An efficient and general library for the definition and use of internal coordinates in large molecular systems

Abschlussarbeit im Rahmen der Prüfungen
zum Master-Studiengang Chemie
der Georg-August-Universität Göttingen

Vorgelegt am 8. November 2017
von Oskar Weser
aus Neustadt a. d. Weinstraße

Erstgutachter: Prof. Dr. Ricardo Mata
Zweitgutachter: Prof. Dr. Jörg Behler

Hiermit versichere ich, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Ort, Datum	Unterschrift

Acknowledgements

I thank Prof. Dr. Ricardo Mata for supervising this work and Prof. Dr. Jörg Behler for judging it. The scientific discussions with Thorsten Stolper (Uni Göttingen) regarding optimization procedures were very helpful. Dr. Rainer Oswald (Uni Göttingen) helped with feedback on code quality and several implementation details. Florian Lauster (Uni Göttingen) gave important mathematical input for group theory related problems. Markus Napierkowski (TU Dresden) helped in the discussion of algorithm scaling. Niels Karschin (Uni Göttingen) helped to improve the work with general chemical discussions.

The following GitHub users contributed to the development of `chemcoord` with pull requests. Dr. Keld Lundgard (then Stanford University) implemented Python 2.7 compatibility for `chemcoord` which supported only Python 3.x before. sgill2 (Unknown institution) added functionality to the xyz-file parser. Lars Berscheid (TU Braunschweig) corrected spelling errors in the documentation.

Already mentioned Thorsten Stolper made an important code contribution to `chemopt` for a chemically tailored BFGS implementation. Because of non smoothness of the gradient discussed in section 3.1 a generic BFGS optimization procedure had to be used in the end.

I thank the proof readers Johanne Scharnick, Prof. Dr. Ulrich Weser, Dr. Daniel Mierwaldt, and Niels Karschin.

In the end I thank my flat mate Tomke Schoss for constantly supplying me with food in the last week of writing.

Contents

1. Introduction

In the field of theoretical chemistry there is a long history of coordinate systems, practices, and preferences to describe molecular structures. All of them have certain advantages for different applications. Cartesian coordinates are probably the easiest to understand and most common coordinate system for saving and distributing structures. On the other hand they have significant drawbacks if structures should be distorted using chemical knowledge. The intuitive way to think of molecules is to think in terms of bond lengths, angles, and dihedrals and to separate larger molecules into certain chemically similar substructures. This way of thinking is not represented in cartesian coordinates but in internal coordinates, of which Z-matrices are a special case. If, for example, a substituent should be bent in respect to a larger molecule in cartesian coordinates, it is necessary to find indices of atoms to be rotated and manually apply a rotation matrix. In a properly constructed Z-matrix the angle bending the whole substituent appears naturally as one coordinate, which greatly simplifies exploration of the potential hyper surface (PHS) of a molecule. There is no clear definition on when a Z-matrix is properly defined, so it may require much manual work beforehand to construct one with the desired properties. It is important to note that there are certain corner cases like linearity that may render a Z-matrix undefined.

If it is the aim to optimize the geometry of a molecule, internal coordinates usually outperform cartesian coordinates, because expressing the PHS in terms of bond lengths, angles, and dihedrals yields a less coupled optimization problem. Z-matrices are the coordinate system of choice for non-linear constrained optimizations, while other internal coordinate systems usually outperform Z-matrices in the case of unconstrained optimizations.

In this work the Python library `chemcoord` was implemented to facilitate construction of a chemically useful Z-matrix and provide functions for the preparation of structures in the field of theoretical chemistry. With `chemcoord` Z-matrices can be generated automatically and the aforementioned corner cases caused by linearity are reliably dealt with. The Jacobian matrices for the conversion between cartesian and Z-matrix coordinates were derived analytically and implemented. In addition to the reliable conversion between Z-

matrices and cartesian coordinates many other features were implemented. An incomplete list includes the Kabsch algorithm for molecule alignment, point group detection and iterative symmetrisation of distorted molecules, as well as support of symbolic expressions in both cartesian coordinates and Z-matrices.

Based on `chemcoord` the library `chemopt` for non-linearly constrained geometry optimization in Z-matrix coordinates was implemented. The unconstrained optimization in Z-matrix coordinates was compared with *redundant internal coordinates* and examples for non-linear constrained optimizations using Z-matrices are given.

The code examples given in this work can be executed with the setup as described in section A.2. The documentations are hosted under `http://chemcoord.readthedocs.io/` and `http://chemopt.readthedocs.io/`. The source code is hosted under `https://github.com/mcocdawc/chemcoord` and `https://github.com/mcocdawc/chemopt`. Further examples for the capabilities of `chemcoord` are given in the tutorial (`http://chemcoord.readthedocs.io/en/v2.0.3/tutorial.html`).

2. The ChemCoord library

2.1. Introduction of Z-matrix coordinates

2.1.1. The Z-matrix

All internal coordinate systems are based on the so called *internal primitives*, which is the set of all bond lengths, planar bends, and proper torsions that can be generated based on the atomic connectivity. The difference between the internal coordinate systems comes from the different selection of subsets and/or linear combinations of these primitives. This allows the important classification into either redundant or non redundant internal coordinates. The redundant internal coordinate systems use more coordinates than the $3n - 6(5)$ internal degrees of freedom in a molecule, which implies an overspecified transformation to cartesian coordinates (n the number of atoms).

Z-matrix coordinates are a non-redundant internal coordinate system that uses a subset of internal primitives *without* forming linear combinations. With the exception of the first three atoms, each atom i at the position $\mathbf{v}_i$ in cartesian coordinates uses three previously defined atoms as reference. The label of the bond defining atom is denoted with b, the angle defining atom with a, and the dihedral defining atom with d. Using these notations the bond length r, the angle α and the dihedral δ are then given by:

$$\begin{aligned} r &= \|\mathbf{v}_i - \mathbf{v}_b\| \\ \alpha &= \arccos\left(\frac{\langle \mathbf{v}_i - \mathbf{v}_b | \mathbf{v}_a - \mathbf{v}_b \rangle}{\|\mathbf{v}_i - \mathbf{v}_b\| \|\mathbf{v}_a - \mathbf{v}_b\|}\right) \\ \delta &= -\operatorname{sgn}(\langle \mathbf{n}(\mathbf{v}_i, \mathbf{v}_b, \mathbf{v}_a) \times \mathbf{n}(\mathbf{v}_b, \mathbf{v}_a, \mathbf{v}_d) | \mathbf{v}_a - \mathbf{v}_b \rangle) \arccos\left(\langle \mathbf{n}(\mathbf{v}_i, \mathbf{v}_b, \mathbf{v}_a) | \mathbf{n}(\mathbf{v}_b, \mathbf{v}_a, \mathbf{v}_d) \rangle\right) \end{aligned} \tag{2.1.1}$$

$\mathbf{n}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is the normalised normal vector on the plane given by the points $\mathbf{x}, \mathbf{y}$, and $\mathbf{z}$.

$$\mathbf{n}(\mathbf{x}, \mathbf{y}, \mathbf{z}) := \frac{(\mathbf{y} - \mathbf{x}) \times (\mathbf{z} - \mathbf{y})}{\|(\mathbf{y} - \mathbf{x}) \times (\mathbf{z} - \mathbf{y})\|} \tag{2.1.2}$$

The definition of rotation directions follows the IUPAC standard.[1] An illustration of

the definitions is given in Figure 2.1.

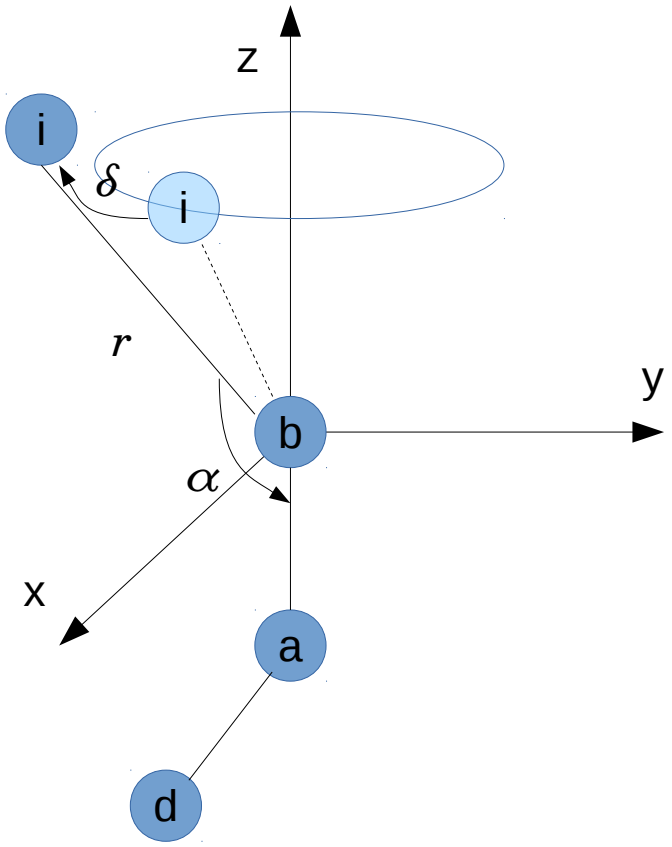


Figure 2.1.: Illustration of the bond length r, angle α and dihedral δ definition.

The example Z-matrix of methane (Table 2.1) illustrates how the first three atoms are usually specified. Corresponding values for r, α and δ are simply omitted. This implies that the three translational and three rotational degrees of freedom do not appear in the Z-matrix and the barycenter's position and alignment of the principal axes of inertia depend on convention. Usually the second atom is inserted along the z-axis from the first atom, which is at the origin. This convention also explains the name Z-matrix.

In `chemcoord` an approach similar to Hoft et al. was implemented to store the information about absolute positions in a Z-matrix.[2] In the first three rows the origin and canonical unitvectors can be treated as if they were reference atoms, to specify translational and rotational degrees of freedom. These fixed points in cartesian space that are used to specify the absolute position will be called absolute references. This method works for all molecules, but does couplee translation and rotation of the molecule. An example Z-matrix with specified absolute position can be seen in Table 2.2.

Table 2.1.: Example Z-matrix of methane with unspecified translational and rotational degrees of freedom. Bond lengths in Å, angles in degrees.

	atom	b	bond	a	angle	d	dihedral
2	C						
1	H	2	1.087				
3	H	2	1.087	1	109.479		
4	H	2	1.087	1	109.472	3	-120.000
5	H	2	1.087	1	109.474	3	120.002

Table 2.2.: Example Z-matrix of methane with fixed translational and rotational degrees of freedom. Bond lengths in Å, angles in degrees.

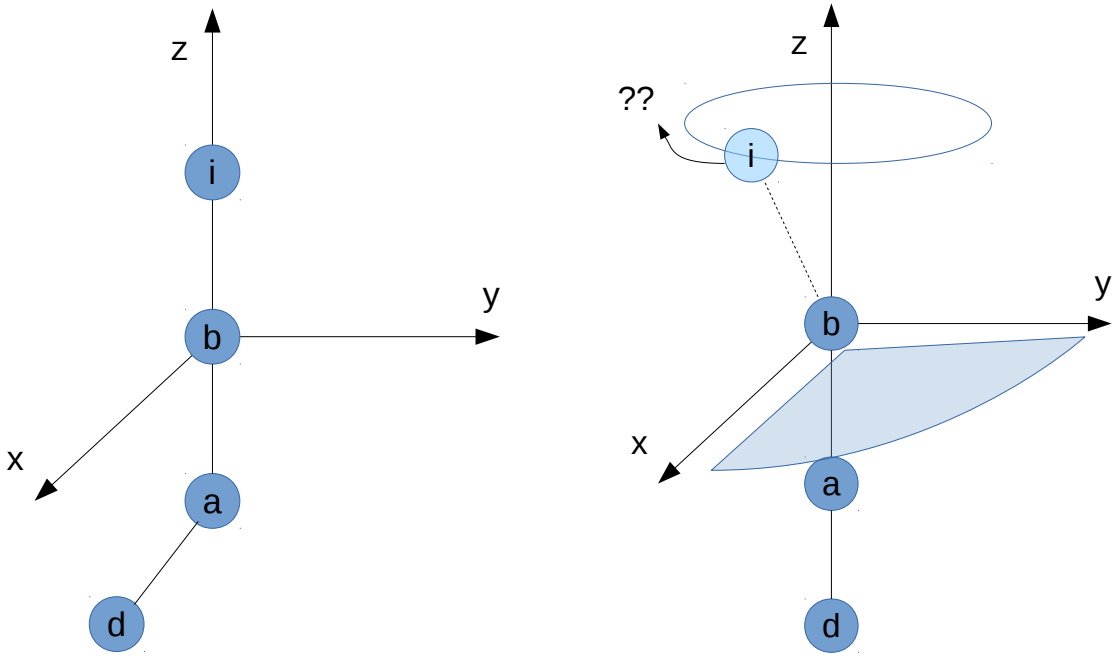
	atom	b	bond	a	angle	d	dihedral
2	C	$\vec{0}$	0.000	$\vec{e_z}$	0.000	$\vec{e_x}$	0.000
1	H	2	1.087	$\vec{e_z}$	30.567	$\vec{e_x}$	16.934
3	H	2	1.087	1	109.479	$\vec{e_x}$	64.784
4	H	2	1.087	1	109.472	3	-120.000
5	H	2	1.087	1	109.474	3	120.002

Another possibility to keep absolute positions in a Z-matrix is to specify the molecule's barycenter in the first row using spherical coordinates. The remaining three angles may then be used as Euler angles to specify how the principal axes of inertia are aligned. This would allow the rotation of a molecule along principal axes of inertia by just assigning an angle. The main drawback is that the rotation is undefined, if there are degenerate inertia moments. Even if this method is implemented only for asymmetric rotors, they can become symmetric rotors if the internal coordinates change. In order to keep a consistent conversion to cartesian coordinates this method was not implemented.

2.1.2. Conversion to cartesian coordinates and corner cases

The conversion of a Z-matrix to cartesian coordinates happens iteratively starting from the first row, because the i-th atom can only be placed if the reference positions $\mathbf{v}_b$, $\mathbf{v}_a$, and $\mathbf{v}_d$ are defined. The position of the i-th atom is then given by inverting equation 2.1.1. Before formulating exact equations, we will qualitatively look unto certain corner cases which are best illustrated by Figure 2.2.

The undefined dihedral in Figure 2.2a appears as definition problem when converting from cartesian to Z-matrix coordinates. If the angle α equals 180° there are infinitely many possible values for the dihedral, which is a direct consequence of the fact that there



(a) Undefined dihedral. **(b)** Undefined local coordinate system.

Figure 2.2.: Schematic drawing of the two corner cases.

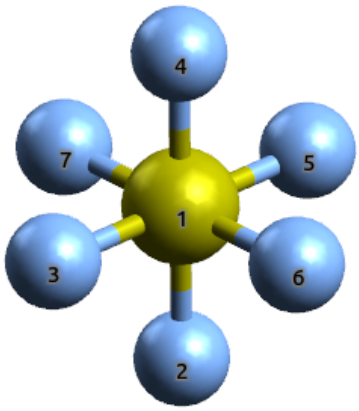
exists no continuous (hence no differentiable) invertible mapping between $\mathbb{R}^2$ and the two dimensional unit sphere. But since energy and all other quantities of interest can be formulated solely using cartesian coordinates, it does not matter which dihedral is used, so it is set to $0°$ in this case. When an angle α of $180°$ is encountered upon converting from Z-matrix to cartesian coordinates, the atom i is simply set to

$$\mathbf{v}_i = \mathbf{v}_b + r\frac{\mathbf{v}_b - \mathbf{v}_a}{\|\mathbf{v}_b - \mathbf{v}_a\|}$$

For the undefined local coordinate system in Figure 2.2b it is advantageous to adopt the view of the *Natural Extension Reference Frame* as described by Parsons et al.[3] This algorithm is used for the conversion from Z-matrix to cartesian coordinates and was introduced to reduce the number of floating point operations. Apart from its better performance it offers a more elegant mathematical description of the conversion operations. Using spherical coordinates, the atom i is put into the canonical reference frame. This frame is then rotated and translated into the orthonormalised, righthanded reference frame which is spanned locally by the three atoms b, a, and d. The conversion to cartesian coordinates is now elegantly factored, because the first positioning into the canonical reference frame is a function of only r, α, and δ and the affine-linear transformation into

the local reference frame is a function of only $\mathbf{v}_b$, $\mathbf{v}_a$, and $\mathbf{v}_d$. Now it is immediately obvious that the undefined coordinate system in Figure 2.2b appears independently of r, α, and δ depending solely on the position of the reference atoms in cartesian coordinates. It is not possible to test for this pathological case without converting the Z-matrix to cartesian coordinates. A Z-matrix that cannot be converted to cartesian coordinates will be referred to as undefined or invalid Z-matrix.

An undefined Z-matrix may easily appear, as it is demonstrated in the following example using the octahedral sulfur hexafluoride displayed in Figure 2.3a. In the same Figure it may be seen that any atom that uses the atoms 1, 2, and 4 as references suffers from an undefined reference frame. The example Z-matrix (Table 2.3b) of Sulfur hexafluoride is only defined up to the fluorine atom with label four. The flourine atoms 5, 6, and 7 have no defined positions due to an invalid reference frame. It will be explained in section 2.3 how a Z-matrix is created and kept defined.



(a) Structure of Sulfur hexafluoride.

(b) Undefined Z-matrix of Sulfur hexafluoride. Bond lengths in Å, angles in degrees.

	atom	b	bond	a	angle	d	dihedral
1	S	$\vec{0}$	1.000	$\vec{e_z}$	90.000	$\vec{e_x}$	90.000
2	F	1	1.561	$\vec{e_z}$	90.000	$\vec{e_x}$	90.000
3	F	1	1.561	2	90.000	$\vec{e_x}$	90.000
4	F	1	1.561	2	180.000	3	-90.000
5	F	1	1.561	2	90.000	4	NaN
6	F	1	1.561	2	90.000	4	NaN
7	F	1	1.561	2	90.000	4	NaN

Figure 2.3.: Structure and Z-matrix of Sulfur hexafluoride.

It has to be emphasized that due to floating point arithmetic the aforementioned corner cases have to be caught already before reaching the exact value where they are mathematically undefined. Throughout this project the canonical numpy functions (as for example `numpy.isclose`) for dealing with floating point arithmetic were used for this purpose.[4]

2.2. Mathematical equations for the coordinate transformation

In this section the exact equations for the transformation between Z-matrix and cartesian coordinates and their gradients are derived. It will be assumed that the corner case of an undefined coordinate system as discussed in subsection 2.1.2 and shown in Figure 2.2a does not appear. Due to mathematical elegance and superior perfomance the *Self-Normalizing Natural Extension Reference Frame* (SN-NeRF) algorithm by Parsons et al. is used.[3] This means that the "intuitive" equations 2.1.1 are not used. The SN-NeRF algorithm works conceptionally similar to the *Natural Extension Reference Frame* described in subsection 2.1.2 but does so without applying mathematically redundant normalisation steps.

The section will heavily rely on tensor formulation. This might seem hyperbolic for the transformation functions in the subsections 2.2.2 and 2.2.4, but it is necessary for deriving the gradients in subsections 2.2.3 and 2.2.5. An advantage of the tensor formulation lies in the similarity between derived equations and library source code which heavily relies on vectorized calls.

2.2.1. Notation

The following typographic rules for the notation of variables will be used:

Scalars will be denoted with italic letters:	x
Tensors of order one (vectors) will be denoted with lowercase, bold letters:	$\mathbf{x}$
Tensors of order higher than one will be denoted with uppercase, bold letters:	$\mathbf{X}$

The typography of functions depends on the tensorial order of their image.
In the case of tensors the colon notation of array slices as in FORTRAN is used. So the i-th column vector of the matrix $\mathbf{X}$ may be written as $\mathbf{X}_{:,i}$ and the j-th row vector may be written as $\mathbf{X}_{j,:}$.
The index notation for deriving tensors in respect to tensors follows a simple rule: The swapped indices of the denominator are appended to the indices of the numerator:

$$\left(\frac{\partial \mathbf{Y}}{\partial \mathbf{X}}\right)_{i,j,k,l} = \frac{\partial \mathbf{Y}_{i,j}}{\partial \mathbf{X}_{l,k}} \tag{2.2.1}$$

The swapping of indices in the denominator allows an easier formulation of calculus in higher dimensions because sum contraction takes place if neighbouring indices use the

same letter. Applying the above rules allows for example the succinct notation of the gradient of a potential energy function V_C (e. g. spherical coordinates) using the potential energy function V_X expressed in another coordinate system (e. g. cartesian coordinates).

$$\begin{aligned} V_X &: \mathbb{R}^3 \to \mathbb{R} \\ \mathbf{t} &: \mathbb{R}^3 \to \mathbb{R}^3 \\ V_C &:= V_X \circ \mathbf{t} \end{aligned}$$

$$\begin{aligned} V_C(\mathbf{c}_0 + \mathbf{h}) &\approx V_C(\mathbf{c}_0) + \frac{\partial V_C}{\partial \mathbf{c}}(\mathbf{c}_0) \cdot \mathbf{h} \qquad \text{For } \|\mathbf{h}\| \text{ small} \\ &= \underbrace{V_X(\mathbf{t}(\mathbf{c}_0))}_{\text{scalar}} + \underbrace{\frac{\partial V_X}{\partial \mathbf{x}}(\mathbf{t}(\mathbf{c}_0))}_{\text{row-vector}} \cdot \underbrace{\frac{\partial \mathbf{t}}{\partial \mathbf{c}}(\mathbf{c}_0)}_{3\times 3\text{-matrix}} \cdot \underbrace{\mathbf{h}}_{\text{column-vector}} \\ &= V_X(\mathbf{t}(\mathbf{c}_0)) + \sum_{i=1}^{3}\sum_{j=1}^{3} \left(\frac{\partial V_X}{\partial \mathbf{x}}(\mathbf{t}(\mathbf{c}_0))\right)_{1,i} \cdot \left(\frac{\partial \mathbf{t}}{\partial \mathbf{c}}(\mathbf{c}_0)\right)_{i,j} \cdot \mathbf{h}_j \end{aligned} \tag{2.2.2}$$

Another important convention is the layout of coordinates. The row wise alignment of XYZ files makes sense for these CSV like files. But in the following section it is mathematically advantageous to use a column wise alignment of coordinates. This implies that the cartesian positions and Z-matrix entries of a molecule with n atoms are an element of $\mathbb{R}^{3,n}$.

