



Geometry optimization methods for modeling large molecules

Ödön Farkas^{a,*}, H. Bernhard Schlegel^b

^aDepartment of Organic Chemistry, Eötvös Loránd University, Budapest, Hungary

^bDepartment of Chemistry, Wayne State University, Detroit, USA

Abstract

Geometry optimization is an essential part of quantum chemical applications. The diversity of the scaling of different methods from linear to exponential implies that there are different requirements for a chosen optimization method. The proposed method aims to meet two requirements, good scaling with size and reliability, which would be a good match for redundant internal coordinate system-based optimization techniques with linear scaling coordinate transformation. The new optimization algorithm uses screened Cholesky decomposition for coordinate transformations and an iterative subspace optimization method. The iterative subspace appears in the course of any optimization. However, few methods are available for using such information efficiently. The Geometry Optimization using Direct Inversion in the Iterative Subspace method is known to have good scaling and efficiency, but poor reliability. Building a Hessian-like matrix in the iterative subspace allows one to take advantage of the reliability offered by Rational Function Optimization, Eigenvector Following and Trust Radius Method (TRM), but still avoid a consequent computational penalty. Also, the new approach steps away from the regular quadratic approximation related to the Newton methods by assuming a simple linear connection between gradient and coordinate changes.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Geometry optimization using direct inversion in the iterative subspace; Rational function optimization; Trust radius method; Sparse Cholesky decomposition

1. Introduction

Quantum chemical geometry optimization methods evolved rapidly over the last three decades. A major developmental milestone included the analytic gradients of the potential energy and the methods based on them, such as the quasi-Newton methods and their modifications. Hessian update techniques allowed information to be collected for the potential energy surface (PES), which accelerated the optimization process. The quadratic line search (QLS) [1–5], rational function optimization

(RFO) [6], trust radius model (TRM) [1–5,7,8] and trust radius update [1–5,7,8] made such methods more reliable. Originally, the geometry optimization was performed in Cartesian or in Z-matrix internal coordinates. Redundant internal coordinates [9,10] took precedence only in the last decade of the 20th century. Other improvements, such as the Geometry Optimization using Direct Inversion in the Iterative Subspace (GDIIS) [11,12] or using natural [13] or delocalized [14,15] internal coordinates has also been considered. It is now commonly agreed that an efficient optimization method [7,8] for quantum chemical applications should use the Hessian update, line search or GDIIS, and RFO in the framework of redundant internal coordinates. The general use of

* Corresponding author.

E-mail address: farkas@para.chem.elte.hu (Ö. Farkas).

such optimization techniques is, however, prevented by the quadratic ($O(N^2)$) scaling of their memory usage and the regularly cubic, $O(N^3)$, scaling of their computational demand with the number of variables to optimize. Current advances in the scaling of some frequently used quantum chemical methods, like Hartree–Fock, density functional theory or semi-empiricals, necessitate the development of new optimization techniques to match the required scaling with size, without compromising traditional efficiency and reliability. Our previous studies [16,17] revealed that the overall computational bottleneck can be reduced to an asymptotic quadratic, $O(N^2)$, scaling using an updated inverse technique for solving the systems of linear equations in question for both, the coordinate transformations and the RFO or TRM optimization step. The updated inverse technique provides similar efficiency and reliability as regular quantum chemical optimization methods but it still needs large, $O(N^2)$, storage of full matrices in memory. Other efforts have been focused on achieving an overall linear, $O(N)$, scaling for the coordinate transformations [18–21] to make the use of redundant internal coordinates affordable for much larger systems. The present paper describes an alternative way, a linear scaling equivalent of singular value decomposition (SVD) or generalized inverse, for solving linear equations with non-definite sparse matrixes for the coordinate transformations necessary in redundant internal coordinate-based optimizations. We also outline a new optimization technique similar to GDIIS, which can employ a step size control in the spirit of RFO and TRM through a generalized Hessian built in the iterative subspace.

The regular optimization process, such as the ‘Benny’ optimization algorithm of the GAUSSIAN [22] program contains two practically equivalent computational bottlenecks; one is the transformation of the forces into internal coordinates and then the transformation of the optimization step back to Cartesian coordinates. The coordinate transformations are based on the use of the Wilson B -matrix [23], which collects the partial derivatives of the internal coordinates with respect to Cartesians

$$B_{i,j} = \frac{\partial q_i}{\partial x_j} \Rightarrow d\mathbf{q} = \mathbf{B} d\mathbf{x} \text{ and } \mathbf{f}_x = \mathbf{B}^t \mathbf{f}_q \quad (1)$$

where q and x denote internal and Cartesian coordinates, respectively. The optimization process needs the transformation of forces (\mathbf{f}_x) given in Cartesian coordinates, and also the transformation of the internal coordinate step ($\Delta\mathbf{q}$). Due to the curvilinear nature of the internal coordinates, the finite internal coordinate step can be transformed by a few iterations of solving the corresponding equations ($\Delta\mathbf{q} \approx \mathbf{B}\Delta\mathbf{x}$). For detailed remarks on solving such equations, see Appendix A. In the case of constrained optimizations, an extra projection of the forces and the optimization step is required.

