

Close Conjunction Detection on Parallel Computer

Liam M. Healy

Naval Research Laboratory, Washington, D.C. 20375-5355

Close conjunction detection is the task of finding which satellites will come within a given distance of other satellites. The algorithms described here are implemented on the Connection Machine (CM) in a program called CM-COMBO. It will find close conjunctions of satellites over a time range for one, a few, or all satellites against the original or another catalog and works with an arbitrary propagator. The problem of comparing an entire catalog against itself is beyond the computing power of current serial machines. This program does not prefilter any orbits and does not make any assumptions about the type of orbit (that it be nearly circular, for instance). This paper describes the algorithm for this computation, the implementation on the CM, and results of several studies using this program.

I. Introduction

A CLOSE conjunction determination algorithm is one that takes a catalog of satellites and determines which satellites come within some specified distance of another over a period of time. Such an algorithm should work with any propagator and make as few assumptions as possible about the types of orbits of satellites in the catalog. A propagator computes the phase space points at some time step after the initial time, given the phase space values at an initial time. The algorithm described here does not depend on any particular propagator.

Traditionally, a single satellite is compared against a catalog of satellites to find close encounters. This is the function of a program called COMBO (Calculation of Miss Distance Between Objects) running on serial computers at U.S. space surveillance centers. More challenging, however, is the problem of the comparison of a large number of satellites, comparisons that are either all to all within a given catalog or between two large sets of satellites. This capability would be useful, for example, in the study of debris, the hazard to a proposed constellation of satellites, or the danger due to overcrowding of the orbital environment. By taking advantage of parallel processing on a computer such as the Connection Machine (CM), it is feasible to do this.

The algorithms described here have all been implemented in CM Fortran on a CM-5E with 256 nodes.

II. Single Satellite

For comparison of a catalog against a single satellite at a particular time, a parallel processor is a natural. With the Cartesian coordinates of the primary satellite available to all processors and the Cartesian coordinates of all the other satellites at the other processors, a direct-distance computation may be performed in parallel. This method as described is adequate only for finding close pairs at a particular time and not for pairs that come briefly within the desired critical distance and then pass out in the time interval.

In order to find close conjunctions in a time interval, we do a Taylor series extrapolation on the motion. We consider at one time

step a continuous time range, say an interval of time T centered on the time t_0 . Thus, the propagator would compute positions and momenta at time t_0 and take a step to compute them at a time T later, and so on. We wish to find all close conjunctions in the range $t \in [t_0 - T/2, t_0 + T/2]$. With the velocity information that comes back from the propagator, we can extrapolate to find the closest approach. Approximating the relative displacement vector of the two satellites with a Taylor series

$$\mathbf{x} = \mathbf{x}_0 + (t - t_0)\dot{\mathbf{x}}_0 + \frac{1}{2}(t - t_0)^2\ddot{\mathbf{x}}_0 + \frac{1}{6}(t - t_0)^3\ddot{\mathbf{x}}_0^{(3)} + \dots \quad (1)$$

we may find the point of closest approach by the minimum value of $\mathbf{x} \cdot \mathbf{x}$. Truncating Eq. (1) at the linear term, we have

$$\mathbf{x} \cdot \mathbf{x} = \mathbf{x}_0 \cdot \mathbf{x}_0 + 2(t - t_0)\mathbf{x}_0 \cdot \dot{\mathbf{x}}_0 + (t - t_0)^2\dot{\mathbf{x}}_0 \cdot \dot{\mathbf{x}}_0 \quad (2)$$

The time at which this is an extremum can be solved by finding the root of the derivative,

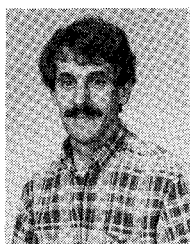
$$\frac{d}{dt}\mathbf{x} \cdot \mathbf{x} = 2\mathbf{x}_0 \cdot \dot{\mathbf{x}}_0 + 2(t - t_0)\dot{\mathbf{x}}_0 \cdot \dot{\mathbf{x}}_0 = 0 \quad (3)$$

To lowest order, the time of closest approach occurs at

$$t_{\text{closest}} = t_0 - \frac{\mathbf{x}_0 \cdot \dot{\mathbf{x}}_0}{\dot{\mathbf{x}}_0 \cdot \dot{\mathbf{x}}_0} \quad (4)$$

The vector \mathbf{x}_0 and its derivative $\dot{\mathbf{x}}_0$ are usually calculated by the propagator. The distance at the time of closest approach is easily calculated from this time, the initial values, and the Taylor series (1).