2.2.2. Transformation from Z-matrix to cartesian coordinates

Let n be the number of atoms in a molecule. Let $I := [1...n]$ be the index set of the molecule, and $I^R := [-2...n]$ the index set of the molecule with three indices of absolute references. Then the following functions are defined:

$$\begin{aligned} b &: I \to I^R \\ &\quad i \mapsto \text{The bond defining atom of } i \\ a &: I \to I^R \\ &\quad i \mapsto \text{The angle defining atom of } i \\ d &: I \to I^R \\ &\quad i \mapsto \text{The dihedral defining atom of } i \end{aligned} \tag{2.2.3}$$

For the previously defined functions, the following relationship holds true:

$$i > b(i), a(i), d(i) \tag{2.2.4}$$

$\mathbf{C} \in ([0,\infty) \times [0,\pi) \times [0,2\pi))^n$ is defined to be the matrix of all bond lengths r, angles α and dihedrals δ:

$$\mathbf{C}_{:,i} = (r_i, \alpha_i, \delta_i)^\mathsf{T}$$

The distance $r_i = \mathbf{C}_{1,i}$ is the distance between atom i and $b(i)$, the angle $\alpha_i = \mathbf{C}_{2,i}$ is the angle between atom i, $b(i)$, and $a(i)$ and the dihedral $\delta_i = \mathbf{C}_{3,i}$ is the dihedral between atom i, $b(i)$, $a(i)$, and $d(i)$. Let $\mathbf{X} \in \mathbb{R}^{3,n}$ be the matrix of all atom positions. Let $\mathbf{X}^R \in \mathbb{R}^{n+3,3}$ be the matrix of all atom positions plus the three absolute references. (This means that the row index of $\mathbf{X}^R$ starts at -2.) So $\mathbf{X}_{:,i}$ and $\mathbf{X}^R_{:,i}$ are the position of the i-th atom. The position of the i-th atom ($i \in I$) is given by:

$$\begin{aligned}\mathbf{X}_{:,i}\colon [0,\infty) \times [0,\pi) \times [0,2\pi) \times \mathbb{R}^{3,3} &\to \mathbb{R}^3 \\ (\mathbf{C}_{:,i}, \mathbf{X}^R_{:,(b(i),a(i),d(i))}) &\mapsto \mathbf{X}_{:,i}(\mathbf{C}_{:,i}, \mathbf{X}^R_{:,(b(i),a(i),d(i))})\end{aligned}$$

Let $\mathbf{B}$ be the third rank tensor, which contains the orthonormal bases spanned by the references of the i-th atom. So $\mathbf{B}_{:,:,i}$ is the orthonormalized basis spanned by $\mathbf{X}^R_{:,(b(i),a(i),d(i))}$.

$$\begin{aligned}\mathbf{B}\colon \mathbb{R}^{(n+3),3} &\to \mathbb{R}^{n,3,3} \\ \mathbf{ba}_i &:= \frac{\mathbf{X}^R_{:,a(i)} - \mathbf{X}^R_{:,b(i)}}{\left\|\mathbf{X}^R_{:,a(i)} - \mathbf{X}^R_{:,b(i)}\right\|} \\ \mathbf{ad}_i &:= \frac{\mathbf{X}^R_{:,d(i)} - \mathbf{X}^R_{:,a(i)}}{\left\|\mathbf{X}^R_{:,d(i)} - \mathbf{X}^R_{:,a(i)}\right\|} \\ \mathbf{B}_{:,3,i} &:= -\mathbf{ba}_i \\ \mathbf{B}_{:,2,i} &:= \frac{\mathbf{ad}_i \times \mathbf{ba}_i}{\|\mathbf{ad}_i \times \mathbf{ba}_i\|} \\ \mathbf{B}_{:,1,i} &:= \frac{\mathbf{B}_{:,2,i} \times \mathbf{B}_{:,3,i}}{\|\mathbf{B}_{:,2,i} \times \mathbf{B}_{:,3,i}\|}\end{aligned} \tag{2.2.5}$$

Using the IUPAC defined rotation directions the function for transforming spherical to cartesian coordinates is defined as:[1]

$$\begin{aligned}\mathbf{S}\colon ([0,\infty) \times [0,\pi) \times [0,2\pi))^n &\to \mathbb{R}^{3,n} \\ \begin{pmatrix} r \\ \alpha \\ \delta \end{pmatrix} := \mathbf{C}_{:,i} &\mapsto \mathbf{S}_{:,i} := \begin{pmatrix} r\sin(\alpha)\cos(\delta) \\ -r\sin(\alpha)\sin(\delta) \\ -r\cos(\alpha) \end{pmatrix}\end{aligned} \tag{2.2.6}$$

Combining these equations gives:

$$\mathbf{X}_{i,j}(\mathbf{C}_{:,j}, \mathbf{X}^R_{:,(b(j),a(j),d(j))}) = \mathbf{B}_{i,:,j}(\mathbf{X}^R_{:,(b(j),a(j),d(j))}) \cdot \mathbf{S}_{:,j}(\mathbf{C}_{:,j}) + \mathbf{X}^R_{i,b(j)} \tag{2.2.7}$$

$$\mathbf{X}^R_{:,i}(\mathbf{C}_{:,i}, \mathbf{X}^R_{:,(b(i),a(i),d(i))}) = \begin{cases} \mathbf{X}_{:,i}(\mathbf{C}_{:,i}, \mathbf{X}^R_{:,(b(i),a(i),d(i))}) & i \in I \\ \mathbf{X}^R_{:,i} & i \in I^R \setminus I \end{cases} \tag{2.2.8}$$

Since $\mathbf{X}^R$ itself depends on $\mathbf{C}$, the function $\mathbf{X}$ can be nested until the references are one of the three absolute references.

$$\begin{aligned} &\mathbf{X}_{:,i}(\mathbf{C}_{:,i}, \mathbf{X}^R_{:,(b(i),a(i),d(i))}) \\ =&\mathbf{X}_{:,i}(\mathbf{C}_{:,i}, \mathbf{X}^R_{:,(b(i),a(i),d(i))}(\mathbf{C}_{:,(b(i),a(i),d(i))}, \mathbf{X}^R_{:,(b(b(i)),b(a(i)),b(d(i)),a(b(i)),a(a(i)),a(d(i)),d(b(i)),d(a(i)),d(d(i)))})) \\ =&\dots \end{aligned} \tag{2.2.9}$$

2.2.3. Gradient for the transformation from Z-matrix to cartesian coordinates

The derivatives of cartesian coordinates with respect to bonds, angles, and dihedrals in the Z-matrix have to be calculated. Using the defined variables of section 2.2.2, the following expression should be evaluated:

$$\left(\frac{\partial \mathbf{X}}{\partial \mathbf{C}}\right)_{i,j,l,k} = \frac{\partial \mathbf{X}_{i,j}}{\partial \mathbf{C}_{k,l}} \tag{2.2.10}$$

Inserting equation 2.2.7 into equation 2.2.10 yields:

$$\begin{aligned} \frac{\partial \mathbf{X}_{i,j}}{\partial \mathbf{C}_{k,l}} &= \frac{\partial \mathbf{B}_{i,:,j}(\mathbf{X}^R_{:,(b(j),a(j),d(j))}) \cdot \mathbf{S}_{:,j} + \mathbf{X}^R_{i,b(j)}}{\partial \mathbf{C}_{k,l}} \\ &= \frac{\partial \mathbf{B}_{i,:,j}(\mathbf{X}^R_{:,(b(j),a(j),d(j))}) \cdot \mathbf{S}_{:,j}}{\partial \mathbf{C}_{k,l}} + \frac{\partial \mathbf{X}^R_{i,b(j)}}{\partial \mathbf{C}_{k,l}} \\ &= \mathbf{B}_{i,:,j}(\mathbf{X}^R_{:,(b(j),a(j),d(j))}) \frac{\partial \mathbf{S}_{:,j}}{\partial \mathbf{C}_{k,l}} + \frac{\partial \mathbf{B}_{i,:,j}(\mathbf{X}^R_{:,(b(j),a(j),d(j))})}{\partial \mathbf{C}_{k,l}} \cdot \mathbf{S}_{:,j} + \frac{\partial \mathbf{X}^R_{i,b(j)}}{\partial \mathbf{C}_{k,l}} \\ &= \underbrace{\mathbf{B}_{i,:,j}(\mathbf{X}^R_{:,(b(j),a(j),d(j))}) \frac{\partial \mathbf{S}_{:,j}}{\partial \mathbf{C}_{k,l}}}_{A} + \underbrace{\frac{\partial \mathbf{B}_{i,:,j}}{\partial \mathbf{X}^R_{:,(b(j),a(j),d(j))}} \frac{\partial \mathbf{X}^R_{:,(b(j),a(j),d(j))}}{\partial \mathbf{C}_{k,l}} \cdot \mathbf{S}_{:,j} + \frac{\partial \mathbf{X}^R_{i,b(j)}}{\partial \mathbf{C}_{k,l}}}_{B} \end{aligned}$$

Now there are three cases to consider $(j < l)$, $(j = l)$, and $(j > l)$ which will be treated separately.

Derivative with $j < l$

The variation of a Z-matrix entry can never affect the cartesian position of atoms previous to this entry. Therefore, the corresponding derivative has to be zero.

$$\forall\, i, j, k, l \in \{1 \dots n\} \wedge j < l : \frac{\partial \mathbf{X}_{i,j}}{\partial \mathbf{C}_{k,l}} = 0 \tag{2.2.11}$$

It should be noted that the derivative of absolute references after Z-matrix entries is always zero:

$$\forall\, j \in I^R \setminus I : \frac{\partial \mathbf{X}^R_{i,j}}{\partial \mathbf{C}_{k,l}} = 0 \tag{2.2.12}$$

Derivative with $j = l$

In this case we know, because of equation 2.2.4, that $l > b(l), a(l), d(l) = b(j), a(j), d(j)$. Together with equation 2.2.11 it may be concluded, that:

$$\begin{aligned} \frac{\partial \mathbf{X}^R_{:,(b(j),a(j),d(j))}}{\partial \mathbf{C}_{k,j}} &= 0 \\ \frac{\partial \mathbf{X}^R_{i,b(j)}}{\partial \mathbf{C}_{k,j}} &= 0 \\ \Rightarrow \quad B &= 0 \end{aligned}$$

Which implies:

$$\frac{\partial \mathbf{X}_{i,j}}{\partial \mathbf{C}_{k,j}} = \mathbf{B}_{i,:,j}(\mathbf{X}^R_{:,(b(j),a(j),d(j))}) \frac{\partial \mathbf{S}_{:,j}}{\partial \mathbf{C}_{k,j}} \tag{2.2.13}$$

The only missing term is the gradient of $\mathbf{S}$ (equation 2.2.7) which is given by:

$$\frac{\partial \mathbf{S}_{:,j}}{\partial \mathbf{C}_{:,j}}(\mathbf{C}_{:,j}) = \begin{pmatrix} \sin(\mathbf{C}_{2,j})\cos(\mathbf{C}_{3,j}) & \mathbf{C}_{1,j}\cos(\mathbf{C}_{2,j})\cos(\mathbf{C}_{3,j}) & -\mathbf{C}_{1,j}\sin(\mathbf{C}_{2,j})\sin(\mathbf{C}_{3,j}) \\ -\sin(\mathbf{C}_{2,j})\sin(\mathbf{C}_{3,j}) & -\mathbf{C}_{1,j}\sin(\mathbf{C}_{3,j})\cos(\mathbf{C}_{2,j}) & -\mathbf{C}_{1,j}\sin(\mathbf{C}_{2,j})\cos(\mathbf{C}_{3,j}) \\ -\cos(\mathbf{C}_{2,j}) & \mathbf{C}_{1,j}\sin(\mathbf{C}_{2,j}) & 0 \end{pmatrix}$$

Derivative with $j > l$

A closer inspection of equation 2.2.6 reveals, that the j-th column of $\mathbf{S}$ solely depends on the j-th column of the argument matrix. It may be concluded, that

$$\begin{aligned} \forall\, j \neq l : \frac{\partial \mathbf{S}_{i,j}}{\partial \mathbf{C}_{k,l}} &\equiv 0 \\ \Rightarrow \quad A &= 0 \end{aligned}$$

which implies:

$$\frac{\partial \mathbf{X}_{i,j}}{\partial \mathbf{C}_{k,l}} = \frac{\partial \mathbf{B}_{i,:,j}}{\partial \mathbf{X}^R_{:,(b(j),a(j),d(j))}} \frac{\partial \mathbf{X}^R_{:,(b(j),a(j),d(j))}}{\partial \mathbf{C}_{k,l}} \cdot \mathbf{S}_{:,j} + \frac{\partial \mathbf{X}^R_{i,b(j)}}{\partial \mathbf{C}_{k,l}} \tag{2.2.14}$$

Combined algorithm

The equations 2.2.11, 2.2.13, and 2.2.14 can be used to formulate one combined algorithm:

Algorithm 1 Calculation of the analytical gradient for the coordinate transformation from Z-matrix to cartesian coordinates

---

for $j = 1 \ldots n$ **do**
  for $l = (j+1) \ldots n$ **do**
    ▷ Alternatively the whole array may be initialized with zeros.

$$\left(\frac{\partial \mathbf{X}}{\partial \mathbf{C}}\right)_{:,j,l,:}(\mathbf{C}, \mathbf{X}^R) := \frac{\partial \mathbf{X}_{:,j}}{\partial \mathbf{C}_{:,l}}(\mathbf{C}, \mathbf{X}^R) := 0$$

  end for
end for
for $j = 1 \ldots n$ **do**

$$\left(\frac{\partial \mathbf{X}}{\partial \mathbf{C}}\right)_{:,j,j,:}(\mathbf{C}, \mathbf{X}^R) := \frac{\partial \mathbf{X}_{:,j}}{\partial \mathbf{C}_{:,j}}(\mathbf{C}, \mathbf{X}^R) = \mathbf{B}_{:,:,j}(\mathbf{X}^R_{:,(b(j),a(j),d(j))}) \frac{\partial \mathbf{S}_{:,j}}{\partial \mathbf{C}_{:,j}}(\mathbf{C})$$

end for
for $j = 1 \ldots n$ **do**
  for $l = 1 \ldots j-1$ **do**

$$
\begin{aligned}
\left(\frac{\partial \mathbf{X}}{\partial \mathbf{C}}\right)_{:,j,l,:}(\mathbf{C},\mathbf{X}^R) &:= \frac{\partial \mathbf{X}_{:,j}}{\partial \mathbf{C}_{:,l}}(\mathbf{C},\mathbf{X}^R) \\
&= \frac{\partial \mathbf{B}_{:,:,j}}{\partial \mathbf{X}^R_{:,(b(j),a(j),d(j))}}(\mathbf{X}^R)\frac{\partial \mathbf{X}^R_{:,(b(j),a(j),d(j))}}{\partial \mathbf{C}_{:,l}}(\mathbf{C},\mathbf{X}^R)\cdot \mathbf{S}_{:,j}(\mathbf{C}) + \frac{\partial \mathbf{X}^R_{:,b(j)}}{\partial \mathbf{C}_{:,l}}(\mathbf{C},\mathbf{X}^R) \\
&= \left(\sum_{m_1=1}^{3}\sum_{k\in\{b(j),a(j),d(j)\}}\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}^R}(\mathbf{X}^R)\right)_{:,:,j,k,m_1}\left(\frac{\partial \mathbf{X}^R}{\partial \mathbf{C}}(\mathbf{C},\mathbf{X}^R)\right)_{m_1,k,l,:}\right)\cdot \mathbf{S}_{:,j}(\mathbf{C}) \\
&\quad + \left(\frac{\partial \mathbf{X}^R}{\partial \mathbf{C}}(\mathbf{C},\mathbf{X}^R)\right)_{:,b(j),l,:} \\
&= \sum_{m_1=1}^{3}\sum_{k\in\{b(j),a(j),d(j)\}}\sum_{m_2=1}^{3}\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}^R}(\mathbf{X}^R)\right)_{:,m_2,j,k,m_1}\mathbf{S}_{m_2,j}(\mathbf{C})\left(\frac{\partial \mathbf{X}^R}{\partial \mathbf{C}}(\mathbf{C},\mathbf{X}^R)\right)_{m_1,k,l,:} \\
&\quad + \left(\frac{\partial \mathbf{X}^R}{\partial \mathbf{C}}(\mathbf{C},\mathbf{X}^R)\right)_{:,b(j),l,:}
\end{aligned}
$$

end for
end for

The term

$$\frac{\partial \mathbf{X}^R_{:,(b(j),a(j),d(j))}}{\partial \mathbf{C}_{:,l}}(\mathbf{C},\mathbf{X}^R)$$

in the third loop, does not have to be calculated, because it is already known from previous runs and can be looked up in the same array.
Besides it might be advantageous to test if

$$\frac{\partial \mathbf{X}^R_{m_1,b(j)}}{\partial \mathbf{C}_{m_2,l}}(\mathbf{C},\mathbf{X}^R)\neq 0 \qquad \frac{\partial \mathbf{X}^R_{m_1,a(j)}}{\partial \mathbf{C}_{m_2,l}}(\mathbf{C},\mathbf{X}^R)\neq 0 \qquad \frac{\partial \mathbf{X}^R_{m_1,d(j)}}{\partial \mathbf{C}_{m_2,l}}(\mathbf{C},\mathbf{X}^R)\neq 0$$

and calculate on demand:

$$\frac{\partial \mathbf{B}_{:,:,j}}{\partial \mathbf{X}^R_{m_1,b(j)}}(\mathbf{X}^R) \qquad \frac{\partial \mathbf{B}_{:,:,j}}{\partial \mathbf{X}^R_{m_1,a(j)}}(\mathbf{X}^R) \qquad \frac{\partial \mathbf{B}_{:,:,j}}{\partial \mathbf{X}^R_{m_1,d(j)}}(\mathbf{X}^R)$$

The symbolically evaluated derivatives $\frac{\partial \mathbf{B}}{\partial \mathbf{X}^R}$ are given in the appendix under section A.4.