The other bottleneck arises from the computation of the RFO or TRM step, which is in fact a Newton–Raphson step with a modified, shifted Hessian matrix

$$-\Delta\mathbf{f} = (\mathbf{H} + \lambda\mathbf{I})\Delta\mathbf{x} \quad (2)$$

where \mathbf{H} is an approximate Hessian (force constant matrix) in the case of quasi-Newton methods. The appropriate diagonal shift to the Hessian is defined by the method of choice (RFO or TRM), and is usually carried out by solving Eq. (2) for $\Delta\mathbf{x}$ each time during the course of determining λ iteratively. A practical way of solving Eq. (2) implies the diagonalization of the Hessian, which is a cubically scaling computational bottleneck.

2. Method

2.1. Coordinate transformations

Paizs et al. pointed out that any set of redundant internal coordinates could be constructed from their complete, but non-redundant, subsets as linear combinations [20]. They also concluded that this also applies to matrices $\mathbf{B}^t\mathbf{B}$ and $\mathbf{B}\mathbf{B}^t$, and their rows. The full Cholesky factorization of positive semi-definite matrices results in zero diagonal values. Because of consequent divisions by zero, the full Cholesky factorization (regardless of the sparsity of the matrix in question) can only be applied to positive definite matrices. The zero (or in practice very small) diagonal values, however, indicate rows that can be produced as a linear combination of previously processed rows. Removing such rows from further examination results in the screened Cholesky factorization. This creates a non-redundant set of linear

combinations. In practice, we use a small positive threshold value, such as 2×10^{-5} , for diagonal elements to select the rows for removal. We also noted that the rank (number of non-zero diagonal values) of the decomposition might change by varying the threshold value. With proper ordering, the screened Cholesky factors can reproduce the original symmetric matrix accurately, although in general, the multiplication of a vector by the ‘screened Cholesky inverse’ of the matrix via forward–backward substitution using the screened Cholesky factors does not give the same results as multiplying by the generalized inverse. Nevertheless, linear equations of the form $\mathbf{y} = \mathbf{A}\mathbf{x}$ can be solved using the screened Cholesky factors of positive semi-definite matrices $\mathbf{A}\mathbf{A}^t$ and $\mathbf{A}^t\mathbf{A}$. The screened Cholesky factorization-based equation solving method can be directly applied to the equations arising from the coordinate transformations (see Eq. (1)). The screened Cholesky decomposition has been tested in practice and give identical results identical to the regular generalized inverse-based transformations; however, its thorough mathematical proof should be given later. For details and the constrained optimization formulae, see Appendix B. The main advantage of the screened Cholesky factorization over the approximate or shifted Cholesky decomposition is that it can substantially reduce the required storage and computational demand for highly redundant systems.

2.2. Iterative subspace optimization

The overall efficiency of the optimization method also depends on the choiced algorithm of choice. Quasi-Newton related algorithms could greatly benefit from using redundant internal coordinates, since a good approximation to the PES can be used for a much wider range than in Cartesian coordinates. The problem is that the approximation to the PES regularly stored in a full Hessian matrix results in at least an $O(N^2)$ scaling computational bottleneck. A starting guess Hessian matrix can be stored in sparse, or even diagonal form and limited memory Hessian update techniques have been formulated to achieve linear scaling [24–27]. These techniques proved to be more efficient than available alternatives, such as pure conjugate gradient (CG)-based methods. Limited memory update techniques, have some disadvantages,

which include the lack of convergence acceleration via RFO/TRM type shifting to the Hessian. In addition, they are not advised to use for transition state optimizations. The GDIIS method is also a possible choice to consider, since its memory and computational demand can be easily controlled by the number of vectors stored and used in the iterative subspace. The reliability of the GDIIS method is unfortunately very poor, especially for large-scale non-linear problems, such as for the geometry optimization of biomolecules. Our previously described method for controlling GDIIS [12] appears to be satisfactory, but requires an expensive computation for a reference step, which is currently RFO. It is, however, an instructive task to find the reason for the otherwise surprisingly good performance of the GDIIS technique closed to convergence. The first thing to note is that GDIIS assumes a simple linear connection between coordinate and force (or gradient) changes. This condition seems to be similar to the quadratic approximation for the PES. The quadratic approximation, however, presumes that the linear connection is related to the second derivative, Hessian matrix of the PES, in the form of a symmetric real matrix:

$$-\Delta\mathbf{f} = \mathbf{H}\Delta\mathbf{x} \quad (3)$$

The GDIIS method only assumes that the force change related to any linear combination of the collected optimization steps (coordinate changes) can be formed using the same combination of the corresponding force changes:

$$\sum \gamma_i \Delta\mathbf{x}_i = \Delta\mathbf{x} \Leftrightarrow \sum \gamma_i \Delta\mathbf{f}_i = \Delta\mathbf{f} \quad (4)$$

