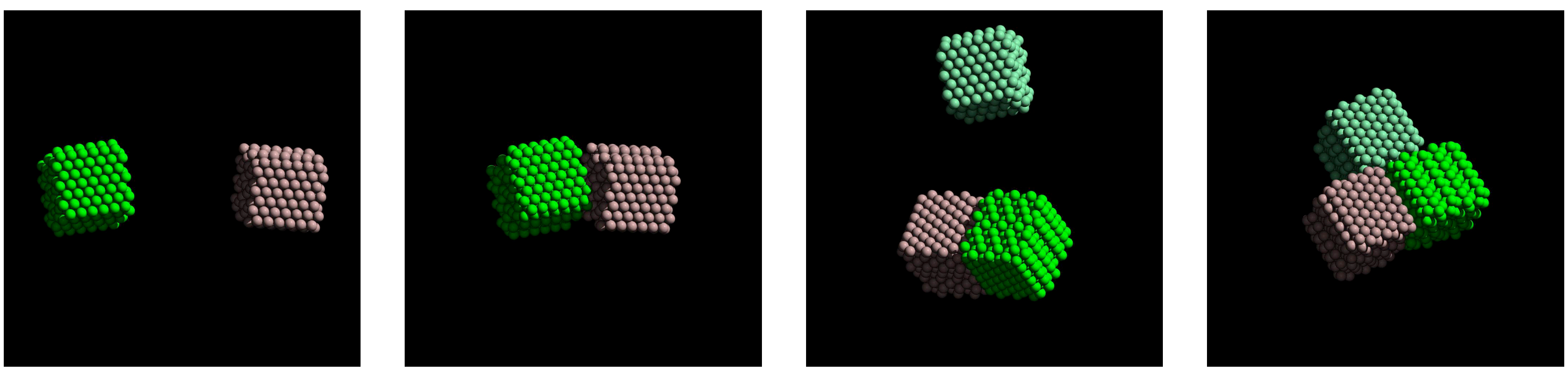
N Rigid-body Dynamics

Derek C. Richardson

University of Maryland

With:
Patrick Michel (Obs. Côte d'Azur)
Randall Perrine (UMd)
Stephen Schwartz (UMd)
Kevin Walsh (Obs. Côte d'Azur)



Very Brief Outline

- CollisionAL systems
  - With real collisions!
- Simulating sphere-sphere collisions
  - Methods and complications.
- Simulating (non-spherical) rigid bodies
  - Methods and applications.
- New directions
  - Cohesion, granular dynamics, etc.

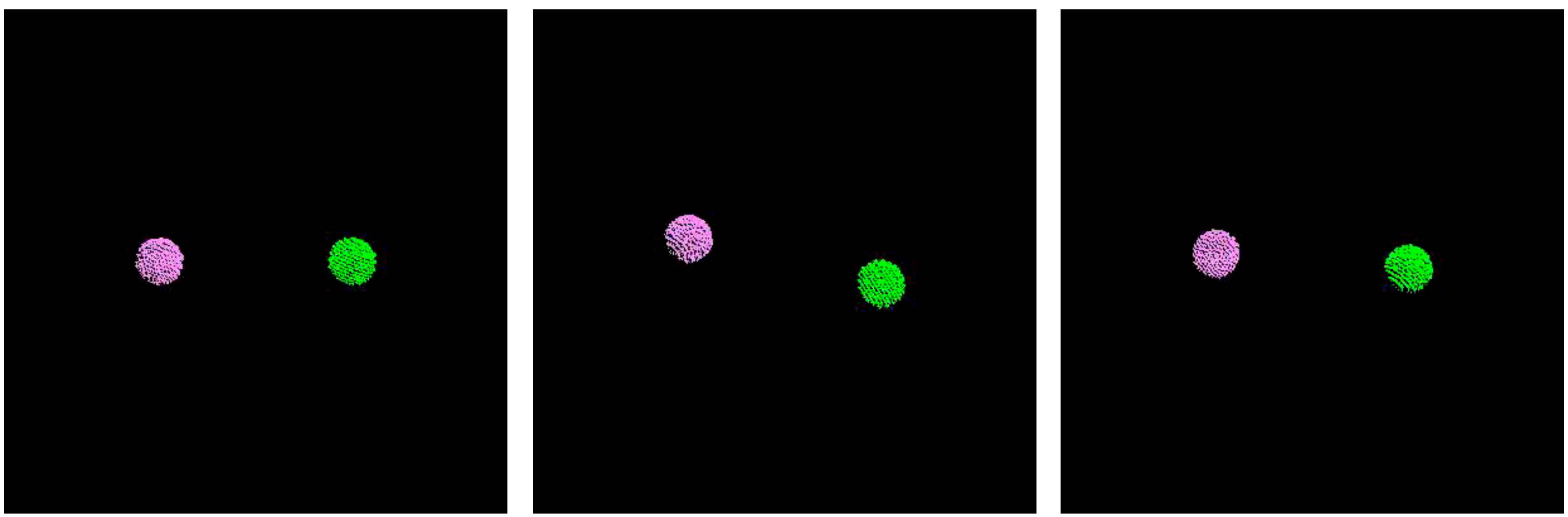
REVIEW: Richardson et al. 2009, P&SS 57, 183

Collisional Systems

- Here we are concerned not only with close gravitational encounters, but also physical collisions: $|\mathbf{r}_i - \mathbf{r}_j| = s_i + s_j$.
- In astrophysics, usually restricted to planetary dynamics:
  - Planet formation (planetesimal accretion).
  - Planetary rings.
  - Granular dynamics.

Physical Collisions in Astrophysics

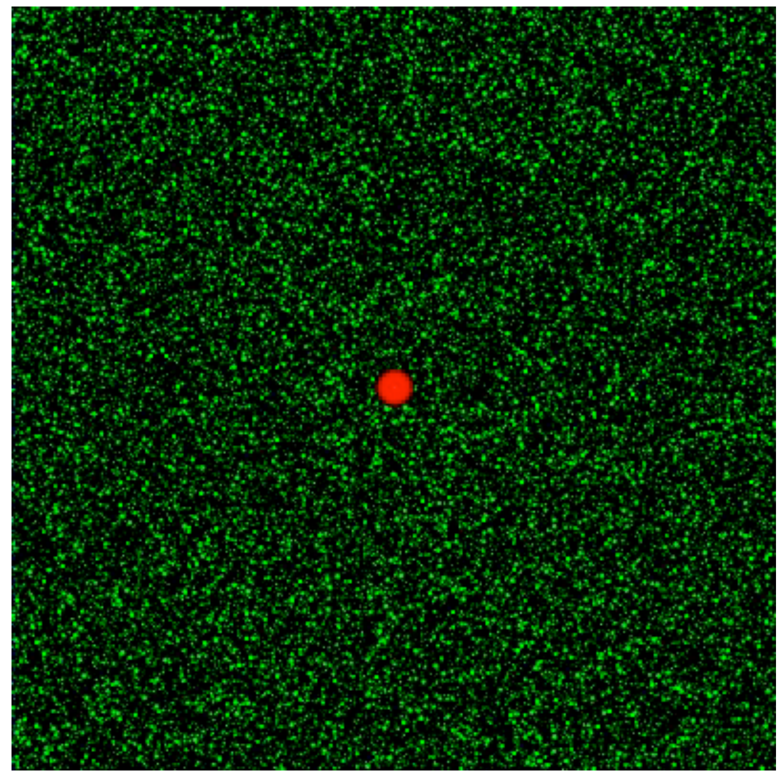
- Planetesimal accretion
  - Gravity + collisions involving rigid particles or groups of rigid particles with some dissipation law and possible fragmentation, etc.



Leinhardt et al. 2000, Icarus 146, 133

Physical Collisions in Astrophysics

- Planetary rings
  - Gravity + collisions in tidal field of a planet, with dissipation and possible sticking and/or fragmentation.

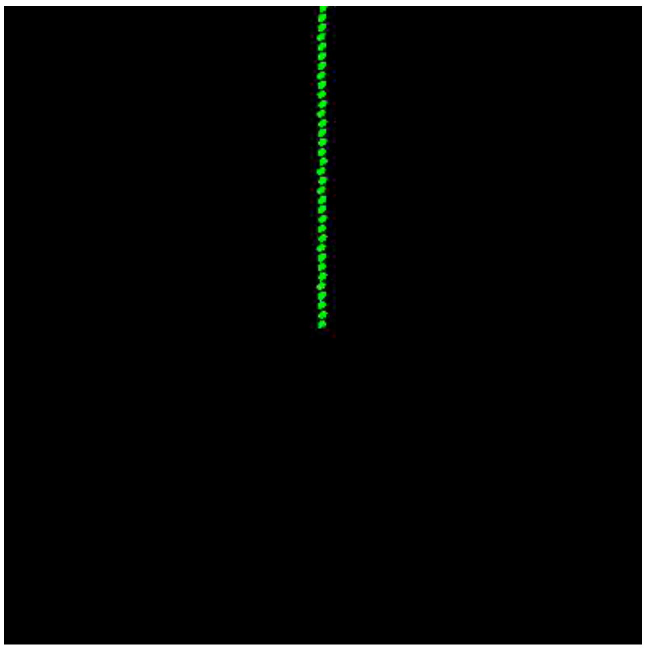


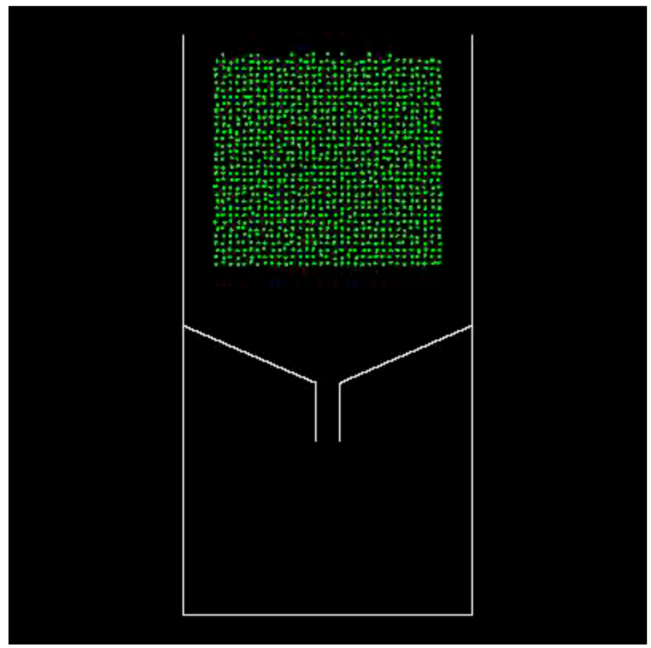
Ring patch with embedded moonlet

Tiscareno et al. 2006, Nature 440, 648

Physical Collisions in Astrophysics

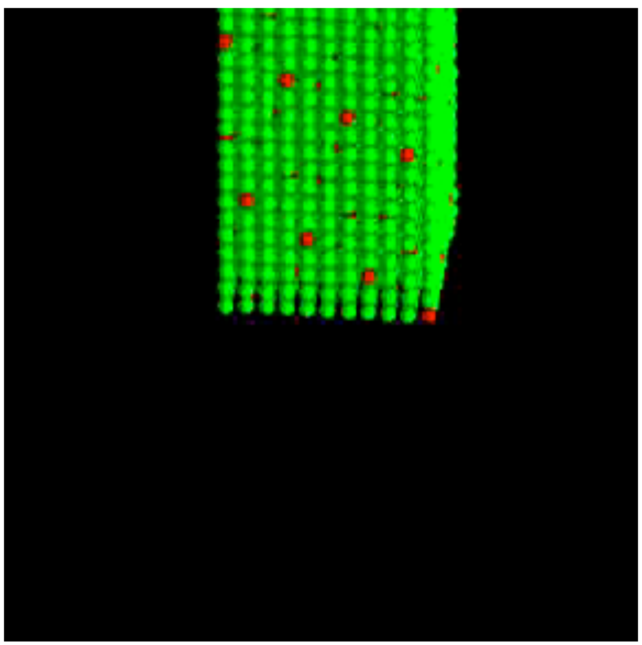
- Granular dynamics
  - Collisions in uniform gravity field, usually with bouncing only, but possibly with sticky "walls."
  - Applications: regolith motion, sample return.