After all the estimated closest approach distances and times have been computed and likely close conjunctions within the time band extracted, they may be refined using a Newton method iteration in parallel using the actual propagator. Thus, the final answer is available to any desired accuracy within the capabilities of the specified propagator. Although computing $\mathbf{x} \cdot \mathbf{x}$ with more terms in the Taylor series (1) and solving for the time of closest approach by reversion



Liam M. Healy received an A. B. in physics from Harvard in 1979, an M.S. in 1981 and Ph.D. in 1986, both from the University of Maryland, College Park, in Physics. His Ph.D. dissertation was on beam dynamics in particle accelerators. Since 1987 he has been at the Naval Research Laboratory, first as a NRC Cooperative Research Associate and currently as a Research Physicist in the Mathematics and Orbit Dynamics section of the Naval Center for Space Technology. His research interests are in orbit dynamics and Hamiltonian systems, particularly using symbolic and parallel computation.

of series give more accurate results, this turns out to be an ineffective use of computer time. For step sizes of 2 min the error made by this linear approximation on the actual catalog is typically about 10 ms on average, resulting in an average distance error of about 2 m (the reason that the distance of closest approach changes so little relative to the change in time is that the distance–time curve is usually quite flat near the closest approach). The addition of higher orders in the Taylor series, although improving these figures, greatly slows the program. Since the final answer will be refined with the actual propagator anyway, there is little point in doing the extra computations earlier.

Aside from being designed for a parallel computer, this algorithm has advantages over others previously reported. Hoots et al.¹ describe a method that uses extensive prefiltering of orbits, designed to remove from consideration those satellites that could never be near the primary. This requires careful consideration of different kinds of orbits and detailed knowledge of the perturbations to be considered. Furthermore, they use a Newton method iteration with an idealized motion model to solve for the point of closest approach for all satellite pairs not removed by the prefilter; the final stages of iteration use the full propagator. Prefiltering the set requires extra analysis and programming; furthermore, it is of very little advantage in a parallel computer. Mietz² also uses prefiltering, which he calls a “discard phase.” Alfano and Negron³ use a curve-fitting approach, which requires values at multiple time steps. They use a set of four points at once, blending two parabolas formed from the first three and last three points. The blending process is repeated with successive sets of points through the whole data set. The method presented here is simpler conceptually and computationally and requires only positions and velocities at a particular time in order to determine conjunctions over an interval. Dybczyński et al.⁴ attempt to solve the problem of the minimum distance between two orbits, rather than two orbiting satellites. Their method is iterative and is based on strictly Keplerian orbits. So too is the method of Beerer and Bauer.⁵ Theirs is based on spherical trigonometric relationships and is valid only for circular orbits.

None of the previous work mentioned attempts to handle the problem of an all-to-all comparison, only of a particular (primary) satellite against the catalog. For a small number of primaries, one can compute in succession using this one-to-all comparison, but with too many primaries, a different approach is required.

III. Close Conjunction Sieve

The previous section has shown how to compute close conjunctions between a primary (or limited number of primaries) and the remaining catalog of satellites. Of different interest, for example, in verifying debris hazard models, is the hazard of any satellite to any other. In this case, specific conjunction detection is the same, but the logistics of comparison will be different. To make matters easier, consider first the conjunction determination at a particular time.

