Theor Chem Acc (2016) 135:84
DOI 10.1007/s00214-016-1847-3

REGULAR ARTICLE

Exploration of some refinements to geometry optimization methods

Adam B. Birkholz[1] **· H. Bernhard Schlegel**[1]

Received: 15 January 2016 / Accepted: 18 February 2016 / Published online: 12 March 2016

Abstract The optimization of equilibrium geometries is a key first step in most investigations that utilize quantum chemistry calculations. We present three modifications to standard methods for geometry optimization: a flow chart approach to Hessian updating, a scaled RFO approach to controlling the step size and direction, and a quasi-rotation approach to handling the propagation of internal coordinate data over the course of the optimization. These modifications are compared against a standard optimization method. We provide a new test set for geometry optimization consisting of 20 structures which range from ca 20 atoms to ca 100 atoms.

Keywords Geometry optimization · Quasi–Newton · Hessian updating · RFO · Optimization test set

1 Introduction

Geometry optimization is an important tool in the computational chemistry toolbox and has become ubiquitous in modern studies of chemical properties and reactions. There are a wide variety of different algorithms that exist for optimization (see Ref. [1] for a recent review of methods), with the most common utilizing a combination of quasi-Newton steps in redundant internal coordinates [2, 3], with an approximate Hessian stored and updated using the gradient information computed at each step of the optimization. Methods also exist that avoid the storage and update of the Hessian matrix, such as the conjugate gradient [4] and limited memory quasi-Newton [5] optimizers. These approaches typically require more gradient evaluations to achieve convergence when compared with full storage quasi-Newton methods and are frequently used to optimize very large molecules where the memory costs associated with storing the Hessian are prohibitive. For a recent benchmark of these such methods, see Ref. [6].

For quasi-Newton optimizations, approximate, positive definite Hessian matrices [7] are typically used to avoid the cost of computing the full Hessian exactly. These approximate Hessians are then updated by the method of Broyden, Fletcher, Goldfarb and Shanno (BFGS [8–11]) when seeking a minimum structure, and either the Powell's symmetric Broyden (PSB [12]) or the symmetric rank 1 (SR1 [13]) updates, or some combination of the two [14], is used when attempting to locate a transition state. Additionally, sequence acceleration methods such as line searches and direct inversion of the iterative subspace (GDIIS [15]) are also used to reduce the number of potential energy surface (PES) calculations necessary to converge to the desired minimum or transition-state structure.

Since geometry optimization using quasi-Newton methods is a relatively mature field, improvements are likely to be modest. Nevertheless, it is worthwhile to explore the effects of various modifications to existing optimization algorithms. Described below are three refinements that may offer additional benefits over existing methodologies for locating minimum energy structures using quasi-Newton optimizers.

- Flowchart update—This approach seeks to improve Hessian updating by using different update methods only

Electronic supplementary material The online version of this article (doi:10.1007/s00214-016-1847-3) contains supplementary material, which is available to authorized users.

✉ H. Bernhard Schlegel
hbs@chem.wayne.edu

[1] Department of Chemistry, Wayne State University, Detroit, MI 48202, USA

when they are expected to be well behaved, and falling back to more reliable but less ideal updates when necessary. Additionally, a new modification to the PSB method is used by using scaled displacements to compute the update.

- Scaled RFO method—This approach seeks to improve the use of the rational function optimization method for controlling step size and direction by modifying the shift matrix to better represent the expected relative stiffness of the bond stretches versus the other coordinates.
- Quasi-rotation method—This is an alternative approach to handling the redundancy in an internal coordinate system. Rather than store the approximate Hessian in the full redundant space, a quasi-rotation matrix is used to rotate the approximate Hessian from the non-redundant space at one point, to the non-redundant space at another. In addition to reducing the memory requirements for storing the Hessian for large systems, this could also lead to a more consistent approximation to the Hessian even when the non-redundant space changes over the course of the optimization and may help improve Hessian updating since the change in the gradient can be expressed entirely in the non-redundant space at one set of coordinates.

In the appendix, we summarize the procedures that have become standard over the past few decades for the optimization of molecular geometries using the quasi-Newton method. In Sects. 2–4, we present some new contributions to improve the stability and efficiency of such optimizations. The standard method will serve as the benchmark for evaluating the performance of each of the modifications to the standard optimization method.

To test the performance of the geometry optimization methods, we have compiled a new set of 20 mol with between 10 and 50 heavy atoms (Fig. 1). This test set is more representative of molecules typically being optimized using ab initio and DFT methods on modern architectures than the older test sets that were developed according to the computer time available in previous decades [16, 17].

2 Flowchart update

2.1 Motivation

Three of the most commonly used Hessian update formulae are the BFGS, SR1 and PSB updates. They use the displacement ($\mathbf{s} = \mathbf{\Delta q}$), change in gradient ($\mathbf{y} = \mathbf{\Delta g}_q$), and quadratic error ($\mathbf{z} = \mathbf{y} - \mathbf{Hs}$) to correct the approximation to the Hessian according to data computed at two separate geometries.

$$\mathbf{\Delta H}_{BFGS} = \frac{\mathbf{yy}^T}{\mathbf{y}^T\mathbf{s}} - \frac{\mathbf{sHHs}^T}{\mathbf{sHs}} \tag{1}$$

$$\mathbf{\Delta H}_{SR1} = \frac{\mathbf{zz}^T}{\mathbf{z}^T\mathbf{s}} \tag{2}$$

$$\mathbf{\Delta H}_{PSB} = \frac{\mathbf{sz}^T + \mathbf{zs}^T}{\mathbf{s}^T\mathbf{s}} - \mathbf{s}^T\mathbf{z}\frac{\mathbf{ss}^T}{\left(\mathbf{s}^T\mathbf{s}\right)^2} \tag{3}$$

It is fairly well understood that the BFGS (Eq. 1) and SR1 (Eq. 2) updates can become numerically unstable when $\mathbf{y}^T\mathbf{s}$ and $\mathbf{z}^T\mathbf{s}$ are small, respectively. The PSB update (Eq. 3), on the other hand, is always numerically stable since the only quantity in the denominator ($\mathbf{s}^T\mathbf{s}$) is always nonzero for a finite step, but may have undesirable properties when the quadratic error is large. After observing that, in general, the SR1 update produced the most reasonable results when $\mathbf{z}^T\mathbf{s}$ is less than 0, the following flowchart method was developed.

2.2 Method

2.2.1 Flowchart method

1. If $\frac{\mathbf{z}^T\mathbf{s}}{|\mathbf{z}||\mathbf{s}|} < -0.1$, use the SR1 update.
2. If $\frac{\mathbf{y}^T\mathbf{s}}{|\mathbf{y}||\mathbf{s}|} > 0.1$, use the BFGS update.
3. Otherwise, use PSB method.

This has the benefit of attempting to use the SR1 and BFGS methods, which are often far superior to the PSB method for minimization, as much as possible, but relying on the numerical stability of PSB when necessary.