2.2.4. Transformation from cartesian to Z-matrix coordinates

The inverse function of $\mathbf{S}$ (equation 2.2.6) is given by:

$$
\begin{aligned}
&\mathbf{S}^{-1}\colon \mathbb{R}^{3,n} \to ([0,\infty)\times[0,\pi)\times[-\pi,\pi))^n \\
&\begin{pmatrix} x \\ y \\ z \end{pmatrix} := \mathbf{V}_{:,i} \mapsto \mathbf{S}_{:,i} := \begin{pmatrix} r \\ \alpha \\ \delta \end{pmatrix} := \begin{pmatrix} \sqrt{x^2+y^2+z^2} \\ \operatorname{acos}\left(\frac{-z}{r}\right) \\ \operatorname{atan2}\left(\frac{-y}{r}, \frac{x}{r}\right) \end{pmatrix}
\end{aligned} \tag{2.2.15}
$$

The affine-linear transformation $\mathbf{T}$ of atom i into the coordinate system spanned by its references is given by:

$$
\begin{aligned}
&\mathbf{T}\colon \mathbb{R}^{3,n} \to \mathbb{R}^{3,n} \\
&\quad \mathbf{X}_{:,i} \mapsto \mathbf{T}_{:,i} = \mathbf{B}^{-1}_{:,:,i}\left(\mathbf{X^R}_{:,(b(i),a(i),d(i))}\right)\cdot\left(\mathbf{X}_{:,i} - \mathbf{X^R}_{:,b(i)}\right) \\
&\qquad\qquad = \mathbf{B}^{T}_{:,:,i}\left(\mathbf{X^R}_{:,(b(i),a(i),d(i))}\right)\cdot\left(\mathbf{X}_{:,i} - \mathbf{X^R}_{:,b(i)}\right)
\end{aligned} \tag{2.2.16}
$$

The second equality is due to the orthogonality of $\mathbf{B}$. This gives for the bond, angle and dihedral of the i-th atom the following equation:

$$
\begin{aligned}
\mathbf{C}_{:,i} &= \mathbf{S}^{-1}_{:,i}(\mathbf{T}_{:,:,i}(\mathbf{X}_{:,i})) \\
&= \mathbf{S}^{-1}_{:,i}\left(\mathbf{B}^{T}_{:,:,i}\left(\mathbf{X^R}_{:,(b(i),a(i),d(i))}\right)\cdot\left(\mathbf{X}_{:,i} - \mathbf{X^R}_{:,b(i)}\right)\right)
\end{aligned} \tag{2.2.17}
$$

Note that the use of the atan2 function in equation 2.2.15 automatically solves the corner case of an undefined dihedral as discussed in subsection 2.1.2. Typical numerical implementations of atan2 are defined as well in cases where one or two arguments are zero or infinite.[5]

2.2.5. Gradient for the transformation from cartesian to Z-matrix coordinates

The aim is to calculate the derivative of bonds, angles, and dihedrals in the Z-matrix with respect to x, y, and z coordinates in cartesian space. Using the defined variables of subsection 2.2.4, the following expression should be evaluated:

$$
\left(\frac{\partial \mathbf{X}}{\partial \mathbf{C}}\right)_{i,j,l,k} = \frac{\partial \mathbf{X}_{i,j}}{\partial \mathbf{C}_{k,l}} \tag{2.2.18}
$$

Inserting equation 2.2.17 into 2.2.18 yields:

$$\begin{aligned}\left(\frac{\partial \mathbf{C}}{\partial \mathbf{X}}\right)_{:,j,l,:}(\mathbf{X}) &= \frac{\partial \mathbf{C}_{:,j}}{\partial \mathbf{X}_{:,l}}(\mathbf{X}) \\ &= \sum_{m_1=1}^{n} \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,m_1}}(\mathbf{T}_{:,:,m_1}(\mathbf{X}_{:,m_1})) \cdot \frac{\partial \mathbf{T}_{:,m_1}}{\partial \mathbf{X}_{:,l}}(\mathbf{X}_{:,l})\end{aligned}$$

A closer inspection of equation 2.2.15 reveals that the j-th column of $\mathbf{S}^{-1}$ solely depends on the j-th column of the argument matrix $\mathbf{V}$:

$$\left(\frac{\partial \mathbf{C}}{\partial \mathbf{X}}\right)_{:,j,l,:}(\mathbf{X}) = \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \cdot \frac{\partial \mathbf{T}_{:,j}}{\partial \mathbf{X}_{:,l}}(\mathbf{X}_{:,l}) \tag{2.2.19}$$

The first factor in equation 2.2.19 is obtained by straightforwardly deriving equation 2.2.15. For a succinct notation it is necessary to define the following variables:

$$\begin{aligned}(x, y, z)^{\mathsf{T}} &:= \mathbf{X}_{:,j} \\ r &:= \sqrt{x^2 + y^2 + z^2}\end{aligned}$$

This gives:

$$\frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) = \begin{pmatrix} \frac{x}{r} & \frac{y}{r} & \frac{z}{r} \\ \frac{-xz}{r^2\sqrt{x^2+y^2}} & \frac{-yz}{r^2\sqrt{x^2+y^2}} & \frac{\sqrt{x^2+y^2}}{r^2} \\ \frac{y}{x^2+y^2} & \frac{-x}{x^2+y^2} & 0 \end{pmatrix} \tag{2.2.20}$$

For $x = y = 0$ a removable singularity exists, which is due to the undefined dihedral case from Figure 2.2a. Removing this singularity yields for $x = y = 0$:

$$\begin{pmatrix} 0 & 0 & 1 \\ -\frac{1}{z} & -\frac{1}{z} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The second factor in equation 2.2.19 may be derived by applying the product rule on equation 2.2.16.

$$\begin{aligned}\frac{\partial \mathbf{T}_{:,j}}{\partial \mathbf{X}_{:,l}}(\mathbf{X}_{:,l}) = &\frac{\partial \mathbf{B}^T_{:,:,j}}{\partial \mathbf{X}_{:,l}}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right) \cdot \left(\mathbf{X}_{:,j} - \mathbf{X^R}_{:,b(j)}\right) \\ &+ \mathbf{B}^T_{:,:,j}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right) \cdot \left(\frac{\partial \mathbf{X}_{:,j}}{\partial \mathbf{X}_{:,l}}(\mathbf{X}_{:,l}) - \frac{\partial \mathbf{X^R}_{:,b(j)}}{\partial \mathbf{X}_{:,l}}(\mathbf{X}_{:,l})\right)\end{aligned} \tag{2.2.21}$$

Before inserting equations 2.2.20 and 2.2.21 into 2.2.19 it is advantageous to rule out several cases which have to be zero.

Derivative with $l \notin \{j, b(j), a(j), d(j)\}$

The Z-matrix entries of the j-th atom can only change if itself or its reference frame moves in cartesian space. For this reason we can write:

$$\forall\, l \notin \{j, b(j), a(j), d(j)\}\colon \left(\frac{\partial \mathbf{C}}{\partial \mathbf{X}}\right)_{:,j,l,:} \equiv \mathbf{0} \tag{2.2.22}$$

Derivative with $l = j$

Deriving the reference frame of the j-th atom with respect to $\mathbf{v}_j$ itself, has to be zero:

$$\frac{\partial \mathbf{B}^T_{:,:,j}}{\partial \mathbf{X}_{:,j}} \equiv \mathbf{0}$$

The positions of each atom in cartesian space are independent of each other:

$$\frac{\partial \mathbf{X^R}_{:,b(j)}}{\partial \mathbf{X}_{:,j}} \equiv \mathbf{0}$$

The following equation is trivial:

$$\frac{\partial \mathbf{X}_{:,j}}{\partial \mathbf{X}_{:,j}} \equiv \mathbf{1}$$

Combining the three previous equations and inserting into 2.2.21 yields:

$$\frac{\partial \mathbf{T}_{:,j}}{\partial \mathbf{X}_{:,j}}(\mathbf{X}) = \mathbf{B}^T_{:,:,j}\big(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\big)$$

Hence by inserting into 2.2.19:

$$\begin{aligned}\left(\frac{\partial \mathbf{C}}{\partial \mathbf{X}}\right)_{:,j,j,:}(\mathbf{X}) &= \frac{\partial \mathbf{C}_{:,j}}{\partial \mathbf{X}_{:,j}}(\mathbf{X}) \\ &= \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \cdot \mathbf{B}^T_{:,:,j}\big(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\big)\end{aligned} \tag{2.2.23}$$

Derivative with $l = b(j)$

With the same argumentation as in subsubsection 2.2.5 the following relationships hold:

$$\frac{\partial \mathbf{X^R}_{:,b(j)}}{\partial \mathbf{X}_{:,b(j)}} \equiv \mathbf{1}$$
$$\frac{\partial \mathbf{X}_{:,j}}{\partial \mathbf{X}_{:,b(j)}} \equiv \mathbf{0}$$

By inserting into equation 2.2.21 one obtains:

$$\frac{\partial \mathbf{T}_{:,j}}{\partial \mathbf{X}_{:,b(j)}}(\mathbf{X}) = \frac{\partial \mathbf{B}^T_{:,:,j}}{\partial \mathbf{X}_{:,b(j)}}\big(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\big)\big(\mathbf{X}_{:,j} - \mathbf{X^R}_{:,b(j)}\big) - \mathbf{B}^T_{:,:,j}\big(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\big)$$

Which finally gives by inserting into 2.2.19:

$$\begin{aligned}
&\left(\frac{\partial \mathbf{C}}{\partial \mathbf{X}}\right)_{:,j,b(j),:}(\mathbf{X}) = \frac{\partial \mathbf{C}_{:,j}}{\partial \mathbf{X}_{:,b(j)}}(\mathbf{X}) \\
=&\frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \\
&\cdot \left(\frac{\partial \mathbf{B}^T_{:,:,j}}{\partial \mathbf{X}_{:,b(j)}}\big(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\big)\big(\mathbf{X}_{:,j} - \mathbf{X^R}_{:,b(j)}\big) - \mathbf{B}^T_{:,:,j}\big(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\big)\right)
\end{aligned} \tag{2.2.24}$$

Derivative with $l = a(j)$ or $l = d(j)$

With the same argumentation as in subsubsection 2.2.5 the following relationships hold:

$$\frac{\partial \mathbf{X^R}_{:,b(j)}}{\partial \mathbf{X}_{:,a(j)}} \equiv \frac{\partial \mathbf{X^R}_{:,b(j)}}{\partial \mathbf{X}_{:,d(j)}} \equiv \mathbf{0}$$
$$\frac{\partial \mathbf{X}_{:,j}}{\partial \mathbf{X}_{:,a(j)}} \equiv \frac{\partial \mathbf{X}_{:,j}}{\partial \mathbf{X}_{:,d(j)}} \equiv \mathbf{0}$$

It may be concluded by inserting into equation 2.2.21:

$$\frac{\partial \mathbf{T}_{:,j}}{\partial \mathbf{X}_{:,a(j)}}(\mathbf{X}) = \frac{\partial \mathbf{B}^T_{:,:,j}}{\partial \mathbf{X}_{:,a(j)}}\big(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\big)\big(\mathbf{X}_{:,j} - \mathbf{X^R}_{:,b(j)}\big)$$
$$\frac{\partial \mathbf{T}_{:,j}}{\partial \mathbf{X}_{:,d(j)}}(\mathbf{X}) = \frac{\partial \mathbf{B}^T_{:,:,j}}{\partial \mathbf{X}_{:,d(j)}}\big(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\big)\big(\mathbf{X}_{:,j} - \mathbf{X^R}_{:,b(j)}\big)$$

Which finally gives by inserting into 2.2.19:

$$\begin{aligned}
\left(\frac{\partial \mathbf{C}}{\partial \mathbf{X}}\right)_{:,j,a(j),:}(\mathbf{X}) &= \frac{\partial \mathbf{C}_{:,j}}{\partial \mathbf{X}_{:,a(j)}}(\mathbf{X}) \\
&= \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \cdot \left(\frac{\partial \mathbf{B}^{T}_{:,:,j}}{\partial \mathbf{X}_{:,a(j)}}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right)\left(\mathbf{X}_{:,j} - \mathbf{X^R}_{:,b(j)}\right)\right) \\
\left(\frac{\partial \mathbf{C}}{\partial \mathbf{X}}\right)_{:,j,d(j),:}(\mathbf{X}) &= \frac{\partial \mathbf{C}_{:,j}}{\partial \mathbf{X}_{:,d(j)}}(\mathbf{X}) \\
&= \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \cdot \left(\frac{\partial \mathbf{B}^{T}_{:,:,j}}{\partial \mathbf{X}_{:,d(j)}}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right)\left(\mathbf{X}_{:,j} - \mathbf{X^R}_{:,b(j)}\right)\right)
\end{aligned} \tag{2.2.25}$$

Combined Algorithm

The equations 2.2.22, 2.2.23, 2.2.24, and 2.2.25 can be used to formulate one combined algorithm:

Algorithm 2 Calculation of the analytical gradient for the coordinate transformation from cartesian to Z-matrix coordinates

---

for $j = 1 \dots n$ **do**
  for $l = 1 \dots n$ **do**
    if $l \notin \{j, b(j), a(j), d(j)\}$ **then**

▷ Alternatively the whole array may be initialized with zeros.

$$\left(\frac{\partial \mathbf{C}}{\partial \mathbf{X}}\right)_{:,j,l,:}(\mathbf{X}) := 0$$

    end if
  end for
end for
for $j = 1 \dots n$ **do**

$$\frac{\partial \mathbf{C}_{:,j}}{\partial \mathbf{X}_{:,j}}(\mathbf{X}) := \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \cdot \mathbf{B}^{T}_{:,:,j}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right)$$

$$\begin{aligned}\frac{\partial \mathbf{C}_{:,j}}{\partial \mathbf{X}_{:,b(j)}}(\mathbf{X}) &:= \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \\ &\quad \cdot \left(\frac{\partial \mathbf{B}^{T}_{:,:,j}}{\partial \mathbf{X}_{:,b(j)}}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right)\left(\mathbf{X}_{:,j} - \mathbf{X^R}_{:,b(j)}\right) - \mathbf{B}^{T}_{:,:,j}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right) \right) \\ &= \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \\ &\quad \cdot \left(\sum_{m_1=1}^{3} \left(\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right) \right)_{m_1,:,j,b(j),:} \left(\mathbf{X}_{m_1,j} - \mathbf{X^R}_{m_1,b(j)}\right) \right) \right. \\ &\quad \left. - \mathbf{B}^{T}_{:,:,j}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right) \right)\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathbf{C}_{:,j}}{\partial \mathbf{X}_{:,a(j)}}(\mathbf{X}) &:= \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \cdot \left(\frac{\partial \mathbf{B}^{T}_{:,:,j}}{\partial \mathbf{X}_{:,a(j)}}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right)\left(\mathbf{X}_{:,j} - \mathbf{X^R}_{:,b(j)}\right) \right) \\ &= \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \\ &\quad \cdot \left(\sum_{m_1=1}^{3} \left(\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right) \right)_{m_1,:,j,a(j),:} \left(\mathbf{X}_{m_1,j} - \mathbf{X^R}_{m_1,b(j)}\right) \right) \right)\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathbf{C}_{:,j}}{\partial \mathbf{X}_{:,d(j)}}(\mathbf{X}) &:= \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \cdot \left(\frac{\partial \mathbf{B}^{T}_{:,:,j}}{\partial \mathbf{X}_{:,d(j)}}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right)\left(\mathbf{X}_{:,j} - \mathbf{X^R}_{:,b(j)}\right) \right) \\ &= \frac{\partial \mathbf{S}^{-1}_{:,j}}{\partial \mathbf{V}_{:,j}}(\mathbf{T}_{:,:,j}(\mathbf{X}_{:,j})) \\ &\quad \cdot \left(\sum_{m_1=1}^{3} \left(\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\left(\mathbf{X^R}_{:,(b(j),a(j),d(j))}\right) \right)_{m_1,:,j,d(j),:} \left(\mathbf{X}_{m_1,j} - \mathbf{X^R}_{m_1,b(j)}\right) \right) \right)\end{aligned}$$

end for

The derivative $\frac{\partial \mathbf{S}^{-1}}{\partial \mathbf{V}}$ is given by equation 2.2.20. The evaluated derivatives $\frac{\partial \mathbf{B}^{\mathsf{T}}}{\partial \mathbf{X}^{R}}$ are given in the appendix under section A.4 if the transposition of the first two indices is taken into account.

2.2.6. Application example of the gradient

The best illustration of the gradient is given by linearising the transformation to cartesian coordinates. So let:

$\mathbf{C_0} \in ([0,\infty) \times [0,\pi) \times [0,2\pi))^n$	be a Z-matrix,
$\mathbf{H} \in ([0,\infty) \times [0,\pi) \times [0,2\pi))^n$	be a small distortion of the Z-matrix,
$\mathbf{X}\colon ([0,\infty) \times [0,\pi) \times [0,2\pi))^n \to \mathbb{R}^{3,n}$	be the transformation to cartesian coordinates.

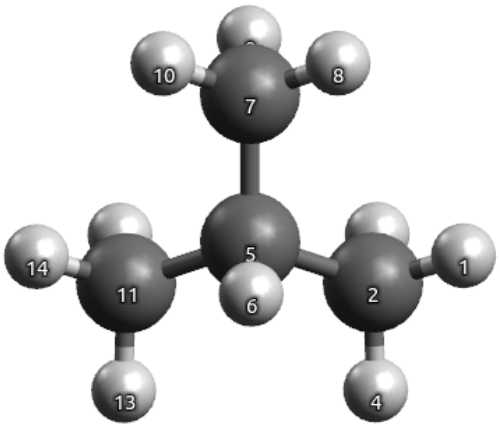
Then the following linear approximation is assumed:

$$\mathbf{X}(\mathbf{C_0} + \mathbf{H}) \approx \mathbf{X}(\mathbf{C_0}) + \frac{\partial \mathbf{X}}{\partial \mathbf{C}}(\mathbf{C_0})\mathbf{H} \tag{2.2.26}$$

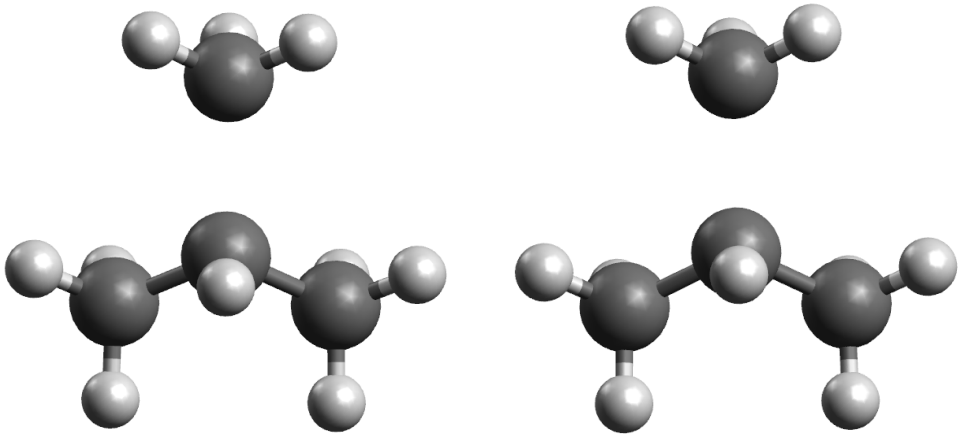
This equation was applied on the Z-matrix of 2-methylpropane (Table 2.3), which was automatically generated using `chemcoord` (code example 2.1). The assignment of labels to the atoms is illustrated in Figure 2.4a. The Z-matrix shows that modifying the bond length of carbon atom 7 moves the whole methyl group away. It can be seen in Figure 2.4b, that the actual distorted molecule is very similar to the molecule generated from linearising the distortion (Figure 2.4c). The structures were calculated using the following code example 2.1.

Code example 2.1: Linearising the conversion to cartesian coordinates.

```
isopropane = cc.Cartesian.read_xyz('2_methylpropane.xyz', start_index=1)
C_0 = isopropane.get_zmat()
C_0.to_latex(escape=False, float_format='{:.3f}'.format)
H = C_0.copy()
H.unsafe_loc[:, ['bond', 'angle', 'dihedral']] = 0.
# Compare Table 2.3
H.unsafe_loc[7, 'bond'] = 0.5
grad_C_to_X = C_0.get_grad_cartesian()
structures = [C_0.get_cartesian(), C_0.get_cartesian() + grad_C_to_X(H),
              (C_0 + H).get_cartesian()]
cc.xyz_functions.to_molden(structures, 'gradient_example.molden')
```



(a) The undistorted molecule. $\mathbf{X}(\mathbf{C_0})$



(b) The distorted molecule. $\mathbf{X}(\mathbf{C_0} + \mathbf{H})$

(c) Linear approximation to the distorted molecule. $\mathbf{X}(\mathbf{C_0}) + \frac{\partial \mathbf{X}}{\partial \mathbf{C}}(\mathbf{C_0})\mathbf{H}$

Figure 2.4.: Comparison of the linearised with the actual distortion. Structures were generated from code example 2.1.

Table 2.3.: Zmatrix of 2-methylpropane as generated by code example 2.1. The assignment of labels is illustrated in Figure 2.4a. Bond lengths in Å, angles in degrees.

	atom	b	bond	a	angle	d	dihedral
5	C	$\vec{0}$	0.373	$\vec{e_z}$	66.392	$\vec{e_x}$	-100.576
2	C	5	1.520	$\vec{e_z}$	95.609	$\vec{e_x}$	-10.578
11	C	5	1.520	2	110.506	$\vec{e_x}$	-99.902
7	C	5	1.519	2	110.512	11	122.639
6	H	5	1.117	2	108.405	11	-118.668
1	H	2	1.098	5	111.344	11	178.701
3	H	2	1.098	5	111.575	11	-61.305
4	H	2	1.098	5	111.338	11	58.673
12	H	11	1.098	5	111.582	2	61.295
13	H	11	1.098	5	111.348	2	-58.695
14	H	11	1.098	5	111.348	2	-178.720
8	H	7	1.098	5	111.343	2	58.688
9	H	7	1.098	5	111.567	2	-61.304
10	H	7	1.098	5	111.344	2	178.716

2.3. A well defined chemist's Z-matrix

2.3.1. Obtaining a well defined chemist's Z-matrix

Up to this point the Z-matrix was introduced as a coordinate system, where the i-th position is given by spherical coordinates in the reference frame of three other positions. Also the discussion of corner cases in subsection 2.1.2 and the derivation of equations for the transformation in section 2.2 can be applied to arbitrary problems expressed in cartesian coordinates. It is the aim in this section to obtain a Z-matrix suited for chemical needs and keep it valid if bonds, angles, and dihedrals change. For this purpose the construction table is defined as a Z-matrix with only the b, a, and d columns (compare e. g. Table 2.3) to obtain a representation of the functions from equation 2.2.3. The problem of defining a chemical Z-matrix reduces to finding a "good" construction table. Besides if a certain problem requires a specific construction table, the user can create one manually, while the calculation of Z-matrix entries bond, angle, and dihedral is performed by the computer. It has to be emphasized that there is not one single correct solution for choosing a construction table, but the following rules of thumb should give similar results to a human chemist constructing a Z-matrix manually:

- The bond defining atom should lie in the the first coordination sphere of i.

