Grid-based state space exploration for molecular binding

Hana Zupan, Frederick Heinz, and Bettina G. Keller
Department of Biology, Chemistry, Pharmacy, Freie Universität Berlin, Arnimallee 22, 14195 Berlin, Germany

(Dated: 2 November 2022)

Binding processes are difficult to sample with molecular-dynamics (MD) simulations. In particular, the state space exploration is often incomplete. Evaluating the molecular interaction energy on a grid circumvents this problem but is heavily limited by state space dimensionality. Here, we make the first steps towards a low-dimensional grid-based model of molecular binding. We discretise the state space of relative positions and orientations of the two molecules under the rigid body assumption. The corresponding program is published as the Python package molgri. For the rotational component of the grids, we test algorithms based on Euler angles, polyhedra and quaternions, of which the polyhedra-based are the most uniform. The program outputs a sequence of molecular structures that can be easily processed by standard MD programs to calculate grid point energies. We demonstrate the grid-based approach on two molecular systems: a water dimer and a coiled-coil protein interacting with a chloride anion. For the second system we relax the rigid-body assumption and improve the accuracy of the grid point energies by an energy minimisation. In both cases, oriented bonding patterns and energies confirm expectations from chemical intuition and MD simulations. We also demonstrate how analysis of energy contributions on a grid can be performed and demonstrate that electrostatically-driven association is sufficiently resolved by point-energy calculations. Overall, grid-based models of molecular binding are potentially a powerful complement to molecular sampling approaches, and we see the potential to expand the method to quantum chemistry and flexible docking applications.

Keywords: molecular association, state space discretisation, rotation grids, scientific programming, coiled-coils

I. INTRODUCTION

Molecular binding, i.e. the process in which two or more molecules form a long-lived complex, is ubiquitous but very difficult to study, both experimentally and computationally. Binding processes range across several length and time scales. Examples include dimers of small solvent molecules such as H₂O dimers or HF dimers solvation shells around ions in solution chelate-complexes that can mask more complex organic molecules including biological membranes, binding of small molecular inhibitors to protein receptors, the formation of protein-protein complexes, and self-assembly of molecular monomers into almost macroscopic structure such as hydrogels or fibrils.

Using molecular-dynamics (MD) simulations combined with rare-event techniques, one can now model and analyze molecular binding at atomistic detail. However, MD simulations rely on sampling the conformational space and assume that the ergodic hypothesis is fulfilled. This is problematic for binding processes, because binding often occurs via multiple distinct binding pathways. These pathways predominantly manifest when the molecules are already in direct contact, but even with large distances between associating molecules their long-range electrostatic interactions are not isotropic but can “guide” them towards forming a complex.

Exhaustively sampling all relative positions and relative orientations of the binding partners is challenging, and even determining whether or not the simulation has converged is difficult.

An alternative to sampling is to discretize the conformational space on a grid and to explore its details by changing the resolution of the grid. Recently, methods that use this approach to calculate grid-based models of conformational dynamics have been proposed. Selecting the grid points so that interactions in all relevant degrees of freedom are resolved uniformly is a crucial step in this process. Here we make the first steps towards grids for the binding process of two molecules. As a first approximation, we model each of the molecules as a rigid body. Fixing the first molecule at the origin of the coordinate system, the relative positions and orientations of the second molecule can be described by a combination of three subspaces: (i) translation in radial direction, (ii) rotation around the origin, and (iii) rotation around its centre of mass. Discretising each of the three subspaces yields a regular grid which we call a trans-rot-rot grid. Each of the grid-points can then be transformed into a set of matrices acting on position vectors of the atoms that yield a specific relative arrangement of the two molecules, i.e. a molecular structure.

A critical point in this process is the construction of uniform rotational grids. Since rotations can be represented as points on a unit sphere, this problem is mathematically closely intertwined with uniformly distributing N points on a sphere. We test rotation grids based on random and systematic Euler angles, random quaternion, polyhedra, and polytopes.
For the construction and analysis of rotational grids, we can build on a vast body of previous research since most areas of scientific computing need to consider grids on spheres or rotation spaces (e.g. geo, and atmospheric sciences, robotics, computer animation). Rotation matrices and the corresponding quaternion representations of rotations are also frequently used in the context of molecular sampling, most notably for the superposition of molecular structures. Other applications include integrators of the equations of motion for rigid bodies and the analysis of rotational distributions, rotational entropies, or rotational correlation functions.

We implemented trans-rot-rot discretisation of association space in the program molgr and provide it as a free Python package. The program takes structure files of two molecules as inputs and returns the relative positions and orientations of the molecules on a trans-rot-rot grid. The output is written in the format of a MD-trajectory, so that the grid point energies can be calculated using a standard MD program.

In this contribution, we briefly discuss the theory of rotations and explain the implementation and the usage of the program molgr. Next, we test rotational grids generated by six different algorithms regarding their uniformity and suitability for molecular studies. We then demonstrate how trans-rot-rot grids can be used to analyze a molecular binding process. We discuss the position- and orientation-dependent contributions to the interaction energies of two water molecules, i.e. molecules which are modelled as rigid bodies in classical force fields. As a second example, we consider the interaction between a Cl−-anion and a coiled-coil peptide.

II. THEORY

We introduce three common descriptions of rotations and briefly review how they relate to each other. We also discuss measures of uniformity applicable to rotation grids. For a systematic overview of rotation representations and conversions between different formats we recommend a comprehensive overview article by Diebel.

A. Rotation matrices

A 3D rotation matrix \( R \) is an orthogonal \( 3 \times 3 \) matrix, i.e. a matrix which fulfills \( R R^T = R^T R = I \), where \( R^T \) is the transpose of \( R \) and \( I \) is the \( 3 \times 3 \) identity matrix. When applied to two vectors \( \mathbf{v}' = R \mathbf{v} \) and \( \mathbf{u}' = R \mathbf{u} \), an orthogonal matrix preserves the length of these vectors and the angle between them, which are two key properties of rotations. The determinant of an orthogonal matrix always equals \( \pm 1 \). We define orthogonal matrices with \( \det R = 1 \) as those describing proper rotations while those with \( \det R = -1 \) describe improper rotations or rotation-inversions.

B. Euler angles

Euler angles, first introduced by Leonhard Euler, are based on the idea that (i) any two Cartesian axes in 3D space span a plane in which elementary rotations can be parametrized with a single angle, and (ii) any rotation in 3D space can be written as a sequence of three such elementary rotations around non-repeating axes.

The triple of angles \( \mathbf{u}_{xyz} = [\phi, \theta, \psi]^T \) is called an Euler angle set and is a description of a 3D rotation in which a vector undergoes a sequence of three elementary (planar) rotations: first around axis \( z \) for an angle \( \psi \), then around \( y \) for \( \theta \) and finally around \( x \) for \( \phi \). Other sequences of axes are also possible, another common choice is the \( xzx \) set.