2.2.2 SSB method

All Hessian update methods are based upon computing a correction to the Hessian which satisfies the secant equation

$$(\mathbf{H} + \mathbf{\Delta H})\mathbf{s} = \mathbf{y} \tag{4}$$

Whenever the dimensionality of the PES is greater than 1, this equation is under-determined for the correction ($\mathbf{\Delta H}$) and the imposition of different constraints, such as the requirement that the correction be symmetric, and that it has a minimum size according to some metric, leads to the different update formula. Since all update methods satisfy Eq. 4, the difference in performance between the methods must depend on how they treat the remaining space. For SR1 and BFGS, the numerator is constructed from the outer products of vectors that resemble gradient change terms ($\mathbf{y}$, $\mathbf{Hs}$ or $\mathbf{z}$), while the numerator for the PSB formula is constructed from outer products of displacements or displacements and gradient change terms. Similarly, the denominators in the SR1/BFGS updates are scalar products of the displacement and a gradient change terms, while the denominators in the PSB update are scalar products of the displacement alone. If these features play a role in the

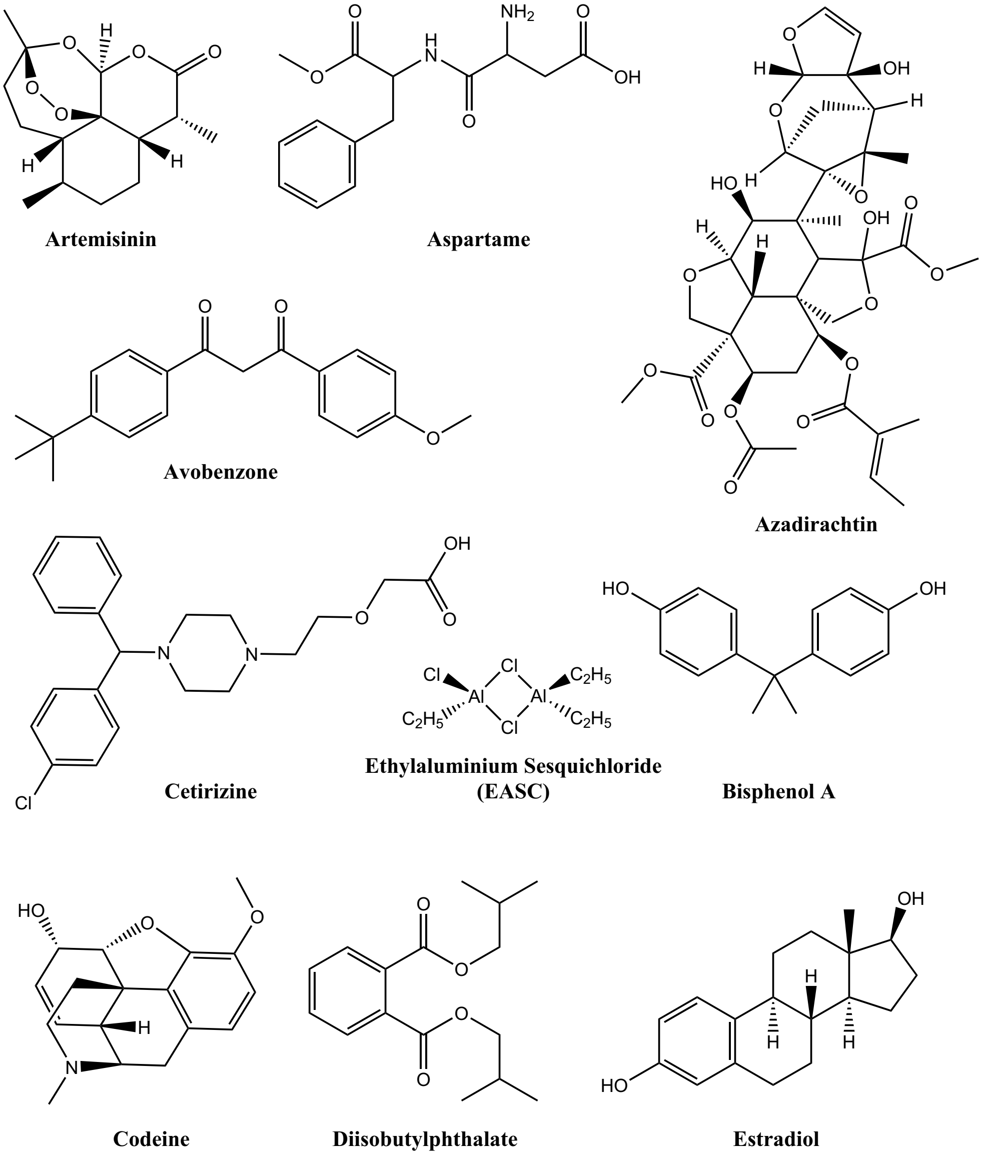

Fig. 1 Structures for test set

improved behavior observed with the SR1/BFGS updates, then perhaps the PSB update can be modified to produce more reasonable updates as well.

A more general symmetric rank 2 update can be constructed using any vector **v** as follows

$$\boldsymbol{\Delta}\mathbf{H}_{SR2} = \frac{\mathbf{v}\mathbf{z}^T + \mathbf{z}\mathbf{v}^T}{\mathbf{v}^T\mathbf{s}} - \mathbf{s}^T\mathbf{z}\frac{\mathbf{v}\mathbf{v}^T}{\left(\mathbf{v}^T\mathbf{s}\right)^2} \tag{5}$$

This update is valid for any choice of **v** that has a nonzero overlap with **s**, with $\mathbf{v} = \mathbf{s}$ giving the PSB update, and $\mathbf{v} = \mathbf{z}$ giving the SR1 update. Greenstadt [18] observed that the choice of $\mathbf{v} = \mathbf{M}\mathbf{s}$, where **M** is a positive definite weighting matrix, minimizes the magnitude of update matrix relative to **M**. By letting **M** be the approximate Hessian that is used to initialize a quasi-Newton optimization, we obtain a sensible **v** that has characteristics of **y**, the change in the gradient, while also having a nonzero overlap with **s**, the displacement. This modified Greenstadt formula will be refered to as the scaled symmetric Broyden (SSB) update and is also tested in the flowchart method.

3 Scaled RFO method

3.1 Motivation

The shifted (quasi-)Newton methods control the direction and magnitude of a (quasi-)Newton step by a shift of the Hessian eigenvalues.

