

# Geometry optimization speedup through a geodesic approach to internal coordinates

Eric D. Hermes,<sup>1</sup> Khachik Sargsyan,<sup>1</sup> Habib N. Najm,<sup>1</sup> and Judit Zádor<sup>1</sup>

*Combustion Research Facility, Sandia National Laboratories, Livermore,  
CA 94551-0969*

(\*Electronic mail: jzador@sandia.gov)

(Dated: 4 August 2021)

We present a new geodesic-based method for geometry optimization in a basis of redundant internal coordinates. Our method updates the molecular geometry by following the geodesic generated by a displacement vector on the internal coordinate manifold, which dramatically reduces the number of steps required to converge to a minimum. Our method can be implemented in any existing optimization code, requiring only implementation of derivatives of the Wilson B-matrix and the ability to numerically solve an ordinary differential equation.

## I. INTRODUCTION

Geometry optimization is a crucial first step in the computational modeling of molecules, solids, and other atomic systems. The most obvious way to optimize molecular geometries is to directly manipulate the Cartesian positions of each atom in the molecule. This approach is appropriate for condensed phase systems, in which the motion of each individual atom is constrained by the presence of nearby atoms. However, this is not true of gas-phase molecules, in which a group of atoms can easily move long distances through concerted motion. These kinds of displacements are large in magnitude in a Cartesian coordinate space, and therefore would require a large number of individual geometry optimization steps.

It is for this reason that optimization of molecular geometries is instead usually performed in an internal coordinate basis. In an internal coordinate basis, the geometry of a molecule is represented by a collection of intramolecular coordinates such as bond stretch distances, bending angles, and dihedral angles.<sup>1</sup> These chemically relevant features are directly modified by the optimization algorithm, which reduces the number of steps required to realize large-magnitude displacements through curvilinear steps in the internal coordinate space.

Several factors complicate optimization in an internal coordinate basis. In general, there are more chemically relevant bonds, bending angles, and dihedral angles than there are degrees of freedom in a molecule. Using a minimal basis of internal coordinates necessarily leaves some chemically relevant features out of the coordinate system, which reduces the efficiency by which these indirectly represented features can be optimized. If all chemically relevant coordinates are used instead, then the coordinate system will have a higher dimensionality than the number of degrees of freedom in the molecule. This latter approach is referred to as a redundant internal coordinate system.<sup>2,3</sup>

In a redundant internal coordinate system, a seemingly valid displacement vector may lead to a point in internal coordinate space that does not correspond to any physical arrangement of atoms. For example, while an infinitesimal displacement of a C-C-C-C dihedral angle in benzene may appear to preserve all C-C bond distances, any finite displacement will necessarily perturb at least one bond distance. The traditional method for solving this problem is to find a valid point in internal coordinate space which is closest to the invalid coordinates generated by the optimization algorithm.<sup>2,4,5</sup> While computationally facile, this

approach does not account for the coupling between internal coordinates; this coupling manifests as a curvature of the space of valid internal coordinates.

One may consider the space of valid internal coordinates as a manifold embedded in the higher-dimensional redundant internal coordinate space.<sup>6</sup> In order for a displacement vector to be valid, it must be tangent to the manifold at the starting geometry, and the final geometry after displacement must also lie in the manifold. Using this perspective, we propose a new method for realizing displacements in a redundant internal coordinate space in which the displacement vector is interpreted as tangent to a geodesic embedded in the manifold of internal coordinates. The geodesic follows the curvature of the manifold as the geometry evolves along the displacement trajectory. Applied to geometry optimization, this approach generates intermediate structures which converge to a minimum-energy structure in fewer steps than the standard approach.

## II. THEORY

The Cartesian coordinate vector of an  $n$ -atom molecule  $\mathbf{x} \in \mathbb{R}^{3n}$  encodes the geometry of a molecule as the Cartesian positions of each atom in that molecule. The internal coordinate vector  $\mathbf{q} \in \mathbb{R}^m$  encodes the geometry of a molecule in a set of  $m$  local coordinates, usually consisting of bond distances, bending angles, and dihedral angles.<sup>1</sup> These internal coordinates cannot represent net translation or rotation of the molecule, so in general only  $3n - 6$  internal coordinates are required to fully specify the geometry of non-linear molecules. When  $m$  is greater than its minimum possible value of  $3n - 6$ , it is said that the internal coordinate representation is redundant.<sup>2,3</sup> Even in a redundant internal coordinate basis, the set of all valid internal coordinate vectors  $\mathbf{q}$  only spans a  $(3n - 6)$ -dimensional space due to coupling between redundant internal coordinates. As described by Zhu *et. al.*,<sup>6</sup> the space of valid internal coordinates can be considered as a  $(3n - 6)$ -dimensional manifold embedded in a larger  $m$ -dimensional space. It is necessary to ensure that all geometry optimization steps in a redundant internal coordinate basis stay on the  $(3n - 6)$ -dimensional manifold, i.e. the steps correspond to valid internal coordinates. This means both that displacement vectors  $\Delta\mathbf{q} \in \mathbb{R}^m$  must be tangent to the internal coordinate manifold and that new structures obtained during optimization must be found in a way that accounts for curvature of the manifold.

