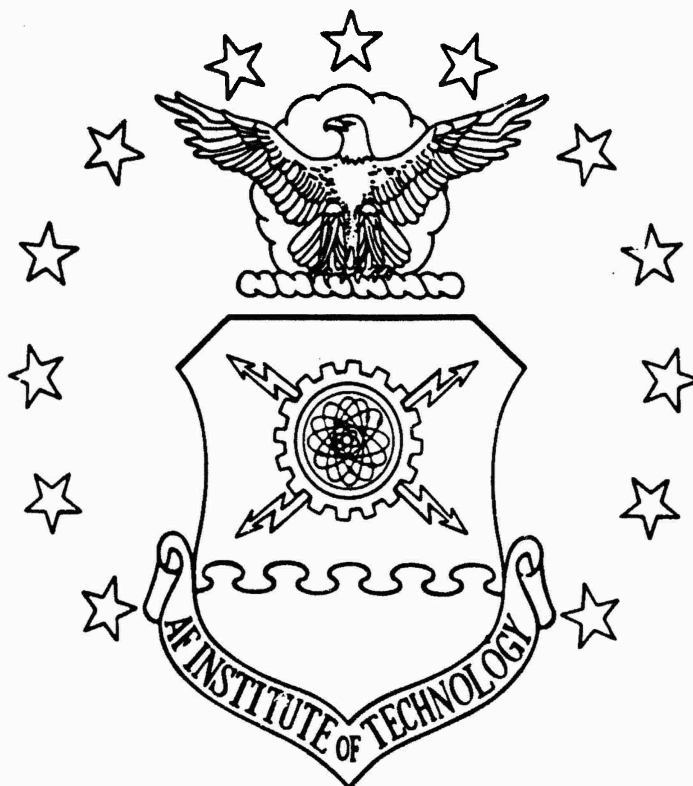


AD-A179 385



DETERMINING THE PROBABILITY
OF CLOSE APPROACH
BETWEEN TWO SATELLITES

THESIS

Randal L. Richey
Captain, USAF

AFIT/GA/AA/85D-8

This document has been approved
for public release and sale; its
distribution is unlimited.

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

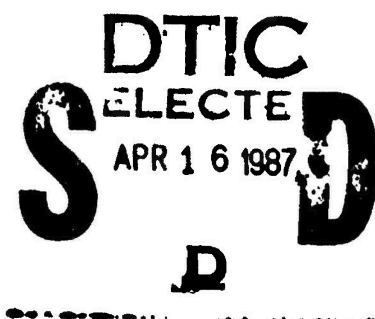
PII Redacted

87 4

16 02

DTIC
ELECTE
APR 16 1987
S
D

AFIT/GA/AA/85D-8



DETERMINING THE PROBABILITY
OF CLOSE APPROACH
BETWEEN TWO SATELLITES

THESIS

Randal L. Richey
Captain, USAF

AFIT/GA/AA/85D-8

Approved for public release; distribution unlimited

DETERMINING THE PROBABILITY OF CLOSE APPROACH
BETWEEN TWO SATELLITES

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Astronautical Engineering

Randal L. Richey, B.S.
Captain, USAF

December 1986



Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Acknowledgement

I would like to thank Dr. William Wiesel for his technical assistance and even more importantly for the constant encouragement he gave me along the way.

Randal L. Richey

Table of Contents

	Page
Acknowledgements	ii
List of Figures	v
List of Tables	vi
Abstract	vii
I. Introduction	1
II. Background	3
III. Probability of Close Approach Between Satellites in Elliptical Orbits	5
Theory	5
Mean Anomaly as a Function of True Anomaly	17
Derivative of Mean Anomaly With Respect to True Anomaly	20
Finding Close Approach Boundaries Given the True Anomaly of Satellite 1	22
Algorithm Summary	37
IV. Probability of Close Approach Between Satellites in Circular Orbits	40
Full Circular Orbit Method	40
Circular Orbit Approximation Method	45
V. Algorithm Verification	49
Analysis Software	49
Statistical Simulation of Probability of Close Approach	50
Test Cases	51
VI. Results and Discussion	55
VII. Suggestions and Recommendations	58
Appendix A: Types of Close Approach	60

	Page
Appendix B: Test Results	69
Appendix C: Program Listings	80
Bibliography	122
Vita	123

List of Figures

Figure	Page
1. Description of Close Approach Boundaries	13
2. Example of a Type 2 Close Approach	36
3. Spherical Geometry of Circular Orbit Close Approach	41

List of Tables

Table	Page
1. Conditions Required for Each Type of Close Approach	35
2. Orbital Parameters Held Constant Through All Test Cases	52
3. Orbital Parameters of Test Cases.	53
4. Simulated Probability of Close Approach Versus COPCA Probability of Close Approach	55
5. Simulated Probability of Close Approach Versus EOPCA Probability of Close Approach	56
B-1. COPCA Test Results For Distance Thresholds Less Than 4000 km	70
B-2. COPCA Test Results For Distance Thresholds Not Less Than 4000 km	71
B-3. COPCA Test Results Using Circular Orbit Approximation For Distance Thresholds Less Than 4000 km	72
B-4. COPCA Test Results Using Circular Orbit Approximation For Distance Thresholds Not Less Than 4000 km.	73
B-5. EOPCA Test Results For Circular Orbits With Distance Thresholds Less Than 4000 km	74
B-6. EOPCA Test Results For Circular Orbits With Distance Thresholds Not Less Than 4000 km	75
B-7. EOPCA Test Results For Eccentricity of .1 With Distance Thresholds Less Than 4000 km	76
B-8. EOPCA Test Results For Eccentricity of .1 With Distance Thresholds Not Less Than 4000 km	77
B-9. EOPCA Test Results For Eccentricity of .3	78
B-10. EOPCA Test Results For Eccentricity of .5	79

Abstract

Probability of close approach is the probability that two satellites will be within some specified distance threshold of each other at a random time within a specified time interval. In this paper, methods were developed to calculate probability of close approach between two satellites. To simplify the analysis, the investigation was restricted to satellite orbits and time intervals where the mean anomaly of both satellites can be treated as independent, uniformly distributed random variables. In addition, all orbital parameters, except for mean anomaly, were assumed to be constant over time. This means that all the methods developed in this paper to calculate the probability of close approach will only be valid over very long time intervals where the ratio of the orbital periods of the two satellites can be approximated as an irrational number. Likewise, there can be no perturbations in the orbital parameters of both satellites.

The first method developed was a general method for calculating the probability of close approach between two satellites in elliptical orbits. The method requires numerical integration and direct solution of the roots of a 4th order polynomial during each numerical integration step.

Another method was developed for calculating the probability of close approach between two satellites in circular orbits. This method still requires numerical integration to obtain a solution, but in this case a direct solution was found for the limits of integration. Furthermore, the calculations required during each numerical integration step are much simpler than those required to calculate the probability of close approach with elliptical orbits.

Finally, a direct solution for approximate probability of close approach between two satellites in circular orbits was developed for the case where the angle between the orbital planes of both satellites is not small and the probability of close approach is small.

Both the elliptical orbit and the circular orbit methods of computing probability of close approach yielded results that compare favorably with estimates of probability of close approach derived from statistical simulations.

I. Introduction

There are a variety of problems where the close approach of two satellites is of interest. Here, close approach of two satellites is defined as occurring whenever the distance between two satellites is less than or equal to some distance threshold d_{TH} . When the position and velocity of both satellites are well known the actual time and duration of each close approach can be predicted. However, if the time of interest cannot be predicted, then a deterministic approach to the close approach of two satellites can no longer be used.

The purpose of this paper is to develop methods to calculate the probability of close approach between two satellites at a uniformly distributed random time within a specified time interval. To simplify the analysis, the investigation is restricted to satellite orbits and time intervals where the mean anomaly of both satellites can be treated as independent, uniformly distributed random variables. In addition, all other orbital parameters are assumed to be constant over time. Because of these restrictions, the methods developed to calculate probability of close approach are only valid over very long time intervals with some restrictions to the ratio of the orbital periods of the two satellites (see the Theory section of chapter III).

In general, the goal is to come up with a way to calculate the probability of close approach between two satellites in elliptical orbits. This general method can also be used to calculate the probability of close approach between two satellites in circular orbits, but it is computationally

cheaper to use a method designed specifically to calculate the probability of close approach between two satellites in circular orbits. Similarly, there are some special cases where an approximate method for calculating the probability of close approach gives adequate accuracy at much less computational expense. For these reasons, three different methods will be developed to calculate probability of close approach. The first method is a general method for calculating the probability of close approach between two satellites in elliptical orbits (see chapter III). The second method is for calculating the probability of close approach between two satellites in circular orbits (see chapter IV). The last method is for calculating the approximate probability of close approach between two satellites in circular orbits where the probability of close approach is small and the angle between the orbital planes of the two satellites is not small (see chapter IV).

Finally, to verify that the three methods are correct, the probability of close approach will be computed (using the three methods, where applicable) for a variety of orbital test cases, and the results will be compared to values derived from statistical simulations.

II. Background

Much work has been done in investigating the probability of collision between orbiting bodies (3, 6, and 8). Probability of collision is typically defined as the probability that one orbiting body/satellite will come within some distance threshold of another satellite one or more times within a specified time interval. When dealing with probability of satellite collision, the distance threshold used is typically very small, since it is directly related to the physical size of the satellites involved.

More recently, work has been done in investigating the probability of satellite intercept between satellites in circular orbits (9). Here, a satellite intercept is defined as occurring when two satellites are within some distance threshold of each other for at least some specified length of time. Probability of satellite intercept is the probability that one or more satellite intercepts will occur within some specified time interval that begins at some uniformly distributed random time within another much larger time interval.

Probability of collision and probability of intercept are two examples of probabilistic measures dealing with satellite proximity. This paper introduces a new probabilistic measure of satellite proximity, called probability of close approach. Probability of close approach is the expected fraction of a specified time interval over which the distance between the two satellites is less than or equal to some distance threshold. For very long time intervals, the probability of close approach equals the sum of the durations of all the close approaches that occur within the specified time interval divided by the length of the specified time interval.

Probability of close approach is very different from probability of

collision. When the probability of one or more collisions between two satellites within a very long time interval approaches 1.0, the computed probability of close approach can approach zero. The reason for this is that, regardless of the number of collisions, the actual time that two satellites spend within the collision distance threshold can be a very small fraction of the length of the time interval of interest.

Probability of close approach is closer in concept to probability of intercept, but there are still major differences. Probability of close approach places no requirement on the duration of the close approach, and close approaches that occur after the uniformly distributed random time are of no interest. Despite these differences, probability of intercept and probability of close approach share three major assumptions. First, the time of interest is assumed to be a uniformly distributed random time within some very long time interval. Second, the mean anomalies of both satellites are assumed to be independent, uniformly distributed random variables (probability of intercept was derived only for circular orbits, where mean anomaly always equals true anomaly). Finally, all other orbital parameters are assumed to be constant over time.

Probability of close approach is different from probability of intercept, just as their purposes are different. When it is important that one or more intercepts occur between two satellites, all within a specified time interval starting at some random time, a high probability of intercept is desirable. When it is important that one satellite spend as much of its orbital lifetime as possible within some arbitrary distance threshold of another satellite, then a high probability of close approach is desirable.

III. Probability of Close Approach Between Satellites in Elliptical Orbits

The purpose of this chapter is to develop a method for calculating the probability of close approach between two satellites in elliptical orbits.

Theory

The derivation of probability of close approach can be broken up into three major parts. This section identifies these major parts and describes how they can be put together to calculate probability of close approach. The next three sections of chapter III then completes the solution for each of the three parts.

By definition, probability of close approach is the probability that the distance between two satellites will be less than some distance threshold at a uniformly distributed random time within a specified time interval. To simplify the analysis, two basic assumptions were made. First, all orbital elements, except for mean anomaly, are assumed to be constant over time. Second, the mean anomalies of both satellites are assumed to be independent random variables that are uniformly distributed between 0 and 2π . The first assumption is valid when there are no perturbations to the orbital elements of the two satellites. When is the second assumption valid? At a uniformly distributed time within a specified time interval, the mean anomaly of both satellites can be represented by (1:33, 185)

$$M_1 = (2\pi t/P_1 + M_{10}) \text{ MOD } 2\pi \quad (1)$$

$$M_2 = (2\pi t/P_2 + M_{20}) \text{ MOD } 2\pi \quad (2)$$

where

t = time from start of time interval of interest.

M_1 = mean anomaly of satellite 1.

M_{10} = mean anomaly of satellite 1 at $t = 0$.

M_2 = mean anomaly of satellite 2.

M_{20} = mean anomaly of satellite 2 at $t = 0$.

P_1 = orbital period of satellite 1.

P_2 = orbital period of satellite 2.

and the general function $X \text{ MOD } Y$ represents the remainder of X divided by Y . For purposes of this analysis, t is a random variable uniformly distributed between 0 and the duration of the time interval of interest. Since mean anomaly is a linear function of t (see Eqs (1) and (2)), M_1 and M_2 are also uniformly distributed random variables when the duration of the time interval is less than the orbital period of both satellites, or when the duration of the time interval is equal to some integer multiple of the period of both satellites. Furthermore, over very long time intervals (over 100 orbital periods) M_1 and M_2 approximate (within 1%) random variables that are uniformly distributed between 0 and 2π . Therefore, over long time intervals the mean anomalies of both satellites can be treated as uniformly distributed random variables (not necessarily independent).

When can M_1 and M_2 be considered independent? Let t_0 be some arbitrary time within the time interval of interest, and let n be some nonnegative integer. Using Eqs (1) and (2), at $t = t_0 + nP_1$ the mean anomalies of both satellites can be represented by the following:

$$M_1 = M_{1t} \quad (3)$$

$$M_2 = [2\pi (P_1/P_2)n + M_{2t}] \text{ MOD } 2\pi \quad (4)$$

where

M_{1t} = the mean anomaly of satellite 1 at $t = t_0$.

M_{2t} = the mean anomaly of satellite 2 at $t = t_0$.

Similarly, at $t = t_0 + nP_2$, the mean anomalies of both satellites can be represented by

$$M_1 = [2\pi (P_2/P_1)n + M_{1t}] \text{ MOD } 2\pi \quad (5)$$

$$M_2 = M_{2t} \quad (6)$$

M_1 and M_2 can be considered independent so long as the results of Eqs (4) and (5) are uniformly distributed between 0 and 2π . Once again, let n be some nonnegative integer. Also let $n_{\max}$ be the maximum number of orbital periods within the time interval of interest. There are two cases in which M_1 and M_2 can be considered independent. The first case is where P_1/P_2 and P_2/P_1 are irrational, and for n between 0 and $n_{\max}$, the distances from $(P_1/P_2)n$ and $(P_2/P_1)n$ to the nearest integer are not less than one divided by $n_{\max}$. For practical purposes, the ratios P_1/P_2 and P_2/P_1 can be considered irrational when $(P_1/P_2)n$ and $(P_2/P_1)n$ do not equal integers for n between 0 and $n_{\max}$. The second case is where $(P_1/P_2)n$ and $(P_2/P_1)n$ equal integers for some value of n less than $n_{\max}$, where n is large (1000+).

and either $n_{\max} \text{ MOD } n$ is large (1000+), or the distance from $n_{\max}$ divided by n to the nearest integer is small. For members of the second case where $n_{\max} \text{ MOD } n$ is large, the orbital ratios must also meet the criteria of the first case for n between 0 and $n_{\max} \text{ MOD } n$.

Generally speaking, if P_1/P_2 and P_2/P_1 are irrational, and the time interval of interest is very long (1000+ orbital periods), then the mean anomalies of both satellites can be considered independent, uniformly distributed random variables.

As discussed above, the mean anomalies of both satellites are assumed to be independent random variables that are uniformly distributed between 0 to 2π . This means that the probability density functions of the mean anomalies of the two satellites can be represented by (5:72-73)

$$f(M_1) = \frac{1}{2\pi} \quad \text{for } 0 \leq M_1 < 2\pi \quad (7)$$

$$f(M_2) = \frac{1}{2\pi} \quad \text{for } 0 \leq M_2 < 2\pi \quad (8)$$

where

M_1 = the mean anomaly of satellite 1.

M_2 = the mean anomaly of satellite 2.

Likewise, the joint probability density function of the mean anomalies of both satellites can be represented by (5:135, 139-140)

$$f(M_1, M_2) = \frac{1}{4\pi^2} \quad \begin{array}{l} \text{for } 0 \leq M_1 < 2\pi \\ 0 \leq M_2 < 2\pi \end{array} \quad (9)$$

The probability of close approach (P_{CA}) can be computed by integrating the joint density function over the region of M_1 and M_2 where the distance between the satellites (d) is less than or equal to some distance threshold d_{TH} . Eq (10) is the formula for probability of close approach between the two satellites:

$$P_{CA} = \int_{R_1} \int_{R_2} \frac{dM_2 dM_1}{4\pi^2} \quad (10)$$

where R_2 is the region of M_2 over which a close approach occurs, given M_1 , and R_1 is the region of M_1 over which some close approach with satellite 2 is possible.

