Derivation of Dihedral Torsion Forces

E.A.J.F. Peters

PBS Supplementary Material

Abstract

This note provides a detailed, step-by-step derivation of the analytical forces arising from a dihedral torsion potential. The primary goal is to derive expressions for the gradient of the dihedral angle with respect to atomic positions, which is the key component for calculating forces in a molecular dynamics simulation. We employ a method based on vector cross products and the chain rule, providing all necessary vector identities and intermediate steps. The final expressions are presented in a form that can be directly implemented in code. This document is intended to serve as a pedagogical guide for students implementing force fields in MD simulation codes.

1. Introduction

In molecular dynamics (MD) simulations, the force on each particle is calculated as the negative gradient of a potential energy function, $\mathbf{F} = -\nabla U$. While simple pairwise potentials depend only on the distance between two atoms, many force fields include more complex terms that depend on the geometry of three (angles) or four (dihedrals) atoms. The dihedral, or torsion, angle potential is crucial for modeling the conformational flexibility of molecules.

This note details the derivation of the forces arising from such a potential. We begin by defining the dihedral angle ϕ using the cross products of bond vectors. We then use the chain rule to express the force on each atom in terms of the gradient of $\cos\phi$.

2. Defining the Dihedral Angle

Let the four atoms defining the dihedral be indexed i, j, k, l in sequence. We first define three connecting vectors based on their positions $\mathbf{r}_p$. Note

that these definitions are chosen to match a specific code implementation and may differ from standard conventions.

$$\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$$
$$\mathbf{r}_{kj} = \mathbf{r}_k - \mathbf{r}_j$$
$$\mathbf{r}_{kl} = \mathbf{r}_k - \mathbf{r}_l$$

The dihedral angle, ϕ, is the angle between the plane containing vectors $(\mathbf{r}_{ij}, \mathbf{r}_{kj})$ and the plane containing $(\mathbf{r}_{kj}, \mathbf{r}_{kl})$. We can determine this angle from the dot product of the unit normal vectors of these two planes.

The normal vectors, $\mathbf{n}_j$ and $\mathbf{n}_k$, are found using the cross product:

$$\mathbf{n}_j = \mathbf{r}_{ij} \times \mathbf{r}_{kj}, \text{ and } \mathbf{n}_k = \mathbf{r}_{kj} \times \mathbf{r}_{kl} \tag{1}$$

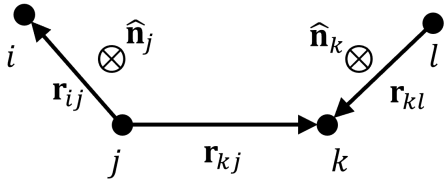
Normalizing these vectors yields the unit normals $\hat{\mathbf{n}}_j$ and $\hat{\mathbf{n}}_k$:

$$\hat{\mathbf{n}}_j = \frac{\mathbf{n}_j}{|\mathbf{n}_j|} \quad , \quad \hat{\mathbf{n}}_k = \frac{\mathbf{n}_k}{|\mathbf{n}_k|} \tag{2}$$

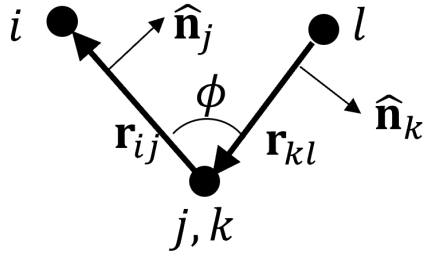
See Fig. 1 for a graphical representation of the dihedral configuration.

The cosine of the dihedral angle is then given by their dot product:

$$\cos\phi = \hat{\mathbf{n}}_j \cdot \hat{\mathbf{n}}_k. \tag{3}$$



(a) Side view of the particles i, j, k and l forming a dihedral. The bond vectors $\mathbf{r}_{ij}$, $\mathbf{r}_{kj}$, $\mathbf{r}_{kl}$ each connect two bonded particles. The normal vectors $\hat{\mathbf{n}}_j$ and $\hat{\mathbf{n}}_k$ point out of the planes defined by $(\mathbf{r}_{ij}, \mathbf{r}_{kj})$, and $(\mathbf{r}_{kj}, \mathbf{r}_{kl})$, respectively.