In fact, Eq. (4) states a more generic linear connection between coordinate and force changes, which is valid for the whole set of stored coordinate and force changes when no redundancy occurs:

$$-\mathbf{F} = \mathbf{H}_G\mathbf{X} \quad (5)$$

Matrices \mathbf{F} and \mathbf{X} collect the coordinate and force changes, respectively. Matrix \mathbf{H}_G , expressing the linear connection, is the generalized Hessian that is not necessarily symmetric, not even for quadratic PESs. For the purpose of optimization, it may serve a similar purpose as the Hessian in the quasi-Newton methods. Based on Eq. (5), it is also possible to find a suitable way of obtaining \mathbf{H}_G .

Orthonormalizing the column vectors of matrix \mathbf{X} via Schmidt orthogonalization results in a unitary matrix on the right side that leads us to the expression for obtaining \mathbf{H}_G :

$$-\mathbf{FV} = \mathbf{H}_G\mathbf{XV} = \mathbf{H}_G\mathbf{U} \Rightarrow -\mathbf{FVU}^t = \mathbf{H}_G \quad (6)$$

where matrix \mathbf{V} orthonormalizes \mathbf{X} . Matrix \mathbf{VU}^t is the generalized inverse of \mathbf{X} if no redundancy occurs. The purpose of using Schmidt orthogonalization instead of a generalized inverse provides higher priority for the latest coordinate and force changes. \mathbf{H}_G is usually not symmetric; therefore, its direct diagonalization is not feasible. The tools of SVD can provide the singular values and two sets of eigenvectors:

$$\mathbf{H}_G = \mathbf{LAR}^t \quad (7)$$

The singular values are positive numbers by definition; however, if we change the sign of one of the eigenvector pairs and the corresponding singular value in order to constrain the scalar product of the corresponding eigenvectors to be non-negative

$$\mathbf{l}_i^t \mathbf{r}_i \geq 0 \quad (8)$$

they may gain signs. The resulting diagonal values of \mathbf{A} can serve as pseudo-eigenvalues, allowing RFO/TRM type step size control of the optimization step. It is important to note that the latest force usually cannot be represented in the subspace of collected force changes. Therefore, only a part of the optimization step can be computed using \mathbf{H}_G . The resulting step corresponds to minimizing the force in the iterative subspace, such as in GDIIS. The residual force then can be used to compute the residual optimization step. Examination of the difference between GDIIS and a line search reveals that since the residual force is perpendicular to the subspace of force (or error vector) changes, the quadratic line search finds an energy extreme and its residual force is perpendicular to the latest step, or if it is generalized to higher dimensions to the subspace of coordinate changes, while GDIIS ends to minimize the force (or an arbitrary error vector). This difference suggests that a projected generalized Hessian for the iterative subspace should be constructed, to serve the purpose of multidimensional search (MDS) using a generalized quadratic line search to higher dimensions:

$$-\mathbf{UU}^t\mathbf{FVU}^t = \mathbf{H}_{\text{MDS}} \quad (9)$$

The advantage of using the projection of the force changes into the coordinate change subspace is that the left and right side eigenvectors of \mathbf{H}_{MDS} span the same subspace. In addition, the generalized Hessian for MDS can more sensitively detect the proximity of higher order critical points with negative pseudo-eigenvalues. RFO/TRM style step size controls can also be applied to \mathbf{H}_{MDS} , but unfortunately, these kinds of corrections are not capable of ensuring an energy lowering step direction, since \mathbf{H}_{MDS} is generally non-symmetric for more than one dimension. The construction of an efficient, but symmetric Hessian for the iterative subspace is a key target for further studies in that field.

The presented iterative subspace optimization (ISSO) scheme produces dense Hessian matrixes in the full space, resulting in the same computational demand than the regular $O(N^3)$ scaling techniques. The rank of these matrixes, however, is not larger than the number of coordinate changes used. The common subspace of coordinate and force changes can be constructed reducing the required storage and computational cost significantly. If the maximum number of used vectors remains constant, then the ISSO method requires linearly scaling storage and computational effort. The details about the construction of the iterative subspace efficiently can be found in Appendix C.