- The angle defining atom should lie in the first coordination sphere of the bond defining atom and in the second coordination sphere of atom i.
- The dihedral defining atom should lie in the first coordination sphere of the angle defining atom, in the second coordination sphere of the bond defining atom, and in the third coordination sphere of atom i.
- The Z-matrix definition should start in the geometric center of the molecule i. e. start with the atom nearest to the geometric center.
- High valency atoms should appear higher up in the Z-matrix and are preferred reference atoms. Valency in this context is simply the number of connected atoms. There is no distinction between single, double, and triple bonds.
- The variation in the b, a, and d columns should be kept minimal. If it is possible for atom i to use the same references as $i-1$, without vioaling previous constraints, it should do so.
- The used references should propagate through, which means that the bond defining atom of the bond defining atom of i should be the angle defining atom of i and so on.

It is worth noting that apart from starting near the molecule's geometric center, all other constraints are just depending on the connectivity graph and not on specific positions in cartesian space. Hence it is very important to use a suitable data structure representing the connectivity graph. Since the following algorithm involves a lot of tests for membership and set intersections, it is advantageous to use hashtable based data structures. The best representation of the connectivity graph in terms of performance and readability of code is a dictionary of atom labels pointing to totally ordered sets of atom labels. The total order of the sets is obtained from ordering by descending valency of atoms. Another datastructure that can be used to great advantage is the ordered dictionary, which keeps the insertion order of keys. `chemcoord` made great use of the implementations from the `sortedcontainers` library.[6] Since sorted sets and ordered dictionaries are not available in every language, the pseudocode algorithm 3 only assumes the availability of sets and dictionaries, which are sorted *ad hoc* when needed.

It is important to note that a single connected structure is assumed for algorithm 3, but in certain situation it is necessary to construct a Z-matrix for an ensemble of molecules/fragments. In this case algorithm 3 is applied onto each fragment/substructure separately. The resulting Z-matrices are then concatenated, which is performed by

substituting the absolute references in the first three rows with atom labels from other Z-matrices.

Algorithm 3 Finding a chemical construction table.

```
function GET_CONSTRUCTION_TABLE(X)
    B := GET_CONNECTIVITY_GRAPH(X)
    i := GET_ATOM_LABEL_CLOSEST_TO_CENTROID(X)
    visited := {i}
    R[i] := {i: {'b': 'origin', 'a': 'e_z', 'd': 'e_x'}}  ▷ This is the construction table.
    parent := {j : i for j in B[i]}
    work_bond_dict := {j : (B[j] \ visited) for j in B[i]}
    while not empty work_bond_dict do
        for i in SortByValency(Keys(work_bond_dict)) do
            if i ∈ visited then
                continue
            else
                b := parent[i]
                if b in Index(R)[:3] then
                    if Length(R) = 1 then
                        R[i] := {'b': b, 'a': 'e_z', 'd': 'e_x'}
                    else if Length(R) = 2 then
                        R[i] := {'b': b, 'a': MaxV(B[b] ∩ Keys(R)) , 'd': 'e_x'}
                                ▷ MaxV returns the atom label with the highest valency.
                    else
                        if Exists(parent[b]) then
                            a := parent[b]
                        else
                            a := MaxV(B[b] ∩ Keys(R))
                        end if
                        if Exists(parent[a]) and parent[a] not in {b, a} then
                            d := parent[a]
                        else
                            if not empty (B[a] ∩ Keys(R)) \ {b, a} then
                                d := MaxV((B[a] ∩ Keys(R)) \ {b, a})
                            else
                                d := MaxV((B[b] ∩ Keys(R)) \ {b, a})
                            end if
```

```
                    end if
                    R[i] := {'b': b, 'a': a, 'd': d}
                end if
            else
                R[i] := {'b': b, 'a': R[b]['b'], 'd': R[b]['a']}
            end if
        end if
        visited := visited ∩{i}
        for j in SortByValency(Keys(work_bond_dict[i])) do
            new_work_bond_dict[j] := B[j]\ visited
            parent[j] := i
        end for
    end for
    work_bond_dict := new_work_bond_dict
  end while
  return R
end function
```

A test to prevent a locally undefined coordinate system (Figure 2.2b) is still missing in algorithm 3. This test is performed afterwards by checking if there exists an atom i whose reference atoms are linearly aligned. If such an atom is found, its dihedral defining atom d is changed (preferably) to an atom from the same coordination sphere relative to i. Instead of testing afterwards, it is possible to include additional conditionals in algorithm 3 to reject a selection for $d(i)$, if the reference of i is linear. This inline testing could be faster than the additional test afterwards, but it would further increase the amount of nested conditionals up to a point where the code becomes very hard to understand.

2.3.2. Keeping a Z-matrix well defined

A well defined reference frame of atom i may become undefined by changing the bond, angle, and dihedral entries in rows of the Z-matrix further up (relative to i). As discussed in subsection 2.1.2 this problem can be detected only by converting to cartesian coordinates. For this reason it is necessary to convert to cartesian coordinates after *every* assignment to bonds, angles, and dihedrals. If an invalid reference for atom i is found, a dummy atom has to be inserted by using the positions of reference atoms *before the* assignment. The frequent transformation operations make it very important to implement the equations from section 2.2 in an efficient manner. For this purpose the LLVM compiler `numba` was used in `chemcoord` to implement "close to metal" functions extending the Python code.[7]

2.4. Additional implemented functionality

2.4.1. Symbolic algebra

Using the symbolic algebra library `sympy` it was possible to implement support for arbitray symbolic expressions.[8] As in the case of assignments, it is necessary to convert the Z-matrix to cartesian coordinates if all symbolic expressions are substituted with real numbers, to test for a locally undefined coordinate system. If an invalid reference for atom i is encountered, a dummy atom has to be inserted by using the last known valid positions of reference atoms.

2.4.2. Symmetry

Automatically detecting the symmetry point group of a molecule in cartesian space and preserving it during the modification of structures can be a huge advantage. Especially if a hamiltonian for electronic calculations is used that gains in performance by making use of symmetry. Since there are libraries available that offer the detection of point groups and other symmetry related functions, the symmetry code of `chemcoord` mainly consists of wrappers to functions from the `pymatgen` library.[9] Since `pymatgen` did not offer the iterative symmetrisation of a distorted molecule and had performance issues in the generation of a group from a given generator set, some functions of `pymatgen` had to be newly implemented or improved and pushed back to the `pymatgen` library (`https://github.com/materialsproject/pymatgen/pull/760`). These improvements were carried out in the context of this work. In the following section the improved algorithms are discussed and application examples are given.

Group generation

It is the aim of this subsection to generate a group G from a given (not necessarily minimal) generator set S. In the case of point groups of molecules the group elements are orthogonal $\mathbb{R}^{3,3}$ matrices. The previous implementation in `pymatgen` used the recursive algorithm 4 which scales with $|G|^3$. The proof is as following: In line 3 $|S|^2$ multiplications and comparisons are performed. The **for each** loop in line 2 is invoked up to $|G|-|S|$

times. Which gives:

$$\begin{aligned}\sum_{i=|S|}^{|G|} |i|^2 &= \frac{|G|}{3} \cdot (|G|+1) \cdot \left(|G|+\frac{1}{2}\right) - \frac{|S|}{3} \cdot (|S|+1) \cdot \left(|S|+\frac{1}{2}\right) \\ &\approx \frac{|G|}{3} \cdot (|G|+1) \cdot \left(|G|+\frac{1}{2}\right)\end{aligned} \tag{2.4.1}$$

The performance of the algorithm can be improved by using the property that every element of G can be generated by multiplying elements of the generator set with each other.[10] This is the defining property of a generator set. Hence line 2 of algorithm 4 may be replaced by (**for each** $f, g \in G \times S$). In addition, the called function in line 4 repeats a lot of already performed multiplications. This may be circumvented by using an ordered iteration (**for in**) in line 2 and passing the current position in the iterator to the called function. All points combined it is easier to implement an iterative solution which yields algorithm 5, which scales with $|G| \cdot |S|$.[10] Another (language dependent) argument for an iterative solution is, that `pymatgen` is implemented in Python, where recursive solutions are explicitly not optimized.[11]

Algorithm 4 Generation of a group from a given generator set S.

```
1: function GENERATE_GROUP(S)
2:     for each f, g ∈ S × S do
3:         if fg ∉ S then
4:             return GENERATE_GROUP(S ∪ {fg})          ▷ Recursive call
5:         end if
6:     end for
7:     return S
8: end function
```

In order to compare actual implementations of the algorithms, it has to be emphasized that scaling depends on the number of symmetry elements and not on the number of atoms. For this reason the icosahedral point group I_h gives an upper bound independent of molecule size because I_h is assumed to be the highest symmetry encountered in molecules. Comparing the algorithms 4 and 5 on C_{60}-fullerene which possesses I_h symmetry, shows a speedup by a factor of ≈ 70. Even more important is that algorithm 5 performs so well compared to other numerical bottlenecks, that further improvements as for example Diminio's algorithm would not enhance overall performance.[10]

Algorithm 5 Improved generation of a group from a given generator set S.

```
1: function GENERATE_GROUP(S)
2:     G := COPY(S)                    ▷ Not needed if language defaults to pass-by-value.
3:     for g in G do                   ▷ Note that iterator G is changed during the loop.
4:         for each s ∈ S do
5:             if gs ∉ G then
6:                 append gs to G
7:             end if
8:         end for
9:     end for
10:    return G
11: end function
```

Application example: Modifying the asymmetric unit

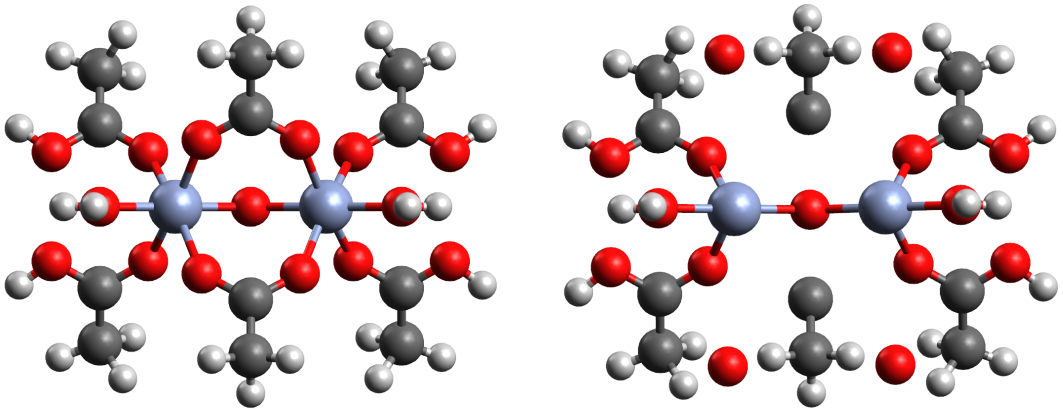
During detection of the symmetry group, symmetry equivalence classes are formed, which are represented by dictionaries of atom labels pointing to sets of atom labels. Another dictionary keeps track of symmetry operations. Its keys are ordered pairs of atom labels pointing to the $\mathbb{R}^{3,3}$-matrix that maps the two atoms from the key unto each other. The `AsymmetricUnitCartesian` representing the asymmetric unit keeps those dictionaries as hidden attributes and exposes only one atom of each symmetry equivalence class. After modifying the asymmetric unit, it can be transformed back to a complete molecule as it was done in code example 2.2 and illustrated in Figure 2.5.

Code example 2.2: Symmetry preserving modifications

```
molecule = cc.Cartesian.read_xyz('MIL53as_cluster.xyz', start_index=1)
asymm_unit = molecule.get_asymmetric_unit()
asymm_unit.view()  # Displayed in Figure 2.5a
asymm_unit.loc[8, ['x', 'y', 'z']] += [2, 3, 2]
asymm_unit.get_cartesian().view()  # Displayed in Figure 2.5b
```

Symmetrisation

For various reasons an overall symmetric molecule may appear to have C_1 symmetry. As discussed in subsubsection 2.4.2 the detection of the symmetry group automatically yields symmetry equivalence classes and the symmetry operations for converting equivalent atoms. Using this information all equivalent atoms are mapped onto the position of one representative atom. Averaging the position yields one best representative of the equivalence class which gets mapped back using the known symmetry operations, to return



(a) Unmodified C_{2v} symmetric cluster.

(b) Symmetry preserving modifications on the cluster from Figure 2.5a.

Figure 2.5.: Comparison of the original and modified version of a MIL53(Cr) cluster while preserving symmetry. Structures were generated by code example 2.2.

a more symmetric molecule. The procedure can be repeated iteratively until the difference between two symmetrisation steps is smaller than a given threshhold. Code example 2.3 shows the distortion and symmetrisation of a C_{2v} symmetric cluster (Figure 2.5a).[12]

Code example 2.3: Point group detection and symmetrisation

```
molecule = cc.Cartesian.read_xyz('MIL53as_cluster.xyz', start_index=1)
print(molecule.get_pointgroup())
np.random.seed(77)
distorted = molecule.copy()
distorted += np.random.randn(len(distorted), 3) / 25
print(distorted.get_pointgroup(tolerance=0.1))
sym = distorted.symmetrize(max_n=25, tolerance=0.3, epsilon=1e-5)['sym_mol']
print(sym.get_pointgroup(tolerance=0.1))
STDOUT:
'C2v'
'C1'
'C2v'
```

2.4.3. Alignment and Interpolating structures

In order to calculate activation barriers or optimize transition states, it might be necessary to interpolate geometries. The most intuitive approach is to interpolate in cartesian space,

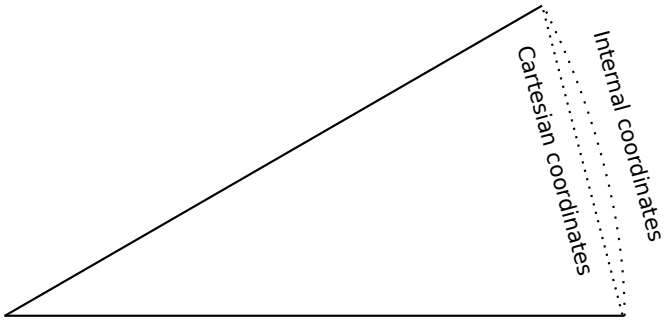


Figure 2.6.: Illustration of the linear angle approximation. The dashed lines represent the interpolated paths.

which assumes the small angle approximation shown in Figure 2.6. If the change of bond angles is large, this may lead to shortened bond lengths and overestimated activation barriers. Another approach is to interpolate in Z-matrix coordinates, which requires the difference between two Z-matrices. In order to have a physically defined difference between Z-matrices, their construction tables have to be the same. If this constraint is fullfilled, the difference of two Z-matrices is straightforwardly the difference of their bonds, angles, and dihedrals. In both cases it is necessary that the start and end structures are labelled in the same way and are aligned i. e. the sum of euclidean distances between same atoms is minimised. In the following section the Kabsch algorithm for aligning molecules will be discussed, followed by an algorithm for labelling aligned molecules. In the end there will be an example for the transformation of the MIL53(Cr) high temperature into the MIL53(Cr) low temperature allotrope.[12]

Alignment and Kabsch Algorithm

The Kabsch Algorithm gives the optimal rotation matrix $\mathbf{U} \in \mathrm{SO}(3)$ to rotate $\mathbf{P} \in \mathbb{R}^{n,3}$ unto $\mathbf{Q} \in \mathbb{R}^{n,3}$ under the assumption that they are both centered around their centroid.[13] The abbreviation $\mathrm{SO}(n)$ is used to denote the special orthogonal group in $\mathbb{R}^{n,n}$. This allows to formally write the problem as:

$$d = \inf_{\mathbf{U} \in \mathrm{SO}(3)} \sum_{i=1}^{n} \left\| (\mathbf{U}\mathbf{P}^{\mathsf{T}})_{:,i} - \mathbf{Q}_{i,:} \right\| \tag{2.4.2}$$

For the algorithm 6 it is assumed that Singular Value Decomposition (SVD) routines are available, to factorice a given matrix $\mathbf{A}$ as: $\mathbf{A} = \mathbf{V}\mathbf{S}\mathbf{W}^{\mathsf{T}}$.

Algorithm 6 Kabsch Algorithm for finding the optimal rotation matrix $\mathbf{U}$.[13]

function GET_OPTIMAL_ROTATION($\mathbf{P}, \mathbf{Q}$)
  $\mathbf{P} := \mathbf{P} - \text{CENTROID}(\mathbf{P})$
  $\mathbf{Q} := \mathbf{Q} - \text{CENTROID}(\mathbf{Q})$
  $\mathbf{A} := \mathbf{P}^\mathsf{T}\mathbf{Q}$
  $\mathbf{V}, \mathbf{S}, \mathbf{W}^\mathsf{T} := \text{SVD}(\mathbf{A})$
  $\mathbf{U} := \mathbf{V} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V}\mathbf{W}^\mathsf{T}) \end{pmatrix} \mathbf{W}^\mathsf{T}$
  return $\mathbf{U}$
end function

Reassignment of labels

One important underlining assumption of the Kabsch algorithm is that the molecules to be aligned are labelled in the same way i. e. the i-th row in $\mathbf{P}$ and $\mathbf{Q}$ denote the same atom's position. This assumption may be wrong. In this case the user has to manually provide a subset of labels for each molelecule to obtain a preliminar alignment of molecules. Using this preliminar alignment the optimal permutation $\mathcal{P}$ of the index I has to be found. Using the abbreviation Aut for the automorphism group i. e. the set of all permutations, the problem may be formally written as:

$$d = \min_{\mathcal{P} \in \text{Aut}(I)} \sum_{i=1}^{n} \left\| \mathbf{P}_{\mathcal{P}(i),:} - \mathbf{Q}_{i,:} \right\| \tag{2.4.3}$$

In algorithm 7 the permutation $\mathcal{P}$ is represented by a dictionary.

Application example

The crystal structure MIL53(Cr) was chosen as application example because it undergoes large structural changes from high to low temperature allotropes without changing connectivity.[12] Truncated clusters of both forms are displayed in Figure 2.7. The alignment and interpolation in both coordinate systems was performed in code example 2.4 which illustrates several important points. First, it is necessary to create both Z-matrices with the same construction table to gain a well defined difference. Secondly, the difference of two Z-matrices may be undefined. For this reason it is, only in this special case, necessary to *not* test the result of the $-$ operator for invalid references. Thirdly, angles are actually equivalence classes in $\mathbb{R}$ that are related via (mod 360°). After calculating the difference in Z-matrix coordinates, it is advantageous to choose the representative with the least

Algorithm 7 Find the optimal permutation that minimizes the distance in equation 2.4.3.

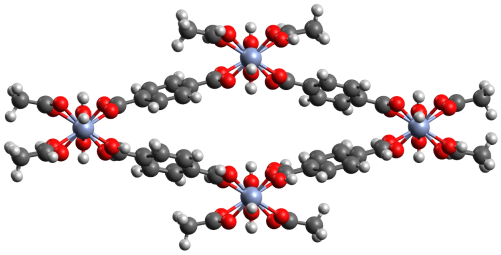
```
1: function GET_OPTIMAL_PERMUTATION(P, Q)
2:     function GET_SHORTEST_DISTANCE(i_Q)
3:         I_P := INDEX(P)
4:         for each i ∈ I_P do
5:             d[i] = ‖P[i,:] − Q[i_Q,:]‖
6:         end for
7:         I_(opt,P) := ARGSORT(d)
8:                 ▷ ARGSORT returns index that sorts d ascending; d is unchanged!
9:         return I_(opt,P), d
10:    end function
11:
12:    I_Q := INDEX(Q)
13:    for i_Q in I_Q do                          ▷ Iterator may change during loop
14:        I_(opt,P), d := GET_SHORTEST_DISTANCE(i_Q)
15:        for j in I_(opt,P) do
16:            if j not in KEYS(P) then
17:                P[j] := i_Q
18:                break
19:            else if d[j] < ‖P[j,:] − Q[P[j],:]‖ then
20:                append P[j] to I_Q
21:                P[j] := i_Q
22:                break
23:            end if
24:        end for
25:    end for
26:    return P
27: end function
```

absolute value (For example $-10°$ instead of $350°$).

