

# An Exact Expression to Calculate the Derivatives of Position-Dependent Observables in Molecular Simulations with Flexible Constraints

Pablo Echenique<sup>1,2,3,4\*</sup>, Claudio N. Cavasotto<sup>5</sup>, Monica De Marco<sup>3</sup>, Pablo Garca-Risueño<sup>1,2,3</sup>, J.L. Alonso<sup>3,2,4</sup>

**1** Instituto de Química Física "Rocasolano", Consejo Superior de Investigaciones Científicas (CSIC), Madrid, Spain, **2** Instituto de Biocomputación y Física de Sistemas Complejos (BIFI), Universidad de Zaragoza, Zaragoza, Spain, **3** Departamento de Física Teórica, Universidad de Zaragoza, Zaragoza, Spain, **4** Unidad Asociada IQFR-BIFI, Zaragoza, Spain, **5** School of Biomedical Informatics, University of Texas Health Science Center at Houston, Houston, Texas, United States of America

## Abstract

In this work, we introduce an algorithm to compute the derivatives of physical observables along the constrained subspace when flexible constraints are imposed on the system (i.e., constraints in which the constrained coordinates are fixed to configuration-dependent values). The presented scheme is exact, it does not contain any tunable parameter, and it only requires the calculation and inversion of a sub-block of the Hessian matrix of second derivatives of the function through which the constraints are defined. We also present a practical application to the case in which the sought observables are the Euclidean coordinates of complex molecular systems, and the function whose minimization defines the flexible constraints is the potential energy. Finally, and in order to validate the method, which, as far as we are aware, is the first of its kind in the literature, we compare it to the natural and straightforward finite-differences approach in a toy system and in three molecules of biological relevance: methanol, N-methyl-acetamide and a tri-glycine peptide.

**Citation:** Echenique P, Cavasotto CN, De Marco M, Garca-Risueño P, Alonso JL (2011) An Exact Expression to Calculate the Derivatives of Position-Dependent Observables in Molecular Simulations with Flexible Constraints. PLoS ONE 6(9): e24563. doi:10.1371/journal.pone.0024563

**Editor:** Darren R. Flower, Aston University, United Kingdom

**Received:** December 30, 2010; **Accepted:** August 15, 2011; **Published:** September 12, 2011

**Copyright:** © 2011 Echenique et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Funding:** This work was funded by FIS2009-13364-C02-01 (MICINN, Spain), Grupo de Excelencia "Biocomputación y Física de Sistemas Complejos", E24/3 (DGA, Spain), and ARAID and Ibercaja grant for young researchers (DGA and Ibercaja, Spain). P. G.-R. is supported by a JAE-Predoc scholarship (CSIC, Spain). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing Interests:** The authors have declared that no competing interests exist.

\* E-mail: echenique.p@gmail.com

## Introduction

In the theoretical and computational modeling of physical systems, including but not limited to condensed-matter materials [1], fluids [2], and biological molecules [3], it is very common to appeal to the concept of *constraints*. When a given quantity related to the system under study is constrained, it is not allowed to depend explicitly on time (or on any other parameter that describes the evolution of the system in the problem at hand). Instead, a constrained quantity is either set to a constant value (*hard* or *rigid* constraints) or to a function of the rest of degrees of freedom (*flexible*, *elastic* or *soft* constraints); in such a way that, if it depends on time, it does so through the latter and not in an explicit manner.

The imposition of constraints is useful in a wide variety of contexts in the fields of computational physics and chemistry: For example, we can use constraints to maintain an exact symmetry of the equations of motion; like in Car-Parrinello molecular dynamics (MD) [4], where the time-dependent Kohn-Sham orbitals need to be orthonormal along the time evolution of the quantum-classical system, a requirement that can be fulfilled by imposing constraints over their scalar product [5]. In a different context, we can use constraints, as in the Blue Moon Ensemble technique [6], to fix some macroscopic, representative degrees of freedom of molecular systems (normally called *reaction coordinates*),

in order to be able to compute free energy profiles along them that would take an unfeasibly long time if we used an unconstrained simulation. Probably the most common application of the idea of constraints, and the one that will be mainly discussed in this work, appears when we fix the fastest, hardest degrees of freedom of molecular systems, such as bond lengths or bond angles, in order to allow for larger time-steps in MD simulations [7,8].

In any of these cases (assuming that the dimensions of the spaces involved are all finite) the imposition of constraints can be described in the following way: If the state of the system is parameterized by a given set of coordinates  $q = (q^\mu)_{\mu=1}^N$ , spanning the *whole space*,  $\mathcal{W}$ , and the associated momenta  $p = (p_\mu)_{\mu=1}^N$ , a given *constrained subspace*,  $\mathcal{K}$ , of dimension  $K < N$ , can be defined by giving a set of  $L := N - K$  independent relations among the coordinates (In this work, we will only deal with holonomic, scleronomous constraints, i.e., those that are independent both of the momenta and (explicitly) of time.):

$$h^I(q) = 0, \quad I = K + 1, \dots, N. \quad (1)$$

The condition of these constraints being independent amounts to asking the set of  $L$  vectors of  $N$  components

$$\partial h^I(q) := \left( \frac{\partial h^I}{\partial q^\mu}(q) \right)_{\mu=1}^N, \quad I=K+1, \dots, N, \quad (2)$$

to be linearly independent at the relevant points  $q$  satisfying (1), and it means that  $\mathcal{K}$  is a manifold of constant dimension in these points, which are called *regular*. Moreover, this independence condition allows, in the vicinity of each point  $q$  and by virtue of the Implicit Function Theorem [9,10], to (formally) solve (1) for  $L$  of the coordinates, which we arbitrarily place at the end of  $q$ , splitting the original set as  $q=(u,d)$ , with  $u := (u^r)_{r=1}^K$  and  $d := (d^I)_{I=K+1}^N$ . Then, in the vicinity of each point  $q$  satisfying (1), we can express the relations defining the constrained subspace,  $\mathcal{K}$ , *parametrically* by

$$d^I = f^I(u), \quad I=K+1, \dots, N. \quad (3)$$

where the functions  $f(u) := (f^I(u))_{I=K+1}^N$  are the ones whose existence the Implicit Function Theorem guarantees. The coordinates  $u$  are thus termed *unconstrained* and they parameterize  $\mathcal{K}$ , whereas the coordinates  $d$  are called *constrained* and their value is determined at each point of  $\mathcal{K}$  according to (3). In general, the functions  $f^I$  will depend on  $u$ , and the constraints will be said to be *flexible* [11]. In the particular case in which all the functions  $f^I$  are constant along  $\mathcal{K}$ , the constraints are called *hard*, and all the calculations are considerably simplified. In this work, we tackle the general, more involved, flexible case.

Of course, even if  $\mathcal{K}$  is regular in all of its points, the particular coordinates  $d^I$  that can be solved need not be the same along the whole space. One of the simplest examples of this being the circle in  $\mathbb{R}^2$ , which is given by  $f(x,y) := x^2 + y^2 - R^2 = 0$ , an implicit expression whose gradient is non-zero for all  $(x,y) \in \mathcal{K}$ . However, if we try to solve, say, for  $y$  in the whole space  $\mathcal{K}$ , we will run into trouble at  $y=0$ ; if we try to solve for  $x$ , we will find it to be impossible at  $x=0$ . I.e., the Implicit Function Theorem does guarantee that we can solve for *some* of the original coordinates at each regular point of  $\mathcal{K}$ , but sometimes the solved coordinate has to be  $x$  and sometimes it has to be  $y$ . Nevertheless, we will assume this to be the case throughout this work, as is normally done in the literature [12–15], and thus we will consider that  $\mathcal{K}$  is parameterized by the *same* subset of coordinates  $u$  for all of its points.

It is also worth mentioning at this point that, not only from the physical point of view all the constraints dealt with in this work are just holonomic constraints, but also the wording used to refer to the two *flexible* and *hard* sub-types is multiple in the literature. The first sub-type is called *flexible* in refs. [13–16], *elastic* in [17], and *soft* in [15]; whereas the second sub-type is called *hard* in refs. [15,16], just *constrained* in [13], or *holonomic* in [17], *rigid* in [14,15], and *fully constrained* in [15]. Some of these terms are clearly misleading (*elastic*, *holonomic* or *fully constrained*), and, in any case, so many names for such simple concepts is detrimental to understanding in the field.

The situation is further complicated by the fact that, when studying the statistical mechanics of constrained systems, one can think about two different models for calculating the equilibrium probability density, whose names often collide with the ones used for defining the type of constraints applied. On the one hand, one can implement the constraints by the use of very steep potentials around the constrained subspace; a model sometimes called *flexible* [18,19], sometimes called *stiff* [12,20]. On the other hand, one can assume the D'Alembert principle [21] and hypothesize that the

forces are just the ones needed for the system to never leave the constrained subspace during its dynamical evolution; a model normally called *rigid* [12,18,19]. The two statistical mechanics models have long been recognized to present different equilibrium probability distributions [18–20,22], and this is the major concern in the literature when discussing them. In refs. [11,12], the reader can find a very detailed discussion of this issue, which we only touch here briefly for completeness.

It is worth remarking that the two types of constraints and the two types of statistical mechanics models can be independently combined; one can have either the *stiff* or the *rigid* model, with either *flexible* or *hard* constraints, hence making any interference between the two sets of words undesirable. The wording chosen is this work is, on the one hand, fairly common, and on the other hand, non-misleading.

Now, if we take any physical observable  $X(q)$ , depending only on the coordinates (not on the momenta), and originally defined on the whole space,  $\mathcal{W}$ , its *restriction* to  $\mathcal{K}$  is given by

$$Z(u) := X(u, f(u)), \quad (4)$$

where the symbol has been deliberately changed in order to indicate that  $Z$  and  $X$  are different functions.

The derivatives of this observable along  $\mathcal{K}$  are thus

$$\frac{\partial Z}{\partial u^r}(u) = \frac{\partial X}{\partial u^r}(u, f(u)) + \frac{\partial X}{\partial d^I}(u, f(u)) \frac{\partial f^I}{\partial u^r}(u), \quad (5)$$

where we have assumed the convention that repeated indices (like  $I$  above) indicate a sum over the relevant range, and we have omitted (as we will often do) the range of variation of the index  $r$ .

In the case of hard constraints, i.e., when the functions  $f^I$  are all constant numbers  $d_0^I$ , the above expression reduces to

$$\frac{\partial Z}{\partial u^r}(u) = \frac{\partial X}{\partial u^r}(u, d_0), \quad (6)$$

where  $X(q)$  must be a known function of  $q$  (in order to have a well-defined problem), and its derivative is typically easy to compute. However, if the constraints are of the more general, flexible form (the ones tackled in this work), the calculation of the partial derivatives  $(\partial f^I / \partial u^r)(u)$  cannot be avoided.

If the constraints are assumed to be flexible, it is common in the literature of molecular modeling to define these functions  $f^I(u)$  as the values taken by the coordinates  $d$  if we minimize either the total or the potential energy with respect to all  $d^I$  at fixed  $u$  [11–15]. Since the energy functions used in molecular simulation are typically rather complicated, such as the ones in classical force fields, with a large number of distinct functional terms [23–28], or the effective nuclear potential arising from the solution of the electronic Schrödinger equation in the ground-state Born-Oppenheimer approximation [12,29], the minimization of the energy with respect to the coordinates  $d$  has to be performed numerically. Hence, the functions  $f^I(u)$ , which are the output of this process, do not have a compact analytical expression that can be easily differentiated to include it in eq. (5) (this is even the case in very simple toy systems; see the Results and Discussion section).

In this work, we present a parameter-free, exact algorithm (up to machine precision) to calculate the derivatives  $(\partial f^I / \partial u^r)(u)$  in such a case. Although several methods exist in the literature [13–15] for performing MD simulations with flexible constraints, nobody has dealt, as far as we are aware, with the computation of these derivatives. Since the general idea can be applied to any

situation in which (1) we have flexible constraints, (2) that are defined in terms of the minimization of some quantity with respect to the constrained coordinates, we first introduce, the essential part of the algorithm based on these two points. Then, we develop a more sophisticated application of this idea to the calculation of the derivatives along the constrained subspace of the Euclidean coordinates of molecular systems; a problem that we faced in our group when trying to calculate the correcting terms associated to mass-metric tensor determinants that appear in the equilibrium probability density when constraints are imposed [11,12,30]. Finally, we perform a comparison between the results obtained with our exact algorithm and the calculation of the derivatives by finite differences; this serves the double purpose of numerically validating the algorithm and showing the limitations of the latter method, which needs the tuning of a parameter for each particular problem.