One way to ensure that the displacement vector  $\Delta \mathbf{q}$  lies tangent to the manifold is to temporarily switch to a minimal local coordinate system in which only valid displacement vectors are possible. The delocalized internal coordinate approach constructs a new minimal coordinate system  $\mathbf{p} \in \mathbb{R}^{3n-6}$  as a linear transformation of the redundant internal coordinates. This is accomplished with the projection matrix  $\mathbf{U}$ , which is the  $(m \times (3n-6))$  matrix of left singular vectors of the Jacobian matrix  $\mathbf{B}$ , also known as the Wilson B-matrix,<sup>1,7</sup>

$$\mathbf{B} = \frac{\partial \mathbf{q}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{U} & \mathbf{U}' \end{bmatrix} \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}^T \\ \mathbf{V}'^T \end{bmatrix}, \quad (1)$$

where  $\mathbf{U}$  are the aforementioned left singular vectors,  $\mathbf{S}$  is the diagonal matrix of non-zero singular values,  $\mathbf{V}$  are the right singular vectors, and  $\mathbf{U}'$  and  $\mathbf{V}'$  are respectively the left and right singular vectors spanning the null space of  $\mathbf{B}$ .

The coordinates  $\mathbf{q}$ , the gradient at the current geometry  $\mathbf{g}$ , and the approximate Hessian matrix  $\mathbf{H}$  are projected into the delocalized internal coordinate space with  $\mathbf{U}$ ,

$$\mathbf{p} = \mathbf{U}^T \mathbf{q} \quad (2)$$

$$\hat{\mathbf{g}} = \mathbf{U}^T \mathbf{g} \quad (3)$$

$$\hat{\mathbf{H}} = \mathbf{U}^T \mathbf{H} \mathbf{U}. \quad (4)$$

The  $\mathbf{U}$  matrix is recalculated for every geometry visited during optimization in order to ensure that the delocalized internal coordinates remain a complete basis. Using these projected quantities, a variety of optimization algorithms such as rational function optimization (RFO)<sup>8</sup> or quasi-Newton BFGS<sup>9-12</sup> can be used to determine a displacement vector  $\Delta \mathbf{p}$  in the delocalized internal coordinate space. In RFO,  $\Delta \mathbf{p}$  is obtained from the leftmost eigenvector of the RFO eigenvalue equation,

$$\begin{pmatrix} \alpha^2 \hat{\mathbf{H}} & \alpha \hat{\mathbf{g}} \\ \alpha \hat{\mathbf{g}}^T & 0 \end{pmatrix} \begin{pmatrix} \Delta \mathbf{p} / \alpha \\ 1 \end{pmatrix} = 2\mu \begin{pmatrix} \Delta \mathbf{p} / \alpha \\ 1 \end{pmatrix}, \quad (5)$$

where  $\alpha$  is chosen such that  $\Delta \mathbf{p}$  satisfies the trust region condition (see appendix A for more details), and the eigenvalue  $\mu$  is not used. The displacement vector obtained from equation 5 is then projected back into the full redundant internal coordinate space,

$$\Delta \mathbf{q} = \mathbf{U} \Delta \mathbf{p}. \quad (6)$$

Though the process described above ensures that  $\Delta\mathbf{q}$  is tangent to the internal coordinate manifold at the current geometry  $\mathbf{q}_0$ , the point  $\mathbf{q}_0 + \Delta\mathbf{q}$  still may not lie on the manifold due to curvature. This can be a particularly severe problem for molecules with many highly-coupled internal coordinates, such as systems with multiple fused ring structures. This problem is traditionally solved by updating the geometry to be the point on the internal coordinate manifold which is closest to  $\mathbf{q}_0 + \Delta\mathbf{q}$ . This can be accomplished with Newton’s root-finding method, in which a series of rectilinear displacements are taken in the Cartesian coordinate basis according to the equation

$$x_{(k+1)}^i = x_{(k)}^i + \left(\mathbf{B}_{(k)}^+\right)_\lambda^i \left(q_0^\lambda + \Delta q^\lambda - q_{(k)}^\lambda\right), \quad i = 1, \dots, 3n, \quad (7)$$

where we have used the Einstein summation convention,  $\mathbf{q}_{(k)}$  and  $\mathbf{x}_{(k)}$  are respectively the internal and Cartesian coordinates at iteration  $k$ , and  $\mathbf{B}_{(k)}^+$  is the Moore-Penrose pseudo-inverse of the Jacobian matrix evaluated at  $\mathbf{x}_{(k)}$ .<sup>2,4,5</sup> In equation 7 and below, Latin indices correspond to quantities represented in Cartesian coordinates while Greek indices correspond to quantities represented in redundant internal coordinates. The converged Cartesian coordinates  $\mathbf{x}_{(k)}$  obtained from equation 7 are then used to calculate the new internal coordinates  $\mathbf{q}_{\text{Newton}}$ , which correspond to the point on the internal coordinate manifold closest to  $\mathbf{q}_0 + \Delta\mathbf{q}$ . Though each iteration of equation 7 consists of a rectilinear displacement in Cartesian coordinates, the Newton method results in an overall curvilinear displacement.