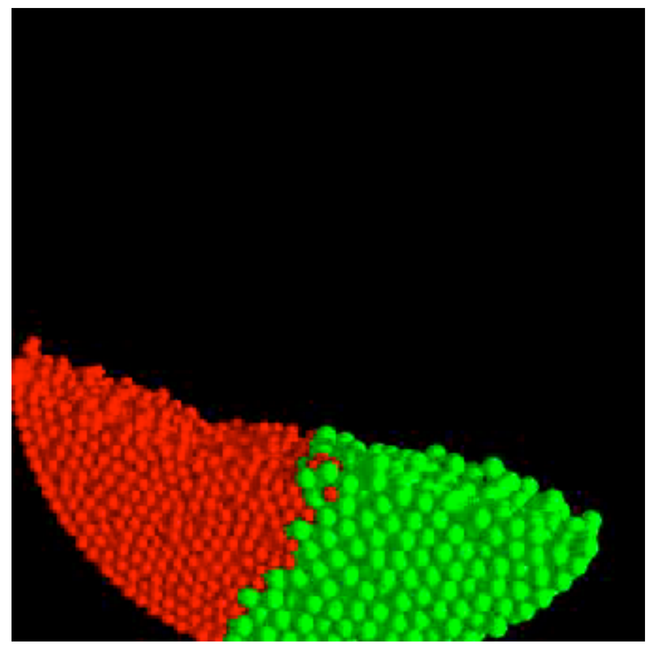




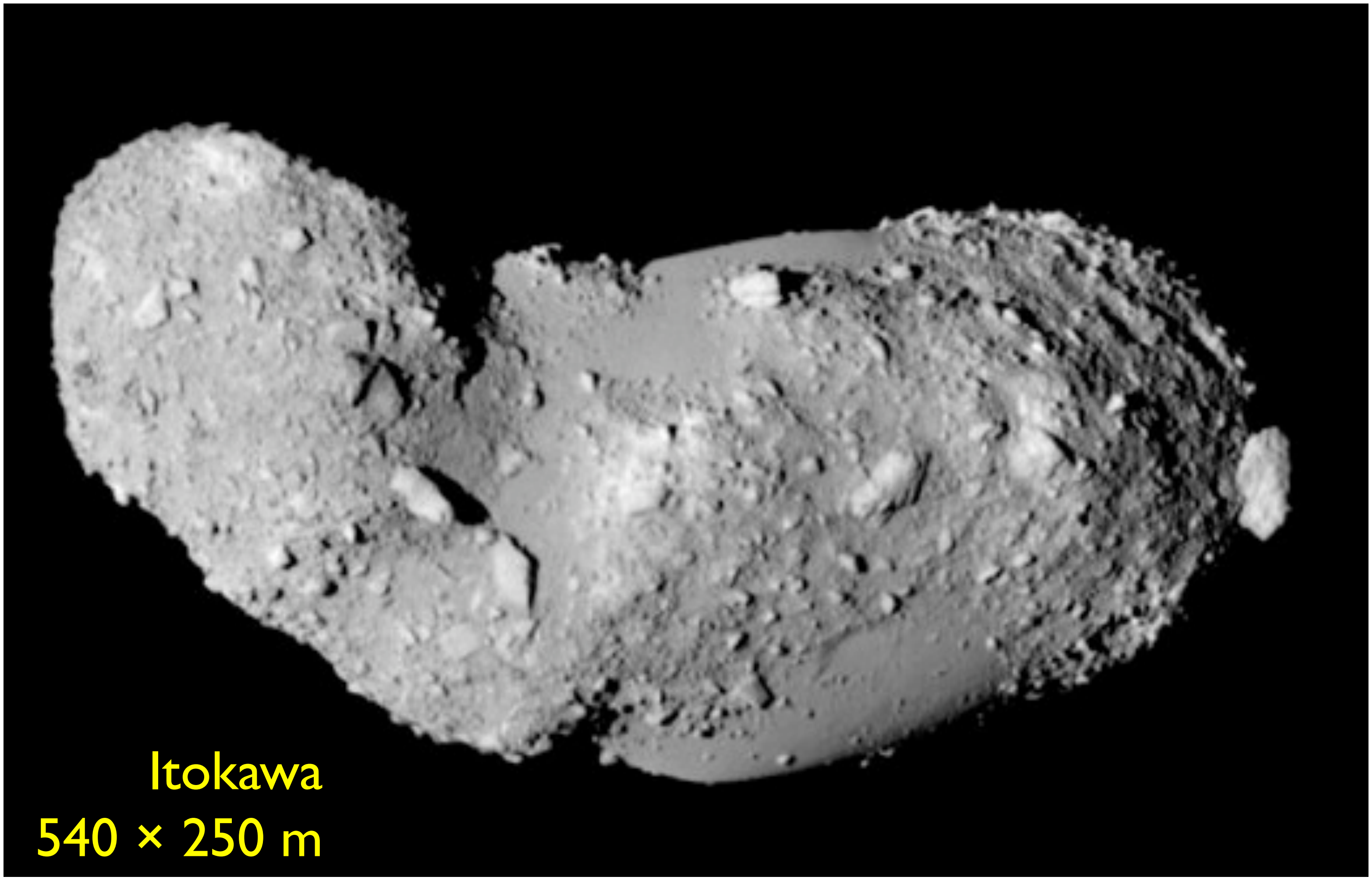
Physical Collisions in Astrophysics

- Granular dynamics
  - Collisions in uniform gravity field, usually with bouncing only, but possibly with sticky "walls."
  - Applications: regolith motion, sample return.

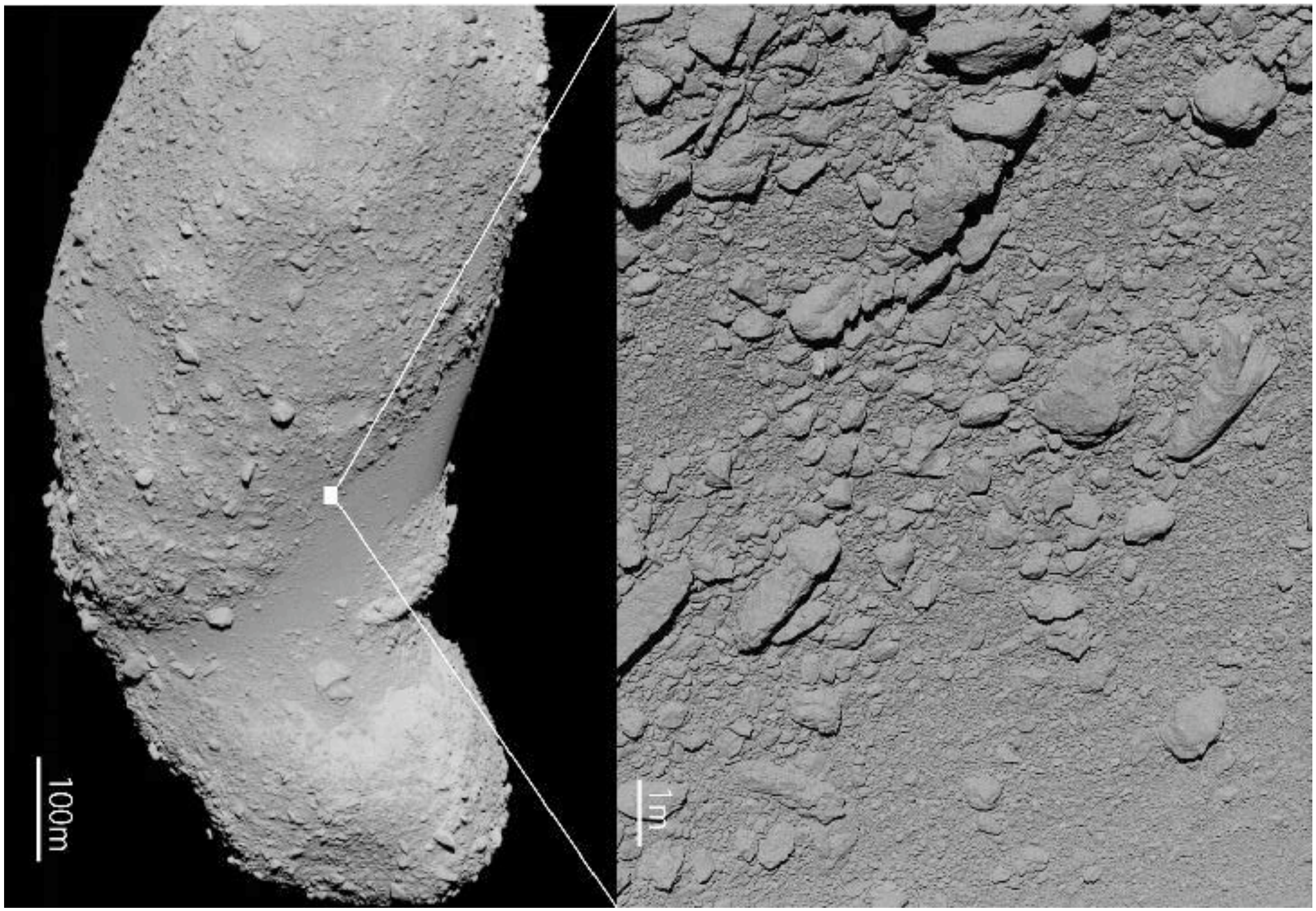




Rubble is out there…



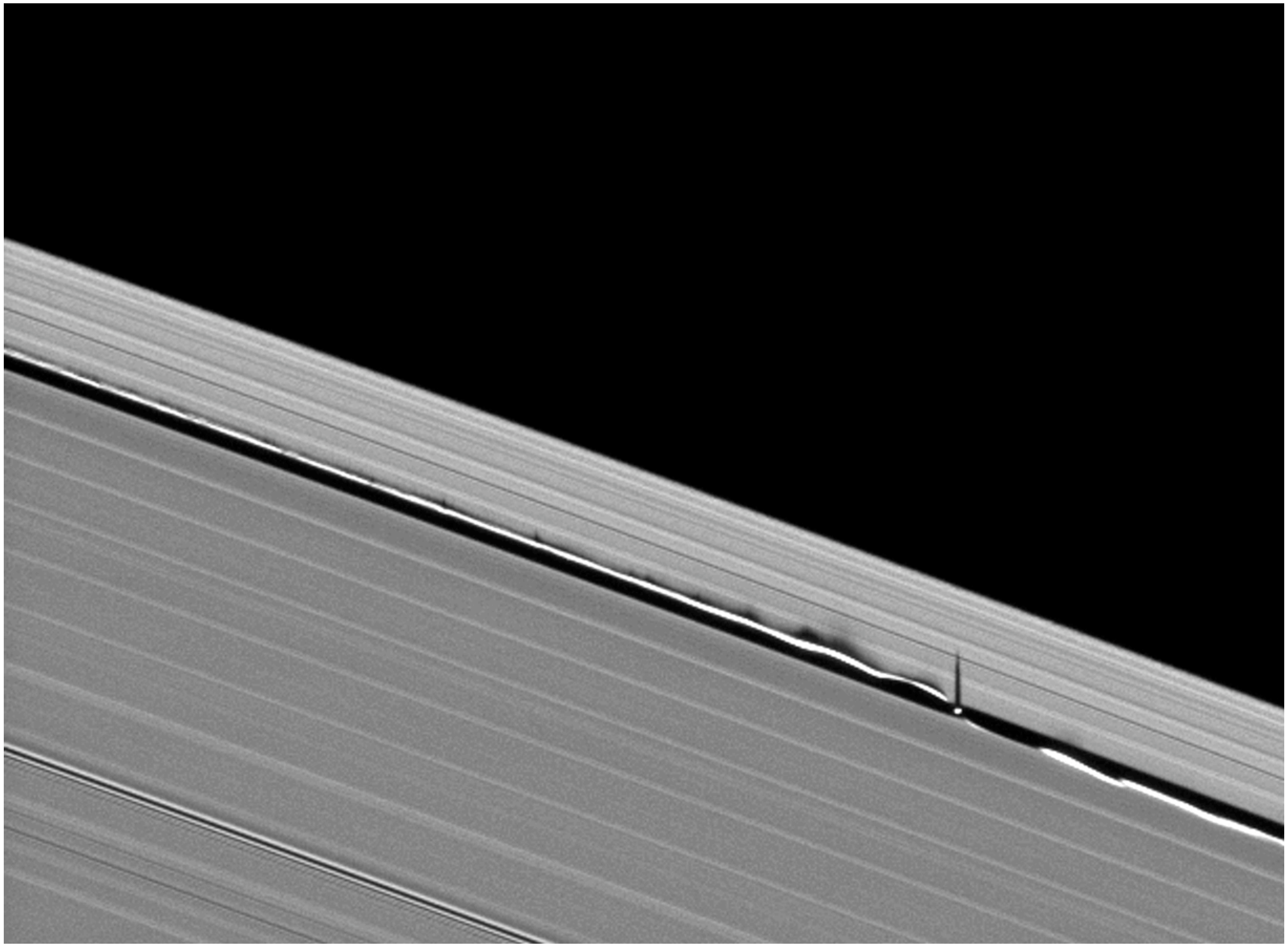
Rubble is out there…



Rubble is out there…

Image courtesy JAXA/ISIS

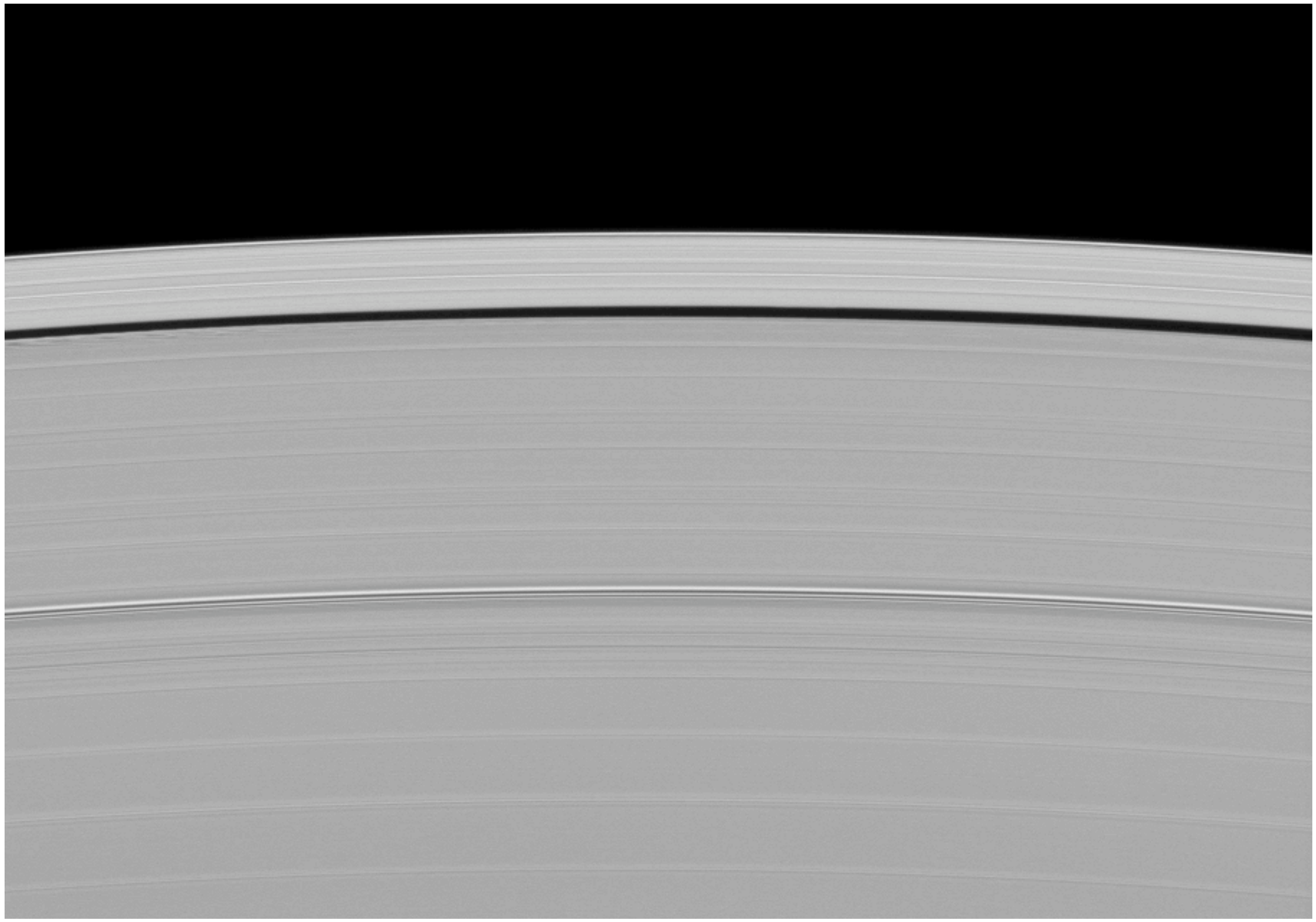
Rubble is out there…



Daphnis casting a shadow

Cassini Equinox Mission

Rubble is out there…



Daphnis casting a shadow (movie)

Cassini Equinox Mission

Collisional systems

- ADVANTAGES:
  1. No singularities.
     - Particles touch before $|\mathbf{r}| \rightarrow 0$. No softening!
  2. Minimum (gravitational) timestep bounded.
     - $h = \eta/(G\rho)^{1/2}$, ρ = maximum density, η ~ 0.03.
- CHALLENGE:
  - Need to predict when collisions occur (or deal with them after the fact), therefore need efficient *neighbor-finding algorithm*.