## Methods

### General Algorithm

As we mentioned in the Introduction, we assume that we are dealing with a constrained problem in which the functions  $f^I(u)$  in eq. (3) are defined as taking the values of the constrained coordinates  $d^I$  that minimize a given function,  $V(q) = V(u, d)$ , for each fixed  $u$ , i.e.,

$$V(u, f(u)) \leq V(u, d), \forall d \in \mathcal{D}(f(u)), \quad (7)$$

where  $\mathcal{D}(f(u))$  is a suitable open set in  $\mathbb{R}^L$  containing the point  $f(u)$ . Depending on the particular application, one can ask the minimum that defines the functions  $f^I(u)$  to be global or just local. However, in the cases in which  $V$  is the total or the potential energy of a complex molecular system, it may become very difficult to find its global minimum (due to the sheer number of dimensions of the search space), and the local choice is the only reasonable one [11].

In order to calculate the derivatives along  $\mathcal{K}$  of any physical observable function of the coordinates  $Z(u) := X(u, f(u))$ , like the one defined in (4), we can always follow the finite-differences approach. However, as we discuss in the Results and Discussion section, finite differences presents intrinsic inaccuracies which are difficult to overcome, specially as the system grows larger. Let us now introduce a different way to calculate  $(\partial Z / \partial u^r)(u)$  which does not suffer from this drawback.

The starting point is eq. (5) in the Introduction, which we copy here for the comfort of the reader:

$$\frac{\partial Z}{\partial u^r}(u) = \frac{\partial X}{\partial u^r}(u, f(u)) + \frac{\partial X}{\partial d^I}(u, f(u)) \frac{\partial f^I}{\partial u^r}(u). \quad (8)$$

As we mentioned, the expression of  $X(q)$ , as well as the functions  $f^I(u)$ , must be known if we wish to have a well-defined constrained problem to begin with. Therefore, the only objects that remain to be computed are the partial derivatives  $(\partial f^I / \partial u^r)(u)$ .

If we assume that we have available some method to check that the order of the stationary point is the appropriate one (i.e., that it is a minimum, and not a maximum or a saddle point), we can write a set of equations which are equivalent to eq. (7), and which (implicitly) define the functions  $f^I(u)$ :

$$\frac{\partial V}{\partial d^I}(u, f(u)) = 0, \quad I = K + 1, \dots, N. \quad (9)$$

Now, we can take the derivative of this expression with respect to a given unconstrained coordinate  $u^r$ :

$$\underbrace{\frac{\partial^2 V}{\partial u^r \partial d^I}(u, f(u))}_{H_{rI}(u)} + \underbrace{\frac{\partial^2 V}{\partial d^J \partial d^I}(u, f(u))}_{H_{JI}(u)} \underbrace{\frac{\partial f^J}{\partial u^r}(u)}_{F_r^J(u)} = 0, \quad (10)$$

where  $H_{\mu\nu}(u)$ , with  $\mu, \nu = 1, \dots, N$ , is the Hessian matrix of  $V$  evaluated at  $(u, f(u)) \in \mathcal{K}$ , and  $F_r^J(u)$  is the matrix of unknowns that we want to solve for. In the whole document, we adhere to the practice of using different types of indices in order to indicate different ranges of variation. Here, for example,  $\mu, \nu, \rho, \dots$  run from 1 to  $N$ ;  $r, s, t, \dots$  run from 1 to  $K$ ; and  $I, J, M, \dots$  run from  $K + 1$  to  $N$ . In the next section, we need to use more types of indices, but the idea is the same.

It is worth mentioning that similar equations to the ones above can be found in classical mechanics anytime that local coordinates are used (the coordinates  $u$  in this work). For example, the force in such a case is defined as  $\partial V / \partial u^r$  and the chain rule can be used in a similar way to what we do here. Note, however, that eq. (9) does not contain derivatives with respect to  $u^r$ , but to the constrained coordinate  $d^I$ . This makes the approach slightly different and, indeed, eq. (10) would become trivial in the most common hard situation tackled in the literature, where  $\partial f^J / \partial u^r = 0, \forall J, r$ .

Since we are, by hypothesis, in a minimum of  $V$  with respect to the constrained coordinates  $d$ , the constrained sub-block  $H_{JI}(u)$  of the Hessian is a positive definite matrix, and therefore invertible. Hence, if we multiply eq. (10) by its inverse, denoted by  $H^{JM}(u)$ , sum over  $I$ , exploit the fact that  $H_{\mu\nu}(u)$  and  $H^{\mu\nu}(u)$  are symmetric, and conveniently rename the indices, we arrive at:

$$\frac{\partial f^I}{\partial u^r}(u) = : F_r^I(u) = -H^{IJ}(u) H_{Jr}(u), \quad (11)$$

which, as promised, allows us to find the exact derivatives  $(\partial f^I / \partial u^r)(u)$  with the only knowledge of the Hessian of  $V$  at the point  $(u, f(u))$ , and, upon introduction of the result in eq. (8), also the derivatives along the constrained subspace  $\mathcal{K}$  of any physical observable  $X(q)$ .

As mentioned, several methods exist in the literature [13–15] to perform MD simulations with flexible constraints, however, none of them has tackled the calculation of these derivatives, which are very basic objects presumably to be needed in many future applications (see, e.g., refs. [11,12]). Of course, it is always possible to compute derivatives using the simple and straightforward method of finite differences. In this work, we use finite differences as a way of validating the new, exact method and ensuring it is error free.

The accuracy of the new algorithm is only limited by the accuracy with which we can calculate the Hessian of  $V$  at  $(u, f(u))$  and invert it; there is no tunable parameter that we need to adjust for optimal accuracy, as in the case of finite differences (see below and also Results and Discussion). This makes a difference because,

in classical force fields [26] and even in some quantum chemical methods (e.g., see chap. 10 of [31]), the Hessian can be calculated analytically, without the need of finite differences.

Although no optimization of the numerical cost has been pursued in this work, some remarks can be made about it, in comparison with the cost of the finite-differences approach. In order to calculate the partial derivatives  $\partial Z/\partial u^s$  with respect to the unconstrained coordinates  $u$  using finite differences, we need to:

1. Minimize  $V(u, d)$  at fixed  $u$  to find  $f(u)$  (this step is common with the new method introduced here).
2. Calculate  $Z(u) := X(u, f(u))$  (this step is common with the new method introduced here).
3. Choose a displacement  $\Delta$  and minimize  $V(\tilde{u}, d)$  at the point  $\tilde{u} := (\tilde{u}^r)_{r=1}^K$ , where  $\tilde{u}^r = u^r + \Delta$  if  $r = s$  and  $\tilde{u}^r = u^r$  if  $r \neq s$ . This yields  $f(\tilde{u})$  at a nearby point in  $\mathcal{K}$  with  $u^s$  displaced a quantity  $\Delta$  and the rest of unconstrained coordinates kept the same.
4. Calculate  $Z(\tilde{u}) := X(\tilde{u}, f(\tilde{u}))$ .
5. Calculate

$$\frac{\Delta Z}{\Delta u^s} := \frac{Z(\tilde{u}) - Z(u)}{\Delta} \quad (12)$$

as the finite-difference approximation to the sought derivative  $\partial Z/\partial u^s$  at the point  $u$ .

Note that the third point of this finite-differences approach is essentially a linear stability analysis. When strongly non-equilibrium points are present, such as in the examples discussed in the last section, this approximation fails and the fact that the new algorithm introduced in this work uses only quantities defined at the point  $u$  becomes even more important.

Now, assuming that we have a good enough guess for the parameter  $\Delta$ , the cost of this procedure is dominated by the need to perform  $K$  minimizations of the function  $V$ , one in each of the directions corresponding to the unconstrained coordinates  $u^r$ . If we denote by  $N_{it}$  the average number of iterations needed for these minimizations to converge, and we define  $C_V$  and  $C_{dV}$  as the numerical costs (in computer time) of computing  $V$  and its first derivatives with respect to the constrained coordinates  $d$ , respectively, we have that the average cost of calculating the sought derivatives  $\partial Z/\partial u^r$  using finite differences will be  $KN_{it}(C_V + C_{dV})$  for local optimization methods such as the steepest descent or the conjugate gradient, or  $KN'_{it}C_V$  for Monte Carlo-based methods in which the derivatives of  $V$  are not needed, such as simulated annealing [32].

On the other hand, the new algorithm does not require the extra minimizations, but it does require the calculation of the Hessian of  $V$  with respect to the internal coordinates (whose cost we call  $C_H$ ), and the computation of the inverse of its constrained sub-block,  $H_{JJ}$ , applied to each one of the  $K$   $L$ -vectors  $F_r$  in eqs. (10) and (11), of cost  $C_{iH}$ ; resulting in a total cost of  $C_H + KC_{iH}$ .

The comparison between the two costs is not trivial and some remarks about it must be made: First, one must notice that the different individual costs involved,  $C_V$ ,  $C_{dV}$ ,  $C_H$  and  $C_{iH}$ , are strongly dependent on the characteristics (1) of the coordinates  $q$  used and (2) of the function  $V(q)$ . For example, if the coordinates  $q$  are the Euclidean ones and the function  $V(q)$  is the potential energy of a molecular system as modeled by a typical force field [23–28], the most direct algorithms for calculating  $V$  and its derivatives yield costs  $C_V$ ,  $C_{dV}$  and  $C_H$  which are of order  $N^2$ ,  $NK$  and  $N^2$ , respectively [3]. However, if more advanced long-range techniques are used, such as the particle-particle particle-

mesh (PPPM) method [33], the fast multipole method [34] or the particle-mesh Ewald summation [35], these costs can be reduced to order  $N \log N$  or even  $N$  (for large  $N$  and forgetting prefactors). Also, as mentioned, if the coordinates used are not the Euclidean ones but some internal coordinates such as the ones used in this work, these costs must change in order to account for the transformation between the two. If force fields are not used but, instead,  $V(q)$  is the ground-state Born-Oppenheimer energy as calculated using Hartree-Fock [29], then the most naive implementations yield costs for  $C_V$ ,  $C_{dV}$  and  $C_H$  which are of order  $N^4$  [31]. The cost,  $C_{iH}$ , of calculating the inverse of  $H_{JJ}$  applied to a vector  $F_r$  can range from order  $N$  to order  $N^3$  depending on the sparsity of the matrix [32], which, in turn, depends again on the coordinates used and on the structure of  $V(q)$ . Finally, additional qualifications may complicate the comparison, such as the architecture of the computers in which the algorithms are implemented, parallelization issues, or the fact that, e.g., if we need the Hessian for a different purpose in our simulation, such as the calculation of the corresponding correcting term that appears both in the constrained stiff model and in the Fixman potential [12], then the ‘only’ computational step we are adding is the inversion of a matrix.

Despite the complexity and problem-dependence of the cost assessment, it must be stressed that, even in the cases in which the new algorithm turns out to be more expensive than the alternatives, the fact that it is exact and parameter-free might still make it the preferred choice in problems where high accuracy is needed. Although a parameter-free structure does not guarantee higher accuracy, in this case it does, since our method can be identified as the proper limit of the finite-differences scheme when  $\Delta \rightarrow 0$ . This is illustrated in Results and Discussion.

It is also worth remarking that the new method, as mentioned, is not needed to perform MD simulations, which can be run without calculating any of the derivatives tackled in this work [13–15]. Our method is only needed when some observable in which these derivatives are included (such as the aforementioned mass-metric tensor determinants) needs to be computed. In such cases, the only two options to get to the final result are either finite differences or our method, and the most convenient of the two has to be chosen; even if its cost is a burden.

## Application to Euclidean Coordinates of Molecules

In this section, we will apply the general algorithm introduced above to calculate the derivatives along the constrained subspace of the Euclidean coordinates of molecular systems in a frame of reference (FoR) fixed in the molecule. This problem has been faced by our group when trying to calculate the correcting terms associated with mass-metric tensor determinants that appear in the equilibrium probability density when flexible constraints are imposed [11,12,30]. More specifically, these derivatives are needed to calculate the determinant of the induced mass-metric tensor  $g$  that appears in the constrained rigid model, according to the formulae derived in ref. [30].

