NNLSQ

Grid in Cartesian coordinates

Two ways to generate noise/remove bias: coordinate-wise, or for each grid point (more expensive)
Construct velocity moment matrix and solve

Grid in spherical coordinates

Equal volume/equal radius?

TODO

- [x] NTC collisions
- [x] I/O of particle properties (basic)
- [x] Particle generation (equal weights)
- [x] Properties computation
  - [x] basic ones
  - [x] moments
- [x] I/O of output
- [x] Maxwellian test case (equal weights)
- [x] Two species relaxation (equal weights)
- [x] BKW test case (equal weights)
- [x] Particle generation (variable weight)
- [x] Grid merging
- [ ] Octree merging
  - [ ] Final tests for bin splitting
  - [x] Basic compute (ndens, nparticles) for refinement
    - [x] Tests
  - [ ] Bin property computation
    - [ ] Tests
  - [ ] New particle computes
    - [ ] Tests

- [ ] mixing rule VHS creator
- [ ] The science begins

TODO: features

- [ ] Add time to output (since we can change dt on the fly)
- [ ] Compute sigma_g_vhs directly (to avoid additional multiplication)
- [ ] Avoid duplicate computation of particle indices in grid-based merging? Avoid second loop? Use Welford's algorithm + initial estimate of mean as middle of box? Or just shift data?
- [ ] Logging struct? (Union of nothing/actual logging struct, write out stuff like "increased octree merging buffer", "increase particle array size")

TODO: tests

- [ ] energy / momentum conservation in scattering
- [x] correct indexing
- [x] BKW var weight reference solution
- [ ] 1D - no merging, particles don't switch cells during variable weight collisions!!!

Misc thoughts

Best to do octree merging in reverse order? So that less shifting around of particle indices? Loop of copying instead of deepcopy of slice in octree merging - would that allocate less?

Comparisons

- [ ] Merging grid-based: pass slice/without species? is that any different?
- [ ] Make more structs immutable - is that better?