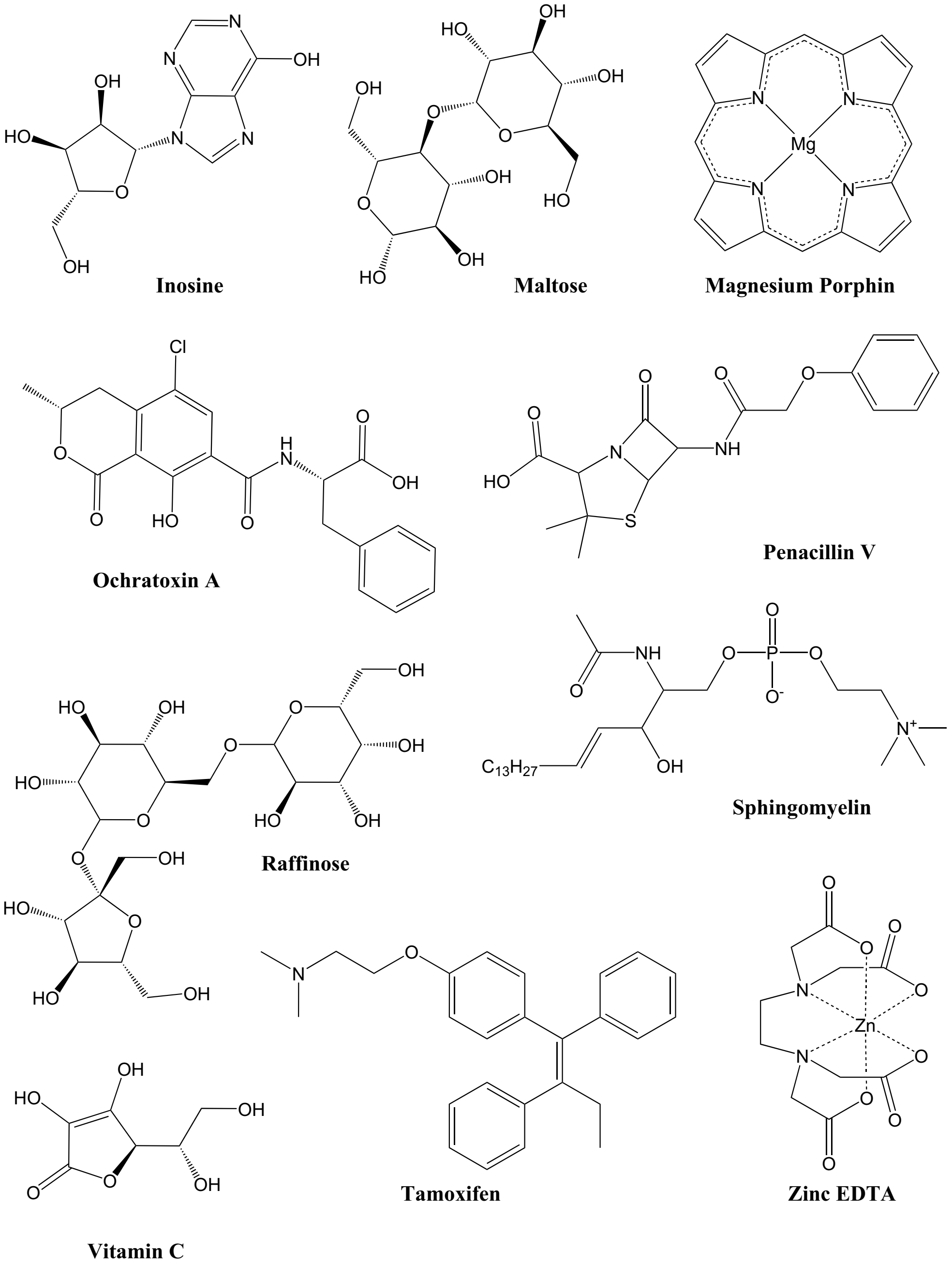

Fig. 1 continued

$$\Delta \mathbf{q}_r = -(\mathbf{H}_r + \xi \mathbf{I})^{-1} \mathbf{g}_r \tag{6}$$

where ξ is computed in such a way that the shifted quantity $\mathbf{H}_r + \xi \mathbf{I}$ has the correct number of negative eigenvalues (zero for minima, one for transition states) for the structure being optimized, and the subscript r indicates that the quantities are projected into the reduced, non-redundant space (see “Projection into non-redundant space” section in the Appendix for further details). The RFO method computes this shift factor as the negative of an eigenvalue (i.e., $\xi_{RFO} = -\lambda_{\text{aug}}$) of the Hessian augmented with the gradient (A more detailed discussion of the RFO method may be found in “RFO” section in Appendix.)

$$\mathbf{H}_{\text{aug}} = \begin{bmatrix} \mathbf{H}_r & \mathbf{g}_r \\ \mathbf{g}_r^T & 0 \end{bmatrix} \tag{7}$$

When the shift factor is large enough to dominate the Hessian, the result is a scaled steepest descent (or shallowest ascent, depending on the sign of the shift factor) step. This tends to be a poor step on a chemical PES, since the vibrational modes in a molecule tend to be a combination of both soft angular/torsional and stiff bond stretch coordinates, which can lead to oscillations and poor convergence behavior. Standard methods to compute the shift factor often result in a shift that is small enough that the Hessian remains dominant, and the vibrational modes are scaled appropriately to avoid the oscillatory behavior. Consideration of better ways to account for the stiffness of the vibrational modes when computing the direction and magnitude of the shifted (quasi-)Newton step has resulted in the scaled RFO method.

3.2 Method

While the RFO method is normally derived with a general matrix $\mathbf{S}$ (see "RFO" section in the Appendix), it is usually implemented with $\mathbf{S} = \mathbf{I}$. In the scaled RFO method, $\mathbf{S}$ is a positive definite matrix that is chosen to better account for the difference in stiffness between the coordinates. Additionally, for behavior similar to the RFO method, $\mathbf{S}$ should be normalized so that the magnitude of $\mathbf{S}$ is comparable to the magnitude of the identity matrix. This ensures that the only function of the $\mathbf{S}$ matrix is to change the contribution from each coordinate in the shifted matrix relative to one another without increasing or decreasing the magnitude of the shift itself. These requirements are met by using an approximate, positive definite Hessian $\tilde{\mathbf{H}}$ used to initialize the Hessian in a quasi-Newton optimization, projected into the active space at the current geometry ($\tilde{\mathbf{H}}_r = \mathbf{U}_{\text{act}}\tilde{\mathbf{H}}\mathbf{U}_{\text{act}}^T$), with the normalization factor computed using the inverse of the determinant.

$$\mathbf{S} = \frac{\widetilde{\mathbf{H}}_r}{|\widetilde{\mathbf{H}}_r|^{1/n_{\text{act}}}} \tag{8}$$

Löwdin orthogonalization [19] can be used to scale the Hessian, gradient and computed step, allowing for the computation of the shift parameter in the standard way.

$$\begin{aligned}\mathbf{\Delta q}_{SRFO} = \mathbf{S}^{-1/2}\Big(&\mathbf{S}^{-\mathbf{1/2}}\mathbf{H}_r\mathbf{S}^{-1/2} \\ &-\lambda_{RFO}\mathbf{I}\big)^{-1}\mathbf{S}^{-1/2}\mathbf{g}_r\end{aligned} \tag{9}$$

4 Quasi-rotation internal coordinate propagation

4.1 Motivation

Regardless of the coordinate system chosen, the number of internal degrees of freedom defining a molecular geometry is fixed at $n_{\text{act}} = n_{crt} - 6$ for nonlinear molecules. Whenever more than n_{act} primitive coordinates are used to define the molecular geometry, a local basis of n_{act} linear combinations of the primitive coordinates completely defines the allowed internal motions of the molecule, and the remaining coordinates orthogonal to that basis are redundant. This basis is considered local because it depends on the corresponding geometry, in the case of Cartesian coordinates, the rotational orientation defines the direction of the overall infinitesimal rotations, while with redundant internal coordinates, the local basis is given by the left singular vectors of the B-matrix (see "Transformation of gradient/Hessian" section in the Appendix), which is a function of the Cartesian coordinates.

A method for transforming the derivatives in the local basis for one geometry into the local basis for another geometry is outlined below. This reduces the amount of space required to store approximate Hessian information down to the amount required for a symmetric n_{act} matrix and may help keep approximate Hessian information relevant over longer optimizations when the local space changes dramatically. The transformation is constructed by projecting the B-matrix of a previous geometry into the local basis at the current geometry, and the resulting $n_{\text{act}} \times n_{\text{act}}$ matrix will have an unsigned determinant of approximately, but not less than, 1. For this reason, the transformation is referred to as a quasi-rotation coordinate propagation method.