In such a case, the system of interest is a set of  $n$  mass points termed *atoms*. The three Euclidean coordinates of atom  $\alpha$  in a FoR fixed in the laboratory are denoted by  $\vec{x}_\alpha$ , and its mass by  $m_\alpha$ , with  $\alpha = 1, \dots, n$ . However, when no explicit mention to the atom index needs to be made, we will use  $x := (x^\mu)_{\mu=1}^N$  to denote the  $N$ -tuple of all the  $N := 3n$  Euclidean coordinates of the system. The masses  $N$ -tuple,  $m := (m_\mu)_{\mu=1}^N$ , in such a case, is formed by consecutive groups of three identical masses, corresponding to each of the atoms.

Apart from the Euclidean coordinates, one can also use a given set of *curvilinear coordinates* (also called sometimes *general* or

generalized), denoted by  $q := (q^\mu)_{\mu=1}^N$ , to describe the system. Both the coordinates  $x$  and  $q$  parameterize the whole space  $\mathcal{W}$ , and the transformation between the two sets and its inverse are respectively given by

$$x^\mu = X^\mu(q), \quad \mu = 1, \dots, N, \quad (13a)$$

$$q^\mu = Q^\mu(x), \quad \mu = 1, \dots, N. \quad (13b)$$

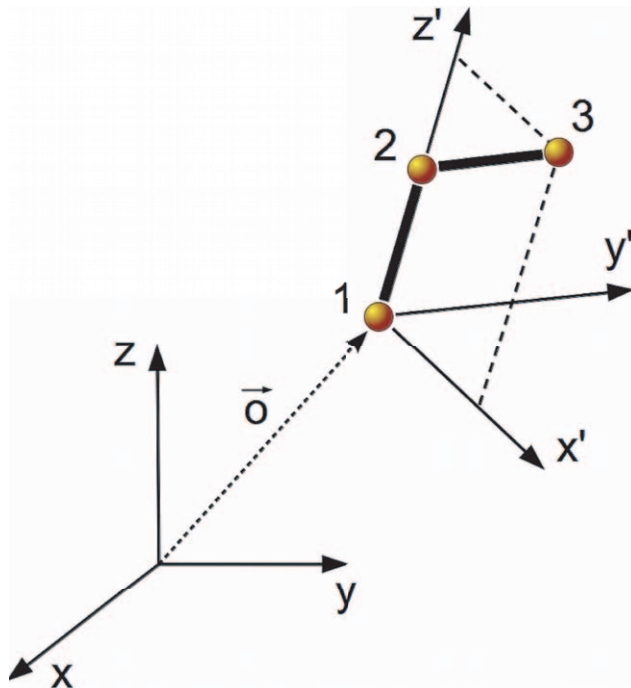
We will additionally assume that, for the points of interest, this is a *proper* change of coordinates, i.e., that the *Jacobian matrix*

$$J_v^\mu := \frac{\partial X^\mu(q)}{\partial q^v} \quad (14)$$

has non-zero determinant.

Now, we define a particular FoR *fixed in the system* to perform some of the calculations. To this end, we select three atoms (denoted by 1, 2 and 3) in such a way that  $\vec{o}$ , the position in the FoR of the laboratory of the origin of the FoR fixed in the system, is the Euclidean position of atom 1 (i.e.,  $\vec{o} := \vec{x}_1$ ). The orientation of the FoR ( $x', y', z'$ ) fixed in the system is chosen such that atom 2 lies in the positive half of the  $z'$ -axis, and atom 3 is contained in the  $(x', z')$ -plane, with projection on the positive half of the  $x'$ -axis (see fig. 1). The position of any given atom  $\alpha$  in the new FoR fixed in the system is denoted by  $\vec{x}_\alpha$ . Also, let  $E(\phi, \theta, \psi)$  be the Euler rotation matrix (in the ZYZ convention) that takes a free 3-vector of primed components,  $\vec{a}'$ , to the FoR fixed in the laboratory, i.e.,  $\vec{a} = E(\phi, \theta, \psi) \vec{a}'$  [21].

Although the aforementioned curvilinear coordinates  $q$  are a priori general, it is very common to take into account the fact that



**Figure 1. Definition of the frame of reference fixed in the system.**

doi:10.1371/journal.pone.0024563.g001

the typical potential energy functions of molecular systems in absence of external fields do not depend on  $\vec{\sigma}^T := (o_x, o_y, o_z)$  nor on the angles  $(\phi, \theta, \psi)$ , and to consequently choose a set of curvilinear coordinates split into  $q = (e, r)$ , where the first six are these *external coordinates*,  $e := (e^A)_{A=1}^6 = (o_x, o_y, o_z, \phi, \theta, \psi)$ . As we mentioned before,  $\vec{\sigma} := (o_x, o_y, o_z)$  describes the overall position of the system with respect to the FoR fixed in the laboratory, and its overall orientation is specified by the angles  $(\phi, \theta, \psi)$ . The remaining  $N-6$  coordinates  $r := (r^a)_{a=7}^N$  are called *internal coordinates* and determine the positions of the atoms in the FoR fixed in the system [36,37]. They parameterize what we shall call the *internal subspace* or *conformational space*, denoted by  $\mathcal{I}$ , and the coordinates  $e$  parameterize the *external subspace*, denoted by  $\mathcal{E}$ ; consequently splitting the whole space as  $\mathcal{W} = \mathcal{E} \times \mathcal{I}$  (denoting by  $\times$  the Cartesian product of sets).

The position,  $\vec{x}_\alpha$ , of any given atom  $\alpha$  in the axes fixed in the system is a function,  $\vec{X}'_\alpha(r)$ , of only the internal coordinates,  $r$ , and the transformation from the Euclidean coordinates  $x$  to the curvilinear coordinates  $q$  in (13) may be written more explicitly as follows:

$$\vec{x}_\alpha = \vec{X}'_\alpha(q) = \vec{\sigma} + E(\phi, \theta, \psi) \vec{X}'_\alpha(r). \quad (15)$$

Although general constraints affecting all the coordinates  $q$  [like those in (1)] can be imposed on the system, the already mentioned property of invariance of the potential energy function under changes of the external coordinates,  $e$ , together with the fact that the potential energy can be regarded as ‘producing’ the constraints [12], make physically frequent the use of constraints involving only the internal coordinates,  $r$ :

$$h^I(r) = 0, \quad I = K+1, \dots, N. \quad (16)$$

Under the common assumptions in the Introduction, these constraints allow us to split the internal coordinates as  $r = (s, d)$ , where the first  $M := K-6 = N-L-6$  ones,  $s := (s^i)_{i=6+1}^K$ , are called *unconstrained internal coordinates* and parameterize the *internal constrained subspace*, denoted by  $\Sigma$ . The last  $L := N-K$  ones,  $d := (d^I)_{I=K+1}^N$ , correspond to the *constrained coordinates* in the Introduction and are called. The external coordinates,  $e$ , together with the unconstrained internal coordinates,  $s$ , constitute the set of all *unconstrained coordinates* of the system,  $u = (e, s)$ , which parameterize the constrained subspace  $\mathcal{K}$ , being  $\mathcal{K} = \mathcal{E} \times \Sigma$ .

In this situation, the constraints in eq. (16) are equivalent to

$$d^I = f^I(s), \quad I = K+1, \dots, N, \quad (17)$$

and the functions  $f^I(s)$  are defined as taking the values of the coordinates  $d^I$  that minimize the potential energy with respect to all  $d^I$  at fixed  $s$  [11,12,15].

Finally, if these constraints are used, together with (16), the Euclidean position of any atom in the constrained case may be parameterized with the set of all unconstrained coordinates,  $u$ , as follows:

$$\begin{aligned} \vec{x}_\alpha &= \vec{Z}'_\alpha(u) := \vec{X}'_\alpha(e, s, f(s)) \\ &= \vec{\sigma} + E(\phi, \theta, \psi) \vec{X}'_\alpha(s, f(s)) \\ &=: \vec{\sigma} + E(\phi, \theta, \psi) \vec{Z}'_\alpha(s), \end{aligned} \quad (18)$$

where the name of the transformation functions has been changed from  $X$  to  $Z$ , and from  $X'$  to  $Z'$ , in order to emphasize that the dependence on the coordinates is different between the two cases.

In order to calculate the derivatives along  $\Sigma$  of the primed atoms positions,  $\vec{Z}'_\alpha$ , with respect to the unconstrained internal coordinates  $s$  (needed, for example, in eq. (28) of ref. [30] to compute the determinant of the induced mass-metric tensor  $g$ ), we first differentiate with respect to  $s^i$  in  $\vec{Z}'_\alpha(s) = \vec{X}'_\alpha(s, f(s))$ , arriving to the analogue to eq. (8):

$$\frac{\partial \vec{Z}'_\alpha}{\partial s^i}(s) = \frac{\partial \vec{X}'_\alpha}{\partial s^i}(s, f(s)) + \frac{\partial \vec{X}'_\alpha}{\partial d^I}(s, f(s)) \frac{\partial f^I}{\partial s^i}(s). \quad (19)$$

Now, the derivatives  $(\partial f^I / \partial s^i)(s)$  can be calculated using the general algorithm introduced in the previous section simply noticing that, in this case,  $V$  is precisely the potential energy of the system. Therefore, the only objects that remain to be computed are the derivatives  $\partial \vec{X}'_\alpha / \partial s^i$  and  $\partial \vec{X}'_\alpha / \partial d^I$ , which can be known analytically (they are *geometrical* [or *kinematical*] objects, i.e., they do not depend of the potential energy). We now turn to the derivation of an explicit algorithm for finding them and thus completing the calculation that is the objective of this section.

In the supplementary material of ref. [30], we give a detailed and explicit way for expressing any 'primed' vector  $\vec{X}'_\alpha$  as a function of all the internal coordinates, in the particular coordination scheme known as SASMIC [37]. We could take the final expression there [eq. (5)] and explicitly perform the partial derivatives, however, we shall follow a different approach that is both more straightforward and applicable to a larger family of Z-matrix-like schemes for defining internal coordinates.

In non-redundant internal coordinates schemes, whether they are defined as in ref. [37] or not, each atom is commonly regarded as being incrementally added to the growing molecule for its coordination. This means that the position of the  $(\alpha > 3)$ -th atom in the body-fixed axes is uniquely specified by the values of three internal coordinates that are defined with respect to the positions of three other atoms with indices  $\beta(\alpha), \delta(\alpha), \gamma(\alpha) < \alpha$ . This is a very convenient practice, and we will assume that we are dealing with a scheme that adheres to it.

Normally, the first of the three internal coordinates used to position atom  $\alpha$  is the length of the vector joining  $\alpha$  and  $\beta(\alpha)$ . Atom  $\beta(\alpha)$  is commonly chosen to be covalently attached to  $\alpha$  and, then, the length of this vector is naturally termed *bond length*, and denoted by  $b_\alpha$ . A given function  $\beta(\alpha)$  embodies the protocol used for defining this atom,  $\beta(\alpha)$ , to which each 'new' atom  $\alpha$  is (mathematically) attached; a superindex, as in  $\beta^p(\alpha)$ , indicates composition of functions, and the iteration of such compositions allows us to trace a single-branched chain of atoms that takes from atom  $\alpha$  to atom 1, at the origin of the 'primed' axes. If  $N_\alpha$  is a number such that  $\beta^{N_\alpha}(\alpha) = 1$ , this chain is given by the following set:

$$\mathcal{B}_\alpha := \{\alpha = \beta^0(\alpha), \beta(\alpha), \beta^2(\alpha), \dots, \beta^{N_\alpha-3}(\alpha), 3, 2, 1\}. \quad (20)$$

It is clear that, if we now change a given bond length  $b_\epsilon$  associated to atom  $\epsilon$ , atom  $\alpha$  will move if  $\epsilon \in \mathcal{B}_\alpha$ ; simply because atom  $\epsilon$  will move and  $\alpha$  has been positioned in reference to atom  $\epsilon$ 's position. Thus, if we define

$$\vec{R}_{\alpha\beta} := \vec{X}'_\beta - \vec{X}'_\alpha, \quad (21)$$

for any  $\alpha, \beta$ , and accordingly denote by  $\hat{R}_{\beta(\epsilon)\epsilon}$  the unitary vector in the 'primed' FoR that points from atom  $\beta(\epsilon)$  to atom  $\epsilon$ , a change in the bond length associated to  $\epsilon$  from  $b_\epsilon$  to  $b_\epsilon + db$  (while keeping the rest of the internal coordinates constant) will translate all atoms  $\alpha$  such that  $\epsilon \in \mathcal{B}_\alpha$  a distance  $db$  along  $\hat{R}_{\beta(\epsilon)\epsilon}$ , having

$$\vec{X}'_\alpha(b_\epsilon + db) = \vec{X}'_\alpha(b_\epsilon) + \hat{R}_{\beta(\epsilon)\epsilon} db, \quad (22)$$

