pkdgrav FAST_AGGS Documentation

Last Update 6/3/20

Created by Joe DeMartini 9/16/19

1 OVERVIEW

FAST_AGGS is a pkdgrav compile option that drastically speeds up aggregate particle calculations performed in the code, under the condition that the number of free bodies is comparable to the number of aggregates, by using a binary search algorithm, followed by a cache line approach which exploits particle ordering on the processor, to find an aggregate's constituent particles, rather than doing a brute-force search, as in the old code. The specific use criterion is: the FAST_AGGS compile option should not be used when there are very few aggregates made up of very many particles in the problem, but **only when there are very many aggregates each containing very few particles**. Only in the latter case will the efficiency increase when using FAST_AGGS.

Notes/Requirements:

1. FAST_AGGS is not compatible with the AGGS_IN_PATCH compile option.
2. FAST_AGGS requires that particles and aggregates be in the same order on the processor: if aggregate 0 is a dumbbell, it must contain particles with iOrder number 0 and 1, aggregate 1 must contain particles with iOrder 2 and 3, etc. This is a key stipulation of the updated searching algorithm in order to gain maximum increased efficiency, and is not always upheld when aggregates are added using rpx.
3. FAST_AGGS requires that, in simulations with both aggregates and single spheres, the particles in aggregates be listed first on the processor, followed by the individual spherical particles. When ordering particles in these types of simulations, be careful to abide by the above ordering rule in bullet 2, and the necessary size ordering for DEM_FIXED_BALL (See SSDEM.pdf page 4).
4. Different options of DEM_FIXED_BALL require different kinds of particle size ordering. Since aggregates must be in iOrder number order for this algorithm, aggregates constructed of different-sized particles are not compatible with both FAST_AGGS and DEM_FIXED_BALL.

For a specific, well-commented example of how the new code is written, see pkdAggsGetAccel and its associated functions in SOURCE/src/pkdgrav/aggs.c.

2 Setup

In order to make use of FAST_AGGS, one must uncomment both of the following options in Makefile.in:

```
USE_AGGREGATES=true
USE_FAST_AGGS=true
```

Then one can recompile the code and the new routines will take effect.

Again, we note that FAST_AGGS is not compatible with the AGGS_IN_PATCH compile option. Otherwise, all restrictions that previously applied to Aggregates will apply here as well.

3 Backwards Compatibility

FAST_AGGS is entirely backwards-compatible with the older aggregate routines in pkdgrav. To use the older aggregate routines, simply leave the USE_FAST_AGGS=true option commented in Makefile.in, while uncommenting USE_AGGREGATES=true. This will revert to the older versions of the functions used for aggregate calculations.

4 Affected Functions

```
pkdAggsCountPart
pkdAggsSetSpacePos
pkdAggsSetSpaceVel
pkdAggsSetSpaceSpins
pkdAggsGetAccel
pkdAggsGetTorque
pkdAggsSetBodyPos
pkdAggsGetCOM
pkdAggsGetAxesAndSpin

pkdAggsSetMassDEM (formerly pkdDEMAggsSetMass)
pstAggsSetMassDEM

msrAggsGravity
msrAggsFind
```

The first set are the functions to which a binary search was added for improved efficiency. The first function in the second set had a binary search was added, and both functions were renamed, updated, and moved to files/locations where they fit better (pkdAggsSetMassDEM to aggs.c and pstAggsSetMassDEM changed and updated in pst.c under the AGGS banner, rather than DEM). The last set are functions in master.c to which a call to msrReorder was added, so that the particles are in aggregate number order on the processor - this is an important step for the new algorithm that searches for particles in an aggregate.