Since the rotation axes for the elementary rotations are the axes of the Cartesian coordinate system, their corresponding elementary rotation matrices are simple trigonometric functions of the rotation angle. For rotation of \( \phi \) around an \( x \)-axis we have:

\[
R_x(\phi) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(\phi) & \sin(\phi) \\
0 & -\sin(\phi) & \cos(\phi)
\end{bmatrix},
\]

and the \( R_y \) and \( R_z \) elementary matrices are simple permutations of the matrix above. To transform an Euler angle set into a single rotation matrix, we multiply the corresponding elementary rotation matrices in the correct order:

\[
R_{xyz}(\phi, \theta, \psi) := R_x(\phi)R_y(\theta)R_z(\psi).
\]

C. Quaternions

Another common representation of rotations is that of unit quaternions. Quaternions have been invented by William Rowan Hamilton and are mostly known for their application in space rotations. A quaternion \( q \) is a 4-dimensional complex number

\[
q = [q_0, q_1, q_2, q_3]^T = a + bi + cj + dk,
\]

where \( a, b, c, \) and \( d \) are real numbers and \( i, j, \) and \( k \) are the basic quaternions that can be interpreted as unit vectors along three perpendicular coordinate axes with the property \( i^2 = j^2 = k^2 = -1 \). The length of a quaternion is defined as

\[
||q|| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}.
\]
A unit quaternion is a quaternion with $\|q\| = 1$. We state, but omit the proof, that there always exist exactly two unit quaternions that map to one rotation matrix. For the derivation see Refs. 33, 46 and 47. A rotation of any vector $v$ to $v'$ can be described in terms of quaternion multiplication. For this, the three-dimensional vector $v = [v_1, v_2, v_3]^T$ is interpreted as a quaternion in which the first element (the scalar element) is zero: $v = 0 + v_1 i + v_2 j + v_3 k$. The rotation then is

$$v' = qvq^{-1},$$

where $q$ is a unit quaternion and

$$(6) \quad q^{-1} = \frac{q}{\|q\|}, \quad q = [q_0, -q_1, -q_2, -q_3]^T.$$

is its reciprocal. The multiplication of quaternions in eq. 5 is defined by the Hamilton product $\epsilon^{34567}$. Quaternions have proven to be the most elegant solution in computational implementation of rotations. Since there is a natural and smooth mapping from the space of quaternions to the space of rotations (the $SO(3)$ group), interpolation of rotations can be easily performed in quaternion space and no singularities occur within this description, as opposed to the Euler angles parametrisation.

### D. Rodrigues’ formula

Rotational matrices can be inferred from an initial vector $v$ and rotated vector $v'$ using Rodrigues’ formula that has been independently discovered by Euler as well as Rodrigues. For the derivation, see chapter 3 of Ref. 50. In essence, we can calculate the unit vector $\hat{\omega}$ along the axis of a rotation that transforms $v$ into $v'$ and the angle of rotation $\theta$ as

$$\hat{\omega} = \frac{v \times v'}{\|v\| \|v'\|}, \quad \sin(\theta) = \|\hat{\omega}\|, \quad \cos(\theta) = \frac{v \cdot v'}{\|v\| \|v'\|}.$$

We additionally define the skew-symmetric matrix of a vector $\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T \in \mathbb{R}^3$ as

$$[\omega]_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.$$

The transformation $v \rightarrow v'$ is then performed with

$$v' = e^{[\omega]_\times \theta} v = (I + \sin \theta [\hat{\omega}]_\times + (1 - \cos \theta)[\hat{\omega}]_\times^2) v,$$

which is known as Rodrigues’ rotation formula. Writing the skew-symmetric matrices in eq. 8 in matrix form, we can directly convert this representation into a rotation matrix multiplication.

### E. Grid coverage and uniformity

We use different parameterisations of rotations to create rotation grids. In order to compare how well the generated set of grid points covers the entire space of rotations, we introduce a measure of uniformity. Consider a rotation grid with $N$ grid points, each represented by a point on a unit sphere (radius $r = 1$, total surface area $A = 4\pi$). We calculate the local grid density in various sectors of the unit sphere by counting the number of grid points $N(S(v, \alpha))$ within a well-defined sub-area $S(v, \alpha)$ of the sphere surface and calculating the coverage ratio

$$c(v, \alpha) = \frac{N(S(v, \alpha))}{N}.$$

The vector $v$ denotes the center of $S$, and the parameter $\alpha$ determines its surface area $A_S$. In a fully uniform grid, the coverage would be equal for all sub-areas of equal size $A_S$ and further equal to the ratio $A_S/A$.

A suitable choice of sub-areas is a class of curved surfaces called spherical caps, defined by the intersection of the unit sphere with a plane (Figure 1C). The intersection itself is a circle, called the base of the spherical cap. Here, $v$ is the unit vector that connects the center of the unit sphere with the center of the cap and defines the center of $S$. The parameter $\alpha$ is determined by the opening angle of the spherical cap, i.e. the angle $\alpha$ between $v$ and a unit vector that points to rim of the base, and is the quantity defining the size of $S$. The area of a spherical cap is

$$A_S = 2\pi(1 - \cos \alpha).$$
And so the expected coverage for a uniform grid is

\[ c_{\text{uniform}}(\alpha) = \frac{A_S}{A} = \frac{1}{2}(1 - \cos \alpha) \]

See Refs. [30] and [31] for a more detailed discussion on measures of uniformity.

III. SOFTWARE

The Python package molgri is freely available within the framework of Python Package Index. Documentation and source code are available via GitHub. Dependencies are listed in section I of the supplementary information.

The package molgri implements a range of generation, analysis and plotting tools. A short user guide is provided in Section IIIB. As the workflow in Figure 2 indicates, the main output of molgri are the of relative positions and orientations of two molecules on a trans-rot-rot grid. The output is written in the file-format of a GROMACS molecular-dynamics trajectory (.gro-file), such that can be further processed by standard MD programs to obtain and analyse point energies. In this contribution, we use GROMACS for the energy calculation, but since the .gro-file format is a standard file-format for MD trajectories and can readily be converted in a other formats, the user is free to use the MD program of their choice for this step. We use the term pseudo-trajectory for the output, because it has the file format of a MD trajectory (.gro file), but it represents a trans-rot-rot grid and not the result of MD sampling.

molgri generates pseudo-trajectories in two steps. In the first step, the program generates a rotation grid, that is, a collection of \( N_{\text{rot}} \) points distributed on a unit sphere. We implemented and tested six algorithms for this step (see section IIIA). In the second step, the rotation grid is combined with a translation grid with \( N_{\text{trans}} \) elements. Overall we obtain \( N_{\text{trans}} \cdot N_{\text{rot}} \cdot N_{\text{rot}} \) relative positions and orientations for the two-molecule system (see section IIIB).

We took particular care to build a modular and flexible program structure. Therefore, each step outputs a standard file format that can be reused by other applications. This also means that a rotation grid of particular size can be reused for any pair of molecules.

A. Step 1: rotational grid algorithms

The six algorithms we implemented to generate rotational grids are: Systematic Euler grid, Random Euler angles, Random unit quaternions, 3D cube grid, Icosahedron grid, and 4D cube grid. The algorithms are detailed in section II of the supplementary information, but we sketch the ideas behind them here.