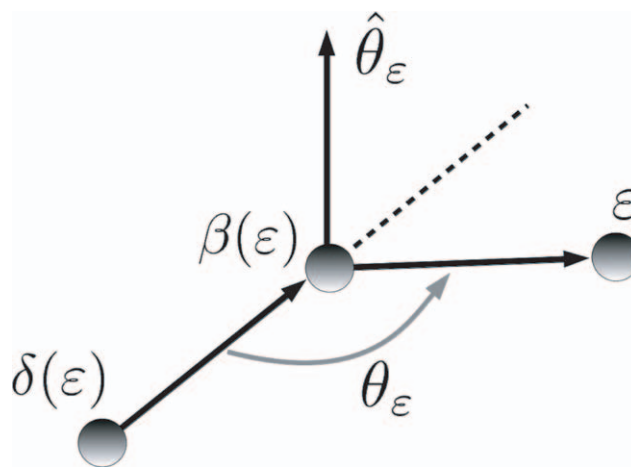
and hence

$$\frac{\partial \vec{X}'_\alpha}{\partial b_\epsilon} := \lim_{db \rightarrow 0} \frac{\vec{X}'_\alpha(b_\epsilon + db) - \vec{X}'_\alpha(b_\epsilon)}{db} = \hat{R}_{\beta(\epsilon)\epsilon}. \quad (23)$$

The second internal coordinate, after  $b_\alpha$ , that is typically defined to position atom  $\alpha$  with respect to the 'already positioned' part of the molecule is a so-called *bond angle*  $\theta_\alpha$ . To define this angle, we need an additional atom associated with  $\alpha$ , which we could denote by  $\delta(\alpha)$ . Although one can in principle think of the possibility of using different atoms  $\beta_\theta(\alpha)$  and  $\delta_\theta(\alpha)$  to define the bond angle than the one used to define the bond length, the common practice in the literature is to use the same three atoms  $\beta(\alpha)$ ,  $\delta(\alpha)$  and  $\gamma(\alpha)$ , to define the three internal coordinates associated to  $\alpha$ . This is also the choice in the SASMIC scheme and the one in this work. The angle  $\theta_\alpha$  is thus defined as  $180^\circ$  minus the angle formed between the vectors  $\vec{R}_{\delta(\alpha)\beta(\alpha)}$  and  $\vec{R}_{\beta(\alpha)\alpha}$  (see fig. 2).

Now, the reasoning is the same as in the case of the derivative with respect to  $b_\epsilon$ : For every atom  $\epsilon > 2$  that is the 'tip' of the bond angle  $\theta_\epsilon$ , the changes in this angle (keeping the rest of internal coordinates constant) will move atom  $\epsilon$  and therefore all atoms  $\alpha$  that contain atom  $\epsilon$  in the chain  $\mathcal{B}_\alpha$  that links them to atom 1.

If we now look at fig. 2, we see that a change from  $\theta_\epsilon$  to  $\theta_\epsilon + d\theta$  amounts to rotate all atoms  $\alpha$  that contain  $\epsilon$  in their chain to the origin an angle  $d\theta$  around the unitary vector  $\hat{\theta}_\epsilon$ , which is defined by



**Figure 2. Rotation associated to a change in a bond angle.** Definition of the *bond angle*  $\theta_\epsilon$ , associated to atom  $\epsilon$ , and the unitary vector  $\hat{\theta}_\epsilon$  corresponding to the direction around which all atoms  $\alpha$  with chains  $\mathcal{B}_\alpha$  containing  $\epsilon$  rotate if  $\theta_\epsilon$  is varied while the rest of internal coordinates are kept constant.

doi:10.1371/journal.pone.0024563.g002

$$\hat{\theta}_\varepsilon = \frac{\vec{R}_{\beta(\varepsilon)\varepsilon} \times \vec{R}_{\delta(\varepsilon)\beta(\varepsilon)}}{|\vec{R}_{\beta(\varepsilon)\varepsilon} \times \vec{R}_{\delta(\varepsilon)\beta(\varepsilon)}|}. \quad (24)$$

The result,  $\vec{v}_{\text{rot}}$ , of rotating a vector  $\vec{v}$  around the direction given by the unitary vector  $\hat{\theta}$  an amount  $\theta$  is given by the well-known *Rodrigues' rotation formula* [21,38]:

$$\vec{v}_{\text{rot}} = \vec{v} \cos \theta + (\hat{\theta} \times \vec{v}) \sin \theta + \hat{\theta}(\hat{\theta} \cdot \vec{v})(1 - \cos \theta). \quad (25)$$

However, notice that, in order to define a rotation, it is not enough to specify the angle  $\theta$  and the rotation axis  $\hat{\theta}$ , but one additionally needs to specify a fixed point (which can actually be any of the points in a fixed line in the direction of  $\hat{\theta}$ ). Therefore, the above expression is only correct for either 'free' vectors  $\vec{v}$  (i.e., those that are not associated to a given point in space), or for vectors  $\vec{v}$  whose starting point lies in the aforementioned fixed line.

The fixed point for the rotation we are interested in can be chosen to be  $\beta(\varepsilon)$  and, using eq. (25), we have that

$$\begin{aligned} \vec{R}_{\beta(\varepsilon)\alpha}(\theta_\varepsilon + d\theta) &= \vec{R}_{\beta(\varepsilon)\alpha}(\theta_\varepsilon) \cos d\theta \\ &+ [\hat{\theta}_\varepsilon \times \vec{R}_{\beta(\varepsilon)\alpha}(\theta_\varepsilon)] \sin d\theta \\ &+ \hat{\theta}_\varepsilon [\hat{\theta}_\varepsilon \cdot \vec{R}_{\beta(\varepsilon)\alpha}(\theta_\varepsilon)] (1 - \cos d\theta). \end{aligned} \quad (26)$$

Then, keeping the terms up to first order in  $d\theta$ , we can easily compute the derivative:

$$\frac{\partial \vec{R}_{\beta(\varepsilon)\alpha}}{\partial \theta_\varepsilon} = \hat{\theta}_\varepsilon \times \vec{R}_{\beta(\varepsilon)\alpha}, \quad (27)$$

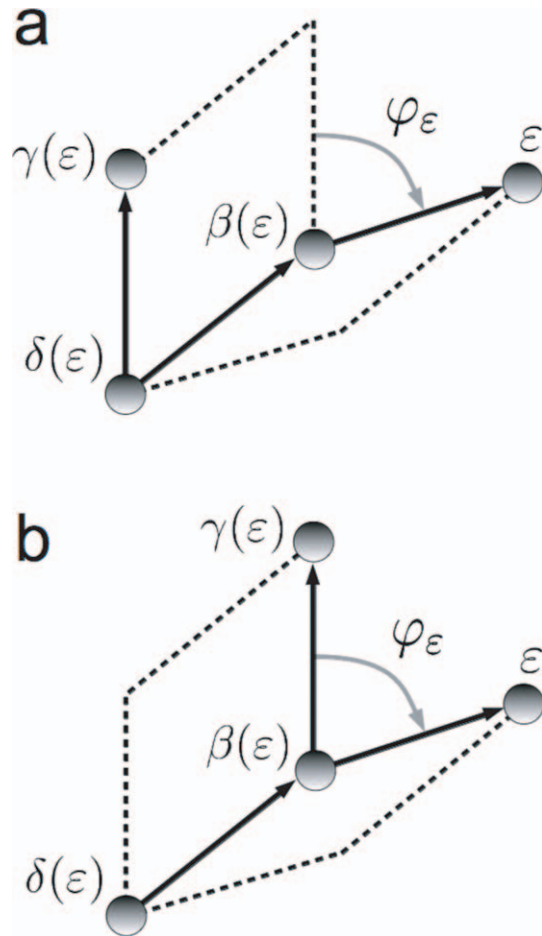
which, since a variation of  $\theta_\varepsilon$  does not move atom  $\beta(\varepsilon)$  (i.e.  $\partial \vec{X}'_{\beta(\varepsilon)}/\partial \theta_\varepsilon = 0$ ), allows us to conclude that

$$\frac{\partial \vec{X}'_\alpha}{\partial \theta_\varepsilon} = \frac{\partial}{\partial \theta_\varepsilon} (\vec{X}'_{\beta(\varepsilon)} + \vec{R}_{\beta(\varepsilon)\alpha}) = \frac{\partial \vec{R}_{\beta(\varepsilon)\alpha}}{\partial \theta_\varepsilon} = \hat{\theta}_\varepsilon \times \vec{R}_{\beta(\varepsilon)\alpha}, \quad (28)$$

if  $\varepsilon \in \mathcal{B}_\alpha$ .

The third and last internal coordinate that is usually defined to position atom  $\alpha$  is a so-called *dihedral angle*  $\vartheta_\alpha$ . To define this angle, we need a third atom associated with  $\alpha$ , which we could denote by  $\gamma(\alpha)$ . The angle  $\vartheta_\alpha$  is thus defined as the oriented angle formed between the plane containing atoms  $\beta(\alpha)$ ,  $\delta(\alpha)$  and  $\gamma(\alpha)$  and the plane containing atoms  $\alpha$ ,  $\beta(\alpha)$  and  $\delta(\alpha)$ . The positive sense of  $\vartheta_\alpha$  is the one indicated in fig. 3, and, although it is common to find two different covalent arrangements of the four atoms  $\alpha$ ,  $\beta(\alpha)$ ,  $\delta(\alpha)$  and  $\gamma(\alpha)$ , termed *principal* and *phase* dihedral angles, respectively [37], this does not affect the mathematical definition of  $\vartheta_\alpha$  given in this paragraph, nor the subsequent calculations.

Regarding the derivative of the 'primed' position of atom  $\alpha$  with respect to a given  $\vartheta_\varepsilon$ , the only difference with the bond angle case is that, now, the rotation is performed around the direction given by the unitary vector  $\hat{R}_{\delta(\varepsilon)\beta(\varepsilon)}$  (see fig. 3). The fixed point can be again chosen as  $\beta(\varepsilon)$ , and eq. (27) (changing  $\theta_\varepsilon$  by  $\vartheta_\varepsilon$  and  $\hat{\theta}_\varepsilon$  by  $\hat{R}_{\delta(\varepsilon)\beta(\varepsilon)}$ ), as well as the fact that changes in  $\vartheta_\varepsilon$  do not move atom  $\beta(\varepsilon)$ , still hold. Therefore,



**Figure 3. Rotation associated to a change in a dihedral angle.** Definition of the *dihedral angle*  $\vartheta_\varepsilon$ , associated to atom  $\varepsilon$ . The positive sense of rotation is indicated in the figure, and we can distinguish between two situations regarding covalent connectivity: **a**) *principal dihedral angle*, and **b**) *phase dihedral angle* (see ref. [37]). doi:10.1371/journal.pone.0024563.g003

$$\frac{\partial \vec{X}'_\alpha}{\partial \vartheta_\varepsilon} = \hat{R}_{\delta(\varepsilon)\beta(\varepsilon)} \times \vec{R}_{\beta(\varepsilon)\alpha}, \quad (29)$$

if  $\varepsilon \in \mathcal{B}_\alpha$ .