To simplify analysis, four functions will be defined. At this point, these functions are strictly symbolic, and no solution for these functions exist. The four functions are $M_1(\nu_1)$, $M_2(\nu_2)$, $d(M_1, M_2)$, and $\Delta M_2(M_1)$. $M_1(\nu_1)$ is the mean anomaly of satellite 1 as a function of the true anomaly of satellite 1. Similarly, $M_2(\nu_2)$ is the mean anomaly of satellite 2 as a function of the true anomaly of satellite 2. The function $d(M_1, M_2)$ represents the distance between the two satellites as a function of their mean anomalies. The last function is $\Delta M_2(M_1)$, which is defined as

$$\Delta M_2(M_1) = \int_{R_2} dM_2 \quad (11)$$

Substituting Eq (11) into Eq (10) results in

$$P_{CA} = \int_{R_1} \left(\int_{R_2} dM_2 \right) \frac{dM_1}{4\pi^2}$$

$$P_{CA} = \int_{R_1} \Delta M_2(M_1) \frac{dM_1}{4\pi^2} \quad (12)$$

Given M_1 , the distance between the two satellites is purely a function of M_2 (all other orbital elements are assumed constant). Each M_2 solution to the equation $d(M_1, M_2) = d_{TH}$ will be referred to as a mean anomaly close approach boundary. The reason for this is that they delimit the mean anomaly regions of close approach between both satellites. For each two M_2 solutions to $d(M_1, M_2) = d_{TH}$, there is a mean anomaly close approach region such that

$$d(M_1, M_2) \leq d_{TH} \quad \text{for } M_{2/i1} \leq M_2 \leq M_{2/i2}$$

$$M_{2/i1} \leq M_{2/i2}$$

where $M_{2/i1}$ is used to represent the beginning of the i th close approach region, and $M_{2/i2}$ is used to represent the end of the i th close approach region. If there is more than one close approach region, then the regions are numbered so that the mean anomaly of the beginning of the $i+1$ close approach region is greater than or equal to the mean anomaly of the end of the i th close approach region. The possible range of each mean anomaly

close approach boundary is from 0 to 2π . The exception to this is when a close approach region crosses 2π . In this case, $M_{2/11}$ (the beginning of the first close approach region) can range from -2π to 0 so that the close approach region that crosses 2π does not have to be broken into two parts. Given all this, when there are n close approach regions and n is greater than zero, the general solution for $\Delta M_2(M_1)$ is

$$\Delta M_2(M_1) = \sum_{i=1}^n (M_{2/12} - M_{2/11}) \quad (12)$$

where

$$-2\pi \leq M_{2/11} \leq 2\pi$$

$$0 \leq M_{2/11} \leq 2\pi \quad \text{for } i > 1$$

$$0 \leq M_{2/12} \leq 2\pi$$

$$d(M_1, M_2) \leq d_{TH} \quad \text{for } M_{2/11} \leq M_2 \leq M_{2/12}$$

$$d(M_1, M_{2/11}) = d_{TH}$$

$$d(M_1, M_{2/12}) = d_{TH}$$

$$M_{2/11} \leq M_{2/12}$$

$$M_{2/12} \leq M_{2/j1} \quad \text{for } n > 1 \text{ and } j > i$$

For example, when there are four M_2 solutions to $d(M_1, M_2) = d_{TH}$, the solution to $\Delta M_2(M_1)$ can be represented by (also see Figure 1)

$$\Delta M_2(M_1) = M_{2/22} - M_{2/21} + M_{2/12} - M_{2/11} \quad (13)$$

where

$$\begin{aligned}
 -2\pi &\leq M_{2/11} \leq 2\pi & 0 &\leq M_{2/12} \leq 2\pi \\
 0 &\leq M_{2/21} \leq 2\pi & 0 &\leq M_{2/22} \leq 2\pi \\
 d(M_1, M_{2/11}) &= d_{TH} & d(M_1, M_{2/12}) &= d_{TH} \\
 d(M_1, M_{2/21}) &= d_{TH} & d(M_1, M_{2/22}) &= d_{TH} \\
 M_{2/12} &\geq M_{2/11} & M_{2/22} &\geq M_{2/21} \\
 M_{2/21} &\geq M_{2/12} \\
 d(M_1, M_2) &\leq d_{TH} \quad \text{for } M_{2/11} \leq M_2 \leq M_{2/12} \\
 d(M_1, M_2) &\leq d_{TH} \quad \text{for } M_{2/21} \leq M_2 \leq M_{2/22}
 \end{aligned}$$

When there are no M_2 solutions to $d(M_1, M_2) = d_{TH}$, $\Delta M_2(M_1)$ can possess one of two possible values. If $d(M_1, M_2) > d_{TH}$ for $0 \leq M_2 \leq 2\pi$, then $\Delta M_2(M_1)$ must equal zero. If $d(M_1, M_2) \leq d_{TH}$ for $0 \leq M_2 \leq 2\pi$, then $\Delta M_2(M_1)$ must equal 2π . However, even when there are no M_2 solutions to $d(M_1, M_2) = d_{TH}$, Eq (12) can still be used to calculate $\Delta M_2(M_1)$ through the following procedure:

$$\text{If } d(M_1, M_2) > d_{TH} \text{ for } 0 \leq M_2 \leq 2\pi \quad (14)$$

$$n = 1$$

$$M_{2/11} = 0$$

$$M_{2/12} = 0$$

$$\text{If } d(M_1, M_2) \leq d_{TH} \text{ for } 0 \leq M_2 \leq 2\pi \quad (15)$$

$$n = 1$$

$$M_{2/11} = 0$$

$$M_{2/12} = 2\pi$$

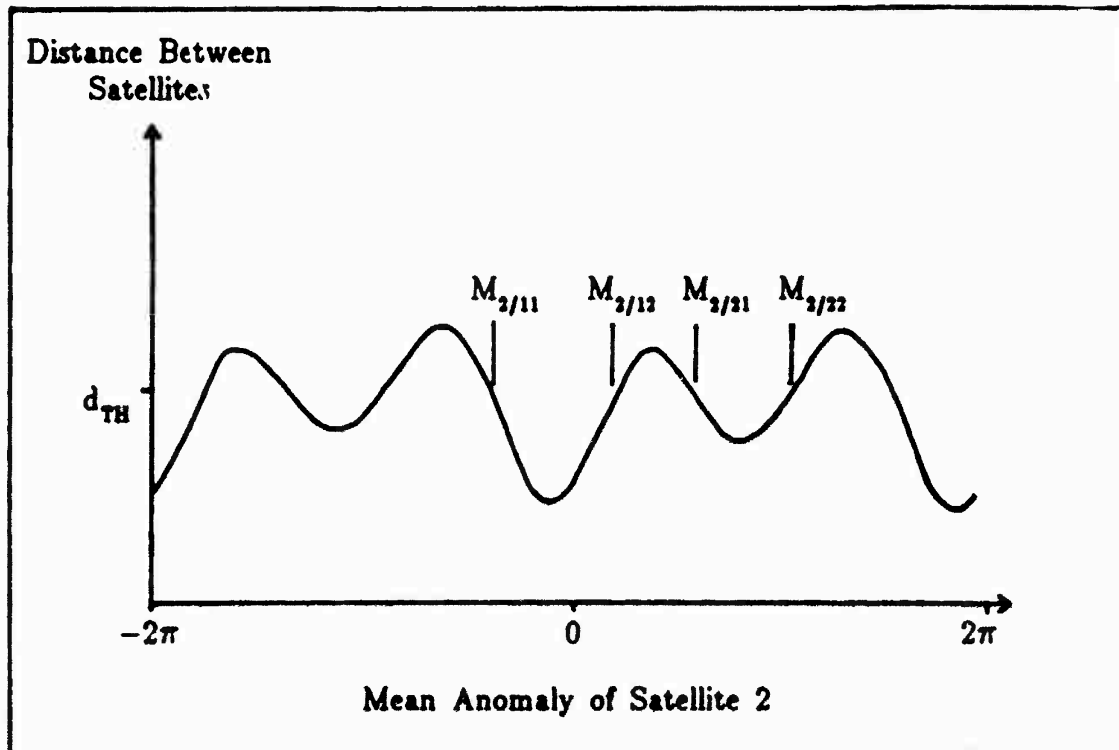


Figure 1. Description of Close Approach Boundaries

With non-circular orbits, true anomaly is much easier to work with than mean anomaly. With this in mind, let d now be represented as a function of true anomaly instead of mean anomaly. In other words, $d(\nu_1, \nu_2)$ now represents the distance between the two satellites as a function of the true anomalies of both satellites (ν_1 and ν_2). Given the true anomaly of

satellite 1 (ν_1), the distance between the two satellites is purely a function of the true anomaly of satellite 2 (ν_2). Each ν_2 solution to the equation $d(\nu_1, \nu_2) = d_{TH}$ will be referred to as true anomaly close approach boundary. For each two ν_2 solutions to $d(\nu_1, \nu_2) = d_{TH}$, there is a true anomaly close approach region such that

$$d(\nu_1, \nu_2) \leq d_{TH} \quad \text{for } \nu_{2/11} \leq \nu_2 \leq \nu_{2/12}$$

$$\nu_{2/11} \leq \nu_{2/12}$$

where $\nu_{2/11}$ is used to represent the true anomaly of the beginning of the i th close approach region, and $\nu_{2/12}$ represents the true anomaly of the end of the i th close approach region. Mean anomaly close approach boundaries and true anomaly close approach boundaries are related in the following way:

$$M_{2/11} = M_2(\nu_{2/11}) \quad (16)$$

$$M_{2/12} = M_2(\nu_{2/12}) \quad (17)$$

where $M_2(\nu_2)$ is the mean anomaly of satellite 2 as a function of true anomaly.

Given ν_1 , ΔM_2 can now be expressed as a function of the true anomaly of satellite 1. When there are n close approach regions and n is greater than zero, the general solution for $\Delta M_2(\nu_1)$ is (also see Eq (12))

$$\Delta M_2(\nu_1) = \sum_{i=1}^n [M_2(\nu_{2/12}) - M_2(\nu_{2/11})] \quad (18)$$

where

$$-2\pi \leq \nu_{2/11} \leq 2\pi$$

$$0 \leq \nu_{2/11} \leq 2\pi \quad \text{for } i > 1$$

$$0 \leq \nu_{2/12} \leq 2\pi$$

$$d(\nu_1, \nu_2) \leq d_{TH} \quad \text{for } \nu_{2/11} \leq \nu_2 \leq \nu_{2/12}$$

$$d(\nu_1, \nu_{2/11}) = d_{TH}$$

$$d(\nu_1, \nu_{2/12}) = d_{TH}$$

$$\nu_{2/11} \leq \nu_{2/12}$$

$$\nu_{2/12} \leq \nu_{2/11} \quad \text{for } n > 1 \text{ and } j > i$$

When there are no ν_2 solutions to $d(\nu_1, \nu_2) = d_{TH}$, Eq (18) can still be used to compute $\Delta M_2(\nu_1)$ through the following procedure (also see Eqs (14) and (15)):

$$\text{If } d(\nu_1, \nu_2) > d_{TH} \text{ for } 0 \leq \nu_2 \leq 2\pi \quad (19)$$

$$n = 1$$

$$\nu_{2/11} = 0$$

$$\nu_{2/12} = 0$$

$$\text{If } d(\nu_1, \nu_2) \leq d_{TH} \text{ for } 0 \leq \nu_2 \leq 2\pi \quad (20)$$

$$n = 1$$

$$\nu_{2/11} = 0$$

$$\nu_{2/12} = 2\pi$$

By changing the integration variable in Eq (12) from mean anomaly of satellite 1 to true anomaly of satellite 1, the equation for probability of close approach becomes (4:212)

$$P_{CA} = \int_{T_1} \Delta M_2(\nu_1) \left(\frac{dM_1(\nu_1)}{d\nu_1} \right) \frac{1}{4\pi^2} d\nu_1 \quad (21)$$

where T_1 is the region of ν_1 over which a close approach is possible.

Using Eq (21) to compute P_{CA} requires integration over the region of ν_1 where some close approach with satellite 2 is possible. Since $\Delta M_2(\nu_1)$ can be discontinuous for some ν_1 , this requires that the limits of integration be found over which the function $\Delta M_2(\nu_1)$ is continuous before Eq (21) can be integrated analytically. Unfortunately, there is normally no closed form solution for the limits of integration. Therefore, to calculate probability of close approach, Eq (21) must be numerically integrated over the complete 2π range of ν_1 .

The final equation for probability of close approach between two satellites in elliptical orbits is

$$P_{CA} = \int_{-\pi}^{\pi} \Delta M_2(\nu_1) \left(\frac{dM_1(\nu_1)}{d\nu_1} \right) \frac{1}{4\pi^2} d\nu_1 \quad (22)$$

where for each numerical integration step, $\Delta M_2(\nu_1)$ is computed using Eqs (18) – (20).

Three more things are needed to complete the solution for probability of close approach:

1. An equation for $M_2(\nu_2)$ (see Eq (18))
2. An equation for $dM_1(\nu_1)/d\nu_1$ (see Eq (22))
3. Given ν_1 and d_{TH} , a method to determine the close approach boundaries ($\nu_{2/11}$ and $\nu_{2/12}$) (see Eq (18) – (20))

The solutions to these problems are the subject of the next three sections of chapter III.

Mean Anomaly as a Function of True Anomaly

The purpose of this section is to derive an equation for mean anomaly as a function of true anomaly.

Eqs (23) and (24) are the well known equations relating true anomaly to eccentric anomaly (2:62), and mean anomaly to eccentric anomaly (1:85):

$$\tan(\nu/2) = [(1+e)/(1-e)]^{1/2} \tan(E/2) \quad (23)$$

$$M = E - e \sin E \quad (24)$$

where

- ν – true anomaly
- e – eccentricity
- E – eccentric anomaly

Substituting

$$\beta = [(1-e)/(1+e)]^{1/2} \quad (25)$$

into Eq (23), and then rewriting it as an equation for eccentric anomaly as a function of true anomaly, results in

$$E = 2 \tan^{-1}[\beta \tan(\nu/2)] \quad (26)$$

Substituting Eq (26) into Eq (24), yields an equation for mean anomaly as a function of true anomaly:

$$M = 2 \tan^{-1}[\beta \tan(\nu/2)] - e \sin\{2 \tan^{-1}[\beta \tan(\nu/2)]\} \quad (27)$$

Substituting

$$\psi = \beta \tan(\nu/2) \quad (28)$$

into Eq (27), results in an equation for mean anomaly as a function of ψ :

$$M = 2 \tan^{-1}\psi - e \sin[2 \tan^{-1}\psi] \quad (29)$$

By applying the trigonometric relationship (7:190) $\sin 2\theta = 2 \sin\theta \cos\theta$, where $\theta = \tan^{-1}\psi$, Eq (29) becomes

$$M = 2 \tan^{-1}\psi - e [2 \sin(\tan^{-1}\psi) \cos(\tan^{-1}\psi)] \quad (30)$$

Substituting (7:193)

$$\sin(\tan^{-1}\psi) = \psi/(1+\psi^2)^{1/2} \quad (31)$$

$$\cos(\tan^{-1}\psi) = 1/(1+\psi^2)^{1/2} \quad (32)$$

into Eq (30), results in

$$\begin{aligned} M &= 2 \tan^{-1}\psi - 2e [\psi/(1+\psi^2)^{1/2}][1/(1+\psi^2)^{1/2}] \\ M &= 2 [\tan^{-1}\psi - e \psi/(1+\psi^2)] \end{aligned} \quad (33)$$

Substituting Eq (28) back into Eq (33), yields

$$M = 2 \{ \tan^{-1}[\beta \tan(\nu/2)] - e \beta \tan(\nu/2)/[1 + \beta^2 \tan^2(\nu/2)] \} \quad (34)$$

As a last step, multiply the right half of Eq (34) by $\cos^2(\nu/2)/\cos^2(\nu/2)$ and simplify:

$$M = 2 \left[\tan^{-1}[\beta \tan(\nu/2)] - \frac{e \beta \sin(\nu/2) \cos(\nu/2)}{[\cos^2(\nu/2) - \beta^2 \sin^2(\nu/2)]} \right] \quad (35)$$

Eq (35) is the equation for mean anomaly as a function of true anomaly, which is one of the things needed to compute the probability of close approach. However, when using Eq (35) to compute mean anomaly in a computer program, the program should first check the value of ν . If ν equals π , then the program should directly set mean anomaly to π instead of trying to calculate mean anomaly using Eq (35), because $\tan(\pi/2)$ is infinite. Likewise, if ν equals $-\pi$, then the program should directly set mean anomaly to $-\pi$. When ν is not equal to $\pm\pi$, then Eq (35) can safely be used to compute the mean anomaly.

Derivative of Mean Anomaly With Respect to True Anomaly

$M(\nu)$ represents mean anomaly as a function of true anomaly (see Eq (35)). The purpose of this section is to solve for the derivative of $M(\nu)$ with respect to true anomaly.