Sphere-sphere Equations of Motion

- Same as for point particles:

$$\ddot{\mathbf{r}}_i = -\sum_{j \neq i} \frac{G m_j (\mathbf{r}_i - \mathbf{r}_j)}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

- Can use any standard ordinary differential equation integrator (see Scott's talk!).
- Turns out 2nd-order leapfrog is particularly advantageous.

Second-order Leapfrog

- Kick-drift-kick (KDK) scheme:

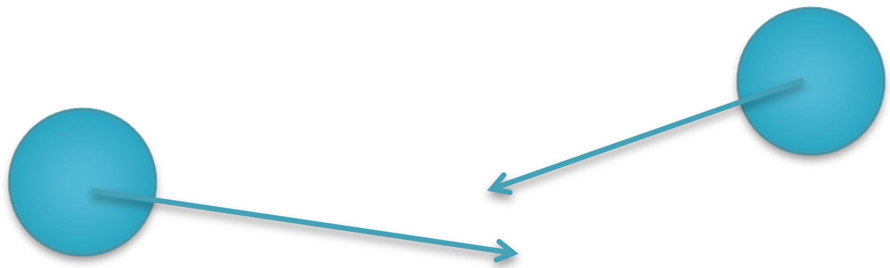
$$\dot{\mathbf{r}}_{i,n+1/2} = \dot{\mathbf{r}}_{i,n} + (h/2)\ddot{\mathbf{r}}_{i,n} \quad \text{“kick”},$$

$$\mathbf{r}_{i,n+1} = \mathbf{r}_{i,n} + h\dot{\mathbf{r}}_{i,n+1/2} \quad \text{“drift”},$$

$$\dot{\mathbf{r}}_{i,n+1} = \dot{\mathbf{r}}_{i,n+1/2} + (h/2)\ddot{\mathbf{r}}_{i,n+1} \quad \text{“kick”},$$

- Notice the drift is linear in the velocities —exploit this to search for collisions.

Collision Prediction



$$\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$$
$$\mathbf{v} = \mathbf{v}_2 - \mathbf{v}_1$$

Collision condition at time *t*:

$$v^2 t^2 + 2(\mathbf{r} \cdot \mathbf{v})t + r^2 = (s_1 + s_2)^2$$

Solve for *t* (take smallest positive root):

$$t = \frac{-(\mathbf{r} \cdot \mathbf{v}) \pm \sqrt{(\mathbf{r} \cdot \mathbf{v})^2 - [r^2 - (s_1 + s_2)^2]v^2}}{v^2}$$

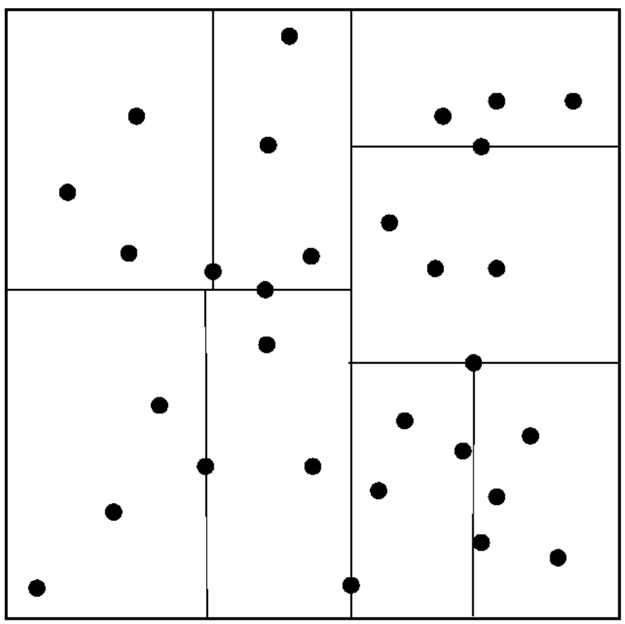
Neighbor Finding

- To check all particle pairs for possible collision carries the same penalty as direct force summation: $O(N^2)$.
- Instead, take advantage of the hierarchical nature of a *tree code* to reduce the neighbor search to $O(N_s \log N)$, where N_s = number of neighbors to find.
  - This is equivalent to what is needed for an SPH "gather" step.

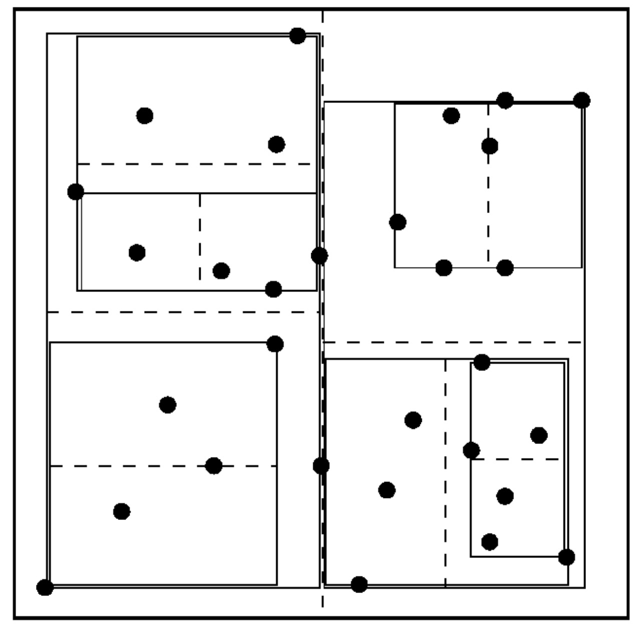
Some words about pkdgrav/gasoline

- First developed at U Washington, this is a parallel, hierarchical gravity solver for problems ranging from cosmology to planetary science.
- "Parallel *k*-D Gravity code" = pkdgrav.
- Gasoline is pkdgrav with SPH.
- Not released into the public domain (yet).
- If you're interested in using it, see me!

Spatial Binary Tree



k-D Tree

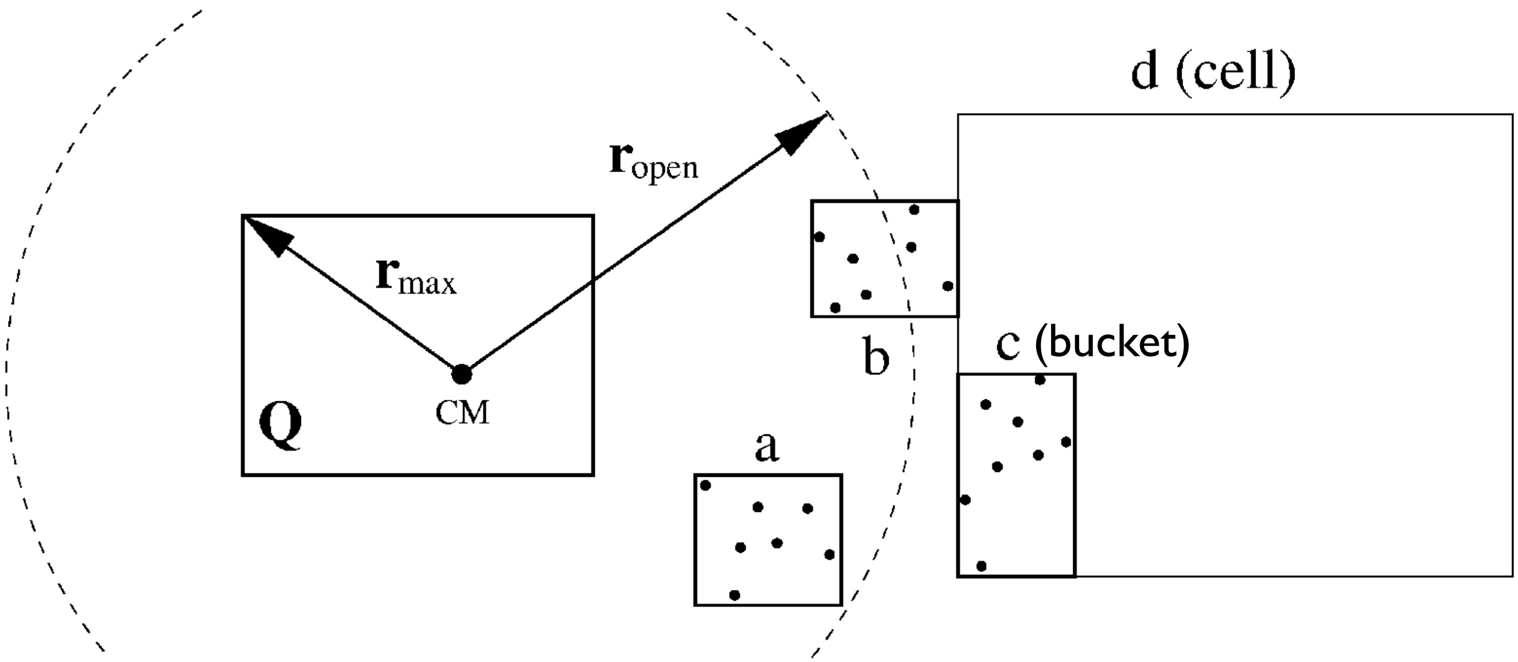


k-D with Squeeze

Tree Walking

- Construct particle-particle and particle-cell interaction lists from top down.
- Define opening ball (based on *critical opening angle* θ) to test for ball-bucket intersection.
  - If bucket outside ball, apply multipole (c-list).
  - Otherwise open cell and test its children, etc., until leaves reached (which go on p-list).
- Nearby cells have similar lists: amortize.

Tree Walking



Note multipole Q acceptable to all particles in cell d.

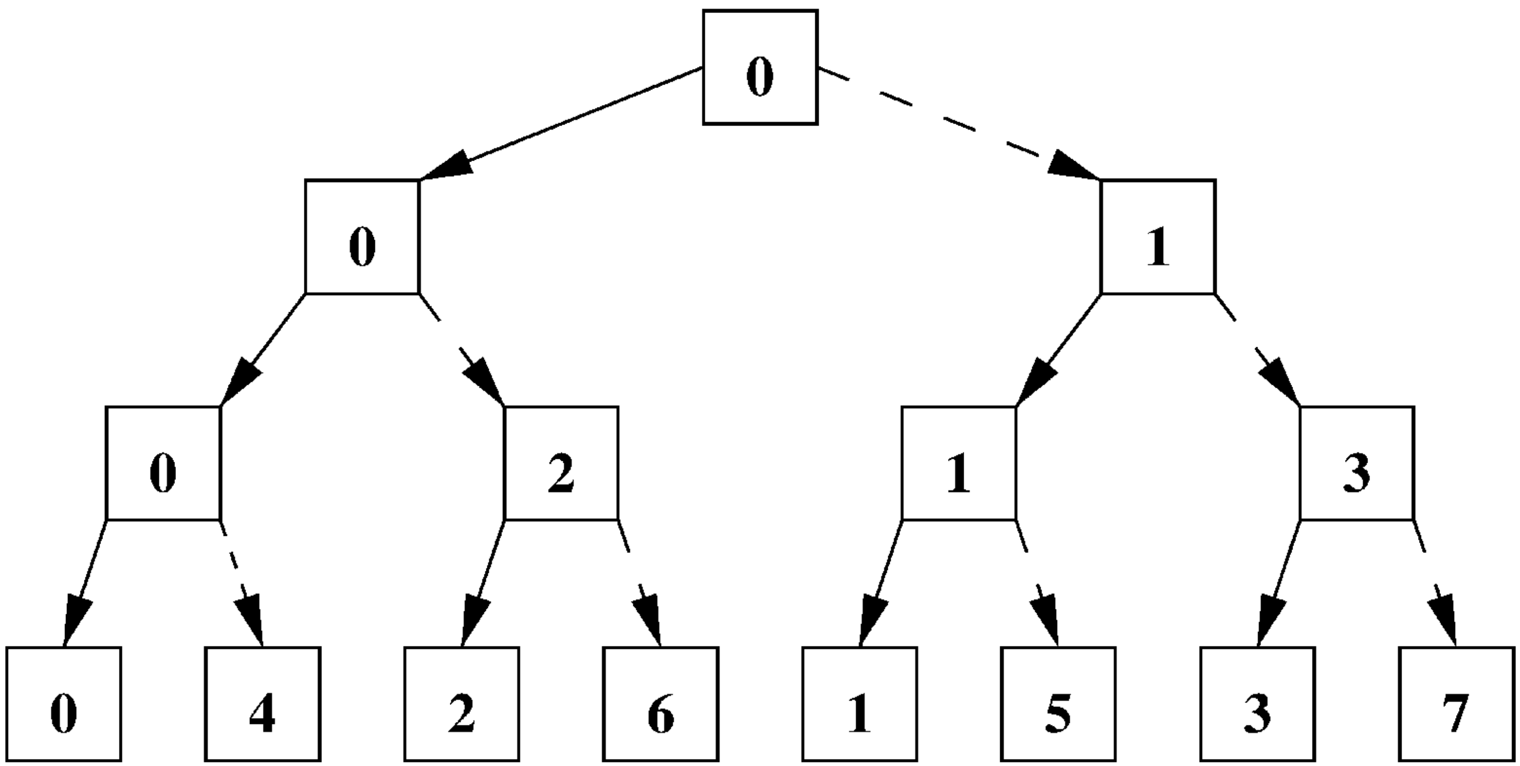
Other Issues

- Multipole expansion order.
  - Use hexadecapole (best bang for buck).
- Force softening (for cosmology).
  - Use spline-softened gravity kernel.
- Periodic boundary conditions.
  - Ewald summation technique available.
- Time steps.
  - Multistepping available (adaptive leapfrog).

Parallel Implementation

- Master layer (serial).
  - Controls overall flow of program.
- Processor Set Tree (PST) layer (parallel).
  - Assigns tasks to processors.
- Parallel *k*-D (PKD) layer (serial).
  - MIMD execution of tasks on each processor.
- Machine-dependent Layer (MDL, separate set of functions).
  - Interface to parallel primitives.

