{q}_i))\frac{\partial^2 f}{\partial(q_k)^2} \tag{29}$$

It follows that the l values can be set, for a Matérn-5/2 kernel, Eq. (21), from the following expression

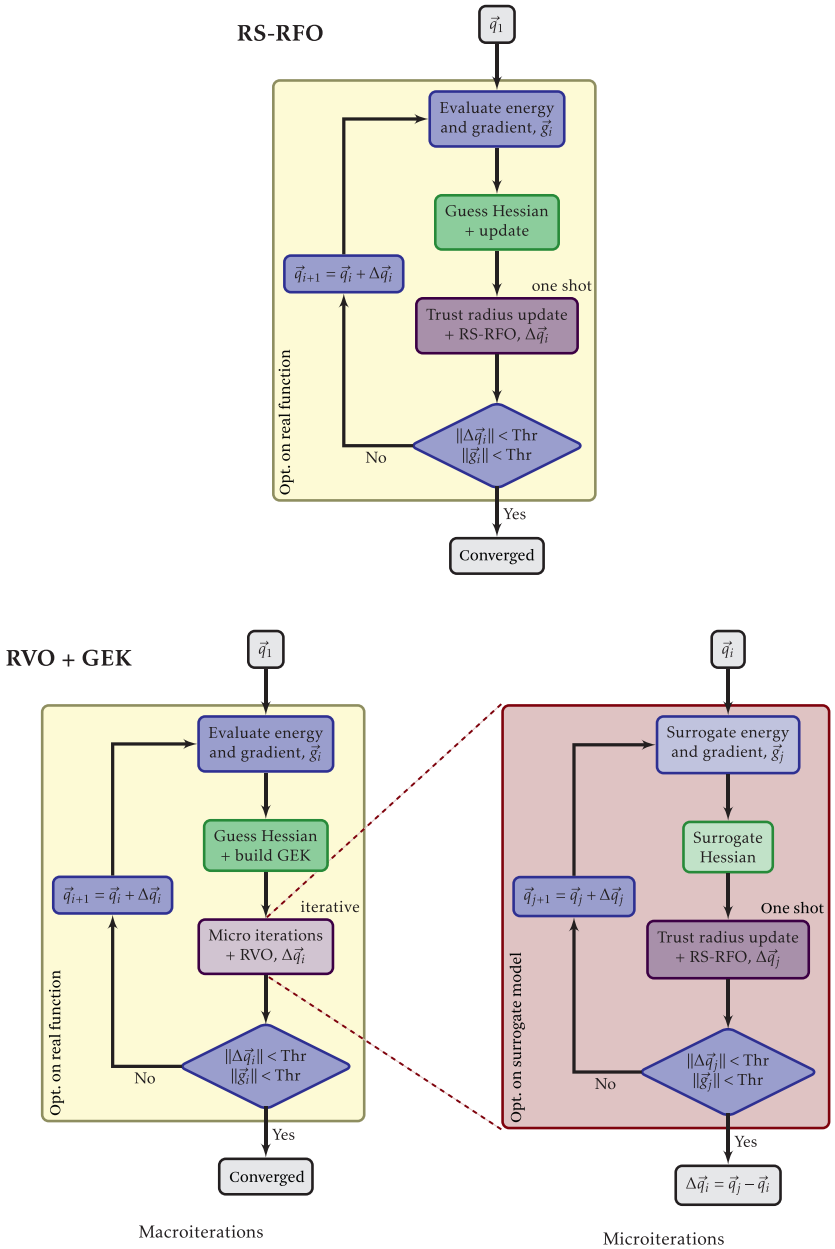


FIG. 1 Schematic comparison of the conventional RS-RFO optimization method (*top*) and the proposed RVO algorithm, based on a GEK surrogate model, and using RS-RFO in the microiterations (*bottom*).

$$l_k = \sqrt{\frac{5(\mu - E_{\max})}{3H_{\mathrm{HMF}}(\boldsymbol{q}_i)_{kk}}} \qquad E_{\max} = \max_i \{E(\boldsymbol{q}_i)\} \tag{30}$$

where l_k is the characteristic length of coordinate k and H_{kk} is the corresponding eigenvalue of the approximate Hessian (which is always positive definite [17]). Thus, in this GEK implementation the individual characteristic length scales are set to reproduce the HMF Hessian PES curvature at the most recent point. The GEK with the full set of data points will effectively serve the purpose of a Hessian-update procedure. We note that for too small values of the eigenvalues of the Hessian large characteristic length scales will result. For that purpose $H_{\mathrm{HMF}}(\boldsymbol{q}_i)_{kk}$ is set to be not smaller than 0.025 $E_\mathrm{h} a_0^{-2}$.

Restricted-variance optimization

As discussed earlier, the RS-RFO method has been shown to be an extremely efficient and robust optimization procedure. The step restriction has been demonstrated to work best in a setting in which its threshold is dynamically updated as the optimization proceeds. This has to be based on some ad hoc definition on how trustworthy the rational function is at the current point and how large should the trust radius be. The GEK surrogate model has here a significant advantage, there is an analytic estimation of the variance $s^2(\boldsymbol{q})$ at any point in space. In this respect, we have developed a type of restricted-step RFO procedure where the step restriction is not based on the associated length of the step but rather on the value of the variance of the surrogate model at the suggested new molecular structure—the restricted-variance optimization procedure. The variance restriction is enforced by making sure that every microiteration (see Fig. 1, bottom right) produces a 95% confidence interval within the specified threshold, σ_{RVO}, that is,

$$1.96\sqrt{s^2(\boldsymbol{q}_j)} \leq \sigma_{\mathrm{RVO}} \tag{31}$$

As the optimization is converging, the threshold is reduced. In the case of constrained optimization, the variance threshold for the individual subspaces is roughly speaking half of the total threshold (for details consult Ref. [3]).

This RVO implementation has been benchmarked against a state-of-the-art second-order method, RS-RFO, for equilibrium, transition state, and constrained structure optimizations, and for the computation of reaction paths. Both methods have been implemented in the open-source quantum chemistry program package OpenMolcas [91]. We will briefly reiterate the results here but recommend the reader to consult the original papers for a more verbose representation of the benchmark test suites and the results. The performance for the optimization of equilibrium structures was benchmarked on the extended Baker [15] (e-Baker), the Baker-TS [92], and the S22 [93] suites of molecules. The e-Baker test suite is a collection of 33 molecular structures preoptimized at a lower level of theory. For these benchmarks both the HF and the DFT level of theory was employed. The calculations at the HF level were primarily used to train the approach on the values of the trend function and some of the threshold values used therein. The Baker-TS test suite is a set of 25 molecules close to a transition state structure, while the S22 test suite is a set of 22 molecular complexes stabilized by hydrogen bonding and/or van der Waal forces. The results from these benchmarks are presented in Fig. 2. In the case of the e-Baker/DFT benchmarks not much improvement was expected, since the starting structures are close to the equilibrium structures. However, the RVO shows

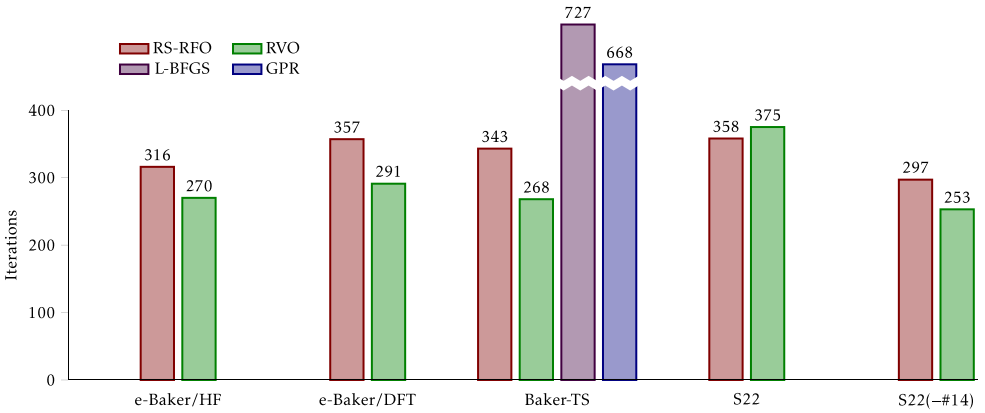


FIG. 2 Total iteration counts in the different benchmark sets, obtained with the RS-RFO and RVO methods. For comparison, the results from Ref. [79] for the Baker-TS set are also provided (L-BFGS and GPR). In the Baker-TS results, Cases 3, 5, 9, and 25 have been excluded for all methods.

a reduced total iteration count of some 22%. This is not bad considering that RS-RFO methods have been developed and refined for a very long time while the use of ML in the RVO approach is very novel. There must be room for some more improvement. For the Baker-TS benchmark we also have access to the results of Denzel and Kästner [79] where they report iteration statistics for a conventional implementation—L-BFGS [81]—and the GPR implementation of their own making. Both implementations are available in the open-source optimization library DL-FIND [94]. While both ML-based approaches demonstrate a considerable speed with respect to their conventional counterparts—the RVO with a reduction in the number of iterations by some 28%—it is noteworthy that the conventional implementation of RS-RFO in OpenMolcas is twice as efficient as L-BFGS and the RVO implementation in the same package is 2.49 times faster than the GPR implementation in DL-FIND. This is possibly partially associated by the use of Cartesian coordinates—the surrogate model is not translation and rotation invariant—and a single characteristic length scale in the latter. For the benchmark optimization of molecular complexes stabilized by weak forces and with an expectation of a PES with significant anharmonic characteristics—the S22 test suite—the RVO procedure demonstrated superior performance.