Differentiating Eq (33) yields

$$dM = 2 \left[\frac{1}{(1+\psi^2)} - \frac{e}{(1+\psi^2)} - 2 \frac{\psi^2}{(1+\psi^2)^2} \right] d\psi \quad (36)$$

By using $(1+\psi^2)^2$ as a common denominator in all three terms above, Eq (36) can be simplified into

$$\begin{aligned} dM &= 2 \left[\frac{1+\psi^2 - e(1+\psi^2) - 2\psi^2}{(1+\psi^2)^2} \right] d\psi \\ dM &= 2 \left[\frac{1-e + (1+e)\psi^2}{(1+\psi^2)^2} \right] d\psi \end{aligned} \quad (37)$$

Differentiating Eq (28), yields

$$d\psi = (\beta/2) \sec^2(\nu/2) d\nu \quad (38)$$

Substituting Eqs (28), and (38) into Eq (37), results in

$$dM = 2 \left[\frac{1-e + (1+e) \beta^2 \tan^2(\nu/2)}{[1 + \beta^2 \tan^2(\nu/2)]^2} \right] (\beta/2) \sec^2(\nu/2) d\nu \quad (39)$$

Substituting

$$(1+e)\beta^2 = 1-e$$

into Eq (39), results in

$$dM = 2 \left[\frac{1-e + (1-e)\tan^2(\nu/2)}{[1 + \beta^2 \tan^2(\nu/2)]^2} \right] (\beta/2) \sec^2(\nu/2) d\nu$$

$$dM = 2 \left[\frac{(1-e)(1 + \tan^2(\nu/2))}{[1 + \beta^2 \tan^2(\nu/2)]^2} \right] (\beta/2) \sec^2(\nu/2) d\nu \quad (40)$$

Using the trigonometric relationship

$$1 + \tan^2(\nu/2) = \sec^2(\nu/2)$$

Eq (40) can be simplified to

$$dM/d\nu = 2 \left[\frac{(1-e)\sec^2(\nu/2)}{[1 + \beta^2 \tan^2(\nu/2)]^2} \right] (\beta/2) \sec^2(\nu/2)$$

$$dM/d\nu = \frac{(1-e)\beta \sec^4(\nu/2)}{[1 + \beta^2 \tan^2(\nu/2)]^2} \quad (41)$$

As a last step, multiply the right side of Eq (41) by $\cos^4(\nu/2)/\cos^4(\nu/2)$, and then simplify:

$$dM/d\nu = \frac{(1-e)\beta}{[\cos^2(\nu/2) + \beta^2 \sin^2(\nu/2)]^2} \quad (42)$$

This removes any potential numerical problems at $\nu = \pm\pi$.

Eq (42) is the equation for the derivative of mean anomaly with

respect to true anomaly, which is the second of three things needed to compute the probability of close approach.

Finding Close Approach Boundaries Given the True Anomaly of Satellite 1

Given ν_1 , when there are $2n$ ν_2 solutions to $d(\nu_1, \nu_2) = d_{TH}$ and n is positive, there are n true anomaly close approach regions. The close approach boundaries of the i th close approach region can be represented by $\nu_{2/i1}$ and $\nu_{2/i2}$, where

$$-2\pi \leq \nu_{2/i1} \leq 2\pi$$

$$0 \leq \nu_{2/i1} \leq 2\pi \quad \text{for } i > 1$$

$$0 \leq \nu_{2/i2} \leq 2\pi$$

$$d(\nu_1, \nu_2) \leq d_{TH} \quad \text{for } \nu_{2/i1} \leq \nu_2 \leq \nu_{2/i2}$$

$$d(\nu_1, \nu_{2/i1}) = d_{TH}$$

$$d(\nu_1, \nu_{2/i2}) = d_{TH}$$

$$\nu_{2/i1} \leq \nu_{2/i2}$$

$$\nu_{2/j2} \leq \nu_{2/j1} \quad \text{for } n > 1 \text{ and } j > i$$

The close approach boundaries for each close approach region are required to calculate $\Delta M_2(\nu_2)$ (see Eq (18)), and a method to determine the close approach boundaries of each close approach region is the last thing required to complete the solution for probability of close approach between two satellites in elliptical orbits. The purpose of this section is to develop a

method to calculate $\nu_{2/11}$ and $\nu_{2/12}$ for each close approach region given the true anomaly of satellite 1.

When numerically integrating Eq (22), only ν_1 and ν_2 are known at the beginning of each numerical integration step. To compute the distance between satellite 1 and 2, the position vectors of both satellites must be determined within a common cartesian coordinate frame. For convenience, the perifocal frame of satellite 2 was selected.

The position vector of satellite 2 in the perifocal frame of satellite 2 can be represented by (1:72)

$$\mathbf{r}_{2/p2} = \{ x_2 \ y_2 \ z_2 \} \quad (43)$$

where

$$x_2 = r_2 \cos(\nu_2) \quad (44)$$

$$y_2 = r_2 \sin(\nu_2) \quad (45)$$

$$z_2 = 0 \quad (46)$$

$$r_2 = p_2 / [1 + e_2 \cos(\nu_2)] \quad (47)$$

and

e_2 = eccentricity of satellite 2

p_2 = semi-latus rectum of satellite 2

ν_2 = true anomaly of satellite 2

r_2 = the magnitude of $\mathbf{r}_{2/p2}$

Likewise, the position vector of satellite 1 in the perifocal frame of satellite 1 can be represented by (1:72)

$$\underline{r}_{1/p1} = \{ x_1 \ y_1 \ z_1 \} \quad (48)$$

where

$$x_1 = r_1 \cos(\nu_1) \quad (49)$$

$$y_1 = r_1 \sin(\nu_1) \quad (50)$$

$$z_1 = 0 \quad (51)$$

$$r_1 = p_1 / [1 + e_1 \cos(\nu_1)] \quad (52)$$

and

e_1 = eccentricity of satellite 1

p_1 = semi-latus rectum of satellite 1

ν_1 = true anomaly of satellite 1

r_1 = the magnitude of $\underline{r}_{1/p1}$

Transforming the coordinate frame of $\underline{r}_1$ from the perifocal frame of satellite 1 to the perifocal frame of satellite 2 can be performed in two steps. The first step is to transform $\underline{r}_1$ from the perifocal frame of satellite 1 to the earth centered inertial reference frame. The last step is to transform $\underline{r}_1$ from the inertial frame to the perifocal frame to the perifocal frame of satellite 2.

The transformation from the perifocal frame to the inertial frame can be done by multiplying the position vector in the perifocal frame by the following transformation matrix (1:82-83):

$$\underline{R}(i, \omega, \Omega) = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (53)$$

where

$$R_{11} = \cos \Omega \cos \omega - \sin \Omega \sin \omega \cos i \quad (54)$$

$$R_{12} = -\cos \Omega \cos \omega - \sin \Omega \sin \omega \cos i \quad (55)$$

$$R_{13} = \sin \Omega \sin i \quad (56)$$

$$R_{21} = \sin \Omega \cos \omega + \cos \Omega \sin \omega \cos i \quad (57)$$

$$R_{22} = -\sin \Omega \sin \omega + \cos \Omega \cos \omega \cos i \quad (58)$$

$$R_{23} = -\cos \Omega \sin i \quad (59)$$

$$R_{31} = \sin \omega \sin i \quad (60)$$

$$R_{32} = \cos \omega \sin i \quad (61)$$

$$R_{33} = \cos i \quad (62)$$

and

i = orbital inclination
 ω = argument of perigee
 Ω = longitude of the ascending node

Regardless of the perifocal plane that the position vector of satellite 1 is transformed into, the position vector of satellite 1 in the inertial frame is unchanged. This means that

$$\underline{r}_{1/I} = \underline{R}(i_1, \omega_1, \Omega_1) \underline{r}_{1/p1} \quad (63)$$

$$\underline{r}_{1/I} = \underline{R}(i_2, \omega_2, \Omega_2) \underline{r}_{1/p2} \quad (64)$$

$$\underline{R}(i_1, \omega_1, \Omega_1) \underline{r}_{1/p1} = \underline{R}(i_2, \omega_2, \Omega_2) \underline{r}_{1/p2} \quad (65)$$

where

- i_1 = inclination of satellite 1
- i_2 = inclination of satellite 2
- ω_1 = argument of perigee of satellite 1
- ω_2 = argument of perigee of satellite 2
- Ω_1 = longitude of the ascending node of satellite 1
- Ω_2 = longitude of the ascending node of satellite 2
- $\underline{r}_{1/I}$ = the position vector of satellite 1 in the inertial frame
- $\underline{r}_{1/p1}$ = the position vector of satellite 1 in the perifocal plane of satellite 1
- $\underline{r}_{1/p2}$ = the position vector of satellite 1 in the perifocal plane of satellite 2
- $\underline{R}(i_1, \omega_1, \Omega_1)$ = the transformation matrix to transform from the perifocal frame of satellite 1 to the inertial frame
- $\underline{R}(i_2, \omega_2, \Omega_2)$ = the transformation matrix to transform from the perifocal frame of satellite 2 to the inertial frame

Multiplying both sides of Eq (65) by $\underline{R}^{-1}(i_2, \omega_2, \Omega_2)$ results in an equation for the position vector of satellite 1 in the perifocal frame of satellite 2:

$$\underline{r}_{1/p2} = \underline{R}^{-1}(i_2, \omega_2, \Omega_2) \underline{R}(i_1, \omega_1, \Omega_1) \underline{r}_{1/p1} \quad (66)$$

The transformation matrix $\underline{R}$ is orthogonal (1:79-83), so

$$\underline{R}^T(i_2, \omega_2, \Omega_2) = \underline{R}^{-1}(i_2, \omega_2, \Omega_2) \quad (67)$$

Substituting Eq (67) into Eq (66), yields the final equation for $\underline{r}_{1/p2}$:

$$\underline{r}_{1/p2} = \underline{R}^T(i_2, \omega_2, \Omega_2) \underline{R}(i_1, \omega_1, \Omega_1) \underline{r}_{1/p1} \quad (68)$$

Eqs (43) through (62), and Eq (68) make it possible to compute the position vectors of both satellites within the perifocal frame of satellite 2. Given that

$$\mathbf{r}_{1/p2} = \{ x_1 \ y_1 \ z_1 \} \quad (69)$$

$$\mathbf{r}_{2/p2} = \{ x_2 \ y_2 \ 0 \} \quad (70)$$

the distance between satellite 1 and satellite 2 can be represented by

$$d = [(x_2 - x_1)^2 + (y_2 - y_1)^2 + z_1^2]^{1/2} \quad (71)$$

Simplifying Eq (70) further, yields

$$\begin{aligned} d &= (x_1^2 - 2x_1x_2 + x_2^2 + y_1^2 - 2y_1y_2 + y_2^2 + z_1^2)^{1/2} \\ d &= (x_1^2 + y_1^2 + z_1^2 + x_2^2 + y_2^2 - 2x_1x_2 - 2y_1y_2)^{1/2} \end{aligned} \quad (72)$$

Substituting r_1^2 for $x_1^2 + y_1^2 + z_1^2$ and r_2^2 for $x_2^2 + y_2^2$, Eq (72) becomes

$$d = (r_1^2 + r_2^2 - 2x_1x_2 - 2y_1y_2)^{1/2} \quad (73)$$

Substituting Eqs (44) - (47) into Eq (73) results in

$$d = \left[r_1^2 + \frac{p_2^2}{[1+e_2\cos(\nu_2)]^{1/2}} - \frac{2x_1p_2\cos(\nu_2)}{1+e_2\cos(\nu_2)} - \frac{2y_1p_2\sin(\nu_2)}{1+e_2\cos(\nu_2)} \right]^{1/2} \quad (74)$$

After squaring both sides, subtracting r_1^2 from both sides, and then

multiplying both sides by $|1+e_2\cos(\nu_2)|^2$, Eq (74) becomes

$$(d^2-r_1^2)[1+e_2\cos(\nu_2)]^2 = p_2^2 - 2x_1p_2\cos(\nu_2)[1+e_2\cos(\nu_2)] - 2y_1p_2\sin(\nu_2)[1+e_2\cos(\nu_2)] \quad (75)$$

Simplifying Eq (75) further

$$(d^2-r_1^2)[1+2e_2\cos(\nu_2)+e_2^2\cos^2(\nu_2)] - p_2^2 + 2x_1p_2\cos(\nu_2)[1+e_2\cos(\nu_2)] = -2y_1p_2\sin(\nu_2)[1+e_2\cos(\nu_2)] \quad (76)$$

Now Eq (76) can be expressed as a polynomial of $\cos(\nu_2)$:

$$A \cos^2(\nu_2) + B \cos(\nu_2) + C = -2y_1p_2\sin(\nu_2)[1+e_2\cos(\nu_2)] \quad (77)$$

where

$$A = e_2[(d^2-r_1^2) + 2x_1p_2] \quad (78)$$

$$B = 2[e_2(d^2-r_1^2) + x_1p_2] \quad (79)$$

$$C = d^2-r_1^2-p_2^2 \quad (80)$$

Squaring both sides again, Eq (77) becomes

$$A^2\cos^4(\nu_2) + 2AB\cos^3(\nu_2) + (2AC+B^2)\cos^2(\nu_2) + 2BC\cos(\nu_2) + C^2 = 4y_1^2p_2^2\sin^2(\nu_2)[1+2e_2\cos(\nu_2)+e_2^2\cos^2(\nu_2)] \quad (81)$$

Using the trigonometric relationship $1-\cos^2\nu_2 = \sin^2\nu_2$ in Eq (70) yield:

$$A^2\cos^4(\nu_2) + 2AB\cos^3(\nu_2) + (2AC+B^2)\cos^2(\nu_2) + 2BC\cos(\nu_2) + C^2$$

$$\begin{aligned}
&= 4y_1^2 p_2^2 [1 - \cos^2(\nu_2)] [1 + 2e_2 \cos(\nu_2) + e_2^2 \cos^2(\nu_2)] \\
&A^2 \cos^4(\nu_2) + 2AB \cos^3(\nu_2) + (2AC + B^2) \cos^2(\nu_2) + 2BC \cos(\nu_2) + C^2 \\
&= 4y_1^2 p_2^2 [1 + 2e_2 \cos(\nu_2) + (e_2^2 - 1) \cos^2(\nu_2) - 2e_2 \cos^3(\nu_2) - e_2^2 \cos^4(\nu_2)] \quad (82)
\end{aligned}$$

Now Eq (82) can be expressed as a single 4th order polynomial:

$$P_1 \cos^4(\nu_2) + P_2 \cos^3(\nu_2) + P_3 \cos^2(\nu_2) + P_4 \cos(\nu_2) + P_5 = 0 \quad (83)$$

where

$$K = 4y_1^2 p_2^2 \quad (84)$$

$$P_1 = A^2 + Ke_2^2 \quad (85)$$

$$P_2 = 2AB + 2Ke_2 \quad (86)$$

$$P_3 = 2AC + B^2 + K(1 - e_2^2) \quad (87)$$

$$P_4 = 2CB - 2Ke_2 \quad (88)$$

$$P_5 = C^2 - K \quad (89)$$

The roots of a 4th order polynomial can be solved for directly (7:103-106). This means that, given ν_1 , Eq (83) can be used to solve for all possible values of $\cos(\nu_2)$ where the distance between the two satellites is equal to d_{TH} .

A special case occurs when the absolute value of y_1 approaches zero. When y_1 equals zero, Eq (83) reduces to (see Eqs (77) to (80))

$$A \cos^2(\nu_2) + B \cos(\nu_2) + C = 0 \quad (90)$$

where

$$A = e_2[(d^2 - r_1^2) + 2x_1p_2] \quad (91)$$

$$B = 2[e_2(d^2 - r_1^2) + x_1p_2] \quad (92)$$

$$C = d^2 - r_1^2 - p_2^2 \quad (93)$$

This also means that, when y_1 equals zero, Eq (83) is equal to the square of a 2nd order polynomial. Theoretically, when y_1 equals zero, the roots of the square of Eq (90) are the same as two copies of the roots of Eq (90). However, in practical applications, this is not the case. When using IEEE double precision arithmetic, the direct solution of the 4th order roots of the square of Eq (90) can result in a pair of complex conjugate roots for each real root of Eq (90), where the real component of each pair of complex conjugate roots would equal one of the real roots of Eq (90), and the imaginary component would be some small value on the order of 10^{-8} .

The addition of any imaginary number to an otherwise valid solution for $\cos(\nu_2)$, makes that solution unusable. Because of this, when the absolute value of y_1 is small, two copies of the roots of Eq (90) should be used, instead of directly solving for the 4th order roots of Eq (83).

Under some conditions, it is possible that both P_1 and P_2 in Eq (83) will equal zero. For example, when satellite 2 is in a circular orbit (e_2 equals zero), both P_1 and P_2 in Eq (83) are always equal to zero. This case

can be handled by checking the values of P_1 and P_2 , before solving for the 4th order roots of Eq (83). If both P_1 and P_2 in Eq (83) are equal to zero, then the desired solutions are the roots of the remaining 2nd order equation. Another special case occurs when y_1 approaches zero and satellite 2 is in a circular orbit. Ignoring the fact that satellite 2 is in a circular orbit, since y_1 is approximately zero, the desired solutions should be same as two copies of the roots of Eq (90). The difference is that the A coefficient of Eq (90) is equal to zero because the eccentricity of satellite 2 is equal to zero. This reduces Eq (90) to a 1st order polynomial (see Eq (94)) with one solution (see Eq (95)):

$$B \cos(\nu_2) + C = 0 \quad (94)$$

$$\cos(\nu_2) = -C/B \quad (95)$$

However, similar to above, this can be handled by checking the value of A before solving for the roots of Eq (90). If A equals zero, then the desired solutions are the same as two copies of the single root of Eq (94) (see Eq (95)).

The procedures above yield two or four values of $\cos(\nu_2)$ that are the roots of Eq (83), or Eq (90) when the absolute value of y_1 is small (on the order of 10^{-8} km). After discarding solutions that are complex or have an absolute value greater than 1, there will be zero, two, or four valid solutions left. Ultimately, when there are two valid solutions, the close approach boundaries $\nu_{2/11}$ and $\nu_{2/12}$ must be found that meet the criteria described in

Eq (18) for n equal to one. Similarly, when there are four valid solutions, the close approach boundaries $\nu_{2/11}$, $\nu_{2/12}$, $\nu_{2/21}$, and $\nu_{2/22}$ must be found that meet the criteria described in Eq (18) for n equal to two.