This approach is computationally facile and generally converges in only a few iterations, with the greatest cost being the evaluation and inversion of the Jacobian matrix. However, when the manifold has a high degree of redundancy or coupling between coordinates, such as in systems with rings, equation 7 may fail to converge. In this scenario, a common solution is to iterate equation 7 only a single time, which is equivalent to taking the rectilinear Cartesian displacement  $\mathbf{x}_0 + \mathbf{B}_0^+ \Delta\mathbf{q}$ .<sup>5</sup> This fallback approach can have substantial deleterious effects on optimization performance, as these rectilinear displacements tend to perturb bond distances when modifying bending angles or dihedral angles. Even when equation 7 does converge, it cannot fully account for the changing coupling between internal coordinates during the displacement because it does not explicitly consider the curvature of the manifold at any point.

As an alternative to the Newton approach, we suggest a new method for realizing a displacement vector  $\Delta\mathbf{q}$  based on geodesics of the internal coordinate manifold. Geodesics are

curves which trace the shortest path between two points on a manifold. In our application, the geodesic is determined from the starting geometry  $\mathbf{q}_0$  and a vector which is tangent to the geodesic, which we choose to be  $\Delta\mathbf{q}$ . The orientation of  $\Delta\mathbf{q}$  determines the trajectory of the geodesic  $\mathbf{q}(\tau)$ , where  $\tau$  is the dimensionless geodesic parameter, while the magnitude  $\|\Delta\mathbf{q}\|$  determines the distance along the trajectory to travel. The trajectory can be found by solving the geodesic equation,

$$\ddot{q}^\lambda + \Gamma_{\mu\nu}^\lambda \dot{q}^\mu \dot{q}^\nu = 0, \quad \lambda = 1, \dots, m \quad (8)$$

where Newton's dot notation is used to refer to derivatives with respect to  $\tau$  and  $\Gamma_{\mu\nu}^\lambda$  are the Christoffel symbols of the second kind for the internal coordinates (see appendix B for more details).<sup>13</sup> Equation 8 is solved for the initial conditions  $\mathbf{q}(0) = \mathbf{q}_0$  and  $\dot{\mathbf{q}}(0) = \Delta\mathbf{q}$ . The new geometry is determined by integrating  $\mathbf{q}(\tau)$  until  $\tau = 1$  using equation 8, which corresponds with a displacement distance of  $\|\Delta\mathbf{q}\|$  along the geodesic.

Equation 8 cannot be solved directly, as the internal coordinates  $\mathbf{q}$  are calculated from the Cartesian coordinates  $\mathbf{x}$  and are therefore not independent variables. Instead, we solve the geodesic equation in the Cartesian coordinate basis,

$$\ddot{x}^i + (\mathbf{B}^+)^i_\lambda \frac{\partial^2 q^\lambda}{\partial x^k \partial x^l} \dot{x}^k \dot{x}^l = 0, \quad i = 1, \dots, 3n \quad (9)$$

where  $\mathbf{x}(0) = \mathbf{x}_0$  are the Cartesian coordinates corresponding to  $\mathbf{q}_0$  and  $\dot{\mathbf{x}}(0) = \mathbf{B}^+ \Delta\mathbf{q}$ . The point  $\mathbf{x}(1)$  obtained from this differential equation is used to calculate the new internal coordinates  $\mathbf{q}(1)$ . Though equation 9 depends on the second derivative of  $\mathbf{q}$  with respect to  $\mathbf{x}$ , this quantity is not prohibitively onerous to implement for commonly-used internal coordinate types, and it has sparse structure that can be exploited to accelerate the summation over indices  $k$  and  $l$ . These second derivatives can be evaluated numerically from the Jacobian matrix,<sup>1</sup> analytically,<sup>14,15</sup> or through automatic differentiation.<sup>16</sup> Equation 9 can be solved using an off-the-shelf ODE solver such as LSODA<sup>17</sup> or CVODE<sup>18</sup> using a standard order reduction strategy.

Following a geometry step, it is typical for optimization algorithms to update an approximate Hessian matrix  $\mathbf{H}$  in order to satisfy the secant condition,

$$H_{\lambda\mu} (\mathbf{q}_1 - \mathbf{q}_0)^\mu = (\mathbf{g}_1 - \mathbf{g}_0)_\lambda, \quad \lambda = 1, \dots, m, \quad (10)$$

where  $\mathbf{g}$  is the gradient vector in the internal coordinate basis. In order for the approximate curvature to lie in the tangent space of the manifold at the new point  $\mathbf{q}_1$ , this secant condition

must be modified to

$$H_{\lambda\mu}(\dot{\mathbf{q}}(1))^\mu = (\mathbf{g}_1 - \tilde{\mathbf{g}}_0)_\lambda, \quad \lambda = 1, \dots, m, \quad (11)$$

where  $\dot{\mathbf{q}}(1)$  is obtained from the solution to equation 8 and  $\tilde{\mathbf{g}}_0$  is the gradient vector at point  $\mathbf{q}_0$  which has been parallel transported along the geodesic to the point  $\mathbf{q}_1$ .<sup>19</sup> Parallel transport is the process of translating vectors that are tangent to a manifold along a curvilinear trajectory on that manifold (such as a geodesic) in such a way that the vectors remain both tangent to the manifold along the entire trajectory and self-parallel along infinitesimal displacements. Details on how  $\tilde{\mathbf{g}}_0$  is determined are presented in appendix C. In the Hessian update scheme of our geodesic approach, the raw displacement  $\mathbf{q}_1 - \mathbf{q}_0$  is replaced by  $\dot{\mathbf{q}}(1)$ , and the initial gradient vector  $\mathbf{g}_0$  is replaced by its parallel transported equivalent  $\tilde{\mathbf{g}}_0$ . Our implementation of the algorithm initializes the Hessian matrix using the scheme of Fischer and Almlof,<sup>20</sup> while the Hessian matrix is updated using the TS-BFGS algorithm, as it is suitable for both minimization and saddle point optimization.<sup>21,22</sup>