3. Results and discussion

The new methods presented here have been implemented in the development version of Gaussian [28]. The preliminary test results on storage and CPU requirements of the coordinate transformations (namely the computation of the Cholesky factors) can be found in Figs. 1 and 2. Proper reordering is essential for the efficiency of the factorization. Thus, a divide-and-conquer-based approach will be developed for this purpose, as suggested by Nemeth et al. [21]. Note that the factorization of $\mathbf{G}_q = \mathbf{BB}^t$ is less demanding than the factorization of $\mathbf{G}_x = \mathbf{B}^t\mathbf{B}$ in the screened Cholesky formalism. It is not easy to find examples to compare the performance of different optimization algorithms on large flexible molecules because they tend to converge to different local minima. However, the optimization of taxol

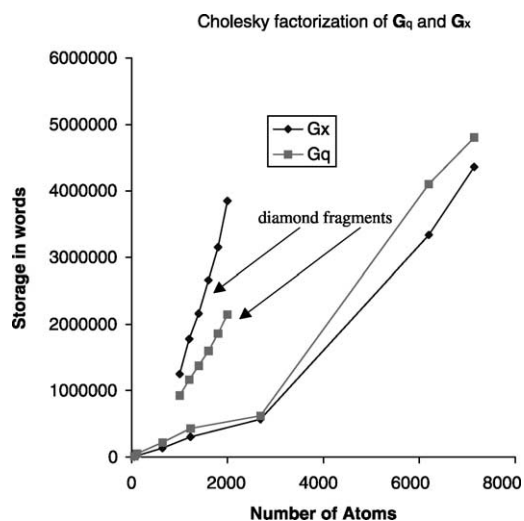


Fig. 1. Diagram for required storage (including indexing) vs. number of atoms. The diamond fragments (3D network examples) are clearly separated from the others, which are mostly proteins. For the definition of G_q and G_x , see Appendix B.

(113 atoms) in the UFF [29] forcefield results the same minimum for the three selected optimization methods used for large molecules. The results are summarized in Table 1. The linear scaling ISSO method uses only a diagonal updated guess Hessian

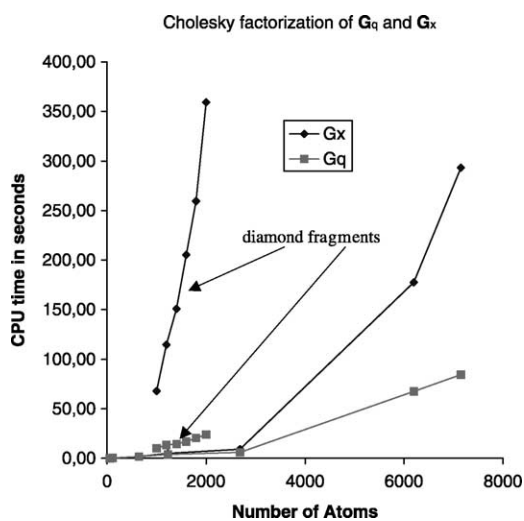


Fig. 2. Diagram for required CPU time on 500 MHz Alpha processor vs. number of atoms. The diamond fragments (3D network examples) are clearly separated from the others, which are mostly proteins. For the definition of G_q and G_x , see Appendix B.

Table 1

The number of optimization steps required optimizing taxol in the UFF [29] molecular mechanics forcefield

Optimization method	Number of optimization steps
Controlled GDIIS, full Hessian [12]	40
ISSO, diagonal full space Hessian	127
CG, Cartesian coordinates	1153

for the full space and therefore, needs substantially more steps to converge than the controlled GDIIS. However, it is still only a small fraction of the steps used by the CG optimizer in Cartesian coordinates. Further improvements are expected using a sparse updated Hessian in the full space and implementing a symmetric MDS Hessian for guiding the optimization to the quadratic region of a proper critical point more efficiently.

4. Summary

The aim of the new methods presented is to provide efficient linear scaling optimization tools for the emerging linear scaling quantum chemical methods, including QM/MM and MM calculations. The use of redundant internal coordinates allows more accurate calculations for large molecules with the aid of ISSO and MDS, which also opens the opportunity for the development of transition state optimization algorithms for large molecules. In general, the screened Cholesky decomposition can be considered for solving large, redundant systems of linear equations.

Acknowledgements

This work has been supported by Gaussian, Inc. (<http://www.gaussian.com>) and the Hungarian Research Foundation (OTKA D29446). The authors would also like to thank NSF, MTA and OTKA for an international collaboration grant. The computations have been performed in part using the High Performance Computing Center of the Eötvös Loránd University.