Two problems remain. The first problem is that both $\cos(\nu_2)$ and $\cos(2\pi-\nu_2)$ are equal to $\cos(\nu_2)$. Given that $\theta = \cos^{-1}[\cos(\nu_2)]$, it is not known whether $\nu_2 = \theta$ or $\nu_2 = 2\pi-\theta$. Of course, ν_2 can be found through the following procedure:

$$\text{if } d(\nu_1, \theta) = d_{TH}$$

$$\nu_2 = \theta$$

else

$$\nu_2 = 2\pi-\theta$$

Unfortunately, this method requires a lot of CPU time to implement, so another way is needed. The second problem is that once the values of ν_2 are found where $d(\nu_1, \nu_2) = d_{TH}$, the solutions for $\nu_{2/11}$, $\nu_{2/12}$, $\nu_{2/21}$, and $\nu_{2/22}$ are still not known. For example, with two ν_2 solutions, there is no way to tell which of the two solutions is $\nu_{2/11}$ or $\nu_{2/12}$ without some additional work. If there were some way to compute $\nu_{2/11}$, $\nu_{2/12}$, $\nu_{2/21}$, and $\nu_{2/22}$ directly from θ , then both problems would be solved.

Let ϕ_1 through ϕ_n equal the valid solutions for $\cos(\nu_2)$, such that

$$1 \leq i \leq n$$

$$\phi_i \geq \phi_{i+1}$$

where

n = number of valid solutions for $\cos(\nu_2)$

Let $\theta_i = \cos^{-1}(\phi_i)$ where $1 \leq i \leq n$ and $0 \leq \theta_i \leq \pi$. Note, because ϕ_i is sorted in descending order (greater i , smaller ϕ_i), θ_i will be sorted in ascending order (greater i , larger θ_i). The goal is now is to find some way to relate $\theta_i - \theta_n$, to true anomaly of the close approach boundaries of each close approach region.

This process is simplified considerably by redefining close approach so that a close approach occurs when satellite 2 is within some distance threshold pd_{TH} of the projection of the position of satellite 1 onto the orbital plane of satellite 2. This new definition of close approach effectively makes close approach a two dimensional problem, and the new definition of close approach is completely equivalent to the old definition, so long as

$$pd_{TH} = (d_{TH}^2 - z_1^2)^{1/2} \quad (96)$$

where

z_1 = the distance from satellite 1 to the orbital plane of satellite 2
(see Eq (69))

From Eq (69), the projection of the position vector of satellite 1 in the perifocal frame of satellite 2 on to the orbital plane of satellite 2 can be represented by

$$r_{1/p2} = \{ x_1 \ y_1 \ 0 \} \quad (97)$$

Two final terms of interest are:

$$B_1 = \{[a_2(1-e_2)-x_1]^2+y_1^2\}^{1/2} \quad (98)$$

$$B_2 = \{[a_2(1+e_2)+x_1]^2+y_1^2\}^{1/2} \quad (99)$$

where B_1 is the distance from r_{1p} to satellite 2's perigee, and B_2 is the distance from r_{1p} to satellite 2's apogee.

Now all the tools are in place. Given the results of Eqs (96) - (99), there are three basic checks, that along with the number of valid solutions to Eq (83), can be used to find a set of equations relating $\nu_{2/11}$, $\nu_{2/12}$, $\nu_{2/21}$, and $\nu_{2/22}$ to θ_1 . These three checks are

1. Is $y_1 > 0$
2. Is $pd_{TH} > B_1$
3. Is $pd_{TH} > B_2$

Since the results of each check is either true or false, there are 8 possible combinations of results. Each one of these combinations represents a different type of close approach which requires up to three different sets of equations to represent possible cases with zero, two, and four valid solutions. Table 1 lists the type of close approach that corresponds with each possible result of the three checks.

TABLE 1

Conditions Required for Each Type of Close Approach

Results of Close Approach Type Checks			Close Approach Type
$y_1 \leq 0$	$pd_{TH} \leq B_1$	$pd_{TH} \leq B_2$	1
$y_1 \leq 0$	$pd_{TH} \leq B_1$	$pd_{TH} > B_2$	2
$y_1 \leq 0$	$pd_{TH} > B_1$	$pd_{TH} \leq B_2$	3
$y_1 \leq 0$	$pd_{TH} > B_1$	$pd_{TH} > B_2$	4
$y_1 > 0$	$pd_{TH} \leq B_1$	$pd_{TH} \leq B_2$	5
$y_1 > 0$	$pd_{TH} \leq B_1$	$pd_{TH} > B_2$	6
$y_1 > 0$	$pd_{TH} > B_1$	$pd_{TH} \leq B_2$	7
$y_1 > 0$	$pd_{TH} > B_1$	$pd_{TH} > B_2$	8

Appendix A contains the actual equations for each type of close approach. As a general convention within each type of close approach, when there are two valid solutions, both $\nu_{2/21}$ and $\nu_{2/22}$ are set to zero. When there are no valid solutions, then the following procedure is used (also see Eqs (19) and (20)):

$$\text{if } pd_{TH} > B_1 \text{ and } pd_{TH} > B_2 \quad (100)$$

$$\nu_{2/11} = 0 \quad \nu_{2/21} = 0$$

$$\nu_{2/12} = 2\pi \quad \nu_{2/22} = 0$$

$$\text{if } pd_{TH} < B_1 \text{ and } pd_{TH} < B_2 \quad (101)$$

$$\nu_{2/11} = 0 \quad \nu_{2/12} = 0$$

$$\nu_{2/12} = 0 \quad \nu_{2/22} = 0$$

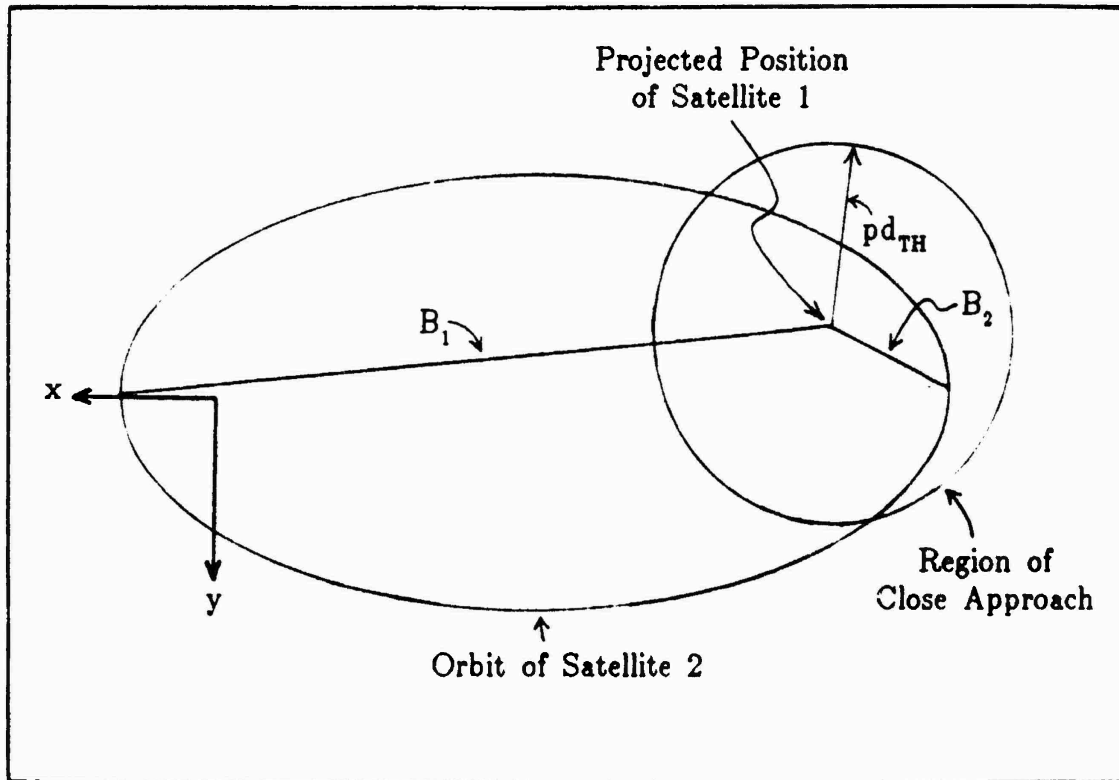


Figure 2. Example of a Type 2 Close Approach

The following example will demonstrate how to use the type of close approach to calculate $\nu_{2/11}$, $\nu_{2/12}$, $\nu_{2/21}$, and $\nu_{2/22}$ from θ_1 through θ_n . Inspection of Figure 2 reveals that $y_1 \leq 0$, $pd_{TH} < B_1$, and $pd_{TH} > B_2$. This means that Figure 2 is an example of a type 2 close approach. Since the orbit of satellite 2 enters and exits the area of close approach once, there are two valid solutions. Once again by inspection, θ_1 is about 160° ,

and θ_2 is about 170° . For two valid solutions, Table A-2 contains the equations to calculate $\nu_{2/11}$ and $\nu_{2/12}$ from θ_1 and θ_2 :

$$\nu_{2/11} = \theta_2 \quad (102)$$

$$\nu_{2/12} = 2\pi - \theta_1 \quad (103)$$

Applying Eqs (102) and (103), $\nu_{2/11}$ equals 170° , and $\nu_{2/12}$ equals 200° . Since there are only two valid solutions, $\nu_{2/21}$ and $\nu_{2/22}$ are by definition equal to zero.

Algorithm Summary

The probability of close approach between two elliptical orbits can be found by numerically integrating the following equation (also see Eq (22)):

$$P_{CA} = \int_{-\pi}^{\pi} \Delta M_2(\nu_1) \left(\frac{dM_1(\nu_1)}{d\nu_1} \right) \frac{1}{4\pi^2} d\nu_1 \quad (104)$$

Where for each numerical integration step, $dM_1(\nu_1)/d\nu_1$ is computed using (also see Eqs (25) and (42))

$$\beta_1 = [(1-e_1)/(1+e_1)]^{1/2} \quad (105)$$

$$dM(\nu_1)/d\nu_1 = \frac{(1-e_1)\beta_1}{[\cos^2(\nu_1/2) + \beta_1^2 \sin^2(\nu_1/2)]^2} \quad (106)$$

and $\Delta M_2(\nu_1)$ is computed using the following procedure:

1. Compute the positions of both satellites within the perifocal frame of satellite 2 using Eqs (43) – (62), and (68).
2. If the absolute value of y_1 is small (on the order of 10^{-8} km) and e_2 is non zero, then find the roots of Eq (90) and use two copies of those roots to obtain four solutions. If the absolute value of y_1 is small and e_2 is zero, then find the root of Eq (94) and use two copies of that root to obtain two solutions. If the absolute value of y_1 is not small, find the roots of Eq (83).
3. Discard those roots that are complex, or those with absolute values that exceed one. The roots that remain are valid solutions.
4. Let ϕ_1 through ϕ_n equal the valid solutions for $\cos(\nu_2)$ such that

$$1 \leq i \leq n$$

$$\phi_i \leq \phi_{i+1}$$

where n is the number of valid solutions. Let $\theta_i = \cos^{-1}(\phi_i)$ where $1 \leq i \leq n$ and $0 \leq \theta_i \leq \pi$.

5. Using Eqs (98) and (99), compute the two distance bounds, B_1 and B_2 . Use Eq (96) to compute the close approach projected distance threshold, pd_{TH} .

6. Perform the close approach checks, and locate the appropriate type of close approach in Table 1. Look up the desired type of close approach in Appendix A, then using the number of valid solutions, select the proper equations relating $\nu_{2/11}$, $\nu_{2/12}$, $\nu_{2/21}$, and $\nu_{2/22}$ to θ_1 . Compute $\nu_{2/11}$, $\nu_{2/12}$, $\nu_{2/21}$, and $\nu_{2/22}$.

7. Calculate $\Delta M_2(\nu_1)$ with the following equation:

$$\Delta M_2(\nu_1) = M_2(\nu_{2/12}) - M_2(\nu_{2/11}) + M_2(\nu_{2/22}) - M_2(\nu_{2/21}) \quad (107)$$

where (also see Eqs (25) and (35))

$$\beta_2 = [(1-e_2)/(1+e_2)]^{1/2} \quad (108)$$

$$M_2(\nu_2) = 2 \left[\tan^{-1}[\beta_2 \tan(\nu_2/2)] - \frac{e_2 \beta_2 \sin(\nu_2/2) \cos(\nu_2/2)}{[\cos^2(\nu_2/2) - \beta_2^2 \sin^2(\nu_2/2)]} \right] \quad (109)$$

IV. Probability of Close Approach Between Satellites in Circular Orbits

Full Circular Orbit Method

The purpose of this section is to develop a method for calculating the probability of close approach between two satellites in circular orbits.

To simplify the analysis, the close approach of two satellites is redefined to be whenever the angle between the radius vectors of satellite 1 and satellite 2 (angle D) is less than or equal to some angle threshold D_{TH} , where D_{TH} is equal to the angle between the radius vector of satellite 1 and satellite 2 when the distance between the two satellites is equal to the distance threshold of close approach, d_{TH} . When a close approach is possible ($|r_1 - r_2| < d_{TH}$), the plane trigonometry law of cosines (7:196) can be used to solve for D_{TH} as a function of d_{TH} :

$$\begin{aligned} d_{TH}^2 &= r_1^2 + r_2^2 - 2r_1r_2\cos(D_{TH}) \\ D_{TH} &= \cos^{-1}\left[\frac{r_1^2 + r_2^2 - d_{TH}^2}{2r_1r_2} \right] \end{aligned} \quad (110)$$

where

- r_1 = magnitude of the radius vector of satellite 1
- r_2 = magnitude of the radius vector of satellite 2
- d_{TH} = distance threshold for close approach
- D_{TH} = angular distance threshold for close approach

Note that using an angle threshold of D_{TH} is completely equivalent to using

a distance threshold of d_{TH} so long as both satellites are in circular orbits and D_{TH} is computed using Eq (110). The use of D_{TH} simplifies the remaining mathematics, because by projecting the position of satellite 2 onto a sphere of radius r_1 , spherical trigonometry can be used to obtain the limits of integration.

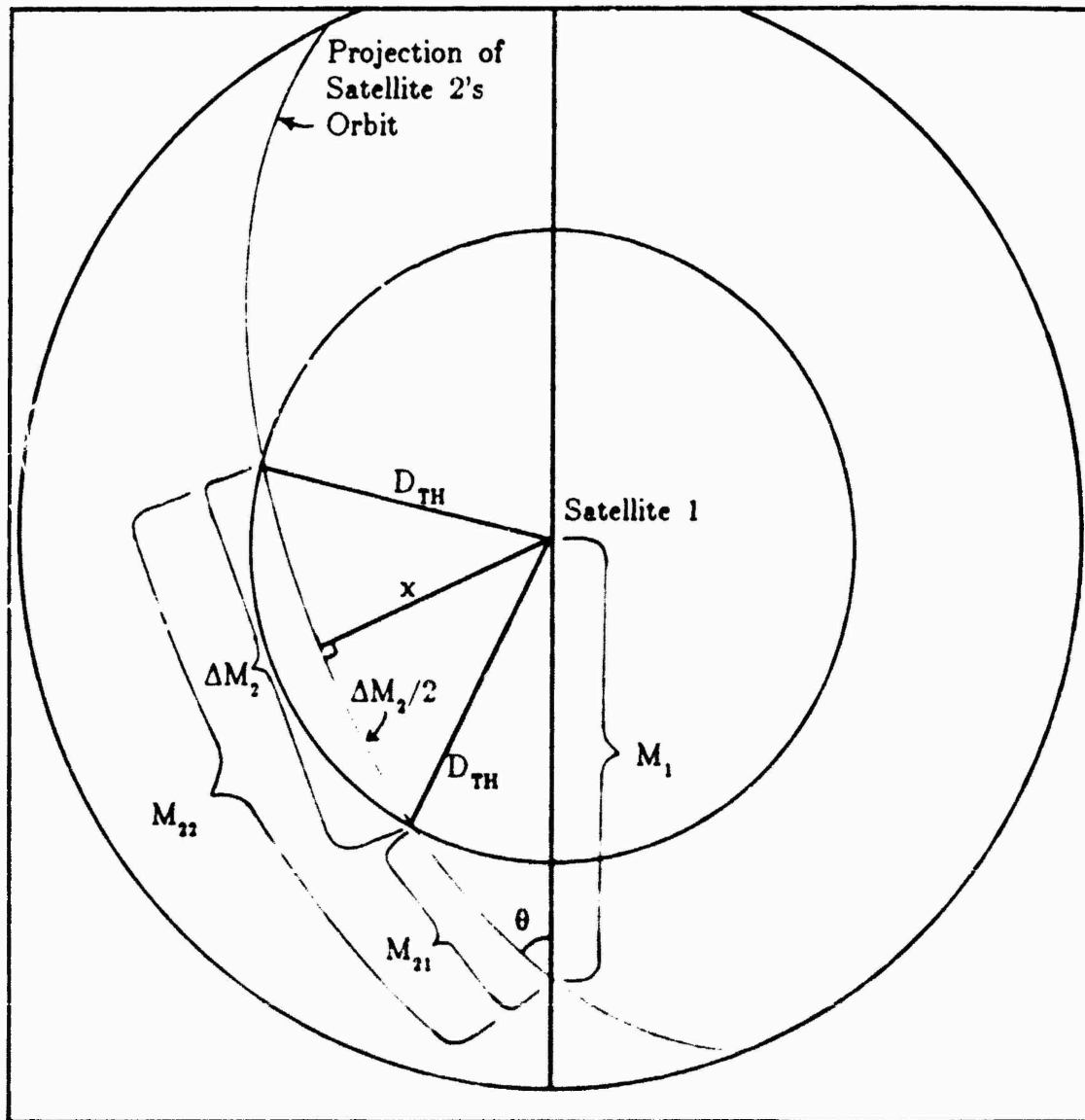


Figure 3. Spherical Geometry of Circular Orbit Close Approach

Figure 3 shows the orbital path of satellite 1, and the projection of the orbital path of satellite 2. For circular orbits, the position of zero mean anomaly is arbitrary, so for convenience, the mean anomaly of both satellites (M_1 and M_2) are assumed to be zero where the two orbital planes cross in the northern hemisphere. When the mean anomaly of satellite 1 is known, a close approach is possible whenever some portion of the projection of satellite 2's orbit comes within D_{TH} (great circle arc) of satellite 1. The probability of close approach (P_{CA}) can be determined by integrating the joint density function (see Eq (10)) over the region of M_1 and M_2 , where D is less than or equal to D_{TH} . If M_{12} and M_{11} are the unknown limits of integration over M_1 , then P_{CA} , expressed in terms of M_{12} and M_{11} , is equal to

$$P_{CA} = \int_{M_{11}}^{M_{12}} \int_{M_{21}}^{M_{22}} \frac{dM_2 dM_1}{4\pi^2} \quad (111)$$

$$P_{CA} = \int_{M_{11}}^{M_{12}} \left(\frac{M_2}{4\pi^2} \right) \Big|_{M_{21}}^{M_{22}} dM_1$$

$$\Delta M_2 = M_{22} - M_{21} \quad (112)$$

$$P_{CA} = \int_{M_{11}}^{M_{12}} \frac{\Delta M_2}{4\pi^2} dM_1 \quad (113)$$

Two more things are needed to compute P_{CA} . First, an equation for ΔM_2 as a function of M_1 is needed. Second, the integration limits over M_1

must be found for which a close approach with satellite 2 is possible. Using spherical trigonometry (7:198-200), Eqs (114) and (115) can be derived (see Figure 3):

$$\sin(M_1)/\sin(\pi/2) = \sin(x)/\sin(\theta)$$

$$\sin(x) = \sin(\theta) \sin(M_1) \quad (114)$$

$$\cos(D_{TH}) = \cos(x)\cos(\Delta M_2/2)$$

$$\Delta M_2 = 2 \cos^{-1}[\cos(D_{TH})/\cos(x)] \quad (115)$$

where

x = angle between the radius vector of satellite 1 and the orbital plane of satellite 2