4.2 Method

Instead of updating the Hessian in the full redundant space and then projecting into the non-redundant space at every step of the optimization, the Hessian may be stored and updated in only the active space. To account for the changes in the active space from one point to the next point, a quasi-rotation scheme is used. If $\mathbf{B}^{\text{old}}$ and $\mathbf{B}^{\text{new}}$ are the B-matrix computed at the previous and current point, respectively, while $\mathbf{U}_{\text{act}}^{\text{old}}$ and $\mathbf{U}_{\text{act}}^{\text{new}}$ are the active space at the previous and current point, the quasi-rotation matrix that transforms a matrix or vector from the active space at the previous point to the active space at the current point is given by

$$\mathbf{M}^{\text{rot}} = \mathbf{U}_{\text{act}}^{\text{old}}\mathbf{B}^{\text{old}}\left(\mathbf{U}_{\text{act}}^{\text{new}}\mathbf{B}^{\text{old}}\right)^{-1} \tag{10}$$

The rotated Hessian at the new point may be computed as follows

$$\mathbf{H}_{\text{act}}^{\text{rot}} = \left(\mathbf{M}^{\text{rot}}\right)^T\mathbf{H}_{\text{act}}^{\text{old}}\mathbf{M}^{\text{rot}} \tag{11}$$

This Hessian may then be updated by using modified $\mathbf{\Delta q}$ and $\mathbf{\Delta g}$ vectors

$$\mathbf{\Delta q}^{prj} = \mathbf{U}_{\text{act}}^{\text{new}}\left(\mathbf{q}^{\text{new}} - \mathbf{q}^{\text{old}}\right) \tag{12}$$

$$\mathbf{\Delta g}^{\text{rot}} = \mathbf{g}_r^{\text{new}} - \left(\mathbf{M}^{\text{rot}}\right)^T\mathbf{g}_r^{\text{old}} \tag{13}$$

The projected displacement will not be 100 % accurate, as some of the displacement may exist in the redundant space at the new point. However, this amount is typically small relative to the magnitude of the displacement vector, and the BFGS and SR1 updates only modify space that is nonzero in $\boldsymbol{\Delta}\mathbf{g}$ and $\mathbf{H}\boldsymbol{\Delta}\mathbf{q}$.

5 Results and discussion

The performance of the modified algorithms are compared to the standard algorithm outlined in the appendix: optimization in redundant internal coordinates using the BFGS formula to update the Hessian and RFO to compute the step. An adaptive trust region is also used to control the maximum step size, with the RFO step scaled back whenever the computed step would exceed the trust region. Six modified algorithms were constructed by replacing one component of the standard algorithm:

- Cartesian—Optimize using Cartesian coordinates rather than redundant internal coordinates
- QN—Use a quasi-Newton step (i.e., Eq. 27 with $\lambda_m = 0$) rather than the RFO step
- FlowPSB—Replace BFGS updating with the flowchart update, using the PSB update as the fallback option
- FlowSSB—Replace BFGS updating with the flowchart update, using the SSB update as the fallback option
- SRFO—Compute the step using the scaled RFO approach rather than the standard RFO step
- RotCrd—Store only the non-redundant $n_{\mathrm{act}} \times n_{\mathrm{act}}$ Hessian, using the quasi-rotation matrix to propagate the Hessian and gradient at the previous point prior to applying the update

In each case, the initial Hessian was estimated as a diagonal matrix in redundant internal coordinates, and the initial trust region was set to 0.5 bohr or radian. No ceiling on the maximum step size was found to be necessary, as steps longer than approximately 1 bohr were uncommon, particularly when RFO was employed, and so the trust region was limited in practice to a maximum of 2 bohr. For the Cartesian optimization, the initial Hessian was transformed to Cartesian coordinates using the B-matrix, $\widetilde{\mathbf{H}}_{\mathrm{cart}} = \mathbf{B}\widetilde{\mathbf{H}}_{\mathrm{int}}\mathbf{B}^T$, and the reduced space at each step is defined as the $3n_{\mathrm{cart}} - 6$ vectors orthogonal to the translation and rotation vectors defined in Eqs. 16 and 17. For the QN algorithm, if the Hessian is found to have a negative eigenvalue following the update, the optimization is terminated since the Hessian is not shifted prior to computing the step.

These algorithms were implemented in a Mathematica [20] program, using Gaussian 09 [21] to evaluate the potential energy surface. Table 1 shows the number of iterations, consisting of an energy and gradient calculation, that each method required to achieve the convergence criteria listed in Sect. 1 for the examples in the test set shown in Fig. 1 (initial coordinates of the structures are provided in the supporting information). So that the different algorithms could be surveyed quickly, Hartree–Fock with the 3–21 G basis set was used to compute the energies for most cases. The behavior for B3LYP [22–25] with a 6–31 G(d,p) basis set should be similar, with the latter used in the EASC and vitamin C optimizations. For comparison, the total number of iterations required to converge all of the structures in the test set is listed in Table 1 for each algorithm that successfully optimized the complete test set. Additionally, since the QN method fails to converge on five of the structures, the total number of iterations required to converge only the remaining 15 structures is also listed.

Immediately apparent from the table is that optimization in redundant internal coordinates is vastly superior to optimization in Cartesian coordinates. Simple quasi-Newton steps perform slightly better, in general, than the steps shifted by the RFO method; however, the RFO method benefits from greater stability due to the ability to produce a good optimization step even when the Hessian has negative eigenvalues. It is common in optimization programs using the quasi-Newton method to employ additional fail-safes, such as restarting the optimization with a new approximate Hessian or rejecting updates that result in an indefinite Hessian, but the performance and merits of these approaches were not studied in the present work. The SRFO method offers a level of efficiency that is closer to the unshifted quasi-Newton method, while still providing the reliability of the RFO method.

For the new methods introduced in this paper, the RotCrd method was the only one that required more iterations (701) than the standard method (689) to optimize the test set. In general, the unsigned determinant of the quasi-rotation matrix was <1.05 even for large steps; however, when the BFGS update produced an indefinite Hessian, the subsequent step not only tended to increase the energy, but the resulting quasi-rotation matrix had a determinant much larger than previous or subsequent steps, in some cases being as large as 1.2. Since corresponding poor updates were observed in the standard algorithm, it suggests that changes to the local basis may play some role in these poor updates, and that a large determinant of the quasi-rotation matrix may have some use in rejecting what is likely to be a poor step prior to evaluating the energy/gradient. Additionally, the results here indicate that the RotCrd method adequately handles changes in the active coordinates during the course of the optimization, suggesting that it may be worthwhile to use this approach in future investigations of new strategies that involve the evolution of the redundant internal coordinate set during the optimization.