The Baker-TS benchmarking gave the possibility to benchmark both constrained and TS optimizations. In the constrained calculations, the Baker-TS structures were optimized with conditions such that the final structure should have the structural characteristics of the TS geometry. In the case of the TS optimization, this initial constrained optimization was relaxed as soon as the surrogate model provided a Hessian with one negative eigenvalue. At this point, the optimization adapted the RS-I-RFO approach under the umbrella of RVO. The results of the constrained optimization are presented in Fig. 3. With no exceptions, the RVO procedure is superior to state-of-the-art second-order restricted-step optimization methods. Moreover, it displays an impressive robustness and quickly converges four cases in which conventional methods fail. For the transition state structure optimizations (see Fig. 4) we again observe a familiar pattern, RVO represents an iteration reduction of some 26.5%. This efficiency is attributed to a significant improvement during the latter part of the optimization, probably due to the improved quantitative accuracy of the surrogate model's Hessian.

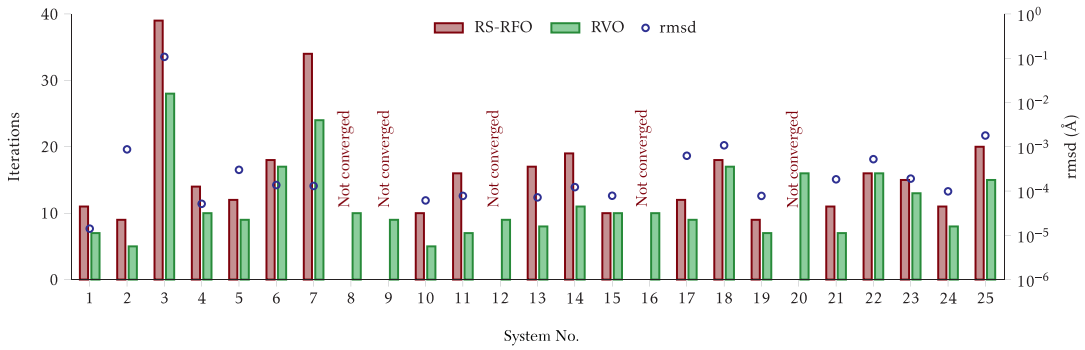


FIG. 3 Number of iterations to converge the Baker-TS structures to a constrained minimum. The *circles* represent root mean square displacement between the converged structures.

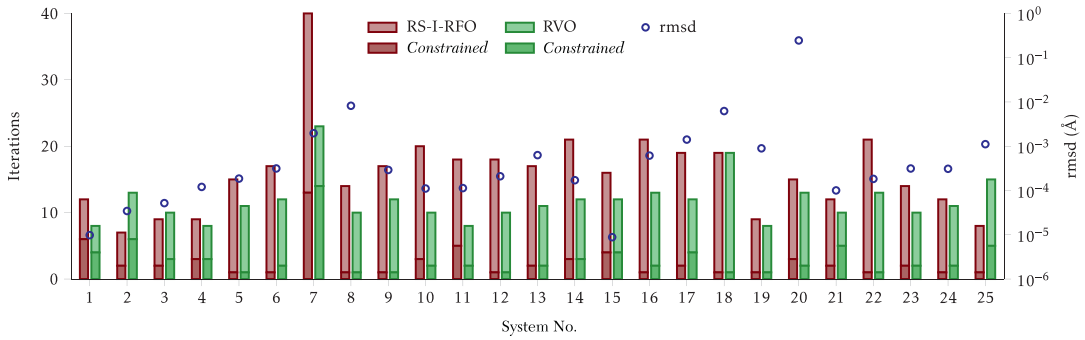


FIG. 4 Number of iterations to converge the Baker-TS structures to a saddle point. The *darker color* in the bars indicates the iterations with active constraints. The *circles* represent the root mean square displacement between the converged structures.

Finally, we take a look at benchmark calculations in the computation of reactions paths. A collective analysis of all 25 reactions paths is presented in Fig. 5. Each reaction path calculation is done as a series of constrained optimizations, one for each point along the path, going downhill from the TS, and each of these optimizations takes a number of (macro)iterations to converge. The graph represents the number of optimizations that converged on a given number of iterations. Thus, the tallest bar means that 250 points converged on three iterations with RVO. This case demonstrated an almost 50% reduction in the number of iterations to establish the reaction path. A possible source of this improvement is that the RVO procedure brings along information from each point on the path, while the conventional approach starts losing information already at the third iteration.

Before we rest our case on the benefits of the RVO approach some words on the CPU timings. As addressed earlier, the RVO should have a scaling of the order of d_M^3. Here we report that, for example, for the histamine–H^+ molecule ($3N - 6 = 48$ degrees of freedom) of the e-Baker test suite, while the DFT time accounts for little more than 360 s per iteration both the RS-RFO and the RVO procedures account for less than 1 s per iteration; or for a manganese cluster of 61 atoms (177 degrees of freedom) in which the ab initio energy and gradient calculations took 3480 s, the RS-RFO and the RVO algorithms took, respectively, 2 and 7 s per (macro)iteration.

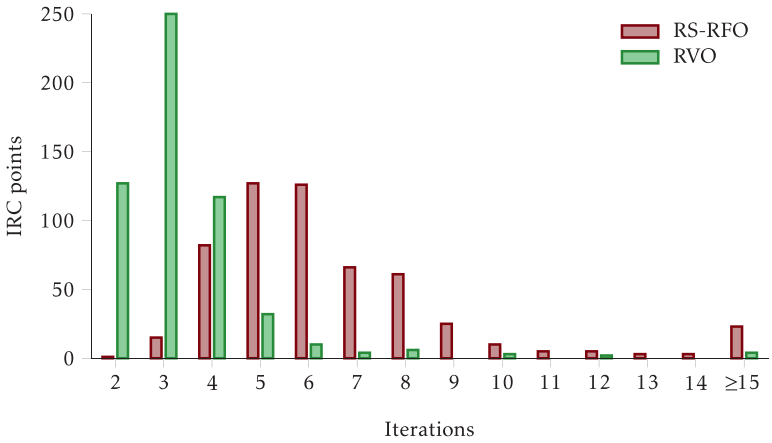


FIG. 5 Histogram of the number of iterations needed for optimizing each reaction path point.

Case studies

In this section, we discuss some concrete practical examples of geometry optimization, comparing conventional second-order optimization methods with the RVO method, which makes use of ML techniques. The examples are very simple, although realistic. The first one is a one-dimensional system, and we will use an ad hoc implementation, which will allow us to better understand the innards of the optimization, as well as play with the different parameters. The latter examples will make use of the existing implementation in the open-source electronic structure software OpenMolcas [91]. Complementary materials for these case studies can be found as described on the book's companion website.

One-dimensional system (H_2)

Our first example is an optimization of the H_2 molecule. In this case, the selection of internal coordinates is trivial, as there is only one sensible choice: the interatomic distance. It is therefore a one-dimensional system which allows easy visualizations.

We provide a full Python implementation (although it depends on the `numpy` and `scipy` modules), with comments in the code. Here we will discuss the main features.

Instead of computing the energy of H_2 at different interatomic distances with some approximate method, we use an analytical expression, which was obtained as a fit to high-quality computational results [95]. In this way, we can easily compare with the "true" function (i.e., the analytical fit) and quickly get results. But keep in mind that our goal is to apply these methods to other functions that are unknown and/or much more expensive to compute. The functional form of the fit is given by

$$E(r) = 2E(\mathrm{H}) - D_\mathrm{e}\left(1 + \sum_{i=1}^{4} c_i (r - R_\mathrm{e})^i\right) \exp\left[-c_1 (r - R_\mathrm{e})\right] \tag{32}$$

where $E(\mathrm{H})$ is the energy of an isolated hydrogen atom (which is 0.5 E_h), and the values of the parameters are:

$$\begin{aligned} D_\mathrm{e} &= 0.173108\ E_\mathrm{h} & c_1 &= 2.208257\ {a_0}^{-1} \\ R_\mathrm{e} &= 1.401315\ a_0 & c_2 &= 1.468554\ {a_0}^{-2} \\ & & c_3 &= 0.759565\ {a_0}^{-3} \\ & & c_4 &= -0.033099\ {a_0}^{-4} \end{aligned}$$

With this analytical form it is also straightforward to compute the gradient (i.e., $\mathrm{d}E(r)/\mathrm{d}r$), which is left as an exercise to the reader.

Another piece we will need for our code is the HMF [17] that, for the specific case of H_2, reduces to

$$H_\mathrm{HMF}(r) = k \exp\left(r_\mathrm{ref}^2 - r^2\right) \tag{33}$$

with $k = 0.45 E_\mathrm{h} a_0^{-2}$ and $r_\mathrm{ref} = 1.35 a_0$. You will notice that this is *not* the true Hessian for the analytical function at hand. In fact, it does not even have the correct behavior, since it will only yield positive values, and it is very clear (once you plot it) that there are some places where the curvature of our analytical function is negative. The purpose of the HMF is not to approximate the true Hessian at all points, but only at a minimum, or to guess what the Hessian would be *if* there were a minimum at r.

At this point we follow two different paths. First, we will create a program to find the minimum of $E(r)$ with RS-RFO, the conventional method. Then we will do it with RVO, using GEK as a surrogate model.