The computation of close miss distances involves finding all pairs of satellites within a set that are within some specified critical distance c at a given time. The general approach is as follows. One may take the Cartesian coordinates $\mathbf{x} = [x, y, z]^T$, in arrays stored on the parallel computer and duplicate the set:

$$\begin{array}{c} \boxed{\mathbf{x}_1} \boxed{\mathbf{x}_2} \cdots \boxed{\mathbf{x}_n} \\ \boxed{\mathbf{x}_1} \boxed{\mathbf{x}_2} \cdots \boxed{\mathbf{x}_n} \end{array}$$

Then, with one copy fixed, we shift all three arrays by one and compute in parallel the distance of aligned coordinates,

$$\begin{array}{c} \boxed{\mathbf{x}_1} \boxed{\mathbf{x}_2} \cdots \boxed{\mathbf{x}_n} \\ \boxed{\mathbf{x}_1} \cdots \boxed{\mathbf{x}_{n-1}} \boxed{\mathbf{x}_n} \\ \downarrow \text{Compute distances} \downarrow \\ \boxed{|\mathbf{x}_2 - \mathbf{x}_1|} \cdots \boxed{|\mathbf{x}_n - \mathbf{x}_{n-1}|} \end{array}$$

saving in a separate array pairs that are closer than some critical distance c . Then we repeat the shift–distance–save operation until

all possible pairs have been compared. Counting the displacement of the upper array to the left relative to the lower array by s , we compare $|\mathbf{x}_{i+s} - \mathbf{x}_i|$ in parallel for all $1 \leq i \leq n-1$ and serially $1 \leq s \leq n-1$.

This method will determine all close pairs at the time, but it is unnecessarily time consuming. We may greatly reduce the computations by sorting the elements in advance. The position vectors are rearranged so that one Cartesian coordinate of the vector \mathbf{x} , say z , is sorted in ascending order. First observe that if two satellites differ by more than c in one coordinate, they are more than c apart:

$$|z_i - z_j| > c \Rightarrow |\mathbf{x}_i - \mathbf{x}_j| > c \quad (5)$$

As a consequence of the sorted coordinates, if two satellites in the array differ by more than c in one coordinate, then a pair formed by the first satellite and another later in the array than the second will also differ by more than c in the coordinate:

$$k > j > i \quad \text{and} \quad |z_j - z_i| > c \Rightarrow |z_k - z_i| > c \quad (6)$$

These together imply that if, at a certain stage of the computation, the minimum of the differences of the z values is greater than c , no further shifts will produce an aligned pair whose Cartesian distance is less than c . Thus we modify the algorithm above; if we let p be the ranking of satellites by the z value, i.e.,

$$z_{p^{-1}(1)} \leq z_{p^{-1}(2)} \leq \cdots \leq z_{p^{-1}(n)}$$

shifting and finding distances produce

$$\begin{array}{c} \boxed{\mathbf{x}_{p^{-1}(1)}} \boxed{\mathbf{x}_{p^{-1}(2)}} \cdots \boxed{\mathbf{x}_{p^{-1}(n)}} \\ \boxed{\mathbf{x}_{p^{-1}(1)}} \cdots \boxed{\mathbf{x}_{p^{-1}(n-1)}} \boxed{\mathbf{x}_{p^{-1}(n)}} \\ \downarrow \text{Compute distances} \downarrow \\ \boxed{|\mathbf{x}_{p^{-1}(2)} - \mathbf{x}_{p^{-1}(1)}|} \cdots \boxed{|\mathbf{x}_{p^{-1}(n)} - \mathbf{x}_{p^{-1}(n-1)}|} \end{array}$$

Now, however, after each shift an extra step checks the minimum z differences,

$$\min_{1 \leq i \leq n-s} |z_{p^{-1}(i+s)} - z_{p^{-1}(i)}| \quad (7)$$

for shift s in parallel of all aligned pairs. If this minimum is greater than c , we terminate the computation, confident that all pairs that are within a distance c of each other have been found. Since the need for most of the shifts, which are done sequentially, is eliminated, sorting improves the speed of the close conjunction determination by a factor of more than 20.

Having found a fast way to determine all conjunctions within a catalog at a particular time, we seek to expand this to the case of the continuous-time interval $t \in [t_0 - T/2, t_0 + T/2]$ using the method described in Sec. II. We shall need to modify the rearrangement of the arrays, however. If we sort merely by the value at a particular time and then shift until the minimum difference of the z values is at least c , we might miss a pair that is, at t_0 , greater than c apart but will (or had) approached to less than c within the time interval. Thus we choose a different arrangement: The two arrays that are to be compared will not have the same ordering.