An illustration comparing the geodesic and Newton stepping methods is presented in figure 1. In this figure, the purple surface represents the manifold of valid internal coordinates in a methane molecule in which only a single hydrogen atom is free to move at a fixed distance from the carbon atom. Thus, the only unconstrained internal coordinates of this molecule are three of the bending angles, as depicted in figure 1c. Though this system has three free bending angle coordinates, only two degrees of freedom remain due to the coupling between the angular coordinates. Figures 1a and 1b depict the entire manifold in a basis of the three free bending angles from two different perspectives. In this basis, the internal coordinate manifold takes the form of an octahedron with smoothed edges. The geodesic approach follows the curvature of the manifold to find the new point  $\mathbf{q}_{\text{geodesic}}$ . In contrast, the Newton method converges to the point  $\mathbf{q}_{\text{Newton}}$  on the manifold which is closest to  $\mathbf{q}_0 + \Delta\mathbf{q}$ . Figure 1d shows the same manifold, but rotated and zoomed to better illustrate the difference between the Newton and geodesic stepping methods. From figure 1d, it is clear that  $\mathbf{q}_{\text{Newton}}$  does not lie on the geodesic curve. This is to be expected, as the Newton stepping method is not aware of the curvature of the manifold, unlike the geodesic method which follows the curvature of the manifold by construction. For a convex manifold, the geodesic step length will always be greater than or equal to the Newton step length as a consequence of the triangle inequality. Despite this, the primary difference between the geodesic and Newton