For the RS-RFO implementation, we start with Eq. (4) and simplify it for the case where $\boldsymbol{q}$ (and $\boldsymbol{g}$, and $\boldsymbol{H}$) is one-dimensional, and replace $\boldsymbol{q}$ with r:

$$E(r) = E(r_0) + \frac{g\Delta r + \frac{1}{2} H (\Delta r)^2}{1 + \alpha (\Delta r)^2} \tag{34}$$

and (another exercise) we find that this expression has a minimum where

$$\alpha g (\Delta r)^2 - H \Delta r = g \tag{35}$$

which we can solve for Δr, once we know g, H, and α.

In general, we would like to use a Hessian update method to improve the approximate Hessian as the optimization proceeds. This means taking the above H_HMF result and applying some formula dependent on the previous coordinates and gradients. The most common formula is the BFGS [18–21] one (Eq. 8). However, for the specific case of a one-dimensional system, the update procedure removes all previous information and simply assigns to the Hessian a value that depends on the last two points:

$$H_k = \frac{g_k - g_{k-1}}{r_k - r_{k-1}} \tag{36}$$

that is, as soon as we have computed the second point, we will not need the HMF any more, since the approximate Hessian will be fully determined by the previous coordinates and gradients.

The final piece we need is a dynamic step restriction, that is, a way to modify α in Eqs. (34), (35) to ensure that the resulting Δr does not exceed a specified limit. For this we simply use the `fsolve` function of `scipy`, to give us a proper value of α if the default $\alpha = 1$ yields too long a step.

The core of this optimization can then be expressed as the pseudocode in Algorithm 1.

When we run this example (`optim_RFO.py`), we obtain a series of graphs (press a key or click the mouse to see the next one) showing our "true" function of Eq. (32) as a dashed line and initially a single red point. This first point represents the initial structure, H–H with a bond length of 5.5 a_0. As we iterate, we also plot the RFO surrogate model, Eq. (34) (blue line), and the minimum on this line, for which we compute the "true" energy (arrow to blue point); this is the next guess for the minimum of the true function. Fig. 6 shows two graphs, after a few iterations and at convergence. We can note several things about the process. The RFO surrogate curve only fits the last computed point and is in general not a very good approximation to the target function. At convergence, once a minimum is located, the RFO curve is a much better approximation, but still only around the minimum. The step size restriction is a severe limitation for a speedy progress; we could increase it with no harm in this case (`max_step`), but the default value of `0.3` is safer on most occasions.

Let us try this with the RVO method. The main differences are that the surrogate curve is given by a GEK model instead of RFO, and that we will use a variance restriction instead of a direct step length restriction. The code for building the GEK model is provided in the file `kriging.py`, and it looks more complicated than it needs to, because it allows any number of dimensions, not just 1. Two different kernels are included, Matérn-5/2 and Gaussian. The characteristic length, or l value, is set according to the H_{HMF} value as in Eq. (30) (for a Matérn-5/2 kernel), and the number of data points is limited to 5 in this example.

For the variance restriction, the strategy is as follows: We first find a minimum on the surrogate curve with the `optimize` function from `scipy`. If the variance computed at that point

ALGORITHM 1

RS-RFO algorithm.

```
X ← initial r
Y ← E(X)
dY ← E'(X)
ddY ← H_HMF(X)                          ▷ Initial guessed Hessian
repeat
    alpha ← step_restriction(X,dY,ddY)   ▷ Find value of α
    new ← RFO_min(X,dY,ddY,alpha)        ▷ Find value of Δr (new = X + Δr)

    X_old ←X                             ▷ Save previous values
    Y_old ←Y
    dY_old ←dY
    ddY_old ←ddY

    X ←new
    Y ← E(X)
    dY ← E'(X)
    ddY ← (dY−dY_old)/(X−X_old)          ▷ Approximate Hessian
until dY < threshold
```

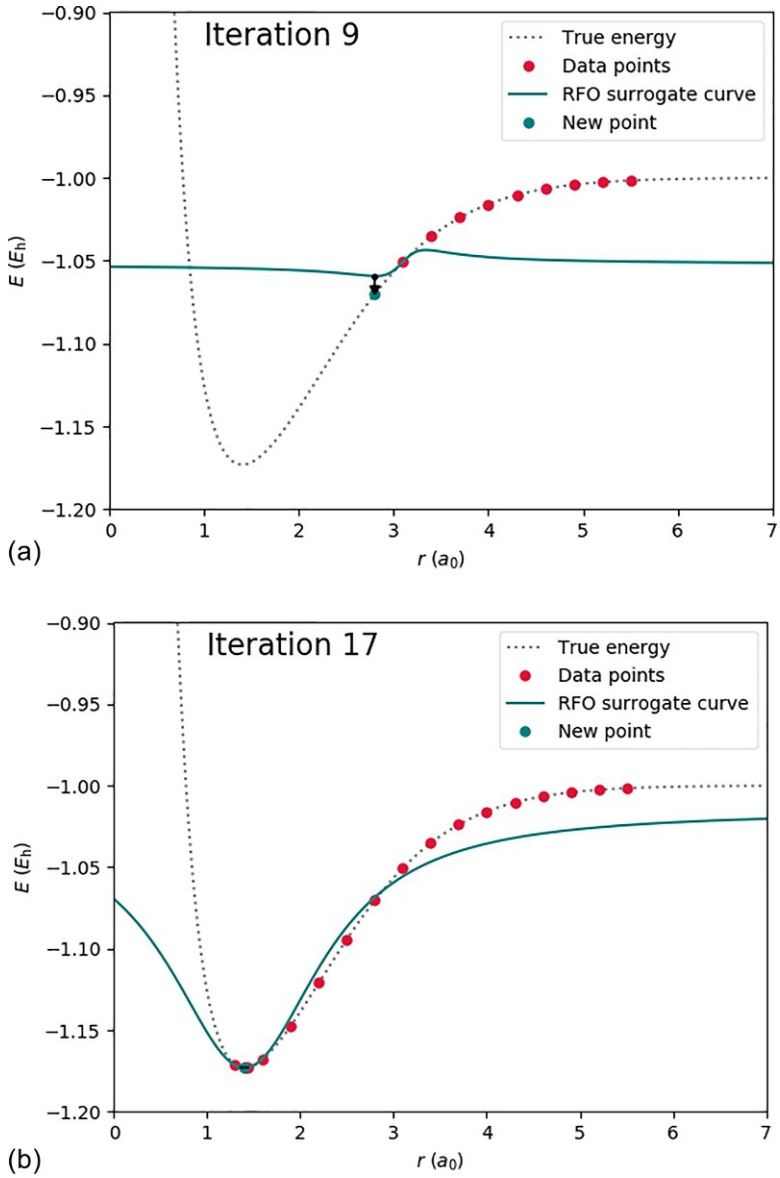


FIG. 6 Two graphs from the H_2 RS-RFO example. *Top*: Iteration 9. *Bottom*: At convergence.

is larger than the threshold, we use the `fsolve` function to find a point where the variance equals the threshold. This is somewhat different to the microiterations used in other implementations and discussed previously, but it is essentially the same result and allows for a more transparent implementation in this simple example.

With these functions in place, the optimization is very similar to RS-RFO, as given by the pseudocode in Algorithm 2.

ALGORITHM 2

RVO algorithm.

```
X ← initial r
Y ← E(X)
dY ← E′(X)
repeat
    l ←set_l(H_HMF(X))
    model.create(l,X,Y,dY) ▷ Build GEK model from previous data
    new ←var_restrict(model,X)                    ▷ Find next trial point
    X ← X ∪ new                                   ▷ Append new values
    Y ← Y ∪ E(new)
    dY ← dY ∪ E′(new)
until dY < threshold
```

If we run this (`optim_RVO.py`), we get something similar to the RS-RFO case, but now the estimated 95% confidence interval is plotted as a shaded area. Fig. 7 displays again the graph after a few iterations and at convergence. We note that the variance restriction is more obvious than the step restriction, but this is only because the step restriction modifies the surrogate curve, such that the next point is still its minimum, while the variance restriction only changes the way to choose the next point, and we clearly see that it is often not the minimum of the surrogate curve. In fact, in the example at the top of Fig. 7, the minimum would be somewhere on the negative r side, which is unphysical (this could also happen with RS-RFO, with longer step lengths). It is also clear that the variance restriction allows for longer step lengths as the surrogate model has more data and gets more confident. During the optimization, the surrogate curve closely follows the "true" curve between the last five points, and at convergence the fit around the minimum is much better than the RFO curve. Outside the region of the last five points, the surrogate curve and the variance rise strongly (although it is not always visible at this scale), this is due to the baseline or trend function we are using, which is at +9 E_h! At the end, at least for the settings used here (which match typical default settings in any molecular structure optimization) the RVO converges in about half the number iterations required with RS-RFO.

We hope that by playing with this simple example you will be able to better understand the similarities and differences between the two methods, and the relationship between the different settings.