Given the trigonometric relationship (7:188)

$$\cos(x) = [1 - \sin^2(x)]^{1/2}$$

Eq (114) can be used to obtain an equation for $\cos(x)$:

$$\cos(x) = [1 - \sin^2(\theta)\sin^2(M_1)]^{1/2} \quad (116)$$

Substituting Eq (116) into Eq (115), yields the final equation for ΔM_2 :

$$\Delta M_2 = 2 \cos^{-1}\{\cos(D_{TH})/[1 - \sin^2(\theta)\sin^2(M_1)]^{1/2}\} \quad (117)$$

The integration regions can be found through Eq (114), by replacing x with D_{TH} and M_1 with M , and then solving for M .

$$\sin(D_{TH}) = \sin(\theta) \sin(M)$$

$$\sin(M) = \sin(D_{TH})/\sin(\theta)$$

$$M = \sin^{-1}[\sin(D_{TH})/\sin(\theta)] \quad (118)$$

If $\sin(D_{TH})/\sin(\theta)$ is less than 1, then there are two regions. The first region is from $-M$ to M , and the second is from $\pi-M$ to $\pi+M$.

If $\sin(D_{TH})/\sin(\theta)$ is greater than 1, then the integration is from 0 to 2π , because in this case, for $0 < M_1 < 2\pi$, there is always some chance of a close approach with satellite 2.

Assuming that there are some places in satellite 1's orbit where there is no possibility of a close approach by satellite 2, then the final equation for P_{CA} is

$$P_{CA} = \int_{-M}^M \frac{\cos^{-1}\{\cos(D_{TH})/[1-\sin^2(\theta)\sin^2(M_1)]^{1/2}\}}{2\pi^{1/2}} dM_1 + \int_{\pi-M}^{\pi+M} \frac{\cos^{-1}\{\cos(D_{TH})/[1-\sin^2(\theta)\sin^2(M_1)]^{1/2}\}}{2\pi^{1/2}} dM_1 \quad (119)$$

There is no closed form solution for the equation above, so numerical integration must be used to obtain the final solution for probability of close approach between two satellites in circular orbits.

Circular Orbit Approximation Method

In the previous section no closed form solution could be found for P_{CA} between two satellites in circular orbits. However, there is a special case which does have a closed form solution for P_{CA} .

Assume that satellite 1 and satellite 2 are in circular orbits where D_{TH} is small, and θ is not small. This also implies that M_2 and M_1 (or $M_1 - \pi$) are also small. For small absolute values of u , the following approximations will be of use (with 3rd order effects and higher discarded)(7:454-457):

$$\sin(u) \cong u$$

$$\cos(u) \cong 1 - u^2/2$$

$$(1 - u^2)^{1/2} \cong 1 - u^2/2$$

$$1/(1 - u^2) \cong 1 + u^2$$

Eq (117) can be re-written as

$$\cos(\Delta M_2/2) = \cos(D_{TH}) / [1 - \sin^2(\theta) \sin^2(M_1)]^{1/2} \quad (120)$$

Substituting the small value approximations into Eq (120) and then simplifying yields

$$(1 - \Delta M_2^2/8) = (1 - D_{TH}^2/2) [1 + \sin^2(\theta) M_1^2/2]$$

$$1 - \Delta M_2^2/8 = 1 + \sin^2(\theta) M_1^2/2 - D_{TH}^2/2 - \cancel{D_{TH}^2 \sin^2(\theta) M_1^2/4} \quad \text{4th Order}$$

$$\Delta M_2^2/8 = D_{TH}^2/2 - \sin^2(\theta) M_1^2/2$$

$$\Delta M_2 = 2 [D_{TH}^2 - \sin^2(\theta) M_1^2]^{1/2}$$

$$\Delta M_2 = 2 \sin(\theta) [D_{TH}^2 / \sin^2(\theta) - M_1^2]^{1/2} \quad (121)$$

When $M_1 - \pi$ is very small, then another set of equations must be used. If $M_1 = \pi + u$, where the absolute value of u is small, then the following approximations can be used:

$$\sin(M_1) = \sin(\pi - u)$$

$$\sin(M_1) = \sin(\pi) \cos(u) + \sin(u) \cos(\pi)$$

$$\sin(M_1) = -\sin(u)$$

$$\sin(M_1) \cong -u$$

$$\sin(M_1) \cong \pi - M_1 \quad (122)$$

Substituting the small value approximations (with small $M_1 - \pi$) into Eq (120), results in

$$(1 - \Delta M_2^2 / 8) = (1 - D_{TH}^2 / 2) [1 + \sin^2(\theta) (\pi - M_1)^2 / 2]$$

$$1 - \Delta M_2^2 / 8 = 1 + \sin^2(\theta) (\pi - M_1)^2 / 2 - D_{TH}^2 / 2 - \cancel{D_{TH}^2 \sin^2(\theta) (\pi - M_1)^2 / 4} \quad \text{4th Order}$$

$$\Delta M_2^2 / 8 = D_{TH}^2 / 2 - \sin^2(\theta) (\pi - M_1)^2 / 2$$

$$\Delta M_2 = 2 [D_{TH}^2 - \sin^2(\theta) (\pi - M_1)^2]^{1/2}$$

$$\Delta M_2 = 2 \sin(\theta) [D_{TH}^2 / \sin^2(\theta) - (\pi - M_1)^2]^{1/2} \quad (123)$$

By substituting the small value approximations into Eq (118), the

limits of integration can be found:

$$M = D_{TH}/\sin(\theta) \quad (124)$$

Eqs (121) and (122) are approximations for Eq (117), and Eq (124) is an approximation for Eq (118). Substituting Eqs (121), (123), and (124) into Eq (119), results in:

$$P_{CA} = \int_{-M}^M \frac{\sin(\theta) [D_{TH}^2/\sin^2(\theta) - M_1^2]^{1/2}}{2\pi^2} dM_1 \\ + \int_{\pi-M}^{\pi+M} \frac{\sin(\theta) [D_{TH}^2/\sin^2(\theta) - (\pi-M_1)^2]^{1/2}}{2\pi^2} dM_1 \quad (125)$$

where Eq (124) is used to compute M. Given that (7:411)

$$\int (a^2 - u^2)^{1/2} du = [u(a^2 - u^2)^{1/2} + a^2 \sin^{-1}(u/a)]/2 \quad (126)$$

Eq (125) can be directly integrated by making the following substitutions into Eq (126):

For small M_1 :	For small $M_1 - \pi$:
$a = D_{TH}/\sin(\theta)$	$a = D_{TH}/\sin(\theta)$
$u = M_1$	$u = \pi - M_1$
$du = dM_1$	$du = -dM_1$

Integrating Eq (125) and simplifying results in

$$\begin{aligned}
P_{CA} &= \frac{\sin(\theta)}{4\pi^2} \left[M_1 \left(\frac{D_{TH}^2}{\sin^2(\theta)} - M_1^2 \right)^{1/2} \right. \\
&\quad \left. + \frac{D_{TH}^2}{\sin^2(\theta)} \sin^{-1} \left(\frac{M_1}{|D_{TH}/\sin(\theta)|} \right) \right] \begin{vmatrix} D_{TH}/\sin(\theta) \\ -D_{TH}/\sin(\theta) \end{vmatrix} \\
&\quad - \frac{\sin(\theta)}{4\pi^2} \left[(\pi - M_1) \left(\frac{D_{TH}^2}{\sin^2(\theta)} - (\pi - M_1)^2 \right)^{1/2} \right. \\
&\quad \left. + \frac{D_{TH}^2}{\sin^2(\theta)} \sin^{-1} \left(\frac{\pi - M_1}{|D_{TH}/\sin(\theta)|} \right) \right] \begin{vmatrix} \pi + [D_{TH}/\sin(\theta)] \\ \pi - [D_{TH}/\sin(\theta)] \end{vmatrix} \\
P_{CA} &= \frac{\sin(\theta)}{4\pi^2} \left[\frac{D_{TH}^2}{\sin^2(\theta)} \left(\frac{\pi}{2} \right) - \frac{D_{TH}^2}{\sin^2(\theta)} \left(-\frac{\pi}{2} \right) \right] \\
&\quad - \frac{\sin(\theta)}{4\pi^2} \left[\frac{D_{TH}^2}{\sin^2(\theta)} \left(-\frac{\pi}{2} \right) - \frac{D_{TH}^2}{\sin^2(\theta)} \left(\frac{\pi}{2} \right) \right] \\
P_{CA} &= \frac{\sin(\theta)}{4\pi^2} \left(\frac{2\pi D_{TH}^2}{\sin^2(\theta)} \right) \\
P_{CA} &= \frac{D_{TH}^2}{2\pi \sin(\theta)} \tag{127}
\end{aligned}$$

Equ (128) is the closed form approximate solution for P_{CA} , where satellite 1 and satellite 2 are in circular orbits, D_{TH} is small, and θ is not small.

V. Algorithm Verification

Analysis Software

To verify the algorithms from the previous chapters, three computer programs were created.

The first program is called Statistical Simulation of Probability of Close Approach or SSPCA. SSPCA queries the user for a close approach distance threshold and for a set of orbital parameters for satellite 1 and satellite 2 and then computes the probability of close approach between both satellites through a statistical simulation. The first step in this process is to select two random numbers that are uniformly distributed between 0 and 2π to represent the mean anomalies of the two satellites. For the selected mean anomalies, SSPCA calculates the distance between the two satellites. If the computed distance between the two satellites is less than or equal to the input distance threshold, then a close approach occurs. This process is repeated 100,000 times, and a count is kept of how many close approaches occurred. Eq (128) is then used to calculate the simulated probability of close approach:

$$P_s = n_{CA} / 100,000 \quad (128)$$

where P_s is the simulated probability of close approach, and n_{CA} is the number of close approaches that occurred in the simulation.

The second program is called Circular Orbit Probability of Close Approach or COPCA. COPCA first queries the user for a close approach

distance threshold and for a set of orbital parameters for two circular orbits. COPCA then computes the probability of close approach between the two satellites using both the full and the approximate circular orbit methods of calculating probability of close approach that were described in chapter IV.

The third program is called Elliptical Orbit Probability of Close Approach or EOPCA. EOPCA queries the user for a distance threshold and for a set of orbital parameters for the two satellites and then computes the probability of close approach using the elliptical orbit method of calculating probability of close approach that was described in chapter III.

The numerical integration in both the COPCA and EOPCA programs were performed using Simpson's rule with an step size of approximately $2\pi/10,000$ radians.

Statistical Simulation of Probability of Close Approach

For analysis purposes, assume that the analytically derived probability of close approach (P_{CA}) is correct. For large sample sizes and values of both P_{CA} and $(1-P_{CA})$ which are not small, the number of close approaches that occur in the statistical simulation can be approximated by a normal distribution with a mean of (5:225-226)

$$n_{CA} = n_S P_{CA} \quad (129)$$

where

- n_{CA} = mean number of close approaches in the simulation.
- n_S = number of samples or iterations in the simulation.

and a standard deviation of (5:225-226)

$$\sigma_N = [n_S P_{CA} (1 - P_{CA})]^{1/2} \quad (130)$$

Likewise, by dividing both Eqs (129) and (130) by n_S , P_S can be approximated by a normal distribution with a mean of P_{CA} , and a standard deviation of

$$\sigma_P = [P_{CA} (1 - P_{CA}) / n_S]^{1/2} \quad (131)$$

Using Eq (131), the difference between P_S (see Eq (128)) and P_{CA} can now be found in terms of standard deviations. For a perfect normal distribution, the absolute value of the difference between P_S and P_{CA} will be less than $.6745 \sigma_P$ for .5 of the simulation runs, and less than $1.96 \sigma_P$ for .95 of the simulation runs (7:578). These two thresholds are tests that can determine how well the simulated solution for probability of close approach matches the analytical solutions for P_{CA} from the COPCA and EOPCA programs.

Test Cases

Table 2 contains a list of the orbital parameters of satellite 1 and satellite 2 that are held constant through all test cases. Table 3 contains a list of the orbital parameters of satellite 1 and satellite 2 for each of the 16 test cases that were used to verify the probability of close approach algorithms described in chapters III and IV. Through all test cases, the

eccentricity of both satellites varied from 0 to .5, with the eccentricity of both satellites being the same within each test case. For convenience, the longitude of the ascending node of both satellites and the inclination of satellite 1 were set to 0° so that the inclination of satellite 2 would equal to the angular separation of the two orbital planes.

TABLE 2
Orbital Parameters Held Constant Through All Test Cases

Orbital Parameters	Satellite 1	Satellite 2
Perigee Radius	7000 km	7500 km
Inclination	0°	varies
Argument of Perigee	0°	90°
Longitude of the Ascending Node	0°	0°

Test cases 1-8 (orbits with an eccentricity of 0 or .1) were run through the SSPCA program with distance thresholds of 1000, 2000, 4000, 8000, 12000, and 20000 km. Test cases 9-16 (orbits with an eccentricity of .3 or .5) were run through the SSPCA program with distance thresholds of 4000, 8000, 12000, and 20000 km, making a total of 80 SSPCA runs.

All circular orbit test cases (test cases 1-4) were run through the COPCA program with distance thresholds of 1000, 2000, 4000, 8000, 12000, 20000 km, making a total of 24 COPCA runs.

TABLE 3
Orbital Parameters of Test Cases

Orbit Test Case #	Eccentricity of Both Satellites	Angular Separation of Orbital Planes
1	.0	0°
2	.0	0°
3	.0	30°
4	.0	60°
5	.0	90°
6	.1	0°
7	.1	30°
8	.1	60°
9	.1	90°
10	.3	0°
11	.3	30°
12	.3	60°
13	.5	0°
14	.5	30°
15	.5	60°
16	.5	90°

All test cases were run through the EOPCA program. All test cases with an eccentricity of 0 or .1 (test cases 1-8) were run with distance thresholds of 1000, 2000, 4000, 8000, 12000, and 20000 km. All test cases with an eccentricity of .3 to .5 (test cases 9-16) were run with distance thresholds of 4000, 8000, 12000, and 20000 km, making a total of 80 EOPCA runs.

Note that the distance thresholds of 1000, and 2000 km were only used in SSPCA, COPCA, and EOPCA runs involving test cases with an eccentricity of .1 or less. This is because when the distance threshold drops below 4000 km and the test case eccentricity is .3 or larger, the probability of close approach is generally too small for a statistical simulation to be of much value.

VI. Results and Discussion

Tables B-1 and B-2 in Appendix B contain the test results of the COPCA program, along with the corresponding simulation results. Table 4 shows how well the simulated P_{CA} matched the P_{CA} computed in COPCA for all runs where P_{CA} is not equal to 1. Runs with a P_{CA} of one were excluded from Table 4 because the computed P_{CA} always equaled the simulated P_{CA} when the computed P_{CA} was equal to one, and because σ_p equals zero when P_{CA} equals one. The mean error listed in Table 4 is the mean of the error (scaled by σ_p) between simulated P_{CA} and analytical P_{CA} for all COPCA runs with a P_{CA} less than one.