Domain Decomposition



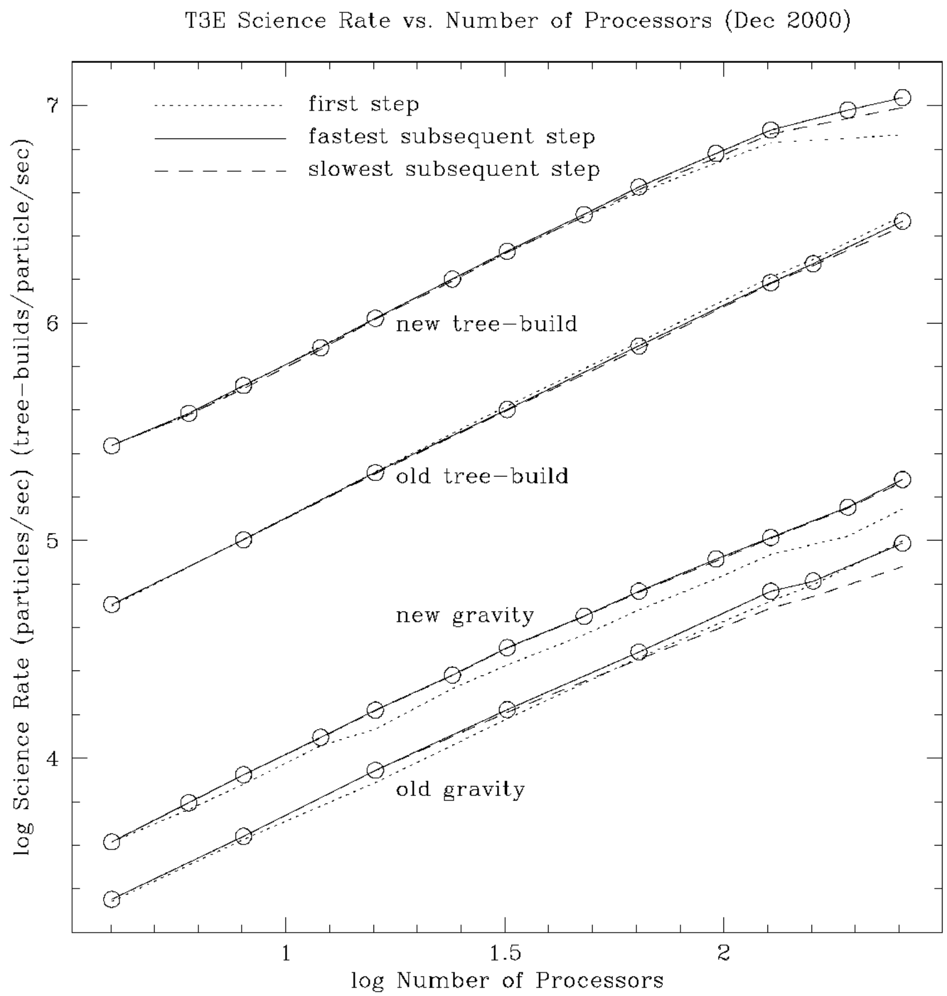
Binary tree balanced by work factors. Nodes construct local trees.

Scaling at Fixed Accuracy

Clustered cosmology simulation

($N = 3 \cdot 10^6$)

($\theta = 0.8$)



Back to collisions…

- How many neighbors to search?
  - Close-packed equal-size spheres have a maximum of 12 touching neighbors.
  - For less-packed situations, only concern is a more distant fast-moving particle.
  - Typically use $N_s \sim 16$–32, with h small enough to ensure no surprises.
  - Can also search for all neighbors within a fixed ball radius (e.g. $R = 3vh$), but can end up with many more neighbors to check.

Collision Resolution

Post-collision velocities and spins:

$$\begin{aligned}
\boldsymbol{v}_1' &= \boldsymbol{v}_1 + \frac{m_2}{M}\left[(1+\epsilon_n)\boldsymbol{u}_n + \beta(1-\epsilon_t)\boldsymbol{u}_t\right], \\
\boldsymbol{v}_2' &= \boldsymbol{v}_2 - \frac{m_1}{M}\left[(1+\epsilon_n)\boldsymbol{u}_n + \beta(1-\epsilon_t)\boldsymbol{u}_t\right], \\
\boldsymbol{\omega}_1' &= \boldsymbol{\omega}_1 + \beta\frac{\mu}{I_1}(1-\epsilon_t)(\,\boldsymbol{s}_1 \times \boldsymbol{u}), \\
\boldsymbol{\omega}_2' &= \boldsymbol{\omega}_2 - \beta\frac{\mu}{I_2}(1-\epsilon_t)(\,\boldsymbol{s}_2 \times \boldsymbol{u}),
\end{aligned}$$

where:

$M = m_1 + m_2$, $\mu = m_1 m_2/M$, $\boldsymbol{u} = \boldsymbol{v} + \boldsymbol{\sigma}$, $\hat{\boldsymbol{n}} = \boldsymbol{r}/r$, $\boldsymbol{u}_n = (\boldsymbol{u}\bullet\hat{\boldsymbol{n}})\hat{\boldsymbol{n}}$, $\boldsymbol{u}_t = \boldsymbol{u} - \boldsymbol{u}_n$, $\boldsymbol{s}_1 = s_1\hat{\boldsymbol{n}}$, $\boldsymbol{s}_2 = -s_2\hat{\boldsymbol{n}}$, $\boldsymbol{\sigma}_i = \boldsymbol{\omega}_i \times \boldsymbol{s}_i$, $\boldsymbol{\sigma} = \boldsymbol{\sigma}_2 - \boldsymbol{\sigma}_1$, $\beta = 2/7$ for spheres, and $I_i = (2/5)\, m_i R^2$.

What about ε_n & ε_t?


GROVE

What about ε_n & ε_t?

What about ε_n & ε_t?

Collision Handling in Parallel

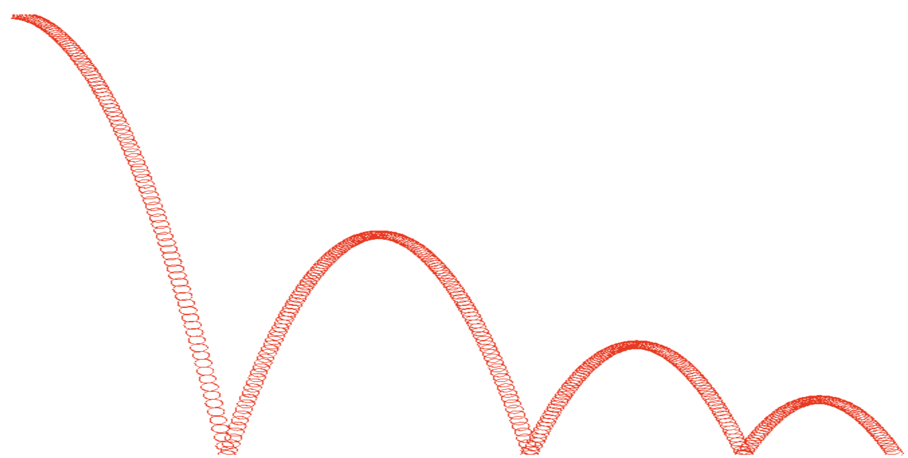
- Each processor checks its particles for next collision during current drift interval (could involve off-processor particle).
- Master determines which collision goes next and allows it to be carried out.
- Check whether any future collision circumstances changed.
- Repeat until all collisions occurring within this drift step resolved.

Complications

- The "restitution" model of billiard-ball collisions is only an approximation of what really happens.
- Collisions are treated as instantaneous (no flexing) and single-point contact.
- This leads to problems:
  - Inelastic collapse.
  - Missed collisions due to round-off error.

Inelastic Collapse

- A rigid ball bouncing on a rigid flat surface must come to rest, but in the restitution model this requires an infinite number of increasingly smaller bounces to occur in a finite time (Zeno's paradox!).



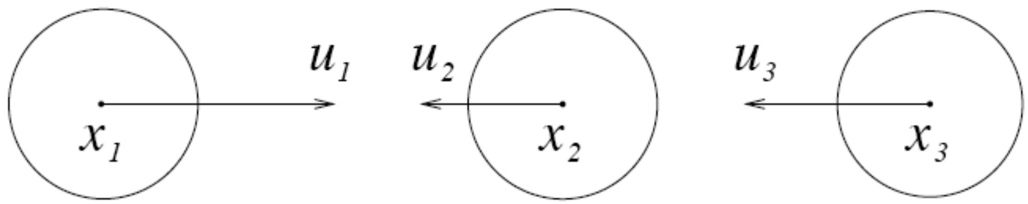
Could also occur between 2 self-gravitating spheres in free space.

Inelastic Collapse

- How to fix it?
  - Impose minimum impact speed $v_{\min}$ below which $\varepsilon_n \rightarrow 1$ (no dissipation).
  - Choose $v_{\min}$ so that this "vibration energy" is small compared to energy regimes of interest.
    - Petit & Hénon 1987a "sliding phase."
  - OR, force particles/surfaces to come to rest with one another—but this causes other complications, especially with self-gravity.
    - Requires introducing surface normal forces.

Inelastic Collapse

- Can occur in other circumstances, even *without* gravity, e.g.



$$\begin{pmatrix} u_1'' \\ u_2'' \\ u_3'' \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(1-\epsilon) & \frac{1}{2}(1+\epsilon) & 0 \\ \frac{1}{4}(1-\epsilon^2) & \frac{1}{4}(1-\epsilon)^2 & \frac{1}{2}(1+\epsilon) \\ \frac{1}{4}(1+\epsilon)^2 & \frac{1}{4}(1-\epsilon^2) & \frac{1}{2}(1-\epsilon) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

For collapse to occur, the matrix must have at least one real eigenvalue between 0 & 1. This is satisfied if $0 < \epsilon < 7 - 4\sqrt{3}$ (~ 0.072).

Inelastic Collapse, continued

- Previous example was in 1-D, but problem occurs in 2- & 3-D as well.
- Crucially, as $N \rightarrow \infty$, $\varepsilon_{n,\mathrm{crit}} \rightarrow 1$!
- How to fix it?
  - If distance travelled since last collision small (factor f_{crit}) compared to the particle radius, set $\varepsilon_n = 1$ for next collision (typically $f_{\mathrm{crit}} \sim 10^{-6}$–$10^{-3}$).
  - Other strategy (not implemented): store some fraction of impact energy as internal vibration to be released stochastically.

Round-off Error and Overlaps

- Despite precautions, if there are many collisions between many particles in a timestep, round-off error can cause a collision to be missed.
- In this case, some particles may be overlapping at start of next step.
  - Minimize by good choices of h, v_{min}, and f_{crit}.
  - But sometimes that's not enough…

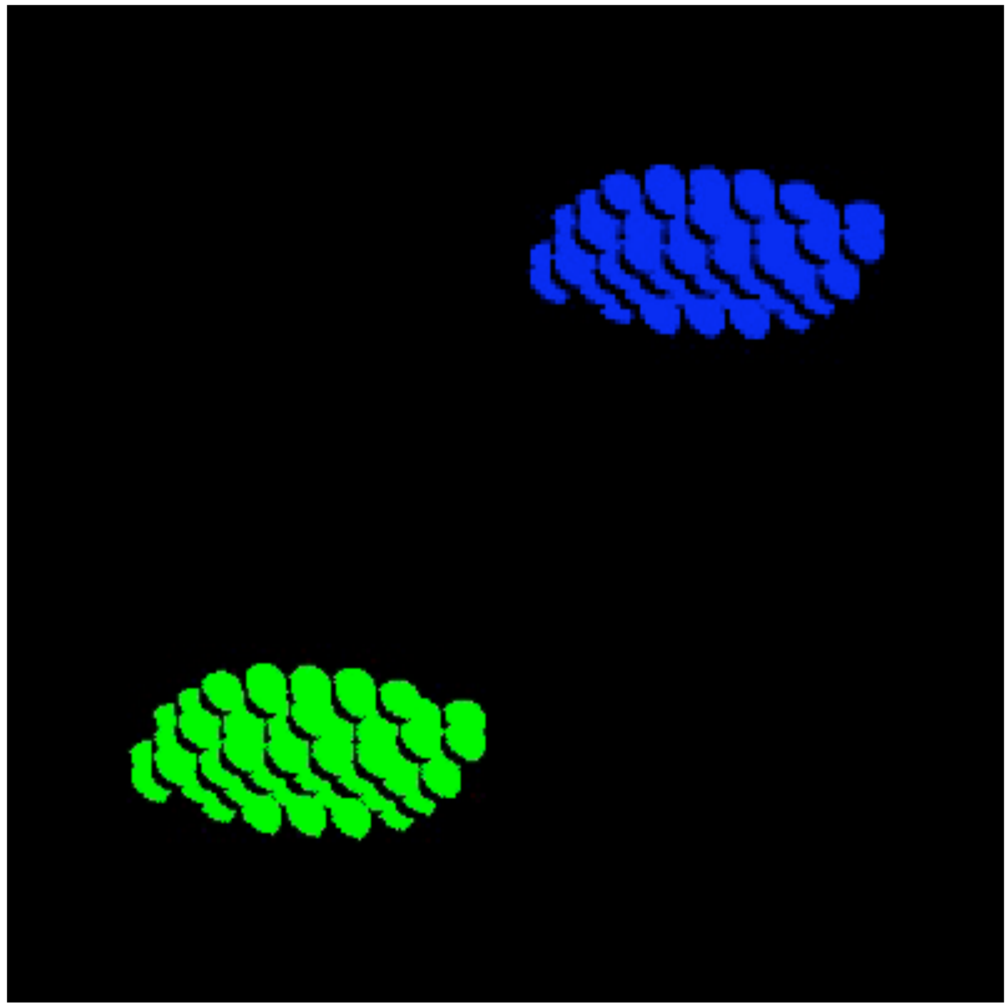
Round-off Error and Overlaps

- Overlap handling strategies:
  - Abort with error (default).
  - Trace particles back in time until touching.
  - Push particles directly away until touching.
  - Merge particles (if merging enabled).
  - Apply repulsive force.
- For single particles, trace-back is best. For rigid bodies, repulsive force is best.

Finally, Rigid Bodies!

- Spheres are a special (easy, ideal) case.
- Perfect spheres are rarely encountered in nature, and may give misleading results when used to model granular flow, aggregation in planetary rings, etc.
- Simplest generalization: allow spheres to stick together in more complex shapes ("bonded aggregates"). Advantages:
  - Can still use tree code for gravity & collisions.
  - Collisions are still sphere point-contact.

Rigid Body Gravity Torques



Euler's Equations of Rigid Body Rotation

$$I_1\dot{\omega}_1 - \omega_2\omega_3(I_2 - I_3) = N_1,$$

$$I_2\dot{\omega}_2 - \omega_3\omega_1(I_3 - I_1) = N_2,$$

$$I_3\dot{\omega}_3 - \omega_1\omega_2(I_1 - I_2) = N_3,$$

where I_i, ω_i are principal moments and body spin components, respectively, and $\mathbf{N}$ is the external torque expressed in the body frame.

