Available online at www.sciencedirect.com
SCIENCE DIRECT®

Icarus 176 (2005) 432–439

ICARUS

www.elsevier.com/locate/icarus

A fast method for finding bound systems in numerical simulations: Results from the formation of asteroid binaries

Zoë M. Leinhardt *, Derek C. Richardson

Department of Astronomy, University of Maryland, College Park, MD 20740-2421, USA

Received 24 August 2004; revised 3 November 2004

Available online 21 April 2005

Abstract

We present a new code (`companion`) that identifies bound systems of particles in $\mathcal{O}(N \log N)$ time. Simple binaries consisting of pairs of mutually bound particles and complex hierarchies consisting of collections of mutually bound particles are identifiable with this code. In comparison, brute force binary search methods scale as $\mathcal{O}(N^2)$ while full hierarchy searches can be as expensive as $\mathcal{O}(N^3)$, making analysis highly inefficient for multiple data sets with $N \gtrsim 10^3$. A simple test case is provided to illustrate the method. Timing tests demonstrating $\mathcal{O}(N \log N)$ scaling with the new code on real data are presented. We apply our method to data from asteroid satellite simulations [Durda et al., 2004. Icarus 167, 382–396; Erratum: Icarus 170, 242; reprinted article: Icarus 170, 243–257] and note interesting multi-particle configurations. The code is available at http://www.astro.umd.edu/zoe/companion/ and is distributed under the terms and conditions of the GNU Public License.

Keywords: Asteroids; Dynamics; Data analysis techniques

1. Introduction

1.1. Binaries in the Solar System

Recent technical advances in observational techniques, specifically radar and adaptive optics (Merline, 2001), have resulted in the detection of dozens of binaries among the Near-Earth Asteroid (NEA), Main Belt Asteroid (MBA), and Jupiter Trojan populations. Detailed lightcurve analysis (Pravec et al., 2000, 2002) and even a spacecraft flyby (Belton and Carlson, 1994; Belton et al., 1995) have also revealed binaries among asteroids. Binaries also exist in the trans-neptunian region (Margot, 2002; Pluto and Charon represent the most extreme example). Binary asteroids appear to represent a significant fraction of the asteroid population (10–20%) (Merline, 2001). Given the relatively short lifetimes of MBAs and NEAs binaries (Bottke, personal communication, Chauvineau and Farinella, 1995), the Solar System is evidently still dynamically active, continuously forming new binaries.

1.2. Numerical simulations of binary formation

The diverse physical and dynamical properties of binary asteroids suggest at least three distinct formation mechanisms: (1) NEA binaries may have been formed by tidal disruption during close planetary encounters (Richardson, 2001; Walsh et al., in preparation) or by fission following thermal spin-up (Margot et al., 2002); (2) MBA binaries may result from highly energetic collisions between asteroids, including family forming events (e.g., Michel et al., 2001; Durda et al., 2004); and (3) Kuiper belt binaries, given their large separations, may have formed through three-body encounters or capture following energy loss via dynamical friction from small bodies (Goldreich et al., 2002; Weidenschilling, 2002).

Simulations of MBA binary formation (e.g., Michel et al., 2001; Durda et al., 2004) are suitable for modest computer

* Corresponding author.
E-mail addresses: zoe@astro.umd.edu, zoe@eps.harvard.edu (Z.M. Leinhardt).

0019-1035/$ – see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.icarus.2004.11.008

clusters, employing $N \sim 10^5$ particles with a two-phase numerical method. In the first phase, the physical collision and resulting fracture propagation is modeled with smoothed particle hydrodynamics (SPH) code (Benz and Asphaug, 1999). In the second phase, after the collisional shock has propagated through the bodies, the simulation is switched to an N-body code (Richardson et al., 2000) which follows the debris for timescales of days under the mutual effects of gravity. Typical projectile and target asteroids are between 1 and 100 km in size, with impact speeds of kilometers per second. All simulations of this type require $N \sim 10^5$ in order to accurately model the collisional shock wave. Both groups found that binary asteroids formed as the result of catastrophic collision. In addition, (Richardson, 2001) showed that NEA binaries could be formed via tidal disruption of a "rubble pile" and (Walsh et al., in preparation) have begun a systematic study of binary asteroid formation via tidal disruption.

These simulations raise an interesting problem for data analysis. In order to understand the formation of binary asteroids fully, a fast, complete search method is needed that can identify both simple binary and hierarchical systems for $N \gg 10^3$. Once binaries and/or systems have been identified, their properties can be measured and compared to observed populations (with some assumptions on long-term stability). A brute-force search would require $\mathcal{O}(N^2)$ comparisons if each particle is compared with every other particle. A more complete and complex search would naively require $\mathcal{O}(N^3)$ comparisons if in addition every particle is compared with every system. Both searches are prohibitive for large N ($\gtrsim 10^4$; less if multiple data sets or time series are considered).