Appendix A

A.1. Solving linear equations

A system of linear equations can be formulated as $\mathbf{y} = \mathbf{A}\mathbf{x}$, which can only be solved if the proper $\mathbf{P}\mathbf{y}$ projection does not alter \mathbf{y} :

$$(\mathbf{A}\mathbf{A}^t)(\mathbf{A}\mathbf{A}^t)^{-1}\mathbf{y} = \mathbf{P}\mathbf{y} = \mathbf{y}' = \mathbf{A}\mathbf{x} \quad (\text{A1})$$

Otherwise, the corresponding equations with \mathbf{y}' can be solved using the generalized inverse or SVD formalism:

$$(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{y} = \mathbf{A}^t(\mathbf{A}\mathbf{A}^t)^{-1}\mathbf{y} = \mathbf{x} \quad (\text{A2})$$

If matrix $\mathbf{A}^t\mathbf{A}$ or $\mathbf{A}\mathbf{A}^t$ is non-singular, then $\mathbf{A}\mathbf{A}^t = \mathbf{L}\mathbf{L}^t$ or $\mathbf{A}^t\mathbf{A} = \mathbf{K}\mathbf{K}^t$ Cholesky decomposition is the most efficient way of solving the equations:

$$(\mathbf{A}^t\mathbf{A})^{-C}\mathbf{A}^t\mathbf{y} = \mathbf{x} \text{ or } \mathbf{A}^t(\mathbf{A}\mathbf{A}^t)^{-C}\mathbf{y} = \mathbf{x} \quad (\text{A3})$$

The $-C$ superscript denotes the Cholesky inverse which is in practice multiplication by the inverse via forward–backward substitution using the corresponding lower/upper triangular Cholesky factors. If $\mathbf{A}^t\mathbf{A}$ and $\mathbf{A}\mathbf{A}^t$ are positive semi-definite, then incomplete, shifted or approximate Cholesky factorization and iterative solving is necessary. The removal of columns of the lower and the corresponding rows of the upper triangular factors with diagonals smaller than a small positive ε during the decomposition results in a non-redundant subset:

$$\mathbf{A}^t\mathbf{A} = \mathbf{L}_S\mathbf{L}_S^t \text{ and } \mathbf{A}\mathbf{A}^t = \mathbf{K}_S\mathbf{K}_S^t \quad (\text{A4})$$

The number of remaining non-zero diagonals indicates the rank of \mathbf{A} . We found that changing ε may change the obtained rank if the original limit is too large. The corresponding Cholesky inverse can be used as a projector

$$\mathbf{A}(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{y} = \mathbf{P}\mathbf{y} = \mathbf{A}(\mathbf{A}^t\mathbf{A})^{-S}\mathbf{A}^t\mathbf{y} = \mathbf{y}' \quad (\text{A5})$$

like the generalized inverse, but solving the equations cannot be done in the usual way, since:

$$(\mathbf{A}^t\mathbf{A})^{-S}\mathbf{A}^t\mathbf{y} \neq \mathbf{x}$$

$$\mathbf{A}^t(\mathbf{A}\mathbf{A}^t)^{-S}\mathbf{y} \neq \mathbf{x} \quad (\text{A6})$$

$$(\mathbf{A}^t\mathbf{A})^{-S}\mathbf{A}^t\mathbf{y}' \neq \mathbf{x}$$

On the other hand,

$$\mathbf{A}^t(\mathbf{A}\mathbf{A}^t)^{-S}\mathbf{y}' = \mathbf{x} \quad (\text{A7})$$

leads to the solution in the screened Cholesky formalism:

$$\mathbf{A}^t(\mathbf{A}\mathbf{A}^t)^{-S}\mathbf{A}(\mathbf{A}^t\mathbf{A})^{-S}\mathbf{A}^t\mathbf{y} = (\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t = \mathbf{x} \quad (\text{A8})$$

which is a potential linear scaling equivalent of the generalized inverse. The ‘screened Cholesky inverse’, denoted by the $-S$ superscript, is not provided explicitly but the multiplications with vectors can be computed via forward–backward substitutions, like in the case of regular Cholesky decomposition.

Appendix B

B.1. Coordinate transformations

Using the first order approximation to the connection between internal and Cartesian coordinates, one can carry out the coordinate transformations for the purpose of redundant internal coordinates

$$B_{ij} = \frac{\partial q_i}{\partial x_j} \Rightarrow d\mathbf{q} = \mathbf{B} d\mathbf{x} \text{ and } \mathbf{f}_x = \mathbf{B}^t\mathbf{f}_q$$

$$\mathbf{G}_q = \mathbf{B}\mathbf{B}^t(N_q \times N_q) \text{ and } \mathbf{G}_x = \mathbf{B}^t\mathbf{B}(N_x \times N_x) \quad (\text{A9})$$

$$\mathbf{P}_q^S = \mathbf{B}\mathbf{G}_x^{-S}\mathbf{B}^t \text{ and } \mathbf{P}_x^S = \mathbf{B}^t\mathbf{G}_q^{-S}\mathbf{B}$$

where \mathbf{B} is the Wilson B -matrix, q denotes internal, while x denotes Cartesian coordinates and the $-S$ subscript indicates screened Cholesky inverse. The equations for the internal forces, \mathbf{f}_q , and the Cartesian coordinate step, $\Delta\mathbf{x}$, can be solved using the screened Cholesky decomposition:

$$\mathbf{f}_q = \mathbf{B}\mathbf{G}_x^{-S}\mathbf{P}_x^S\mathbf{f}_x = \mathbf{G}_q^{-1}\mathbf{B}\mathbf{f}_x \quad (\text{A10})$$

$$d\mathbf{x} = \mathbf{B}^t\mathbf{G}_q^{-S}\mathbf{P}_q^S d\mathbf{q} = \mathbf{B}^t\mathbf{G}_q^{-1} d\mathbf{q}$$