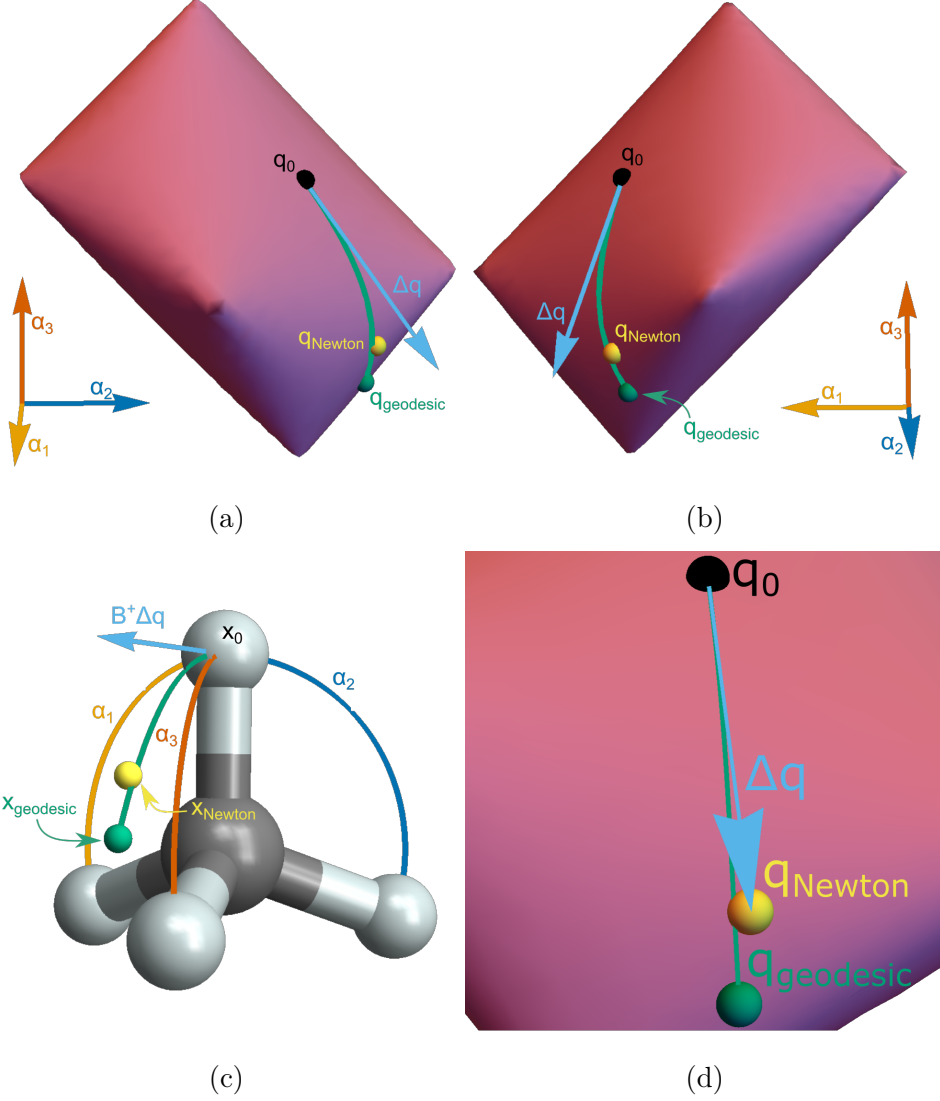


FIG. 1: (a), (b) The internal coordinate manifold (in purple) of a methane molecule with all internal coordinates fixed except three bending angles, from two different perspectives. Labeled are the initial structure  $\mathbf{q}_0$  (black), the displacement vector  $\Delta \mathbf{q}$  (light blue), the final structure of the Newton method  $\mathbf{q}_{\text{Newton}}$  (yellow), and the final structure of the geodesic method  $\mathbf{q}_{\text{geodesic}}$  (green). (c) A real-space representation of the same methane molecule with the three free bending angles labeled  $\alpha_1$  (orange),  $\alpha_2$  (dark blue), and  $\alpha_3$  (red). Additionally, the Cartesian equivalents of the initial structure, displacement vector, final Newton structure, and final geodesic structure are also labeled. (d) A zoomed-in perspective of the manifold in the region around the displacement, which shows more clearly that the point  $\mathbf{q}_{\text{Newton}}$  does not lie on the geodesic curve.



methods is not the magnitude of the displacement, but rather its trajectory.

### III. RESULTS

Though figure 1 demonstrates that the Newton and geodesic methods result in different structures, it is not immediately obvious which of the two stepping methods is better for geometry optimization. In order to determine the difference in performance between the Newton and geodesic methods, we use a geometry optimization benchmark originally developed by Birkholz and Schlegel consisting of 20 molecules that have between 20 and 95 atoms.<sup>5</sup> Potential energies were evaluated using DFTB+ with the DFTB3 parameterization.<sup>23-27</sup> Structure optimization was performed by Sella, an open source Python package primarily focused on saddle point optimization which is also capable of performing geometry minimization.<sup>28,29</sup> We note that because Sella is primarily intended to be used for saddle point optimization, the performance of its RFO minimization algorithm is likely lower than that of other purpose-built minimization codes. The focus is therefore only on the relative performance of the Newton and geodesic stepping approaches, with all other aspects of the minimization algorithm remaining identical. Of the original 20 molecules in the benchmark, one molecule was excluded due to a missing initial structure from the reference and another was excluded as DFTB3 lacks parameters for Aluminum. Scripts to reproduce these calculations can be found in the supplementary material.

The results in table I indicate that the geodesic approach requires fewer steps to reach convergence in all tested systems. For the molecule sphingomyelin, the geodesic approach converges to a lower energy structure than the Newton approach while also requiring fewer steps to converge. The optimization trajectories of two of the molecules, azadirachtin and sphingomyelin, are illustrated in figure 2. Trajectories of all tested molecules can be found in the supplementary material. Initially, the largest components of the gradients of the structures in this test set tend to lie in bond-stretching coordinates, and so early stages of geometry optimization are dominated by bond stretch displacements. These bond stretch displacements tend to be rectilinear or nearly rectilinear, meaning the manifold has very low curvature in these directions, and therefore the two methods tend to take very similar steps near the beginning of optimization. After the bond stretching modes are largely relaxed, the larger amplitude angle bending and dihedral angle coordinates begin to dominate the

Species (# of atoms)	Newton	Geodesic
Artemisinin (42)	122	33
Avobenzene (45)	307	90
Azadirachtin (95)	108	45
Bisphenol A (33)	268	89
Cetirizine (52)	178	34
Codeine (43)	270	107
Diisobutyl phthalate (42)	119	40
Estradiol (44)	174	47
Inosine <sup>a</sup> (31)	351	92
Maltose (45)	197	104
Mg Porphyrin (37)	86	15
Ochratoxin A (45)	212	47
Penicillin V (42)	167	55
Raffinose (66)	300	168
Sphingomyelin (84)	217 <sup>b</sup>	164
Tamoxifen (57)	211	58
Vitamin C (20)	160	60
Zn EDTA (33)	126	41
Mean	198.5	71.6
Standard deviation	73.7	42.0

<sup>a</sup> The structure provided by reference 5 erroneously replaced a nitrogen atom with a carbon atom. We have corrected this error in our calculations.

<sup>b</sup> Converges to a higher-energy structure

TABLE I: Number of gradient evaluations required to converge for the standard and geodesic stepping methods.

optimization; it is at this point that the Newton and geodesic methods begin to exhibit different performance characteristics. In this regime, the Newton method more frequently takes steps that result in an increase in the potential energy as evidenced by the many spikes

