NcorpiON, A $\mathcal{O}(N)$ N-body integrator for collisional and fragmenting systems.

Jérémy COUTURIER, Alice Quillen, Miki Nakajima

University of Rochester

Features

- Written in C language.
- Entirely open source.
- A python add-on to produce animations.
- Four built-in modules for mutual interactions :

1. Brute-force $\mathcal{O}(N^2)$ method
2. Barnes-Hut $\mathcal{O}(N \ln N)$ tree code
3. Mesh-grid $\mathcal{O}(N)$ method
4. FalcON $\mathcal{O}(N)$ algorithm

About NcorpiON

NcorpiON is a N-body software developed for the time-efficient integration of collisional and fragmenting systems. It features a fragmentation model able to realistically simulate a violent impact.
Previous N-body integrators did not implement the possibily that, upon a violent collision, particles may fragment instead of just bouncing or merging. NcorpiON solves this issue with a built-in fragmentation model that relies on crater scaling and ejecta models to implement realistic fragmentations.
The most difficult challenge in a N-body code with a large number of particles is to time-efficiently treat mutual interactions between them (collisions and gravity). NcorpiON comes with four different built-in modules for mutual interactions computation, one of them being the fast multipole method Falcon of Walter Dehnen[2], that NcorpiON adapts so it can detect collisions as well as compute mutual gravity.

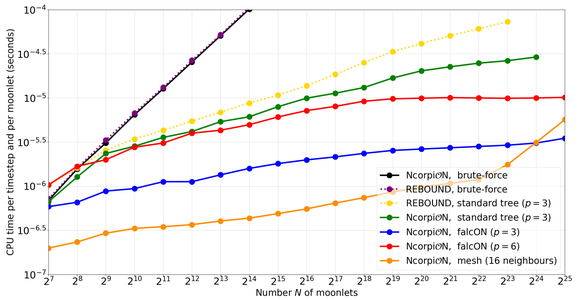

Figure 1:Performances of NcorpiON for different modules.

The fragmentation model

I used the litterature on crater scaling for the fragmentation of NcorpiON (*e. g.* [1]). The distribution of speeds in the debris is given by

$$\frac{M^\star(v)}{m_1} = \frac{3k}{4\pi}\left(\frac{C_1 \Delta v \cos\theta}{v}\right)^{3\mu}, \qquad (1)$$

where m_1 is the impactor's mass and $M^\star(v)$ is the mass of debris with velocity higher than v.

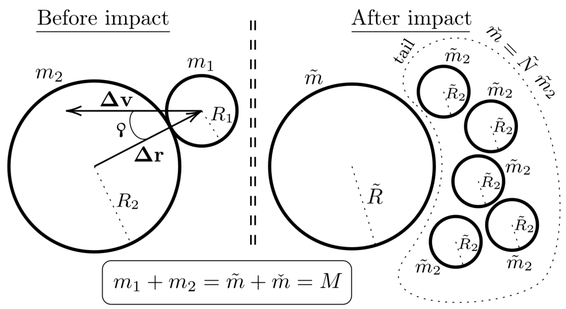

Figure 2:Outcome of a fragmentation in NcorpiON

Website

You can visit NcorpiON's website at https://ncorpion.com and download the software at https://github.com/Jeremycouturier/NcorpiON. You can contribute too !

FalcON algorithm

FalcON is by far the best of NcorpiON's modules for gravity computation. It groups particles in cell in an octree and relies on fast multipole expansion in order to compute mutual gravity in time $\mathcal{O}(N)$ instead of $\mathcal{O}(N^2)$ for a brute-force direct computation. FalcON approximates the gravitational potential

$$\phi(\boldsymbol{x}) = \sum_{i \in B} \mu_i \, g(\boldsymbol{x} - \boldsymbol{x}_i), \qquad (2)$$