In order to decide whether or not atom  $\alpha$  will move upon changes in internal coordinates associated to atoms  $\varepsilon$  that *do not* belong to  $\mathcal{B}_\alpha$  we must first finish the story about internal coordinates definition. Since the argument above to show that  $\alpha$  moved when  $\varepsilon \in \mathcal{B}_\alpha$  was that  $\varepsilon$  *itself* moved and it was used to position  $\alpha$ , we must ask

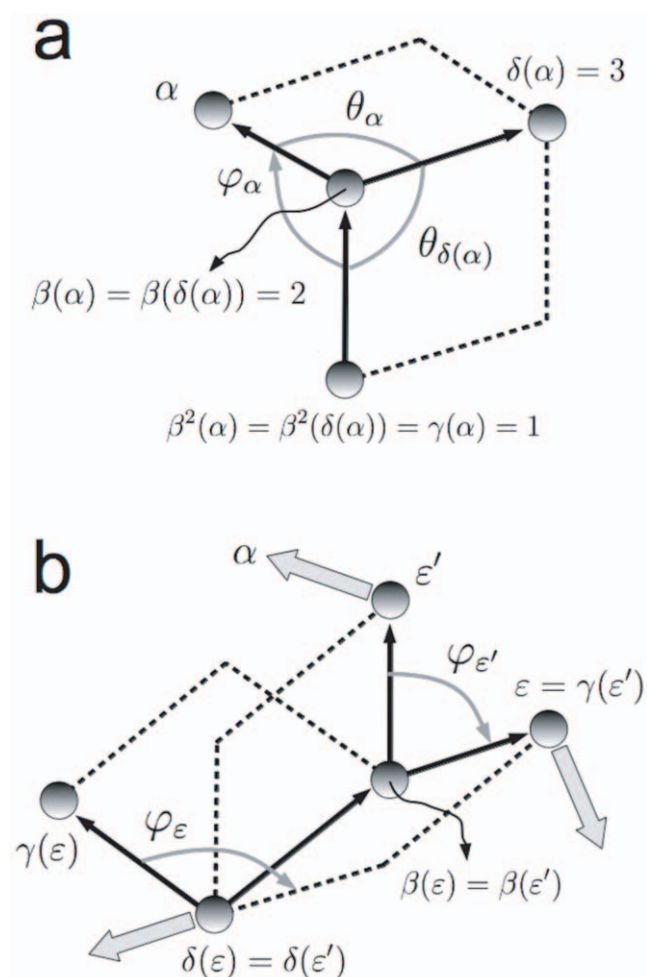
1. whether or not there can be atoms that are also used to position  $\alpha$  but that do not belong to  $\mathcal{B}_\alpha$ , and
2. what happens when we change the internal coordinates associated to them.

The answers to these two questions depend on the particular scheme used to define the internal coordinates, and we will tackle them referring to the SASMIC scheme [37], which is the one used in this work: According to the SASMIC rules, there are only two

situations in which an atom  $\varepsilon \notin \mathcal{B}_\alpha$  can be used to position atom  $\alpha$ , and they are depicted in fig. 4.

The first case, in fig. 4a, attains only the first atoms of the molecule. Typically, atom 1 is not a first-row atom, but a hydrogen (such is the case of the three molecules studied, for example, in Results and Discussion). Hence, after positioning atoms 2 and 3, which are typically first-row, it is more representative to choose atom 3 as  $\delta(\alpha)$  and atom 1 as  $\gamma(\alpha)$  when positioning the rest of the atoms  $\alpha$  attached to atom 2. This makes  $1 = \beta^2(\alpha) \neq \delta(\alpha)$  and hence  $\delta(\alpha)$  qualifies as an atom that is used to position  $\alpha$  but which is not included in the chain  $\mathcal{B}_\alpha$ .

The second case, in fig. 4b, corresponds to the situation in which the molecule divides in two branches, and it can happen all along its chemical structure. If atom  $\varepsilon$  is the atom that defines the only principal dihedral over the bond connecting  $\delta(\varepsilon)$  and  $\beta(\varepsilon)$  (in the SASMIC scheme, only one principal dihedral can be defined on a given bond [37]), and atom  $\alpha$  belongs to a different branch than the one beginning in  $\varepsilon$  (the branches are indicated with grey broad arrows), then the starting atom  $\varepsilon'$  of the branch to which  $\alpha$  belongs ( $\varepsilon'$  can be  $\alpha$  itself) must be positioned using a phase dihedral in which  $\varepsilon = \gamma(\varepsilon')$ . Thus,  $\varepsilon$  is an atom that is used to position  $\alpha$ , but which does not belong to the chain  $\mathcal{B}_\alpha$  connecting  $\alpha$  to atom 1.



**Figure 4. Special cases.** Special cases of atoms that do not belong to the chain  $\mathcal{B}_\alpha$  connecting  $\alpha$  to atom 1, but that are nevertheless used to position  $\alpha$ . doi:10.1371/journal.pone.0024563.g004

In principle, any change in the internal coordinates of atom  $\delta(\alpha)$ , in the first case, or in those of atom  $\varepsilon$ , in the second case, may move atom  $\alpha$ , however, due to the geometrical characteristics of the internal coordinates, this is not the case.

For example, it is easy to see that, in the case depicted in fig. 4a, a variation of the bond length  $b_\varepsilon$  (denoting  $\varepsilon := \delta(\alpha)$ ) does not move atom  $\alpha$ . Regarding the angles, the dihedral  $\vartheta_\varepsilon$  is not defined because  $\varepsilon = 3$ , and a change in  $\theta_\varepsilon$  can be seen to rotate atom  $\alpha$  with fixed point  $\beta(\varepsilon) = \beta(\alpha)$  and around the axis given exactly by  $\hat{\theta}_\varepsilon$  as defined in eq. (24). (It is not trivial to see that this motion keeps all the rest of internal coordinates constant, specially the phase dihedral  $\vartheta_\alpha$ . The authors found it helpful to imagine that atoms 1, 2 and 3 lie in the plane of the paper, with  $\hat{\theta}_\varepsilon$  and  $\alpha$  coming out of it towards the reader; the first orthogonally and the second not.) Therefore, the derivative of the Euclidean position of atom  $\alpha$  with respect to  $\theta_\varepsilon$  is also given by eq. (28) in this special case.

In the situation shown in fig. 4b, one can see that neither a change in  $b_\varepsilon$  nor in  $\theta_\varepsilon$  move atom  $\varepsilon'$  nor  $\alpha$ . However, if we change  $\vartheta_\varepsilon$ , we need to move atom  $\varepsilon'$  if we want to keep  $\vartheta_{\varepsilon'}$  constant. Therefore, atom  $\alpha$  moves in such a case and it does so by rotating with the same fixed point  $\beta(\varepsilon)$  and the same axis  $\hat{\mathbf{R}}_{\delta(\varepsilon)\beta(\varepsilon)}$  as in the simpler cases depicted in fig. 3. This means that, again, we can calculate the sought derivative using the already justified eq. (29).

In summary, only changes in bond lengths associated to atoms  $\varepsilon \in \mathcal{B}_\alpha$  affect the position of atom  $\alpha$ :

$$\frac{\partial \vec{X}'_\alpha}{\partial b_\varepsilon} = \begin{cases} \hat{\mathbf{R}}_{\beta(\varepsilon)\varepsilon} & \text{if } \varepsilon \in \mathcal{B}_\alpha \\ 0 & \text{if } \varepsilon \notin \mathcal{B}_\alpha \end{cases}; \quad (30)$$

changes both in bond angles associated to atoms  $\varepsilon \in \mathcal{B}_\alpha$  and to  $\delta(\alpha)$  in fig. 4a affect the position of atom  $\alpha$ :

$$\frac{\partial \vec{X}'_\alpha}{\partial \theta_\varepsilon} = \begin{cases} \hat{\theta}_\varepsilon \times \vec{\mathbf{R}}_{\beta(\varepsilon)\alpha} & \text{if } \varepsilon \in \mathcal{B}_\alpha, \\ & \text{or } [\beta(\alpha) = 2, \delta(\alpha) = 3] \\ 0 & \text{otherwise} \end{cases}; \quad (31)$$

and changes both in dihedral angles associated to atoms  $\varepsilon \in \mathcal{B}_\alpha$  and to those that define the principal dihedral at a branching point that leads to atom  $\alpha$  (see fig. 4b) can affect the position of atom  $\alpha$ :

$$\frac{\partial \vec{X}'_\alpha}{\partial \vartheta_\varepsilon} = \begin{cases} \hat{\mathbf{R}}_{\delta(\varepsilon)\beta(\varepsilon)} \times \vec{\mathbf{R}}_{\beta(\varepsilon)\alpha} & \text{if } \varepsilon \in \mathcal{B}_\alpha, \\ & \text{or } [\vartheta_\varepsilon \text{ ppal.}, \beta(\varepsilon) \in \mathcal{B}_\alpha] \\ 0 & \text{otherwise} \end{cases}. \quad (32)$$

Finally, the outline of the algorithm for calculating the sought derivatives  $\partial \vec{X}'_\alpha(s, f(s)) / \partial s^I$  along the constrained subspace  $\Sigma$  is:

1. Calculate the chain  $\mathcal{B}_\alpha$  that connects atom  $\alpha$  with atom 1 and identify the special cases depicted in fig. 4.
2. Calculate the derivatives  $\partial f^I / \partial s^I$  by solving the system of linear equations in (11).
3. Calculate the geometric derivatives  $\partial \vec{X}'_\alpha(r) / \partial s^I$  and  $\partial \vec{X}'_\alpha(r) / \partial d^I$ , for  $I = K + 1, \dots, N$ , using eqs. (30), (31) and (32).
4. Plug all the calculated quantities into eq. (19) et voilà.



## Results and Discussion

In this section, we compare the finite-differences approach (see Methods) to the new algorithm introduced in this work with two objectives in mind: the validation of the new scheme, and the identification of the most important pitfalls of the finite-differences technique, which are absent in the new method. It is worth stressing again that the method presented here is the first of its kind, as far as we are aware, and the finite-differences scheme is just a very natural and straightforward method that is always available when derivatives need to be calculated. In fact, the pitfalls of finite differences which we highlight in this section are very well known, although they have been seldomly presented in the context of molecular force fields. We hope that this section can be additionally useful to revisit this classical topic from a new angle.

To these two ends, we have applied the more specific algorithm introduced in the previous section for the calculation of the derivatives of the Euclidean coordinates of molecular systems to the three biological species in fig. 5: methanol, N-methyl-acetamide (abbreviated NMA), and the tripeptide N-acetyl-glycyl-glycyl-glycyl-amide (abbreviated GLY3). For each one of these molecules, a number of dihedral angles describing rotations around single bonds (and indicated with light-blue arrows in fig. 5) have been chosen as the unconstrained internal coordinates,  $s$ , spanning the corresponding constrained internal subspace  $\Sigma$ . The rest of internal coordinates  $d$  (bond lengths, bond angles, phase dihedrals, and principal dihedrals over non-single bonds) are flexibly constrained as described in the previous sections. The numeration of the atoms and the definition of the internal coordinates follow the SASMIC scheme, which is specially adapted to deal with constrained molecular systems [37].

For methanol and NMA, due to the small dimensionality of their constrained subspaces, the working sets of conformations have been generated by systematically scanning their unconstrained internal coordinates at finite steps. For methanol, we produced 19 conformations, in which the central dihedral,  $\vartheta_6$ , ranges from  $0^\circ$  to  $180^\circ$  in steps of  $10^\circ$ . Similarly, the systematic scanning of the unconstrained dihedrals in NMA produced a set of

588 conformations in which the first and last angles,  $\vartheta_6$  and  $\vartheta_{10}$ , range from  $0^\circ$  to  $180^\circ$ , and the central one,  $\vartheta_8$ , ranges from  $0^\circ$  to  $330^\circ$ , all in steps of  $30^\circ$ . For GLY3, and in view of the dimensionality of its constrained subspace, 1368 conformations were generated through a Monte Carlo with minimization procedure.

At each one of these conformations, defined by the value of the unconstrained internal coordinates  $s$ , the constrained coordinates  $d$  were found by minimizing the potential energy  $V(s,d)$  at fixed  $s$ , thus enforcing the constraints  $d=f(s)$  described in Methods. Let us remark that this fixing of the coordinates  $s$  is just an algorithmic way of sampling the constrained subspace defined by the relations  $d=f(s)$ , and it does not imply that the coordinates  $s$  are constrained; indeed, they could take any value in the set of conformations, whereas the constrained coordinates  $d$  are fixed by the aforementioned relations. The potential energy and force-field parameters were taken from the AMBER 96 parameterization [39,40], and local energy minimization with respect to the constrained coordinates was performed with Gaussian 03 [41]. At the minimized points, the Euclidean coordinates,  $\vec{Z}'_x(s) = \vec{X}'_x(s, f(s))$ , of all atoms in the system-fixed axes defined in the Methods section were also computed.