Euler's Equations of Rigid Body Rotation

- Previous equations represent a set of coupled ODEs that evolve the spin axis in the body frame. Need 3 more vector equations to evolve body orientation:

$$\dot{\hat{\mathbf{p}}}_1 = \omega_3 \hat{\mathbf{p}}_2 - \omega_2 \hat{\mathbf{p}}_3,$$

$$\dot{\hat{\mathbf{p}}}_2 = \omega_1 \hat{\mathbf{p}}_3 - \omega_3 \hat{\mathbf{p}}_1,$$

$$\dot{\hat{\mathbf{p}}}_3 = \omega_2 \hat{\mathbf{p}}_1 - \omega_1 \hat{\mathbf{p}}_2,$$

where $\hat{p}_i$ are the principal axes of the body.

Euler's Equations of Rigid Body Rotation

- The moments of inertia (eigenvalues) and principal axes (eigenvectors) are found by diagonalizing the inertia tensor—only need to do this when particles added to/removed from aggregate.
- Solve this set of 12 coupled ODEs any way you like (up to next collision, or end of drift). I use a fifth-order adaptive Runge-Kutta (for strongly interactive systems, dissipation not a concern).

For Completeness

- Inertia tensor:

$$\mathbf{I}_{\mathrm{agg}} = \sum_i \left[\mathbf{I}_i + m_i(\boldsymbol{\rho}_i^2 \mathbf{1} - \boldsymbol{\rho}_i \boldsymbol{\rho}_i)\right]$$

  with $\mathbf{I}_i = \frac{2}{5} m_i R_i^2 \mathbf{1}$ and $\boldsymbol{\rho}_i = \mathbf{r}_i - \mathbf{r}_a$

- Torques:

$$\mathbf{N} = \boldsymbol{\Lambda}^{\mathrm{T}} \left[\sum_{i \in a} m_i (\mathbf{r}_i - \mathbf{r}_a) \times (\ddot{\mathbf{r}}_i - \ddot{\mathbf{r}}_a)\right]$$

  where the sum is over all particles in aggregate *a* and $\boldsymbol{\Lambda} \equiv (\hat{\mathbf{p}}_1 | \hat{\mathbf{p}}_2 | \hat{\mathbf{p}}_3)$

Rigid Body Collisions

- Collision resolution complicated because impacts generally off-axis (non-central).
- Solutions do not permit surface friction.
  - However, off-axis collisions cause impulsive torques, allowing transfer of translational motion to rotation, and vice versa.
- Collision prediction also more complicated, due to body rotation.

Collision Prediction & Resolution

$$t = \frac{-(\mathbf{r}\cdot\mathbf{u}) \pm \sqrt{(\mathbf{r}\cdot\mathbf{u})^2 - [r^2 - (s_1+s_2)^2][u^2 + (\mathbf{r}\cdot\mathbf{q})]}}{u^2 + (\mathbf{r}\cdot\mathbf{q})}$$

$$\Delta\mathbf{V}_1 = \gamma(1+\varepsilon_n)(M_2/M)w_n\hat{\mathbf{n}},$$

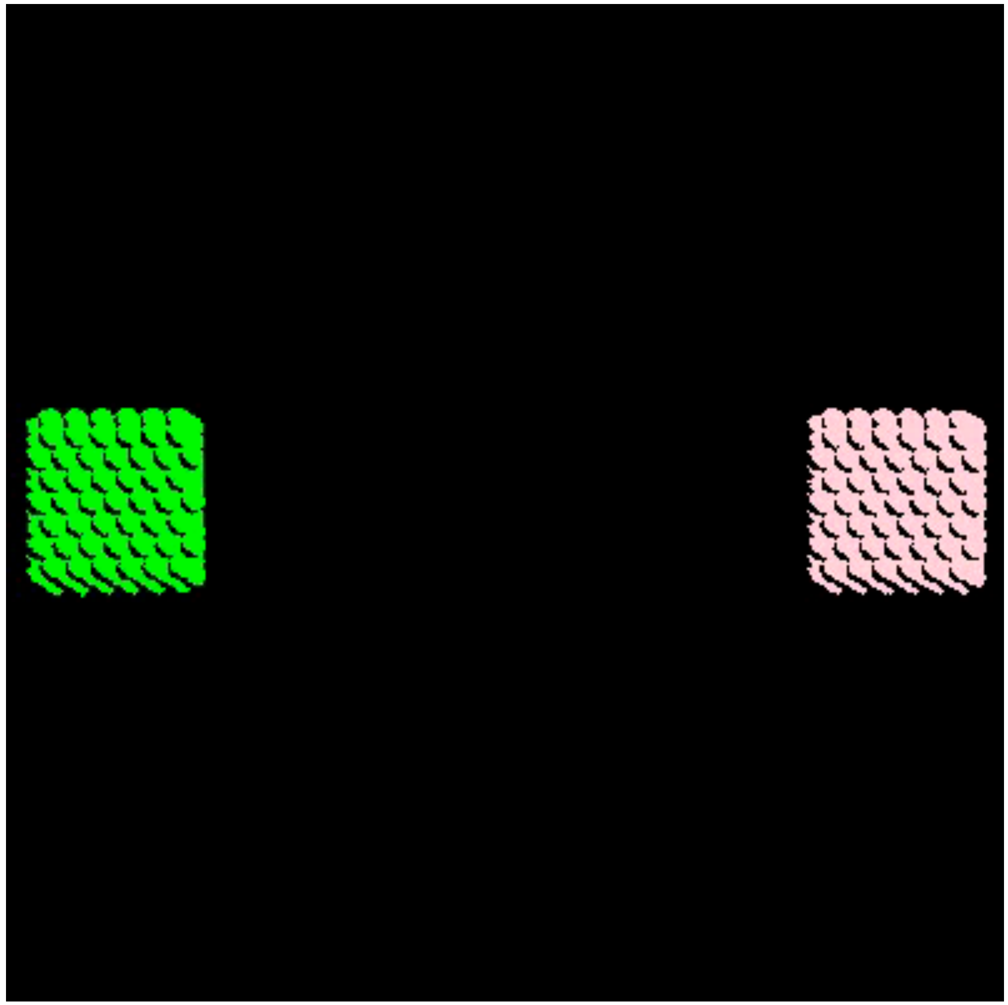
$$\Delta\mathbf{V}_2 = -\gamma(1+\varepsilon_n)(M_1/M)w_n\hat{\mathbf{n}},$$

$$\Delta\mathbf{\Omega}_1 = M_1\mathbf{I}_1^{-1}(\mathbf{c}_1 \times \Delta\mathbf{V}_1),$$

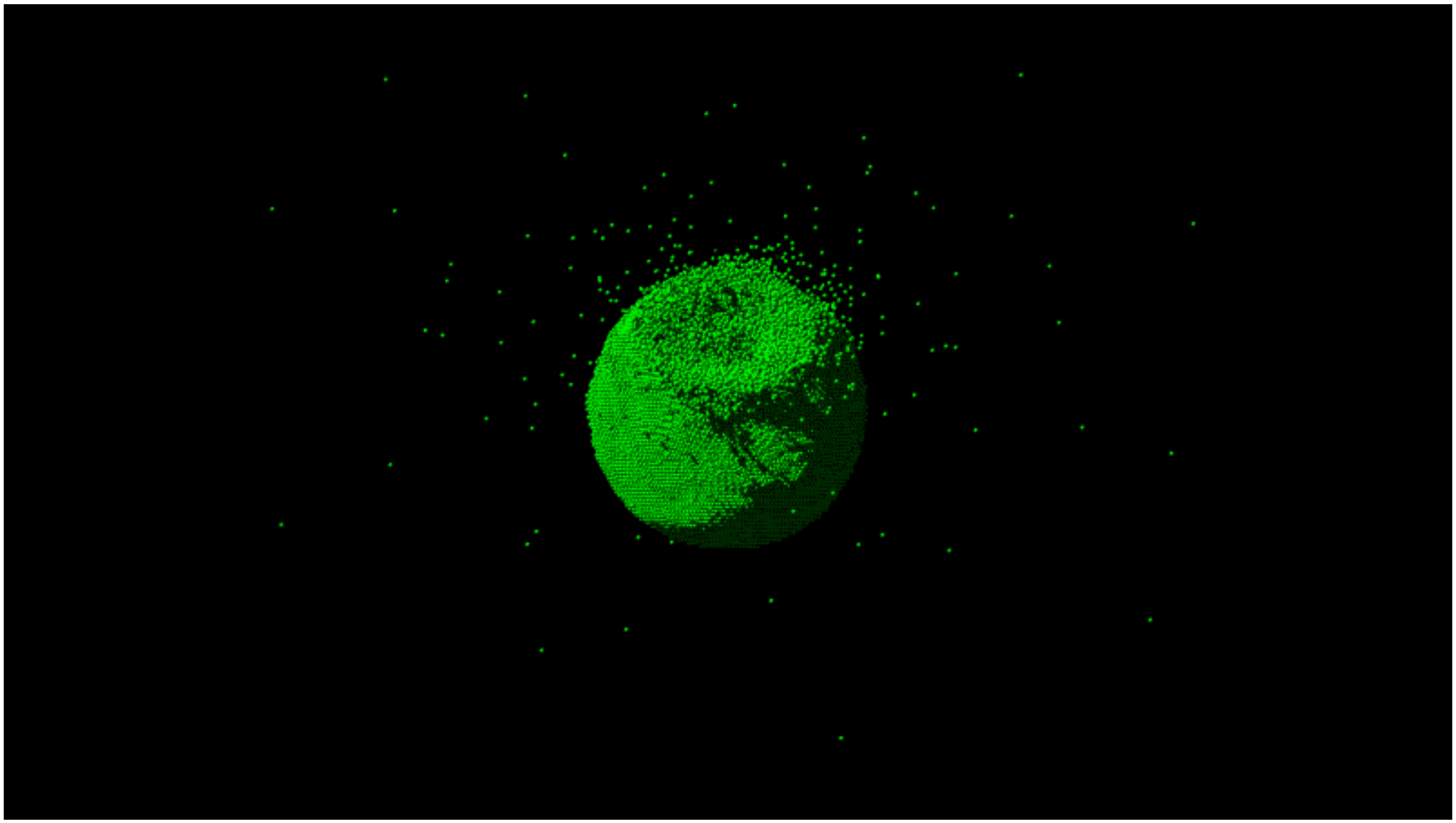
$$\Delta\mathbf{\Omega}_2 = M_2\mathbf{I}_2^{-1}(\mathbf{c}_2 \times \Delta\mathbf{V}_2),$$

See Richardson et al. 2009 for definitions of terms!

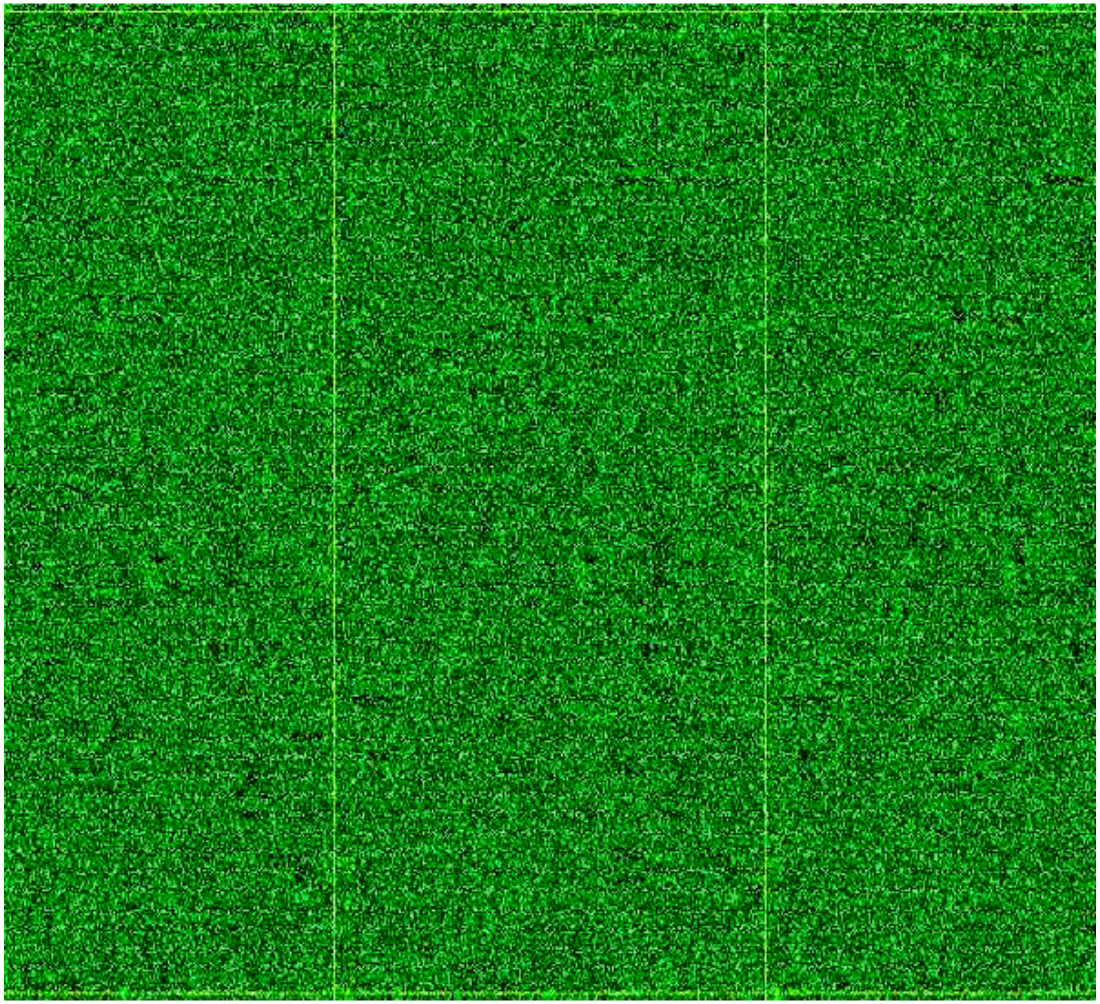
Bouncing Cubes!



Asteroid Family Formation



Bonded Aggregates in Rings



Homework Exercise

- Posted on the PiTP wiki.
- Basic idea: smash stuff up!