For each satellite, we determine the lowest z value and the highest z value in the time interval, i.e.,

$$z^{\max} = \max_{t \in [t_0 - T/2, t_0 + T/2]} z(t) \quad (8)$$

$$z^{\min} = \min_{t \in [t_0 - T/2, t_0 + T/2]} z(t) \quad (9)$$

These are fairly easy to determine. If the sign of the z velocity is the same at the beginning and end of the time interval and the interval is sufficiently small, there is no turning during the interval. Depending on this sign, the z at each end of the time interval is the minimum

Table 1 Arrangement of arrays for direct comparison

Index	B array	C array	Comment
1	α	$a = l^{-1}[h(\alpha) + s(\alpha)]$	Start α L
2	α	$b = l^{-1}[h(\alpha) + s(\alpha) + 1]$	
3	α	$c = l^{-1}[h(\alpha) + s(\alpha) + 2]$	
\vdots	\vdots	\vdots	
$t - s(\alpha)$	α	$l^{-1}[h(\alpha) + t - 1]$	End α L
$1 + t - s(\alpha)$	α	$f = h^{-1}[l(\alpha) - s(\alpha)]$	Start α H
$2 + t - s(\alpha)$	α	$e = h^{-1}[l(\alpha) - s(\alpha) - 1]$	
$3 + t - s(\alpha)$	α	$d = h^{-1}[l(\alpha) - s(\alpha) - 2]$	
\vdots	\vdots	\vdots	
$2[t - s(\alpha)]$	α	$h^{-1}[l(\alpha) - t + 1]$	End α H
$1 + 2[t - s(\alpha)]$	β	$l^{-1}[h(\beta) + s(\beta)]$	Start β L
$2 + 2[t - s(\alpha)]$	β	$l^{-1}[h(\beta) + s(\beta) + 1]$	
\vdots	\vdots	\vdots	

Note: Index is index in two arrays, B array is index in original order of satellite in B array, and C array is index in original order of satellite in C array; α and β are original ordering indices of high movers, and a, \dots, f correspond to example in diagrams.

index in the original array of the Cartesian coordinates that constitute that element. The comments indicate where each block begins and ends. The ordering of the satellites with high rank difference, i.e., what is α, β , etc., is not important. One might sort the rank differences and select α, β, \dots , but it is more convenient and faster to pick a value of t , compute the total number of direct comparisons necessary, see if they will fit in the arrays, and adjust t if necessary. Then the satellites with high rank difference may be picked off in the order they occur in the original array.

V. Categorizing Conjunctions

There are essentially two types of conjunctions observed in the catalog of Earth-orbiting satellites: coincidental conjunctions and traveling partners. The former are satellites that once or periodically are within the critical distance, because they are in different orbits. The latter are satellites that, by accident or by design, are orbiting together, such that they are always or nearly always within the critical distance of one another. A prime example of this is the Russian space station Mir, which is recorded in the U.S. Space Command catalog as seven separate pieces. Clearly, the coincidental conjunctions are of primary interest; without assuming prior knowledge of traveling partners in a given catalog, we would like to sort the conjunctions into the two classes. The traveling partners, although for the most part actually known about in advance, are not assumed to be given. Instead, there are multiple cues that can help determine the type of conjunction for a pair.

At each time step, there are two computations made to identify conjunctions. First, the distances at the current time t_0 are computed. Second, the time and distance of closest approach are estimated according to Eq. (4). If the first distance is less than the critical distance c , or if the second distance is less than c and the time within one-half the interval, $T/2$ of t_0 , then these values are saved along with the identifying numbers for each member of the pair and a key composed from these identifiers that uniquely identifies the pair. At each step, this information is sorted by key and duplicate conjunctions, which are now adjacent in the array, removed.

After the analysis at a particular time interval is done and the information saved, the propagator advances to the next time step and the process is repeated. After the entire period of simulation has been completed, there will be a large collection of conjunction information from all the time steps. To make sense of this, we will need to sort and analyze it.