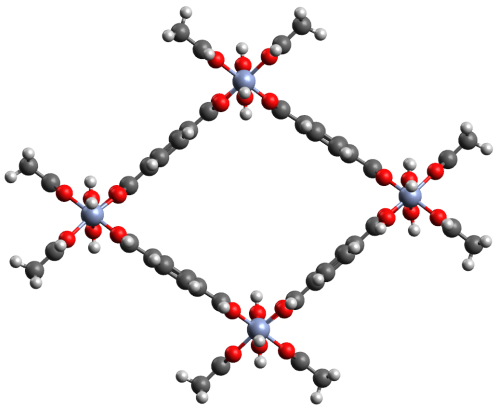
Code example 2.4: Alignment and interpolation of allotropes.

```
MIL53_lt = Cartesian.read_xyz('MIL53_lt.xyz')
MIL53_ht = Cartesian.read_xyz('MIL53_ht.xyz')
MIL53_lt, MIL53_ht = MIL53_lt.align(MIL53_ht)
MIL53_lt.view()  # Displayed in Figure 2.7a
MIL53_ht.view()  # Displayed in Figure 2.7b
steps = 20
delta = MIL53_ht - MIL53_lt
cartesians = [MIL53_lt + delta * i / (steps - 1) for i in range(steps)]
Z_MIL53_lt = MIL53_lt.get_zmat()
construction_table = Z_MIL53_lt.loc[:, ['b', 'a', 'd']]
Z_MIL53_ht = MIL53_ht.get_zmat(construction_table)
with cc.TestOperators(False):
    delta = (Z_MIL53_ht - Z_MIL53_lt).minimize_dihedrals()
    Z matrices = [Z_MIL53_lt + delta * i / (steps - 1) for i in range(steps)]
```

Interpolating the MIL53(Cr) allotropes (Figure 2.7) in cartesian coordinates shortens certain C−C bond lengths by up to 0.05 Å, while they are unchanged in Z-matrix coordinates.



(a) Truncated cluster of low temperature MIL53(Cr).



(b) Truncated cluster of high temperature MIL53(Cr).

Figure 2.7.: Two truncated clusters of different allotropes of MIL53(Cr). They were used for the interpolation of structures in code example 2.4

3. The ChemOpt library

3.1. Unconstrained Optimisation

In order to find the equilibrium geometry of a molecule, its potential hyper surface (PHS) is optimized in Z-matrix coordinates using the Broyden-Fletcher–Goldfarb-Shanno (BFGS) algorithm.[14] This requires the potential energy gradient expressed in internal coordinates, which is obtained by using the analytical gradient of the coordinate transformations from subsection 2.2.3 to transform the energy gradient expressed in cartesian coordinates. Denoting the Hadamard product with $\odot$ and the sum of all matrix elements with grandsum(...) allows to write a generalized version of equation 2.2.2:

$$\left(\frac{\partial V_C}{\partial \mathbf{C}}(\mathbf{C}_0)\right)_{i,j} = \text{grandsum}\left(\left(\frac{\partial V_X}{\partial \mathbf{X}}(\mathbf{X}(\mathbf{C}_0))\right)^{\mathsf{T}} \odot \left(\frac{\partial \mathbf{X}}{\partial \mathbf{C}}(\mathbf{C}_0)\right)_{:,:,i,j}\right) \tag{3.1.1}$$

This equation assumes the coordinate layout defined in subsection 2.2.1, so the gradient in internal coordinates $\frac{\partial V_C}{\partial \mathbf{C}}$ has the following form:

$$\frac{\partial V_C}{\partial \mathbf{C}} = \begin{pmatrix} \frac{\partial V}{\partial r_1} & \frac{\partial V}{\partial \alpha_1} & \frac{\partial V}{\partial \delta_1} \\ \frac{\partial V}{\partial r_2} & \frac{\partial V}{\partial \alpha_2} & \frac{\partial V}{\partial \delta_2} \\ \frac{\partial V}{\partial r_3} & \frac{\partial V}{\partial \alpha_3} & \frac{\partial V}{\partial \delta_3} \\ & \vdots & \end{pmatrix} \tag{3.1.2}$$

The generic BFGS algorithm can be tailored for the context of molecular optimizations, since it is assumed that the PHS exhibits certain smoothness and curvature around the equilibrium structure. Examples of such specialisations include a parametrised model Hessian[15], or a parametrised step length instead of a line search.[16] These improvements which are readily available in redundant internal coordinates can not be used in general in a Z-matrix. The main obstacle is that the potential energy function expressed in Z-matrix coordinates is potentially a lot non-smoother than in redundant internal coordinates. For these reasons the generic BFGS optimicer from the SciPy project was used, which performs a line search imposing strong Wolfe conditions.[14,17] The non-smoothness can be

explained intuitively by looking at the matrix notation of $\frac{\partial V_C}{\partial \mathbf{C}}$ in equation 3.1.2. Changes of bonds, angles, and dihedrals in upper rows of the Z-matrix propagate through the whole molecule and can have a large impact on the energy. If a Z-matrix is visualised as tree structure, it becomes clear that the number of atoms in the n-th coordination sphere grows approximately exponentially with n. Hence any change in a upper row of the Z-matrix affects an exponentially growing number of successing atoms which yields large and fast changing absolute values for the gradient in upper rows of $\frac{\partial V_C}{\partial \mathbf{C}}$.

A set of 30 molecules, widely used in the literature for benchmarking purposes, was optimized using Z-matrix coordinates and the generic BFGS optimisation function from SciPy.[16–18] The electronic calculations were performed using MOLPRO 2012.1 with the same level of theory as in the benchmark publications from Table 3.1: Restricted Hartree Fock on a STO-3G basis set.[19,20] Since an unconstrained optimization should work as "black-box" method, the Z-matrices were used as automatically generated by `chemcoord` on the basis of algorithm 3. The convergence criteria followed the literature: $1 \cdot 10^{-6}\,E_\mathrm{h}$ was the allowed difference in energy and $6 \cdot 10^{-4}\,E_\mathrm{h}\,\text{Å}^{-1}$ the allowed value for the maximum norm of the gradient in cartesian coordinates.[18] On this basis Table 3.1 shows a disillusioning result for the optimization in Z-matrix coordinates. If the optimization converges, it is outperformed by the tailored optimization procedures given in the literature. and more than often it does not even converge. This is mainly due to hidden coupling of angles appearing in ring systems and is well described in the literature.[16,22,23] If for example the Z-matrix of 1,3,5-trifluorobenzene is created manually as seen in Table 3.2 and only the three bond lengths r_{CC}, r_{CH}, and r_{CF} are optimized from the same starting geometry as in Table 3.1, the optimization converges smoothly after 9 iterations. But in this case it is not a "black-box" method anymore.

3.2. Constrained Optimisation

After the results of the previous section it might be asked why Z-matrix coordinates should be used for optimizations at all. The answer is that they allow certain non-linear constraints, which are (at least up to this point) not possible in other coordinate systems. In this context it is worth noting that Baker, Kessi, and Delley have shown that it is possible to include linear constraints in delocalized internal coordinates.[24]

For the purpose of abstracting equation 3.1.1 to include non-linear constraints, the vector $\mathbf{p}$ is defined as an arbitrary length vector containing only expressions differentiable by the symbolic algebra package of use. A Z-matrix $\mathbf{C}$ containing symbolic expressions can then be written as function $\mathbf{C}(\mathbf{p})$. The Z-matrix in Table 3.3 is for example a function

Table 3.1.: Number of electronic single point calculations required by each approach tested. Missing entries denote a failure to converge.

Molecule	This work[a]	Helgaker[16]	Eckert[21]	Lindh[15]	Baker[18]
Water	8	4	4	4	6
Ammonia	10	5	6	5	6
Ethane		3	4	4	5
Acetylene		4	6	5	6
Allene		4	4	5	5
Hydroxysulphane	19	7	7	8	8
Benzene		3	3	3	4
Methylamine	9	4	5	5	6
Ethanol	11	4	5	5	6
Acetone	16	4	5	5	6
Disilylether		8	9	11	8
1,3,5-trisilacyclohexane		9	6	8	8
Benzaldehyde		4	5	5	6
1,5-difluoronaphtalene		4	5	5	5
1,3,5-trifluorobenzene		4	4	4	5
Neopentane	32	4	4	5	5
Furan	29	5	6	7	8
Naphtalene		5	6	6	5
1,3-difluorobenzene		5	6	6	6
2-hydroxybicyclopentane		9[b]	9	10	15
ACHTAR10		8	9	8	12
ACANIL01		7	8	8	8
Benzidine		9	7	10	9
Pterin		8	9	9	10
Difuropyrazine		6	7	7	9
Mesityloxide		5	6	6	7
Histidine		16	14	20	19
Dimethylpentane		9	10	10	12
Caffeine		6	7	7	12
Menthone		12	10	14	13

[a] The generic BFGS optimization from SciPy executes the given function twice with the same parameters in the beginning to test determinicity. Since coordinate systems and not implementations are compared, one iteration was substracted.
[b] Did not converge to correct geometry.

Table 3.2.: Manually created Z-matrix of 1,3,5-trifluorobenzene.

	atom	b	bond	a	angle	d	dihedral
4	C	$\vec{0}$	1.390	$\vec{e_z}$	90.000	$\vec{e_x}$	-120.000
3	C	4	r_{CC}	$\vec{e_z}$	66.064	$\vec{e_x}$	-33.255
5	C	4	r_{CC}	3	120.024	$\vec{e_x}$	-0.000
2	C	3	r_{CC}	4	119.976	5	-0.000
6	C	5	r_{CC}	4	119.976	3	-0.000
1	C	6	r_{CC}	5	120.024	4	-0.000
7	F	3	r_{CF}	4	120.012	5	-180.000
8	F	5	r_{CF}	4	120.012	3	-180.000
9	F	1	r_{CF}	6	120.012	5	-180.000
10	H	2	r_{CH}	3	119.988	4	-180.000
11	H	4	r_{CH}	5	119.988	3	-180.000
12	H	6	r_{CH}	5	119.988	4	-180.000

of the two-dimensional vector $(t, r_{CH})^\mathsf{T}$. The potential function V_p expressed in terms of the parameter vector $\mathbf{p}$ is then:

$$\begin{aligned} V_p &:= V_C \circ C = V_X \circ \mathbf{X} \circ \mathbf{C} \\ V_p(\mathbf{p}) &:= V_X(\mathbf{X}(\mathbf{C}(\mathbf{p}))) \end{aligned} \tag{3.2.1}$$

Deriving this function for the m-th entry in $\mathbf{p}$ yields:

$$\begin{aligned} &\left(\frac{\partial V_p}{\partial \mathbf{p}}(\mathbf{p}_0)\right)_{1,m} = \frac{\partial V_p}{\partial \mathbf{p}_m}(\mathbf{p}_0) = \nabla V_p(\mathbf{p}_0) \\ &= \sum_{i=1}^{3}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{3}\left(\left(\frac{\partial V_X}{\partial \mathbf{X}}(\mathbf{X}(\mathbf{C}(\mathbf{p}_0)))\right)_{j,i} \cdot \left(\frac{\partial \mathbf{X}}{\partial \mathbf{C}}(\mathbf{C}(\mathbf{p}_0))\right)_{i,j,k,l} \cdot \left(\frac{\partial \mathbf{C}}{\partial \mathbf{p}}(\mathbf{p}_0)\right)_{l,k,m}\right) \\ &= \sum_{i=1}^{3}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{3}\left(\left(\frac{\partial V_X}{\partial \mathbf{X}_{i,j}}(\mathbf{X}(\mathbf{C}(\mathbf{p}_0)))\right) \cdot \left(\frac{\partial \mathbf{X}_{i,j}}{\partial \mathbf{C}_{l,k}}(\mathbf{C}(\mathbf{p}_0))\right) \cdot \left(\frac{\partial \mathbf{C}_{l,k}}{\partial \mathbf{p}_m}(\mathbf{p}_0)\right)\right) \end{aligned} \tag{3.2.2}$$

The first factor $\frac{\partial V_X}{\partial \mathbf{X}}$ is returned from the electronic calculations and the second factor $\frac{\partial \mathbf{X}}{\partial \mathbf{C}}$ was derived in subsection 2.2.3. For the factor $\frac{\partial \mathbf{C}}{\partial \mathbf{p}}$, we note the previous restriction that only differentiable expressions are allowed as Z-matrix entries and use a symbolic algebra package. For the usual applications in chemistry the restriction of differentiability should be of no concern.

With the potential function from equation 3.2.1 and its gradient from equation 3.2.2

the BFGS algorithm can be used to optimize a Z-matrix having non-linear expressions to reduce the degrees of freedom. An example is given in Table 3.3, which is the already known Z-matrix of 2-methylpropane from Table 2.3 but with symbolic expressions. The parameter t denotes a coupled movement of C−C stretches and C−C−C bends with one C−C bond growing exponentially. The parameter r_{CH} denotes the C−H bond lengths. The output of simultaneously optimizing t and r_{CH} with starting values 0. and 1. is

Table 3.3.: Z-matrix of 2-methylpropane (Figure 2.4a) with non-linear constraints. Bond lengths in Å, angles in degrees.

	atom	b	bond	a	angle	d	dihedral
5	C	$\vec{0}$	0.3728	$\vec{e_z}$	66.392	$\vec{e_x}$	-100.576
2	C	5	$e^t + 1.5$	$\vec{e_z}$	95.608	$\vec{e_x}$	-10.577
11	C	5	$t + 1.5$	2	$180/\pi \arcsin(t) + 110$	$\vec{e_x}$	-99.902
7	C	5	$t + 1.5$	2	$180/\pi \arcsin(t) + 110$	11	122.638
6	H	5	r_{CH}	2	108.405	11	-118.668
1	H	2	r_{CH}	5	111.344	11	178.701
3	H	2	r_{CH}	5	111.575	11	-61.305
4	H	2	r_{CH}	5	111.338	11	58.672
12	H	11	r_{CH}	5	111.582	2	61.294
13	H	11	r_{CH}	5	111.348	2	-58.695
14	H	11	r_{CH}	5	111.348	2	-178.720
8	H	7	r_{CH}	5	111.343	2	58.688
9	H	7	r_{CH}	5	111.567	2	-61.303
10	H	7	r_{CH}	5	111.344	2	178.716

shown in section A.3. The calculation converges after 11 iterations. If the redundant step performed by the SciPy BFGS function is neglected, it can be counted as 10 iterations.

4. Conclusion

With the `chemcoord` library an important tool for the theoretical chemist was implemented. The creation of a chemically useful Z-matrix from cartesian coordinates was on the one hand automated to become a black-box method and on the other hand allows manual control over each step in the transformation if so desired. The user can be sure that a Z-matrix is created and remains valid troughout its existence and does not have to worry about manual dummy atom insertion. Heuristics for the selection of references were implemented to create a Z-matrix in the same way as a human chemist would usually define it. Gradients for the interconversion of Z-matrix and cartesian coordinates were analytically derived and implemented. In addition to the tools for handling the conversion of Z-matrices a stack of functions and methods were implemented that are in general of great use in the preparation of structures in the field of theoretical chemistry. It was the explicit aim to not only implement algorithms, but to write a performant, well tested, and well documented library with an intuitive and stable API, which is further discussed in section A.5. The library is available through the PyPi package manager which unfortunately does not provide download statistics. But it can be seen as a sign of growing popularity that there are around five unique clones of the repository per week, 18 raised issues by other GitHub users, as well as four pull requests in total.

The possibility to perform non-linear constrained optimizations in Z-matrix coordinates was implemented in the library `chemopt`. As expected from literature the unconstrained optimization in Z-matrix coordinates is outperformed by optimization in redundant internal coordinates, but non-linear constraints allow to reduce degrees of freedom in such a way that optimizations of large molecules can become much cheaper.

There are two remaining open questions. First, if it is possible to use certain assumptions about the PHS expressed in Z-matrix coordinates to tailor the BFGS algorithm to parametrise the step length instead of performing a line search. Secondly, if it is possible to efficiently generalize the ansatz for non linearly constrained optimizations to redundant or delocalized internal coordinates. Especially the second question would greatly enhace the available toolbox of a theoretical chemist.

A. Appendix

A.1. Abbreviations

Table A.1.: Used Abbreviations or function names

Abbreviation	Full word
PHS	Potential Hyper Surface.
SN-NeRF	Self-Normalizing Natural Extension Reference Frame.[3]
SVD	Singular Value Decomposition.
BFGS	Broyden-Fletcher–Goldfarb-Shanno algorithm.[14]
SO(n)	Special orthogonal group of $\mathbb{R}^{n,n}$ matrices.
Aut(S)	Automorphism group of S. Set of all permutations of S.
sgn	Returns the sign of a number.
API	Application programming interface.

A.2. Setup for code examples

Unless otherwise stated, the code examples given throughout this work can be executed, if the following header of imported modules is used.

Code example A.1: Necessary Header.

```
import numpy as np
import pandas as pd
import numba as nb
import sympy
sympy.init_printing()
import chemcoord as cc
from chemcoord import Cartesian, Zmat
import chemopt as co
```

The `chemcoord` library is packaged in the PyPi package index and can be installed via `pip install chemcoord==2.0.3`. The documentation is hosted under the following URL: `http://chemcoord.readthedocs.io/`. The installation of `chemcoord` automatically resolves all its dependencies. The `chemopt` library is not yet (as of 2017-10-27) packaged in the PyPi package index. This makes it necessary to execute the following commands:

```
git clone "https://github.com/mcocdawc/chemopt/tree/v0.1.0"
cd chemopt
pip install .
```

The installation of `chemopt` automatically resolves all its dependencies. The documentation is hosted under the following URL: `http://chemopt.readthedocs.io/en/latest/`. If a manual creation of library environments should be necessary, the used versions can be seen in Table A.2. In the examples Python 3 syntax will be used, although the libraries

Table A.2.: Versions of used libraries and programs.

Library/Program	Version
Python	3.6.2
Numpy	1.13.1
Pandas	0.20.3
Numba	0.35.0
SymPy	1.1.1
ChemCoord	2.0.3
ChemOpt	0.1.0

are compatible to both Python 2.7 and 3.x. The necessary xyz-files are appended in A.6.

A.3. Example output of Chemopt

ChemOpt produces markdown output files, which can be automatically converted to PDF files for presentation purposes. The following four PDF pages are the output of a non-linearly constrained optimization of 2-methylpropane

This is ChemOpt 0.1.0 optimising a molecule in internal coordinates.

Written by Oskar Weser (oskar.weser@gmail.com)

Input File

```
#!/usr/bin/env python

from chemopt.zmat_optimisation import optimise
from chemcoord import Cartesian
from sympy import asin, exp, deg, symbols
from functools import partial

r_CH, t = symbols('r_CH, t', real=True)
molecule = Cartesian.read_xyz('2_methylpropane.xyz', start_index=1)

zmolecule = molecule.get_zmat()
zmolecule.safe_loc[2, 'bond'] = exp(t) + 1.5
zmolecule.safe_loc[[11, 7], ['bond', 'angle']] = t + 1.5, deg(asin(t)) + 110
zmolecule.safe_loc[zmolecule['atom'] == 'H', 'bond'] = r_CH

f = partial(optimise, hamiltonian='RHF', basis='STO-3G', coord_fmt='.2f')
calculated = f(zmolecule, symbols=[(t, 0), (r_CH, 1.)])
```

Settings for the calculations

Backend	molpro
Hamiltonian	RHF
Basis	STO-3G
Charge	0
Spin multiplicity	1
Convergence energy	1e-07
Convergence gradient	0.0005
Maximum of iterations	100
Number of processes	2
Memory per process	200Mb

Starting structures

Starting structure as Zmatrix

	atom	b	bond	a	angle	d	dihedral
5	C	$\vec{0}$	0.37	$\vec{e_z}$	66.39	$\vec{e_x}$	-100.58
2	C	5	$e^t + 1.5$	$\vec{e_z}$	95.61	$\vec{e_x}$	-10.58
11	C	5	$t + 1.5$	2	$\frac{180}{\pi} \arcsin(t) + 110$	$\vec{e_x}$	-99.9
7	C	5	$t + 1.5$	2	$\frac{180}{\pi} \arcsin(t) + 110$	11	122.64
6	H	5	r_{CH}	2	108.41	11	-118.67
1	H	2	r_{CH}	5	111.34	11	178.7
3	H	2	r_{CH}	5	111.57	11	-61.31
4	H	2	r_{CH}	5	111.34	11	58.67
12	H	11	r_{CH}	5	111.58	2	61.29
13	H	11	r_{CH}	5	111.35	2	-58.7
14	H	11	r_{CH}	5	111.35	2	-178.72
8	H	7	r_{CH}	5	111.34	2	58.69
9	H	7	r_{CH}	5	111.57	2	-61.3
10	H	7	r_{CH}	5	111.34	2	178.72

Symbols with starting values

Symbol	Start Value
t	0
r_{CH}	1

Iterations

Starting 2017-11-05T15:28:18

n	$E[E_h]$	$\Delta E[E_h]$	$\max(\lvert\nabla E\rvert)[E_h/\text{Å}]$
1	-155.1651796517	0.0000000000	1.8389522098
2	-155.1651796517	0.0000000000	1.8389522098
3	-153.0920917531	2.0730878986	2.2931335015
4	-155.2240304648	-2.1319387117	0.7029751059
5	-155.1224960402	0.1015344246	1.3189340368
6	-155.2310838021	-0.1085877619	0.6701393859
7	-155.2417188950	-0.0106350929	0.3818829525
8	-155.2465947366	-0.0048758416	0.0899002578
9	-155.2468244570	-0.0002297204	0.0034643899
10	-155.2468251019	-0.0000006449	0.0000895282

n	$E[E_h]$	$\Delta E[E_h]$	$\max(\lvert\nabla E\rvert)[E_h/\text{Å}]$
11	-155.2468251242	-0.0000000223	0.0000014321

Optimised Structures

Symbols with end values

Symbol	End Value
t	-0.0539243
r_{CH}	1.08874

Optimised structure as Zmatrix

	atom	b	bond	a	angle	d	dihedral
5	C	$\vec{0}$	0.37	$\vec{e_z}$	66.39	$\vec{e_x}$	-100.58
2	C	5	2.45	$\vec{e_z}$	95.61	$\vec{e_x}$	-10.58
11	C	5	1.45	2	106.91	$\vec{e_x}$	-99.9
7	C	5	1.45	2	106.91	11	122.64
6	H	5	1.09	2	108.41	11	-118.67
1	H	2	1.09	5	111.34	11	178.7
3	H	2	1.09	5	111.57	11	-61.31
4	H	2	1.09	5	111.34	11	58.67
12	H	11	1.09	5	111.58	2	61.29
13	H	11	1.09	5	111.35	2	-58.7
14	H	11	1.09	5	111.35	2	-178.72
8	H	7	1.09	5	111.34	2	58.69
9	H	7	1.09	5	111.57	2	-61.3
10	H	7	1.09	5	111.34	2	178.72

Optimised structure in cartesian coordinates

	atom	x	y	z
5	C	-0.06	-0.34	0.15
2	C	2.32	-0.15	-0.36
11	C	-0.77	-0.21	-1.11
7	C	-0.37	0.7	1.11
6	H	-0.25	-1.32	0.58
1	H	2.93	-0.26	0.54

	atom	x	y	z
3	H	2.57	0.82	-0.8
4	H	2.64	-0.92	-1.06
12	H	-0.59	0.76	-1.57
13	H	-0.46	-0.98	-1.81
14	H	-1.85	-0.3	-0.96
8	H	0.23	0.59	2.01
9	H	-0.18	1.69	0.7
10	H	-1.42	0.67	1.4

Closing

Structures were written to 2_methylpropane_opt.molden.