(b) View along the central bond axis from k to j. The dihedral angle ϕ is the angle between the two normal vectors. the normal vectors are perpendicular to the connecting vectors. The angle between these vectors equals ϕ.

Figure 1: Geometric definition of the dihedral angle ϕ and the associated vectors.

This expression for $\cos\phi$ is the starting point for finding the forces. For a potential $U(\cos\phi)$, the force on a particle p is given by the chain rule:

$$\mathbf{F}_p = -\frac{dU}{d(\cos\phi)} \nabla_p(\cos\phi), \tag{4}$$

with $p = i, j, k$, or l. The main task is to find the gradient term, $\nabla_p(\cos\phi)$.

3. Derivation of the Gradient

The differentiation of Eq. (3) with respect to atomic positions is a bit tedious. To set the stage, let's first provide some definitions and identities that will be used.

3.1. Vector Identities

We will use the following standard vector identities for the scalar triple product and the vector triple product:

$$\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c}) = \mathbf{c}\cdot(\mathbf{a}\times\mathbf{b}) = \mathbf{b}\cdot(\mathbf{c}\times\mathbf{a}), \text{ and} \tag{5}$$

$$\mathbf{a}\times(\mathbf{b}\times\mathbf{c}) = (\mathbf{a}\cdot\mathbf{c})\,\mathbf{b} - (\mathbf{a}\cdot\mathbf{b})\,\mathbf{c} \tag{6}$$

Furthermore, the differential of a unit vector $\hat{\mathbf{r}}$ can be expressed as:

$$d\hat{\mathbf{r}} = \frac{d\mathbf{r} - (d\mathbf{r}\cdot\hat{\mathbf{r}})\,\hat{\mathbf{r}}}{|\mathbf{r}|} = \frac{\hat{\mathbf{r}}\times(d\mathbf{r}\times\hat{\mathbf{r}})}{|\mathbf{r}|}, \tag{7}$$

where the last expression follows from Eq. (6).

3.2. Differential Expansion

Using these identities, the general differentiation of Eq. (3) is given by the product rule:

$$d\cos\phi = d\hat{\mathbf{n}}_j\cdot\hat{\mathbf{n}}_k + \hat{\mathbf{n}}_j\cdot d\hat{\mathbf{n}}_k. \tag{8}$$

The differentials of the unit normals depend on the differentials of the normal vectors themselves:

$$d\hat{\mathbf{n}}_j = \frac{\hat{\mathbf{n}}_j\times(d\mathbf{n}_j\times\hat{\mathbf{n}}_j)}{|\mathbf{n}_j|} \tag{9}$$

$$d\mathbf{n}_j = d\mathbf{r}_{ij}\times\mathbf{r}_{kj} + \mathbf{r}_{ij}\times d\mathbf{r}_{kj} \tag{10}$$

(and similarly for $d\hat{\mathbf{n}}_k$ and $d\mathbf{n}_k$).

3.3. Chain Rule for Atomic Positions

The torsion angle depends on atomic positions through the three connecting vectors. By means of the chain rule, we can relate the gradient with respect to an atom's position to the gradients with respect to the connecting vectors:

$$\begin{aligned}
\nabla_i(\cos\phi) &= \nabla_{ij}(\cos\phi), \\
\nabla_j(\cos\phi) &= -\nabla_{ij}(\cos\phi) - \nabla_{kj}(\cos\phi) \\
\nabla_k(\cos\phi) &= \nabla_{kj}(\cos\phi) + \nabla_{kl}(\cos\phi) \\
\nabla_l(\cos\phi) &= -\nabla_{kl}(\cos\phi)
\end{aligned} \tag{11}$$

3.4. Calculating the Bond Gradients

The vectorial character of the ∇-operator makes direct evaluation cumbersome. A useful trick is to compute the directional derivative $\mathbf{v}\cdot\nabla$ first, and at the end of the derivation, identify the gradient vector by letting $\mathbf{v}$ be the unit vectors.