Two-dimensional system (H_2O)

For the next example, we will use the OpenMolcas [91] software package, which offers implementations of both the RS-RFO and RFO methods. The system of choice is only slightly more complicated than H_2: a water molecule (H_2O). In this case, the molecular structure is fully determined by three coordinates ($3N - 6$, where N is the number of atoms), but the choice of these coordinates is not as trivial as for H_2. We could choose the three interatomic

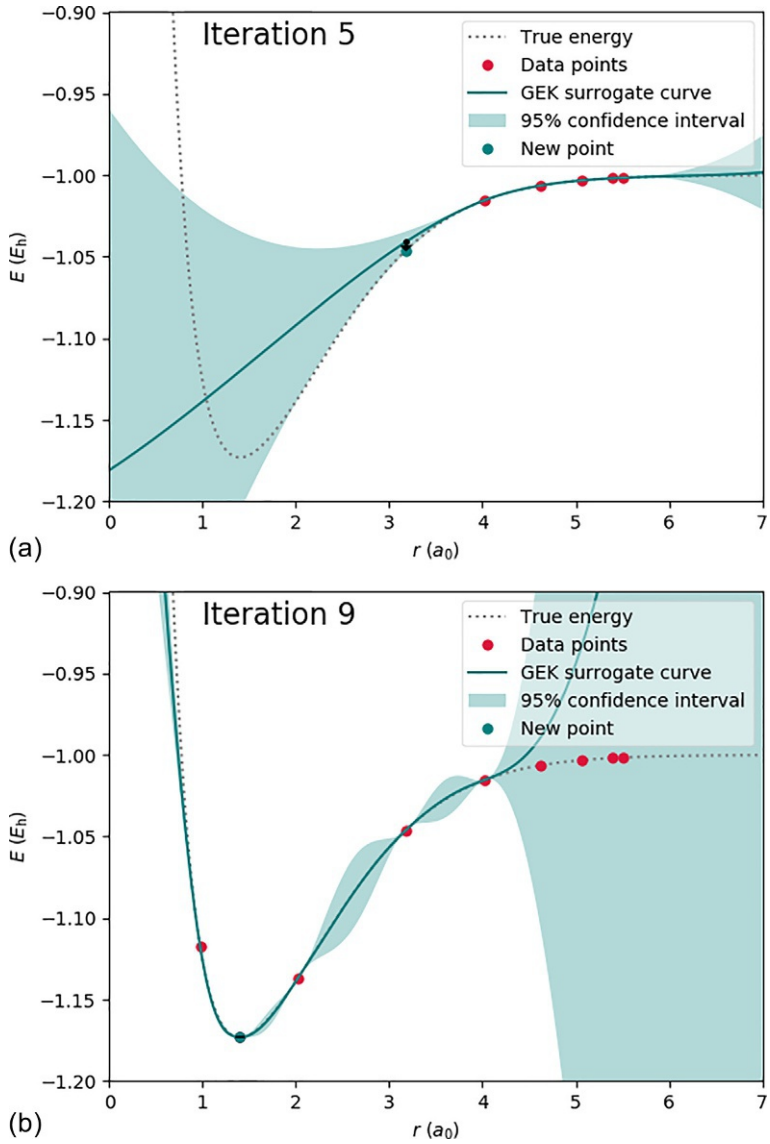


FIG. 7 Two graphs from the H_2 RVO example. *Top*: Iteration 5. *Bottom*: At convergence.

distances, or the three possible angles, or any combination thereof, or the H–O–H angle, the average O–H distance and the difference between the two O–H distances, etc. We will, however, limit the scope to structures with equal O–H distances, so in practice the system has only two degrees of freedom. For the discussion at least we will choose the H–O–H angle and the (equal) O–H distance as our coordinates.

We will examine two kinds of optimization: an unconstrained optimization searching for a local minimum on the PES, and a constrained optimization where we also search for a structure with minimal energy, but imposing some constraint on the geometry, in particular a

specific value for the H–O–H angle. In all cases, we will start from a structure with 1.4 Å O–H distance and an angle of 80 degrees.

An input file for a plain optimization is given in Listing 1. Briefly, the ampersands (&) indicate the beginning of the input of the different modules, and the `> Do While` and `> End Do` lines enclose a loop that will be repeated until convergence. The `&GATEWAY` module defines the system (Cartesian coordinates of the initial structure, basis set, and no symmetry enforced). The `&SEWARD`, `&SCF`, and `&SLAPAF` modules are invoked with their default values, and are used to compute the necessary integrals for the basis functions, to obtain the Hartree-Fock wave function, and to propose a new geometry for the next iteration. If OpenMolcas is properly installed, you should be able to run this as `pymolcas -f filename`.

Listing 1: Basic input for geometry optimization of H_2O with OpenMolcas.

```
&GATEWAY
  Coord = 3

    O    0.00000    0.00000    0.00000
    H    0.89990    1.07246    0.00000
    H   -0.89990    1.07246    0.00000
  Basis = cc-pVDZ
  Group = NoSym

> Do While
  &SEWARD
  &SCF
  &SLAPAF
> End Do
```

The basic input in Listing 1 will perform an RS-RFO optimization, for an RVO simply add `Kriging` below `&SLAPAF`. For a constrained optimization with either method, add the following to the `&GATEWAY` block:

```
Constraints
  a = Angle H2 O1 H3
Values
  a = 170 degrees
End of constraints
```

This defines a single constraint, named `a`, as the H–O–H angle, and we request that its value must be 170 degrees when converged.

Once these optimizations are run successfully, you should find in the output near the end (before a `&LAST_ENERGY` block that computes the energy for the converged structure) a table similar to Listing 2. In the RVO case, the `Geom Update` column should say `RVO` *n*, with *n* being the number of microiterations required, instead of `RS-RFO`, and the `Hessian Update` column should always be `None`.

Listing 2: Part of the OpenMolcas output for the RS-RFO optimization of H_2O.

```
**************************************************************************************************************
*                              Energy Statistics for Geometry Optimization                                     *
**************************************************************************************************************
                    Energy      Grad       Grad              Step                 Estimated   Geom      Hessian
Iter  Energy        Change      Norm       Max    Element    Max      Element     Final Energy Update Update Index
  1 -75.83687495  0.00000000 0.289466 -0.190847 nrc001 -0.297386* nrc001     -75.86560626 RS-RFO  None    0
  2 -75.89492441 -0.05804946 0.291155 -0.193396 nrc001 -0.298178* nrc001     -75.92373832 RS-RFO  BFGS    0
  3 -75.95096824 -0.05604384 0.266255 -0.178482 nrc001 -0.299548* nrc001     -75.97780593 RS-RFO  BFGS    0
  4 -75.99863829 -0.04767005 0.197909 -0.131147 nrc001 -0.462988  nrc001     -76.02947166 RS-RFO  BFGS    0
  5 -76.02562660 -0.02698831 0.058099 -0.035965 nrc002 -0.057691  nrc002     -76.02733220 RS-RFO  BFGS    0
  6 -76.02699014 -0.00136354 0.006461 -0.006394 nrc003 -0.020427  nrc003     -76.02705073 RS-RFO  BFGS    0
  7 -76.02705328 -0.00006315 0.000394  0.000220 nrc002 -0.001164  nrc003     -76.02705350 RS-RFO  BFGS    0
      +-------------------------------+-------------------------------+
      +    Cartesian Displacements    +    Gradient in internals      +
      + Value   Threshold Converged? + Value   Threshold Converged?  +
+-----+-------------------------------+-------------------------------+
+ RMS + 7.9653E-04  1.2000E-03  Yes   + 2.7882E-04  3.0000E-04  Yes    +
+-----+-------------------------------+-------------------------------+
+ Max + 8.1205E-04  1.8000E-03  Yes   + 2.1978E-04  4.5000E-04  Yes    +
+-----+-------------------------------+-------------------------------+
Geometry is converged in 7 iterations to a Minimum Structure
```

If we take the time to compute the energies of H_2O for a number of structures covering a range of angles and distance, we can visualize the PES (at this level of theory, i.e., HF/cc-pVDZ) and the behavior of the different optimizations. This is presented in the top row of Fig. 8. For reproducing the figures you would need to compute single-point energies for structures in a 20×20 grid of bond lengths and angles in order to obtain the PES contours, but these are not essential for the example. The lines showing the optimization progress can be obtained from the output of the optimizations. We observe that the two methods proceed very similarly for the unconstrained optimization (left panel), although RVO is able to take larger steps at the beginning, so it reaches the vicinity of the minimum earlier. The same happens for the constrained optimization (right panel), but it seems also that the path followed by the RS-RFO optimization is somewhat more erratic or oscillatory; in the case of RVO these oscillations would mostly be "hidden" in the microiterations.

The actual surrogate models used during the optimization are not explicitly provided in the output of OpenMolcas, but with some digging and modifications in the source code, it is possible to obtain all necessary information. (This part requires some familiarity with the OpenMolcas source code and methods, and it is done here only for illustration purposes. You could try to reproduce it if you feel adventurous.) The resulting models are displayed in the middle (RFO model) and bottom (GEK model used in RVO) rows of Fig. 8. The left panels show the surrogate models at the end of the unconstrained optimization, the right panels are for the constrained optimization. When compared with the actual PES in the top row, it is clear that the GEK model is able to capture much better general shape, and can give a good approximation for the minimum even when that area was not explored (bottom right panel). Nevertheless, it should be kept in mind that the purpose of these surrogate models is only to provide a local approximation to the surface near the target structure and not to globally fit the surface. For a global representation a more adequate sampling would have to be used.

Transition state optimization ($CH_3-CH=O \leftrightarrow CH_2=CH-OH$)

In this last example, we will locate the transition state for the keto-enol tautomerism of ethanal (acetaldehyde) and ethenol (vinyl alcohol). This transition state is represented by a first-order saddle point on the PES. The molecular system has seven atoms, so the total number of internal degrees of freedom is 15, and representing the full PES would require a 16-dimensional space. Since this is not feasible, we will not attempt it, and instead of representing only projections or slices, we will simply omit the PES and focus on simpler properties.