1.3. Previous work on binary detection in numerical simulations

The problem of developing an efficient method for finding bound groups of asteroids is related to searching for groups of galaxies in cosmological N-body simulations that contain large numbers of particles. In this case a nearest neighbor algorithm called "friend-of-friends" (FoF) (Davis et al., 1985) is often employed. FoF relies on a linking length test of a particle's nearest neighbors in order to determine what particles should be considered members of the group. For example, if particle B is within one linking length of particle A, particles A and B are in the same group. If particle C is within one linking length of particle B, particles A, B and C are in the same group, and so on. SKID (Governato et al., 1997) and the hierarchical clustering method (Zappalà et al., 1990) are more complex algorithms, but the group search is done in the same way.

We have developed our method along the lines of cosmological search methods, which are quite efficient. Because we are not specifically interested in spatial groups, we have replaced the linking length test with an escape speed test. The relative speed of a particle and its possible companion is compared to their mutual escape speed to see if the pair is bound (in the absence of all other perturbations). To improve efficiency, we employ a Barnes and Hut (1986) hierarchical tree to limit the search for possible companions to those that are nearby (in the sense of being contained in a tree cell with a sufficiently large opening angle; cf. Section 2.1) or to those contained in a small or distant tree cell whose center-of-mass speed fails the escape speed test. It is possible that a small fraction of binaries may be missed with this method (see Section 3.2 for a discussion; in particular note that our tests show $\gtrsim$99% completeness in most cases, and it is always possible to set the tree criterion so low that all pairs are considered, at the expense of computation time).

In this paper we present `companion`, a hierarchical tree code that detects binaries, multiple, and complex hierarchical systems in the output from numerical simulations. Section 2 describes the numerical method in detail. Section 3 presents diagnostic and performance tests. In Section 4 we present analysis of published data from (Durda et al., 2004), highlighting newly detected hierarchical systems. A summary and conclusions are given in Section 5.

2. Numerical method

In general, the most stable binaries are those that are the most tightly bound. This means that for a particle of a given mass the likelihood of having stable satellites decreases with increasing distance and relative speed (between the particle and potential satellite). The maximum distance at which a satellite can be bound to a particle depends on the combined mass of the system. As a result, `companion` uses a 3-D spatial tree code (Barnes and Hut, 1986), augmented by a center of mass relative speed test to insure that widely separated systems are found (cf. Section 2.2).

2.1. Hierarchical spatial tree

Our tree construction method closely follows the algorithm of Barnes and Hut (1986). Particles are placed one at a time, according to their spatial coordinates, inside the "root cell," a cubical volume large enough to contain the entire system. Any time two particles end up in the same cell, the cell is divided in half along each coordinate axis, resulting in 8 daughter cells. The two particles in question are then placed into the respective daughter cell appropriate to their spatial coordinates. If they still share the same cell, the daughter cell is itself subdivided, and the process is repeated. The entire procedure continues recursively for all particles, until every particle resides in its own unique cell. At this point the entire tree has been built from the bottom up. Accessing any given particle requires "walking" the tree, beginning at the root cell, opening every cell that contains the particle of interest, and ending when the cell containing the particle has been reached. Figure 1 shows an example of a simple two-dimensional spatial tree with three particles.

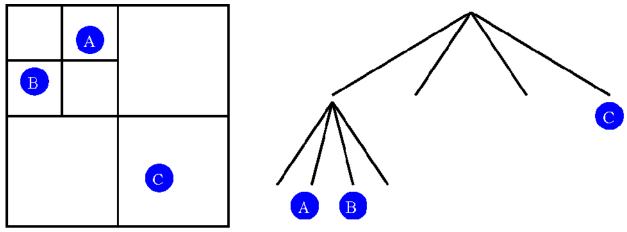

Fig. 1. An example of a simple two-dimensional spatial tree. On the right is a depiction of division of the root cell of the tree after three particles (A, B, and C) have been placed into the tree. On the left is a "tree" diagram that describes the level of each particle in the tree. Starting at the top is the root cell. The first level below the root cell contains one cell with a particle C, two empty cells and a cell that has four daughter cells. At the second level below the root cell there are two cells each with one particle each (A and B) and two empty cells.