The global analysis of conjunction information must be able to make sense of different kinds of distance functions in a way that would match a knowledgeable person's judgment. Consider the four different distance functions given in Fig. 1; in each of these plots the dashed line indicates the critical distance. Plot a is clearly a coincidental conjunction, and there should be no problem categorizing it as such. Likewise, plot b is clearly a traveling partner. Plot c shows the pitfall of judging the two cases based on a hard cutoff c : It is obviously just like Fig. 1b, but at a greater distance, so that part

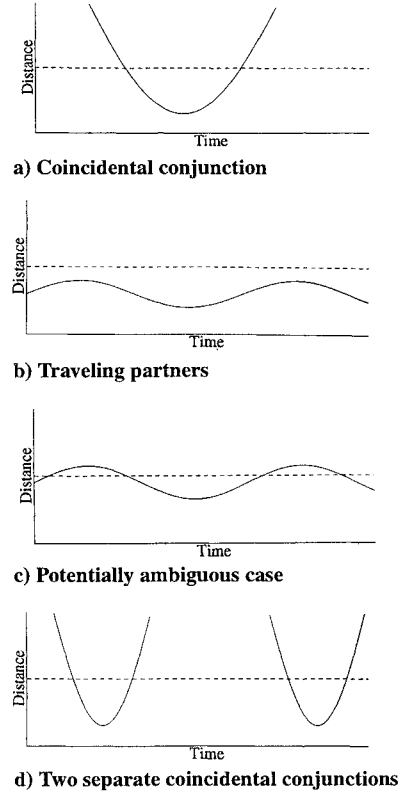


Fig. 1 Some idealized distance functions for satellite pairs; dotted lines indicate critical distance c .

of the time the two satellites are further than c apart. On the other extreme, however, as the far distance increases, it becomes like plot d, which we would want to categorize as two separate coincidental conjunctions. Identifying criteria to accurately discriminate between these cases is difficult; in practice those given below when applied to real catalogs and a realistic critical distance c rarely produce misclassifications.

To formally classify the conjunctions, we sort the conjunction information from all times using the pair key. This information includes the type of conjunction: distance within c at t_0 (a current contact) or estimated within the time interval (an estimated contact) or both. If at least half the time steps have a conjunction of a particular pair and at least 90% of these are a current contact, that pair is considered a traveling partner. Any pair that is not a traveling partner and has an estimated contact is a coincidental conjunction or conjunctions.

A third possibility must be considered, that of two (or more) separate coincidental conjunctions involving the same pair of satellites, which might otherwise be interpreted as traveling partners, such as Fig. 1d. If there is more than one estimated contact, and the standard deviation of the times of the multiple approaches is more than some threshold, say 2.0 s, we shall take it as multiple coincidental conjunctions. Any newly identified multiple coincidental conjunctions are marked with each conjunction separate, and analysis is then redone as two separate coincidental conjunctions.

For the traveling partner, we may compute the mean and standard deviation of the distance and the percentage of time steps in which the distance is within the critical distance. For the coincidental conjunctions, we may find a more precise time and distance of closest approach using a Newton search procedure and the actual propagator (rather than an extrapolation). This is not very computationally intensive, because all coincidental conjunctions obtained through the whole run are refined at the same time in parallel. The use of the actual propagator assures that the resultant values are the closest approach.

VI. Performance

The performance in the sieve mode is dependent on a number of factors. Chief among these are the step size of propagation, the

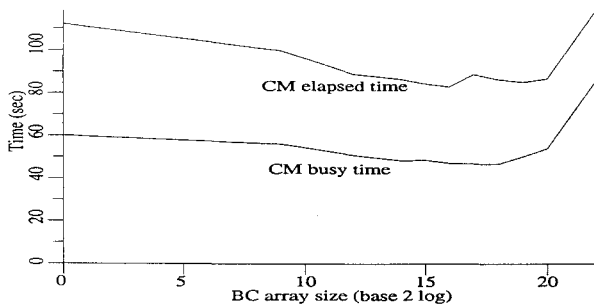


Fig. 2 Time of execution for 1 h of all-to-all simulation for various BC array sizes, with step size of 240 s and $c = 2.5$ km, 256 CM-5 nodes.