Systematic Euler grid is based on a uniformly spaced grid between 0 and \( 2\pi \) for each of the three Euler angles using the \( xyz \) sequence of axes. Random Euler angles algorithm generates a random rotation by selecting a uniform random number in the \([0, 2\pi]\) interval for each of the three Euler angles. Random unit quaternions algorithm generates random rotations by selecting random quaternions using the subgroup algorithm. Icosahedron grid and 3D cube grid use as a regular grid on the surface of a polyhedron to approximate a uniform grid on a unit sphere. This approach starts with inscribing a 3D polyhedron (a cube, or an icosahedron) into a unit sphere, then constructing a grid on the faces of the polyhedron and projecting the grid points onto the unit sphere. The number of grid points \( N_{\text{rot}} \) can be increased by subdividing the faces of the polyhedron. We subdivide square faces (cube) on a layered diagonal square lattice and triangle faces (icosahedron) on a layered triangle lattice. 4D cube grid extends this concept to polytopes (generalization of polyhedra to higher dimensional spaces) and quaternions. In this case, a four-dimensional hyper-cube is inscribed into a four-dimensional unit hyper-sphere, a grid is constructed on its faces, and the grid points are projected onto the hyper-sphere. The resulting four-dimensional vectors are interpreted as quaternions.

Some of the six algorithms generate a set of Euler angles, others a set of quaternions and still others a set of points on a sphere. We unify all outputs in the latter format by letting the quaternions or Euler angles act on a unit vector in the \( z \)-direction and saving the rotated vector as a three-dimensional grid point (algorithm I in section II of the supplementary information).

Systematic Euler grid and polyhedron/polytope-based grid algorithms use regular grids to generate the rotations, and can therefore natively only generate rotational grids with specific numbers of grid points \( N_{\text{rot}} \). To provide complete flexibility in the number of grid points \( N_{\text{rot}} \), we first generate a grid of size \( M \geq N_{\text{rot}} \), then order all points in a way that maximizes coverage and truncate at \( N_{\text{rot}} \) (algorithm II in section II of the supplementary information).
B. Step 2: from rotation grids to pseudo-trajectories

In the second step, the translational grid, the rotational grid of relative positions and the rotational grid of relative orientations are combined. We use a linear equidistant grid for the translations, and currently the same rotational grid for selection of relative positions and relative orientations. Simplifications due to symmetry are not taken into consideration. A full grid in state space is a product (meshgrid) of all three component grids.

To generate a pseudo-trajectory, the rotational grid is first transformed into a series of rotation matrices using Rodrigues formula (Equation 9). Specifically, the unit-vector in the z-direction is the initial vector $v$ and the grid point is the rotated vector $v'$. Rotation axis and angle can then be calculated using Equation 7.

The pseudo-trajectory is generated by keeping molecule 1 fixed at the origin of the three-dimensional coordinate system while molecule 2 is translated and rotated. Initially, the two molecules are positioned in their starting orientations at distance $r_0$ (first value of the translational grid). The $N_{\text{rot}}$ rotation matrices are applied to the second molecule as rotations around the origin and the resulting configurations are recorded. In each of the configurations, the $N_{\text{rot}}$ rotation matrices are applied to the second molecule as rotations around the center of mass, generating $N_{\text{rot}}$ relative orientations. The process is repeated for each radius provided by the translational grid. All resulting configurations are recorded in a single .gro file.

C. Using the molgri package

In this section, we provide short instructions for the installation of molgri package and its use from command line. Assuming a compatible Python installation is already present, the package is installed with the command

```
pip install molgri
```

After installation, molgri can be imported as a package into a Python program as `import molgri`. Alternatively, we also provide three scripts that can be run directly from the command line: 1) `molgri-io` that creates a standard tree of input/output directories, 2) `molgri-grid` that generates rotation grids including their plots, animations and statistical analyses and 3) `molgri-pt` that generates pseudotrajectory files. All three scripts can be run with an optional flag `--help` that returns a short user guide to the script.

Running the script `molgri-io` with the optional flag `--examples`

```
molgri-io --examples
```

provides example input .gro files including all molecules and ions used in Results section of this contribution and generates the following directory structure

```
|--input
 | |--grid_files
 | |--pt_files
 | |--figures
 | |--animations
 | |--statistics_files
|--output
 | |--CL.gro
 | |--H2O.gro
 | |--NA.gro
 | |--NH3.gro
 | |--example_protein.gro
```

Instead of running this script, the user can also manually create an input/ directory and copy any .gro files for which pseudotrajectories should be generated there.

The second script `molgri-grid` implements step 1 from Figure 2. It is necessary to specify the number of grid points $N$ and the algorithm `-algo` (options: systemE, randomE, randomQ, cube4D, cube3D, ico) to generate a rotation grid. In addition, optional flags to this script are: `--draw` that saves a plot of grid points, `--animation` for the corresponding 3D animation, `--statistics` that generates a number of files and figures analysing the uniformity and convergence of this grid, and `--readable` that saves the grid points in a human-readable .txt format in addition to the standard .npy format. The generated files are saved to the correspondingly named subfolder within the output folder. For example,

```
molgri-grid -N 250 -algo ico --draw
--animate --statistics --readable
```

generates a rotational grid with 250 grid points using the algorithm Icosahedron grid, draws and animates the grid, calculates the grid statistics, and additionally stores the grid in human-readable format.

The last command line script `molgri-pt` implements step 2 and creates a pseudotrajectory. It expects two molecular structure files in the .gro-format which should be stored in the `input/` folder. Usually it is preferential to select the smaller molecule as the `-m2` option since this is the molecule undergoing translations and rotations. In addition, the user needs to specify the rotational grid in the form `-rot algorithm N` (see algorithm names above) and the translational grid (flag `-trans`). The translational grid can be specified in one of the following formats: a list of distances, linspace(start, stop, num) or range(start, stop, step), where the units are in nanometers. For example, the translational grid with grid points 1.0, 1.5, 2.0, 2.5, and 3.0 nm can be specified as

```
• -trans "(1, 1.5, 2, 2.5, 3)"
• -trans "linspace(1, 3, 5)"
• -trans "range(1, 3.1, 0.5)"
```

In summary, the command

```
molgri-pt -m1 NH3 -m2 H2O -rot cube3D_15
-trans "linspace(1, 3, 5)"
```
generates a pseudo-trajectory of a water molecule rotating around an ammonium molecule, where the rotational grid is generated with a cube 3D algorithm and contains 15 points (or is read from `/output/grid_files/cube3D_15.npy if already generated). The two structure files are expected as `/input/H2O.gro and `/input/NH3.gro, and the translational grid is the one discussed above. Finally, the user may select an optional flag `only_origin` which suppresses the generation of different orientations of the second molecule and only returns \(N_{\text{rot}}\cdot N_{\text{trans}}\) structures. This is useful if the second molecule is spherically symmetric (e.g. a single ion or an atom).