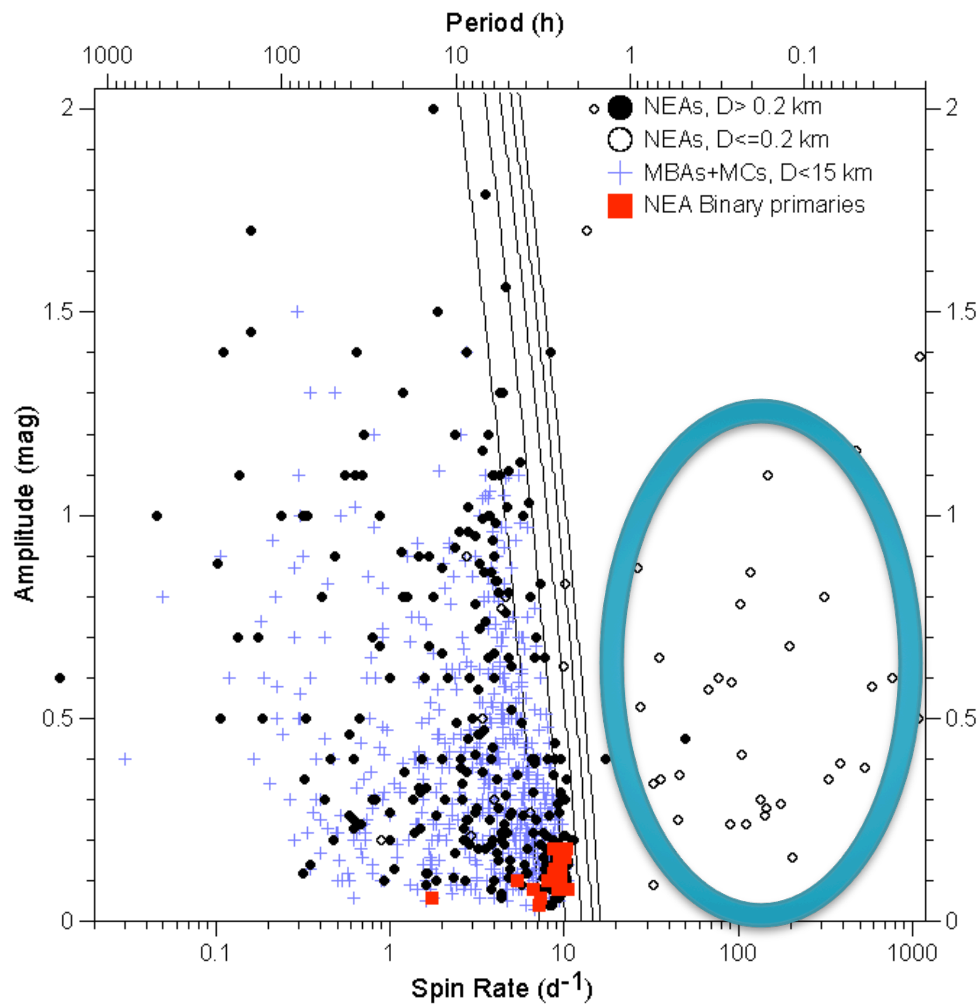
About gravitational aggregates…

- Loose assemblages of coherent pieces held together mostly by gravity.
- May have *some* cohesion between pieces (tensile strength).
- NOTE: under compression, a gravitational aggregate has *shear* strength.
- A rubble pile is a special case of a jumbled body with no cohesion.

What about cohesion?

- Lightcurve and radar data show some very small solar system bodies must have tensile strength/cohesion.

What about cohesion?

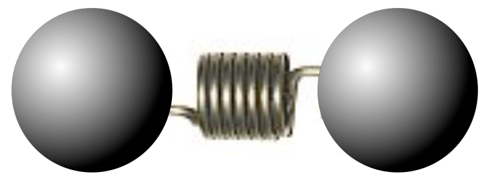


What about cohesion?

- Upper limits from comets SL9 & Tempel 1 ~100 Pa. Essentially no data for asteroids.
- How to model this?
- What is the effect?

Modeling cohesion

- Add simple Hooke's law restoring force between nearby particles.



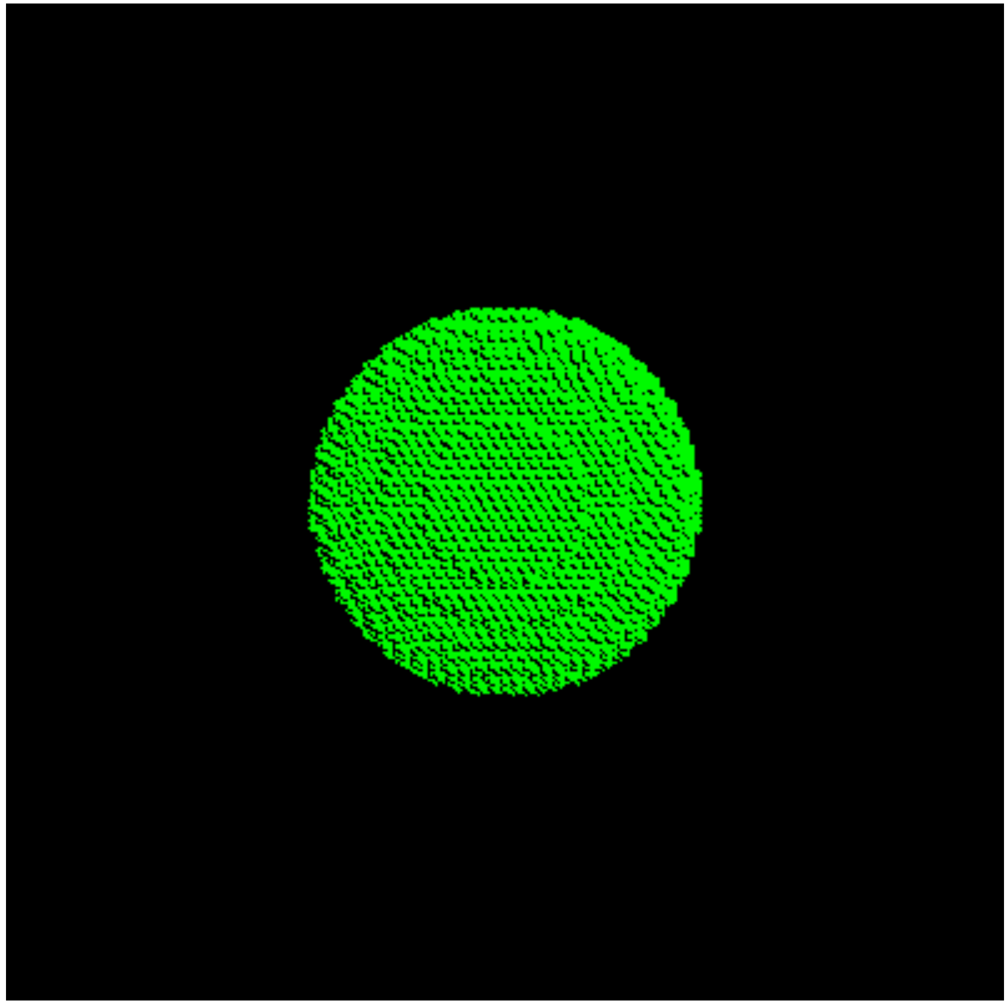
- Deform elastically up to maximum strain (spring rigidity set by Young's modulus).
- Particles act as *tracers* of a continuum solid.

These are NOT bonded aggregates!

Example: excessive initial spin

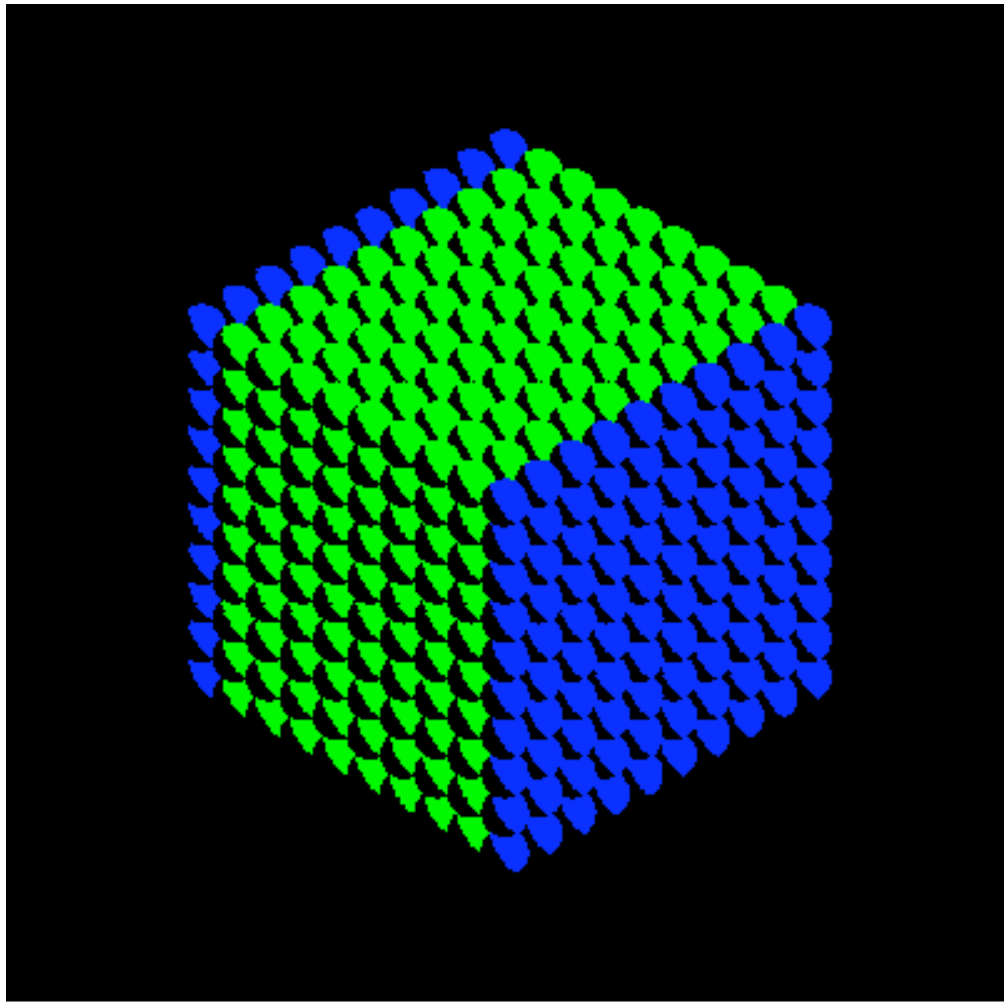
Color legend:

green	3 or more springs
yellow	2 springs only
orange	1 spring only
red	no springs left