TABLE 4

Simulated Probability of Close Approach Versus
COPCA Probability of Close Approach

Fraction of COPCA Runs With Errors	Normal Distribution	Simulated P_{CA}
Less Than $.6745 \sigma_p$	.5000	.5000
Less Than $1.96 \sigma_p$	.9500	1.0000
Mean Error (σ_p)	.0000	.4445

Tables B-3 and B-4 in Appendix B contain the test results of the COPCA program, where the probability of close approach is computed using

both the full circular orbit method and the circular orbit approximation method described in chapter IV. For all runs with a P_{CA} less than .01, the circular orbit approximation method agreed with the full circular orbit method to within 1%. When the angle between the orbital planes is 60° or greater, the error between the full circular orbit method and the circular orbit approximation method was generally less than 3% when the computed P_{CA} was less than 5%. As expected (see Eq (127)), given the distance threshold, the error in the approximation method is inversely proportional to the angle between the two orbital planes. Similarly, given the angle between the orbital planes, the error in the approximation method is directly proportional to the distance threshold.

TABLE 5

Simulated Probability of Close Approach Versus
EOPCA Probability of Close Approach

Fraction of EOPCA Runs With Errors	Normal Distribution	Simulated P_{CA}
Less Than $.6745 \sigma_p$	.5000	.5000
Less Than $1.96 \sigma_p$	.9500	.9657
Mean Error (σ_p)	.0000	.0074

Tables B-5 through B-10 in Appendix B contain the test results of the EOPCA program, along with the corresponding simulation results. Table

5 shows how well the simulated P_{CA} matched the P_{CA} computed in EOPCA for all runs where P_{CA} is not equal to 1. As with the COPCA test results, runs with a P_{CA} of one were excluded from Table 5, because the computed P_{CA} always equaled the simulated P_{CA} when the computed P_{CA} was equal to one, and because $\sigma_p = 0$ when $P_{CA} = 1$. The mean error listed in Table 5 is the mean of the error (scaled by σ_p) between simulated P_{CA} and analytical P_{CA} for all EOPCA runs with a P_{CA} less than one.

The probability of close approach computed by both the COPCA and EOPCA programs favorably matches the simulated probability of close approach computed by the SSPCA program. However, the mean error for the 20 COPCA runs with a computed P_{CA} less than one (see Table 4) indicates that a possible bias exists between simulated P_{CA} and the analytical P_{CA} computed by COPCA. Similar biases exist within the EOPCA test results when EOPCA runs with only the same eccentricity are examined. When all 72 EOPCA runs with a computed P_{CA} less than one (see Table 5) are considered, there does not appear a bias. Given the limited number of circular orbit test cases, the small bias in the COPCA test results is not considered significant.

VII. Suggestions and Recommendations

To develop a method to calculate the probability of close approach between two satellites, two major assumptions were made. First, all orbital elements, except for true (or mean) anomaly, were assumed to be constant over time. Second, the mean anomalies of both satellites were assumed to be independent random variables that are uniformly distributed between 0 and 2π . While these assumptions greatly simplified the derivation of probability of close approach, they also limited its usefulness.

Three follow-up studies are recommended. The goal of the first study would be to develop methods to calculate probability of close approach between two satellites in elliptical orbits, where there are linear perturbations to the argument of perigee and the longitude of the ascending node of both satellites. In this case, both argument of perigee and longitude of the ascending node would be treated as linear functions of time. The goal of the second study would be to develop methods to calculate probability of close approach between two satellites in elliptical orbits where the duration of the specified time interval is too short, or the ratio of the orbital periods of the two satellites are such that the mean anomalies of both satellites are not independent. The goal of the final study would be to find ways to reduce the computational expense involved in calculating probability of close approach. In this paper, numerical integration was used to directly calculate the probability of close approach between two satellites. This approach can be used to accurately calculate the probability of close approach between any two satellites with eccentricities less than 1.0, but it can be computationally

expensive. For two satellites with an eccentricity less than .3, it is possible that a series approximation for probability of close approach could be found that would offer acceptable precision, and at much less computational expense than methods that use numerical integration. For two satellites in circular orbits, even simpler series approximations for probability of close approach could be possible. Both forms of series approximations for probability of close approach merit further investigation.

Appendix A

Types of Close Approach

Type 1 Close Approach

A type 1 close approach occurs when

$$1. \quad y_1 \leq 0$$

$$2. \quad pd_{TH} \leq B_1$$

$$3. \quad pd_{TH} \leq B_2$$

When there are 0 valid solutions, then

$$\nu_{2/11} = 0$$

$$\nu_{2/21} = 0$$

$$\nu_{2/12} = 0$$

$$\nu_{2/22} = 0$$

When there are 2 valid solutions, then

$$\nu_{2/11} = 2\pi - \theta_2$$

$$\nu_{2/21} = 0$$

$$\nu_{2/12} = 2\pi - \theta_1$$

$$\nu_{2/22} = 0$$

When there are 4 valid solutions, then

$$\nu_{2/11} = \theta_2$$

$$\nu_{2/21} = 2\pi - \theta_4$$

$$\nu_{2/12} = \theta_3$$

$$\nu_{2/22} = 2\pi - \theta_1$$

Type 2 Close Approach

A type 2 close approach occurs when

1. $y_1 \leq 0$
2. $pd_{TH} \leq B_1$
3. $pd_{TH} > B_2$

It is not possible to have 0 valid solutions in a type 2 close approach.

When there are 2 valid solutions, then

$$\begin{array}{ll} \nu_{2/11} = \theta_2 & \nu_{2/21} = 0 \\ \nu_{2/12} = 2\pi - \theta_1 & \nu_{2/22} = 0 \end{array}$$

When there are 4 valid solutions, then

$$\begin{array}{ll} \nu_{2/11} = \theta_2 & \nu_{2/21} = \theta_4 \\ \nu_{2/12} = \theta_3 & \nu_{2/22} = 2\pi - \theta_1 \end{array}$$

Type 3 Close Approach

A type 3 close approach occurs when

1. $y_1 \leq 0$
2. $pd_{TH} > B_1$
3. $pd_{TH} \leq B_2$

It is not possible to have 0 valid solutions in a type 3 close approach.

When there are 2 valid solutions, then

$$\begin{array}{ll} \nu_{2/11} = -\theta_2 & \nu_{2/21} = 0 \\ \nu_{2/12} = \theta_1 & \nu_{2/22} = 0 \end{array}$$

When there are 4 valid solutions, then

$$\begin{array}{ll} \nu_{2/11} = -\theta_4 & \nu_{2/21} = \theta_2 \\ \nu_{2/12} = \theta_1 & \nu_{2/22} = \theta_4 \end{array}$$

Type 4 Close Approach

A type 4 close approach occurs when

1. $y_1 \leq 0$
2. $pd_{TH} > B_1$
3. $pd_{TH} > B_2$

When there are 0 valid solutions, then

$$\begin{array}{ll} \nu_{2/11} = 0 & \nu_{2/21} = 0 \\ \nu_{2/12} = 2\pi & \nu_{2/22} = 0 \end{array}$$

When there are 2 valid solutions, then

$$\begin{array}{ll} \nu_{2/11} = \theta_2 - 2\pi & \nu_{2/21} = 0 \\ \nu_{2/12} = \theta_1 & \nu_{2/22} = 0 \end{array}$$

When there are 4 valid solutions, then

$$\begin{array}{ll} \nu_{2/11} = \theta_4 - 2\pi & \nu_{3/21} = \theta_2 \\ \nu_{2/12} = \theta_1 & \nu_{2/22} = \theta_3 \end{array}$$

Type 5 Close Approach

A type 5 close approach occurs when

1. $y_1 > 0$
2. $pd_{TH} \leq B_1$
3. $pd_{TH} \leq B_2$

When there are 0 valid solutions, then

$$\nu_{2/11} = 0$$

$$\nu_{2/21} = 0$$

$$\nu_{2/12} = 0$$

$$\nu_{2/22} = 0$$

When there are 2 valid solutions, then

$$\nu_{2/11} = \theta_1$$

$$\nu_{2/21} = 0$$

$$\nu_{2/12} = \theta_2$$

$$\nu_{2/22} = 0$$

When there are 4 valid solutions, then

$$\nu_{2/11} = \theta_1$$

$$\nu_{2/21} = 2\pi - \theta_1$$

$$\nu_{2/12} = \theta_2$$

$$\nu_{2/22} = 2\pi - \theta_2$$

Type 6 Close Approach

A type 6 close approach occurs when

1. $y_1 > 0$
2. $pd_{TH} \leq B_1$
3. $pd_{TH} > B_2$

It is not possible to have 0 valid solutions in a type 6 close approach.

When there are 2 valid solutions, then

$$\begin{array}{ll} \nu_{2/11} = \theta_1 & \nu_{2/21} = 0 \\ \nu_{2/12} = 2\pi - \theta_2 & \nu_{2/22} = 0 \end{array}$$

When there are 4 valid solutions, then

$$\begin{array}{ll} \nu_{2/11} = \theta_1 & \nu_{2/21} = 2\pi - \theta_2 \\ \nu_{2/12} = \pi - \theta_4 & \nu_{2/22} = 2\pi - \theta_2 \end{array}$$

Type 7 Close Approach

A type 7 close approach occurs when

1. $y_1 > 0$
2. $pd_{TH} > B_1$
3. $pd_{TH} \leq B_2$

It is not possible to have 0 valid solutions in a type 7 close approach.

When there are 2 valid solutions, then

$$\begin{array}{ll} \nu_{2/11} = -\theta_1 & \nu_{2/21} = 0 \\ \nu_{2/12} = \theta_2 & \nu_{2/22} = 0 \end{array}$$

When there are 4 valid solutions, then

$$\begin{array}{ll} \nu_{2/11} = -\theta_1 & \nu_{2/21} = 2\pi - \theta_1 \\ \nu_{2/12} = \theta_2 & \nu_{2/22} = 2\pi - \theta_2 \end{array}$$

Type 8 Close Approach

A type 8 close approach occurs when

1. $y_1 > 0$
2. $pd_{TH} > E_1$
3. $pd_{TH} > E_2$

When there are 0 valid solutions, then

$$\nu_{2/11} = 0$$

$$\nu_{2/21} = 0$$

$$\nu_{2/12} = 2\pi$$

$$\nu_{2/22} = 0$$

When there are 2 valid solutions, then

$$\nu_{2/11} = -\theta_1$$

$$\nu_{2/21} = 0$$

$$\nu_{2/12} = 2\pi - \theta_2$$

$$\nu_{2/22} = 0$$

When there are 4 valid solutions, then

$$\nu_{2/11} = -\theta_1$$

$$\nu_{2/21} = 2\pi - \theta_1$$

$$\nu_{2/12} = 2\pi - \theta_2$$

$$\nu_{2/22} = 2\pi - \theta_2$$

Appendix B

Test Results

TABLE B-1

COPCA Test Results For Distance Thresholds Less Than 4000 km

Distance Threshold (km)	Angle Between Orbital Planes	COPCA P_{CA}	Simulation σ_p	Simulated P_{CA}	Error (σ_p)	Error (%)
1000	0°	.038068	.000605	.038510	.7306	1.16
2000	0°	.085327	.000883	.086210	1.0000	1.03
1000	30°	.004580	.000214	.004670	.4206	1.97
2000	30°	.023638	.000480	.024010	.7750	1.57
1000	60°	.002632	.000162	.002790	.9753	6.00
2000	60°	.013287	.000362	.013250	-.1022	-.28
1000	90°	.002278	.000151	.002370	.6093	4.04
2000	90°	.011471	.000337	.011100	-1.1009	-3.23

TABLE B-2

COPCA Test Results For Distance Thresholds Not Less Than 4000 km

Distance Threshold (km)	Angle Between Orbital Planes	COPCA P_{CA}	Simulation σ_p	Simulated P_{CA}	Error (σ_p)	Error (%)
4000	0°	.176602	.001206	.178280	1.3914	.95
8000	0°	.371487	.001528	.373250	1.1538	.47
12000	0°	.620316	.001535	.620220	-.0625	-.02
20000	0°	1.000000	.000000	1.000000	-	.00
4000	30°	.128739	.001059	.129460	.6808	.56
8000	30°	.361425	.001519	.363150	1.1356	.48
12000	30°	.629654	.001527	.629590	-.0419	-.01
20000	30°	1.000000	.000000	1.000000	-	.00
4000	60°	.058214	.000740	.058130	-.1135	-.14
8000	60°	.315640	.001470	.316940	.8840	.41
12000	60°	.671401	.001485	.671640	.1609	.04
20000	60°	1.000000	.000000	1.000000	-	.00
4000	90°	.049658	.000687	.050050	.5691	.79
8000	90°	.233877	.001339	.233390	-.3637	-.21
12000	90°	.754522	.001361	.755730	.8876	.16
20000	90°	1.000000	.000000	1.000000	-	.00

TABLE B-3

COPCA Test Results Using Circular Orbit Approximation
For Distance Thresholds Less Than 4000 km

Distance Threshold (km)	Angle Between Orbital Planes	COPCA P_{CA}	COPCA Approximate P_{CA}	Error (%)
1000	30°	.004580	.004553	-.59
2000	30°	.023638	.022873	-3.24
1000	60°	.002632	.002629	-.11
2000	60°	.013287	.013206	-.61
1000	90°	.002278	.002276	-.09
2000	90°	.011471	.011437	-.30

TABLE B-4

COPCA Test Results Using Circular Orbit Approximation
For Distance Thresholds Not Less Than 4000 km

Distance Threshold (km)	Angle Between Orbital Planes	COPCA P_{CA}	COPCA Approximate P_{CA}	Error (%)
4000	30°	.128739	.097981	-23.89
8000	30°	.361425	.433547	19.95
12000	30°	.629654	—	—
20000	30°	1.000000	—	—
4000	60°	.058214	.056569	-2.83
8000	60°	.315640	.250309	-20.70
12000	60°	.671401	.697936	3.95
20000	60°	1.000000	—	—
4000	90°	.049658	.048990	-1.35
8000	90°	.233877	.216774	-7.31
12000	90°	.754522	.604430	-19.89
20000	90°	1.000000	—	—

TABLE B-5

EOPCA Test Results For Circular Orbits With
Distance Thresholds Less Than 4000 km

Distance Threshold (km)	Angle Between Orbital Planes	COPCA P_{CA}	Simulation σ_p	Simulated P_{CA}	Error (σ_p)	Error (%)
1000	0°	.038008	.000605	.038510	.7306	1.16
2000	0°	.085327	.000883	.086210	1.0000	1.03
1000	30°	.004580	.000214	.004670	.4206	1.97
2000	30°	.023638	.000480	.024010	.7750	1.57
1000	60°	.002632	.000162	.002790	.9753	6.00
2000	60°	.013287	.000362	.013250	-.1022	-.28
1000	90°	.002278	.000151	.002370	.6093	4.04
2000	90°	.011471	.000337	.011100	-1.1009	-3.23

TABLE B-6

EOPCA Test Results For Circular Orbits With
Distance Thresholds Not Less Than 4000 km

Distance Threshold (km)	Angle Between Orbital Planes	COPCA P_{CA}	Simulation σ_p	Simulated P_{CA}	Error (σ_p)	Error (%)
4000	0°	.176602	.001206	.178280	1.3914	.95
8000	0°	.371487	.001528	.373250	1.1538	.47
12000	0°	.620316	.001535	.620220	-.0625	-.02
20000	0°	1.000000	.000000	1.000000	-	.00
4000	30°	.128739	.001059	.129460	.6808	.56
8000	30°	.361425	.001519	.363150	1.1356	.48
12000	30°	.629654	.001527	.629590	-.0419	-.01
20000	30°	1.000000	.000000	1.000000	-	.00
4000	60°	.058214	.000740	.058130	-.1135	-.14
8000	60°	.315640	.001470	.316940	.8840	.41
12000	60°	.671401	.001485	.671640	.1609	.04
20000	60°	1.000000	.000000	1.000000	-	.00
4000	90°	.049659	.000687	.050050	.5691	.79
8000	90°	.233877	.001339	.233390	-.3637	-.21
12000	90°	.754522	.001361	.755730	.8876	.16
20000	90°	1.000000	.000000	1.000000	-	.00

TABLE B-7

EOPCA Test Results For Eccentricity of .1 With
Distance Thresholds Less Than 4000 km

Distance Threshold (km)	Angle Between Orbital Planes	COPCA P_{CA}	Simulation σ_p	Simulated P_{CA}	Error (σ_p)	Error (%)
1000	0°	.021034	.000454	.021520	1.0705	2.31
2000	0°	.067767	.000795	.068180	.5195	.61
1000	30°	.002449	.000156	.002030	-2.6859	-17.11
2000	30°	.016417	.000402	.016320	-.2413	-.59
1000	60°	.001414	.000119	.001170	-2.0504	-17.26
2000	60°	.009292	.000303	.008960	-1.0957	-3.57
1000	90°	.001225	.000111	.000950	-2.4775	-22.45
2000	90°	.008027	.000282	.007530	-1.7624	-6.19

TABLE B-8

EOPCA Test Results For Eccentricity of .1 With
Distance Thresholds Not Less Than 4000 km