IV. RESULTS

A. Comparison of rotational grid algorithms

Algorithms presented in Section III A provide several ways to discretise rotation space, each generating a particular set of \(N_{\text{rot}}\) points on a sphere. The insets in Figure 3 A show the grid points generated by each of the algorithms for \(N_{\text{rot}} = 600\). One can easily distinguish grid-based approaches from stochastic ones by the regularity and the symmetry of the grids. Note that in the Systematic Euler grid, the grid points align in circles along the “latitudes” around the sphere and are denser at the “equator” than at the “poles”. Since the random Euler angle approach is a direct stochastic analogue of the Systematic Euler grid, it also over-samples along the equator and under-samples the pole areas. Random unit quaternions algorithm samples from uniform distribution in the unit sphere (analytical proof in [29]), but due to fluctuations in a random sample with finite number of points, this does not necessarily yield the most uniform grid. Polyhedra-based algorithms (3D cube grid and Icosahedron grid) show obvious regularities in their structure. To quantitatively compare the grids, we recall the measure of uniformity introduced in section Section II E where the local density of grid points was measured by counting the number of grid points within a spherical cap area (Equation 11) and evaluating variation from ideal grid density given in Equation 12. Local density is measured around 100 randomly chosen axes and the variation of measurements shown in form of a violin plot in Figure 3 A. The parameter \(\alpha\) indicates the size of the spherical cap area in which counting is performed. The narrower the distribution, the more uniform the grid. The two algorithms that are based on inscribing polyhedra into three-dimensional sphere, 3D cube grid and Icosahedron grid, produce by far the most uniform grids across all values of values of \(\alpha\).

It is important to consider several different angles to test uniformity at all scales, from narrow areas surrounding the axis to ones covering almost entire sphere (we test angles ranging from \(\pi/6\) to \(5\pi/6\)). How variance changes with \(\alpha\) provides additional information. We observe, for example, that variance in random Euler angle grid changes with \(\alpha\), suggesting that the algorithm fails to achieve true randomness (random quaternion algorithm performs better in this regard). Using a single alpha value can also be deceiving as it may reflect inherent symmetries in grid construction. Notable examples are cubic grids with essentially nonexistent variance in grid densities for \(\alpha = \pi/2\). For the 3D and 4D cubic grids, the subdivision is performed in the same way for each of the eight faces, meaning that each spherical cap covering a quarter of the sphere (\(\alpha = \pi/2\)) always contains the same collection of points (\(\pm 1\) per face for \(N\) not divisible by 8).

Grid uniformity is not only of theoretical interest but has real consequences when using rotation grids to discretise the space of molecular rotations. Figure 3 B displays a simple molecular example where the grids discussed above were employed to study a system of two water molecules. In this figure, areas where structures with low potential energy were found are shown in dark blue color. In most cases, we observe three minima: a broad one near the free electron pairs of the central oxygen atom and two more pointed ones near each of the central hydrogens, indicating three possible hydrogen bonding patterns. The difference between grid algorithms becomes apparent when we note that one of the three minima (near bottom hydrogen) is completely missing in the Systematic Euler grid and is only represented by a single point in the 4D cube grid. Although a very dense (600 points) grid was used, grids with poor uniformity failed to identify all regions of interest even in this simple example. This underlines the necessity of examining topology when selecting parametrisation of non-Euclidean spaces.

Figure 4 compares how time-intensive the six rotational grid algorithms are. The timings have been measured on a Debian GNU/Linux 11 operating system, equipped with an Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz and 30 GB RAM. Both random algorithms, Random Euler angles and Random unit quaternions, are very efficient, while the 3D cube grid and Icosahedron grid show much poorer scaling with the number of grid points. The almost constant timing for long intervals of \(N_{\text{rot}}\) originate from the fact that polyhedra-based algorithms do not natively yield grids with arbitrary number of grid points - larger grids are created and truncated according to an unification algorithm found in section II of the Supplementary information. However, even for the more time-consuming algorithms the computational costs are in the range of seconds for grids with up to 1000 points, which is very small compared to the cost of the entire analysis, which is dominated by the energy calculation along the pseudo-trajectory.

In summary, polyhedra-based grids are remarkably uniform. The somewhat larger computational cost for generating these grids are well worth the improved quality.
FIG. 3. Comparison of rotation grid algorithms. A. Rotation grids with N=600 points are shown in upper left corners. The variance in grid densities across sphere surface is displayed as a set of violin plots. As a measure of uniformity we select 100 random axes and measure the density of the grid in this region by counting the number of points \( N_{sc} \) that fall within an angle \( \alpha \) of the axis and calculating the coverage \( N_{sc}/N \). Broad distributions suggest that coverage is not uniform. B. The same rotation grids are used to create pseudo-trajectories for the water-water system. Potential energies [kJ/mol] of structures with various orientations and positions of the second water molecule are plotted, energy minima shown in dark blue. The Systematic Euler algorithm misses one of the three minima (below the hydrogen) and the cube 4D algorithm covers it with only one point, demonstrating the errors that may arise when using grids with poor uniformity.

FIG. 4. Time needed to construct rotation grids using six different algorithms. Polyhedra grids (icosahedron, cube 3D) are orders of magnitude more time intensive than random grids (random Euler angles, quaternions), but total time needed to create grids with up to 1000 points is still under 100 seconds.

B. Water-water system

In the second step of the molgri procedure, rotation and translation grids are combined to systematically parameterise all three movements with which we parameterise association space - translation, relative rotation and internal rotation, see Figure 1 A. After selecting two molecules, a pseudo-trajectory based on the meshgrid of the three grids is generated.

We first demonstrate the use of pseudo-trajectories on a system of two water molecules, since this is a small, well-studied example that nonetheless displays several oriented hydrogen-bonding patterns, enabling us to study how radial and rotational degrees of freedom contribute to potential energy. Moreover, rigid models of water are common in classical MD, meaning that a direct comparison with established methods is possible.

| Potential [kJ/mol] | H-bond length [nm] |
|--------------------|--------------------|
| systematic Euler a. | -27.67 | 0.28 |
| icosahedron grid    | -27.39 | 0.28 |
| random Euler a.     | -27.35 | 0.28 |
| random quaternions  | -27.24 | 0.27 |
| 3D cube grid        | -27.23 | 0.27 |
| MD run              | **-27.09** | **0.28** |
| 4D cube grid        | -27.03 | 0.27 |

TABLE I. Comparing minima of (pseudo)trajectories, each of the methods evaluating 60.000 points in state space.
pare hydrogen bond lengths. While there is some difference between different rotation algorithms, all minima fall within 2.5% of the comparison calculation and all but one molgrigri algorithms even find a deeper minimum than a simulation with the same number of points. The plots in previous section (Figure 3) also suggest that the minima found by molgrigri occur at locations expected by partial charge distribution in water molecules.

**FIG. 5.** A. Coulomb and LJ energies [kJ/mol] for the full pseudotrajectory (icosahedron grid, \( N_{\text{rot}} = 600, N_{\text{trans}} = 5 \)) separated into three plots based on the radial COM-COM distance of both water molecules. B. Pseudo-trajectory structures with lowest (blue box) and highest (red box) Coulomb/LJ contributions.