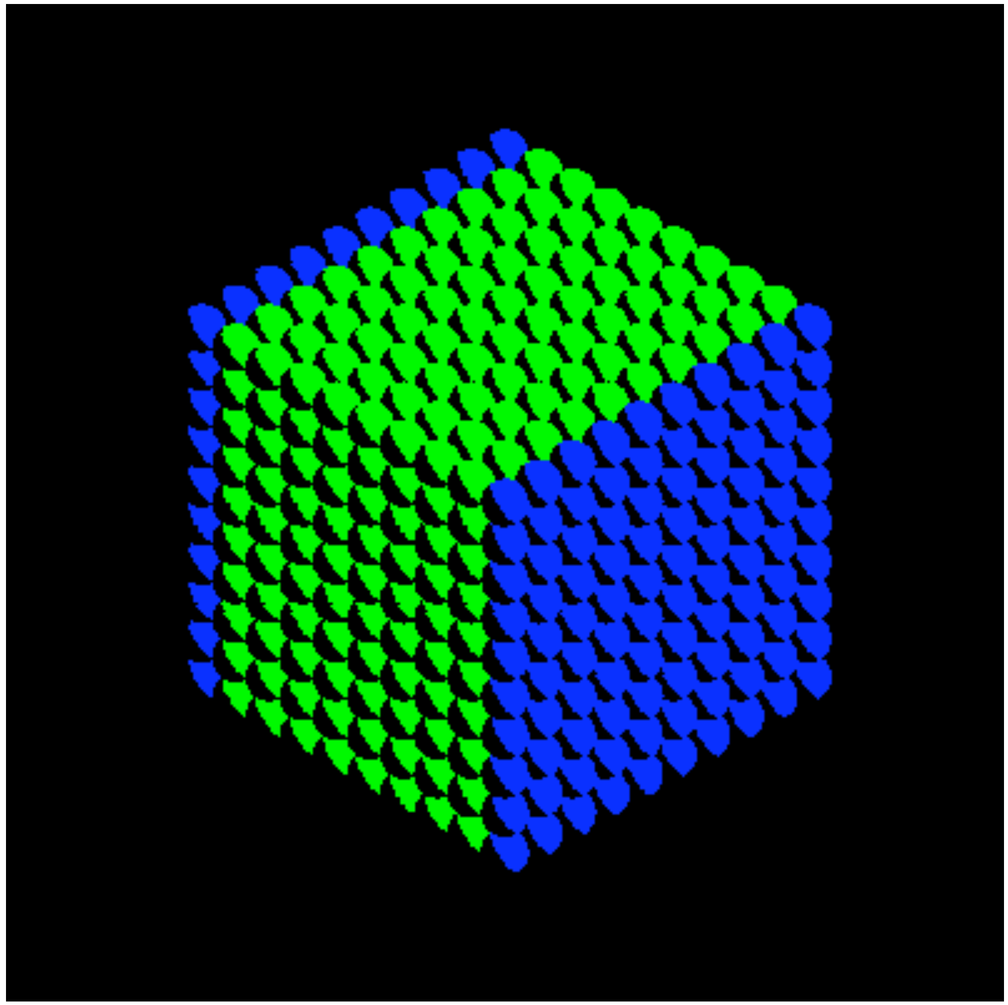


Y = 250 Pa, L = 150 Pa
Spin period P = 0.86 h
Oblate shape α = 0.40

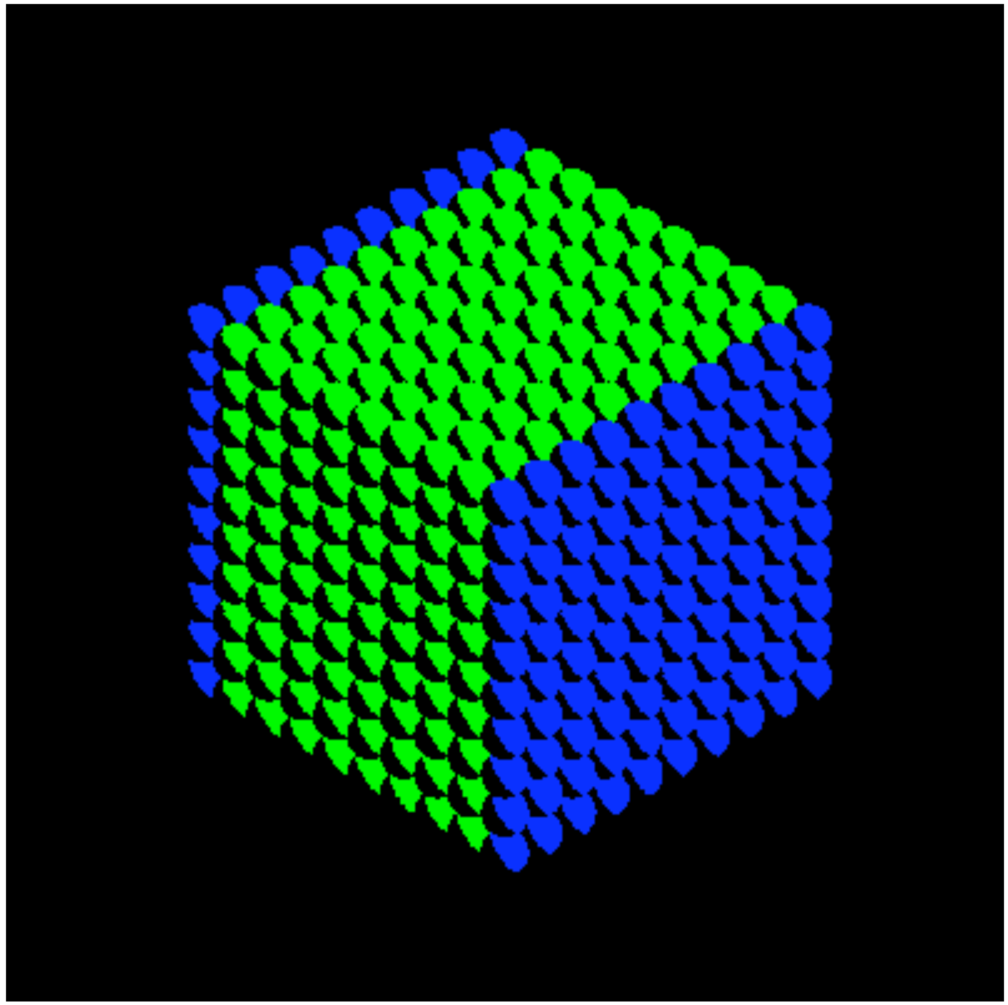
Failure under tension: slow pull



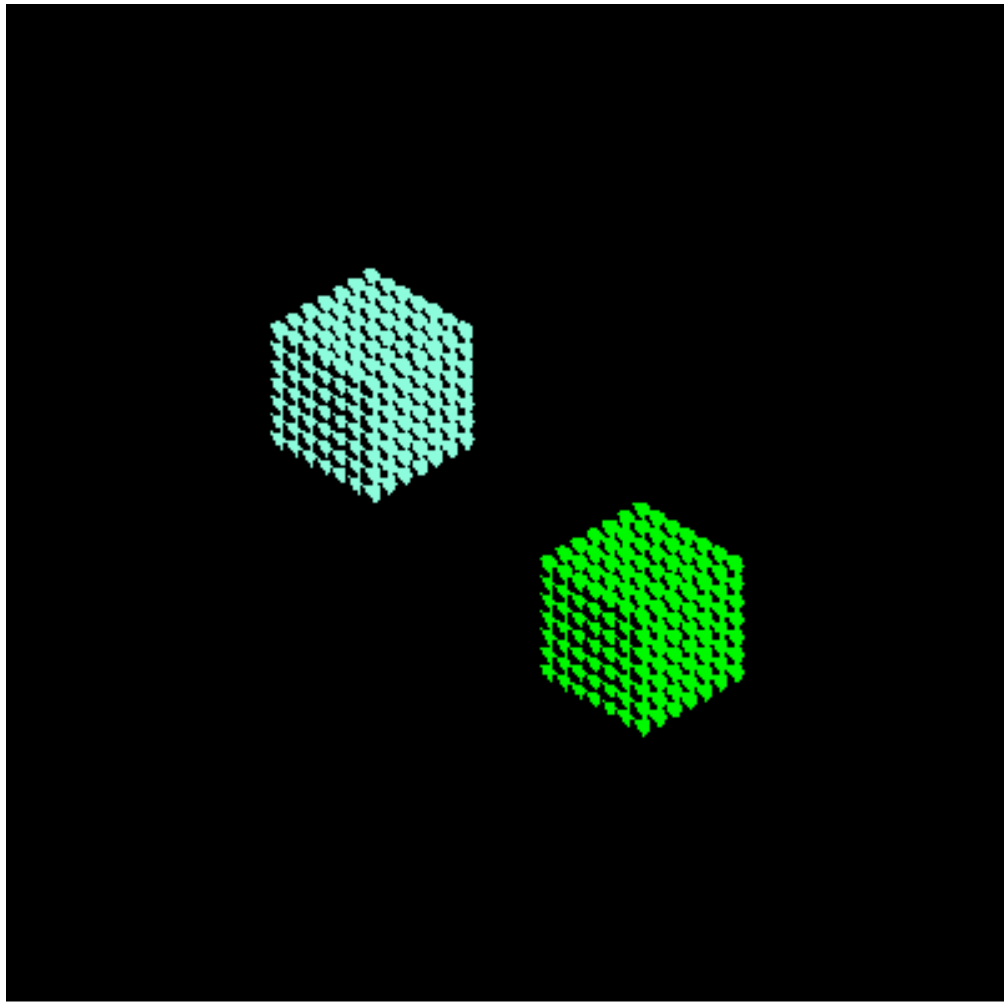
Failure under tension: fast pull



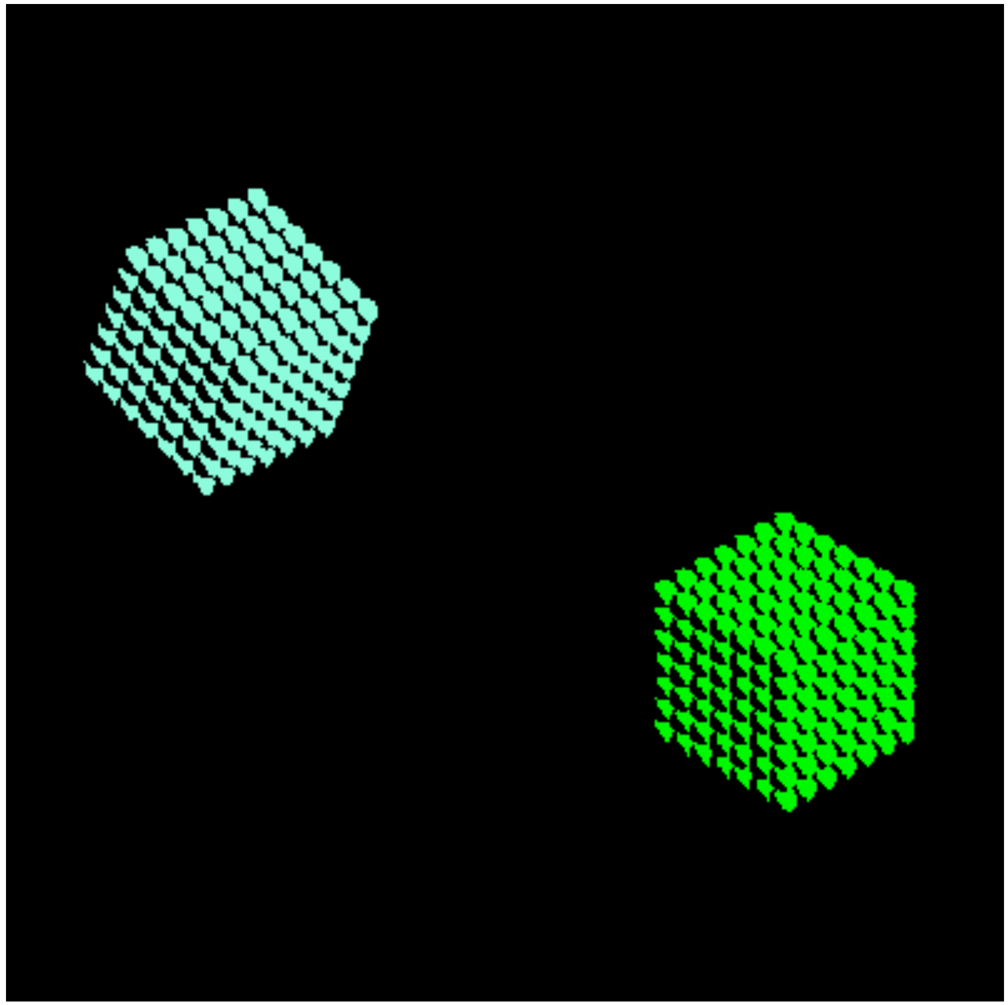
Failure under shear



Colliding cubes



Colliding cubes—faster!



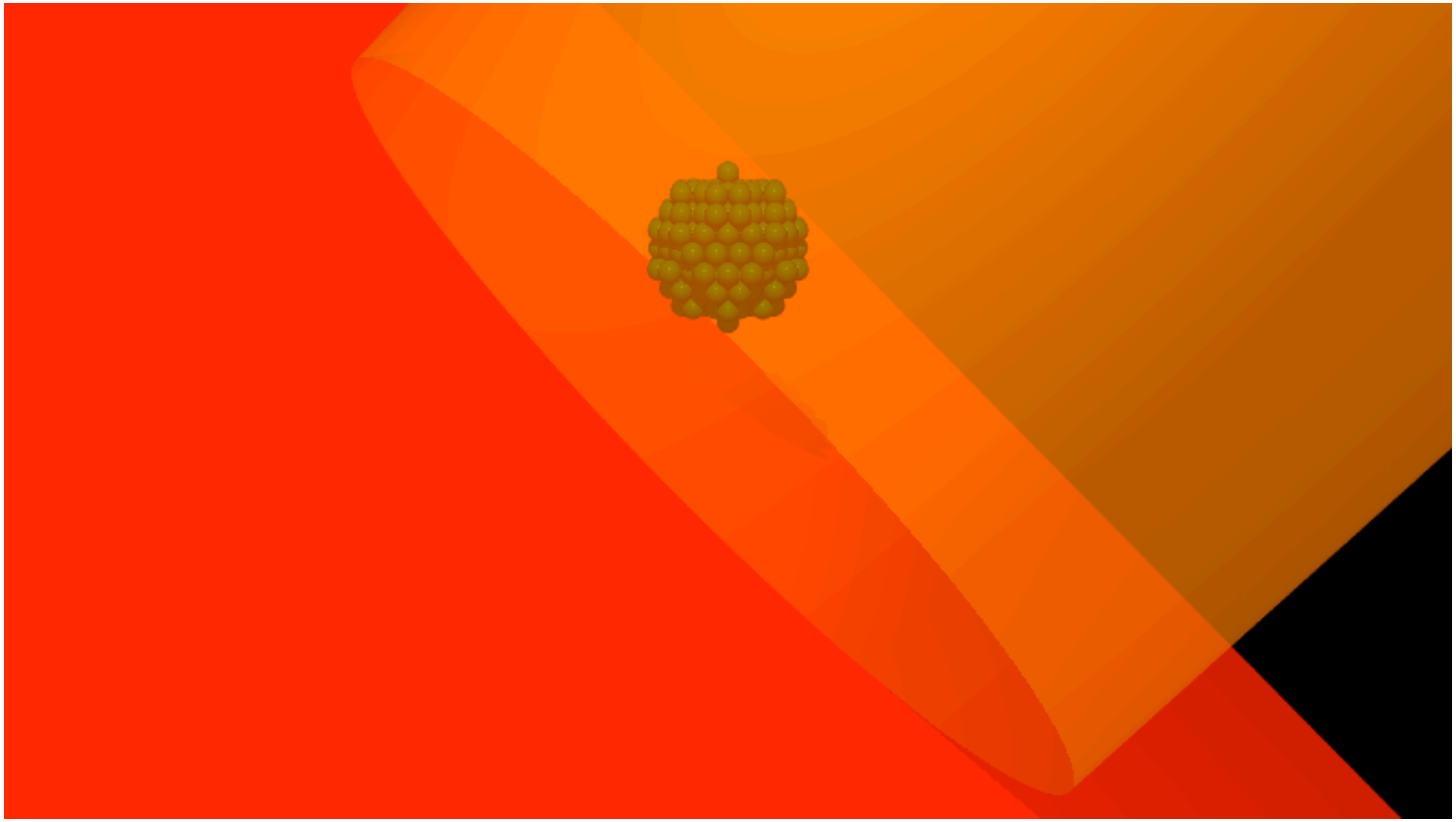
More on Cohesion

- We are applying these models to rotational disruption simulations (binary asteroid formation) and comparing with laboratory experiments.
- Next step: allow for individual spring strengths in order to model pre-existing weaknesses/fractures, e.g. Weibull distribution of flaws.

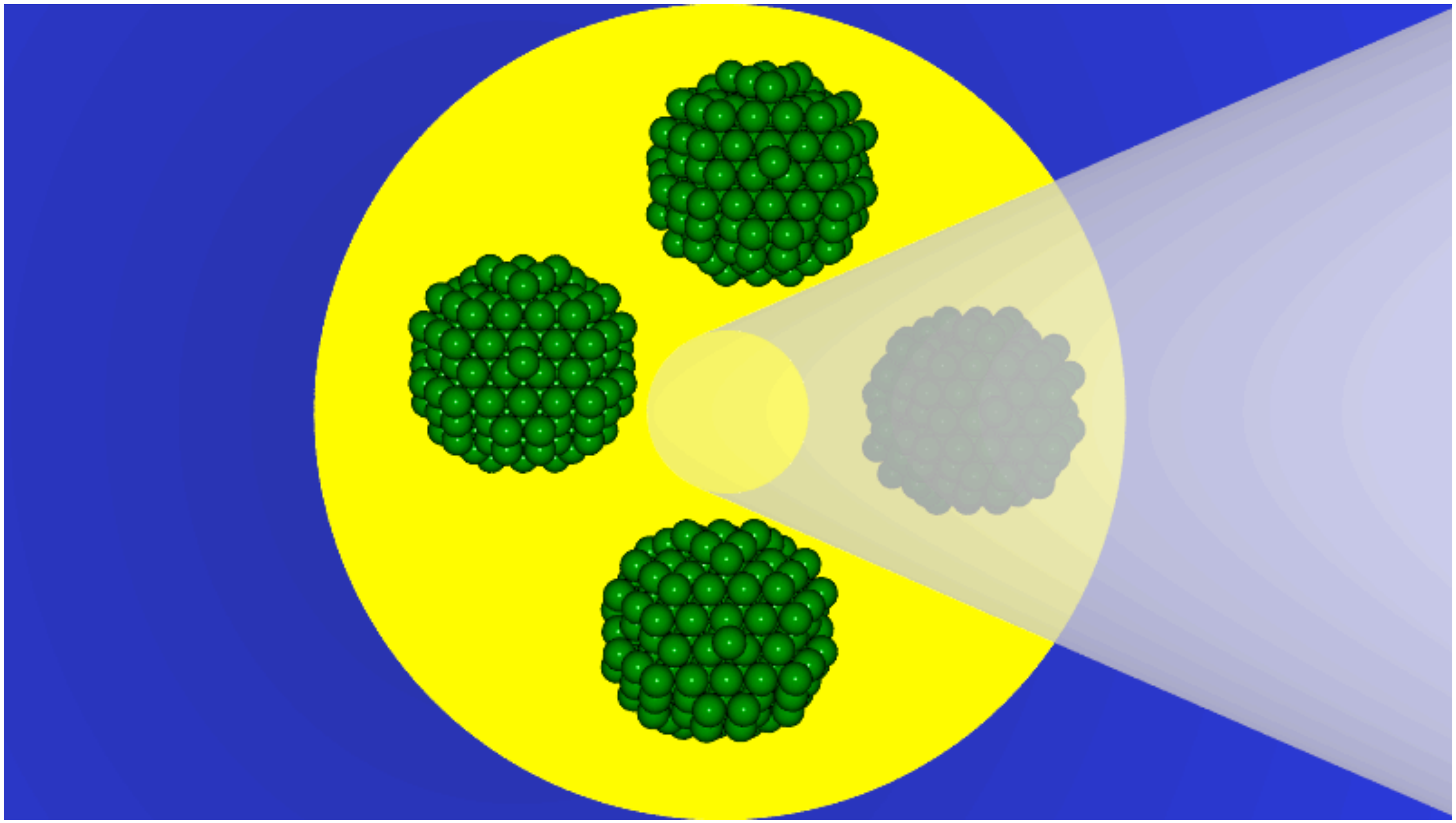
Working with Walls

- Asteroid sample return missions are faced with anticipating the behavior of granular material in very weak gravity.
- Want to develop simulations of these regimes, but be able to compare with physical experiments.
- Approach: provide wall "primitives" that can be combined to replicate experimental apparatus.