The calculation finished successfully after 11 iterations at: 2017-11-05T15:28:49 and needed: 0:00:30.

A.4. Derivatives of the basis tensor B

For the derived gradients in section 2.2.3 and 2.2.5 an expression for $\frac{\partial \mathbf{B}_{:,:,k}}{\partial \mathbf{X}^R_{:,(b(k),a(k),d(k))}}$ is still required. The derivatives of the $\mathbf{B}$ tensor (equation 2.2.5) were calculated with the help of the symbolic algebra library SymPy.[8] In order to write the equations shorter, the following variables were defined for a given k:

$$\begin{aligned} \mathbf{ba} &:= \mathbf{X}_{:,a(k)} - \mathbf{X}_{:,b(k)} & (x_b, y_b, z_b)^\mathsf{T} &:= \mathbf{X}_{:,b(k)} \\ \mathbf{ad} &:= \mathbf{X}_{:,d(k)} - \mathbf{X}_{:,a(k)} & (x_a, y_a, z_a)^\mathsf{T} &:= \mathbf{X}_{:,a(k)} \\ & & (x_d, y_d, z_d)^\mathsf{T} &:= \mathbf{X}_{:,d(k)} \end{aligned}$$

The results are:

$$\begin{aligned} \left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,1,k,b(k),1} &= \frac{\partial \mathbf{B}_{1,1,k}}{\partial \mathbf{X}_{1,b(k)}} \\ &= \frac{1}{\|\mathbf{ad} \times \mathbf{ba}\|^3 \|\mathbf{ba}\|^3} (\Big(x_a - x_b)((y_a - y_b)((x_a - x_b)(y_a - y_d) - (x_a - x_d)(y_a - y_b)) \\ &\quad + (z_a - z_b)((x_a - x_b)(z_a - z_d) - (x_a - x_d)(z_a - z_b))) \left(\Big((x_a - x_b)(y_a - y_d) - (x_a - x_d)(y_a - y_b)\right)^2 \\ &\quad + ((x_a - x_b)(z_a - z_d) - (x_a - x_d)(z_a - z_b))^2 + ((y_a - y_b)(z_a - z_d) - (y_a - y_d)(z_a - z_b))^2\Big) \\ &\quad - ((y_a - y_b)(y_a - y_d) + (z_a - z_b)(z_a - z_d)) \left(\Big(x_a - x_b\right)^2 + (y_a - y_b)^2 \\ &\quad + (z_a - z_b)^2\Big) \left(\Big((x_a - x_b)(y_a - y_d) - (x_a - x_d)(y_a - y_b)\right)^2 + ((x_a - x_b)(z_a - z_d) - (x_a - x_d)(z_a - z_b))^2 \\ &\quad + ((y_a - y_b)(z_a - z_d) - (y_a - y_d)(z_a - z_b))^2\Big) + ((y_a - y_b)((x_a - x_b)(y_a - y_d) - (x_a - x_d)(y_a - y_b)) \\ &\quad + (z_a - z_b)((x_a - x_b)(z_a - z_d) - (x_a - x_d)(z_a - z_b))) \left((y_a - y_d)((x_a - x_b)(y_a - y_d) - (x_a - x_d)(y_a - y_b))\right. \\ &\quad \left. + (z_a - z_d)((x_a - x_b)(z_a - z_d) - (x_a - x_d)(z_a - z_b))) \left(\Big(x_a - x_b\right)^2 + (y_a - y_b)^2 + (z_a - z_b)^2\Big)\right) \end{aligned}$$

$$
\begin{aligned}
\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,1,k,b(k),2} &= \frac{\partial \mathbf{B}_{1,1,k}}{\partial \mathbf{X}_{2,b(k)}} \\
&= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big(y_a-y_b)((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad +(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\left(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)^2 \\
&\quad +((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big) \\
&\quad +((-x_a+x_b)(y_a-y_d)+(2x_a-2x_d)(y_a-y_b))\left(\Big(x_a-x_b\right)^2+(y_a-y_b)^2 \\
&\quad +(z_a-z_b)^2\Big)\left(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)^2+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2 \\
&\quad +((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)-((x_a-x_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad -(z_a-z_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad +(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\left(\Big(x_a-x_b\right)^2+(y_a-y_b)^2+(z_a-z_b)^2\Big)\Big)
\end{aligned}
$$

$$
\begin{aligned}
\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,1,k,b(k),3} &= \frac{\partial \mathbf{B}_{1,1,k}}{\partial \mathbf{X}_{3,b(k)}} \\
&= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big(z_a-z_b)((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad +(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\left(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)^2 \\
&\quad +((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big) \\
&\quad +((-x_a+x_b)(z_a-z_d)+(2x_a-2x_d)(z_a-z_b))\left(\Big(x_a-x_b\right)^2+(y_a-y_b)^2 \\
&\quad +(z_a-z_b)^2\Big)\left(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)^2+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2 \\
&\quad +((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)-((x_a-x_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)) \\
&\quad +(y_a-y_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad +(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\left(\Big(x_a-x_b\right)^2+(y_a-y_b)^2+(z_a-z_b)^2\Big)\Big)
\end{aligned}
$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,1,k,a(k),1} = \frac{\partial \mathbf{B}_{1,1,k}}{\partial \mathbf{X}_{1,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big(-x_a+x_b)((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)$$
$$+((y_a-y_b)(y_b-y_d)+(z_a-z_b)(z_b-z_d))\,(\big(x_a-x_b\big)^2+(y_a-y_b)^2$$
$$+(z_a-z_b)^2)\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2$$
$$+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)-((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\,((y_b-y_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_b-z_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\,(\big(x_a-x_b\big)^2+(y_a-y_b)^2+(z_a-z_b)^2)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,1,k,a(k),2} = \frac{\partial \mathbf{B}_{1,1,k}}{\partial \mathbf{X}_{2,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big(-y_a+y_b)((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)$$
$$+((x_b-x_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_b-z_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\,(\big(x_a-x_b\big)^2+(y_a-y_b)^2+(z_a-z_b)^2)$$
$$+((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)-(x_b-x_d)(y_a-y_b))\,(\big(x_a-x_b\big)^2+(y_a-y_b)^2$$
$$+(z_a-z_b)^2)\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2$$
$$+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,1,k,a(k),3} = \frac{\partial \mathbf{B}_{1,1,k}}{\partial \mathbf{X}_{3,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big(-z_a+z_b)((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\left(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)^2$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)$$
$$+((x_b-x_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_b-y_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\left((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\left(\Big(x_a-x_b\Big)^2+\left(y_a-y_b\right)^2+\left(z_a-z_b\right)^2\right)$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)-(x_b-x_d)(z_a-z_b)\right)\left(\Big(x_a-x_b\Big)^2+\left(y_a-y_b\right)^2\right.$$
$$\left.+(z_a-z_b)^2\right)\left(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2\right.$$
$$\left.+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)\right)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,1,k,d(k),1} = \frac{\partial \mathbf{B}_{1,1,k}}{\partial \mathbf{X}_{1,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|}(-\Big((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))\Big)^2$$
$$+\left(\Big(y_a-y_b\Big)^2+\left(z_a-z_b\right)^2\right)\left(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2\right.$$
$$\left.+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)\right)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,1,k,d(k),2} = \frac{\partial \mathbf{B}_{1,1,k}}{\partial \mathbf{X}_{2,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|}(\Big(-x_a+x_b)(y_a-y_b)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)$$
$$+\big((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))\big)\big((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))\big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,1,k,d(k),3} = \frac{\partial \mathbf{B}_{1,1,k}}{\partial \mathbf{X}_{3,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|}(\Big(-x_a+x_b)(z_a-z_b)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)$$
$$+\big((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))\big)\big((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))\big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,2,k,b(k),1} = \frac{\partial \mathbf{B}_{1,2,k}}{\partial \mathbf{X}_{1,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))\big((y_a-y_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))\big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,2,k,b(k),2} = \frac{\partial \mathbf{B}_{1,2,k}}{\partial \mathbf{X}_{2,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(\Big(-z_a+z_d)(\Big(\big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2$$
$$+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)-\big((x_a-x_d)\big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)$$
$$-(z_a-z_d)\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)\big)\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,2,k,b(k),3} = \frac{\partial \mathbf{B}_{1,2,k}}{\partial \mathbf{X}_{3,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(\Big(y_a-y_d)(\Big(\big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2$$
$$+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)-\big((x_a-x_d)\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)$$
$$+(y_a-y_d)\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)\big)\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,2,k,a(k),1} = \frac{\partial \mathbf{B}_{1,2,k}}{\partial \mathbf{X}_{1,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(-\big(y_a-y_b)(z_a-z_d)+(y_a-y_d)(z_a-z_b)\big)\big((y_b-y_d)\big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)$$
$$+(z_b-z_d)\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)\big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,2,k,a(k),2} = \frac{\partial \mathbf{B}_{1,2,k}}{\partial \mathbf{X}_{2,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(\Big(z_b-z_d)(\Big(\big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2$$
$$+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)+\big((x_b-x_d)\big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)$$
$$-(z_b-z_d)\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)\big)\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,2,k,a(k),3} = \frac{\partial \mathbf{B}_{1,2,k}}{\partial \mathbf{X}_{3,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(\Big(-y_b+y_d)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2$$
$$+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)+((x_b-x_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_b-y_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,2,k,d(k),1} = \frac{\partial \mathbf{B}_{1,2,k}}{\partial \mathbf{X}_{1,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(-(y_a-y_b)(z_a-z_d)+(y_a-y_d)(z_a-z_b))\,((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,2,k,d(k),2} = \frac{\partial \mathbf{B}_{1,2,k}}{\partial \mathbf{X}_{2,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(\Big(z_a-z_b)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2$$
$$+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)+((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,2,k,d(k),3} = \frac{\partial \mathbf{B}_{1,2,k}}{\partial \mathbf{X}_{3,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(\Big(-y_a+y_b)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2$$
$$+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)+((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,3,k,b(k),1} = \frac{\partial \mathbf{B}_{1,3,k}}{\partial \mathbf{X}_{1,b(k)}} = \frac{1}{\|\mathbf{ba}\|^3}(\big(y_a - y_b\big)^2 + (z_a - z_b)^2\big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,3,k,b(k),2} = \frac{\partial \mathbf{B}_{1,3,k}}{\partial \mathbf{X}_{2,b(k)}} = \frac{1}{\|\mathbf{ba}\|^3}(-x_a + x_b)(y_a - y_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,3,k,b(k),3} = \frac{\partial \mathbf{B}_{1,3,k}}{\partial \mathbf{X}_{3,b(k)}} = \frac{1}{\|\mathbf{ba}\|^3}(-x_a + x_b)(z_a - z_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,3,k,a(k),1} = \frac{\partial \mathbf{B}_{1,3,k}}{\partial \mathbf{X}_{1,a(k)}} = \frac{1}{\|\mathbf{ba}\|^3}(-\big(y_a - y_b\big)^2 - (z_a - z_b)^2\big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,3,k,a(k),2} = \frac{\partial \mathbf{B}_{1,3,k}}{\partial \mathbf{X}_{2,a(k)}} = \frac{1}{\|\mathbf{ba}\|^3}(x_a - x_b)(y_a - y_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,3,k,a(k),3} = \frac{\partial \mathbf{B}_{1,3,k}}{\partial \mathbf{X}_{3,a(k)}} = \frac{1}{\|\mathbf{ba}\|^3}(x_a - x_b)(z_a - z_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,3,k,d(k),1} = \frac{\partial \mathbf{B}_{1,3,k}}{\partial \mathbf{X}_{1,d(k)}} = 0$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,3,k,d(k),2} = \frac{\partial \mathbf{B}_{1,3,k}}{\partial \mathbf{X}_{2,d(k)}} = 0$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{1,3,k,d(k),3} = \frac{\partial \mathbf{B}_{1,3,k}}{\partial \mathbf{X}_{3,d(k)}} = 0$$

$$\begin{aligned}
\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,1,k,b(k),1} &= \frac{\partial \mathbf{B}_{2,1,k}}{\partial \mathbf{X}_{1,b(k)}} \\
&= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big(-x_a+x_b)((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad -(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2 \\
&\quad +((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big) \\
&\quad -((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad -(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((y_a-y_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad +(z_a-z_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\,(\big(x_a-x_b\big)^2+\big(y_a-y_b\big)^2+\big(z_a-z_b\big)^2\big)+(-(x_a-x_d)(y_a-y_b) \\
&\quad +(2x_a-2x_b)(y_a-y_d))\,(\big(x_a-x_b\big)^2+\big(y_a-y_b\big)^2+\big(z_a-z_b\big)^2\big)\,(\Big(\big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2 \\
&\quad +((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)\Big)
\end{aligned}$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,1,k,b(k),2} = \frac{\partial \mathbf{B}_{2,1,k}}{\partial \mathbf{X}_{2,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big(-y_a+y_b)((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)$$
$$-((x_a-x_b)(x_a-x_d)+(z_a-z_b)(z_a-z_d))\,(\big(x_a-x_b\big)^2+(y_a-y_b)^2$$
$$+(z_a-z_b)^2)\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2$$
$$+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)+((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((x_a-x_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\big(x_a-x_b\big)^2+(y_a-y_b)^2+(z_a-z_b)^2)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,1,k,b(k),3} = \frac{\partial \mathbf{B}_{2,1,k}}{\partial \mathbf{X}_{3,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big(-z_a+z_b)((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)$$
$$+((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((x_a-x_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\big(x_a-x_b\big)^2+(y_a-y_b)^2+(z_a-z_b)^2\big)+((-y_a+y_b)(z_a-z_d)$$
$$+(2y_a-2y_d)(z_a-z_b))\,(\big(x_a-x_b\big)^2+(y_a-y_b)^2+(z_a-z_b)^2)\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)\Big)$$

$$
\begin{aligned}
\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,1,k,a(k),1} &= \frac{\partial \mathbf{B}_{2,1,k}}{\partial \mathbf{X}_{1,a(k)}} \\
&= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big((x_a-x_b)((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad -(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\Big(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2 \\
&\quad +((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big) \\
&\quad +((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad -(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))((y_b-y_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad +(z_b-z_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\Big(\Big(x_a-x_b\Big)^2+\Big(y_a-y_b\Big)^2+\Big(z_a-z_b\Big)^2\Big) \\
&\quad +((-x_a+x_b)(y_a-y_d)-(x_a-x_b)(y_b-y_d)+(x_a-x_d)(y_a-y_b))\Big(\Big(x_a-x_b\Big)^2+\Big(y_a-y_b\Big)^2 \\
&\quad +(z_a-z_b)^2\Big)\Big(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2 \\
&\quad +((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)\Big)
\end{aligned}
$$