We can now substitute the gradient operators for the differential d in the relations above. We will evaluate the contribution of each of the three bond vectors ('ij', 'kj', 'kl') to the total gradient. The calculation for the ij contribution, for example, proceeds as follows:

$$\begin{aligned}
\mathbf{v}\cdot\nabla_{ij}\mathbf{n}_j &= (\mathbf{v}\cdot\nabla_{ij}\mathbf{r}_{ij})\times\mathbf{r}_{kj} + \mathbf{r}_{ij}\times(\mathbf{v}\cdot\nabla_{ij}\mathbf{r}_{kj}) = \mathbf{v}\times\mathbf{r}_{kj} \\
\mathbf{v}\cdot\nabla_{ij}\mathbf{n}_k &= 0 \\
\mathbf{v}\cdot\nabla_{ij}\hat{\mathbf{n}}_j &= \frac{\hat{\mathbf{n}}_j\times((\mathbf{v}\cdot\nabla_{ij}\mathbf{n}_j)\times\hat{\mathbf{n}}_j)}{|\mathbf{n}_j|} = \frac{\hat{\mathbf{n}}_j\times((\mathbf{v}\times\mathbf{r}_{kj})\times\hat{\mathbf{n}}_j)}{|\mathbf{n}_j|} \\
&= \frac{\hat{\mathbf{n}}_j\times((\hat{\mathbf{n}}_j\cdot\mathbf{v})\,\mathbf{r}_{kj})}{|\mathbf{n}_j|} = \frac{(\hat{\mathbf{n}}_j\times\mathbf{r}_{kj})\,(\hat{\mathbf{n}}_j\cdot\mathbf{v})}{|\mathbf{n}_j|} \\
(\mathbf{v}\cdot\nabla_{ij}\hat{\mathbf{n}}_j)\cdot\hat{\mathbf{n}}_k &= \frac{(\hat{\mathbf{n}}_j\times\mathbf{r}_{kj})\cdot\hat{\mathbf{n}}_k}{|\mathbf{n}_j|}\,(\mathbf{v}\cdot\hat{\mathbf{n}}_j) = -\frac{(\mathbf{r}_{kl}\cdot\hat{\mathbf{n}}_j)\,|\mathbf{r}_{kj}|^2}{|\mathbf{n}_j||\mathbf{n}_k|}\,(\mathbf{v}\cdot\hat{\mathbf{n}}_j)
\end{aligned} \tag{12}$$

Repeating this exercise for all 3 connecting vectors and 2 normal unit vectors

yields the six necessary components:

$$
\begin{aligned}
(\mathbf{v}\cdot\nabla_{ij}\hat{\mathbf{n}}_j)\cdot\hat{\mathbf{n}}_k &= -\frac{(\mathbf{r}_{kl}\cdot\hat{\mathbf{n}}_j)\,|\mathbf{r}_{kj}|^2}{|\mathbf{n}_j||\mathbf{n}_k|}\,(\mathbf{v}\cdot\hat{\mathbf{n}}_j)\\
\hat{\mathbf{n}}_j\cdot(\mathbf{v}\cdot\nabla_{ij}\hat{\mathbf{n}}_k) &= 0\\
(\mathbf{v}\cdot\nabla_{kj}\hat{\mathbf{n}}_j)\cdot\hat{\mathbf{n}}_k &= \frac{(\mathbf{r}_{kl}\cdot\hat{\mathbf{n}}_j)\,(\mathbf{r}_{kj}\cdot\mathbf{r}_{ij})}{|\mathbf{n}_j||\mathbf{n}_k|}\,(\mathbf{v}\cdot\hat{\mathbf{n}}_j)\\
\hat{\mathbf{n}}_j\cdot(\mathbf{v}\cdot\nabla_{kj}\hat{\mathbf{n}}_k) &= \frac{(\mathbf{r}_{ij}\cdot\hat{\mathbf{n}}_k)\,(\mathbf{r}_{kj}\cdot\mathbf{r}_{kl})}{|\mathbf{n}_j||\mathbf{n}_k|}\,(\mathbf{v}\cdot\hat{\mathbf{n}}_k)\\
(\mathbf{v}\cdot\nabla_{kl}\hat{\mathbf{n}}_j)\cdot\hat{\mathbf{n}}_k &= 0\\
\hat{\mathbf{n}}_j\cdot(\mathbf{v}\cdot\nabla_{kl}\hat{\mathbf{n}}_k) &= -\frac{(\mathbf{r}_{ij}\cdot\hat{\mathbf{n}}_k)\,|\mathbf{r}_{kj}|^2}{|\mathbf{n}_j||\mathbf{n}_k|}\,(\mathbf{v}\cdot\hat{\mathbf{n}}_k)
\end{aligned}
\tag{13}
$$