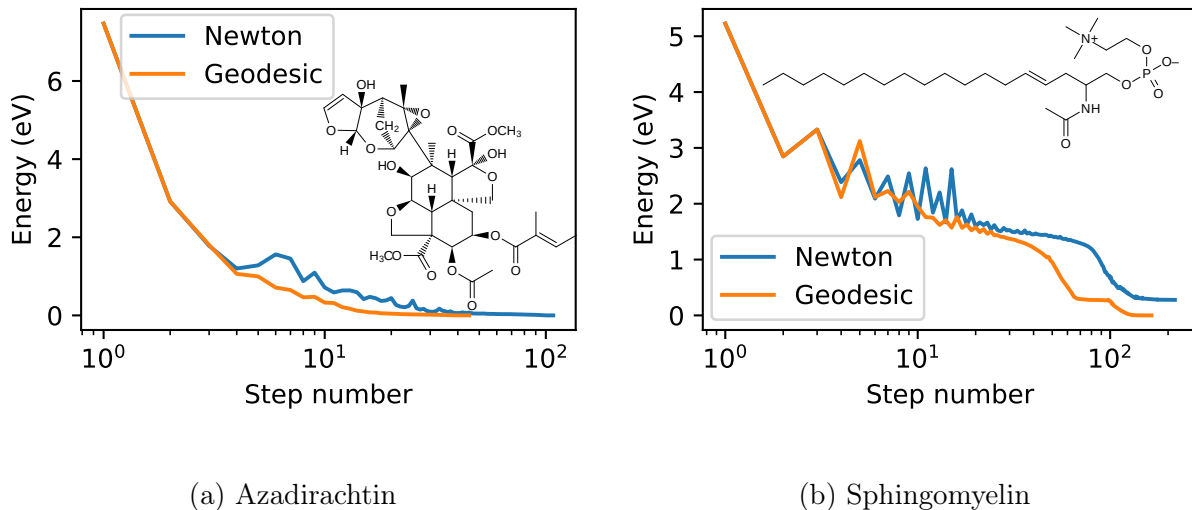


FIG. 2: Optimization trajectories for the azadirachtin (a) and sphingomyelin (b) test molecules using the Newton (blue) and geodesic (orange) methods. A log scale is used for the step number axis to better highlight early optimization steps.

in the optimization trajectories in figure 2. In contrast, the geodesic method is less likely to take steps that increase the potential energy and generally reaches convergence in fewer steps overall compared to the Newton method. Azadirachtin, as illustrated in figure 2a, is an example of a molecule with several fused ring and bicyclic structures which benefits substantially from the geodesic stepping method, as previously suggested.

We believe that this difference in performance is not primarily a consequence of the magnitude of the individual displacement steps, but rather the differing displacement trajectories generated by the two methods. In Sella’s primary application of saddle point optimization, preliminary results suggest a substantial increase in performance compared to other leading algorithms, which we intend to show in a future publication.

#### IV. CONCLUSION

We have presented a new approach for realizing displacements in a redundant internal coordinate space by translating the molecular geometry along geodesics of the internal coordinate manifold. In contrast to the traditional Newton stepping approach, our method substantially reduces the number of steps required to reach convergence, particularly for systems with highly coupled internal coordinates, such as molecules with multiple fused rings.

The improved performance of the geodesic stepping approach is due to its consideration of the coupling between internal coordinates along the entire displacement trajectory. Our method is straightforward to implement, only requiring second derivatives of the internal coordinates (i.e. first derivatives of the Wilson B-matrix) and numerical solution of a non-stiff first-order differential equation.

## **SUPPLEMENTARY MATERIAL**

Additional optimization trajectories such as those presented in figure 2 for all tested systems can be found in the supplementary material. Additionally, the Python scripts used to generate these trajectories and the data presented in table I are also available in the supplementary material.

## **ACKNOWLEDGMENTS**

This work was supported by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Chemical Sciences, Geosciences and Biosciences Division, as part of the Computational Chemistry Sciences Program (Award Number: 0000232253).

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

We would like to thank Dr. Laura McCaslin for useful discussions leading to improved readability and comprehensibility of this work.

## **DATA AVAILABILITY STATEMENT**

The data and code that support the findings of this study are available within the article, its supplementary material, and from Zenodo at <http://doi.org/10.5281/zenodo.4747052>, reference number 29.

## Appendix A: Trust region algorithm

In our implementation of trust-region RFO, the parameter  $\alpha$  of equation 5 is chosen to satisfy the trust region condition  $\|\Delta\mathbf{p}\|_\infty \leq \delta_k$ , where  $\delta_k$  is the size of the trust region at iteration  $k$ . First, equation 5 is solved for  $\alpha = 1$ , and if the resulting  $\Delta\mathbf{p}$  satisfies the trust region condition, the displacement is accepted. Otherwise, bisection is used to determine a value of  $\alpha$  for which  $\|\Delta\mathbf{p}\|_\infty = \delta_k$ . After every iteration,  $\delta_k$  is updated based on the agreement between the expected change in electronic potential energy and the actual change in energy. The ratio between the predicted and true change in energy is

$$\rho = \frac{\mathbf{g}^T \Delta\mathbf{q} + \frac{1}{2} \Delta\mathbf{q}^T \mathbf{H} \Delta\mathbf{q}}{\Delta\epsilon}, \quad (\text{A1})$$

where  $\Delta\epsilon$  is the true change in energy following a geometry optimization step. Two threshold values,  $\rho_{\text{inc}}$  and  $\rho_{\text{dec}}$  are used to determine whether to increase or decrease the trust radius, respectively. If  $\rho > \rho_{\text{dec}}$  or  $\rho < \rho_{\text{dec}}^{-1}$ , this indicates that the quadratic approximation is poor quality and that the trust region should be decreased. For minimization, this condition will be met whenever the energy increases during optimization, as  $\rho$  will become negative while  $\rho_{\text{dec}}$  is always positive. In this case, the new trust region is chosen to be

$$\delta_{k+1} = \sigma_{\text{dec}} \|\Delta\mathbf{p}\|_\infty, \quad (\text{A2})$$

where  $\sigma_{\text{dec}} < 1$  is a parameter of the method. If  $\rho_{\text{inc}}^{-1} < \rho < \rho_{\text{inc}}$ , this indicates that the quadratic approximation is highly accurate and that the trust region can be expanded. In this case, the new trust region is chosen to be

$$\delta_{k+1} = \max(\delta_k, \sigma_{\text{inc}} \|\Delta\mathbf{p}\|_\infty), \quad (\text{A3})$$

where  $\sigma_{\text{inc}} > 1$  is a parameter of the method.