Locating saddle points is a significantly harder task than locating minima or maxima, and there are several methods and strategies available. Here we will use a technique that combines the constrained optimizations of PCO [36, 37] with the saddle-point homing of I-RFO [9]. The idea is to start with a constrained optimization to guide the structure toward the region where we believe a saddle point may exist and, once a surrogate model with the right curvature is found, the constrains are lifted and a "normal" saddle point optimization proceeds. In designing the initial constraints some amount of knowledge, luck and trial and error (what is usually called "chemical intuition") is often involved.

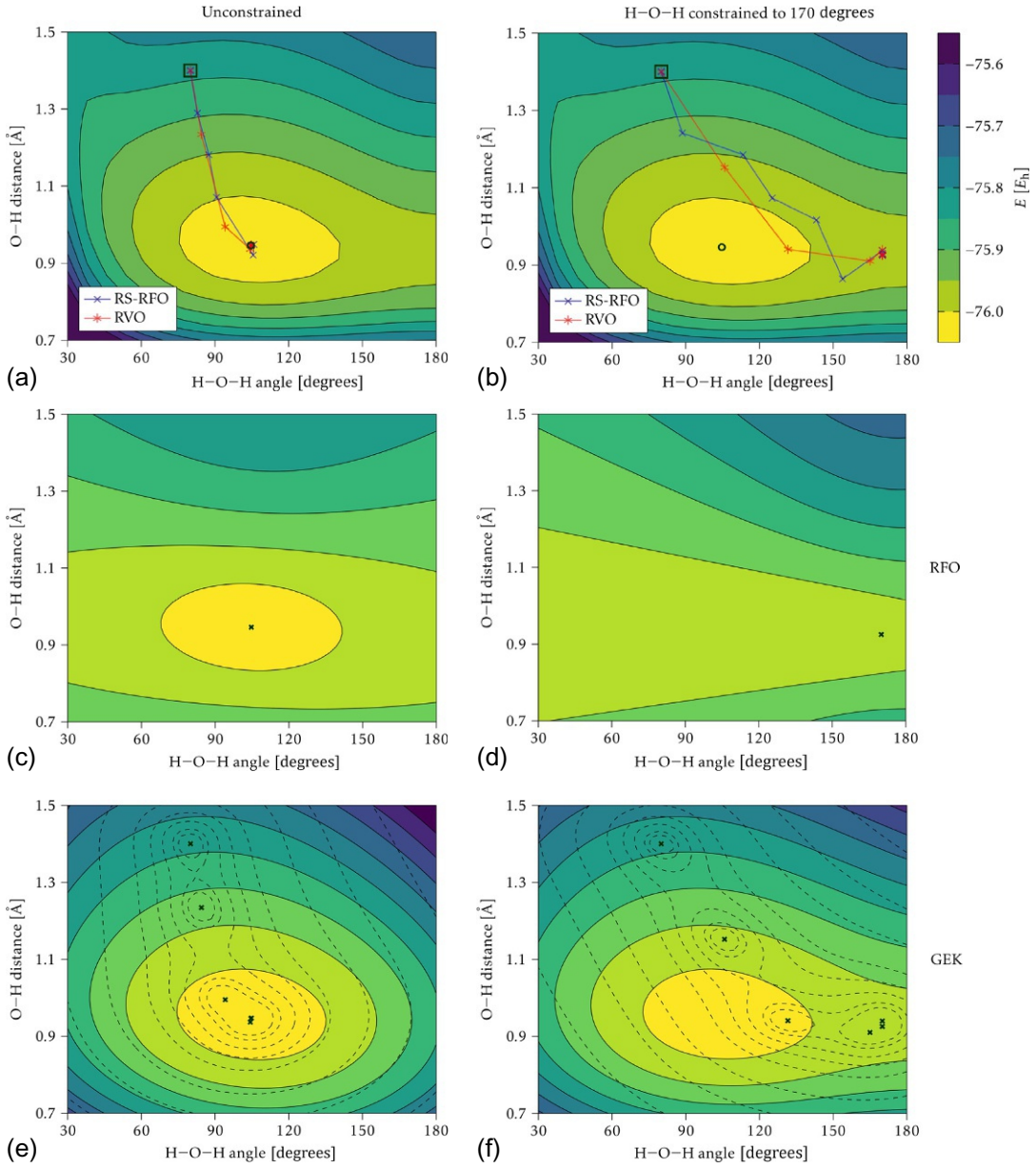


FIG. 8 *Top row*: PES and optimization results for the H_2O example. The *square* surrounds the starting structure, the *circle* marks the minimum of the PES. *Middle row*: Surrogate RFO model after convergence. *Bottom row*: Surrogate GEK model after convergence. The *dashed contours* represent the energy uncertainty for 95% confidence (values of 1, 2, and 5 times 10^n E_h, for $n = -3, -2, -1$). The *crosses* in the middle and bottom rows mark the data points that are used for defining the model.

An input example is given in Listing 3. Apart from trivial changes, like the different number and type of atoms, and the use of DFT method (`KSDFT = B3LYP`) instead of Hartree-Fock, the difference with the minimum optimization of Listing 1 is the `FindTS`, and `TSConstraints` keywords in `&SLAPAF`. The former enables the technique discussed earlier, and the latter specifies the constraints that will be active during the initial part of the process. In this case, since the reaction (starting from the keto) involves forming a new O–H bond and breaking a C–H bond, we specify distances that correspond to an almost formed O–H bond and a rather broken C–H bond, respectively.

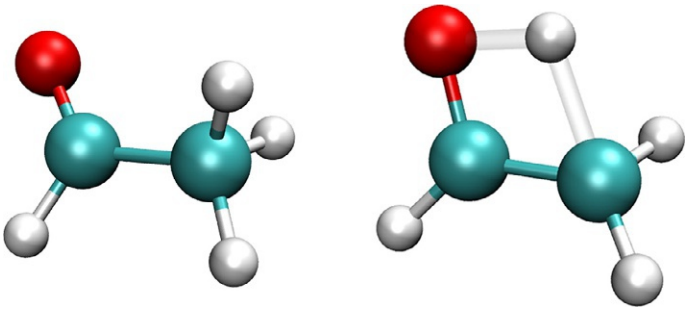
Listing 3: Basic input for TS optimization with RVO and OpenMolcas.

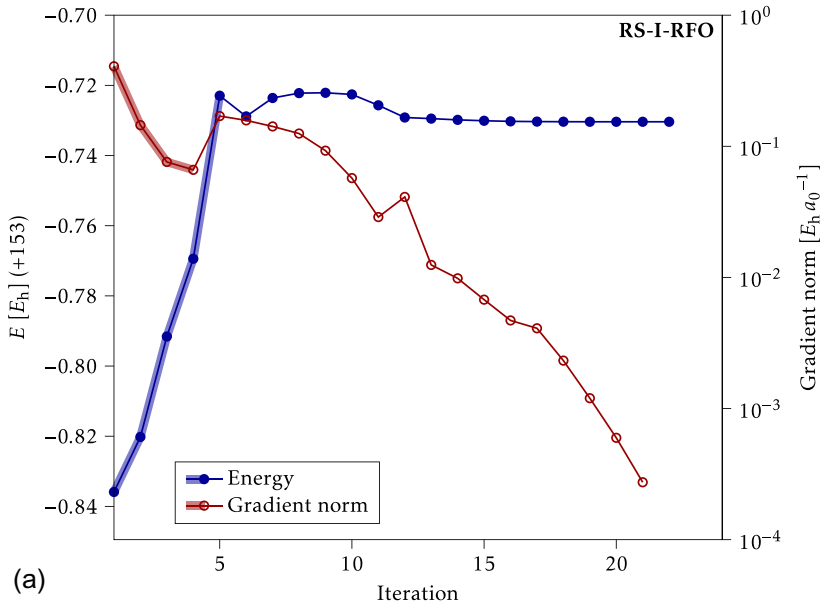
```
&GATEWAY
  Coord = 7
    C    0.96431771    -0.08453257    -0.00253431
    C    2.47833224    -0.05823981     0.03899534
    O    3.12403979    -0.20367373     1.04991836
    H    0.57334700     0.05878500    -1.02049162
    H    0.56720058     0.70514970     0.65700421
    H    0.60388459    -1.04683942     0.39754191
    H    2.99032520     0.10870718    -0.94642421
 Basis = cc-pVDZ
 Group = NoSym

> Do While
  &SEWARD
  &SCF
    KSDFT = B3LYP
  &SLAPAF
    Kriging
    FindTS
    TSConstraints
      b1 = Bond O3 H5
      b2 = Bond C1 H5
    Values
      b1 = 1.2 angstroms
      b2 = 1.7 angstroms
    End of TSConstraints
> End Do
```

If we run this calculation, we should obtain a converged structure for the saddle point, displayed in Fig. 9. We can compare the optimization progress with the RVO method and with the conventional RS-I-RFO method (with and without the `Kriging` keyword, respectively), and this is shown in Fig. 10. Again we see that RVO allows for a faster change toward satisfying the constraints (thicker lines), and also a faster approach to convergence, reducing the total number of iterations to one-third (14 vs. 21).

FIG. 9 Molecular structures for the C_2H_4O example. *Left*: keto minimum. *Right*: optimized transition state.