In the the next step, we examine how different components of potential energy, specifically the Coulomb and Lennard-Jones contributions, vary with distance and orientation. The ability to systematically examine spatial dependence of energy contributions is a unique feature of our approach since classical trajectories rarely enable us to examine the full range of states, strongly biasing the sample towards a few easily accessible minima. For this examination, we construct longer pseudo-trajectories of the same system using Icosahedron grid with \( N_{\text{rot}} = 600 \) and \( N_{\text{trans}} = 5 \) between 0.3 and 0.32 nm, generating 600 \( \cdot \) 600 \( \cdot \) 5 = 1.8 million time steps in total.

In order to separate translational variation from rotational, we plot three shells with different radial distances in separate subplots of Figure 5 A. We plot a point at the center of mass (COM) of the second water molecule and color-code it according to its Coulomb or Lennard-Jones energy. Since only a subset of the 1.8 million structures can be reasonably visualised, we first perform two levels of selection: 1) among all structures with the same COM but different internal orientations, we select the one with lowest energy and 2) of the points selected in step 1), 100 lowest and 100 highest energies are plotted. On the bottom of the same Figure, 5 B, we plot the structures with the lowest and highest energy contribution so that internal orientation of the second water molecule can also be seen.

Completely different patterns arise for the two energy components. In the case of Lennard-Jones potential, radial distance plays a major role (as can be expected from the functional form of this potential) while the hydrogen-bonding-specific patterns are not present. Hydrogen-hydrogen interaction is the most favourable LJ interaction in this system while oxygen-oxygen one, especially at smaller radii, is most repulsive.

On the other hand, Coulomb energies barely vary with radial distance but a strong and persisting orientational pattern is observed instead. This could again be predicted by the form of Coulomb potential which scales with \( 1/r \), \( r \) being the distance between charges. The lowest- and highest energy structures confirm that the cause is electrostatic repel between positively charged hydrogens and attractive interaction of a hydrogen with free electron pairs on an oxygen.

While the contribution analysis presented here is anything but novel for this simple system, we expect this kind of systematic study to be valuable for complex systems featuring a mix of interaction potentials, something that is very typical for protein systems.

**C. Polypeptide-ion system**

The water-water dimer is an often modeled system without any internal degrees of freedom and can be treated accurately as a system of two rigid bodies. To test the implications of the rigid-body assumption on a system with many internal degrees of freedom, we use molecularRotationalGrids to evaluate the system of a chlorine anion interacting with a coiled-coil peptide (Figure 6). The coiled-coil peptide hFF03 is a parallel homodimer of two peptide strands, each of which forms an \( \alpha \)-helix. The resulting coiled-coil dimer is stabilized via a leucine zipper. The ends of the peptide strand are not capped and the peptide is represented at pH= 7. Consequently, the amid group of the N-terminal end of the coiled-coil dimer and amine group of the lysine side chains are positively charged. Whereas the carboxyl group of the C-terminal end and the glutamatic acid side
FIG. 6. A. An example of a pseudo-trajectory with a sparse grid ($N_{\text{rot}} = 30$, $N_{\text{trans}} = 8$, distances 0.5 - 2.25 nm). Molecular structure of the coiled-coil peptide is shown in grey and the probe Cl$^-$ ion in orange. All frames of the pseudo-trajectory are shown overlapped, each contains one Cl$^-$ ion. In B and C parts of the figure, $N_{\text{rot}} = 1000$ is used, otherwise, the set-up is equivalent. B. Change in the potential energy of the coiled-coil-Cl$^-$ system through the usage of energy minimisation of each frame. Potential energy before (black) and after (red) energy minimization is shown for frames ordered from highest to lowest energy. Additionally, the relative change in energy by minimization is shown in blue. Some structures (zone I) exhibit immense relaxation during the minimisation; in these cases the initial structure featured overlapping Lennard-Jones potentials. Other structures (zone III) remain essentially unchanged. C. Example pseudo-trajectory frames from zones I, II and III. The initial pseudo-trajectory structure is shown in grey and the structure after minimisation in red. Only the movement of atoms within 0.6 nm of the Cl$^-$ ion are visualized in red; changes outside this sphere were minimal fluctuations.

Each peptide strand consists of 26 amino acids, and the $\alpha$-helix measures approximately 3.8 nm in length. In the coiled-coil structures, the helices are approximately 1.2 nm apart. In total, the system has just under 1000 atoms.

To calculate the rotational grid for coiled-coil peptide and the chlorine anion, we placed the center of mass of the coiled-coil peptide at the origin of the coordinate system and aligned the $\alpha$-helices with the $z$-axis. The relative positions of the chlorine anion Cl$^-$ were generated using the package molgr with a translational grid of 0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 2.00, and 2.25 nm, i.e. an equidistant grid with $N_{\text{trans}} = 8$. For the rotational grid of the relative positions, we used an Icosahedron grid with $N_{\text{rot}} = 1000$ grid points. Since Cl$^-$ is a single ion, no rotational grid for the relative orientations was needed. The combined grid had 8000 grid points and the resulting pseudo-trajectory as many frames. To give an impression of such a grid, Figure 6 A shows the relative positions of a Cl$^-$ anion on a similar but more sparse grid with $N_{\text{trans}} = 8$ and $N_{\text{rot}} = 30$.

We next calculated the interaction energy between Cl$^-$ and the coiled-coil peptide using GROMACS rerun on the pseudo-trajectory. Since no dynamics are present, we cannot estimate kinetic energy; the interaction energy that we obtain consists of the Coulomb and Lennard-Jones interactions. Since the coiled-coil peptide is rod-like and the set of Cl$^-$ configurations spherical, some of the structures at smallest radii inevitably feature the chlorine anion overlapping with the peptide, leading to enormous Lennard-Jones interaction energies. We assume that many of these frames can be discarded as unphysical cases.

To obtain more realistic grid point energies, the internal degrees of freedom of the coiled-coil peptide can be relaxed to the nearest local minimum. We thus ran a steepest descent energy minimization using GROMACS on every frame of the pseudo-trajectory while constraining the position of the Cl$^-$ anion and the peptide backbone atoms to make sure that the system does not relax into the neighboring grid point. In three of the 8000 grid points, the energy minimization did not lower the energy, despite very high initial energies. We excluded these grid points from the analysis. The energy decrease for the other grid points is shown in Figure 6 B as absolute (black and red line) and relative (blue line) energy difference. On average, the energy after minimisation is (2200 ± 480) kJ/mol per grid point.

The changes in energy after minimisation can be
roughly divided into three categories, labeled I, II and III in Figure 6 B and C. The largest energy differences (I) occur for grid points at which the Cl\(^{-}\) ion overlaps with several atoms of the coiled-coil peptide, usually because it is placed inside the coiled-coil. Moderate energy changes (II) occur when the Cl\(^{-}\) overlaps with a single side-chain. In the example in the Figure 6 C-II Cl\(^{-}\) is initially placed on top of a lysine side chain which is located at the exterior of the coiled-coil and can easily rotate away during energy minimization. In roughly half of the grid points, Cl\(^{-}\) is placed outside of the coiled-coil peptide and only minimal adjustments are made during the energy minimization (III).