For this work, we use the following manually-tuned parameters:  $\delta_0 = 0.2$  (units ambiguous, *vide infra*),  $\rho_{\text{dec}} = 100$  (effectively, the trust region only shrinks when the energy increases),  $\rho_{\text{inc}} = 1.035$ ,  $\sigma_{\text{dec}} = 0.90$ , and  $\sigma_{\text{inc}} = 1.15$ . Note that because  $\mathbf{p}$  is a linear combination of bond stretches, angle bends, and dihedral angles, the units of  $\delta$  are somewhat ambiguous. In our implementation, angstroms are used for bond stretches while radians are used for both angle bends and dihedral angles. Choice of the units impacts the performance of the algorithm. For example, using degrees for the angle coordinates instead of

radians would result in a decrease in the relative magnitude of displacements in the angle coordinates as compared to the bond stretch coordinates. Similarly, using atomic units of distance for bond stretch coordinates would result in a decrease in the relative magnitude of displacements in the bond stretch coordinates as compared to the angle coordinates.

## Appendix B: Christoffel symbols and the metric tensor

The Christoffel symbols  $\Gamma_{kl}^i$  of a manifold contain information about the curvature of the manifold. The symbols are related to the metric tensor  $\mathbf{G}$  through the relation

$$\Gamma_{kl}^i = \frac{1}{2} (\mathbf{G}^{-1})^{im} \left( \frac{\partial G_{mk}}{\partial q^l} + \frac{\partial G_{ml}}{\partial q^k} - \frac{\partial G_{kl}}{\partial q^m} \right), \quad i, k, l = 1, \dots, 3n. \quad (\text{B1})$$

The metric tensor  $\mathbf{G}$  is a representation of the metric function  $g(\mathbf{u}, \mathbf{v})$  which generalizes the notion of the dot product to curved manifolds,

$$g(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{G} \mathbf{v}. \quad (\text{B2})$$

For the current application, the metric tensor is that of the internal coordinate space as represented in a Cartesian coordinate basis, which is given by

$$G_{ik} = B_i^\lambda I_{\lambda\mu} B_k^\mu = \frac{\partial q^\lambda}{\partial x^i} I_{\lambda\mu} \frac{\partial q^\mu}{\partial x^k}, \quad i, k = 1, \dots, 3n, \quad (\text{B3})$$

where  $\mathbf{I}$  is the identity matrix. Evaluating equation B1 with the definition of  $\mathbf{G}$  given by equation B3 results in the definition of  $\Gamma_{kl}^i$  used in equation 9.

## Appendix C: Solving the geodesic equation and parallel transport

The geodesic is found by solving the differential equation outlined in equation 9. This equation is reduced to a first-order differential equation by substituting  $\mathbf{y}_1 = \mathbf{x}$  and  $\mathbf{y}_2 = \dot{\mathbf{x}}$ ,

$$\dot{y}_1^i = y_2^i, \quad i = 1, \dots, 3n \quad (\text{C1})$$

$$\dot{y}_2^i = -(\mathbf{B}^+)_\lambda^i \frac{\partial^2 q^\lambda}{\partial x^k \partial x^l} y_2^k y_2^l, \quad i = 1, \dots, 3n. \quad (\text{C2})$$

Parallel transport of the gradient vector  $\mathbf{g}_0$  from the point at which it is evaluated  $\mathbf{q}_0$  along the geodesic to the new point  $\mathbf{q}_1$  is accomplished by augmenting equations C1 and C2 with a third variable  $\mathbf{y}_3$ ,

$$\dot{y}_3^i = -(\mathbf{B}^+)_\lambda^i \frac{\partial^2 q^\lambda}{\partial x^k \partial x^l} y_2^k y_3^l, \quad i = 1, \dots, 3n, \quad (\text{C3})$$

with the initial condition  $\mathbf{y}_3(0) = \mathbf{B}_0^+ \mathbf{g}_0$ , where  $\mathbf{B}_0^+$  is the Moore-Penrose pseudo-inverse of the Jacobian matrix evaluated for geometry  $\mathbf{x}_0 = \mathbf{y}_1(0)$ .<sup>13</sup> The parallel transported gradient vector  $\tilde{\mathbf{g}}_0$  is given by

$$\tilde{\mathbf{g}}_0 = \mathbf{B}_1 \mathbf{y}_3(1), \tag{C4}$$

where  $\mathbf{B}_1$  is the Jacobian matrix evaluated at  $\mathbf{x}_1 = \mathbf{y}_1(1)$ . The final differential equation is solved in terms of  $\mathbf{y} = [\mathbf{y}_1^T \ \mathbf{y}_2^T \ \mathbf{y}_3^T]^T$  with the SciPy implementation of the LSODA ODE solver.