with a multipole expansion of the form

$$\phi(\boldsymbol{x}) = \sum_{m=0}^{p} \frac{1}{m!} (\boldsymbol{x} - \boldsymbol{s}_A)^{(m)} \odot \boldsymbol{C}^{(m)}(\boldsymbol{s}_A), \qquad (3)$$

where $\boldsymbol{M}_B^{(n)}$ is the cell's multipole and

$$\boldsymbol{C}^{(m)}(\boldsymbol{s}_A) = \sum_{n=0}^{p-m} \frac{(-1)^n}{n!} \boldsymbol{\nabla}^{(n+m)} g(\boldsymbol{R}) \odot \boldsymbol{M}_B^{(n)}(\boldsymbol{s}_B).$$

Octree

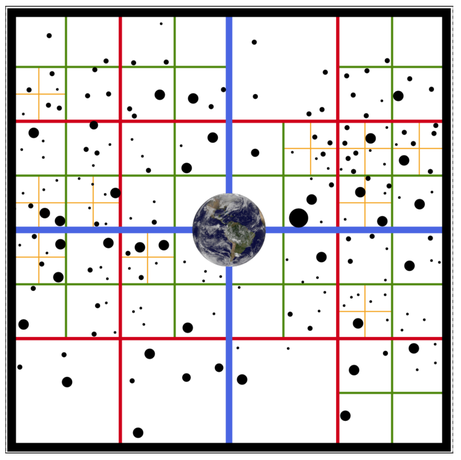

Figure 3:The octree used by falcON to group particles.

Implementation

The user of NcorpiON can choose between four different built-in modules to compute self-gravity and detect collisions. One of these is a $\mathcal{O}(N)$ tree-based algorithm called FalcON. It relies on fast multipole expansion for gravity computation and we adapted it to collision detection as well. Computation time is reduced by building the tree structure using a three-dimensional Hilbert curve. For the same precision in mutual gravity computation, NcorpiON is found to be up to 25 times faster than the famous software REBOUND.
NcorpiON detects collisions and computes mutual gravity faster than REBOUND, and unlike other N-body integrators, it can resolve a collision by fragmentation. The fast mutipole expansions are implemented up to order $p = 6$ in Eq. (3) to allow for a high precision in mutual gravity computation.

Conclusion

NcorpiON is the first $\mathcal{O}(N)$ N-body software to be able to resolve collisions by fragmentation. We are currently using it to gain insight on the formation of the Moon from the protolunar disk that followed the giant impact with Theia. Besides looking like Scorpion, I chose this name due to the time complexity being in $\mathcal{O}(N)$ and because N-body translates to N-corps in French.

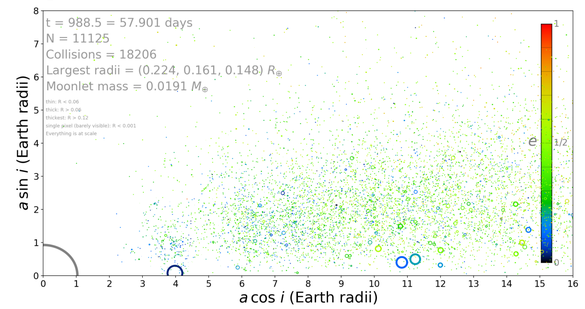

Figure 4:Snapshot of a Moon-forming simulation

References

[1] Kevin R. Housen and Keith A. Holsapple.
Ejecta from impact craters.
Icarus, 211:856–875, January 2011.

[2] Walter Dehnen.
A Hierarchical O(N) Force Calculation Algorithm.
Journal of Computational Physics, 179:27–42, June 2002.

Acknowledgements

Jérémy Couturier thanks Walter Dehnen for helpful transatlantic discussions about the intricacies of FalcON algorithm as well as Hanno Rein for help with REBOUND.

Contact Information

- Web: https://jeremycouturier.com
- Email: jeremycouturier21@hotmail.fr
- Phone: +33 601 461 030