Table 1 Number of iterations required to converge to a minimum structure

	Cartesian	QN	Standard	FlowPSB	FlowSSB	SRFO	RotCrd
Artemisinin	80	25	24	21	21	24	25
Aspartame	98[a]	26	30	30	30[d]	28[a]	27
Avobenzone	193	18[a]	43	41	42	43	47
Azadirachtin	216[a]	57	77	73	74	65	72
Bisphenola A	94	Fail[c]	31	32	31	21	30
Cetirizine	157[a]	31[a]	35	37	40	30	39
Codeine	70	30[b]	18	18	18[d]	18[b]	19
Diisobutylphthalate	150[a]	Fail[c]	27	27	27[d]	28[a]	29
Estradiol	55	25	21	21	21[d]	19	21
EASC	123	28	40	27	25	24	31
Inosine	96[a]	Fail[c]	41	46	42	34	41
Maltose	168[a]	36	35	36	37	36	37
Mg porphyrin	49	16	14	24	24[d]	16	14
Ochratoxin A	174[b]	Fail[c]	31	31	31[d]	43[b]	31
Penicillin V	183[b]	35[b]	31	32	32[d]	30[a]	32
Raffinose	223[b]	50[a]	53	55	55[d]	54[b]	55
Sphingomyelin	429[a]	Fail[c]	56	48	48[d]	51	67
Tamoxifen	238	24	35	35	35[d]	35	36
Vitamin C	48	18	18	18	18[d]	18	17
Zn EDTA	58	27	29	29	29[d]	27	31
Subtotal[e]	1959	446	503	497	501	467	503
Total	2902	–	689	681	680	644	701

[a] The optimization converged to a higher energy structure than the standard method

[b] The optimization converged to a lower energy structure than the standard method

[c] The updated Hessian had a negative eigenvalue, and so the quasi-Newton optimizer was terminated prior to convergence

[d] The symmetric Broyden update was never used in the FlowPSB optimization, so the optimization behavior of the FlowSSB method is identical

[e] Comparison of only the 15 structures which successfully converged using the QN algorithm

Both of the flowchart methods (FlowPSB and FlowSSB) and the SRFO method required fewer iterations to optimize all of the structures in the test set than the standard method. Only 8 of the 20 optimizations needed to fall back to the symmetric Broyden update in the flowchart method, so the differences in performance between the FlowPSB and FlowSSB methods are limited to the differences in those 8 structures with neither approach doing consistently better than the other. Even when the symmetric Broyden update was needed, it was only used once or twice during the course of the optimization, and so the BFGS and SR1 updates were well suited to these problems. It is possible that greater differences between the BFGS and the flowchart updates would be observed when the surface is less regular than ab initio or DFT surfaces (for example, using semi-empirical energies, or solvation models), and future investigation is warranted.

The SRFO method had the best overall performance of the seven methods in Table 1, requiring 45 fewer iterations to converge the test set than the standard approach. The SRFO method was also the only new method that converged to different structures in six of the cases. When the initial structure is high in energy and far from a minimum, there may be multiple minimum energy wells that are accessible, each containing a different stable conformer. Since the SRFO effectively emphasizes changes to dihedral and valence angles early in the optimization, it is reasonable to expect that it will occasionally find a different minimum than the standard RFO approach, even when using the same initial Hessian and same Hessian update formula. For the molecules where the SRFO method had the largest improvement over the standard method (Azadirachtin, Bisphenol A, EASC and Inosine), the SRFO and Standard methods converged to the same structure. Likewise, the only case where the SRFO method needed a significant number of additional iterations to converge (Ochratoxin A), the SRFO method produced a lower energy structure.

6 Summary

The present paper discusses various components of optimization methods, including coordinate systems, transformations, control of the step size and direction, and Hessian updating and provides a diverse set of 20 medium size molecules for testing optimization methods. The best current geometry optimization methods typically use redundant internal coordinates, BFGS updating of the Hessian, RFO and trust region updating for step size control. A number of new refinements have been proposed and tested: updating that switches between SR1, BFGS and PSB or SSB to obtain the most reliable update (FlowPSB and FlowSSB), scaled RFO (SRFO) that varies control of the step size and direction depending on the softness or stiffness of the modes, and quasi-rotation propagation of the internal coordinates (RotCrd). The latter is an initial approach to working in only the active space of the redundant internal coordinates which necessarily changes during the course of the optimization. The results for optimization of the molecules in the test set clearly show that redundant internal coordinates are superior to Cartesian coordinates. RFO is better than quasi-Newton since it can reach a minimum even when the Hessian has negative eigenvalues. The FlowPSB and FlowSSB methods are comparable to the standard BFGS updating in most cases. The SRFO approach takes about 7 % fewer steps for optimization of the test set.

Acknowledgments This work was supported by a Grant from the National Science Foundation (CHE1212281 and CHE1464450). Wayne State University's computing grid provided computational support.

Appendix: Standard method

Described below is an optimization algorithm composed of standard methods and is comparable to the ones used in various electron structure packages. This algorithm is used as a reference for comparing and evaluating the modifications described in the text. The core of the algorithm is the quasi-Newton method applied to the gradient of the potential energy, and the additional components include the choice of coordinates to optimize, how the Hessian (the matrix of second derivatives of the PES) is computed or approximated, and how the Newton method is modified in order to control step size and direction. While there are a variety of different approaches that have been used in the past, a very common combination of methods includes the use of a redundant set of internal coordinates comprised of bond stretches, bends and torsions, approximation of the full Hessian matrix along with the use of updating schemes to improve the approximation over the course of the optimization and the rational functional optimization and trust region schemes to control the step size and direction.

Geometry extrapolation methods, such as line searches and DIIS, are commonly used with an aim to improve the efficiency of an optimization. In practice, they can also have a significant influence on the stability of an optimization as well. Likewise, Hessian updating methods that consider the change in the gradient at the current geometry relative to many previous optimization steps are employed for similar reasons. The performance of these methods depends heavily on the choice of parameters, such as how many previous points should be considered in the DIIS extrapolation or the Hessian update step. In order to better evaluate the merits of the methods described in this paper on their own terms, no extrapolation methods were employed, and the Hessian was updated using only the information at the current point and the most recent point.

Coordinate transformation/projection

Coordinate definitions

The standard redundant internal coordinate (RIC) system includes bond stretches between all bonded atoms, angles between pairs of bonds that share an atom and the dihedral angles between the planes defined by any two angles that share a common bond. Typically, whether or not a pair of atoms is bonded can be determined automatically by comparing the distance between the atoms to a standard bond length; however, for the present work, the connectivity was explicitly provided for all examples in order to ensure that the intended coordinate system is used and the behavior of the resulting optimizations does not depend on any arbitrary numerical thresholds for bondedness.

Two additional coordinate definitions that are frequently used in adjunct to the standard stretch, bend and torsion coordinates are out-of-plane bends and linear angle bends. Out-of-plane bends describe how an atom moves relative to the plane defined by 3 other atoms, while linear angle bend coordinates describe the bending motion of 3 atoms which have become nearly or completely linear. Out-of-plane bend coordinates, such as an improper dihedral angle, may provide some benefit to an optimization but are generally unnecessary for structures containing more than 4 atoms as the relevant motions are well described by the redundancy in the standard RIC set. For the present work, no out-of-plane coordinates are used.

Linear bend coordinates, on the other hand, may be essential when the optimized structures have linear or near-linear angles. Inclusion of standard bend angle coordinates (and the corresponding dihedral angle coordinates) for an angle that has become near linear can be disastrous for an optimization, even when the remaining coordinates in the RIC set are sufficient to fully describe the molecule. Whether or not angles and their corresponding dihedrals can simply be omitted from the RIC definitions depends on how many atoms are bonded to the center atom in the linear angle and may be possible to do so with as few as 3 bonded atoms. When the local structure around a central atom is non-planar and there are at least 4 bonds, the linear angle coordinates can always be safely omitted since the local structure can be completely defined with bonds and nonlinear angles. In the standard method, linear bends were handled by the approach outlined in Sect. 1 whenever an angle exceeds 165° and the central atom is bonded to fewer than 5 atoms, which is assumed to be sufficient to ensure non-planarity around the central atom.