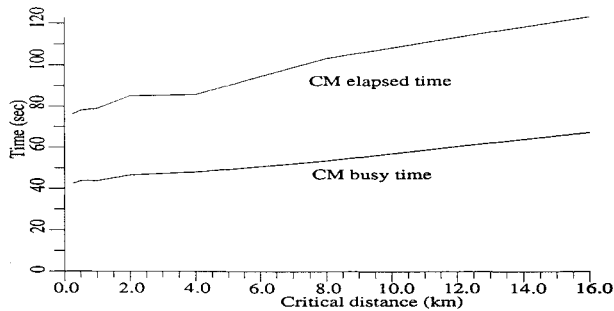


Fig. 3 Time of execution for 1 h of all-to-all simulation for various critical distances c with step size 240 s and broadcast size 16, 256 CM-5 nodes.

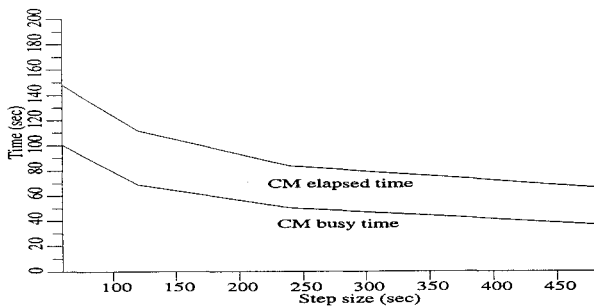


Fig. 4 Time of execution for 1 h of all-to-all simulation for various step sizes, with $c = 1.0$ km and BC size 16,256 CM-5 nodes.

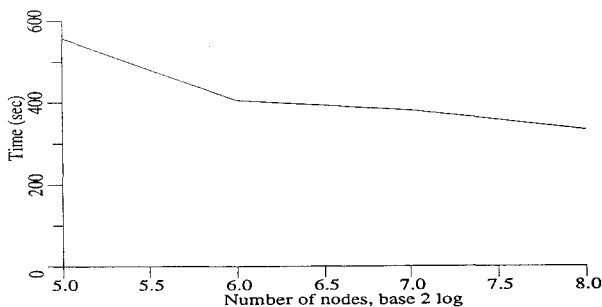


Fig. 5 Time of execution for 30 min of all-to-all simulation, 3-min step size, with $c = 2.5$ km, on different-size machines.

critical distance c , and the BC array size. Also, the number of processors being used is obviously important, as is the size of the satellite dimension for the arrays. The figures below show the relative performance of a 1-h all-to-all close conjunction detection on the catalog of 7613 satellites from May 24, 1993. The figures have both CM elapsed time and CM busy time; the gap between the two times indicates a significant amount of nonparallel work performed on the process manager; as the number of satellites increases, we can expect greater parallelization and a relative closing of the gap. Fig. 3 shows the dependence on the size of the BC arrays, Fig. 4 shows the dependence on critical distance c , and Fig. 5 shows the dependence on the time step. Although Fig. 4 shows that a larger step size

is quicker, care must be taken. In this particular hour, there were two coincidental conjunctions, one at a distance of 0.970 km and the other at a distance of 0.994 km. With c set to 1 km, both were missed by the simulation with step size 480 s and the higher one was missed with a step size of 240 s. This is because the actual distance is so close to the cutoff that the linear extrapolation misses the conjunction for longer step size. One may compensate for this by setting the cutoff slightly higher; for the 240-s step simulation, the program reported that the maximum of the distance adjustments was 0.094 km. Therefore, allowing for estimation errors of about double this (giving a safety factor) should capture both conjunctions; indeed, for $c = 1.2$ km, both appear. Finally, Fig. 5 shows the dependence of execution time on a CM-5 with the \log_2 number of nodes. A CM-5 node consists of a SPARC processor and four vector units. Of course, although an increased number of nodes decreases execution time, it is not "cost effective" for a fixed number of satellites; that is, the product of time and number of nodes increases with an increase in the number of nodes.