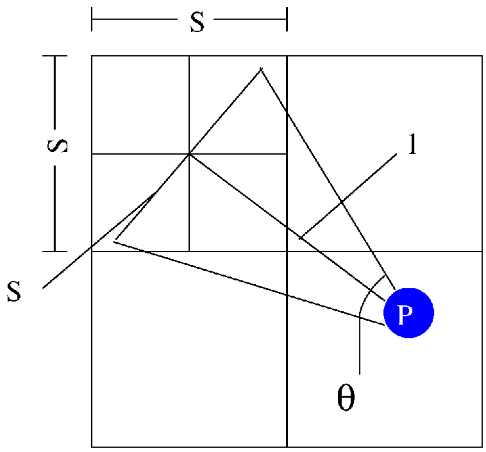

Fig. 2. A graphical depiction of the opening angle test for a particle P, where $\theta = s/l$, s is the length of one side of the cell being tested, l is the distance between particle P and the center of the cell. The cell in question will be opened if $\theta > \theta_{\mathrm{crit}}$.

The premise of a spatial tree code is that particles far from a given particle of interest (called particle P from now on) are generally not as important as those that are nearby. As a result, only particles that exert the most influence on P are considered in detail. In this case, such particles are those residing in cells that open an angle $\theta > \theta_{\mathrm{crit}}$ with respect to P, where $\theta = s/l$, s is the length of one side of the cell being tested, l is the distance between P and the center of the cell,[1] and θ_{crit} is the "critical opening angle" (in radians), specified by the user. Figure 2 shows a diagram of the opening angle test. Tests show that $\theta_{\mathrm{crit}} = 0.5$ rad is a good compromise between speed and completeness (cf. Section 3.2).

2.2. Binary detection

After the tree is built the search for binaries begins. Every particle P is considered as a potential primary in turn and the opening angle test is used to determine whether a cell needs to be opened to search for satellites of P within that cell. If the open cell contains daughter cells, the same test is applied to them recursively. This continues until a cell passes the opening angle test ($\theta \leqslant \theta_{\mathrm{crit}}$) or has no more daughters (i.e., the cell contains a single particle). In either case the speed v of the center of mass of the cell relative to P is then compared to the mutual escape speed $v_{\mathrm{esc}} = \sqrt{2GM/r}$, where G is the gravitational constant, M is the combined mass, and r is the separation. If the cell still has daughters and $v < v_{\mathrm{esc}}$, the daughter cells are forced open and the recursive procedure above resumes. This additional test insures `companion` identifies widely separated systems with low relative speed. Otherwise, if the cell contains a single particle and $v < v_{\mathrm{esc}}$, the particle is tagged as a companion to P.

2.3. System detection

At this point `companion` contains a list of particle–particle binaries. The user has the option to use this list or to have `companion` go further and identify systems of particles (hierarchies). In that case, starting from the initial binary list, `companion` chooses the shortest-period system and replaces its two components with a single particle located at the center of mass of the binary, with the same total mass and linear momentum (angular momentum is ignored). The "radius" of the new particle is set equal to the semi-major axis of the binary orbit, in order to take advantage of filtering options described below (Section 2.4; the collision cross section of the binary depends on the size of the orbit). Any binary in the original list that contained either of the two components of the binary that was replaced is removed from the binary list. `Companion` then performs a binary test for the new center-of-mass particle using the method outlined above (Section 2.2). Any new binaries that are detected are added to the binary list. This process is repeated until all bound systems of particles have been reduced to single center-of-mass particles.

Once the hierarchy option of `companion` has run to completion only two types of particles remain in the spatial tree—those particles that were never part of a binary and thus are original, unbound, single particles, and composite, center-of-mass particles. Each center-of-mass particle represents a separate system and each contains information about the primary and satellite of the system that it replaced. Thus the entire system represented by each composite particle can be reconstructed in the output (see Section 3.1).

2.4. Usage options

`Companion` provides several options to refine and filter searches. The user can choose to search for simple systems (Section 2.2) or complex hierarchical systems (Section 2.3). `Companion` accepts a variety of input and output units (cgs, mks, and "system units" in which $G \equiv 1$). Allowable input formats include plain text and binary, with one particle to

[1] Barnes and Hut (1986) used the center of mass instead of the geometric center of the cell for the opening angle test in order to have the dipole term in the multipole expansion of the gravitational potential vanish. Since `companion` does not use a multipole expansion, the geometric center is sufficient.

Table 1

M_p/M_t	p_ind	p_rad	M_s/M_p	s_ind	s_rad	bind_eng	a	e	i	per
9.99e-01	0	6.82e+08	9.45e-04	3	7.08e+07	-1.42e+35	8.82e+11	0.12	0.00	4.51e+08
			2.94e-06	1	6.24e+06	-2.73e+33	1.43e+11	0.05	0.00	2.93e+07
			3.68e-08	2	1.72e+06	-3.42e+31	1.43e+11	0.06	0.00	2.94e+07
9.44e-04	3	7.08e+07	4.71e-05	4	1.80e+06	-1.31e+31	4.25e+08	0.01	0.00	1.55e+05
			2.54e-05	5	1.54e+06	-4.41e+30	6.84e+08	0.02	0.00	3.17e+05
2.93e-06	1	6.24e+06	1.25e-02	2	1.72e+06	-5.86e+28	2.45e+08	0.55	0.00	1.21e+06

`Summary: 3 systems, 6 binaries, total mass considered = 1.995755e+30`

a line and columns representing mass, radius, 3-D position vector, and 3-D velocity vector, respectively.

