Project: Developing a code for LAMMPS to solve continuum models of granular flow

PI’s: Ken Kamrin (MIT), Jeremy Lechman (Sandia)

Basics and background:

The project we have been engaged in with Sandia and Envalior, which started in July 2023, has the goal of producing an easy-to-use, openly available, well-maintained code that can be used within LAMMPS to enable it to solve a variety of granular continuum models in arbitrary geometries. This project was instigated by LAMMPS’ relatively new capability that enables one to solve basic fluid flow equations by re-imagining the particles as continuum material points and forcing these points to move in accord with the Navier-Stokes equations. This is done using the Smoothed Particle Hydrodynamics (SPH) method, which rewrites the LAMMPS’ particle-interaction law so that the particle forces make them move just as the flow field solutions of the Navier-Stokes equations would have them move. Our goal has been, thus, to modify this code to represent granular continuum models.

Challenges:

1. Accurate granular continuum models are of a different class than fluid models. Namely, granular media have a yield criterion, beneath which they support stress without flow. They flow only when the criterion is met. The correct way to model such rheological behavior is using elasto-plasticity theory, which requires us to treat the continuum body natively as a solid, albeit one that can undergo huge amounts of plastic strain. Thus, we must “teach” LAMMPS how to implement the equations of solid dynamics rather than fluid dynamics.
2. Implementing boundary conditions in SPH is less straightforward than in, e.g., DEM. In SPH interparticle forces have longer interaction distances (each point represents the center of mass of a cloud of particles) such that merely freezing a layer of boundary points is insufficient.
3. The ability to input an arbitrary surface mesh to represent solid objects (such as silo walls, machine parts, etc.) is a developing capability in LAMMPS.
4. The code would need to be completely synced with the latest LAMMPS install to ensure it can be included in the default LAMMPS installation in order for ease of access.
5. We would want the code supplied to run fast, such that the benefit of the continuum model over DEM is exceedingly clear.

Where we are (referencing the Challenges list above):

1. We now have working implementations of generic solid mechanical models running in LAMMPS and producing correct results. Of note, the mu(I) model of dry granular flow combined with gas-like separation/reconsolidation is integrated within the LAMMPS code, as are basic offshoots to include shear dilatancy and strength-dilation coupling.
2. The PI's have created a new method to implement boundary conditions in SPH, which is novel to our knowledge, and has many simplifying factors over existing methods. In particular, the method admits free-slip, no-slip, and frictional boundary interactions with the material. These can all be chosen at will by the user and can vary between surfaces in the same simulation.
3. The boundary condition method has been built to work for a user-supplied STL file for the surface geometry. The mesh that is provided needs to be a surface triangulation of the desired geometry. The ability to import an STL file is a major ease-of-use requirement we have been aiming for. This capability is made possible by recording a distance function on a background voxelization of the space that allows each SPH point to discern which facets of the mesh are potentially closest to it.
4. Several months of work over the last year have been spent on code rebasing, so that the programming done is carefully and correctly merged into the most recent LAMMPS installation. The code rebasing ensures that the latest improvements to the base LAMMPS installation, and particularly the RHEO package, are available with the granular continuum models. For instance, support for multiple phases was implemented in a separate branch concurrently with the initial granular continuum models. Rebasing was required to use the modified interfaces, which will pave the way for merging into mainline LAMMPS as well as allow more complicated interactions (e.g. mixtures) at a later date.
5. Most of the code components run rather quickly, with a similar computational cost compared to the analogous parts of a DEM code with the number of grains equaling the number of SPH points. (Note this is not a fair comparison for physical problems since one SPH point should actually be represented by thousands of grains.) We have, however, identified that the STL-based boundary implementation causes a slow-down due to the extensive list checking that must be done among particles and wall facets to ensure the boundary conditions are satisfied. The slower speed is likely connected to the excessive memory requirements entailed by the background voxelization and lookups.

To-do's:

- Certain surface geometries that are common in engineering contexts are being built into the code as "primitive" elements that will not require a faceted STL mesh. These include cylinders and (truncated) cones, as needed to make typical hoppers, silos, and chutes.
- We are reconsidering the facet-particle lookup method to address the slow-down we have observed when using a meshed STL rigid surface boundary. We expect once this is resolved, it will result in a speedup of about 5x. This expectation is based on how much

faster our beta-tests of the cone-cylinder primitives run compared to the equivalent meshed geometry.

- We will build in a checkable list of constitutive inputs so that the user can easily input the properties of their material. This list will be expanded from our current constitutive models for dry cohesionless media to include cohesion strength and cohesion energy as needed to better model powders.
- Further out, we intend to add a mixture capability to model a flowing air phase mixed within the particles. This will utilize a "two-fluid" type mixture framework, similar to previous frameworks used for fluid-sediment mixtures in the Kamrin Group.
- The finalized code needs to be formally integrated into the LAMMPS default installation and a user-guide needs to be provided in the LAMMPS documentation. An internal guidesheet will also be provided to IFPRI detailing experimental calibration procedures to use the various continuum models.

Effort to date:

The programming effort to date has been headed by my former student (and now project consultant) Sachith Dunatunga. The time spent has been

- Weekly one-hour meetings with me, Sachith, and Joel Clemmer (the Sandia point person on this project). Frequently, these meetings are joined by Pieter Vonk and Eric Grolman, who are the IFPRI correspondents on this project. They have direct access to our codes and have been providing us direct feedback on the usability of the code in key application problems to industry.
- Weekly one-hour coding meetings between Sachith and I.
- Ten hours a week of Sachith's time.

Between Sachith, Joel, and my time, this amounts to many, many man-hours expended so far.

Anticipated costs for the coming year:

I anticipate the costs for the coming year to be the same as those for the previous year (20k). These costs go entirely to cover the consulting and coding time for Sachith. He has conveyed that he is happy to continue at the current rate and pace. Depending on the progress in the next year, it is possible the entire project will be completed by the end of the coming year.