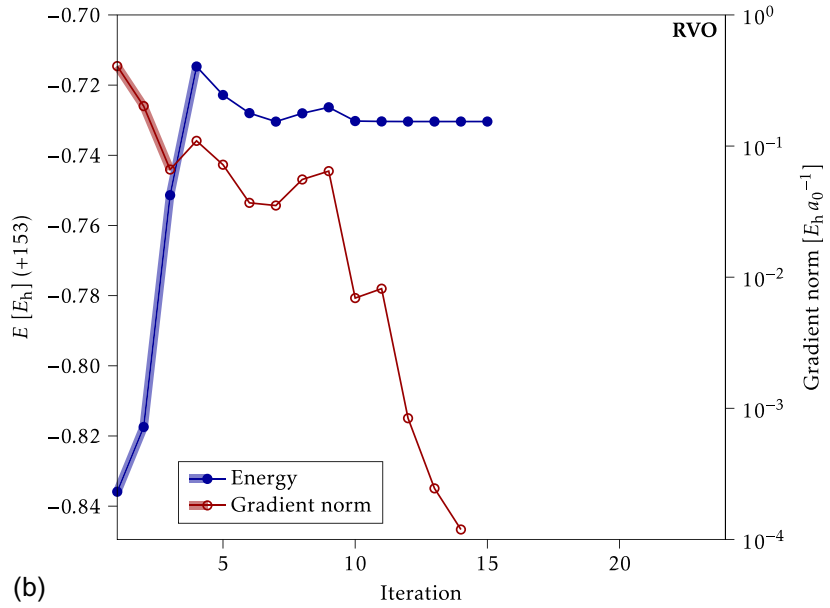


FIG. 10 Evolution of energy and gradient during the TS example optimization. *Top*: RS-I-RFO. *Bottom*: RVO. The *thicker lines* indicate the iterations where the guiding constraints are active.

It is worth reminding that the specified constraints are, for a successful TS optimization, never actually satisfied. At the point when they are turned off (iterations 4 and 5), the distance values are 2.07 and 1.34 Å with RS-I-RFO, and 1.81 and 1.45 Å with RVO (compare with the values in Listing 3). At the final TS structure, the respective distances are 1.30 and 1.51 Å.

Conclusions and outlook

In this chapter, we have presented the advancement of ML technology, especially GPR, in the field of molecular structure optimization. The initial driving force behind this was the need to replace expensive ab initio methods in the simulations of molecular dynamics or scattering processes. In this respect, both NN and kernel methods have been introduced and proven to be indispensable. In particular, it has been demonstrated that the GPR approach—which can learn as it goes and also through the benefits of an analytic expression of the expected dispersion—can guide a procedure toward optimal learning for the purpose of molecular structure optimization with as few data points as possible. In that respect, GPR is the optimal surrogate model for accurate local representations of the PES while optimizing molecular structures. Works over the last few years have explored this and the results have been impressive. The key to success has been the selection of coordinates in which the surrogate model is translational and rotational invariant, exploitation of the benefits of access to the estimated variance, individual characteristic length scales for each coordinate, and combining heuristics of the estimated Hessian into the ML procedure. The RVO implementation in OpenMolcas has all these qualities, where the later point is achieved by setting the characteristic length scale in a procedure, ignoring standard protocols to maximize the likelihood, in order for single-point GPR Hessian to reproduce an approximative guessed Hessian. The new procedure has aced standard restricted-step second-order quasi-Newton optimization procedures in equilibrium and transition state optimization, in optimizations in association with geometrical constraints, and in the computation of minimum reaction paths. Developments toward optimization with nongeometrical constraints as in intersystem crossings and conical intersections are on their way. Preliminary results show expected superior performance for the case of intersystem crossings, the latter is a bit more problematic. The possible reason could be the partial ambiguity in a unique and consistent way to present the branching space that optimal learning can be reached in few iterations. Despite these problems there is no reason to expect anything else than success. Considering that GPR only has been developed for molecular structure calculations over the last few years and that it already outperforms established methods, suggests that there must be more development potential. Further developments of GEK should be considered. In particular, the poor scaling with the number of degrees of freedom needs to be addressed. This bottleneck should be resolved in order to achieve these very impressive efficiency improvements in higher-dimensional problems such as macromolecular systems or the optimization of wave function parameters.

Acknowledgments

Funding from the Swedish Research Council (grants 2016-03398 and 2020-03182) and the Olle Engkvist foundation (grant 18-2006) are recognized.

References

[1] H.B. Schlegel, Geometry optimization, Wiley Interdiscip. Rev. Comput. Mol. Sci. 1 (5) (2011) 790–809, https://doi.org/10.1002/wcms.34.

[2] G. Raggi, I. Fdez. Galván, C.L. Ritterhoff, M. Vacher, R. Lindh, Restricted-variance molecular geometry optimization based on gradient-enhanced kriging, J. Chem. Theory Comput. 16 (6) (2020) 3989–4001, https://doi.org/10.1021/acs.jctc.0c00257.

[3] I. Fdez. Galván, G. Raggi, R. Lindh, Restricted-variance constrained, reaction path, and transition state molecular optimizations using gradient-enhanced kriging, J. Chem. Theory Comput. 17 (1) (2020) 571–582, https://doi.org/10.1021/acs.jctc.0c01163.

[4] W.C. Davidon, Variable metric method for minimization, 1959, https://doi.org/10.2172/4222000. Technical Report.

[5] R. Fletcher, M.J.D. Powell, A rapidly convergent descent method for minimization, Comput. J. 6 (2) (1963) 163–168, https://doi.org/10.1093/comjnl/6.2.163.

[6] A. Banerjee, N. Adams, J. Simons, R. Shepard, Search for stationary points on surfaces, J. Phys. Chem. 89 (1) (1985) 52–57, https://doi.org/10.1021/j100247a015.

[7] J. Baker, An algorithm for the location of transition states, J. Comput. Chem. 7 (4) (1986) 385–395, https://doi.org/10.1002/jcc.540070402.

[8] C.M. Smith, How to find a saddle point, Int. J. Quantum Chem. 37 (6) (1990) 773–783, https://doi.org/10.1002/qua.560370606.

[9] T. Helgaker, Transition-state optimizations by trust-region image minimization, Chem. Phys. Lett. 182 (5) (1991) 503–510, https://doi.org/10.1016/0009-2614(91)90115-p.

[10] R. Fletcher, Practical Methods of Optimization, John Wiley & Sons, Ltd, 1987, https://doi.org/10.1002/9781118723203.

[11] V. Bakken, T. Helgaker, The efficient optimization of molecular geometries using redundant internal coordinates, J. Chem. Phys. 117 (20) (2002) 9160–9174, https://doi.org/10.1063/1.1515483.

[12] E. Besalú, J.M. Bofill, On the automatic restricted-step rational-function-optimization method, Theor. Chem. Acc. 100 (5–6) (1998) 265–274, https://doi.org/10.1007/s002140050387.

[13] H.B. Schlegel, Estimating the Hessian for gradient-type geometry optimizations, Theor. Chim. Acta 66 (5) (1984) 333–340, https://doi.org/10.1007/bf00554788.

[14] T.H. Fischer, J. Almlöf, General methods for geometry and wave function optimization, J. Phys. Chem. 96 (24) (1992) 9768–9774, https://doi.org/10.1021/j100203a036.

[15] J. Baker, Techniques for geometry optimization: A comparison of Cartesian and natural internal coordinates, J. Comput. Chem. 14 (9) (1993) 1085–1100, https://doi.org/10.1002/jcc.540140910.

[16] J.D. Head, M.C. Zerner, An approximate Hessian for molecular geometry optimization, Chem. Phys. Lett. 131 (4–5) (1986) 359–366, https://doi.org/10.1016/0009-2614(86)87166-4.

[17] R. Lindh, A. Bernhardsson, G. Karlström, P.-Å. Malmqvist, On the use of a Hessian model function in molecular geometry optimizations, Chem. Phys. Lett. 241 (4) (1995) 423–428, https://doi.org/10.1016/0009-2614(95)00646-l.

[18] C.G. Broyden, The convergence of a class of double-rank minimization algorithms 1. General considerations, IMA J. Appl. Math. 6 (1) (1970) 76–90, https://doi.org/10.1093/imamat/6.1.76.

[19] R. Fletcher, A new approach to variable metric algorithms, Comput. J. 13 (3) (1970) 317–322, https://doi.org/10.1093/comjnl/13.3.317.

[20] D. Goldfarb, A family of variable-metric methods derived by variational means, Math. Comput. 24 (109) (1970) 23, https://doi.org/10.1090/s0025-5718-1970-0258249-6.

[21] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, Math. Comput. 24 (111) (1970) 647, https://doi.org/10.1090/s0025-5718-1970-0274029-x.

[22] B.A. Murtagh, Computational experience with quadratically convergent minimisation methods, Comput. J. 13 (2) (1970) 185–194, https://doi.org/10.1093/comjnl/13.2.185.

[23] J.M. Bofill, Updated Hessian matrix and the restricted step method for locating transition structures, J. Comput. Chem. 15 (1) (1994) 1–11, https://doi.org/10.1002/jcc.540150102.

[24] J.M. Bofill, Remarks on the updated Hessian matrix methods, Int. J. Quantum Chem. 94 (6) (2003) 324–332, https://doi.org/10.1002/qua.10709.