`Companion` also contains several filter options so that only binaries and hierarchical systems that meet certain criteria are reported. The user can specify a maximum eccentricity, minimum binding energy, maximum semimajor axis, and minimum periapse (including a criterion to reject binaries on re-impact trajectories). If a system is particularly interesting, it can be extracted from the original data, with or without the filtering options applied, and studied further in isolation. The user may also change the critical opening angle θ_{crit} used in the opening angle test—reducing θ_{crit} will improve completeness but increase the computation time, and vice versa.

3. Tests

3.1. Illustrative test

To test `companion` and demonstrate its capabilities, we created a hierarchical system based on our Solar System that includes the Sun, the Earth and Moon, Jupiter, Io, and Europa, all in the $z = 0$ plane. We chose this system because it contains two subsystems (Earth–Moon and Jupiter-satellites) that `companion` should detect. Table 1 shows the normal (non-hierarchy) output from `companion` for this system. Each line of data output corresponds to a binary. In order, the columns are: mass ratio of the primary to the total system mass; index number of the primary (an integer assigned to each line of input data, starting at 0); radius of the primary; mass ratio of the satellite to the primary; index of the satellite; radius of the satellite; binary binding energy; semimajor axis; eccentricity; inclination; and orbital period. In this example output units are mks (inclination is always in radians). In this human-readable format, satellites sharing the same primary only show data from the fourth column on to emphasize associations. `Companion` also outputs a text machine-readable format for ease of interfacing with analysis routines.

In this example `companion` has identified three systems: (1) the Sun (particle 0) with Jupiter (particle 3), the Earth (particle 1), and the Moon (particle 2) as satellites; (2) Jupiter with Io (particle 4) and Europa (particle 5) as satellites; and (3) the Earth with the Moon as its satellite. The summary line at the end gives the number of systems (i.e., number of primaries), the number of binaries (primary-satellite pairs), and the total mass considered in the search. Note that since the relative speed between Jupiter's satellites and the Sun is greater than the escape speed from the Sun at their distance, `companion` does not identify them as members of the Sun system, even though they are members of the Jupiter system and Jupiter is a member of the Sun system.

Table 2 shows `companion` output for the same system with the hierarchy option turned on. The first column is the index number of the center-of-mass particle that has replaced the primary (third column) and satellite (sixth column). The other columns have the same meaning as in the normal `companion` output. Note that index numbers in the third and sixth column that are above 5 are also center-of-mass particles (recall numbering starts at 0 and there are 6 original particles in this test). Each separate system is identified by a new header line; in this case there is only one system identified (everything, including the jovian satellites, is determined to belong to one system). The summary line for each system shows the total mass of the system with respect to the total mass of all particles considered, the maximum semimajor axis (a rough indication of the physical "size" of the system), and the total binding energy. After all systems have been listed, a global summary reports the total number of systems found (broken down into two-particle and multiple-particle systems), the total number of original particles, and the total mass considered in the search.

Figure 3 shows a visual representation of the hierarchical output for this test.[2] Jupiter (particle 3) and Io (particle 4) have the shortest period so they become the first center-of-mass particle (particle 6, shown in Fig. 3 as the black dot one level above Jupiter and Io). The next shortest period is the Jupiter–Io system with Europa (particle 5 in Fig. 3). The Jupiter–Io system is combined with Europa to form a new center-of-mass particle (7) that represents the entire Jupiter system. The next shortest period is the Earth–Moon system,

[2] The software used to create the diagram Fig. 3 is also publically available at http://www.astro.umd.edu/~zoe/companion/. After `companion` has been run on the user's data with the hierarchy option run the plotting script with the index of the center of mass particle at the top of the desired system. The plotting script will produce a super mongo script.