The transformation of the optimization step is valid for infinitesimally small steps only. In practice, we iterate $\Delta\mathbf{x} = \mathbf{B}^t\mathbf{G}_q^{-S}\mathbf{P}_q^S\Delta\mathbf{q} = \mathbf{B}^t\mathbf{G}_q^{-1}\Delta\mathbf{q}$ until convergence. Constrained optimization necessitates the definition of the constrained B -matrix, which has empty rows for the non-constrained internal

coordinates, and then the related matrixes:

$$\mathbf{G}_{q,c} = \mathbf{B}_c \mathbf{B}_c^t \text{ and } \mathbf{G}_{x,c} = \mathbf{B}_c^t \mathbf{B}_c \quad (\text{A11})$$

$$\mathbf{P}_{q,c}^S = \mathbf{B}_c \mathbf{G}_{x,c}^{-S} \mathbf{B}_c^t \text{ and } \mathbf{P}_{x,c}^S = \mathbf{B}_c^t \mathbf{G}_{q,c}^{-S} \mathbf{B}_c$$

The force transformation is straightforward

$$\mathbf{f}_q = \mathbf{B} \mathbf{G}_x^{-S} (\mathbf{I} - \mathbf{P}_{x,c}^S) \mathbf{P}_x^S \mathbf{f}_x = \mathbf{G}_q^{-1} \mathbf{B} (\mathbf{I} - \mathbf{P}_{x,c}^S) \mathbf{f}_x \quad (\text{A12})$$

but the step transformation has to treat optional non-zero step in the constrained subspace

$$\Delta \mathbf{x} \approx \mathbf{B}^t \mathbf{G}_q^{-S} (\mathbf{P}_{q,c}^S \Delta \mathbf{q}_c + (\mathbf{I} - \mathbf{P}_{q,c}^S) \mathbf{P}_q^S \Delta \mathbf{q}) \quad (\text{A13})$$

which is a little bit more complicated and can be formulated in numerous equivalent ways.

Appendix C

C.1. Iterative subspace

Definition. \mathbf{W} collects M of L dimensional vectors, then a transformation matrix, \mathbf{V} ($N \times M$) is applied to achieve:

$$(\mathbf{WV})^T \mathbf{WV} = \mathbf{V}^T \mathbf{W}^T \mathbf{WV} = \mathbf{I} \quad (\text{A14})$$

The $\mathbf{Q} = \mathbf{WV}$ ($N \times L$) set of vectors forms an orthonormal basis to define a subspace. Any \mathbf{a} vector of that subspace (but represented in the original space) can then be represented as \mathbf{p} in the basis of the subspace coordinates (\mathbf{Q}) as:

$$\mathbf{p} = \mathbf{Q}^T \mathbf{a} \quad (\text{A15})$$

In general, Eq. (A15) provides the projection of vector \mathbf{a} on to the subspace. The number of dimensions for the original vectors (\mathbf{W}, \mathbf{a}) might be large. Therefore, their direct use might be impractical. The necessary values can be obtained using the $\mathbf{O} = \mathbf{W}^T \mathbf{W}$ overlap matrix instead. It is also practical to give any vector \mathbf{a} as a \mathbf{b} linear combination of the original \mathbf{W} vectors

$$\mathbf{a} = \mathbf{Wb} \quad (\text{A16})$$

then the \mathbf{p} representation can be formed as:

$$\mathbf{p}^T = \mathbf{b}^T \mathbf{OV} \quad (\text{A17})$$

Furthermore, all work with the represented vectors in the subspace can be done using only N dimensional vectors and accordingly formed

matrixes with a maximum size of $N \times N$. The resulting vectors in the subspace, e.g. \mathbf{y} , then can be transformed back to the original space, denoted as \mathbf{z}

$$\mathbf{z} = \mathbf{Qy} \quad (\text{A18})$$

because as it is defined by Eq. (A15):

$$\mathbf{Q}^T \mathbf{z} = \mathbf{Q}^T \mathbf{Qy} = \mathbf{y} \quad (\text{A19})$$