[25] H.L. Sellers, V.J. Klimkowski, L. Schäfer, Normal coordinate ab initio force relaxation, Chem. Phys. Lett. 58 (4) (1978) 541–544, https://doi.org/10.1016/0009-2614(78)80014-1.
[26] H.L. Sellers, J.F. Pinegar, L. Schäfer, Normal coordinate ab initio calculations of cubic anharmonicity constants, Chem. Phys. Lett. 61 (3) (1979) 499–502, https://doi.org/10.1016/0009-2614(79)87159-6.
[27] W.J. Hehre, L. Radom, P.vonR. Schleyer, J. Pople, Ab initio molecular orbital theory, John Wiley & Sons, Ltd, 1986. https://www.wiley.com/en-us/AB+INITIO+Molecular+Orbital+Theory-p-9780471812418.
[28] P. Pulay, G. Fogarasi, F. Pang, J.E. Boggs, Systematic ab initio gradient calculation of molecular geometries, force constants, and dipole moment derivatives, J. Am. Chem. Soc. 101 (10) (1979) 2550–2560, https://doi.org/10.1021/ja00504a009.
[29] G. Fogarasi, X. Zhou, P.W. Taylor, P. Pulay, The calculation of ab initio molecular geometries: efficient optimization by natural internal coordinates and empirical correction by offset forces, J. Am. Chem. Soc. 114 (21) (1992) 8191–8201, https://doi.org/10.1021/ja00047a032.
[30] P. Pulay, G. Fogarasi, Geometry optimization in redundant internal coordinates, J. Chem. Phys. 96 (4) (1992) 2856–2860, https://doi.org/10.1063/1.462844.
[31] R. Lindh, A. Bernhardsson, M. Schütz, Force-constant weighted redundant coordinates in molecular geometry optimizations, Chem. Phys. Lett. 303 (5–6) (1999) 567–575, https://doi.org/10.1016/s0009-2614(99)00247-x.
[32] N. Koga, K. Morokuma, Determination of the lowest energy point on the crossing seam between two potential surfaces using the energy gradient, Chem. Phys. Lett. 119 (5) (1985) 371–374, https://doi.org/10.1016/0009-2614(85)80436-x.
[33] M.R. Manaa, D.R. Yarkony, On the intersection of two potential energy surfaces of the same symmetry. Systematic characterization using a Lagrange multiplier constrained procedure, J. Chem. Phys. 99 (7) (1993) 5251–5256, https://doi.org/10.1063/1.465993.
[34] A. Farazdel, M. Dupuis, On the determination of the minimum on the crossing seam of two potential energy surfaces, J. Comput. Chem. 12 (2) (1991) 276–282, https://doi.org/10.1002/jcc.540120219.
[35] M.J. Bearpark, M.A. Robb, H.B. Schlegel, A direct method for the location of the lowest energy point on a potential surface crossing, Chem. Phys. Lett. 223 (3) (1994) 269, https://doi.org/10.1016/0009-2614(94)00433-1.
[36] J.M. Anglada, J.M. Bofill, A reduced-restricted-quasi-Newton-Raphson method for locating and optimizing energy crossing points between two potential energy surfaces, J. Comput. Chem. 18 (8) (1997) 992–1003, https://doi.org/10.1002/(sici)1096-987x(199706)18:8¡992::aid-jcc3¿3.0.co;2-l.
[37] L. De Vico, M. Olivucci, R. Lindh, New general tools for constrained geometry optimizations, J. Chem. Theory Comput. 1 (5) (2005) 1029–1037, https://doi.org/10.1021/ct0500949.
[38] P. Császár, P. Pulay, Geometry optimization by direct inversion in the iterative subspace, J. Mol. Struct. 114 (1984) 31–34, https://doi.org/10.1016/s0022-2860(84)87198-7.
[39] H.B. Schlegel, Exploring potential energy surfaces for chemical reactions: an overview of some practical methods, J. Comput. Chem. 24 (12) (2003) 1514–1527, https://doi.org/10.1002/jcc.10231.
[40] J. Cheng, A.N. Tegge, P. Baldi, Machine learning methods for protein structure prediction, IEEE Rev. Biomed. Eng. 1 (2008) 41–49, https://doi.org/10.1109/rbme.2008.2008239.
[41] A.W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A.W.R. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D.T. Jones, D. Silver, K. Kavukcuoglu, D. Hassabis, Improved protein structure prediction using potentials from deep learning, Nature 577 (7792) (2020) 706–710, https://doi.org/10.1038/s41586-019-1923-7.
[42] M. Torrisi, G. Pollastri, Q. Le, Deep learning methods in protein structure prediction, Comput. Struct. Biotechnol. J. 18 (2020) 1301–1310, https://doi.org/10.1016/j.csbj.2019.12.011.
[43] Q. Zhao, Z. Zhao, X. Fan, Z. Yuan, Q. Mao, Y. Yao, Review of machine learning methods for RNA secondary structure prediction, PLoS Comput. Biol. 17 (8) (2021) e1009291, https://doi.org/10.1371/journal.pcbi.1009291.
[44] K. Ryan, J. Lengyel, M. Shatruk, Crystal structure prediction via deep learning, J. Am. Chem. Soc. 140 (32) (2018) 10158–10168, https://doi.org/10.1021/jacs.8b03913.
[45] S. Wengert, G. Csányi, K. Reuter, J.T. Margraf, Data-efficient machine learning for molecular crystal structure prediction, Chem. Sci. 12 (12) (2021) 4536–4546, https://doi.org/10.1039/d0sc05765g.
[46] E. Mansimov, O. Mahmood, S. Kang, K. Cho, Molecular geometry prediction using a deep generative graph neural network, Sci. Rep. 9 (1) (2019), https://doi.org/10.1038/s41598-019-56773-5.
[47] D. Lemm, G.F. von Rudorff, O.A. von Lilienfeld, Machine learning based energy-free structure predictions of molecules, transition states, and solids, Nat. Commun. 12 (1) (2021), https://doi.org/10.1038/s41467-021-24525-7.

[48] M.Z. Makoś, N. Verma, E.C. Larson, M. Freindorf, E. Kraka, Generative adversarial networks for transition state geometry prediction, J. Chem. Phys. 155 (2) (2021) 024116, https://doi.org/10.1063/5.0055094.
[49] C. Peng, H.B. Schlegel, Combining synchronous transit and quasi-Newton methods to find transition states, Isr. J. Chem. 33 (4) (1993) 449–454, https://doi.org/10.1002/ijch.199300051.
[50] J. Behler, First principles neural network potentials for reactive simulations of large molecular and condensed systems, Angew. Chem. Int. Ed. 56 (42) (2017) 12828–12840, https://doi.org/10.1002/anie.201703114.
[51] O.T. Unke, S. Chmiela, H.E. Sauceda, M. Gastegger, I. Poltavsky, K.T. Schütt, A. Tkatchenko, K.-R. Müller, Machine learning force fields, Chem. Rev. 121 (16) (2021) 10142–10186, https://doi.org/10.1021/acs.chemrev.0c01111.
[52] A.A. Peterson, Acceleration of saddle-point searches with machine learning, J. Chem. Phys. 145 (7) (2016) 074106, https://doi.org/10.1063/1.4960708.
[53] A. Kamath, R.A. Vargas-Hernández, R.V. Krems, T. Carrington, S. Manzhos, Neural networks vs Gaussian process regression for representing potential energy surfaces: a comparative study of fit quality and vibrational spectrum accuracy, J. Chem. Phys. 148 (24) (2018) 241702, https://doi.org/10.1063/1.5003074.
[54] W. Liu, S. Batill, Gradient-enhanced response surface approximations using kriging models, in: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics, 2002, p. 5456, https://doi.org/10.2514/6.2002-5456.
[55] Z.-H. Han, S. Görtz, R. Zimmermann, Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function, Aerosp. Sci. Technol. 25 (1) (2013) 177–189, https://doi.org/10.1016/j.ast.2012.01.006.
[56] S. Ulaganathan, I. Couckuyt, T. Dhaene, J. Degroote, E. Laermans, Performance study of gradient-enhanced Kriging, Eng. Comput. 32 (1) (2016) 15–34, https://doi.org/10.1007/s00366-015-0397-y.
[57] D.G. Krige, A statistical approach to some basic mine valuation problems on the Witwatersrand, J. South. Afr. Inst. Min. Metall. 52 (6) (1951) 119–139.
[58] G. Matheron, Principles of geostatistics, Econ. Geol. 58 (8) (1963) 1246–1266, https://doi.org/10.2113/gsecongeo.58.8.1246.
[59] S. Lorenz, A. Groß, M. Scheffler, Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks, Chem. Phys. Lett. 395 (4–6) (2004) 210–215, https://doi.org/10.1016/j.cplett.2004.07.076.
[60] J. Behler, M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, Phys. Rev. Lett. 98 (14) (2007) 146401, https://doi.org/10.1103/physrevlett.98.146401.
[61] N. Artrith, T. Morawietz, J. Behler, High-dimensional neural-network potentials for multicomponent systems: applications to zinc oxide, Phys. Rev. E 83 (15) (2011) 153101, https://doi.org/10.1103/physrevb.83.153101.
[62] L. Raff, R. Komanduri, M. Hagan, S. Bukkapatnam, Neural Networks in Chemical Reaction Dynamics, Oxford University Press, 2012, https://doi.org/10.1093/oso/9780199765652.001.0001.
[63] Y. Yang, O.A. Jiménez-Negrón, J.R. Kitchin, Machine-learning accelerated geometry optimization in molecular simulation, J. Chem. Phys. 154 (23) (2021) 234704, https://doi.org/10.1063/5.0049665.
[64] Z.D. Pozun, K. Hansen, D. Sheppard, M. Rupp, K.-R. Müller, G. Henkelman, Optimizing transition states via kernel-based machine learning, J. Chem. Phys. 136 (17) (2012) 174101, https://doi.org/10.1063/1.4707167.
[65] K. Ahuja, W.H. Green, Y.-P. Li, Learning to optimize molecular geometries using reinforcement learning, J. Chem. Theory Comput. 17 (2) (2021) 818–825, https://doi.org/10.1021/acs.jctc.0c00971.
[66] A.P. Bartók, M.C. Payne, R. Kondor, G. Csányi, Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons, Phys. Rev. Lett. 104 (13) (2010) 136403, https://doi.org/10.1103/physrevlett.104.136403.
[67] A.P. Bartók, G. Csányi, Gaussian approximation potentials: a brief tutorial introduction, Int. J. Quantum Chem. 115 (16) (2015) 1051–1057, https://doi.org/10.1002/qua.24927.
[68] J. Behler, Perspective: machine learning potentials for atomistic simulations, J. Chem. Phys. 145 (17) (2016) 170901, https://doi.org/10.1063/1.4966192.
[69] R. Meyer, K.S. Schmuck, A.W. Hauser, Machine learning in computational chemistry: an evaluation of method performance for nudged elastic band calculations, J. Chem. Theory Comput. 15 (11) (2019) 6513–6523, https://doi.org/10.1021/acs.jctc.9b00708.
[70] J. Cui, R.V. Krems, Gaussian process model for collision dynamics of complex molecules, Phys. Rev. Lett. 115 (7) (2015) 073202, https://doi.org/10.1103/physrevlett.115.073202.