Table 2

c_ind	M_p/M_t	p_ind	p_rad	M_s/M_p	s_ind	s_rad	bind_eng	a	e	i	per
10	9.99e-01	9	1.43e+11	9.45e-04	7	6.84e+08	-1.42e+35	8.82e+11	0.12	0.00	4.51e+08
9	9.99e-01	0	6.82e+08	2.97e-06	8	2.45e+08	-2.77e+33	1.43e+11	0.05	0.00	2.93e+07
8	2.93e-06	1	6.24e+06	1.25e-02	2	1.72e+06	-5.86e+28	2.45e+08	0.55	0.00	1.21e+06
7	9.45e-04	6	4.25e+08	2.54e-05	5	1.54e+06	-4.41e+30	6.84e+08	0.02	0.00	3.17e+05
6	9.44e-04	3	7.08e+07	4.71e-05	4	1.80e+06	-1.31e+31	4.25e+08	0.01	0.00	1.55e+05

```
System summary: mass = 2.00e+30, max semimajor axis = 8.82e+11, total binding energy = -1.45e+35

1 system found: 0 2-particle systems and 1 multi-particle system
Total number of original particles: 6
Total mass in original particles: 2.00e+30
```

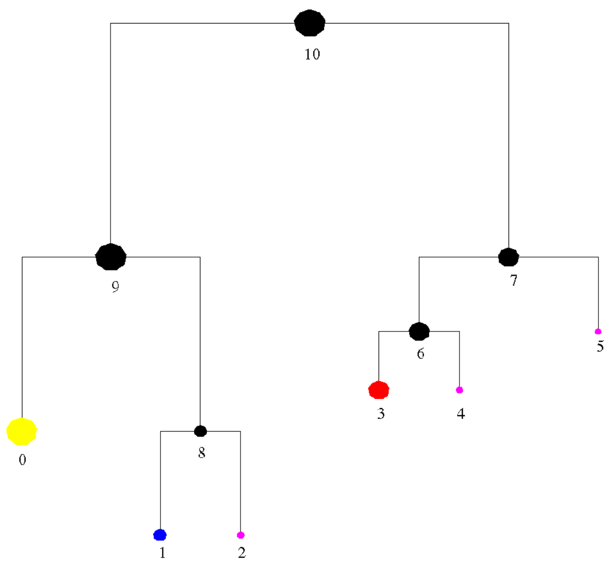

Fig. 3. A visual representation of the output from the hierarchical option in `companion` for a pseudo Solar System that included six particles: the Sun, the Earth, the Moon, Jupiter, Io, and Europa, all in a coplanar configuration. Particle 0 represents the Sun, 1 the Earth, 2 the Moon, 3 Jupiter, 4 Io, 5 Europa. All particles with particle indices above 5 are center-of-mass particles. The radius of the dots corresponds to the mass of the particle with the most massive five times the radius of the smallest. Similarly, the lengths of the vertical branches in the tree correspond to the orbital period of each binary with the longest period four times that of the shortest.

particles 1 and 2 at the bottom of Fig. 3. They are combined to form another center-of-mass particle (8). The period of the Earth–Moon system around the Sun is shorter than the period of the Jupiter system around the Sun, thus the Earth–Moon system is combined with the Sun (particle 0) to form center-of-mass particle 9. Finally, the Jupiter system is combined with the Sun–Earth–Moon system to form particle 10. Ultimately the system is reduced to one center-of-mass particle.

3.2. Performance tests

The development goal for `companion` was to find binaries, including hierarchical systems, in better than $\mathcal{O}(N^2)$ time. Figure 4 indicates this goal has been achieved: shown is the time needed to run `companion` on six numerical simulations of catastrophic asteroid collisions with various N and initial conditions. The default value $\theta_{\text{crit}} = 0.5$ rad was used and no filtering was performed. Figure 4 shows that the time it takes `companion` to complete the search for binary systems scales linearly with $N \log N$ for both normal and hierarchical search options. The scatter in both plots is due to the fact that several different simulations with different initial conditions were used in these tests. The hierarchy version of `companion` takes longer because each time a center-of-mass particle is replaced, a search for companions to that new particle is performed. In general, the number of binaries in a simulation is significantly less than the number of particles in the simulation.

To test the completeness of `companion` (the ability for it to identify all binaries in the data set being tested), we used $\theta_{\text{crit}} = 0$, effectively forcing `companion` to behave as an inefficient N^2 code, without any chance of missing a binary. From this test we found that for $\theta_{\text{crit}} = 0.5$ rad, `companion` is at least 99% complete for all data sets tested (N-body simulations of catastrophic asteroid collision events which have been run a few days past the collision) and two orders of magnitude faster than a traditional N^2 search method. For catastrophic asteroid collision simulations, $\theta_{\text{crit}} = 0.5$ rad optimizes completeness and speed. In other scenarios it is possible that a more conservative opening angle is required.