We draw the conclusion that minimization is crucial in the areas where Lennard-Jones interaction, especially its repulsive part, is the dominant contribution to energy. Structures from region I and II benefit from subsequent optimization while its effect in zone III is minimal. We can imagine that in docking studies, for example, molgri approach can be used on its own for electrostatically-lead process of the ligand approaching the protein while the subsequent process of the ligand docking within the protein structure can be first studied on a grid and then followed by short optimisation runs.

Since we are using a single anion as our association particle, we can regard it as an electric field probe and expect it to identify positively charged regions of state space. Figure 7 shows 100 frames of coiled-coil-Cl\(^{-}\) pseudo-trajectory that had lowest potential energies (before and after minimisation). We indeed observe that electrostatic interaction is the driving force of association since all low energy structures feature anions congregating around the positively charged N-terminus or the amino groups of the lysine side chains. In addition, changes to the lowest-energy structures with minimization are small. This supports our hypothesis that molgri on its own is a suitable tool to identify regions of interest (those featuring an appropriate charge and a free approach vector) in association studies while energy minimization can be used to fine tune the results.

V. COMPUTATIONAL METHODS

To test the pseudo-trajectories we generated with molgri we used GROMACS 2022\(^{[55-58]}\) rerun function to perform point energy calculations of configurations proposed by our algorithm. We also used GROMACS to perform comparison MD simulations. Plots of molecular structures were obtained with VMD 1.9\(^{[59]}\). Some figures were combined and labelled in Inkscape 1.6\(^{[60]}\).

For comparison calculations, we used GROMACS simulations with geometry optimisation (without the rerun option). For the water-water example system, we used the rigid TIP3P\(^{[61]}\) model of water as a force field and performed simulations with 10 fs time steps. In the polypeptide example systems (hFF03-Cl\(^{-}\)) we used the Amber ff99SB-ILDN protein force field\(^{[62]}\), dielectric constant of water 78.4 (at 300K). All simulations were performed at 300 K with particle mesh Ewald (PME) electrostatics and various trajectory lengths (usually up to \(10^5\) time steps). All parameters were kept constant between the pseudo-trajectory calculation and comparison MD run, although some parameters were irrelevant for a rerun (e.g. time step).

GROMACS was also used for the frame-wise minimisation through the steepest descent algorithm. During this process, the backbone and ion where restrained with GROMACS restraint function, which applies a symmetric force on the chosen atoms. This approach allows changes to the local peptide structure without deforming the whole structure or landing in a neighbouring grid point. To encourage the adaptation of the backbone over ion movement we applied position restraints of 1,000,000 kJ/mol nm\(^2\) to Cl\(^{-}\) and 1000 kJ/mol nm\(^2\) to protein backbone atoms. Due to program limitations, the pseudo-trajectory needs to be cut apart, each frame minimised individually and assembled together again.

VI. CONCLUSION

We offer a fresh perspective on state space exploration for molecular binding processes by replacing force-field-based sampling of the two-body interaction with a regular grid. We developed a python workflow that generates relative positions and orientations of two molecules on a regular trans-rot-rot grid, and made the program, including ready-made scripts, available as the Python package.
The quality of a grid-based approach depends heavily on the selection of grid points. To describe a binding process, we used a mesh of translation and rotation grids. For translations, we employed a linear discretisation of the radial distance between the two molecules. For rotations, we implemented six different algorithms and compared them using a quantitative uniformity measure. Algorithms based on polyhedra (Cube 3D and Icosahedron) yielded by far the most uniform rotational grids. To us this was surprising, because given that the quaternions most naturally describe rotations, we expected quaternion-based algorithms to outperform the other algorithms.

By contrast, algorithms based on Euler angles missed some of the known minima of the water dimer, even with high numbers of grid points, and should only be used with caution for molecular systems. In terms of computational costs, polyhedra-based algorithms showed the poorest scaling with increasing number of grid points. However, since the run time to generate large grids with up to 1000 points is below 100 s on a standard workstation, the computational costs for grid generation is negligible compared to the calculation of grid point energies. Moreover, rotational grids are generated once and can be re-used for any molecular system.

Grid points in state space are converted into corresponding molecular structures and stored in an MD-trajectory file format, so that trans-rot-rot grids can easily be interfaced with standard MD programs. The calculation of the grid point energies can thus be accomplished by already existing, highly optimised programmes. The entire method was first applied to two approaching water molecules, showing that the position and depth of potential energy minima found by molgri in combination with polyhedron- or quaternion-based rotational grids matched those found by a stochastic sampling approach. Moreover, we studied the radial and orientational dependency of energy contributions, reproducing known orientated hydrogen bonding patterns for this system and the dependencies of Lennard-Jones and Coulomb energies on interaction distance. While this is not a new result for this simple system, it demonstrates how the untangling of energy contributions on a grid could be performed for more complex systems.

A drawback of our current approach is that it relies on the rigid body approximation, which may be valid for small molecules but is insufficient for induced fit interactions that are commonly encountered in macromolecular systems. This problem can be tackled by replacing point energy calculations with short energy minimisations. We demonstrated this on the system of a coiled-coil peptide binding with a Cl\(^{-}\) anion. The GROMACS energy minimisation routine could handle starting structures with strongly overlapping Lennard-Jones spheres, and in most cases the Cl\(^{-}\) position could be accommodated by the rotation of a side chain or a local adjustment of the backbone. This example also showed that, at large distances, the long-range electrostatic interactions are well-represented by the single-point energy calculations and do not need an energy minimization. One can further improve the accuracy of the grid point energies by running a short constraint simulation and by including solvent molecules in the energy calculation. One could further improve the accuracy of the grid point energies by running a short constraint simulation and by including solvent molecules in the energy calculation.

We view the molgri package as a tool to represent the binding process on a grid with approximate energies, to guide the choice of initial structures for unconstrained sampling, or to guide the placement of umbrella potentials for the construction of free-energy surfaces. With these use cases in mind, the package can be extended in the following directions: (i) providing other non-linear grids, e.g. for cylindrical coordinates, (ii) improving the interface with MD programs, such that molgri seamlessly calls a specific MD program, (iii) extending the interface to quantum chemistry packages, such that grid-point energies can be obtained from electronic-structure calculations, (iv) extending the interface to flexible docking algorithms to improve the handling of induced-fit binding. We are also interested in extending the package to another use case: a rate-model of the binding process constructed on a trans-rot-rot grid using the square-root approximation of the Fokker-Planck equation\(^{23,27}\). For the square-root approximation, the adjacency matrix of the grid, the high-dimensional volumes of the grid cells, and the area of the interface between two neighboring grid cells are required. Thus, full control over the grid is critical. On a systematic trans-rot-rot grid, all mentioned parameters could be calculated once and applied to various molecular systems.