For the purpose of geometry optimization, the \mathbf{W} set of column vectors contains the previously obtained K coordinate and force values as

$$\mathbf{w}_{2i-1} = \mathbf{x}_i, \quad \mathbf{w}_{2i} = \mathbf{f}_i, \quad i \in \{1, \dots, K\} \quad (\text{A20})$$

where the \mathbf{x} and \mathbf{f} vectors denote the coordinate vectors and the corresponding forces (negative of the gradient of the PES), respectively. Later, we assume that \mathbf{x}_1 and \mathbf{f}_1 belong to the latest point. The iterative subspace can be constructed based on difference vectors from the optimization history. As we noted before, it is practical to use linear combinations of the previously defined vector set (see Eq. (A16)), such that

$$\begin{aligned} \mathbf{x}_{j+1} - \mathbf{x}_j &= \mathbf{w}_{2(j+1)-1} - \mathbf{w}_{2j-1} = \mathbf{Wv}_{2j-1}^S, \\ \mathbf{f}_{j+1} - \mathbf{f}_j &= \mathbf{w}_{2(j+1)} - \mathbf{w}_{2j} = \mathbf{Wv}_{2j}^S, \end{aligned} \quad j \in \{1, \dots, K-1\} \quad (\text{A21})$$

for a ‘sequential’ iterative subspace based on optimization steps or alternatively, a ‘central’ iterative subspace can be built using differences from the latest point:

$$\begin{aligned} \mathbf{x}_{j+1} - \mathbf{x}_1 &= \mathbf{w}_{2(j+1)-1} - \mathbf{w}_1 = \mathbf{Wv}_{2j-1}^C, \\ \mathbf{f}_{j+1} - \mathbf{f}_1 &= \mathbf{w}_{2(j+1)} - \mathbf{w}_2 = \mathbf{Wv}_{2j}^C, \end{aligned} \quad j \in \{1, \dots, K-1\} \quad (\text{A22})$$

Any of the previously defined linear combination sets can then be used to construct an orthonormal basis via Gram–Schmidt orthogonalization, where the necessary scalar products are taken using the \mathbf{O} overlap matrix instead of direct use of the original \mathbf{W} set, as in the following example:

$$(\mathbf{x}_3 - \mathbf{x}_2)^T (\mathbf{x}_4 - \mathbf{x}_3) = (\mathbf{v}_3^S)^T \mathbf{W}^T \mathbf{Wv}_5^S = (\mathbf{v}_3^S)^T \mathbf{Ov}_5^S \quad (\text{A23})$$

The orthogonalization results in a new set of \mathbf{v} vectors, collected in \mathbf{V} , which satisfies Eq. (A14), and therefore, the corresponding \mathbf{Q} set of vectors

can serve as a basis for the iterative subspace. The iterative subspace is constructed in such a way that any \mathbf{a} vector needed to calculate an optimization step can be given as a \mathbf{b} linear combination of the \mathbf{W} set of vectors and its representation in the iterative subspace can be calculated using Eq. (A17). We note that, the \mathbf{p} representation (see Eq. (A15)) of a general vector, \mathbf{a} , in the iterative subspace only represent its projection and the *residuum* vector, \mathbf{r} , can be calculated as:

$$\mathbf{r} = \mathbf{a} - \mathbf{Qp} = (\mathbf{I} - \mathbf{Q}\mathbf{Q}^T)\mathbf{a} \quad (\text{A24})$$

If the *residuum* vector vanishes, then vector \mathbf{a} can be fully represented in the iterative subspace. In the case of geometry optimization, the iterative subspace Hessian matrix is applied only to the represented part of the actual force vector, while its *residuum* is used to calculate the remaining part of the optimization step calculated in the full space.

C.2. Hessian in the iterative subspace

Further simplifications can be applied for constructing the iterative subspace if we assume, according to GDIIS, a linear relationship between the changes in the coordinate values and the corresponding changes in the forces (or gradients). The optimization history contains stored geometries and corresponding forces. If the assumed linearity applies for them then at any \mathbf{x}' point

$$\mathbf{x}' = \mathbf{x}_1 + \sum_{i=1}^K \sum_{j=1}^{i-1} \alpha_{i,j} (\mathbf{x}_i - \mathbf{x}_j) \quad (\text{A25})$$

of the iterative subspace, the forces appear to be

$$\mathbf{f}' = \mathbf{f}_1 + \sum_{i=1}^K \sum_{j=1}^{i-1} \alpha_{i,j} (\mathbf{f}_i - \mathbf{f}_j) \quad (\text{A26})$$

We note, that the following rearrangement of Eqs. (A25) and (A26)

$$\mathbf{x}' = \sum_{i=1}^K \beta_i \mathbf{x}_i \quad (\text{A27})$$

$$\mathbf{f}' = \sum_{i=1}^K \beta_i \mathbf{f}_i$$

leads to the following

$$\sum_{i=1}^K \beta_i = 1 \quad (\text{A28})$$

constraint for the β coefficients, which is a simple result of the fact that the sum of the coefficients for linear combinations of difference vectors is always 0. The linearity condition can be stated in such a way that any linear combination of coordinate changes results in a force change that can be formed with the identical linear combination of the respective force changes. Also, a desired linear combination of force changes implies the same linear combination of coordinate changes, which is the basis of the GDIIS method.