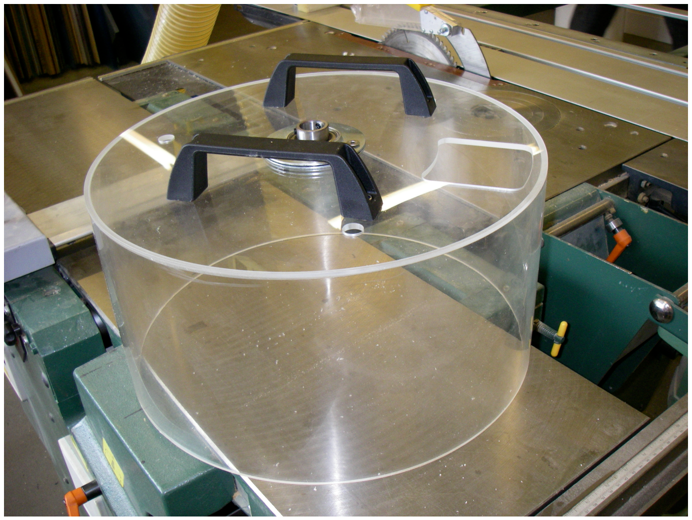
Particles in an Inclined Cylinder



Taylor-Couette Shear Cell



Taylor-Couette Shear Cell



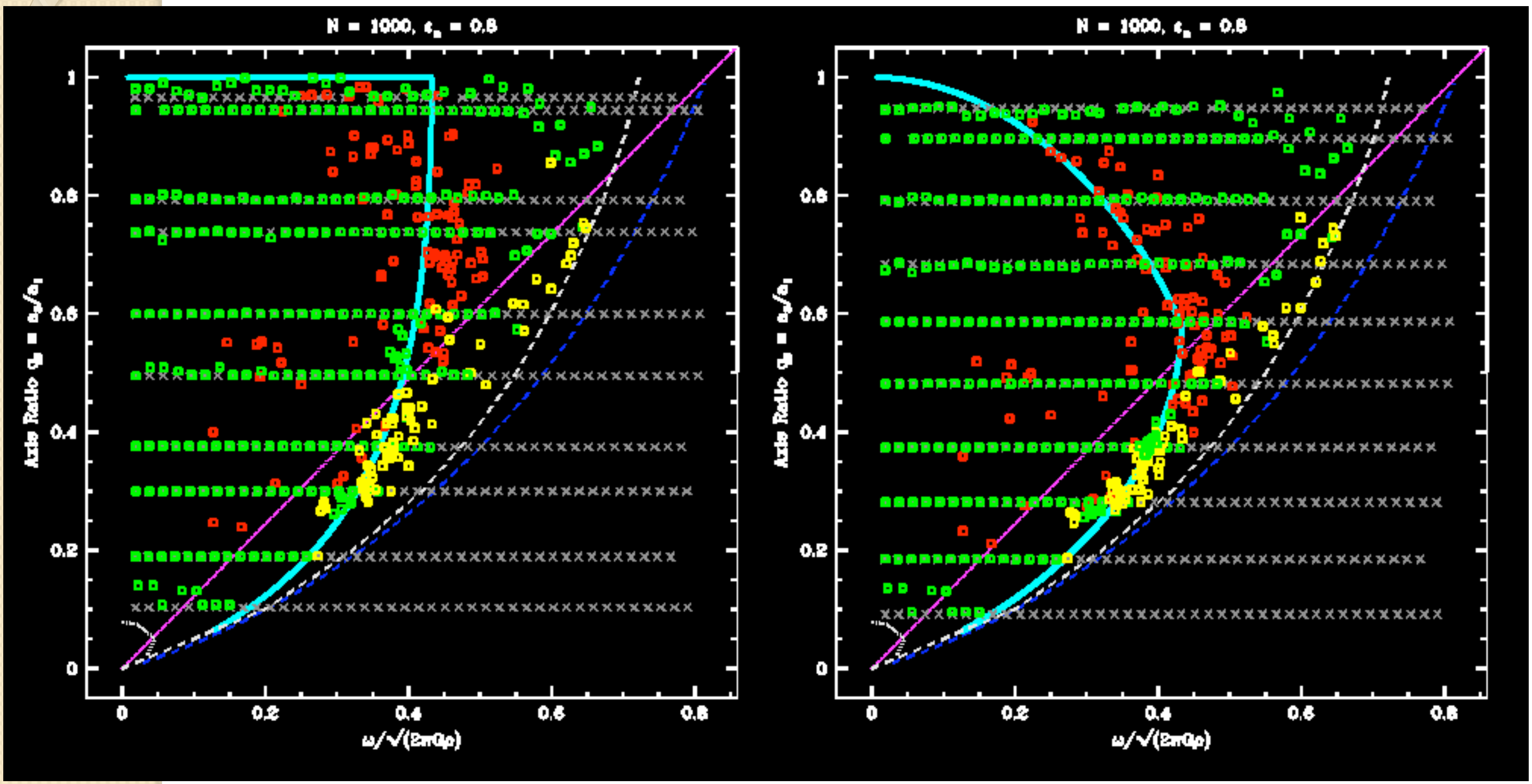
Naomi Murdoch

Summary

- Physical collisions in *N*-body codes enabled by neighbor finding and solving collision equations.
- Rigid body mechanics additionally require solving Euler equations and more complex collision prediction and resolution.
- Many applications, ranging from planet formation to granular dynamics.

Extra Slides

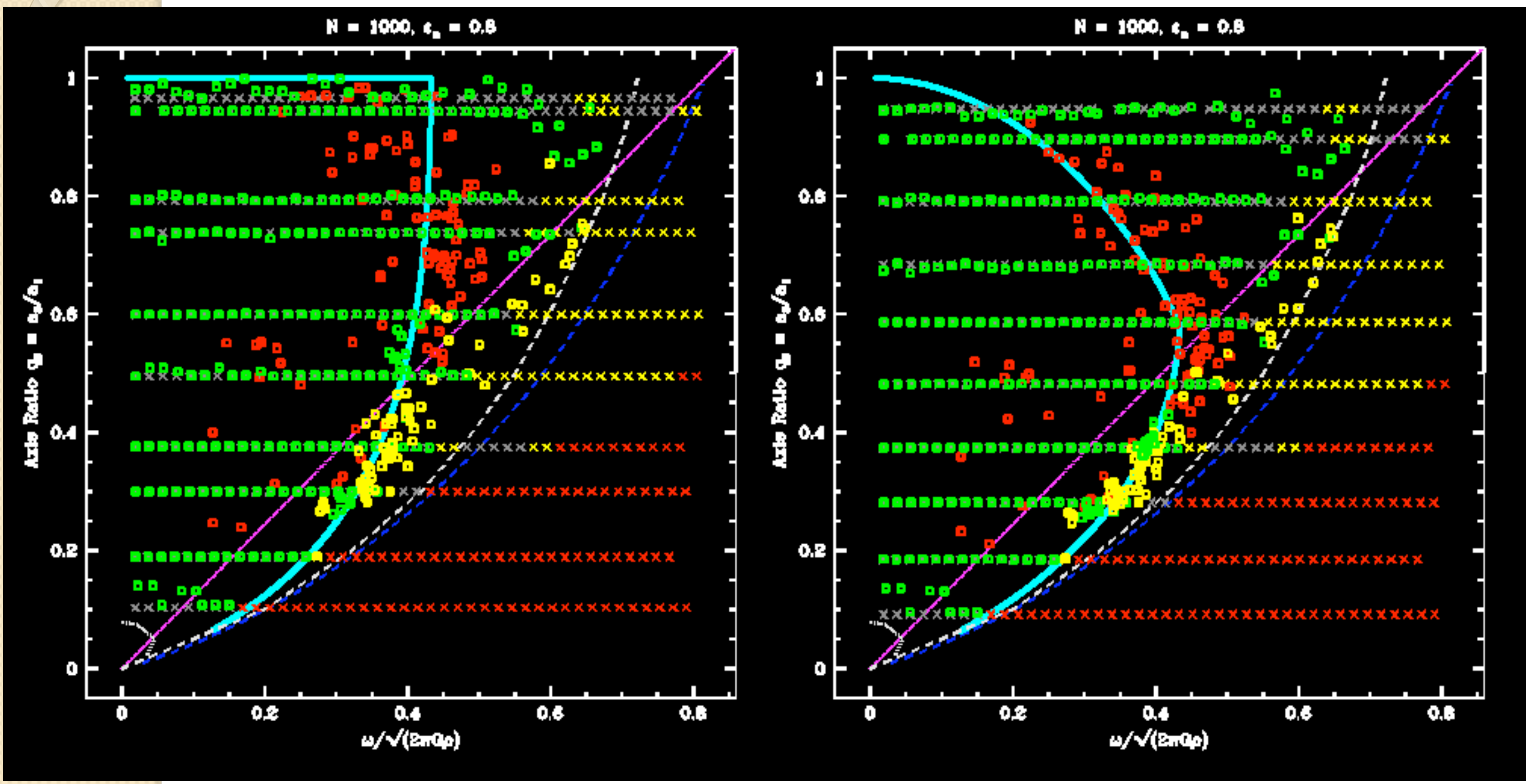
Rubble Pile Equilibrium Shapes



Mass loss: 0% < 10% > 10%   X = initial condition

Richardson et al. "Modeling Cohesion in Gravitational Aggregates" (DPS '08 #55.02)

Rubble Pile Equilibrium Shapes



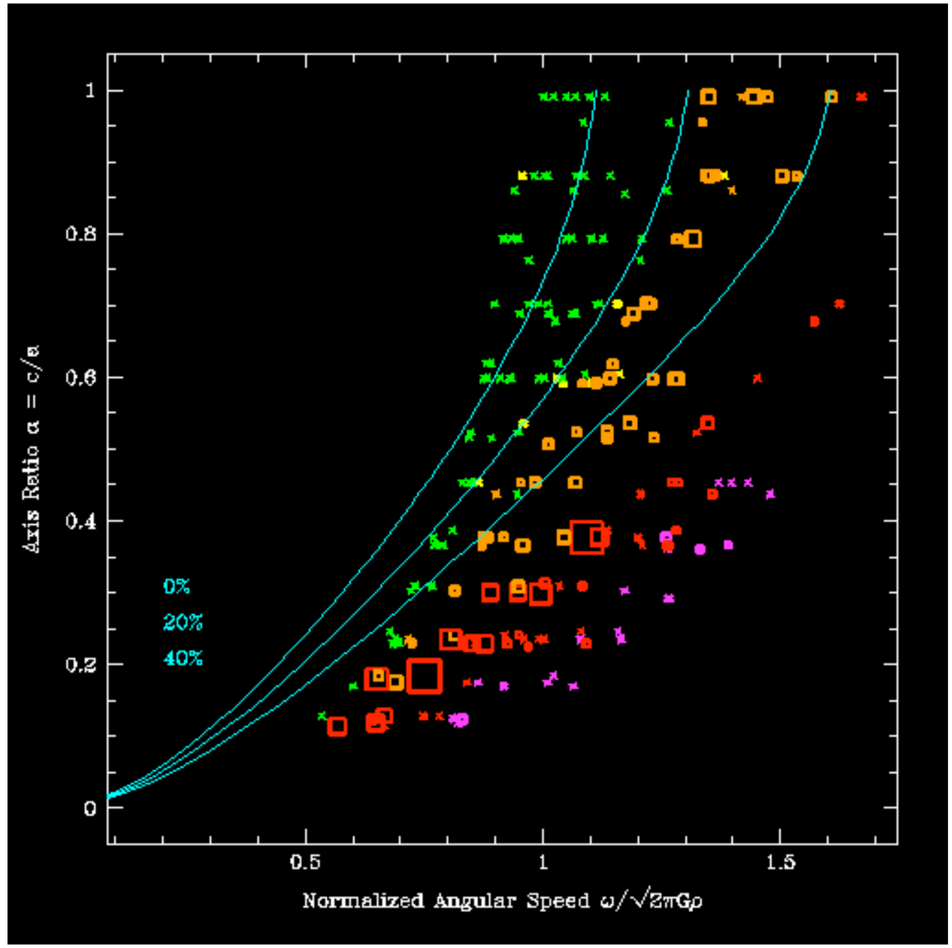
Mass loss: 0% < 10% > 10% X = initial condition

Richardson et al. "Modeling Cohesion in Gravitational Aggregates" (DPS '08 #55.02)

Oblate, *Y*=250, *L*=150 Pa

Color legend:

green	no mass loss
yellow	< 10% mass loss
orange	< 50% mass loss
red	< 90% mass loss
fuchsia	≥ 90% mass loss



Symbol legend:

- × remnant only
- □ mass in orbit
- ★ accreting mass

(symbol size proportional to mass orbiting/accreting)

Richardson et al. "Modeling Cohesion in Gravitational Aggregates" (DPS '08 #55.02)

Damping Oscillations