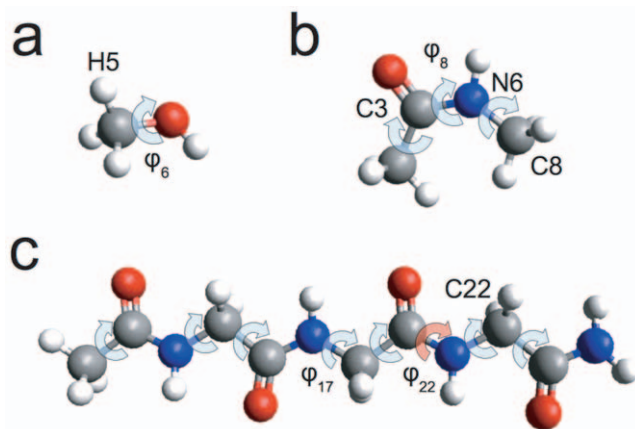
In order to find the partial derivatives  $\partial \vec{Z}'_x / \partial s^i$  at the generated points by finite differences, we produced, for each conformation in the working sets,  $M = K - 6$  additional conformations, each one with a single coordinate  $s^i$  displaced to  $s^i + \Delta$ . After the re-minimization of the constrained coordinates at the new points, we were in possession of all the data needed to compute the estimate of the sought derivative in eq. (12) for all unconstrained coordinates. In order to assess the behaviour and accuracy of the finite-differences approach, we performed these calculations for the values  $\Delta = 0.01^\circ, 0.05^\circ, 0.1^\circ, 0.5^\circ, 1.0^\circ, 5.0^\circ, 10.0^\circ$ .

On the other hand, to calculate the derivatives  $\partial \vec{Z}'_x / \partial s^i$  using the new scheme introduced in Methods, we do not need to perform any additional minimization, but we need to know the Hessian matrix of the second derivatives of  $V(s,d)$  with respect to the internal coordinates. The Hessian in internal coordinates was calculated with the Gaussian 03 package [41].

In order to compare the two methods, we turn first to the smallest system: methanol. In fig. 6a, we can see the value of the derivative  $\partial x_5 / \partial \vartheta_6$  of the  $x$ -coordinate (in the 'primed' axes, but we drop the prime from now on) of hydrogen number 5 (see fig. 5) with respect to the unconstrained dihedral angle  $\vartheta_6$  that describes the rotation of the alcohol group with respect to the methyl one. We can see that the agreement between the new algorithm and the finite-differences approach is good but not perfect, and that the discrepancy between the two is larger for the smallest ( $0.01^\circ$ ) and largest ( $10.0^\circ$ ) values of  $\Delta$  depicted in the graph.

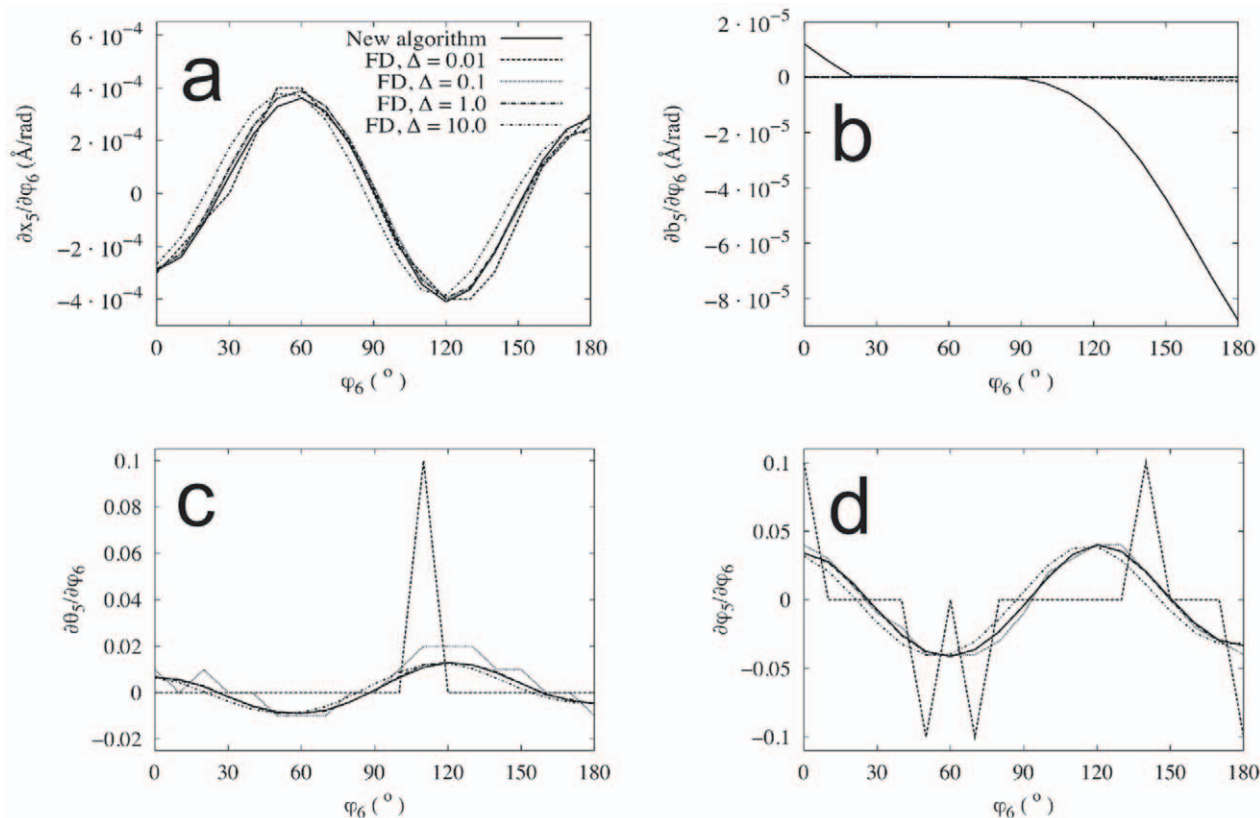
To track the source of this difference, we can take a look at eq. (8), which gives the derivative  $\partial \vec{Z}'_x / \partial s^i$  as a function of simple, 'geometrical' terms,  $\partial \vec{X}'_x / \partial s^i$  and  $\partial \vec{X}'_x / \partial d^l$ , and the numerical derivatives  $\partial f^l / \partial s^i$ . Of course, the choice of one method or another does not affect the former, but only the latter. In the particular case of  $\partial x_5 / \partial \vartheta_6$  in fig. 6a, if we remove the terms that are zero according to the rules in eqs. (30), (31) and (32), eq. (8) becomes

$$\begin{aligned} \frac{\partial x_5}{\partial \vartheta_6} &= \frac{\partial x_5}{\partial b_2} \frac{\partial b_2}{\partial \vartheta_6} + \frac{\partial x_5}{\partial b_3} \frac{\partial b_3}{\partial \vartheta_6} + \frac{\partial x_5}{\partial \theta_3} \frac{\partial \theta_3}{\partial \vartheta_6} \\ &+ \frac{\partial x_5}{\partial b_5} \frac{\partial b_5}{\partial \vartheta_6} + \frac{\partial x_5}{\partial \theta_5} \frac{\partial \theta_5}{\partial \vartheta_6} + \frac{\partial x_5}{\partial \vartheta_5} \frac{\partial \vartheta_5}{\partial \vartheta_6}. \end{aligned} \quad (33)$$



**Figure 5. Molecules used in the numerical calculations in this section.** (a) Methanol, (b) N-methyl-acetamide (abbreviated NMA), and (c) the tripeptide N-acetyl-glycyl-glycyl-glycyl-amide (abbreviated GLY3). Hydrogens are conventionally white, carbons are grey, nitrogens blue and oxygens red. The unconstrained dihedral angles that span the corresponding spaces  $\mathcal{K}$  are indicated with light-blue arrows, and some internal coordinates and some atoms that appear in the discussion are specifically labeled. The constrained dihedral angle  $\vartheta_{22}$  is indicated by a red arrow in GLY3.

doi:10.1371/journal.pone.0024563.g005



**Figure 6. Derivatives of some selected coordinates of methanol.** Derivatives of (a) the  $x$  coordinate of atom 5 in methanol, (b) the bond length  $b_5$  associated to it, (c) the bond angle  $\theta_5$ , and (d) the dihedral angle  $\vartheta_5$  as a function of the unconstrained coordinate  $\varphi_6$ . Both the results of the new algorithm and those obtained by finite differences (FD) are depicted. The key for the different types of line is the same in the four graphs. doi:10.1371/journal.pone.0024563.g006

The numerical derivatives appearing in this expression that are related to the three constrained coordinates associated to atom 5 are shown in figs. 6b, 6c and 6d, respectively, where we can see that the discrepancy between the new algorithm and the finite-differences approach is more significant. For the bond length  $b_5$  in fig. 6b, we see that the derivative predicted by finite differences is close to zero for all values of  $\varphi_6$  and for all the tested  $\Delta$ s, while the behaviour given by the new algorithm is more rich and substantially different. This large discrepancy is produced by the fact that bond lengths are very stiff coordinates in the energy function that we have used here, together with the default precision of the floating point numbers provided by Gaussian 03 outputs. In table 1, we can see indeed that the last significant figure of bond length  $b_5$  only starts to change for  $\Delta = 5.0^\circ$ , which makes any algorithm based on finite differences very unreliable for this particular quantity if small values of  $\Delta$  are used. The bond angles and dihedral angles, on the other hand, are somewhat less stiff than bond lengths, as it can also be seen in tab. 1. This makes their derivatives by finite differences more reliable, as one can observe in figs. 6c and 6d, where the discrepancy with the new method is apparent for small  $\Delta$ , but becomes gradually smaller as we increase it. Of course, since, in the new method presented in this work, all quantities are computed at the non-displaced point  $\Delta = 0^\circ$ , the problem regarding the number of significant figures does not appear. It is also worth remarking that, in the case of finite differences, the point in which this issue will appear depends on the number of bits used to represent coordinates, but it will always appear for some small enough value of  $\Delta$ .

As we noticed in fig. 6a, also in the case of the constrained internal coordinates the difference between the two methods starts to grow again when  $\Delta$  reaches  $5.0^\circ$  or  $10.0^\circ$ . This is easily understood if we think that only in the  $\Delta \rightarrow 0$  limit the estimate in eq. (12) converges to the actual value of the partial derivative. In fact, as the complexity of the system increases, the error introduced at large  $\Delta$  may come not only from continuous changes in the location of the constrained minima, but also, as

**Table 1. Stiffness of the constrained coordinates in methanol.**

| $\Delta$ ( $^\circ$ ) | $b_5$ ( $\text{\AA}$ ) | $\theta_5$ ( $^\circ$ ) | $\vartheta_5$ ( $^\circ$ ) |
|-----------------------|------------------------|-------------------------|----------------------------|
| 0.0                   | 1.090694               | 109.403                 | 119.296                    |
| 0.01                  | 1.090694               | 109.404                 | 119.296                    |
| 0.05                  | 1.090694               | 109.404                 | 119.297                    |
| 0.1                   | 1.090694               | 109.405                 | 119.299                    |
| 0.5                   | 1.090694               | 109.409                 | 119.312                    |
| 1.0                   | 1.090694               | 109.415                 | 119.329                    |
| 5.0                   | 1.090693               | 109.462                 | 119.474                    |
| 10.0                  | 1.090692               | 109.525                 | 119.671                    |

Values of the constrained coordinates associated to atom 5 of methanol for different displacements  $\Delta$  in the unconstrained coordinate  $\varphi_6$ . The values correspond to the conformation with  $\varphi_6 = 110^\circ$ , and the number of significant figures presented is the default one provided by Gaussian 03. doi:10.1371/journal.pone.0024563.t001

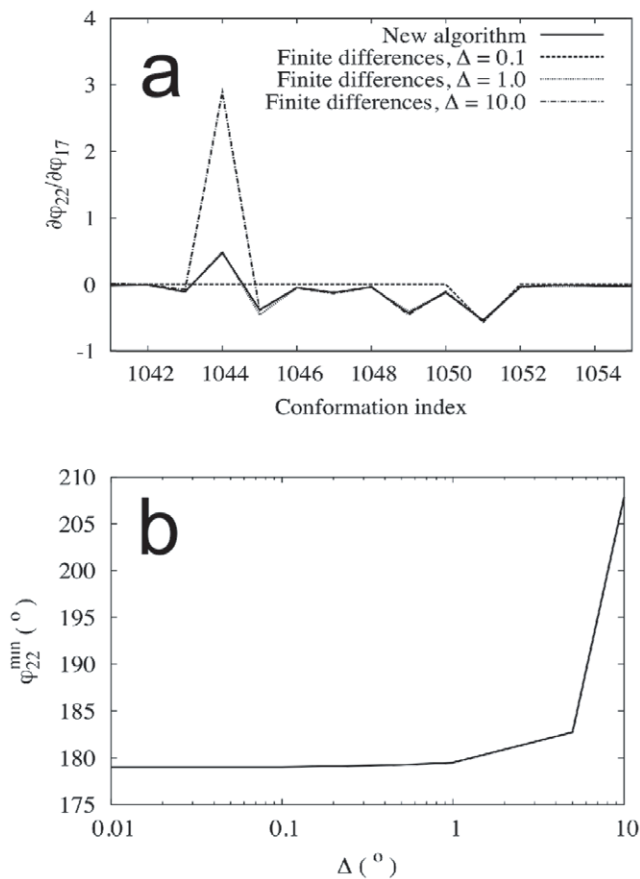
fig. 7 suggests, it may occur that, at a certain value of  $\Delta$ , the very *identity* of the minima is altered, thus introducing potentially larger errors. In fig. 7a, we can see that the derivative  $\partial\vartheta_{22}/\partial\vartheta_{17}$  in GLY3 presents an unusually large error at the conformation 1044. In fig. 7b, we see that the minimum-energy value of  $\vartheta_{22}$ , which is the dihedral angle associated to carbon 22, describing the rotation around a given peptide bond (see fig. 5c), presents an abrupt change when  $\Delta$  reaches  $10^\circ$ . If we think that the energy landscape of GLY3 is indeed a complex and multidimensional one, it is not difficult to imagine that, as we change  $\vartheta_{17}$ , i.e., as we increase  $\Delta$ , the energy landscape is so altered that some minima disappear, some other appear, and the energy ordering among them is changed. In such a case, the structures found by the minimization procedure will be rather different between, say,  $\Delta=0^\circ$  and  $\Delta=10^\circ$ , thus producing a large error in the derivatives calculated by finite differences. Again, the new algorithm, which only uses quantities calculated at  $\Delta=0^\circ$ , does not suffer from this drawback.