VII. New Studies

The unique capability of this program is to find close conjunctions in an all-to-all comparison of a large set of satellites against itself. By using the actual Space Surveillance catalog and running a simulation

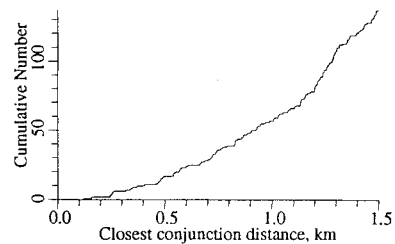
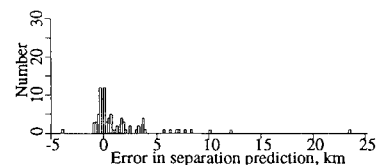
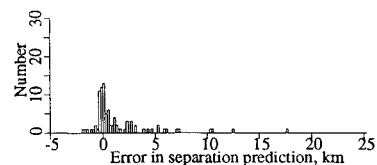


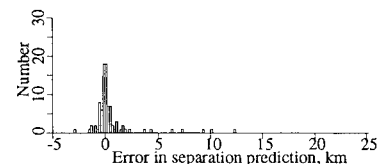
Fig. 6 Number of close conjunctions in 24-h period less than or equal to given distance.



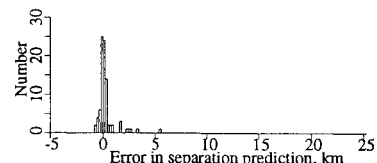
a) Four days



b) Three days



c) Two days



d) One day

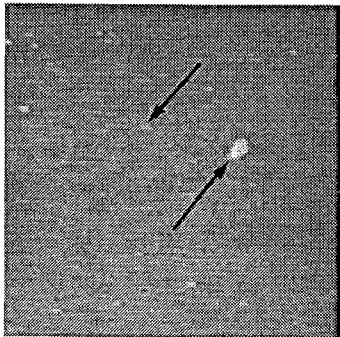
Fig. 7 Histogram of errors in conjunction distance for catalogs of various ages prior to conjunction date.

Table 2 Satellites broken up on July 27, 1993, 00:00 in simulation

Identification number	Semimajor axis, km	Eccentricity	Inclination, deg	Mean anomaly, deg	Argument perigee, deg	Right ascension of ascending node, deg
118	7280.58	0.009158	66.78	-24.3	49.2	159.5
211	7101.36	0.010261	66.58	-158.4	166.1	56.2
4724	7431.89	0.065031	62.81	51.1	-105.9	133.4
11671	6962.73	0.003230	82.51	172.3	-306.5	61.4
11694	7847.57	0.001765	74.03	-2.2	-12.1	11.0
14801	7146.72	0.000728	74.07	155.0	-208.3	83.2
14814	7072.02	0.001526	74.03	11.6	154.2	-44.4
17911	7006.59	0.002355	82.51	-128.7	134.4	8.8
19396	7053.58	0.005222	58.50	-96.8	95.6	-152.5
19440	7730.94	0.018462	73.75	178.2	-262.1	127.8

Table 3 Close conjunctions within 1 km for simulated breakup (July 2, 1993)

Identification number 1	Identification number 2	km	Time
4578	1011270	0.329	17:53:40.308
12347	1009174	0.354	17:50:56.164
5266	1008571	0.481	15:41:59.240
20029	1008917	0.504	13:07:17.362
86654	1009569	0.649	15:07:29.632
5350	1008548	0.712	13:58:09.069
87590	1008679	0.808	16:23:54.327
15472	1009382	0.955	13:47:42.779
9617	1009301	0.976	16:51:23.967
17192	1009065	0.994	17:49:32.322

**Fig. 8 Photograph of TIROS 10 (upper arrow) and NOAA 10 (lower arrow), on May 21, 1993.**

for 24 h, one can assess the degree of crowding in the space environment. Using the catalog starting January 17, 1993, at 14:00:00, we found all satellites that came within 2.0 km of another satellite in the catalog. In this period, there were 140 coincidental conjunctions. There were 57 that came within 1 km; Fig. 6 shows the cumulative number of conjunctions for each distance. The work of Khutorovsky et al.,⁷ taking a different approach, supports these results.