Whether or not the grid-based approach could ever outperform sampling approaches, assuming that ergodicity is fulfilled, is difficult to say. Two aspects of the grid-based approach boost its computational efficiency: First, because grid points are independent, calculation of the grid-point energies can be completely parallelised. Second, because each grid point is evaluated only once, a grid-based approach avoids re-examining the same minima over and over again. Additionally, by their very nature, grid-based energy-calculations should interface easily with adaptive resolution techniques\(^{43}\). On the other hand, because of the high-dimensionality of conformational space, accurate calculations of the grid point energies will have to rely on short stochastic sampling of the orthogonal degrees of freedom in the foreseeable future like we demonstrated in our minimisation routine. If the exploration of the conformational space in orthogonal degrees of freedom is slow compared to the degrees of freedom resolved by the grid, the computational efficiency is dictated by the sampling.

However, the ergodicity assumption is a constant worry in MD simulations. In contrast to sampling approaches, grid-based models of molecular interactions do not need to assume ergodicity. Even with approximate estimates
for the grid point (free) energies, grid-based state exploration is a powerful complement to sampling.

VII. AUTHOR STATEMENTS

A. Competing interests statement

The authors declare there are no competing interests.

B. Author contribution statement

H.Z.: Investigation; Methodology; Formal analysis; Software; Visualization; Writing – original draft; Writing – review & editing. F.H.: Investigation; Methodology; Formal analysis; Visualization; Writing – review & editing. B.G.K.: Conceptualization; Methodology; Funding acquisition; Project administration; Supervision; Writing – review & editing.

C. Funding statement

This research has been funded by Deutsche Forschungsgemeinschaft (DFG) through grant SFB 1349 Fluorin-Specific Interactions - Project ID 387284271, Project A05; SFB 1449 Dynamic Hydrogens at Bio interfaces - Project ID 431232613 - Project C02; RTG 2473 Bioactive Peptides - Project ID 392923329 - Project C01. We acknowledge access to high-performance computers via the Zentraleinrichtung für Datenverarbeitung (ZEDAT) of Freie Universität Berlin.

D. Data availability statement

Input molecular structure files used in this contribution and non-commercial software used to manipulate them are included in our software package molgri. We refer to Section III C for instructions on package acquisition and use.

VIII. SUPPORTING INFORMATION

Detailed pseudo-algorithms and code dependencies for the software implementation described in Section III are provided as supporting information. Documentation and source code are hosted on GitHub [https://github.com/bkellerlab/molecularRotationalGrids].

1 Buckingham, A. The hydrogen bond, and the structure and properties of H2O and (H2O) 2. Journal of molecular structure 1991, 250, 111–118.

2 Rüme, K. F.; Gekle, S.; Netz, R. R. Ion-specific solvation water dynamics: single water versus collective water effects. The Journal of Physical Chemistry A 2014, 118, 11667–11677.

3 Reinmuth, M.; Pramanik, S.; Douglas, J. T.; Day, V. W.; Bowman-James, K. Structural impact of chelation on phosphate, a highly phosphorylated biomolecule. European Journal of Inorganic Chemistry 2019, 2019, 1870–1874.

4 Plattner, N.; Noé, F. Protein conformational plasticity and complex ligand-binding kinetics explored by atomistic simulations and Markov models. Nature communications 2015, 6, 1–10.

5 Casassonovas, R.; Limongelli, V.; Tiwary, P.; Carloni, P.; Parrinello, M. Unbinding kinetics of a p38 MAP kinase type II inhibitor from metadynamics simulations. Journal of the American Chemical Society 2017, 139, 4780–4788.

6 Kahler, U.; Kamenik, A. S.; Waibl, F.; Kraml, J.; Liedl, K. R. Protein-protein binding as a two-step mechanism: Preselection of encounter poses during the binding of BPTI and trypsin. Biophysical journal 2020, 119, 652–660.

7 Wenz, M. T.; Bertazzon, M.; Sticht, J.; Aleksic, S.; Gjorgjevikj, D.; Freund, C.; Keller, B. G. Target recognition in tandem WW domains: complex structures for parallel and antiparallel ligand orientation in h-FBP21 tandem WW. Journal of Chemical Information and Modeling 2021, 89, 1244–1252.

8 Moriss-Andrews, A.; Shea, J.-E. Computational studies of protein aggregation: methods and applications. Annual review of physical chemistry 2015, 66, 619–660.

9 Morozova, S.; Muthukumar, M. Electrostatic effects in collagen fibril formation. The Journal of chemical physics 2018, 149, 163333.

10 Hellmund, K. S.; Von Lospichl, B.; Böttcher, C.; Ludwig, K.; Keiderling, U.; Noirez, L.; Weiß, A.; Mikolajczak, D. J.; Gradzelski, M.; Kosch, B. Functionalized peptide hydrogels as tunable extracellular matrix mimics for biological applications. Peptide Science 2021, 113, e24201.

11 Bolhuis, P. G.; Chandler, D.; Dellago, C.; Geissler, P. L. Transition Path Sampling: Throwing Ropes. Annu. Rev. Phys. Chem. 2002, 53, 291–318.

12 Barducci, A.; Bonomi, M.; Parrinello, M. Metadynamics. Wiley Interdisciplinary Reviews: Computational Molecular Science 2011, 1, 826–843.

13 Zuckerman, D. M.; Chong, L. T. Weighted ensemble simulation: review of methodology, applications, and software. Annual review of biophysics 2017, 46, 43.

14 Bowman, G. R.; Ensign, D. L.; Pande, V. S. Enhanced modeling via network theory: Adaptive sampling of Markov state models. Journal of chemical theory and computation 2010, 6, 787–794.

15 Bruce, N. J.; Ganotra, G. K.; Kokh, D. B.; Sadiq, S. K.; Wade, R. C. New approaches for computing ligand–receptor binding kinetics. Current opinion in structural biology 2018, 49, 1–10.

16 Limongelli, V. Ligand binding free energy and kinetics calculation in 2020. Wiley Interdisciplinary Reviews: Computational Molecular Science 2020, 10, e1455.

17 Plattner, N.; Doerr, S.; De Fabritiis, G.; Noé, F. Complete protein–protein association kinetics in atomic detail revealed by molecular dynamics simulations and Markov modelling. Nature chemistry 2017, 9, 1005–1011.

18 Capelli, R.; Carloni, P.; Parrinello, M. Exhaustive search of ligand binding pathways via volume-based metadynamics. The journal of physical chemistry letters 2019, 10, 3495–3499.

19 Lemeske, C. W.; Fleischer, C.; Kukharenko, O. Efficient Sampling and Characterization of Free Energy Landscapes of Ion–Peptide Systems. Journal of chemical theory and computation 2018, 14, 5476–5488.

20 Batra, J.; Szabó, A.; Caulfield, T. R.; Soares, A. S.; Sahin-Tóth, M.; Radisky, E. S. Long-range electrostatic complementarity governs substrate recognition by human chymotrypsin C, a key regulator of digestive enzyme activation. Journal of Biological Chemistry 2013, 288, 9848–9859.