Distance Threshold (km)	Angle Between Orbital Planes	COPCA P_{CA}	Simulation σ_p	Simulated P_{CA}	Error (σ_p)	Error (%)
4000	0°	.154566	.001143	.155130	.4934	.36
8000	0°	.327955	.001485	.329990	1.3704	.62
12000	0°	.533222	.001578	.533210	-.0076	-.00
20000	0°	1.000000	.000000	1.000000	-	.00
4000	30°	.091240	.000911	.091410	.1866	.19
8000	30°	.313524	.001467	.315190	1.1357	.53
12000	30°	.535690	.001577	.535530	-.1015	-.03
20000	30°	1.000000	.000000	1.000000	-	.00
4000	60°	.045010	.000656	.045230	.3354	.49
8000	60°	.234065	.001339	.233720	-.2577	-.15
12000	60°	.546098	.001574	.547590	.9479	.27
20000	60°	1.000000	.000000	1.000000	-	.00
4000	90°	.038555	.000609	.038810	.4187	.66
8000	90°	.179613	.001214	.178870	-.6120	-.41
12000	90°	.581394	.001560	.582690	.8308	.22
20000	90°	1.000000	.000000	1.000000	-	.00

TABLE B-9

EOPCA Test Results For Eccentricity of .3

Distance Threshold (km)	Angle Between Orbital Planes	COPCA P_{CA}	Simulation σ_p	Simulated P_{CA}	Error (σ_p)	Error (%)
4000	0°	.067156	.000791	.066410	-.9431	-1.11
8000	0°	.226341	.001323	.226230	-.0839	-.05
12000	0°	.374049	.001530	.375610	1.0203	.42
20000	0°	.762220	.001346	.762770	.4086	.07
4000	30°	.027733	.000519	.027410	-.6224	-1.16
8000	30°	.192271	.001246	.191640	-.5064	-.33
12000	30°	.363538	.001521	.364400	.5667	.24
20000	30°	.773937	.001323	.774410	.3575	.06
4000	60°	.013120	.000360	.013530	1.1389	3.13
8000	60°	.109096	.000986	.108750	-.3509	-.32
12000	60°	.302533	.001453	.301440	-.7522	-.36
20000	60°	.819091	.001217	.820710	1.3303	1.98
4000	90°	.010933	.000329	.011360	1.2979	3.91
8000	90°	.090037	.000905	.090000	-.0409	-.04
12000	90°	.269083	.001402	.267690	-.9936	-.52
20000	90°	.849413	.001131	.851050	1.4474	1.93

TABLE B-10

EOPCA Test Results For Eccentricity of .5

Distance Threshold (km)	Angle Between Orbital Planes	COPCA P_{CA}	Simulation σ_p	Simulated P_{CA}	Error (σ_p)	Error (%)
4000	0°	.015125	.000386	.014750	-.9715	-2.48
8000	0°	.066378	.000787	.065430	-1.2046	-1.43
12000	0°	.185335	.001229	.185330	-.0041	-.00
20000	0°	.434822	.001568	.435010	.1199	.04
4000	30°	.002293	.000151	.002270	-.1523	-1.00
8000	30°	.042547	.000638	.041490	-1.6567	-2.48
12000	30°	.150654	.001131	.150520	-.1185	-.09
20000	30°	.427634	.001564	.427660	.0166	.01
4000	60°	.000682	.000083	.000610	-.8675	-10.56
8000	60°	.010386	.000321	.010230	-.4860	-1.50
12000	60°	.108902	.000985	.108390	-.5198	-.47
20000	60°	.390383	.001543	.389900	-.3130	-.12
4000	90°	.000434	.000066	.000430	-.0606	-.92
8000	90°	.008820	.000296	.008590	-.7770	-2.61
12000	90°	.085002	.000882	.085920	1.0408	1.08
20000	90°	.341475	.001500	.338930	-1.6967	-.75

Appendix C
Program Listings

```

C-----C
C
C   SSPCA - Statistical Simulation of the Probability   C
C               of Close Approach                     C
C-----C
C   INTERFACE TO SUBROUTINE RNDINI(C)
C   END
C   INTERFACE TO REAL*8 FUNCTION RANDOM(C)
C   END
C-----C
C
C   PROGRAM SSPCA
C   CHARACTER YESNO*4
C   LOGICAL*4 DEBUG, OBSCUR
C   INTEGER*4 I, J, K, ONE, TWO, CA, OCA
C   REAL*8 DX, DY, DZ, DR
C   REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1(4)
C   REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2(4)
C   REAL*8 DTH, RANDOM, RANGE, DBLE
C   REAL*8 HALFPI, PI, TWOPI, DEGRAD, RADDEG
C   REAL*8 ER, DU, TU
C   COMMON /ADMIN/  DEBUG
C   COMMON /ASTRO/  ER, DU, TU
C   COMMON /COMPAR/ DX, DY, DZ, DR
C   COMMON /CONST/  HALFPI, PI, TWOPI, DEGRAD, RADDEG
C   COMMON /SAT1/   M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1
C   COMMON /SAT2/   M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2
C   PARAMETER (ONE=1, TWO=2)
C   ER = 6378.145D0
C   DU = 6378.145D0
C   TU = 806.8118744D0
C   PI=3.141592653589793D0
C   HALFPI=PI/2.0D0
C   TWOPI=2.0D0*PI
C   DEGRAD=PI/180.0D0
C   RADDEG=180.0D0/PI
C   *-----*
C   *   Initialize Random Number Generator   *
C   *-----*
C   CALL RNDINI
C   *-----*
C   *   Input Orbital Elements of Sat 1     *
C   *-----*
C   WRITE(*,1000) ONE
C   READ (*,1010) A1
C   WRITE(*,1020) ONE
C   READ (*,1010) E1

```

```

WRITE(*,1030) ONE
READ (*,1010) INC1
WRITE(*,1040) ONE
READ (*,1010) ARGPA1
WRITE(*,1050) ONE
READ (*,1010) LONAN1
C  *-----*
C  *   Input Orbital Elements of Sat 2   *
C  *-----*
WRITE(*,1000) TWO
READ (*,1010) A2
WRITE(*,1020) TWO
READ (*,1010) E2
WRITE(*,1030) TWO
READ (*,1010) INC2
WRITE(*,1040) TWO
READ (*,1010) ARGPA2
WRITE(*,1050) TWO
READ (*,1010) LONAN2
C  *-----*
C  *   Input Simulation Limits           *
C  *-----*
WRITE(*,1060)
READ (*,1010) DTH
WRITE(*,1070)
READ (*,1080) ITER
WRITE(*,1090)
READ (*,1100) YESNO
IF (YESNO(1:1).EQ.'Y'.OR.YESNO(1:1).EQ.'y') THEN
  DEBUG = .TRUE.
ELSE
  DEBUG = .FALSE.
ENDIF
C  *-----*
C  *   Convert the input angles from deg *
C  *   to rad                           *
C  *-----*
INC1 = INC1 * DEGRAD
ARGPA1 = ARGPA1 * DEGRAD
LONAN1 = LONAN1 * DEGRAD
INC2 = INC2 * DEGRAD
ARGPA2 = ARGPA2 * DEGRAD
LONAN2 = LONAN2 * DEGRAD
C  *-----*
C  *   Clear the two event counters:     *
C  *   OCA = Obscured Close Approach    *
C  *   CA  = Close Approach              *

```

```

C      *-----*
      OCA = 0
      CA = 0
C      *-----*
C      * Start Simulation      *
C      *-----*
      DO 100 I=1,ITER
        M1 = TWOPI * RANDOM()
        M2 = TWOPI * RANDOM()
        DR = RANGE()
        IF (DEBUG) WRITE (*,1110) M1, M2, DR
        IF (DR .LE. DTH) THEN
          IF (OBSCUR()) THEN
            OCA = OCA + 1
          ELSE
            CA = CA + 1
          ENDIF
        ENDIF
      ENDIF
100    CONTINUE
      WRITE(*,1120) A1, A2, E1, E2, INC1*RADDEG, INC2*RADDEG,
1          ARGPA1*RADDEG, ARGPA2*RADDEG, LONAN1*RADDEG,
2          LONAN2*RADDEG, DTH, ITER,
3          OCA, DBLE(OCA)/DBLE(ITER),
4          CA, DBLE(CA)/DBLE(ITER)
C      *-----*
1000  FORMAT (/ ' Input the semi-major axis of Sat ',I1,
1      ' in km: '\)
1010  FORMAT (F16.12)
1020  FORMAT (' Input the eccentricity of Sat ',I1,': '\)
1030  FORMAT (' Input the inclination of Sat ',I1,
1      ' in degrees: '\)
1040  FORMAT (' Input the argument of perapsis of Sat ',I1,
1      ' in degrees: '\)
1050  FORMAT (' Input the longitude of the ascending node ',
1      ' of Sat ',I1,' in degrees: '\)
1060  FORMAT (/ ' Input distance threshold in km: '\)
1070  FORMAT (' Input the desired number of iterations ',
1      '(7 digits max): '\)
1080  FORMAT (I7)
1090  FORMAT (' Run SSPCA in DEBUG mode (Y/N)? '\)
1100  FORMAT (A4)
1110  FORMAT (/ ' M1=',F12.10,4X,'M2=',F12.10,4X,'RANGE=',F12.6)
1120  FORMAT (/T21,'Sat 1',T41,'Sat 2',T16,'-----',
1      T36,'-----'// ' a (km)',T16,F16.10,4X,
2      F16.10// ' e',T16,F16.10,4X,F16.10//
3      ' inc (deg)',T16,F16.10,4X,F16.10//
4      ' argument of',T16,F16.10,4X,F16.10// ' perigee (deg)'//

```

```

5 ' long of asc',T16,F16.10,4X,F16.10/' node (deg)'///
6 ' Distance Threshold for Close Approach: ',
7 F16.10,' km'///
6 T48,'Fraction'/' Iterations ',I7,T46,'-----'//
7 ' Close',T14,I7,'--Obscured by the Earth--',F12.10/
8 ' Approaches'/T14,I7,'-----Unobserved-----',
9 F12.10//)

```

END

```

C *-----*
C *
C *          Function OBSCUR()
C *
C *-----*

```

LOGICAL*4 FUNCTION OBSCUR()

```

C *-----*

```

LOGICAL*4 DEBUG

INTEGER*4 I, J, K

REAL*8 NDRANG, DX, DY, DZ, DR, ODT, RADICAL

REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1(4)

REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2(4)

REAL*8 HALFPI, PI, TWOPI, DEGRAD, RADDEG

REAL*8 ER, DU, TU

COMMON /ADMIN/ DEBUG

COMMON /ASTRO/ ER, DU, TU

COMMON /COMPAR/ DX, DY, DZ, DR

COMMON /CONST/ HALFPI, PI, TWOPI, DEGRAD, RADDEG

COMMON /SAT1/ M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1

COMMON /SAT2/ M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2

```

C *-----*

```

NDRANG = -1D0 * (X1(1)*DX + X1(2)*DY + X1(3)*DZ)

NDRANG = NDRANG / X1(4) / DR

IF (DABS(NDRANG) .GT. 1D0) NDRANG = DSIGN(1D0, NDRANG)

NDRANG = DACOS(NDRANG)

IF (DEBUG) WRITE(*,1000) NDRANG*RADDEG

IF (NDRANG.GT.HALFPI .OR. X1(4)*DSIN(NDRANG).GE.ER) THEN

OBSCUR = .FALSE.

ELSE

ODTH = X1(4) * DCOS(NDRANG)

RADICAL = ODT - ODT - X1(4)*X1(4) + ER*ER

IF (RADICAL .GE. 0D0) THEN

ODTH = ODT - DSQRT(RADICAL)

ELSE

IF (DSQRT(DABS(RADICAL)) .GT. 5D-11) THEN

WRITE (*,10!0) ODT.

1 -1D0 * DSQRT(-RADICAL)

ENDIF

ENDIF

```

      IF (DEBUG) WRITE (*,1020) DR, ODTN.
1    DSIGN(DSQRT(DABS(RADICAL)),RADICAL)
      IF ( DR .GT. ODTN) THEN
        OBSCUR = .TRUE.
      ELSE
        OBSCUR = .FALSE.
      ENDIF
    ENDIF
  RETURN
1000 FORMAT (// ' The Nadir Angle from Sat 1 to Sat 2 ',
1          'is (deg):',F16.10)
1010 FORMAT (// ' Error! Negative Radical.  ODTN=',F16.10,
2  4X, 'RADICAL=', 1P,D12.5,0P)
1020 FORMAT (// ' DR=',F16.10,4X, 'ODTN=',F16.10,4X, 'RADICAL=',
3  1P,D12.5,0P)
END
C  *-----*
C  *                                     *
C  *           Function RANGE()         *
C  *                                     *
C  *-----*
REAL*8 FUNCTION RANGE()
C  *-----*
LOGICAL*4 DEBUG
INTEGER*4 I, J, K
REAL*8 NU, XP(3), R(3,2), DX, DY, DZ, DR
REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1(4)
REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2(4)
REAL*8 HALFPI,PI,TWOPI,DEGRAD,RADDEG
REAL*8 ER, DU, TU
COMMON /ADMIN/  DEBUG
COMMON /ASTRO/  ER, DU, TU
COMMON /COMPAR/ DX, DY, DZ, DR
COMMON /CONST/  HALFPI, PI, TWOPI, DEGRAD, RADDEG
COMMON /SAT1/   M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1
COMMON /SAT2/   M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2
C  *-----*
C  *   Compute True Anomaly of Sat 1   *
C  *-----*
NU1 = NU( M1, E1)
C  *-----*
C  *   Compute the Radius of Sat 1   *
C  *-----*
X1(4) = A1 * (1D0 - E1*E1) / (1D0 + E1*DCOS(NU1))
C  *-----*
C  *   Compute the position of Sat 1 in   *
C  *   Perifocal Coordinate Frame       *

```

```

C      *-----*
      XP(1) = X1(4)*DCOS(NU1)
      XP(2) = X1(4)*DSIN(NU1)
      XP(3) = 0D0
C      *-----*
C      * Compute the elements of the      *
C      * transformation matrix to transform *
C      * from the Perifocal Coordinate     *
C      * Frame to the Geocentric Equatorial *
C      * Frame.                             *
C      *-----*
      R(1,1) = DCOS(LONAN1)*DCOS(ARGPA1)-
1          DSIN(LONAN1)*DSIN(ARGPA1)*DCOS(INC1)
      R(1,2) = - DCOS(LONAN1)*DSIN(ARGPA1) -
1          DSIN(LONAN1)*DCOS(ARGPA1)*DCOS(INC1)
      R(2,1) = DSIN(LONAN1)*DCOS(ARGPA1)+
1          DCOS(LONAN1)*DSIN(ARGPA1)*DCOS(INC1)
      R(2,2) = - DSIN(LONAN1)*DSIN(ARGPA1)+
1          DCOS(LONAN1)*DCOS(ARGPA1)*DCOS(INC1)
      R(3,1) = DSIN(ARGPA1)*DSIN(INC1)
      R(3,2) = DCOS(ARGPA1)*DSIN(INC1)
      IF (DEBUG) WRITE(*,1000) ((R(I,J),J=1,2),I=1,3)
C      *-----*
C      * Compute the position of Sat 1 in  *
C      * the Geocentric Equatorial Frame.  *
C      *-----*
      X1(1) = R(1,1)*XP(1) + R(1,2)*XP(2)
      X1(2) = R(2,1)*XP(1) + R(2,2)*XP(2)
      X1(3) = R(3,1)*XP(1) + R(3,2)*XP(2)
C      *-----*
C      * Compute the True Anomaly of Sat 2 *
C      *-----*
      NU2 = NU( M2, E2)
C      *-----*
C      * Compute the Radius of Sat 2      *
C      *-----*
      X2(4) = A2 * (1D0 - E2*E2) / (1D0 + E2*DCOS(NU2))
C      *-----*
C      * Compute the position of Sat 1 in  *
C      * Perifocal Coordinate Frame        *
C      *-----*
      XP(1) = X2(4)*DCOS(NU2)
      XP(2) = X2(4)*DSIN(NU2)
      XP(3) = 0D0
C      *-----*
C      * Compute the elements of the      *
C      * transformation matrix to transform *

```

```

C      * from the Perifocal Coordinate      *
C      * Frame to the Geocentric Equatorial *
C      * Frame.                             *
C      *-----*
R(1,1) = DCOS(LONAN2)*DCOS(ARGPA2) -
1      DSIN(LONAN2)*DSIN(ARGPA2)*DCOS(INC2)
R(1,2) = - DCOS(LONAN2)*DSIN(ARGPA2) -
1      DSIN(LONAN2)*DCOS(ARGPA2)*DCOS(INC2)
R(2,1) = DSIN(LONAN2)*DCOS(ARGPA2) +
1      DCOS(LONAN2)*DSIN(ARGPA2)*DCOS(INC2)
R(2,2) = - DSIN(LONAN2)*DSIN(ARGPA2) +
1      DCOS(LONAN2)*DCOS(ARGPA2)*DCOS(INC2)
R(3,1) = DSIN(ARGPA2)*DSIN(INC2)
R(3,2) = DCOS(ARGPA2)*DSIN(INC2)
IF (DEBUG) WRITE(*,1010) ((R(I,J),J=1,2),I=1,3)
C      *-----*
C      * Compute the position of Sat 2 in   *
C      * the Geocentric Equatorial Frame.   *
C      *-----*
X2(1) = R(1,1)*XP(1) + R(1,2)*XP(2)
X2(2) = R(2,1)*XP(1) + R(2,2)*XP(2)
X2(3) = R(3,1)*XP(1) + R(3,2)*XP(2)
DX = X2(1) - X1(1)
DY = X2(2) - X1(2)
DZ = X2(3) - X1(3)
DR = DSQRT(DX*DX + DY*DY + DZ*DZ)
IF (DEBUG) THEN
  WRITE(*,1020) NU1*RADDEG, NU2*RADDEG
  WRITE(*,1030) (X2(I),I=1,4),(X1(I),I=1,4),
1              DX, DY, DZ, DR
ENDIF
RANGE = DR
RETURN
1000 FORMAT (///' Sat 1 R-matrix to transform from Perifocal ',
1          'to Geocentric Equatorial Frame.'///
2          (' ',F15.12,4X,F15.12)//)
1010 FORMAT (///' Sat 2 R-matrix to transform from Perifocal ',
1          'to Geocentric Equatorial Frame.'///
2          (' ',F15.12,4X,F15.12)//)
1020 FORMAT (' Sat 1 True Anomaly: ',F16.10,4X,
1          ' Sat 2 True Anomaly: ',F16.10//)
1030 FORMAT (/T16,'X',T34,'Y',T52,'Z',T66,'Magnitude'/
1          T9,'-----',
2          '-----'/
3          ' Sat 2: ',F16.9,2X,F16.9,2X,F16.9,2X,F16.9//
4          ' Sat 1: ',F16.9,2X,F16.9,2X,F16.9,2X,F16.9/
5          T9,'-----')