Transformation of gradient/Hessian

Since the derivatives of the PES are computed in terms of the absolute Cartesian coordinates of the atomic centers in a molecule, at each step of the optimization the gradient (and Hessian, if it is computed) must be transformed into internal coordinates. This transformation involves the multiplication of the Cartesian gradient by the generalized inverse of Wilson's B-Matrix, which is defined as the partial derivatives of the RIC with respect to a change in the Cartesian coordinates:

$$B_{ia} = \frac{\partial q_i}{\partial x_a} \rightarrow \delta q_i = \sum_a B_{ia}\delta x_a \tag{14}$$

where q_i are the internal coordinates, and x_a are the Cartesian coordinates.

$\mathbf{B}$ is an $n_{ric} \times n_{\text{cart}}$, where n_{ric} is the total number of redundant internal coordinates, and n_{crt} is the total number of Cartesian coordinates (3 times the number of atomic centers). The generalized inverse (also known as the Moore-Penrose pseudo-inverse) is a generalization of the concept of an inverse to non-square or singular matrices such that only the non-singular part of the matrix is inverted. Two approaches for computing the pseudo-inverse of the $\mathbf{B}$ matrix, in particular, are through inversion of B-squared ($\mathbf{B}^T\mathbf{B}$), and by singular value decomposition.

For the first approach, it should be recognized that $\mathbf{B}^T\mathbf{B}$ is a $n_{\text{cart}} \times n_{\text{cart}}$ singular matrix, and so care must be taken when inverting it. So long as the internal coordinates chosen to define $\mathbf{B}$ are sufficient to fully describe the internal motions of the molecule, the singular part of $\mathbf{B}^T\mathbf{B}$ corresponds to the infinitesimal rotations and translations of the molecule. A projector onto this space can be added to $\mathbf{B}^T\mathbf{B}$ without effecting the end result of the pseudo-inverse. This projector $\mathbf{P}_{TR}$ can be computed as a sum of the projector onto the individual infinitesimal translation ($\mathbf{t}_i$) and rotation ($\mathbf{r}_i$) vectors

$$\mathbf{P}_{TR} = \sum_i^3 \mathbf{t}_i\mathbf{t}_i^T + \mathbf{r}_i\mathbf{r}_i^T \tag{15}$$

where the portion of these vectors corresponding to the kth atom are given by

$$\mathbf{t}_{i,k} = \mathbf{e}_i \tag{16}$$

$$\mathbf{r}_{i,k} = \mathbf{x}_k \times \mathbf{e}_i \tag{17}$$

where × denotes the 3-dimensional cross product, $\mathbf{e}_i$ is the unit vector in the ith direction, and $\mathbf{x}_k$ are the 3-dimensional Cartesian coordinates for atom k. These vectors are not orthonormal with respect to each other, but for a nonlinear molecule, they fully span and are entirely contained within the 6 instantaneous translation and rotation degrees of freedom for the molecule defined by the coordinates $\mathbf{x}$. The generalized inverse may then be computed as

$$\mathbf{B}^- = \left(\mathbf{B}^T\mathbf{B} + \mathbf{P}_{TR}\right)^{-1}\mathbf{B}^T \tag{18}$$

The primary benefit of computing the generalized inverse this way is that $\left(\mathbf{B}^T\mathbf{B} + \mathbf{P}_{TR}\right)^{-1}$ allows the use of iterative inversion methods for when the size of the molecule is too large to invert the matrix directly and/or store the result. For the present work, however, the generalized inverse was computed using singular value decomposition (SVD). The SVD of $\mathbf{B}$ is given by $\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ is a square, unitary matrix of size n_{ric}, $\mathbf{V}$ is a unitary matrix of size n_{crt}, and $\boldsymbol{\Sigma}$ is a matrix of the appropriate dimensions that is zero everywhere except for the first $n_{crt} - 6$ diagonal elements. The generalized inverse may be computed as

$$\mathbf{B}^- = \mathbf{V}\boldsymbol{\Sigma}^-\mathbf{U}^T \tag{19}$$

where $\boldsymbol{\Sigma}^-$ is the transpose of $\boldsymbol{\Sigma}$, with the nonzero diagonal elements replaced by their inverse (i.e., $\left[\boldsymbol{\Sigma}^-\right]_{ii} = 1/\Sigma_{ii}$ if $\Sigma_{ii} \neq 0$). Whether Eq. 18 or Eq. 19 are used, the same matrix should result. The primary benefit of using Eq. 19 is that the locally non-redundant *active space* for the current geometry (the $n_{\text{act}} = n_{crt} - 6$ orthogonal linear combinations of the redundant internal coordinates in which $\mathbf{B}$ is defined for a given geometry) is computed as a consequence of doing the SVD. The active space is defined as the first n_{act} rows of $\mathbf{U}$.

With $\mathbf{B}^-$ defined, the typical approach for transformation of the gradient and Hessian begins with an algebraic rearrangement of the chain rule definition of the gradient (using x and q subscripts to differentiate between Cartesian and redundant internal coordinate gradients, respectively)

$$\mathbf{g}_x = \mathbf{B}^T\mathbf{g}_q \rightarrow \mathbf{g}_q = \left(\mathbf{B}^-\right)^T\mathbf{g}_x \tag{20}$$

A similar approach is used to transform the Hessian

$$\mathbf{H}_q = \left(\mathbf{B}^-\right)^T \left(\mathbf{H}_x - \mathbf{g}_q^T \frac{\partial \mathbf{B}}{\partial \mathbf{x}}\right) \left(\mathbf{B}^-\right) \tag{21}$$

Projection into non-redundant space

When using a redundant coordinate system, the total number of coordinates being optimized exceeds the number of internal degrees of freedom, and this must be accounted for when computing a step. Constraints or a penalty function approach could be used to compute a valid step in the full set of redundant coordinates, but here a projection method is used. The active space described in the previous section may be used to project the gradient $\mathbf{g}_r = \mathbf{U}_{\text{act}}\mathbf{g}_q$ and Hessian $\mathbf{H}_r = \mathbf{U}_{\text{act}}\mathbf{H}_q\mathbf{U}_{\text{act}}^T$ into a reduced space. The step, $\mathbf{\Delta q}_r$, is computed in this space and then transformed back to the full $\mathbf{\Delta q}_0 = \mathbf{U}_{\text{act}}^T\mathbf{\Delta q}_r$.

Back-transformation of step

Due to the curvilinear relationship between the redundant internal and Cartesian coordinates, the back-transformation has to be done iteratively. This is done by minimizing the difference between the current geometry $\mathbf{q}(\mathbf{x})$ and a goal geometry, in redundant internal coordinates $\mathbf{q}_g = \mathbf{q}_0 + \mathbf{\Delta q}_0$, where $\mathbf{q}_0$ is the internal coordinate geometry at the current iteration, and $\mathbf{\Delta q}_0$ is the computed step. This can be thought of as the minimization of the following functional

$$\begin{aligned} F(\mathbf{x} + \mathbf{\Delta x}) =& \frac{1}{2}(\mathbf{\Delta q})^2 \\ =& \frac{1}{2}\left(\mathbf{q}_g - \mathbf{q}(\mathbf{x} + \mathbf{\Delta x})\right)^2 \end{aligned} \tag{22}$$

with respect to $\mathbf{\Delta x}$, starting with $\mathbf{x} = \mathbf{x}_0$ where $\mathbf{x}_0$ is Cartesian geometry at the current step. This gives the typical iterate for $\mathbf{\Delta x}$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{B}_i^- \mathbf{\Delta q}_i \tag{23}$$

This process continues until the RMS of $\mathbf{\Delta x}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$ is $<10^{-6}$ bohr.

