The Abrasion Fix

David Scott

31st March, 2023

Introduction

This document describes a LAMMPS fix that has been developed to replace much of James Young's `particle_meshs` fix and Enzo's accompanying script. The new fix is called `abrasion` and it may be found in
`https://github.com/EPCCed/lammps/tree/feature-abrasion/src/ABRASION`
along with a very simple test script (`in.test_tetra3`) and input file (`tetrahedron3.mol`).

One aim of the work undertaken was to remove the dependency on CGAL (`https://www.cgal.org`). The second aim was to move the calculation of the displacement velocity from the LAMMPS script into the fix.

Description

The fix computes the displacement velocity of each atom in a group of atoms (the target) caused by an incident particle (an atom). For each atom it also computes an area and a normal vector. The order in which the data are held is:

x component of normal
y component of normal
z component of normal
area
x component of velocity
y component of velocity
z component of velocity

In a script that uses the fix these data are available as `f_{ID}[1]` etc. where `{ID}` is replaced by the name given to the fix.

In addition to the `ID` and `group-ID` arguments common to all fixes, two further arguments are required: `pf` and `mu`.

Implementation

The bulk of the computation is carried out in the `post_force` method.

Firstly, an area and a normal is computed for each atom (see the method `areas_and_normals`). The area is computed in the same way as in James's fix by summing contributions from the area of each facet incident on the atom. The normal vector is obtained by calculating the weighted average of the normals of the facets contributing to the area, with the weight being proportional to the facet's contribution.

The current implementation of `areas_and_normals` assumes that the input file (which describes the molecules) contains exactly one angle from each (triangular) facet. This is different from the previous implementation which expected three angles per facet. The previous implementation can be viewed by looking at the history of the fix in the repository.

Once the areas and normals have been determined, each atom in turn is examined to see if it is subject to any force. If not, nothing more needs to be done.

If there is a force then the relative motion of the incident atom and the atom in the target is considered (see the method `gap_is_shrinking`). If the gap between the two (when projected onto the normal vector of the atom in the target) is widening then nothing more needs to be done.

If the gap is closing then the displacement velocity is calculated (see the method `displacement_of_atom`). This involves the use of lists of neighbours.

Script

The test script needs to be extended in order to mimic the effect of the original script. It is sufficient, however, to demonstrate the working of the fix.

Note the following lines in the script:

```
atom_style      hybrid molecular sphere
bond_style      zero
angle_style     zero
```

The atoms are declared to be of the style `hybrid molecular sphere` so that they have all of the properties that are needed. This leads to the atom section of the input file needing to specify

```
atom-ID, atom-type, x, y, z, molecule-ID, diameter, density
```

The styles of the bonds and angles are `zero` because they are used to convey geometric information only.

Observations

The calculation of the displacement velocity assumes that each atom in the target is interacting with only one incident particle at a time. If there are interactions with more than one particle then at most one of these interactions will produce a displacement.

This could be a problem if a multiple particles are introduced. Also, it could be a problem if two groups of atoms (rigid bodies, say) are allowed to interact. This restriction could be relaxed but a scheme for combining the effects of multiple interactions would be required.