$$
\begin{aligned}
\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,1,k,a(k),2} &= \frac{\partial \mathbf{B}_{2,1,k}}{\partial \mathbf{X}_{2,a(k)}} \\
&= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big((y_a-y_b)((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad -(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\Big(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2 \\
&\quad +((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big) \\
&\quad +((x_a-x_b)(x_b-x_d)+(z_a-z_b)(z_b-z_d))\Big(\Big(x_a-x_b\Big)^2+\Big(y_a-y_b\Big)^2 \\
&\quad +(z_a-z_b)^2\Big)\Big(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2 \\
&\quad +((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)-((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad -(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))((x_b-x_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)) \\
&\quad -(z_b-z_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\Big(\Big(x_a-x_b\Big)^2+\Big(y_a-y_b\Big)^2+\Big(z_a-z_b\Big)^2\Big)\Big)
\end{aligned}
$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,1,k,a(k),3} = \frac{\partial \mathbf{B}_{2,1,k}}{\partial \mathbf{X}_{3,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}((\Big(z_a-z_b)((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2$$
$$+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)$$
$$-((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((x_b-x_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_b-y_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\Big(x_a-x_b\Big)^2+(y_a-y_b)^2+(z_a-z_b)^2\Big)$$
$$+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)-(y_b-y_d)(z_a-z_b))\,(\Big(x_a-x_b\Big)^2+(y_a-y_b)^2$$
$$+(z_a-z_b)^2\Big)\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2$$
$$+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,1,k,d(k),1} = \frac{\partial \mathbf{B}_{2,1,k}}{\partial \mathbf{X}_{1,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|}((\Big(-x_a+x_b)(y_a-y_b)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2$$
$$+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)$$
$$+((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,1,k,d(k),2} = \frac{\partial \mathbf{B}_{2,1,k}}{\partial \mathbf{X}_{2,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|}(-\Big((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))\Big)^2$$
$$+\left((x_a-x_b)^2+(z_a-z_b)^2\right)\Big(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,1,k,d(k),3} = \frac{\partial \mathbf{B}_{2,1,k}}{\partial \mathbf{X}_{3,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|}(-\Big(y_a-y_b)(z_a-z_b)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)$$
$$-\left((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\left((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,2,k,b(k),1} = \frac{\partial \mathbf{B}_{2,2,k}}{\partial \mathbf{X}_{1,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(\Big(z_a-z_d)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)$$
$$-\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)\left((y_a-y_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)$$
$$+(z_a-z_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,2,k,b(k),2} = \frac{\partial \mathbf{B}_{2,2,k}}{\partial \mathbf{X}_{2,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))\,((x_a-x_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,2,k,b(k),3} = \frac{\partial \mathbf{B}_{2,2,k}}{\partial \mathbf{X}_{3,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(\Big(-x_a+x_d)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)$$
$$+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))\,((x_a-x_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,2,k,a(k),1} = \frac{\partial \mathbf{B}_{2,2,k}}{\partial \mathbf{X}_{1,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(\Big(-z_b+z_d)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)$$
$$+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))\,((y_b-y_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_b-z_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,2,k,a(k),2} = \frac{\partial \mathbf{B}_{2,2,k}}{\partial \mathbf{X}_{2,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))\,((-x_b+x_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_b-z_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,2,k,a(k),3} = \frac{\partial \mathbf{B}_{2,2,k}}{\partial \mathbf{X}_{3,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((\Big(x_b-x_d)(\Big(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2$$
$$+\Big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\Big)^2+\Big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\Big)^2\Big)$$
$$-\Big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\Big)\Big((x_b-x_d)\Big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\Big)$$
$$+(y_b-y_d)\Big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\Big)\Big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,2,k,d(k),1} = \frac{\partial \mathbf{B}_{2,2,k}}{\partial \mathbf{X}_{1,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((\Big(-z_a+z_b)(\Big(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2$$
$$+\Big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\Big)^2+\Big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\Big)^2\Big)$$
$$+\Big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\Big)\Big((y_a-y_b)\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)$$
$$+(z_a-z_b)\Big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\Big)\Big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,2,k,d(k),2} = \frac{\partial \mathbf{B}_{2,2,k}}{\partial \mathbf{X}_{2,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((-x_a+x_b)\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)$$
$$+(z_a-z_b)\Big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\Big)\Big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,2,k,d(k),3} = \frac{\partial \mathbf{B}_{2,2,k}}{\partial \mathbf{X}_{3,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((\Big(x_a-x_b)(\Big(\big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)$$
$$-\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)\big((x_a-x_b)\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)$$
$$+(y_a-y_b)\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)\big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,3,k,b(k),1} = \frac{\partial \mathbf{B}_{2,3,k}}{\partial \mathbf{X}_{1,b(k)}}$$
$$= \frac{1}{\|\mathbf{ba}\|^3}(-x_a+x_b)(y_a-y_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,3,k,b(k),2} = \frac{\partial \mathbf{B}_{2,3,k}}{\partial \mathbf{X}_{2,b(k)}}$$
$$= \frac{1}{\|\mathbf{ba}\|^3}(\big(x_a-x_b\big)^2+(z_a-z_b)^2\big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,3,k,b(k),3} = \frac{\partial \mathbf{B}_{2,3,k}}{\partial \mathbf{X}_{3,b(k)}}$$
$$= \frac{1}{\|\mathbf{ba}\|^3}(-y_a+y_b)(z_a-z_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,3,k,a(k),1} = \frac{\partial \mathbf{B}_{2,3,k}}{\partial \mathbf{X}_{1,a(k)}}$$
$$= \frac{1}{\|\mathbf{ba}\|^3}(x_a-x_b)(y_a-y_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,3,k,a(k),2} = \frac{\partial \mathbf{B}_{2,3,k}}{\partial \mathbf{X}_{2,a(k)}}$$
$$= \frac{1}{\|\mathbf{ba}\|^3}(-\left(x_a - x_b\right)^2 - (z_a - z_b)^2)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,3,k,a(k),3} = \frac{\partial \mathbf{B}_{2,3,k}}{\partial \mathbf{X}_{3,a(k)}}$$
$$= \frac{1}{\|\mathbf{ba}\|^3}(y_a - y_b)(z_a - z_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,3,k,d(k),1} = \frac{\partial \mathbf{B}_{2,3,k}}{\partial \mathbf{X}_{1,d(k)}}$$
$$= 0$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,3,k,d(k),2} = \frac{\partial \mathbf{B}_{2,3,k}}{\partial \mathbf{X}_{2,d(k)}}$$
$$= 0$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{2,3,k,d(k),3} = \frac{\partial \mathbf{B}_{2,3,k}}{\partial \mathbf{X}_{3,d(k)}}$$
$$= 0$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,1,k,b(k),1} = \frac{\partial \mathbf{B}_{3,1,k}}{\partial \mathbf{X}_{1,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big(-x_a+x_b)((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)$$
$$-((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((y_a-y_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\,(\big(x_a-x_b\big)^2+\big(y_a-y_b\big)^2+\big(z_a-z_b\big)^2)+(-(x_a-x_d)(z_a-z_b)$$
$$+(2x_a-2x_b)(z_a-z_d))\,(\big(x_a-x_b\big)^2+\big(y_a-y_b\big)^2+\big(z_a-z_b\big)^2)\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,1,k,b(k),2} = \frac{\partial \mathbf{B}_{3,1,k}}{\partial \mathbf{X}_{2,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big(-y_a+y_b)((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)$$
$$+((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((x_a-x_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\big(x_a-x_b\big)^2+\big(y_a-y_b\big)^2+\big(z_a-z_b\big)^2)+(-(y_a-y_d)(z_a-z_b)$$
$$+(2y_a-2y_b)(z_a-z_d))\,(\big(x_a-x_b\big)^2+\big(y_a-y_b\big)^2+\big(z_a-z_b\big)^2)\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,1,k,b(k),3} = \frac{\partial \mathbf{B}_{3,1,k}}{\partial \mathbf{X}_{3,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}((\Big(-z_a+z_b)((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2$$
$$+\Big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\Big)^2+\Big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\Big)^2\Big)$$
$$-((x_a-x_b)(x_a-x_d)+(y_a-y_b)(y_a-y_d))\,(\Big(x_a-x_b\Big)^2+\Big(y_a-y_b\Big)^2$$
$$+(z_a-z_b)^2)\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2+\Big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\Big)^2$$
$$+\Big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\Big)^2\Big)+((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((x_a-x_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\Big(x_a-x_b\Big)^2+\Big(y_a-y_b\Big)^2+\Big(z_a-z_b\Big)^2)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,1,k,a(k),1} = \frac{\partial \mathbf{B}_{3,1,k}}{\partial \mathbf{X}_{1,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}((\Big(x_a-x_b)((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2$$
$$+\Big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\Big)^2+\Big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\Big)^2\Big)$$
$$+((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\,((y_b-y_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_b-z_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\,(\Big(x_a-x_b\Big)^2+\Big(y_a-y_b\Big)^2+\Big(z_a-z_b\Big)^2)$$
$$+((-x_a+x_b)(z_a-z_d)-(x_a-x_b)(z_b-z_d)+(x_a-x_d)(z_a-z_b))\,(\Big(x_a-x_b\Big)^2+\Big(y_a-y_b\Big)^2$$
$$+(z_a-z_b)^2)\,(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\Big)^2+\Big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\Big)^2$$
$$+\Big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\Big)^2\Big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,1,k,a(k),2} = \frac{\partial \mathbf{B}_{3,1,k}}{\partial \mathbf{X}_{2,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big((y_a-y_b)((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\left(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)^2$$
$$+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)$$
$$-((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))((x_b-x_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_b-z_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\Big((x_a-x_b)^2+(y_a-y_b)^2+(z_a-z_b)^2\Big)$$
$$+((-y_a+y_b)(z_a-z_d)-(y_a-y_b)(z_b-z_d)+(y_a-y_d)(z_a-z_b))\Big((x_a-x_b)^2+(y_a-y_b)^2$$
$$+(z_a-z_b)^2\Big)\left(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)^2+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2$$
$$+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,1,k,a(k),3} = \frac{\partial \mathbf{B}_{3,1,k}}{\partial \mathbf{X}_{3,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|^3}(\Big((z_a-z_b)((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\left(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)^2$$
$$+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)$$
$$+((x_a-x_b)(x_b-x_d)+(y_a-y_b)(y_b-y_d))\Big((x_a-x_b)^2+(y_a-y_b)^2$$
$$+(z_a-z_b)^2\Big)\left(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)^2+((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))^2$$
$$+((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))^2\Big)-((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))((x_b-x_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_b-y_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\Big((x_a-x_b)^2+(y_a-y_b)^2+(z_a-z_b)^2\Big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,1,k,d(k),1} = \frac{\partial \mathbf{B}_{3,1,k}}{\partial \mathbf{X}_{1,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|}((\Big(-x_a+x_b)(z_a-z_b)((\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)$$
$$+\left((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\left((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,1,k,d(k),2} = \frac{\partial \mathbf{B}_{3,1,k}}{\partial \mathbf{X}_{2,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|}(-\Big(y_a-y_b)(z_a-z_b)((\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)$$
$$-\left((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\left((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,1,k,d(k),3} = \frac{\partial \mathbf{B}_{3,1,k}}{\partial \mathbf{X}_{3,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3\|\mathbf{ba}\|}(-\Big((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))^2$$
$$+\left(\left(x_a-x_b\right)^2+\left(y_a-y_b\right)^2\right)(\Big(\left((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)^2$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,2,k,b(k),1} = \frac{\partial \mathbf{B}_{3,2,k}}{\partial \mathbf{X}_{1,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(\Big(-y_a+y_d)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)$$
$$+\left((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)\left((y_a-y_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)$$
$$+(z_a-z_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,2,k,b(k),2} = \frac{\partial \mathbf{B}_{3,2,k}}{\partial \mathbf{X}_{2,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(\Big(x_a-x_d)(\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)$$
$$-\left((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)\left((x_a-x_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)$$
$$-(z_a-z_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,2,k,b(k),3} = \frac{\partial \mathbf{B}_{3,2,k}}{\partial \mathbf{X}_{3,b(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}(-\left(x_a-x_b)(y_a-y_d)+(x_a-x_d)(y_a-y_b)\right)\left((x_a-x_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)$$
$$+(y_a-y_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,2,k,a(k),1} = \frac{\partial \mathbf{B}_{3,2,k}}{\partial \mathbf{X}_{1,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((\Big(y_b-y_d)((\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)$$
$$-\left((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)\left((y_b-y_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)$$
$$+(z_b-z_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,2,k,a(k),2} = \frac{\partial \mathbf{B}_{3,2,k}}{\partial \mathbf{X}_{2,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((\Big(-x_b+x_d)((\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\left((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)^2+\left((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\right)^2\Big)$$
$$+\left((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)\left((x_b-x_d)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\right)$$
$$-(z_b-z_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,2,k,a(k),3} = \frac{\partial \mathbf{B}_{3,2,k}}{\partial \mathbf{X}_{3,a(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))\left((x_b-x_d)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\right)$$
$$+(y_b-y_d)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)))$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,2,k,d(k),1} = \frac{\partial \mathbf{B}_{3,2,k}}{\partial \mathbf{X}_{1,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((\Big(y_a-y_b)((\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)$$
$$-\big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)\big((y_a-y_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$+(z_a-z_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))\big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,2,k,d(k),2} = \frac{\partial \mathbf{B}_{3,2,k}}{\partial \mathbf{X}_{2,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}((\Big(-x_a+x_b)((\Big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)^2$$
$$+\big((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b)\big)^2+\big((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b)\big)^2\Big)$$
$$+\big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)\big((x_a-x_b)((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b))$$
$$-(z_a-z_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))\big)\Big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,2,k,d(k),3} = \frac{\partial \mathbf{B}_{3,2,k}}{\partial \mathbf{X}_{3,d(k)}}$$
$$= \frac{1}{\|\mathbf{ad}\times\mathbf{ba}\|^3}\big((x_a-x_b)(y_a-y_d)-(x_a-x_d)(y_a-y_b)\big)\big((x_a-x_b)((x_a-x_b)(z_a-z_d)-(x_a-x_d)(z_a-z_b))$$
$$+(y_a-y_b)((y_a-y_b)(z_a-z_d)-(y_a-y_d)(z_a-z_b))\big)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,3,k,b(k),1} = \frac{\partial \mathbf{B}_{3,3,k}}{\partial \mathbf{X}_{1,b(k)}}$$
$$= \frac{1}{\|\mathbf{ba}\|^3}(-x_a+x_b)(z_a-z_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,3,k,b(k),2} = \frac{\partial \mathbf{B}_{3,3,k}}{\partial \mathbf{X}_{2,b(k)}} = \frac{1}{\|\mathbf{ba}\|^3}(-y_a + y_b)(z_a - z_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,3,k,b(k),3} = \frac{\partial \mathbf{B}_{3,3,k}}{\partial \mathbf{X}_{3,b(k)}} = \frac{1}{\|\mathbf{ba}\|^3}(\left(x_a - x_b\right)^2 + \left(y_a - y_b\right)^2)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,3,k,a(k),1} = \frac{\partial \mathbf{B}_{3,3,k}}{\partial \mathbf{X}_{1,a(k)}} = \frac{1}{\|\mathbf{ba}\|^3}(x_a - x_b)(z_a - z_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,3,k,a(k),2} = \frac{\partial \mathbf{B}_{3,3,k}}{\partial \mathbf{X}_{2,a(k)}} = \frac{1}{\|\mathbf{ba}\|^3}(y_a - y_b)(z_a - z_b)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,3,k,a(k),3} = \frac{\partial \mathbf{B}_{3,3,k}}{\partial \mathbf{X}_{3,a(k)}} = \frac{1}{\|\mathbf{ba}\|^3}(-\left(x_a - x_b\right)^2 - \left(y_a - y_b\right)^2)$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,3,k,d(k),1} = \frac{\partial \mathbf{B}_{3,3,k}}{\partial \mathbf{X}_{1,d(k)}} = 0$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,3,k,d(k),2} = \frac{\partial \mathbf{B}_{3,3,k}}{\partial \mathbf{X}_{2,d(k)}} = 0$$

$$\left(\frac{\partial \mathbf{B}}{\partial \mathbf{X}}\right)_{3,3,k,d(k),3} = \frac{\partial \mathbf{B}_{3,3,k}}{\partial \mathbf{X}_{3,d(k)}} = 0$$

A.5. Code quality, Documentation, Extendability and Performance

The code is fully complying the official Python code style (`https://www.python.org/dev/peps/pep-0008/`). Code quality is automatically monitored after each push to the central repository using landscape (`https://landscape.io/github/mcocdawc/chemcoord`). The current (2017-11-07) score is 98 %, which is regarded as excellent in the rating of landscape. An integrated testing environment which builds and tests the library after each push to the central repository was set up using Travis CI (`https://travis-ci.org/mcocdawc/chemcoord/builds`). The code coverage of each commit is monitored using codecov (`https://codecov.io/gh/mcocdawc/chemcoord`).

The documentation is automatically compiled from the docstrings of functions using Sphinx to prevent diverging source code and documentation (`http://www.sphinx-doc.org/en/stable/`). The documentation is automatically compiled after each push to a documented branch and hosted by Read the Docs (`https://readthedocs.org/projects/chemcoord/`).

All classes were strictly designed for inheritance and the LGPLv3 license grants far reaching rights to modify and distribute code.

Performance was not a concern in 99 % of the code lines. The few occasions where numerical bottlenecks occured, were implemented as compiled extensions using the LLVM compiler numba.[7]

A.6. XYZ-files

The provided xyz-files which are necessary to execute the code examples.

```
7
Save as SF6.xyz
S  0.000000  1.000000  0.000000
F  0.901244  1.901244  0.901244
F  0.637276  1.637276  1.274551
F  0.901244  0.098756  0.901244
F  0.637276  0.362724  1.274551
F  1.103794  0.103794  0.000000
F  1.103794  2.103794  0.000000
```

```
14
Save as 2_methylpropane.xyz
  H      2.0300      0.3293      0.7396
  C      1.4193      0.2197      0.1658
```

```
 H      1.6626      0.7549      0.6103
 H      1.7384      0.9933      0.8762
 C      0.0627      0.3358      0.1493
 H      0.2505      1.3424      0.5964
 C      0.4771      0.7225      1.1578
 H      0.0950      0.6316      2.0902
 H      0.3132      1.7367      0.7692
 H      1.5414      0.6366      1.4126
 C      0.8881      0.2133      1.1206
 H      0.7415      0.7614      1.6053
 H      0.6160      0.9869      1.8502
 H      1.9607      0.3179      0.9117
```

```
56
Save as MIL53as_cluser.xyz
 O     1.23132200   4.50087200   0.41299920
 C     1.83795900   3.40984500   0.79043420
 O     0.00000000   3.40984500   1.89979780
 O     1.23132200   4.50087200   0.41299920
 C     1.83795900   3.40984500   0.79043420
Cr     0.00000000   1.70508500   0.95030980
 O     0.00000000   0.00000000   0.00000000
 O     1.50870200   1.11184200   2.03363880
 O     1.23132200   2.31881900   0.41299920
 C     2.09794600   0.00000000   2.37449180
Cr     0.00000000   1.70508500   0.95030980
 O     1.50870200   1.11184200   2.03363880
 O     1.23132200   2.31881900   0.41299920
 C     1.83795900   3.40984500   0.79043420
 O     1.23132200   2.31881900   0.41299920
 O     1.50870200   1.11184200   2.03363880
 C     2.09794600   0.00000000   2.37449180
 O     0.00000000   3.40984500   1.89979780
 O     1.23132200   2.31881900   0.41299920
 O     1.50870200   1.11184200   2.03363880
 C     1.83795900   3.40984500   0.79043420
 O     1.23132200   4.50087200   0.41299920
 O     1.23132200   4.50087200   0.41299920
 C     3.36242100   0.00000000   3.10593580
 H     3.44405300   0.88427900   3.74274080
 H     4.21067800   0.00000000   2.41744580
 C     3.07827600   3.40984500   1.56213220
 H     2.94821700   2.69516700   2.44029820
 H     3.39930000   4.31379500   1.89069420
 H     3.88114900   2.89529800   0.95198220
 C     3.07827600   3.40984500   1.56213220
 C     3.36242100   0.00000000   3.10593580
 C     3.07827600   3.40984500   1.56213220
 C     3.07827600   3.40984500   1.56213220
```

```
 H     3.44405300    0.88427900   3.74274080
 H     3.44405300    0.88427900   3.74274080
 H     3.44405300    0.88427900   3.74274080
 H     4.21067800    0.00000000   2.41744580
 H     2.94821700    2.69516700   2.44029820
 H     2.94821700    2.69516700   2.44029820
 H     2.94821700    2.69516700   2.44029820
 H     3.39930000    4.31379500   1.89069420
 H     3.39930000    4.31379500   1.89069420
 H     3.39930000    4.31379500   1.89069420
 H     3.88114900    2.89529800   0.95198220
 H     3.88114900    2.89529800   0.95198220
 H     3.88114900    2.89529800   0.95198220
 H     0.00000000    3.54345083   2.77971228
 H     1.76927217    5.23536892   0.74769890
 H     1.76927217    5.23536892   0.74769890
 H     1.76927217    5.23536892   0.74769890
 H     1.76927217    5.23536892   0.74769890
 H     0.00000000    0.00000000   0.89000000
 H     0.00000000    4.22827990   1.55013465
 H     0.00000000    3.54345083   2.77971228
 H     0.00000000    4.22827990   1.55013465
```

```
236
Save as MIL53_lt.xyz
 C   2.086270  1.521598  3.939204
 C   5.002512  0.362167  2.453004
 C   5.829949  0.516225  3.567793
 C   3.057275  2.334917  2.723780
 C   5.441689  0.030273  4.823603
 C   3.366825  0.799545  3.823749
 C   7.121847  1.189854  3.460392
 C   5.973517  0.451152  4.209980
 C   4.745359  1.109226  4.090376
 C   6.800953  0.297093  3.095191
 C   5.166271  1.459193  1.709932
 C   6.412694  0.843592  1.839380
 C   4.337829  1.612864  2.839235
 C   8.092852  0.376535  3.202591
 C   3.774355  0.295906  2.572607
 C   4.195267  0.645874  4.953052
 H   7.081068  0.743683  0.989877
 H   4.823779  1.842249  0.753569
 H   4.099026  1.239549  4.951737
 H   6.290077  0.034375  5.159883
 H   5.319072  0.778944  1.503100
 H   3.852775  1.028930  5.909415
 H   3.128022  0.426230  1.711246
 H   6.110063  0.069636  5.673107
```

```
 H   2.193348  1.401731  5.703017
 H   7.147719  1.193336  5.262928
 O   1.608047  1.752692  5.131746
 O   1.513442  1.813175  2.767368
 O   7.608718  1.425798  2.268005
 O   7.703324  1.486281  4.632383
 O   2.579051  2.566011  1.531238
 O   2.484446  2.626494  3.895616
 O   8.579722  0.612480  4.394979
 O   8.674327  0.672963  2.030601
 C  13.508106  1.613761  2.718048
 C  12.194920  5.079189  0.119374
 C  11.081417  2.302921  3.908762
 C  12.052416  1.489597  2.754243
 C  10.874887  4.453682  0.164239
 C  12.537107  2.427084  3.944956
Cr   9.423827  2.383802  1.517301
Cr   8.938303  2.790451  1.814164
 H   9.225178  3.811699  3.446472
 H  12.847017  3.162813  4.691450
 H   9.622483  0.955903  0.411844
 H   8.420233  3.659535  4.233628
 H   8.254178  4.625022  3.216534
 H  11.028146  2.811482  5.548815
 H  13.974539  0.658259  2.464961
 H  12.327346  5.643742  0.807090
 H  12.328288  5.768298  0.957191
 H  13.818017  2.349490  1.971555
 H  12.985095  4.325905  0.172134
 H  13.896248  1.932859  3.688676
 H  12.925249  2.746182  2.974328
 H  13.003540  1.471583  4.198044
 O  10.428248  1.921581  2.850885
 O  10.205152  4.133779  0.903563
 O   9.392779  1.804754  0.274770
 O   8.483863  3.776166  3.353599
 O  10.430220  2.573006  4.934230
 O  11.399257  1.108256  3.812142
 O  10.361906  4.213135  1.272193
 O   9.454863  2.962843  3.309405
 O  11.401219  1.759682  1.728775
 H   3.028993  2.304628  4.521692
 H   7.982689  0.291428  4.971673
 H   9.637374  2.640070  4.118483
 H  12.013135  0.959817  4.439214
 C   1.694712  2.573770  3.502852
 C   4.284615  2.932091  2.000916
 C   2.888521  2.940415  2.012807
 C   0.723711  3.387093  3.160153
 C   4.310358  3.955507  4.218599
 C   2.912664  3.920793  4.249972
```