4. Results

An older version of `companion` without the hierarchy option was used for the analysis of satellite formation simulations in Durda et al. (2004). The updated version produces similar results for the three data files from Durda et al. (2004) that we used as test cases. For both versions, `companion` was used with two filters applied: (1) a maximum semimajor axis of one Hill radius (at 3 AU from the Sun); and (2) a minimum periapse distance of twice the primary radius. Due to some improvements in how the filters are applied in `companion`, we found a slight difference in the number of satellites reported by the new version ($<$0.5% dif-

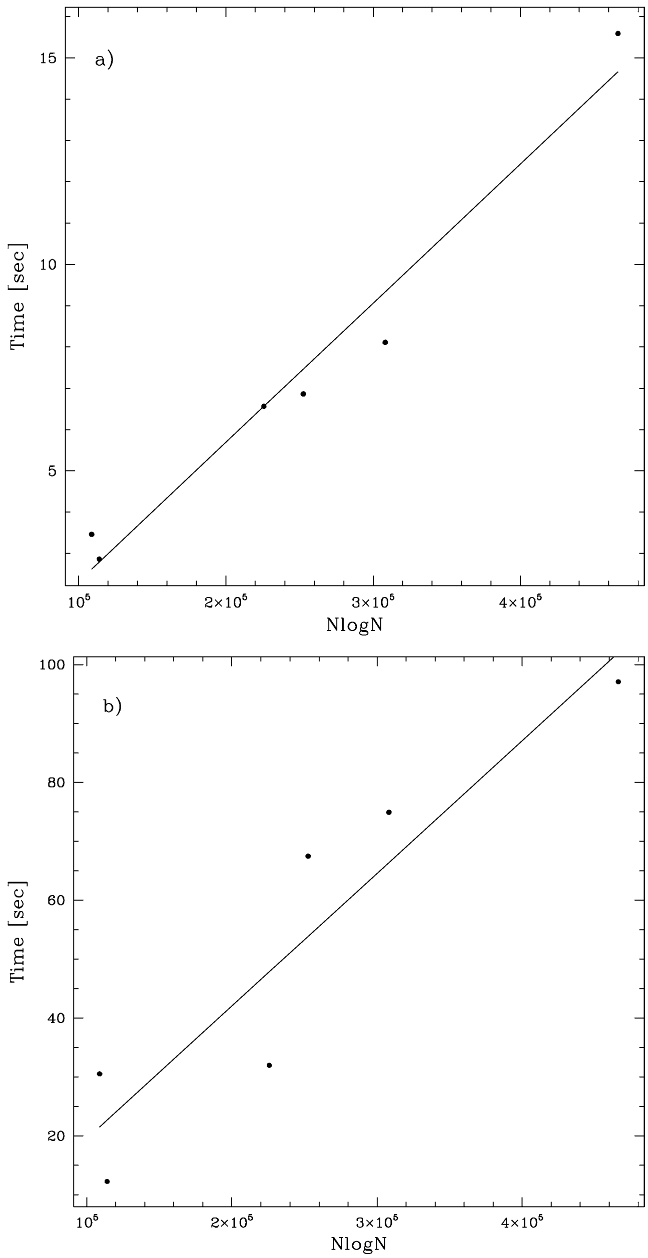

Fig. 4. (a) CPU time versus $N \log N$ in seconds for default `companion` analysis of the results of catastrophic asteroid collision simulations (Durda et al., 2004; Durda, personal communication). (b) Timing results for the same data using the hierarchical search. The data sets contained between 2.6×10^4 and 9.4×10^4 particles. The solid lines are least-squares fits to each data set.

ference). Thus, the overall statistics reported in Durda et al. (2004) are consistent with our tests.

Since Durda et al. (2004) did not have the hierarchy option available, we have done a preliminary analysis with it on the simulation that produced the most binaries. The impact parameters of this simulation are as follows: impact speed $\sim$3 km s^{-1}, impact angle at collision of 30°, diameter of projectile of 34 km, diameter of target of 100 km. We have found a number of interesting hierarchical configurations in their data (Fig. 5 gives an example, using `companion` without any filtering). Most of the more interesting hierarchies occur between smaller particles (what Durda et al. (2004) call "EEBs," or escaping ejecta binaries). These are systems escaping the largest post-collision remnant and that consist of smaller fragments with low relative speeds. We have also run `companion` on the same data with the Hill sphere and periapse cuts mentioned above. `Companion` found 1101 systems with 129 multiple systems and 972 2-particles systems applying the above mentioned cuts without the hierarchy option. With the hierarchy option turned on `companion` found 1020 systems with 862 2-particle systems and 158 multiple systems. This means that about 80 2-particle systems detected without the hierarchy option have their center of mass bound to another system. Figure 6 shows a histogram of the number of N-particle systems. As expected the majority of systems are binaries but there are a significant number of trinary systems ($\sim$10% the number of binaries) and quarternary systems ($\sim$3%) that passed the orbital restrictions.