Linear bends

Definition of terms

There are various ways to handle linear bends. In the present work, a dummy atom is used to define the two degrees of freedom associated with a linear bend. The Cartesian coordinates, bond lengths, bend angle, and torsion angle are denoted by x_i, $r_{ij} > 0$, $0 \leq \theta_{ijk} \leq \pi$ and $-\pi \leq \phi_{ijkl} \leq \pi$ respectively. The atoms involved in the linear angle are indicated by numbers 1 through 3 (i.e., the angle which is nearly/completely linear is θ_{123}), the dummy atom is indicated by the subscript d, and n and m are used to indicate any atoms bonded to 1 and 3, respectively.

Placement of the dummy atom

If θ_{123} is not completely linear, then the plane in which the angle bends may be defined by r_{12} and r_{23}, while if it is linear, the choice of bend direction is arbitrary. For consistency and convenience, the bend direction may be chosen such that either ϕ_{n12d} or ϕ_{d23m} is 0 for some n or m. Once the plane is defined, the dummy atom is placed such that $\theta_{12d} = \theta_{32d}$ and r_{2d} is equal to a constant value. In the present implementation, this is set to 2 bohr since that is a fairly typical bond length, but the choice is more or less arbitrary.

Definition of coordinate set to handle linear bends

With the dummy atom placed, the optimization is carried out by modifying the coordinate definitions and constraining three coordinates per linear bend to define the location of the dummy atom relative to the rest of the molecule. The standard approach is used to construct the coordinate set beginning with all of the stretches corresponding to the bonded atoms, and angle bends between pairs of stretches that share a common atom. The angle θ_{123} is redefined as the equivalent sum $\theta_{12d} + \theta_{32d}$. The dihedral angles are included for all pairs of angle bend coordinates (not including θ_{123}) that share a bond, including all of the dihedrals ϕ_{n12d} and ϕ_{d23m}, and the improper dihedral ϕ_{12d3}.

The constrained coordinates always include r_{2d}, $\theta_{12d} - \theta_{32d}$, and a dihedral, but the choice of dihedral depends on the structure. For the vast majority of cases, the choice of any ϕ_{n12d} or ϕ_{d23m} is appropriate, with special care taken when there are multiple adjacent linear bends (i.e., 4 or more bonded, colinear atoms) to ensure that one dihedral for each dummy atom is included. If the molecule contains only 3 atoms, the improper dihedral ϕ_{12d3} can be constrained as the bend direction is completely arbitrary for a potential that only depends on the internal coordinates.

Coordinate transformation/back-transformation

Let $\mathbf{B}_{\text{opt}}$ be the B-matrix for the redundant internal coordinate set defined in the previous section, and $\mathbf{B}_{\text{constr}}$ be the B-matrix for the set of coordinates that will be used to constrain the motion of the dummy atom. The transformation is carried out using an otherwise identical approach to the standard unconstrained transformation, using a projected B-matrix $\mathbf{B}_{prj} = \mathbf{B}_{\text{opt}} - \mathbf{B}_{\text{opt}}\mathbf{B}_{\text{constr}}^-\mathbf{B}_{\text{constr}}$, and assuming that the gradient and Hessian elements corresponding to the dummy atoms are all 0. $\mathbf{B}_{prj}$ has the correct number of non-singular values ($3 * n_{\text{atoms}}$) for the original system. If analytical Hessians are used, the B-matrix derivatives are similarly projected: for internal coordinate i, $\partial\left[\mathbf{B}_{prj}\right]_i = \left(\mathbf{I} - \mathbf{B}_{cns}^{-}\mathbf{B}_{cns}\right)\partial\left[\mathbf{B}_{\text{opt}}\right]_i\left(\mathbf{I} - \mathbf{B}_{cns}^{-}\mathbf{B}_{cns}\right)$.

Once a step is computed, the back-transformation is done in an unconstrained fashion, with the goal values for the constrained coordinates set according to their current values. A second iterative back-transformation is used, if necessary, to reimpose the dihedral constraint since there may be some redundancy between the dihedral constraint chosen and dihedral coordinates used in the optimization. During the second iterative back-transformation, only the dummy atom is allowed to move.

Step size and direction control

RFO

The rational function optimization [26] (RFO) method begins by approximating the PES as a rational function:

$$E_{RF}(\mathbf{q}_0 + \mathbf{\Delta q}) = E_0 + \frac{\mathbf{\Delta q}^T\mathbf{g}_0 + \frac{1}{2}\mathbf{\Delta q}^T\mathbf{H}_0\mathbf{\Delta q}}{1 + \mathbf{\Delta q}^T\mathbf{S}\mathbf{\Delta q}} \tag{24}$$

with the matrix $\mathbf{S}$ taken to be the identity matrix for convenience. Equation 24 has the property that, for a nonzero gradient, there exactly $n + 1$ stationary points on the surface, where n is the number of coordinates defining the surface, with each stationary point having a different curvature (determined by the number of negative eigenvalues in the second derivative of E_{RF}). By stepping toward the stationary point on the rational function surface with the desired curvature, an optimization may be directed to a particular solution even when the number of negative eigenvalues in $\mathbf{H}_0$ is incorrect. The stationary condition (the value of $\mathbf{\Delta q}$ where $\partial E_{RF}/\partial\mathbf{\Delta q}$ is zero) for all $n + 1$ stationary points is given by diagonalizing the augmented Hessian

$$\mathbf{H}_{\text{aug}} = \begin{bmatrix} \mathbf{H} & \mathbf{g} \\ \mathbf{g}^T & 0 \end{bmatrix} \tag{25}$$

The step may be computed using either the mth eigenvector ($\mathbf{v}_m$) or eigenvalue (λ_m) of the augmented Hessian.

$$\mathbf{\Delta q}_{RFO,m} = \frac{\left(v_{m,1}, v_{m,2}, \ldots, v_{m,n}\right)}{v_{m,n+1}} \tag{26}$$

$$\mathbf{\Delta q}_{RFO,m} = -(\mathbf{H} - \lambda_m\mathbf{I})^{-1}\mathbf{g} \tag{27}$$

Both approaches result in the same displacement, and the resulting step satisfies the following relationship

$$\lambda_m = \left(\mathbf{\Delta q}_{RFO,m}\right)^T\mathbf{g} \tag{28}$$

For minimizations, the most negative eigenvalue of the augmented Hessian is used. For transition states, even though the second most negative eigenvalue results in a shifted Hessian with a single negative eigenvalue, the partitioned RFO (pRFO [26]) approach tends to result in a more stable optimization behavior. The pRFO approach involves searching for a minimum in $n - 1$ directions (usually selected to be eigenvectors of the Hessian), and a maximum in the remaining direction. This is done by combining an RFO minimization step in the $n - 1$ space, and an RFO maximization step in the remaining space.