It is also important to note that as a consequence of the assumed linearity, the following relationship holds between changes in coordinates and forces

$$-\Delta \mathbf{f} = \mathbf{H}_G \Delta \mathbf{x} \quad (\text{A29})$$

where the (otherwise needless) negative sign indicates the similarity with the quadratic approximation to the potential PES, while the G index means 'generic', stating that \mathbf{H}_G is not necessarily symmetric unlike the force constant (or Hessian or second derivative) matrix of the quasi-Newton methods. Further details are discussed in the text.

References

- [1] R. Fletcher, Practical Methods of Optimization, Wiley, Chichester, 1987.
- [2] P.E. Gill, W. Murray, M.H. Wright, Practical Optimization, Academic Press, London, 1981.
- [3] L.E. Scales, Introduction to Non-Linear Optimization, Springer, New York, 1985.
- [4] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [5] M.J.D. Powell (Eds.), Nonlinear Optimization, Academic Press, New York, 1982.
- [6] A. Banerjee, N. Adams, J. Simons, R. Shepard, Journal of Physical Chemistry 89 (1985) 52–57.
- [7] H.B. Schlegel, in: D.R. Yarkony (Ed.), Modern Electronic Structure Theory, World Scientific, Singapore, 1995, pp. 459–500.
- [8] H.B. Schlegel, in: P.v.R. Schleyer, N.L. Allinger, T. Clark, J. Gasteiger, P.A. Kollman, H.F. Schaefer III, P.R. Schreiner,

- (Eds.), *Encyclopedia of Computational Chemistry*, Wiley, Chichester, 1998, pp. 1136–1142.
- [9] P. Pulay, G. Fogarasi, *Journal of Chemical Physics* 96 (1992) 2856–2860.
- [10] C.Y. Peng, P.Y. Ayala, H.B. Schlegel, M.J. Frisch, *Journal of Computational Chemistry* 17 (1996) 49–56.
- [11] P. Csaszar, P. Pulay, *Journal of Molecular Structure* 114 (1984) 31–34.
- [12] Ö. Farkas, H.B. Schlegel, *Physical Chemistry Chemical Physics* 4 (2002) 11–15.
- [13] G. Fogarasi, X.F. Zhou, P.W. Taylor, P. Pulay, *Journal of the American Chemical Society* 114 (1992) 8191–8201.
- [14] J. Baker, A. Kessi, B. Delley, *Journal of Chemical Physics* 105 (1996) 192–212.
- [15] J. Baker, D. Kinghorn, P. Pulay, *Journal of Chemical Physics* 110 (1999) 4986–4991.
- [16] Ö. Farkas, H.B. Schlegel, *Journal of Chemical Physics* 109 (1998) 7100–7104.
- [17] Ö. Farkas, H.B. Schlegel, *Journal of Chemical Physics* 111 (1999) 10806–10814.
- [18] B. Paizs, G. Fogarasi, P. Pulay, *Journal of Chemical Physics* 109 (1998) 6571–6576.
- [19] K. Nemeth, O. Coulaud, G. Monard, J.G. Angyan, *Journal of Chemical Physics* 113 (2000) 5598–5603.
- [20] B. Paizs, J. Baker, S. Suhai, P. Pulay, *Journal of Chemical Physics* 113 (2000) 6566–6572.
- [21] K. Nemeth, O. Coulaud, G. Monard, J.G. Angyan, *Journal of Chemical Physics* 114 (2001) 9747–9753.
- [22] M.J. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, J.A. Montgomery, T. Vreven, K.N. Kudin, J.C. Burant, J.M. Millam, S.S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G.A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J.E. Knox, H.P. Hratchian, J.B. Cross, C. Adamo, J. Jaramillo, R. Gomperts, R.E. Stratmann, O. Yazyev, A.J. Austin, R. Cammi, C. Pomelli, J.W. Ochterski, P.Y. Ayala, K. Morokuma, G.A. Voth, P. Salvador, J.J. Dannenberg, V.G. Zakrzewski, S. Dapprich, A.D. Daniels, M.C. Strain, O. Farkas, D.K. Malick, A.D. Rabuck, K. Raghavachari, J.B. Foresman, J.V. Ortiz, Q. Cui, A.G. Baboul, S. Clifford, J. Cioslowski, B.B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R.L. Martin, D.J. Fox, T. Keith, M.A. Al-Laham, C.Y. Peng, A. Nanayakkara, M. Challacombe, P.M.W. Gill, B. Johnson, W. Chen, M.W. Wong, C. Gonzalez, J.A. Pople, *Gaussian Development Version, Revision B.02*, Gaussian, Inc., Pittsburgh, PA, 2003.
- [29] A.K. Rappe, C.J. Casewit, K.S. Colwell, W.A. Goddard, W.M. Skiff, *Journal of the American Chemical Society* 114 (1992) 10024–10035.