To sum up, the finite-differences method contains two sources of error which the new method does not present: one at small values of  $\Delta$ , related to the finite precision of the floating point numbers representing the internal coordinates, and the other at larger values of  $\Delta$ , stemming from the very definition of the partial derivative by finite differences, and aggravated by the complexity of the energy

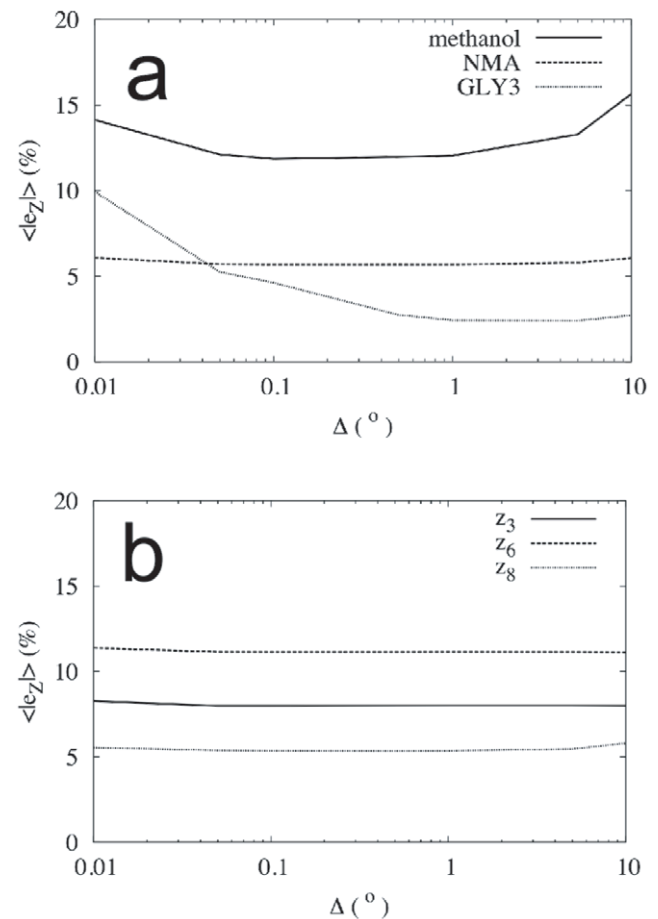
landscapes of large systems. If the derivatives are to be calculated using finite differences, an optimal value of  $\Delta$  must be chosen in each case so that the possible error is minimized. However, already in the simple example of methanol, we saw that the derivatives of different observables, in the same system, may behave differently as we change  $\Delta$  (compare the bond length derivative in fig. 6b with that of the angles in figs. 6c and 6d). In fig. 8, we additionally see that the search for the optimal  $\Delta$  may be further complicated by the fact that the behaviour found also depends (strongly) on the system studied, and, in the case of the derivatives of the Euclidean coordinates, on the position of the atom in the molecule.

In fig. 8a, we have plotted the normalized average of the absolute value of the error in the derivatives of the Euclidean coordinates,  $\langle|e_Z|\rangle$ , as a function of  $\Delta$  for the three molecular systems studied. This quantity is defined, for a given unconstrained coordinate  $s^i$ , as

$$\langle|e_Z|\rangle(\Delta) := \frac{100}{N_c(3n-6)} \times \sum_{m=1}^{N_c} \sum_{\mu=1}^{3n} \frac{1}{\delta_i^\mu} \left| \left( \frac{\partial Z^\mu}{\partial s^i} \right)_{\text{FD}}(\Delta) - \left( \frac{\partial Z^\mu}{\partial s^i} \right)_{\text{NA}} \right|, \quad (34)$$



**Figure 7. Metastability of the local minima in GLY4.** (a) Derivative  $\partial\vartheta_{22}/\partial\vartheta_{17}$  of the constrained dihedral angle  $\vartheta_{22}$ , describing a peptide bond rotation in GLY3, with respect to the unconstrained coordinate  $\vartheta_{17}$  for a selected set of conformations in the working set. (b) Minimum-energy value of the constrained dihedral angle  $\vartheta_{22}$  in the conformation 1044 of GLY3 for different values of the displacement  $\Delta$  in the unconstrained coordinate  $\vartheta_{17}$ . doi:10.1371/journal.pone.0024563.g007



**Figure 8. Dependence of the error as a function of  $\Delta$ .** Average normalized error in the derivatives by finite differences as a function of  $\Delta$  (see the text for a more precise definition). (a) Error averaged to all conformations and all atoms of the three molecular systems studied. (b) Error averaged to all conformations of the  $z$ -coordinate of three particular 1<sup>st</sup>-row atoms in NMA. doi:10.1371/journal.pone.0024563.g008

where the index  $m$  indicates the conformation in the working set, running from 1 to  $N_c$ , FD stands for ‘finite differences’, NA for ‘new algorithm’, and  $\delta_i^\mu$  is a normalizing quantity for each coordinate  $x^\mu$  chosen as

$$\delta_i^\mu := \max_m \left( \frac{\partial Z^\mu}{\partial s^i} \right)_{\text{NA}} - \min_m \left( \frac{\partial Z^\mu}{\partial s^i} \right)_{\text{NA}}. \quad (35)$$

The graphics in fig. 8a of this quantity correspond to the unconstrained dihedral angles  $\vartheta_6$ ,  $\vartheta_8$  and  $\vartheta_{17}$  of methanol, NMA and GLY3, respectively (see fig. 5). We observe that the average error as a function of  $\Delta$  presents significantly different behaviours in the three molecules, never being smaller than a 2%. Additionally, in fig. 8b, we show the same error but this time individualized to the  $z$ -coordinate of three different 1<sup>st</sup>-row atoms of NMA: C3, N6 and C8. Although the overall behaviour of the error is similar for the three atoms, its size is not.

All in all, we see that the need to tune for the optimal  $\Delta$  in the finite-differences approach not only produces unavoidable errors, but also it must be done in a per-system, per-observable basis, clearly complicating and limiting the use of this technique. The new algorithm, on the other hand, is only affected by the source of error related to the accuracy with which the Hessian matrix of the potential energy can be calculated and inverted; apart from this, which is a general drawback of any method implemented in a computer, its mathematical definition is ‘exact’, in the sense that it does not contain any tunable parameter, like  $\Delta$ , that must be adjusted for optimal accuracy in each particular problem.

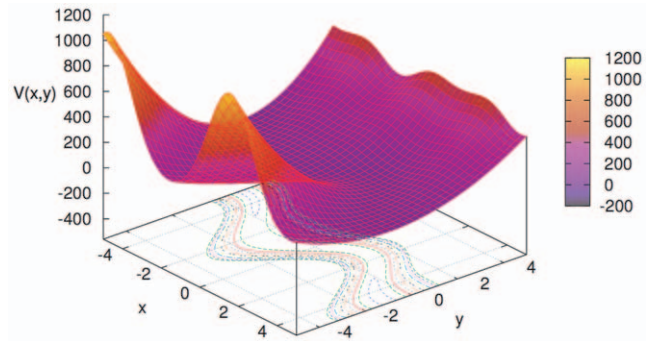
Also, and more importantly (since the failure of finite differences was indeed predictable) the good coincidence between the newly introduced, somewhat more involved method and the straightforward finite-differences scheme for the smallest system and in some intermediate range of values of  $\Delta$  allows us to regard the new scheme as validated and error-free.

Finally, despite what we discussed in the Methods section, namely, that we have not pursued here the numerical optimization of the algorithm introduced, being our main interest to present the general theoretical concepts and to show that the new method is exact and reliable, we close this section with an example of a toy system to provide a clue that the new technique is at least feasible. Before introducing the toy system it is worth noting that the examples tackled in this section are just particular cases, but the technique can be used in different systems and with different potential energy functions. When looking at the computer costs presented below, the reader should bear in mind that they may be not very significant (due to the aforementioned lack of optimization) and not very relevant (due to the choice of a small toy system and a given potential energy function). Of course, if any production runs using the new algorithm are attempted, a thorough numerical optimization and assessment should be performed, which we deem to be a very important next step of our work.

The toy system is a 2-dimensional one, with positions  $x$  and  $y$ , and the following potential energy (see fig. 9):

$$V(x,y) = \frac{1}{2} K(2 + \sin x)(y - \sin x)^2 + \tanh(xy). \quad (36)$$

If we take a large enough  $K$ , say  $K = 20$ , we see that the system will present a strong oscillatory motion in the  $y$  coordinate, around approximately  $y = \sin x$  (but not exactly, since the term  $\tanh(xy)$



**Figure 9. Potential energy of the toy system in eq. (36).** The range of  $x$  and  $y$  corresponds to the one explored in this work. Contour level lines and colour level indication in the surface have been added for visual comfort. All units are arbitrary. doi:10.1371/journal.pone.0024563.g009

slightly modifies the position of the minimum), and with harmonic constant approximately equal to  $K(2 + \sin x)$ . In the spirit of this work, since, due to energetic reasons, the value of  $y$  will seldomly move far away from the value that minimizes  $V(x,y)$  for each  $x$ , denoted by  $f(x)$  and implicitly defined by the following equation:

$$\frac{\partial V}{\partial y}(x, f(x)) = K(2 + \sin x)(f(x) - \sin x) + \frac{x}{\cosh^2(xf(x))} = 0, \quad (37)$$

we can kill this oscillatory motion by assuming that a flexible constraint  $y = f(x)$  exists. In such a case,  $x$  plays the role of the whole set of unconstrained coordinates  $s$  in the general formalism, and  $y$  plays the role of the whole set of constrained coordinates  $d$ .