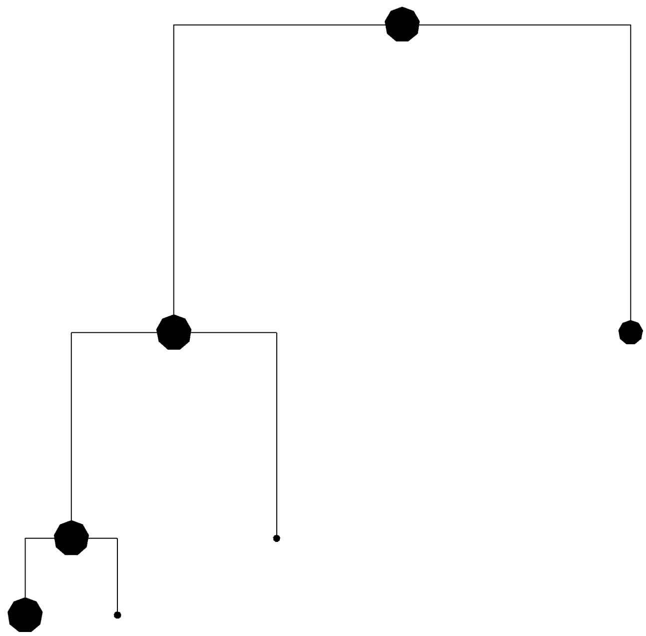

Fig. 5. An example of an interesting hierarchy found by `companion` in a simulation from Durda et al. (2004), with no filtering applied.

We also found 30 multiparticle systems (mostly triples) that seemed to be relatively stable in the sense that they survived for several days. These systems all passed the periapse and semimajor axis filter options described above. In addition, these systems did not contain any particles or binaries that pass within one semimajor axis of any other binary in the system. As a test, some of these systems were extracted from the data file and integrated in isolation for several orbits. Three configurations of particles were found to be most stable: (1) a large primary orbited by two-to-three small particles; (2) a tight binary orbited by a smaller particle; (3) a larger particle orbited by a tight binary. For the inner binary in configuration 2, both equal and unequal-size components worked well. The orbital parameters of the

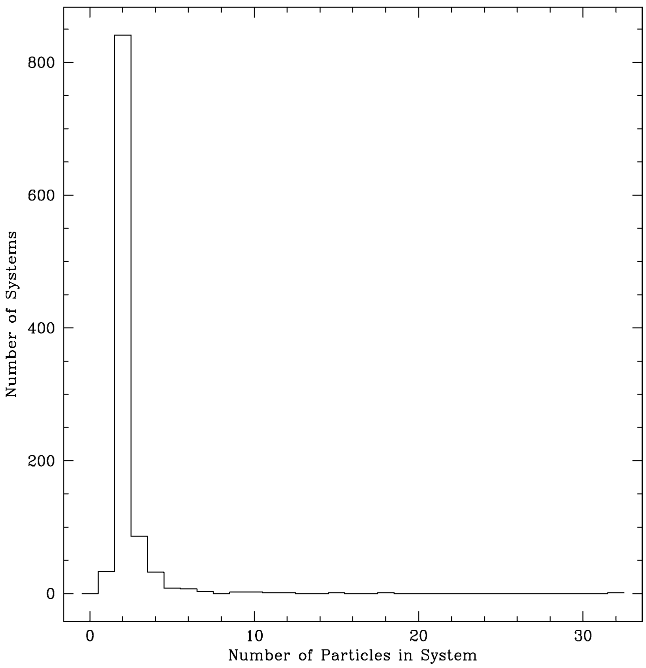

Fig. 6. A histogram of the number of systems found with N particles using the hierarchy option in `companion`. Only original particles are counted, center of mass particles are not included in the calculation of the number of particles in a system.

configurations varied but the tight binaries in configurations 2 and 3 often had relatively moderate-to-low eccentricity ($\leqslant 0.4$).

5. Conclusions

In this paper we presented `companion`, a publicly accessible, efficient code for finding binaries and bound systems in output from numerical simulations. We found that both simple and complex searches scale as $\mathcal{O}(N \log N)$ with the new code. We discussed the capabilities of this code in the context of binary asteroid formation, showing that data from Durda et al. (2004) contains previously unreported hierarchical systems. However, it should be noted that `companion` can in principle be applied to any data set that includes particle mass, radius, position, and velocity.