C	4.999232	3.449201	3.071230
C	6.473173	3.464025	3.079472
C	2.197063	3.401911	3.168392
C	5.281358	3.142184	2.444406
C	3.883665	3.107470	2.413033
C	5.255616	2.118768	4.662089
C	3.168063	2.588588	3.494614
C	3.859522	2.127092	4.650197
C	7.444174	2.650702	3.583532
C	5.970232	2.635878	3.591774
H	7.438042	2.130409	5.275590
H	3.338693	3.461148	1.544615
H	5.845514	3.549299	1.612144
H	2.317106	2.569766	1.167547
H	2.367693	4.274471	5.118389
H	4.874514	4.362622	5.050861
H	4.838418	2.541489	1.152744
H	3.288106	1.756443	5.495457
H	5.809419	1.728166	5.510261
H	2.097508	1.920829	5.126701
O	0.178775	3.012556	2.001819
O	0.083890	3.725345	4.277651
O	7.017520	3.838556	4.237802
O	1.338574	2.089880	4.693643
O	1.054890	2.912022	2.385353
O	7.099620	3.132532	1.984321
O	7.988521	3.025232	2.425202
O	8.070621	2.319209	4.678684
C	0.711167	5.540904	0.577821
C	3.368503	4.365216	1.022468
C	4.824193	4.241053	1.058662
C	2.031194	6.166427	0.622610
Cr	1.225401	3.877681	1.072125
Cr	0.739912	3.471016	2.259382
H	1.909573	5.712233	2.474457
H	5.212335	3.921961	0.088033
H	2.163673	6.730962	1.549079
H	2.164495	6.855556	0.215202
H	2.821373	5.413154	0.569784
H	0.938574	4.898909	4.188547
H	5.290625	5.196554	1.311754
H	0.541253	2.043119	0.330228
H	5.134104	3.505321	1.805153
O	0.041436	5.221000	1.645634
O	0.198189	5.300355	0.530122
O	2.717300	4.095135	2.047919
O	0.770949	2.891971	0.467305
O	0.708866	4.050059	4.051480
O	1.679865	4.863383	2.611524
O	2.715322	4.746560	0.035413
H	0.525417	3.723285	4.858739

```
 H    6.336687  4.007866   4.785432
 H    0.716225  3.914215   4.874795
 H    1.743482  4.746752   3.491553
 C    5.543405   0.172689  0.857688
 C    4.736159   1.180731   1.642361
 C    5.964317   0.522657   1.522757
 C    3.908723   1.334789   0.527572
 C    4.296982   0.788291  0.728239
 C    6.371847   0.019019   0.271616
 C    2.616823   2.008417   0.634972
 C    7.652401  0.703035   0.156161
 H    6.610650   0.392333   2.384118
 H    4.419599   1.597506   2.592264
 H    3.628608   0.888199  1.577742
 H    5.885897  0.210367   1.814050
 O    2.035348   2.304844  0.537019
 O    2.129954   2.244363   1.827359
 O    8.130625  0.934129   1.036381
 O    8.225229  0.994611   1.327996
 C   13.083056  1.882424   0.635195
 C   11.420817  3.909014   3.145443
 C   10.449822  4.722337   3.517550
 C   11.627367  1.758254   0.599061
 C   11.769849  5.347860   3.472761
 C   12.740844  4.534537   3.190232
Cr    8.998760  2.652452   1.835982
Cr    9.484255  2.245791  1.495508
 H   11.260001  3.828702   4.902982
 H   10.168436  0.411238   2.897859
 H   10.002427  1.376740   3.914947
 H   13.468038  1.664754   1.634697
 H   13.394603  2.894879   0.365993
 H   13.551015  1.186969  0.066198
 H   11.902329  5.912397   2.546292
 H   11.903152  6.036987   4.310573
 H   13.531024  3.781264   3.137409
 H    8.800131  4.080356  0.093173
 H    9.197425  1.224558   3.765141
 H   12.560029  4.594587   3.525587
 H   12.873322  5.099076   4.116700
 H   12.874147  5.223662   2.352418
 O    8.967729  2.073412   3.628075
 O   10.907844  3.668466   2.037512
 O    9.938740  1.260091   3.034926
 O   10.974177  1.376912   0.458797
 O    9.029813  3.231501   0.043900
 O    9.780091  4.402433   2.449738
 O   10.976154  2.028336   1.624536
 O    9.936845  4.481789   4.625493
 O   10.751092  3.589110   4.213267
 H   10.523680  4.761335   5.233423
```

```
 H   8.785210  2.396181  4.437153
 C   1.136208  5.272236  3.931169
 C   5.370132  4.785717  2.251104
 C   4.399132  3.972392  4.411901
 C   1.485246  6.711070  2.686988
 C   2.456240  5.897743  3.976033
 C   2.943442  4.096562  4.375767
 C   3.914442  4.909885  2.287237
Cr   0.800352  3.609018  2.281206
Cr   1.285858  4.015681  1.050307
 H   5.836522  5.740908  1.996775
 H   0.239070  6.124557  4.447746
 H   4.787348  3.654544  3.440893
 H   4.709013  3.235719  5.157476
 H   2.248814  6.038810  3.086522
 H   1.484499  5.443581  0.878854
 H   0.116180  1.774466  3.683539
 H   1.087179  2.587790  2.979466
 H   2.472219  6.816894  3.384846
 H   1.761669  6.964559  1.660377
 H   1.501225  7.630222  3.278175
 H   3.875135  5.439653  3.972177
 H   5.680012  4.049051  1.505520
 H   4.865521  4.927582  4.666241
 H   5.758347  4.467858  3.222108
 H   1.500336  3.763414  3.649657
 H   3.219808  5.225483  3.576499
 H   2.732663  6.151231  5.002644
 O   0.466463  4.952334  4.998947
 O   0.347771  5.845018  3.839818
 O   3.261275  5.291221  3.345086
 O   0.504525  5.765661  1.664062
 O   1.316888  3.436640  2.842400
 O   0.345888  2.623317  3.820606
 O   2.292257  3.826469  5.401267
 O   3.263251  4.639797  1.261754
 O   0.623216  5.031690  2.823192
 O   2.290286  4.477893  3.317923
 O   1.254804  4.594729  0.741776
 C   1.239729  3.204107  0.193182
 C   5.515840  3.266221  0.282102
 C   2.713670  3.218931  0.184941
 C   4.800238  2.747340  1.363682
 C   6.989191  3.281039  0.273863
 C   3.428287  3.736041  0.885374
 C   3.402545  2.712625  1.332309
 C   4.824382  3.727718  0.873483
 H   2.874484  4.126643  1.733546
 H   5.395796  4.098366  1.718743
 H   5.345209  2.393662  2.232099
 H   2.838388  2.305510  2.164571
```

```
 O   0.613282   3.535600  0.901968
 O   7.534127   3.655576  0.884470
 O   7.629013   2.942788   1.391361
 O   0.695383   2.829577   1.351512
 H   0.975356   5.191911   5.688679
 H    2.890197   3.587982   6.015834
 H    0.282270   2.739947   4.700635
 C   0.165214   6.085564  2.731852
```

```
236
Save as MIL53_ht.xyz
 C   1.995734   4.966712  3.406221
 C   4.732245   2.693537  2.181383
 C   5.301675   2.263726  3.396334
 C   1.995137   4.962885   3.405777
 C   4.794730   2.737148  4.622767
 C   3.126497   4.031449  3.421421
 C   6.423729   1.337666  3.410827
 C   4.731648   2.689710   4.630616
 C   3.663388   3.592157   4.618605
 C   5.301077   2.259900   3.415665
 C   3.697819   3.597737   2.176185
 C   4.794133   2.733321   2.189233
 C   3.125900   4.027622   3.390578
 C   6.423132   1.333838   3.401172
 C   3.663985   3.595983  2.193394
 C   3.698416   3.601564  4.635814
 H   5.259340   2.399546   1.266822
 H   3.270249   3.949045   1.242309
 H   3.231057   3.953347   5.545350
 H   5.129502   2.312638   5.566674
 H   5.130099   2.316466  1.245324
 H   3.270846   3.952873  5.569690
 H   3.231654   3.957174  1.266649
 H   5.259937   2.403374  5.545176
 H   1.775655   5.120990  5.267662
 H   6.694018   1.471040  5.072832
 O   1.416035   5.433146  4.515644
 O   1.344452   5.349795  2.306014
 O   7.005884   0.867470  2.294803
 O   7.079579   0.950096  4.516636
 O   1.415438   5.429318   2.296355
 O   1.343855   5.345968   4.505985
 O   7.005287   0.863644   4.517196
 O   7.078982   0.946268   2.295362
 C  11.561382   2.609789   3.408707
 C  11.563391    2.886272  0.000382
 C  10.451572   1.663483  3.403924
 C  10.450975   1.659656   3.408076
```

```
 C  10.452392   1.936830  0.000054
 C  11.561979   2.613616  3.403292
Cr   8.360084   0.139238  1.703881
Cr   8.360382   0.137324  1.702118
 H   8.413830   1.821541  3.457769
 H  12.190092   2.479528  4.287649
 H   8.466415   1.672226  0.105564
 H   8.467514   0.653828  4.161418
 H   8.467787   1.774660  3.302373
 H  10.232883   1.472142  5.073956
 H  11.193583   3.639022  3.409253
 H  11.527859   3.528084  0.883592
 H  11.528014   3.527091  0.885082
 H  12.189495   2.475701  2.524350
 H  12.523047   2.363151  0.000003
 H  12.189340   2.474707  4.293023
 H  12.189937   2.478535  2.518976
 H  11.194181   3.642848  3.402746
 O   9.765564   1.075996  2.382514
 O   9.766021   1.350918  1.022124
 O   8.359973   0.696119  0.001350
 O   8.360791   0.970768  3.405587
 O   9.765744   1.077143  4.426113
 O   9.764967   1.072168  4.429485
 O   9.766200   1.349770  1.021476
 O   8.360193   0.974594  3.406412
 O   9.765147   1.073316  2.385886
 H   1.774730   5.115067  5.274176
 H   6.693128   1.465338  5.076107
 H   8.466785   0.658505  4.162617
 H  10.231993   1.466438  5.077854
 C   1.887777   4.831054  3.395959
 C   4.787390   2.732006  2.194942
 C   3.674475   3.574979  2.184628
 C   1.888375   4.827227  3.416040
 C   4.743914   2.703244  4.636958
 C   3.601289   3.509476  4.625321
 C   5.338376   2.310696  3.421388
 C   6.518206   1.459481  3.405706
 C   3.079325   3.969938  3.399718
 C   4.743316   2.707071  2.175041
 C   3.600691   3.513302  2.186678
 C   4.786792   2.735832  4.617057
 C   3.078727   3.973766  3.412281
 C   3.673877   3.578805  4.627371
 C   6.517608   1.463309  3.406293
 C   5.337779   2.314523  3.390611
 H   6.606834   1.380830  5.179141
 H   3.115515   3.802385  1.260794
 H   5.181836   2.378901  1.238729
 H   3.253287   3.938279  1.252440
```

```
 H    3.116113   3.798558   5.551205
 H    5.182434   2.375073   5.573270
 H    5.254245   2.404054   1.271277
 H    3.252690   3.942105   5.559559
 H    5.253648   2.407881   5.540722
 H    1.758736   5.121458   5.266384
 O    1.322469   5.315777   2.309016
 O    1.321672   5.315391   4.521446
 O    7.073075   0.982206   4.532570
 O    1.321871   5.319604   4.502983
 O    1.321075   5.319218   2.290554
 O    7.074572   0.982004   2.280702
 O    7.072478   0.986032   2.279430
 O    7.073974   0.985832   4.531298
 C    2.100554   8.176664   0.004636
 C   2.082745   8.177966   0.005003
 C   3.193152   9.128099   0.005634
 C    3.211553   9.126106   0.005072
Cr    0.008444   6.377159   1.706809
Cr    0.008146   6.379071   1.699191
 H    0.009130   8.100601   3.410777
 H   3.821110   8.993017   0.889951
 H    3.176022   9.767918   0.878902
 H    3.176177   9.766924   0.889772
 H    4.171209   8.602984   0.004694
 H    0.008533   8.104426   3.399622
 H   2.825355  10.157331   0.006181
 H    0.007759   4.653715   0.002840
 H   3.821267   8.994011   0.878723
 O    1.414183   7.590752   1.017434
 O    1.414362   7.589604   1.026166
 O   1.396917   7.591625   1.017187
 O    0.008035   5.543715   0.003340
 O    0.008255   7.214428   3.401722
 O    0.008853   7.210601   3.410277
 O   1.396738   7.590478   1.026413
 H    0.008110   6.986108   4.261937
 H    6.607743   1.375009   5.179983
 H    1.759660   5.115537   5.272279
 H    0.008857   6.981314   4.270235
 C    4.666354   2.921950   1.234212
 C    3.632525   3.829978   1.220219
 C    4.700785   2.927531   1.208209
 C    3.063096   4.259788   0.005268
 C    3.570039   3.786367   1.221164
 C    5.238273   2.492066   0.019819
 C    1.941040   5.185850   0.009225
 C    6.369036   1.556803   0.004619
 H    5.133116   2.566341   2.134954
 H    3.234671   4.207050   2.156277
 H    3.104833   4.120141   2.143574
```

H	5.093923	2.570641	2.168089
O	1.285191	5.573419	1.115034
O	1.358886	5.656044	1.106798
O	6.948735	1.090369	1.114042
O	7.020318	1.173719	1.095588
C	11.574314	2.893470	0.002414
C	10.464727	1.656976	3.405761
C	10.465324	1.653149	3.406238
C	10.463906	1.943336	0.001783
C	11.576324	2.602592	3.406674
C	11.575726	2.606418	3.405325
Cr	8.372916	0.144443	1.702411
Cr	8.372617	0.142529	1.703588
H	10.145701	1.379729	5.178015
H	8.371931	1.865973	3.407557
H	8.372204	0.745140	4.266601
H	11.538538	3.535260	0.881565
H	12.534295	2.370946	0.002205
H	11.538383	3.534266	0.887109
H	11.540792	3.244403	2.522700
H	11.540947	3.243410	4.291374
H	12.535382	2.083297	3.405703
H	8.373303	1.580914	0.000380
H	8.372529	1.869799	3.404442
H	12.535979	2.079470	3.406296
H	11.540195	3.248231	4.289299
H	11.540350	3.247237	2.520625
O	8.372805	0.979800	3.404942
O	9.778535	1.069916	2.384231
O	8.372208	0.975973	3.407057
O	9.777899	1.355849	1.023193
O	8.373026	0.690914	0.000120
O	9.778952	1.067237	2.384169
O	9.778078	1.356997	1.020406
O	9.779132	1.066088	4.427768
O	9.778355	1.071064	4.427831
H	10.146609	1.373910	5.178235
H	8.372952	0.749933	4.264745
C	2.087623	8.460345	3.401657
C	3.205487	9.407953	3.411341
C	3.206085	9.411779	3.400658
C	3.199218	9.405960	3.410779
C	3.198621	9.409788	3.401221
C	2.095678	8.461646	3.401289
C	2.095080	8.457820	3.410710
Cr	0.004387	6.660839	1.699483
Cr	0.004089	6.658925	1.706516
H	2.837689	10.437185	3.411887
H	1.761646	8.181613	5.183890
H	3.834043	9.276698	2.516341
H	3.834198	9.277692	4.285015

```
H   4.158874   8.882838  3.410401
H   0.003702   8.384282  0.004485
H   0.005074   4.937396  3.403453
H   0.004477   4.933571  3.406948
H   3.163245  10.050606  2.516520
H   3.163688  10.047772  2.526805
H   3.163843  10.046778  4.295478
H   1.772744   8.181145  5.182612
H   3.833601   9.273864  2.526983
H   2.838286  10.441012  3.400112
H   3.833446   9.272871  4.295658
H   0.004936   6.051890  4.269261
H   4.158277   8.886666  3.401599
H   3.163090  10.051599  4.285194
O   1.401250   7.874432  4.423726
O   1.402027   7.869458  4.431873
O   1.409073   7.870332  4.432120
O   1.401848   7.870606  2.388273
O   0.004200   5.823569  3.409047
O   0.004797   5.827396  3.402953
O   1.409850   7.875307  4.423479
O   1.409253   7.871479  2.388520
O   1.401429   7.873284  2.380126
O   1.409671   7.874159  2.379879
O   0.003979   7.494282  0.003985
C   1.850621   5.055001  0.003220
C   5.289502   2.544544  0.009209
C   3.030450   4.203786  0.012461
C   4.767537   3.005008  1.216394
C   6.480452   1.687256  0.007113
C   3.581437   3.782477  1.213985
C   3.624913   3.811239  1.228031
C   4.694351   2.939504  1.224299
H   3.114581   4.110429  2.137649
H   5.115540   2.576204  2.156487
H   5.252713   2.715925  2.142279
H   3.186393   4.139409  2.164343
O   1.294254   5.532478  1.128225
O   7.046358   1.198706  1.099910
O   7.047154   1.199091  1.112519
O   1.295751   5.532277  1.123643
H   1.760738   8.187433  5.175456
H   1.773652   8.186964  5.173558
H   0.005684   6.056683  4.262910
C   2.088220   8.456518  3.410343
```

Bibliography

[1] IUPAC, *Compendium of chemical terminology ("the gold book")*, edited by A. D. McNaught and A. Wilkinson., (Blackwell Scientific Publications, 1997).

[2] R. C. Hoft, J. D. Gale, and M. J. Ford, "Implementation of a Z-matrix approach within the SIESTA periodic boundary conditions code and its application to surface adsorption", Mol. Simul. **32**, 595 (2006).

[3] J. Parsons, J. B. Holmes, J. M. Rojas, J. Tsai, and C. E. M. Strauss, "Practical conversion from torsion space to Cartesian space for in silico protein synthesis.", J. Comput. Chem. **26**, 1063 (2005).

[4] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: A structure for efficient numerical computation", Comput. Sci. Eng. **13**, 22 (2011).

[5] International Organization for Standardization, "ISO9899 C99 Working Std.", 453 (2005).

[6] G. Jenks, *Sortedcontainers*, (2015) `http://www.grantjenks.com/docs/sortedcontainers/index.html` (visited on 2016-11-03).

[7] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: A LLVM-based python JIT compiler", Proc. Second Work. LLVM Compil. Infrastruct. HPC - LLVM '15, 1 (2015).

[8] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz, "SymPy: symbolic computing in Python", PeerJ Comput. Sci. **3**, e103 (2017).

[9] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, "Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis", Comput. Mater. Sci. **68**, 314 (2013).

[10] G. Butler, *Fundamental algorithms for permutation groups, Lecture notes in computer science* (Springer, 1991).

[11] G. van Rossum, *Tail recursion elimination*, (2009) `http://neopythonic.blogspot.de/2009/04/tail-recursion-elimination.html` (visited on 2016-11-01).

[12] F. Millange, C. Serre, and G. Férey, "Synthesis, structure determination and properties of MIL-53as and MIL-53ht: the first Cr^{III} hybrid inorganic-organic microporous solids: $Cr^{III}(OH)\cdot\{O_2C{-}C_6H_4{-}CO_2\}\cdot\{HO_2C{-}C_6H_4{-}CO_2H\}_x$", Chem. Commun., 822 (2002).

[13]W. Kabsch, "A solution for the best rotation to relate two sets of vectors", Acta Crystallogr. Sect. A **32**, 922 (1976).

[14]J. Nocedal, S. J. Wright, and SpringerLink (Online service)., *Numerical Optimization* (2006), XXII, 664 s. 85 illus.

[15]R. Lindh, A. Bernhardsson, G. Karlström, and P. Å. Malmqvist, "On the use of a Hessian model function in molecular geometry optimizations", Chem. Phys. Lett. **241**, 423 (1995).

[16]V. Bakken, and T. Helgaker, "The efficient optimization of molecular geometries using redundant internal coordinates", J. Chem. Phys. **117**, 9160 (2002).

[17]T. E. Oliphant, "Python for Scientific Computing", Comput. Sci. Eng. **9**, 10 (2007).

[18]J. Baker, "Techniques for geometry optimization: A comparison of cartesian and natural internal coordinates", J. Comput. Chem. **14**, 1085 (1993).

[19]H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, and M. Schütz, "Molpro: a general purpose quantum chemistry program package", WIREs Comput Mol Sci **2**, 242 (2012).

[20]W. J. Hehre, R. F. Stewart, and J. A. Pople, "Self-Consistent Molecular-Orbital Methods. I. Use of Gaussian Expansions of Slater-Type Atomic Orbitals", J. Chem. Phys. **51**, 2657 (1969).

[21]F. Eckert, P. Pulay, and H.-J. Werner, "Ab initio geometry optimization for large molecules", J. Comput. Chem. **18**, 1473 (1997).

[22]J. Baker, and W. J. Hehre, "Geometry optimization in cartesian coordinates: The end of the Z-matrix?", J. Comput. Chem. **12**, 606 (1991).

[23]P. Pulay, G. Fogarasi, F. Pang, and J. E. Boggs, "Systematic ab initio gradient calculation of molecular geometries, force constants, and dipole moment derivatives", J. Am. Chem. Soc. **101**, 2550 (1979).

[24]J. Baker, A. Kessi, and B. Delley, "The generation and use of delocalized internal coordinates in geometry optimization", J. Chem. Phys. **105**, 192 (1996).

[25]M. D. Hanwell, D. E. Curtis, D. C. Lonie, T. Vandermeersch, E. Zurek, and G. R. Hutchison, "Avogadro: an advanced semantic chemical editor, visualization, and analysis platform", J. Cheminform. **4**, 1 (2012).

List of Figures

All images of chemical structures were generated using Avogadro.[25]