## REFERENCES

- <sup>1</sup>E. Wilson, J. Decius, and P. Cross, *Molecular Vibrations: The Theory of Infrared and Raman Vibrational Spectra*, Dover Books on Chemistry (Dover Publications, 2012).
- <sup>2</sup>P. Pulay and G. Fogarasi, *The Journal of Chemical Physics* **96**, 2856 (1992).
- <sup>3</sup>C. Peng, P. Y. Ayala, H. B. Schlegel, and M. J. Frisch, *Journal of Computational Chemistry* **17**, 49 (1996).
- <sup>4</sup>P. Pulay, G. Fogarasi, F. Pang, and J. E. Boggs, *Journal of the American Chemical Society* **101**, 2550 (1979).
- <sup>5</sup>A. B. Birkholz and H. B. Schlegel, *Theoretical Chemistry Accounts* **135**, 84 (2016).
- <sup>6</sup>X. Zhu, K. C. Thompson, and T. J. Martínez, *The Journal of Chemical Physics* **150**, 164103 (2019).
- <sup>7</sup>J. Baker, A. Kessi, and B. Delley, *The Journal of Chemical Physics* **105**, 192 (1996).
- <sup>8</sup>A. Banerjee, N. Adams, J. Simons, and R. Shepard, *The Journal of Physical Chemistry* **89**, 52 (1985).
- <sup>9</sup>C. G. Broyden, *IMA J. Appl. Math.* **6**, 76 (1970).
- <sup>10</sup>R. Fletcher, *Comput. J.* **13**, 317 (1970).
- <sup>11</sup>D. Goldfarb, *Math. Comput.* **24**, 23 (1970).
- <sup>12</sup>D. F. Shanno, *Math. Comput.* **24**, 647 (1970).
- <sup>13</sup>M. Spivak, *A Comprehensive Introduction to Differential Geometry*, A Comprehensive Introduction to Differential Geometry No. v. 3 (Brandeis University, 1970).
- <sup>14</sup>D. S. Hollman and H. F. Schaefer, *The Journal of Chemical Physics* **137**, 164103 (2012).
- <sup>15</sup>L. M. McCaslin, *From Basis Sets to Force Fields: Improving Methods for High-Accuracy*

- Quantum Chemical Calculations of Small Molecules*, Ph.D. thesis, The University of Texas at Austin (2016).
- <sup>16</sup>J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, “JAX: composable transformations of Python+NumPy programs,” (2018).
  - <sup>17</sup>L. Petzold, SIAM Journal on Scientific and Statistical Computing **4**, 136 (1983).
  - <sup>18</sup>S. D. Cohen, A. C. Hindmarsh, and P. F. Dubois, Computers in Physics **10**, 138 (1996).
  - <sup>19</sup>D. Gabay, Journal of Optimization Theory and Applications **37**, 177 (1982).
  - <sup>20</sup>T. H. Fischer and J. Almlof, The Journal of Physical Chemistry **96**, 9768 (1992).
  - <sup>21</sup>J. M. Anglada and J. M. Bofill, J. Comput. Chem. **19**, 349 (1998).
  - <sup>22</sup>J. M. Bofill, Int. J. Quantum Chem. **94**, 324 (2003).
  - <sup>23</sup>B. Hourahine, B. Aradi, V. Blum, F. Bonafé, A. Buccheri, C. Camacho, C. Cevallos, M. Y. Deshayé, T. Dumitrică, A. Dominguez, S. Ehlert, M. Elstner, T. van der Heide, J. Hermann, S. Irle, J. J. Kranz, C. Köhler, T. Kowalczyk, T. Kubař, I. S. Lee, V. Lutsker, R. J. Maurer, S. K. Min, I. Mitchell, C. Negre, T. A. Niehaus, A. M. N. Niklasson, A. J. Page, A. Pecchia, G. Penazzi, M. P. Persson, J. Řezáč, C. G. Sánchez, M. Sternberg, M. Stöhr, F. Stuckenberg, A. Tkatchenko, V. W.-z. Yu, and T. Frauenheim, The Journal of Chemical Physics **152**, 124101 (2020).
  - <sup>24</sup>M. Gaus, A. Goez, and M. Elstner, Journal of Chemical Theory and Computation **9**, 338 (2013).
  - <sup>25</sup>M. Gaus, X. Lu, M. Elstner, and Q. Cui, Journal of Chemical Theory and Computation **10**, 1518 (2014).
  - <sup>26</sup>X. Lu, M. Gaus, M. Elstner, and Q. Cui, The Journal of Physical Chemistry B **119**, 1062 (2015).
  - <sup>27</sup>M. Kubillus, T. Kubař, M. Gaus, J. Řezáč, and M. Elstner, Journal of Chemical Theory and Computation **11**, 332 (2015).
  - <sup>28</sup>E. D. Hermes, K. Sargsyan, H. N. Najm, and J. Zádor, Journal of Chemical Theory and Computation **15**, 6536 (2019).
  - <sup>29</sup>E. D. Hermes, “Sella 2.0.2.geo1,” (2021).