The completeness of `companion` is dependent on the critical opening angle θ_{crit}. For the evolved asteroid collision simulations tested here, the default value of 0.5 rad provided better than 99% completeness. Other configurations may exist for which a more stringent value of θ_{crit} is required, at the cost of computation time, such as those with large numbers of barely bound, spatially far removed particles. It also must be emphasized that all binaries and multiple systems reported by `companion` are instantaneous, could very well be transient, and may only exist in the context of surrounding particles (i.e., such systems may fly apart when extracted from their broader context). Thus, it may be most useful to apply `companion` to dynamically evolved data sets, as we have done, or to use `companion` to study the statistics and evolution of transient systems.

Acknowledgments

This material is based upon work supported by the National Aeronautics and Space Administration under Grant Nos. NAG511722 and NGT550454 issued through the Office of Space Science and by the National Science Foundation under Grant No. AST0307549. The analysis presented in this paper was carried out using the Beowulf cluster administered by the Center for Theory and Computation of the Department of Astronomy at the University of Maryland. Z.M.L. thanks the Kavli Institute of Theoretical Physics at the University of California, Santa Barbara, as well as D. Hamilton and A.J. Young.

References

Barnes, J., Hut, P., 1986. A hierarchical $\mathcal{O}(N \log N)$ force-calculation algorithm. Nature 324, 446–449.

Belton, M., Carlson, R., 1994. 1993 (243) 1. IAU Circ. 5948.

Belton, M., Chapman, C., Thomas, P., Davies, M., Greenberg, R., Klaasen, K., Byrnes, D., D'Amario, L., Synnott, S., Merline, W., Petit, J.-M., Storrs, A., Zellner, B., 1995. The bulk density of Asteroid 243 Ida from Dactyl's orbit. Nature 374, 785–788.

Benz, W., Asphaug, E., 1999. Catastrophic disruptions revisited. Icarus 142, 5–20.

Chauvineau, B., Farinella, P., 1995. The evolution of Earth-approaching binary asteroids: A Monte Carlo dynamical model. Icarus 115, 36–46.

Davis, M., Efstathiou, G., Frenk, C., White, S.D.M., 1985. The evolution of large scale structure in a universe dominated by cold dark matter. Astrophys. J. 292, 371–394.

Durda, D.D., Bottke Jr., W.F., Enke, B.L., Merline, W.J., Asphaug, E., Richardson, D.C., Leinhardt, Z.M., 2004. The formation of asteroid satellites in large impacts: Results from numerical simulations. Icarus 167, 382–396. Also see Erratum: Icarus 170 (2004) 242, reprinted article: Icarus 170 (2004) 243–257.

Goldreich, P., Lithwick, Y., Sari, R., 2002. Formation of Kuiper-belt binaries by dynamical friction and three-body encounters. Nature 420, 643–646.

Governato, F., Moore, B., Cen, R., Stadel, J., Lake, G., Quinn, T., 1997. The local group as a test of cosmological models. New Astron. 2, 91–106.

Margot, J.-L., 2002. Astronomy: Worlds of mutual motion. Nature 416, 694–695.

Margot, J.-L., Nolan, M.C., Benner, L.A.M., Ostro, S.J., Jurgens, R.F., Giorgini, J.D., Slade, M.A., Campbell, D.B., 2002. Binary asteroids in the near-Earth object population. Science 296, 1445–1448.

Merline, W.J., 2001. Asteroid satellites. Bull. Am. Astron. Soc. 33, 1133.

Michel, P., Benz, W., Tanga, P., Richardson, D.C., 2001. Collisions and gravitational reaccumulation: Forming asteroid families and satellites. Science 294, 1696–1700.

Pravec, P., Sarounová, L., Hicks, M.D., Rabinowitz, D.L., Wolf, M., Scheirich, P., Krugly, Y.N., 2002. Two periods of 1999 HF_1—Another binary NEA candidate. Icarus 158, 276–280.

Pravec, P., Sarounová, L., Rabinowitz, D.L., Hicks, M.D., Wolf, M., Krugly, Y.N., Velichko, F.P., Shevchenko, V.G., Chiorny, V.G., Gaftonyuk, N.M., Genevier, G., 2000. Two-period lightcurves of 1996 FG_3, 1998 PG, and (5407) 1992 AX: One probable and two possible binary asteroids. Icarus 146, 190–203.

Richardson, D.C., 2001. Asteroid satellites from tidal disruption simulations. Bull. Am. Astron. Soc. 33, 1351.

Richardson, D.C., Quinn, T., Stadel, J., Lake, G., 2000. Direct large-scale N-body simulations of planetesimal dynamics. Icarus 143, 45–59.

Weidenschilling, S.J., 2002. On the origin of binary transneptunian objects. Icarus 160, 212–215.

Zappalà, V., Cellino, A., Farinella, P., Knezevic, Z., 1990. Asteroid families. I. Identification by hierarchical clustering and reliability assessment. Astron. J. 100, 2030–2046.