As a measure of the effectiveness of long-term prediction with the existing propagator PPT2⁸ one can predict the conjunctions several days off and then, with daily catalogs updated with new observations, try to predict again for that time period. As the actual date gets closer, the conjunction distance should converge to the actual value. Figure 7 shows the distribution of errors made in predicting the closest conjunction distance with catalogs 4, 3, 2, and 1 day prior to the day of conjunction.

As a dramatic indication of a conjunction, the Naval Space Command ran conjunction predictions daily of 177 satellites with active payloads against the catalog using CM-COMBO. A conjunction was predicted for TIROS 10, launched July 2, 1965, and NOAA 10, launched September 17, 1986 (two weather satellites). The closest distance predicted was 2.113 km on May 21, 1993, 05:53:27.876. This was videotaped by the Air Force Maui Optical Station in Hawaii, and a frame from that videotape showing the conjunction is provided in Fig. 8.

One simulation that can be performed easily with this program is the computation of conjunctions of the existing catalog to a debris cloud. With this CM program and the actual catalog from the Naval Space Command for July 27, 1993, we simulated the explosions of 10 low-Earth-orbit satellites, each into 400 fragments, with peak distribution speeds of 100 m/s. A simulation with 4-min time steps for a period of 6 h after the explosions comparing the debris against the actual remaining catalog showed that 297 different satellites from the catalog would come within 5.0 km of a fragment, with the closest approach being 0.329 km. The run time on the 256-node CM-5 for this simulation was about 17.5 min. The satellites whose breakup was simulated, with their orbital elements, are given in Table 2, and the conjunctions within 1 km are given in Table 3.

VIII. Conclusions

Using a parallel computer, it is possible in a reasonable amount of time to find and classify the conjunctions all to all within a large (8000 or more satellite) catalog or between two large catalogs. This is accomplished with a sieve, which by sorting the Cartesian positions is able to eliminate from consideration the vast majority of possible pairs. This method has been implemented on a CM and was used to predict a conjunction of two weather satellites (which were then subsequently photographed) and has been used for several debris studies.

Acknowledgments

This work was supported by the Naval Space Command. Bernie Kelm of Allied Signal Technical Services was instrumental in the development and operation of this program. Edna Jenkins and Jon Boers of the Naval Space Command were very helpful in performing the studies described in Sec. VII. Shannon Coffey of the Naval Research Laboratory and Steve Knowles of Naval Space Command were essential in bringing this project about.

References

- Hoots, F. R., Crawford, L. L., and Roehrich, R. L., "An Analytic Method to Determine Future Close Approaches Between Satellites," *Celestial Mechanics*, Vol. 33, 1984, pp. 143-158.
- Mietz, L. A., "Predicting Line-of-Sight Conjunction Between a Specific Satellite and Others," M.S. Thesis, Air Force Inst. of Technology, 1973; also document AD-774388, available from National Technical Information Service, Springfield, VA.
- Alfano, S., and Negron, D., "Determining Satellite Close approaches," *Journal of the Astronautical Sciences*, Vol. 41, 1993, pp. 217-226.
- Dybczynski, P. A., Jopek, T. J., and Serafin, R. A., "On the Minimum Distance Between Two Keplerian Orbits with a Common Focus," *Celestial Mechanics*, Vol. 38, 1986, pp. 345-356.
- Beer, J., and Bauer, T., "Determination of Closest Approach and Duration of Encounter for Two Satellites in Circular Non-Coplanar Orbits," *Advances in the Astronautical Sciences*, Vol. 54, 1983, pp. 335-343; also American Astronautical Society, Paper AAS 83-336.
- Phipps, W. E., Neta, B., and Danielson, D. A., "Parallelization of the Naval Space Surveillance Satellite Motion Model," *Journal of the Astronautical Sciences*, Vol. 41, 1993, pp. 207-216.
- Khutorovsky, Z., Boikov, V., Kamensky, S., Makhonin, G., and Smelov, V., "Collision Risk Analysis for Satellites in LEO," NAVSPACOM/Russia Orbit Determination and Prediction Workshop Washington, DC, July 1994.
- Solomon, D., "The NAVSPASUR Satellite Motion Model," Naval Research Lab., Washington, DC, 1991.