Now, if we perform a ‘molecular dynamics’ of this system, then we may need at some point to compute the derivative with respect to  $x$  of some observable  $X(x,y)$  restricted to the constrained subspace  $Z(x) := X(x, f(x))$  (for example, we may need this to calculate mass-metric tensor corrections at each time step [12]). We can do so by using finite differences or the new technique introduced in this work. As we discussed in the Methods section, for both approaches we will need to perform a minimization of  $V(x,y)$  at each fixed  $x$  in order to find  $f(x)$  and  $Z(x) := X(x, f(x))$ , hence, being this step common, we will not consider it for the assessment of the differences in computational cost between the two methods. The additional computations that will decide which method is faster are:

- **For finite differences:** Choose a displaced value of the unconstrained coordinate  $x + \Delta x$ , minimize  $V(x + \Delta x, y)$  with respect to  $y$  in order to find  $f(x + \Delta x)$ , as well as  $X(x + \Delta x, f(x + \Delta x))$ , and finally calculate the finite-differences estimation of the sought derivative:

$$\frac{\partial Z}{\partial x}(x) \simeq \frac{X(x + \Delta x, f(x + \Delta x)) - X(x, f(x))}{\Delta x}. \quad (38)$$

- **For the new method:** Calculate the objects in eq. (11), perform the required inversion to find  $\partial f / \partial x$ , calculate the objects in eq. (5), and finally find  $\partial Z / \partial x$  using this last expression. In this simple case, all the objects to be computed are:

$$\begin{aligned} \frac{\partial^2 V}{\partial y^2}(x, f(x)), \quad \frac{\partial^2 V}{\partial x \partial y}(x, f(x)), \\ \frac{\partial X}{\partial x}(x, f(x)), \quad \text{and} \quad \frac{\partial X}{\partial y}(x, f(x)). \end{aligned} \quad (39)$$

The second-order derivatives of the potential energy can be easily calculated:

$$\frac{\partial^2 V}{\partial y^2}(x, y) = K(2 + \sin x) - \frac{2x^2 \tanh(xy)}{\cosh^2(xy)}, \quad (40a)$$

$$\frac{\partial^2 V}{\partial x \partial y} = K \cos x(y - 2[1 + \sin x]) + \frac{1 - 2xy \tanh(xy)}{\cosh^2(xy)}, \quad (40b)$$

and we can use them to find the derivative of  $f(x)$  through eq. (11):

$$\frac{\partial f}{\partial x}(x) = - \left[ \frac{\partial^2 V}{\partial x \partial y}(x, f(x)) \right] / \left[ \frac{\partial^2 V}{\partial y^2}(x, f(x)) \right]. \quad (41)$$

The particularization of eq. (5) to this simple case is

$$\frac{\partial Z}{\partial x}(x) = \frac{\partial X}{\partial x}(x, f(x)) + \frac{\partial X}{\partial y}(x, f(x)) \frac{\partial f}{\partial x}(x). \quad (42)$$

In this section, for illustrative purposes, we have chosen a simple observable  $X(x, y)$ :

$$X(x, y) = \sqrt{x^2 + y^2}, \quad (43)$$

i.e., the distance of the particle to the origin of coordinates. Hence, the remaining objects that we need to compute in order to apply the new technique to this problem are

$$\frac{\partial X}{\partial x}(x, y) = \frac{x}{\sqrt{x^2 + y^2}}, \quad (44a)$$

$$\frac{\partial X}{\partial y}(x, y) = \frac{y}{\sqrt{x^2 + y^2}}. \quad (44b)$$

We have calculated  $\partial Z/\partial x$  using both techniques for 11 different values of  $x = -5.0, -4.0, \dots, 4.0, 5.0$ . This calculation has been performed in a desktop iMac with a 2.66 GHz Intel Core 2 Duo processor and 4GB of 1067 MHz DD3 RAM memory, running MacOSX Snow Leopard. The same compilation-time optimizations have been used in the two cases, and the common times have been subtracted as indicated before. It is also worth remarking that we have used Brent's method [32] for minimizing

## References

- Rapaport DC (2004) The art of molecular dynamics simulation. Cambridge University Press, 2nd edition.
- Allen MP, Tildesley DJ (2005) Computer simulation of liquids. Oxford: Clarendon Press.
- Frenkel D, Smit B (2002) Understanding molecular simulations: From algorithms to applications. Orlando FL: Academic Press, 2nd edition.
- Car R, Parrinello M (1985) Unified approach for molecular dynamics and density-functional theory. Phys Rev Lett 55: 2471–2474.

$V(x, y)$ , and a different choice will change the comparison. In these conditions, the new technique has proved approximately one order of magnitude faster than finite differences, elapsing 0.13  $\mu\text{s}$ /point vs. 1.08  $\mu\text{s}$ /point.

In summary, in this work, we have introduced a new, exact, parameter-free method for computing the derivatives of physical observables in systems with flexible constraints. The new algorithm has been numerically validated in small molecules against its most natural alternative, finite differences. In doing so, numerous pitfalls of the latter method have been demonstrated, all arising from the fact that it contains a tunable parameter that has to be optimally adjusted in each particular problem at hand. In a number of numerical experiments, we have shown that the finite-differences approach contains two unavoidable sources of error that are not present in the new method: On the one hand, the finite number of significant figures used to represent, in computers, the values of the optimized coordinates, together with the fact that these constrained coordinates are typically very stiff, make the changes in this quantities often unobservable or at least badly resolved, thus rendering the finite-differences derivatives unreliable for small values of the displacement parameter  $\Delta$ . On the other hand, the very fact that finite-differences derivatives only converge to the true ones for  $\Delta \rightarrow 0$ , complicated with the possibility that the energy landscapes of complex molecular systems may significantly change their structure when the unconstrained coordinates are displaced, introduce new errors as  $\Delta$  increases. These two sources of errors combined make compulsory the search of an optimal value of  $\Delta$  in each particular case, and also establish a minimum error below which is not possible to go, as it can be seen in fig. 8. Also, using a simple toy system, we have shown that the new technique can be faster than finite differences in certain situations. The new method introduced here, and it is already being successfully used in a number of works in progress in our group to compute the correcting terms appearing in the equilibrium probability distribution when flexible constraints are imposed on the system [11]. Moreover, given the almost ubiquitous occurrence of the concept of constraints all throughout the fields of computational physics and chemistry, it is expected that the method described in this work will find many applications in present and future problems. Some examples have been already mentioned in the introduction, notably the case of ground-state Born-Oppenheimer MD [42,43] (using, e.g., Hartree-Fock [29]), which can be regarded as a flexibly constrained problem in which the soft coordinates are the nuclear positions  $R$ , the hard ones are the electronic orbitals  $\psi$ , and the function to be minimized is the expected value  $\langle \psi | \hat{H}_e(R) | \psi \rangle$  of the  $R$ -dependent electronic Hamiltonian in the  $N$ -electron Slater determinant  $\psi$ .

## Acknowledgments

We thank the reviewers of the manuscript for their insightful comments that have contributed to improve the final version.

## Author Contributions

Conceived and designed the experiments: PE CNC MDM PGR JLA. Performed the experiments: PE CNC MDM. Analyzed the data: PE CNC MDM. Contributed reagents/materials/analysis tools: PE CNC. Wrote the paper: PE CNC MDM PGR JLA.

5. Hutter J, Curioni A (2005) Car-Parrinello molecular dynamics on massively parallel computers. *ChemPhysChem* 6: 1788–1793.
6. Carter EA, Ciccotti G, Hynes JT, Kapral R (1989) Constrained reaction coordinate dynamics for the simulation of rare events. *Chem Phys Lett* 5: 472–477.
7. Schlick T, Barth E, Mandziuk M (1997) Biomolecular dynamics at long timesteps: Bridging the timescale gap between simulation and experimentation. *Annu Rev Biophys Biomol Struct* 26: 181–222.
8. Ryckaert JP, Ciccotti G, Berendsen HJC (1977) Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes. *J Comput Phys* 23: 327–341.
9. Dubrovin BA, Fomenko AT, Novikov SP (1992) *Modern Geometry — Methods and Applications*. Berlin: Springer.
10. Weisstein EW (last accessed on 03/17/09) Ordinary differential equations. from MathWorld – A Wolfram Web Resource. <http://mathworld.wolfram.com/OrdinaryDifferentialEquations.html>.
11. Echenique P, Cavasotto CN, García-Risueño P (2011) The canonical equilibrium of constrained molecular models. In progress, <http://arxiv.org/abs/1105.0374>.
12. Echenique P, Calvo I, Alonso JL (2006) Quantum mechanical calculation of the effects of stiff and rigid constraints in the conformational equilibrium of the alanine dipeptide. *J Comput Chem* 27: 1748–1755.
13. Christen M, van Gunsteren WF (2005) An approximate but fast method to impose flexible distance constraints in molecular dynamics simulations. *J Chem Phys* 122: 144106.
14. Hess B, Saint-Martin H, Berendsen HJC (2002) Flexible constraints: An adiabatic treatment of quantum degrees of freedom, with application to the flexible and polarizable mobile charge densities in harmonic oscillators model for water. *J Chem Phys* 116: 9602.
15. Zhou J, Reich S, Brooks BR (2000) Elastic molecular dynamics with self-consistent flexible constraints. *J Chem Phys* 112: 7919.
16. Christen M, Christ CD, van Gunsteren WF (2007) Free energy calculations using flexible-constrained, hard-constrained and non-constrained molecular dynamics simulations. *ChemPhysChem* 8: 1557–1564.
17. Cotter CJ, Reich S (2004) Adiabatic invariance and applications from molecular dynamics to numerical weather prediction. *BIT Num Math* 44: 439–455.
18. Helfand E (1979) Flexible vs. rigid constraints in Statistical Mechanics. *J Chem Phys* 71: 5000.
19. Pechukas P (1980) Comment on: ‘Flexible vs. rigid constraints in Statistical Mechanics’. *J Chem Phys* 72: 6320.
20. Van Kampen NG, Lodder JJ (1984) Constraints. *Am J Phys* 52: 419–424.
21. Goldstein H, Poole C, Safko J (2002) *Classical Mechanics*. Addison-Wesley, 3rd edition.
22. Gō N, Scheraga HA (1976) On the use of classical statistical mechanics in the treatment of polymer chain conformation. *Macromolecules* 9: 535.
23. Brooks BR, Brooks III CL, MacKerell AD, Nilsson L, Petrella RJ, et al. (2009) CHARMM: The biomolecular simulation program. *J Comput Chem* 30: 1545–1615.
24. Jorgensen WL, Tirado-Rives J (1988) The OPLS potential functions for proteins. Energy minimization for crystals of cyclic peptides and Crambin. *J Am Chem Soc* 110: 1657–1666.
25. Jorgensen WL, Maxwell DS, Tirado-Rives J (1996) Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *J Am Chem Soc* 118: 11225–11236.
26. Ponder JW, Case DA (2003) Force fields for protein simulations. *Adv Prot Chem* 66: 27–85.
27. Case DA, Darden TA, Cheatham III TE, Simmerling CL, Wang J, et al. (2008) Amber 10. San Francisco: University of California.
28. Pearlman DA, Case DA, Caldwell JW, Ross WR, Cheatham TE, III, et al. (1995) AMBER, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules. *Comp Phys Commun* 91: 1–41.
29. Echenique P, Alonso JL (2007) A mathematical and computational review of Hartree-Fock SCF methods in Quantum Chemistry. *Mol Phys* 105: 3057–3098.
30. Echenique P, Calvo I (2006) Explicit factorization of external coordinates in constrained Statistical Mechanics models. *J Comput Chem* 27: 1733–1747.
31. Jensen F (1998) *Introduction to Computational Chemistry*. Chichester: John Wiley & Sons.
32. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007) *Numerical recipes. The art of scientific computing*. New York: Cambridge University Press, 3rd edition.
33. Eastwood JW, Hockney RW (1974) Shaping the force law in two-dimensional particle mesh models. *J Comput Phys* 16: 342–359.
34. Greengard L, Rokhlin V (1987) A fast algorithm for particle simulations. *J Comput Phys* 73: 325–348.
35. Darden TA, York D, Pedersen L (1993) Particle mesh Ewald: An N log(N) method for ewald sums in large systems. *J Chem Phys* 98: 10089–10092.
36. Echenique P (2007) *Introduction to protein folding for physicists*. *Contemp Phys* 48: 81–108.
37. Echenique P, Alonso JL (2006) Definition of Systematic, Approximately Separable and Modular Internal Coordinates (SASMIC) for macromolecular simulation. *J Comput Chem* 27: 1076–1087.
38. Belongie S (last accessed on 08/12/2009). Rodrigues’ rotation formula. from MathWorld – A Wolfram Web Resource. <http://mathworld.wolfram.com/ImplicitFunctionTheorem.html>.
39. Cornell WD, Cieplak P, Bayly CI, Gould IR, Merz J K M, et al. (1995) A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J Am Chem Soc* 117: 5179–5197.
40. Kollman P, Dixon R, Cornell W, Fox T, Chipot C, et al. (1997) The development/application of a ‘minimalist’ organic/biochemical molecular mechanic force field using a combination of ab initio calculations and experimental data. In: Van Gunsteren WF, Weiner PK, Wilkinson AJ, editors, *Computer Simulations of Biomolecular Systems*, Dordrecht: Kluwer Academic Publishing, volume 3, pp 83–96.
41. Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, et al. (2007) Gaussian 03, Revision E.01. Gaussian, Inc., Wallingford, CT.
42. Alonso JL, Andrade X, Echenique P, Falseto F, Prada-Gracia D, et al. (2008) Efficient formalism for large-scale ab initio molecular dynamics based on time-dependent density functional theory. *Phys Rev Lett* 101: 096403.
43. Andrade X, Castro A, Zucro D, Alonso JL, Echenique P, et al. (2009) Modified Ehrenfest formalism for efficient large-scale ab initio molecular dynamics. *J Chem Theory Comput* 5: 728–742.