[71] R. Krems, J. Cui, Z. Li, Machine learning for molecular scattering dynamics: Gaussian Process models for improved predictions of molecular collision observables, in: APS Division of Atomic, Molecular and Optical Physics Meeting Abstracts, 2016, 2016, p. Q1.190. vol.
[72] J. Cui, Z. Li, R.V. Krems, Gaussian process model for extrapolation of scattering observables for complex molecules: from benzene to benzonitrile, J. Chem. Phys. 143 (15) (2015) 154101, https://doi.org/10.1063/1.4933137.
[73] J. Cui, R.V. Krems, Efficient non-parametric fitting of potential energy surfaces for polyatomic molecules with Gaussian processes, J. Phys. B At. Mol. Opt. Phys. 49 (22) (2016) 224001, https://doi.org/10.1088/0953-4075/49/22/224001.
[74] B. Minasny, A.B. McBratney, The Matérn function as a general model for soil variograms, Geoderma 128 (3–4) (2005) 192–207, https://doi.org/10.1016/j.geoderma.2005.04.003.
[75] B. Kolb, P. Marshall, B. Zhao, B. Jiang, H. Guo, Representing global reactive potential energy surfaces using Gaussian processes, J. Phys. Chem. A 121 (13) (2017) 2552–2557, https://doi.org/10.1021/acs.jpca.7b01182.
[76] O.-P. Koistinen, E. Maras, A. Vehtari, H. Jónsson, Minimum energy path calculations with Gaussian process regression, Nanosyst. Phys. Chem. Math. 7 (6) (2016) 925–935, https://doi.org/10.17586/2220-8054-2016-7-6-925-935.
[77] O.-P. Koistinen, V. Ásgeirsson, A. Vehtari, H. Jónsson, Nudged elastic band calculations accelerated with Gaussian process regression based on inverse interatomic distances, J. Chem. Theory Comput. 15 (12) (2019) 6738–6751, https://doi.org/10.1021/acs.jctc.9b00692.
[78] G. Schmitz, O. Christiansen, Gaussian process regression to accelerate geometry optimizations relying on numerical differentiation, J. Chem. Phys. 148 (24) (2018) 241704, https://doi.org/10.1063/1.5009347.
[79] A. Denzel, J. Kästner, Gaussian process regression for geometry optimization, J. Chem. Phys. 148 (9) (2018) 094114, https://doi.org/10.1063/1.5017103.
[80] A. Denzel, J. Kästner, Gaussian process regression for transition state search, J. Chem. Theory Comput. 14 (11) (2018) 5777–5786, https://doi.org/10.1021/acs.jctc.8b00708.
[81] D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. 45 (1–3) (1989) 503–528, https://doi.org/10.1007/bf01589116.
[82] O.-P. Koistinen, F.B. Dagbjartsdóttir, V. Ásgeirsson, A. Vehtari, H. Jónsson, Nudged elastic band calculations accelerated with Gaussian process regression, J. Chem. Phys. 147 (15) (2017) 152720, https://doi.org/10.1063/1.4986787.
[83] J.A. Garrido Torres, P.C. Jennings, M.H. Hansen, J.R. Boes, T. Bligaard, Low-scaling algorithm for nudged elastic band calculations using a surrogate machine learning model, Phys. Rev. Lett. 122 (15) (2019) 156001, https://doi.org/10.1103/physrevlett.122.156001.
[84] E. Garijo del Río, J.J. Mortensen, K.W. Jacobsen, Local Bayesian optimizer for atomic structures, Phys. Rev. E 100 (10) (2019) 104103, https://doi.org/10.1103/physrevb.100.104103.
[85] A. Denzel, B. Haasdonk, J. Kästner, Gaussian process regression for minimum energy path optimization and transition state search, J. Phys. Chem. A 123 (44) (2019) 9600–9611, https://doi.org/10.1021/acs.jpca.9b08239.
[86] O.-P. Koistinen, V. Ásgeirsson, A. Vehtari, H. Jónsson, Minimum mode saddle point searches using Gaussian process regression with inverse-distance covariance function, J. Chem. Theory Comput. 16 (1) (2019) 499–509, https://doi.org/10.1021/acs.jctc.9b01038.
[87] A. Denzel, J. Kästner, Hessian matrix update scheme for transition state search based on Gaussian process regression, J. Chem. Theory Comput. 16 (8) (2020) 5083–5089, https://doi.org/10.1021/acs.jctc.0c00348.
[88] D. Born, J. Kästner, Geometry optimization in internal coordinates based on Gaussian process regression: comparison of two approaches, J. Chem. Theory Comput. 17 (9) (2021) 5955–5967, https://doi.org/10.1021/acs.jctc.1c00517.
[89] R. Meyer, A.W. Hauser, Geometry optimization using Gaussian process regression in internal coordinate systems, J. Chem. Phys. 152 (8) (2020) 084112, https://doi.org/10.1063/1.5144603.
[90] D.R. Jones, A taxonomy of global optimization methods based on response surfaces, J. Glob. Optim. 21 (4) (2001) 345–383, https://doi.org/10.1023/a:1012771025575.
[91] I. Fdez. Galván, M. Vacher, A. Alavi, C. Angeli, F. Aquilante, J. Autschbach, J.J. Bao, S.I. Bokarev, N.A. Bogdanov, R.K. Carlson, L.F. Chibotaru, J. Creutzberg, N. Dattani, M.G. Delcey, S.S. Dong, A. Dreuw, L. Freitag, L.M. Frutos, L. Gagliardi, F. Gendron, A. Giussani, L. González, G. Grell, M. Guo, C.E. Hoyer, M. Johansson, S. Keller, S. Knecht, G. Kovačević, E. Källman, G. Li Manni, M. Lundberg, Y. Ma, S. Mai, J.P. Malhado, P.Å. Malmqvist, P. Marquetand, S.A. Mewes, J. Norell, M. Olivucci, M. Oppel, Q.M. Phung, K. Pierloot, F. Plasser, M. Reiher, A.M. Sand, I. Schapiro, P. Sharma, C.J. Stein, L.K. Sørensen, D.G. Truhlar, M. Ugandi, L. Ungur, A. Valentini, S. Vancoillie, V. Veryazov, O. Weser, T.A. Wesołowski, P.-O. Widmark, S.

Wouters, A. Zech, J.P. Zobel, R. Lindh, OpenMolcas: from source code to insight, J. Chem. Theory Comput. 15 (11) (2019) 5925–5964, https://doi.org/10.1021/acs.jctc.9b00532.

[92] J. Baker, F. Chan, The location of transition states: a comparison of Cartesian, Z-matrix, and natural internal coordinates, J. Comput. Chem. 17 (7) (1996) 888–904, https://doi.org/10.1002/(sici)1096-987x(199605)17:7¡888::aid-jcc12¿3.0.co;2-7.

[93] P. Jurečka, J. Šponer, J. Černý, P. Hobza, Benchmark database of accurate (MP2 and CCSD(T) complete basis set limit) interaction energies of small model complexes, DNA base pairs, and amino acid pairs, Phys. Chem. Chem. Phys. 8 (17) (2006) 1985–1993, https://doi.org/10.1039/b600027d.

[94] J. Kästner, J.M. Carr, T.W. Keal, W. Thiel, A. Wander, P. Sherwood, DL-FIND: an open-source geometry optimizer for atomistic simulations, J. Phys. Chem. A 113 (43) (2009) 11856–11865, https://doi.org/10.1021/jp9028968.

[95] C.-L. Yang, Y.-J. Huang, X. Zhang, K.-L. Han, MRCI potential curve and analytical potential energy function of the ground state of H_2, J. Mol. Struct. Theochem. 625 (1–3) (2003) 289–293, https://doi.org/10.1016/s0166-1280(03)00031-9.