The RFO method provides a means to compute a step in the correct direction for an optimization even when the Hessian has the incorrect number of negative eigenvalues (0 for minima, 1 for transition states). Often times, however, when the geometry is far from the solution, and especially when the Hessian data are approximate/updated, the computed step size may be unreasonably large and needs to be scaled back to avoid over-shooting the solution. For minimizations, an adaptive step size approach is frequently used and will be included in the standard method (see Ref. [27] for a more indepth discussion of adaptive step size methods). The adaptive step size, or trust region, is initialized to a modest value (0.5 bohr or rad for the present work), and at each step of the optimization, the predicted quadratic change in energy ($\mathbf{\Delta}E_{\text{quad}} = \mathbf{\Delta q}^T\mathbf{g} + 0.5\mathbf{\Delta q}^T\mathbf{H}\mathbf{\Delta q}$) is compared to the actual change in the energy. When the ratio of $\mathbf{\Delta}E_{\text{quad}}$ and the actual change in energy is >75 % and the length of the step is at least 80 % of the current trust region, the trust region is doubled. If the ratio of $\mathbf{\Delta}E_{\text{quad}}$ and the actual change in energy is <25 %, the trust region is reduced to a quarter of the length of the previous step.

Hessian update

To facilitate the discussion of different Hessian update formulas, the following standard notations for the displacement $\mathbf{s} = \mathbf{\Delta q}$, change in gradient $\mathbf{y} = \mathbf{\Delta g}_q$, and quadratic error $\mathbf{z} = \mathbf{y} - \mathbf{Hs}$ is used.

$$\mathbf{\Delta H}_{BFGS} = \frac{\mathbf{yy}^T}{\mathbf{y}^T\mathbf{s}} - \frac{\mathbf{sHHs}^T}{\mathbf{sHs}} \tag{29}$$

$$\mathbf{\Delta H}_{SR1} = \frac{\mathbf{zz}^T}{\mathbf{z}^T\mathbf{s}} \tag{30}$$

$$\mathbf{\Delta H}_{PSB} = \frac{\mathbf{sz}^T + \mathbf{zs}^T}{\mathbf{s}^T\mathbf{s}} - \mathbf{s}^T\mathbf{z}\frac{\mathbf{ss}^T}{\left(\mathbf{s}^T\mathbf{s}\right)^2} \tag{31}$$

$$\mathbf{\Delta H}_{MSP} = \phi_{MSP}\mathbf{\Delta H}_{SR1} + (1-\phi_{MSP})\mathbf{\Delta H}_{PSB} \tag{32}$$

$$\phi_{MSP} = \frac{\left(\mathbf{s}^T\mathbf{z}\right)^2}{\left(\mathbf{z}^T\mathbf{z}\right)\left(\mathbf{s}^T\mathbf{s}\right)} \tag{33}$$

When attempting to locate a minimum energy structure, the BFGS update (Eq. 29) is widely considered to be the gold standard as it is generally quite stable, so long as $\mathbf{s}^T\mathbf{y}$ is positive and the current approximation to the Hessian is positive definite. For transition-state optimizations, which require the Hessian to be indefinite with one negative eigenvalue, there is not such a clear cut winner. The SR1 method (Eq. 30, also referred to as the MS or Murtagh–Sargeant update) tends to produce the most reasonable updates, particularly when the change in the Hessian is large (for example, when the approximation is poor or the change in the gradient is large), but it can suffer from stability issues when $\mathbf{s}^T\mathbf{z}$ is small relative to $\mathbf{z}^T\mathbf{z}$. The PSB method (Eq. 31), on the other hand, is very stable, but the updates tend to be significantly poorer as they increase in magnitude. A combination of these two updates, the MSP method (Eq. 32), was proposed by Bofill [14]. This combined update attempts to leverage the pros and cons of the SR1 and PSB methods by blending them together according to the overlap of the quadratic error and the displacement (Eq. 33).

Convergence

An optimization is considered converged when the RMS change in $\mathbf{\Delta q}$ is less than 1.2×10^{-3} bohr or radians, the maximum absolute element in $\mathbf{\Delta q}$ is less than 1.8×10^{-3} bohr or radians, the RMS of $\mathbf{g}_r$ is less than 1.5×10^{-4} hartree/(bohr or rad), and the maximum absolute element in $\mathbf{g}_r$ is less than 4.5×10^{-4} hartree/(bohr or rad). These convergence criteria are equivalent to those used by Gaussian 09 [21].

References

1. Schlegel HB (2011) WIREs Comput Mol Sci 1(5):790
2. Fogarasi G, Zhou X, Taylor PW, Pulay P (1992) J Am Chem Soc 114(21):8191
3. Pulay P, Fogarasi G (1992) J Chem Phys 96(4):2856
4. Hestenes MR, Stiefel E (1952) Res Nat Bur Stand 49(6):409
5. Nocedal J (1980) Math Comput 35(151):773
6. Chill ST, Stevenson J, Ruehle V, Shang C, Xiao P, Farrell JD, Wales DJ, Henkelman G (2014) J Chem Theory Comput 10(12):5476
7. Schlegel HB (1984) Theor Chim Acta 66(5):333
8. Broyden CG (1970) Math Comput 24(110):365
9. Fletcher R (1970) Comput J 13(3):317
10. Goldfarb D (1970) Math Comput 24(109):23
11. Shanno DF (1970) Math Comput 24(111):647
12. Powell M (1971) IMA J Appl Math 7(1):21
13. Murtagh BA (1970) Comput J 13(2):185
14. Bofill JM (1994) J Comput Chem 15(1):1
15. Császár P, Pulay P (1984) J Mol Struct 114:31
16. Peng C, Ayala PY, Schlegel HB, Frisch MJ (1996) J Comput Chem 17(1):49
17. Baker J, Chan F (1996) J Comput Chem 17(7):888
18. Greenstadt J (1970) Math Comput 24(109):1
19. Löwdin PO (1950) J Chem Phys 18(3):365
20. Wolfram Research, Inc (2013) Mathematica, Version 9.0. Champaign, IL
21. Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, Cheeseman JR, Scalmani G, Barone V, Mennucci B, Petersson GA, Nakatsuji H, Caricato M, Li X, Hratchian HP, Izmaylov AF, Bloino J, Zheng G, Sonnenberg JL, Hada M, Ehara M, Toyota K, Fukuda R, Hasegawa J, Ishida M, Nakajima T, Honda Y, Kitao O, Nakai H, Vreven T, Montgomery JA Jr, Peralta JE, Ogliaro F, Bearpark M, Heyd JJ, Brothers E, Kudin KN, Staroverov VN, Kobayashi R, Normand J , Raghavachari K, Rendell A, Burant JC, Iyengar SS, Tomasi J, Cossi M, Rega N, Millam JM, Klene M, Knox JE, Cross JB, Bakken V, Adamo C, Jaramillo J, Gomperts R, Stratmann RE, Yazyev O, Austin AJ, Cammi R, Pomelli C, Ochterski JW, Martin RL, Morokuma K, Zakrzewski VG, Voth GA, Salvador P, Dannenberg JJ, Dapprich S, Daniels AD, Farkas, Foresman JB, Ortiz JV, Cioslowski J, Fox DJ (2009) Gaussian 09 revision d.01. Gaussian Inc. Wallingford CT
22. Vosko S, Wilk L, Nusair M (1980) Can J Phys 58(8):1200
23. Lee C, Yang W, Parr RG (1988) Phys Rev B 37(2):785
24. Becke AD (1993) J Chem Phys 98(7):5648
25. Stephens P, Devlin F, Chabalowski C, Frisch MJ (1994) J Phys Chem 98(45):11623
26. Banerjee A, Adams N, Simons J, Shepard R (1985) J Phys Chem 89(1):52
27. Dennis JE Jr, Schnabel RB (1983) Numerical methods for unconstrained optimization and nonlinear equations. Prentice-Hall, Englewood Cliffs, NJ