3.5. Final Expressions for Gradients

By combining the components according to Eq. (8) and recognizing that the relation must hold for any vector $\mathbf{v}$, we can extract the final expressions for the gradients with respect to the bond vectors:

$$\nabla_{ij}(\cos\phi) = -\frac{(\mathbf{r}_{kl}\cdot\hat{\mathbf{n}}_j)\,|\mathbf{r}_{kj}|^2}{|\mathbf{n}_j||\mathbf{n}_k|}\,\hat{\mathbf{n}}_j \tag{14}$$

$$\nabla_{kj}(\cos\phi) = \frac{(\mathbf{r}_{kj}\cdot\mathbf{r}_{ij})\,(\mathbf{r}_{kl}\cdot\hat{\mathbf{n}}_j)\,\hat{\mathbf{n}}_j + (\mathbf{r}_{kj}\cdot\mathbf{r}_{kl})\,(\mathbf{r}_{ij}\cdot\hat{\mathbf{n}}_k)\,\hat{\mathbf{n}}_k}{|\mathbf{n}_j||\mathbf{n}_k|} \tag{15}$$

$$\nabla_{kl}(\cos\phi) = -\frac{(\mathbf{r}_{ij}\cdot\hat{\mathbf{n}}_k)\,|\mathbf{r}_{kj}|^2}{|\mathbf{n}_j||\mathbf{n}_k|}\,\hat{\mathbf{n}}_k \tag{16}$$

Finally, one can combine these expressions according to the chain rule in Eq. (11) and insert them into the force equation, Eq. (4), to compute the torsion force on each individual particle in the dihedral.

4. Implementation and Testing

4.1. Implementation Strategy

The derived formulas should be implemented within the provided C function skeleton. The task is to fill in the ‘todo‘ section.

```
// This function calculates dihedral-torsion forces ...
double calculate_forces_dihedral(struct Parameters *p, struct
    Vectors *v)
```

```
{
    double Epot = 0.0;
    // ... variable declarations ...
    for (size_t q = 0; q < num_dihedrals; ++q)
    {
        // ... indices i, j, k, l assigned ...
        // ... vectors rij, rkj, rkl calculated with MIC ...

        /// \todo Provide the dihedral-torsion force calculation
        ///       and assign forces to particles i, j, k, and l.

        // 1. Calculate normal vectors n_j and n_k.
        // 2. Calculate their magnitudes and check for zero length.
        // 3. Normalize to get n_hat_j and n_hat_k.
        // 4. Calculate cos(phi) from the dot product.
        // 5. Calculate the derivative dU/d(cos(phi)) from the potential.
        // 6. Calculate the bond gradients grad_ij, grad_kj, grad_kl
        //    using the final derived expressions.
        // 7. Calculate the atom gradients grad_i, grad_j, grad_k, grad_l
        //    using the chain rule (Eq. 11).
        // 8. Calculate the force on each atom using F_p = -dU/d(cos(phi)) * grad_p
        //    and add it to the global force array v->f.
        // 9. (Optional) Add the potential energy to Epot.
    }
    return Epot;
}
```

Listing 1: C function skeleton for dihedral force calculation.

4.2. Testing and Verification

Correctly implementing complex analytical formulas is notoriously difficult. A robust testing strategy is essential to ensure your code is bug-free.

1. **Numerical Verification:** The most reliable way to test your analytical forces is to compare them against a numerical calculation using finite differences.

- Write a separate test function where you define a simple, fixed geometry for four atoms (e.g., a planar trans or cis conformation).
- Calculate the forces using your analytical implementation.
- Calculate the forces numerically by slightly displacing each atom p in one direction (e.g., along x) by a small amount δ, recalculating the total potential energy U, and approximating the force component as $F_{p,x} \approx -[U(x_p + \delta) - U(x_p - \delta)]/(2\delta)$.
- The analytical and numerical results should match to within a tolerance determined by δ.

2. **Conservation of Momentum:** In a full NVE simulation, the total force on the system from internal interactions must be zero. Verify that for any given dihedral, the sum of your calculated forces is zero: $\mathbf{F}_i + \mathbf{F}_j + \mathbf{F}_k + \mathbf{F}_l = \mathbf{0}$. This is an excellent and easy-to-implement sanity check.