21 Walde, B. J.; Kraml, J.; Kahler, U.; Spinn, A.; Schaperl, M.; Podewitz, M.; Fuchs, J. E.; Cruciani, G.; Liedl, K. R. Electrostatic recognition in substrate binding to serine proteases. Journal of Molecular Recognition 2018, 31, e2727.
Grid-based state space exploration

22 Chodera, J. D. A simple method for automated equilibration detection in molecular simulations. Journal of chemical theory and computation 2016, 12, 1799-1805.
23 Bicout, D.; Szabo, A. Electron transfer reaction dynamics in non-Debye solvents. The Journal of chemical physics 1998, 109, 2325-2338.
24 Donati, L.; Heida, M.; Keller, B. G.; Weber, M. Estimation of the infinitesimal generator by square-root approximation. Journal of Physics: Condensed Matter 2018, 30, 425201.
25 Heida, M. Convergences of the square-root approximation scheme to the Fokker–Planck operator. Mathematical Models and Methods in Applied Sciences 2018, 28, 2599–2635.
26 Donati, L.; Weber, M.; Keller, B. G. Markov models from the square root approximation of the Fokker–Planck equation: calculating the grid-dependent flux. Journal of Physics: Condensed Matter 2021, 33, 115902.
27 Donati, L.; Weber, M.; Keller, B. G. A review of Girsanov Reweighting and of Square Root Approximation for building molecular Markov State Models. arXiv preprint arXiv:2209.10544, 2022.
28 Euler, L. Nova methodus motum corporum rigidorum degerrimi- nandi. Novi commentarii academiae scientiarum Petropolitanae 1770, 208–228.
29 Kirk, D. Graphics Gems III (IBM Version); Elsevier. 2012.
30 Yershova, A.; LaValle, S. M. Deterministic sampling methods for spheres and SO (3). IEEE International Conference on Robotics and Automation. 2004; pp 3974–3980.
31 Yershova, A.; Jain, S.; Lavalle, S. M.; Mitchell, J. C. Generating uniform incremental grids on SO (3) using the Hopf fibration. The International journal of robotics research 2010, 29, 801–812.
32 Karney, C. F. Quaternions in molecular modeling. Journal of Molecular Graphics and Modelling 2007, 25, 595–604.
33 Diebel, J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Matrix 2006, 58, 1–35.
34 Purser, R. J.; Rančič, M. A standardized procedure for the derivation of smooth and partially overlay grids on the sphere, associated with polyhedra that admit regular griddings of their surfaces; Mathematical principles of classification and construction; Part I: 2011.
35 Sadourny, R. Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. Monthly Weather Review 1972, 100, 136–144.
36 Lindemann, S. R.; Yershova, A.; LaValle, S. M. Algorithmic Foundations of Robotics VI; Springer, 2004; pp 313–328.
37 Kneller, G. R. Superposition of molecular structures using quaternions. Molecular Simulation 1991, 7, 113–119.
38 Fincham, D. Leapfrog rotational algorithms. Molecular Simulation 1992, 8, 165–178.
39 Kol, A.; Laird, B. B.; Leimkuhler, B. J. A sympathetic method for rigid-body molecular simulation. The Journal of chemical physics 1997, 107, 2580–2588.
40 Miller III, T.; Elefteriou, M.; Pattnaik, P.; Ndirango, A.; News, D.; Martyna, G. Symplectic quaternion scheme for biophysical molecular dynamics. The Journal of chemical physics 2002, 116, 8469–8469.
41 Nielsen, S. O.; Bulo, R. E.; Moore, P. B.; Ensing, B. Recent progress in adaptive multiscale molecular dynamics simulations of soft matter. Physical Chemistry Chemical Physics 2010, 12, 12401–12414.
42 Stumpe, M. C.; Grubmüller, H. Aqueous urea solutions: structure, energetics, and urea aggregation. The Journal of Physical Chemistry B 2007, 111, 6220–6228.
43 Heinz, L. P.; Grubmüller, H. Computing spatially resolved rotational hydration entropies from atomistic simulations. Journal of Chemical Theory and Computation 2019, 16, 108–118.
44 Lynden-Bell, R.; Stone, A. Reorientational correlation functions, quaternions and Wigner rotation matrices. Molecular Simulation 1989, 3, 271–281.
45 Wikipedia contributors, Rotation matrix - Wikipedia, The Free Encyclopedia. 2022; [https://en.wikipedia.org/wiki/Rotation_matrix#In_three_dimensions] [accessed 24-October-2022].
46 Hamilton, W. R. On a new species of imaginary quantities, connected with the theory of quaternions. Proceedings of the Royal Irish Academy (1836-1869) 1840, 2, 424–433.
47 Hamilton, W. R. On quaternions and the rotation of a solid body. Proceedings of the Royal Irish Academy. 1850; pp 38–56.
48 Evlero, A. L.; Petropolitanae, N. Problema algebraicvm ob affectiones prorsvs singvlares membrabile. Novi Commentarii Academiae imperialis scientiarum Petropolitanae 1770.
49 Rodrigues, O. Des lois g´eom´etriques qui r´egissent les d´eplacements d’un syst`eme solide dans l’espace, et de la variati- tion des coordonn´ees provenant de ces d´eplacements consid´er´es ind´ependamment des causes qui peuvent les produire. J. Math. Pures Appl 1840, 5, 5.
50 Lynch, K. M.; Park, F. C. Modern robotics; Cambridge University Press. 2017.
51 Wikipedia contributors, Spherical cap — Wikipedia, The Free Encyclopedia. 2022; [https://en.wikipedia.org/wiki/Spherical_cap] [accessed 24-October-2022].
52 “pyti package molgri. 2022; [https://pypi.org/project/molgri/] [accessed 29-October-2022].
53 Github repository molgri. 2022; [https://github.com/bkellerlab/molecularRotationalGrids] [accessed 29-October-2022].
54 Woolfson, D. N. The design of coiled-coil structures and assemblies. Advances in protein chemistry 2005, 70, 79–112.
55 Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. GROMACS: fast, flexible, and free. Journal of computational chemistry 2005, 26, 1701–1718.
56 Abraham, M. J.; Murtola, T.; Schulz, R.; P¨all, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX 2015, 1, 19–25.
57 Lindahl, E.; Hess, B.; van Der Spoel, D. GROMACS 3.0: a package for molecular simulation and trajectory analysis. Molecular modeling annual 2001, 7, 306–317.
58 Berendsen, H. J.; van der Spoel, D.; van Drunen, R. GROMACS: A message-passing parallel molecular dynamics implementation. Computer physics communications 1995, 91, 43–56.
59 Humphrey, W.; Dalke, A.; Schulten, K. VMD: visual molecular dynamics. Journal of molecular graphics 1996, 14, 33–38.
60 Harrington, H.; Engelen, J. Inkscape. 2004; Software available at [http://www.inkscape.org].
61 Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of simple potential functions for simulating liquid water. The Journal of chemical physics 1983, 79, 926–935.
62 Lindorff-Larsen, K.; Piana, S.; Palmo, K.; Maragakis, P.; Klepeis, J. L.; Dror, R. O.; Shaw, D. E. Improved side-chain torsion potentials for the Amber ff99SB protein force field. Proteins: Structure, Function, and Bioinformatics 2010, 78, 1950–1958.
63 Cortes-Huerto, R.; Praprotnik, M.; Kremer, K.; Delle Site, L. From adaptive resolution to molecular dynamics of open systems. The European Physical Journal B 2021, 94, 1–22.