```

```

6          '-----'/'
7          ' Delta: ',F16.9,2X,F16.9,2X,F16.9,2X,F16.9//)
      END
C      *-----*
C      *
C      *          Function NU
C      *
C      *-----*
      REAL*8 FUNCTION NU( M, E )
      REAL*8 M, E
C      *-----*
      LOGICAL*4 DEBUG
      INTEGER*4 I, J, K
      REAL*8 EANOM, EANOM1, EANDOT, ERROR
      REAL*8 HALFPI,PI,TWOPI,DEGRAD,RADDEG
      REAL*8 ER, DU, TU
      COMMON /ADMIN/ DEBUG
      COMMON /ASTRO/ ER, DU, TU
      COMMON /CONST/ HALFPI, PI, TWOPI, DEGRAD, RADDEG
C      *-----*
      IF ( E .EQ. 0D0 ) THEN
          NU = M
          IF (DEBUG) WRITE (*,1000) NU
          RETURN
      ENDIF
      EANOM = M
      IF (DEBUG) WRITE (*,1100) M, E, EANOM
      I = 0
100    EANOM1 = EANOM
      I = I + 1
      EANDOT = 1D0 - E * DCOS(EANOM1)
      EANOM = EANOM1 + (M - EANOM1 + E*SIN(EANOM1))/EANDOT
      ERROR = DABS((EANOM1 - EANOM)/EANOM1)
      IF (DEBUG) WRITE (*,1200) I, EANOM1, EANOM, EANDOT, ERROR
      IF (ERROR .GT. 5D-11) GOTO 100
      NU = 2D0 * DATAN( DSQRT((1D0 + E)/(1D0 - E)) *
1    DTAN(5D-1 * EANOM))
      IF (NU.LT.0) NU = TWOPI + NU
      RETURN
1000  FORMAT (/ ' This is a circular orbit, so NU = M.  NU=',
1      F12.10)
1100  FORMAT (//T4,'Mean',T35,'Eccentric'/
1      T3,'Anomaly',T18,'Eccentricity',T36,'Anomaly'/
2      T2,'-----'/'
3      1X,3(F12.10, 4X)///
4      T18,'Old',T35,'New'/T16,'Eccentric',T33,'Eccentric'/
5      ' Iteration',T17,'Anomaly',T34,'Anomaly',T50,'dM/dE',

```

```

6  T66,'Error'/' -----
7  '-----')
1200 FORMAT (2X, I7, 5X, F12.10, 4X, F12.10, 4X, 1P, D12.5, 4X,
1  D12.5, 0P)
END

```

```

#include <stdio.h>
/*-----*/
/*
/*      rndini - this routine initializes the random number    */
/*              generator.                                     */
/*-----*/
rndini()
{
    srand(1);
}
/*-----*/
/*
/*      random - this routine returns a double precision      */
/*              uniform distributed random variable between    */
/*              0 and 1.                                       */
/*-----*/
double *random()
{
    int rand();
    double r;
    r = rand();
    r *= 32768.0;
    r += rand();
    r /= 32768.0 * 32768.0;
    return (&r);
}

```

```

C-----C
C
C COPCA - Circular Orbit Probability of Close Approach C
C
C-----C

PROGRAM COPCA
CHARACTER YESNO*4
LOGICAL*4 DEBUG
INTEGER*4 ONE, TWO
REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1(4)
REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2(4)
REAL*8 ANGPTH, APCA, A2PCA, APXPCA, CADTH
REAL*8 DTH, INCR
REAL*8 M, OPSEP, PCA, PDCA
REAL*8 PDCAM1, PDCAM2, PDCAM3, SADTH, STHETA, THETA
REAL*8 HALFPI, PI, TWOPI, DEGRAD, RADDEG
REAL*8 ER, DU, TU
COMMON /ADMIN/ DEBUG
COMMON /ASTRO/ ER, DU, TU
COMMON /CONST/ HALFPI, PI, TWOPI, DEGRAD, RADDEG
COMMON /SAT1/ M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1
COMMON /SAT2/ M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2
COMMON /TRANS/ SADTH, CADTH, STHETA
PARAMETER (ONE=1, TWO=2)
ER = 6378.145D0
DU = 6378.145D0
TU = 806.8118744D0
PI=DACOS(-1D0)
HALFPI=PI/2D0
TWOPI=2D0*PI
DEGRAD=PI/180D0
RADDEG=180D0/PI
C
C *-----*
C * Input Orbital Elements of Sat 1 *
C *-----*
WRITE(*,1000) ONE
READ (*,1010) A1
WRITE(*,1020) ONE
READ (*,1010) E1
WRITE(*,1030) ONE
READ (*,1010) INC1
WRITE(*,1040) ONE
READ (*,1010) ARGPA1
WRITE(*,1050) ONE
READ (*,1010) LONAN1
C
C *-----*
C * Input Orbital Elements of Sat 2 *

```

```

C      *-----*
      WRITE(*,1000) TWO
      READ (*,1010) A2
      WRITE(*,1020) TWO
      READ (*,1010) E2
      WRITE(*,1030) TWO
      READ (*,1010) INC2
      WRITE(*,1040) TWO
      READ (*,1010) ARGPA2
      WRITE(*,1050) TWO
      READ (*,1010) LONAN2

C      *-----*
C      *   Input Simulation Limits   *
C      *-----*
      WRITE(*,1060)
      READ (*,1010) DTH
      WRITE(*,1070)
      READ (*,1080) ITER
      INCR = TWOPI / DBLE(ITER)
      WRITE(*,1090)
      READ (*,1100) YESNO
      IF (YESNO(1:1).EQ.'Y'.OR.YESNO(1:1).EQ.'y') THEN
        DEBUG = .TRUE.
      ELSE
        DEBUG = .FALSE.
      ENDIF

C      *-----*
C      *   Convert the input angles from deg   *
C      *   to rad                               *
C      *-----*
      INC1 = INC1 * DEGRAD
      ARGPA1 = ARGPA1 * DEGRAD
      LONAN1 = LONAN1 * DEGRAD
      INC2 = INC2 * DEGRAD
      ARGPA2 = ARGPA2 * DEGRAD
      LONAN2 = LONAN2 * DEGRAD

C      *-----*
C      *   Initialize Probability of Close   *
C      *   Approach                           *
C      *-----*
      PCA = 0D0

C      *-----*
C      *   Compute Angular Separation Between *
C      *   The Orbital Planes of Sat 1 and   *
C      *   Sat 2                               *
C      *-----*
      THETA = OPSEP()

```

```

STHETA = DSIN(THETA)
C  *-----*
C  * Compute Angular Distance Threshold *
C  *-----*
ANGDTH = (A1*A1 + A2*A2 - DTH*DTH)/(2D0*A1*A2)
IF (DABS(ANGDTH) .GT. 1D0) THEN
  IF (ANGDTH .LT. -1D0) THEN
    ANGDTH = PI
  ELSE
    ANGDTH = 0D0
  ENDIF
ELSE
  ANGDTH = DACOS(ANGDTH)
ENDIF
SADTH = DSIN(ANGDTH)
CADTH = DCOS(ANGDTH)
C  *-----*
C  * Compute Approximate Probability of *
C  * Close Approach (Using Both Methods)*
C  *-----*
APCA = APXPCA(ANGDTH)
C  *-----*
C  * Start Numerical Integration      *
C  *-----*
IF (ANGDTH .GE. THETA) THEN
  N2 = HALFPI
ELSE
  N2 = DASIN(SADTH / STHETA)
ENDIF
N1 = -N2
IF (DEBUG) THEN
  WRITE(*,1110) N1, N2
  N2 = N1 + 1D1 * INCR
ENDIF
M = N1
C  *-----*
C  * Adjust step size to make the      *
C  * integration range an integer      *
C  * number of step size's.            *
C  *-----*
IF ( N2 .GT. N1) INCR = (N2-N1)/DBLE(IDMINT((N2-N1)/INCR))
N2 = N2 - INCR/2D0
PDCAM1 = PDCA(M)
100 PDCAM2 = PDCA(M+.5D0*INCR)
PDCAM3 = PDCA(M+INCR)
PCA = PCA + INCR/6D0 * (PDCAM1 + 4D0*PDCAM2 + PDCAM3)
IF (DEBUG) WRITE(*,1120) M, PDCAM1, PDCAM2, PDCAM3, PCA

```

```

PDCAM1 = PDCAM3
M = M + INCR
IF (M .LE. M2) GOTO 100
PCA = 2D0 * PCA
WRITE(*,1130) A1, A2, F1, E2, INC1*RADDEG, INC2*RADDEG,
1          ARGPA1*RADDEG, ARGPA2*RADDEG, LONAN1*RADDEG,
2          LONAN2*RADDEG, INCR*1D6, DTH, ANG DTH*RADDEG,
3          THETA*RADDEG, APCA, PCA
C  *-----*
1000 FORMAT (/ ' Input the semi-major axis of Sat ',I1,
1       ' in km: '\)
1010 FORMAT (F16.12)
1020 FORMAT ( ' Input the eccentricity of Sat ',I1,': '\)
1030 FORMAT ( ' Input the inclination of Sat ',I1,
1       ' in degrees: '\)
1040 FORMAT ( ' Input the argument of perapsis of Sat ',I1,
1       ' in degrees: '\)
1050 FORMAT ( ' Input the longitude of the ascending node ',
1       ' of Sat ',I1,' in degrees: '\)
1060 FORMAT (/ ' Input distance threshold in km: '\)
1070 FORMAT ( ' Input the desired number of iterations per radian ',
1       ' (7 digits max): '\)
1080 FORMAT (I7)
1090 FORMAT ( ' Run COPCA in DEBUG mode (Y/N)? '\)
1100 FORMAT (A4)
1110 FORMAT (// ' M1=',F13.10,4X,'M2=',F13.10//)
1120 FORMAT ( ' M=',F16.10// ' PDCAM1=',F16.10// ' PDCAM2=',F16.10//
1       ' PDCAM3=',F16.10// ' PCA  =',F16.10//)
1130 FORMAT (/T21,'Sat 1',T41,'Sat 2',T16,'-----',
1       T36,'-----'// ' a (km)',T16,F16.10,4X,
2       F16.10// ' e',T16,F16.10,4X,F16.10//
3       ' inc (deg)',T16,F16.10,4X,F16.10//
4       ' argument of',T16,F16.10,4X,F16.10// ' perigee (deg)'//
5       ' long of asc',T16,F16.10,4X,F16.10// ' node (deg)'///
6       ' Mean Anomaly Iteration Step Size      : ',
7       F16.10,' micro-radians'//
8       ' Distance Threshold for Close Approach: ',
9       F16.10,' km'//
A       ' Angular Threshold for Close Approach : ',
B       F16.10,' deg'//
C       ' Angular Separation of Orbital Planes : ',
D       F16.10,' deg'//
E       ' Approximate Probability of Close Approach: ',
F       F9.6//
G       ' Computed Probability of Close Approach : ',F9.6//
END
C  *-----*

```

```

C      *
C      *          Function OPSEP()
C      *
C      *-----*
C      REAL*8 FUNCTION OPSEP
C      *-----*
C      LOGICAL*4 DEBUG
C      REAL*8 DLONAN, THETA
C      REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1(4)
C      REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2(4)
C      REAL*8 HALFPI, PI, TWOPI, DEGRAD, RADDEG
C      REAL*8 ER, DU, TU
C      COMMON /ADMIN/  DEBUG
C      COMMON /ASTRO/  ER, DU, TU
C      COMMON /CONST/  HALFPI, PI, TWOPI, DEGRAD, RADDEG
C      COMMON /SAT1/   M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1
C      COMMON /SAT2/   M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2
C      *-----*
C      DLONAN = LONAN2 - LONAN1
C      IF (DABS(DLONAN) .GT. PI)
C      1  DLONAN = DLONAN - DSIGN(TWOPI, DLONAN)
C      THETA = DACOS(DCOS(INC1)*DCOS(INC2)
C      1  + DSIN(INC1)*DSIN(INC2)*DCOS(DLONAN))
C      IF (THETA .GT. HALFPI) THETA = PI - THETA
C      OPSEP = THETA
C      RETURN
C      END
C      *-----*
C      *
C      *          Function PDCA()
C      *
C      *-----*
C      REAL*8 FUNCTION PDCA(N)
C      *-----*
C      LOGICAL*4 DEBUG
C      REAL*8 ARG, M, SM, STHETA, SADTH, CADTH
C      REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1(4)
C      REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2(4)
C      REAL*8 HALFPI, PI, TWOPI, DEGRAD, RADDEG
C      REAL*8 ER, DU, TU
C      COMMON /ADMIN/  DEBUG
C      COMMON /ASTRO/  ER, DU, TU
C      COMMON /CONST/  HALFPI, PI, TWOPI, DEGRAD, RADDEG
C      COMMON /SAT1/   M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1
C      COMMON /SAT2/   M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2
C      COMMON /TRANS/  SADTH, CADTH, STHETA
C      *-----*

```

```

SM = DSIN( M )
ARG = 1D0 - STHETA*STHETA * SM*SM
IF (ARG .EQ. 0D0) THEN
IF (CADTH .GE. 0D0) THEN
PDCA = 0D0
RETURN
ELSE
PDCA = 5D-1 / PI
RETURN
ENDIF
ENDIF
ARG = CADTH / DSQRT(ARG)
IF (DABS(ARG) .GT. 1D0) THEN
IF (ARG .GT. 1D0) THEN
PDCA = 0D0
RETURN
ELSE
PDCA = 5D-1 / PI
RETURN
ENDIF
ENDIF
PDCA = DACOS(ARG)/(2D0 * PI * PI)
RETURN
END

```

C
C
C
C
C

```

*-----*
*          Function APXPCA()          *
*-----*

```

```

REAL*8 FUNCTION APXPCA(ANGDTH)
REAL*8 ANGDTH

```

C

```

*-----*
LOGICAL*4 DEBUG
REAL*8 STHETA, SADTH, CADTH
REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1(4)
REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2(4)
REAL*8 HALFPI, PI, TWOPI, DEGRAD, RADDEC
REAL*8 ER, DU, TU
COMMON /ADMIN/  DEBUG
COMMON /ASTRO/  ER, DU, TU
COMMON /CONST/  HALFPI, PI, TWOPI, DEGRAD, RADDEC
COMMON /SAT1/   M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1
COMMON /SAT2/   M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2
COMMON /TRANS/  SADTH, CADTH, STHETA

```

C

```

*-----*
IF (STHETA .LT. 1D-1) THEN
APXPCA = -1D0

```

```
ELSE
  APXPCA = ANGDTH * ANGDTH / (2D0 * PI * STHETA)
  IF (APXPCA .GT. 1D0) APXPCA = -1D0
ENDIF
RETURN
END
```

```

C-----C
C
C      EOPCA - Elliptical Orbit Probability of Close Approach
C
C-----C

```

```

PROGRAM EOPCA
CHARACTER YESNO*4
LOGICAL*4 DEBUG
INTEGER*4 ONE, TWO
REAL*8 M
REAL*8 BOUND1, BOUND2, DTH, DSQR, PDTH, PR, NU11, NU12,
1 NU21, NU22
REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LOMAN1, X1(4)
REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LOMAN2, X2(4)
REAL*8 SINC1, CINC1, SNU1, CNU1, SARGP1, CARGP1,
1 SLMAN1, CLMAN1, BETA1
REAL*8 SINC2, CINC2, SNU2, CNU2, SARGP2, CARGP2,
1 SLMAN2, CLMAN2, BETA2
REAL*8 R11, R12, R13, R21, R22, R23, R31, R32, R33
REAL*8 S11, S12, S13, S21, S22, S23, S31, S32, S33
REAL*8 APCA
REAL*8 INCR, LIMIT, NU
REAL*8 PCA, PDCA
REAL*8 PDCAN1, PDCAN2, PDCAN3
REAL*8 HALFPI, PI, TWOPI, DEGRAD, RADDEG
REAL*8 ER, DU, TU
COMMON /ADMIN/ DEBUG
COMMON /ASTRO/ ER, DU, TU
COMMON /BOUND/ BOUND1, BOUND2, DTH, DSQR, PDTH, PR, NU11,
1 NU12, NU21, NU22
COMMON /CONST/ HALFPI, PI, TWOPI, DEGRAD, RADDEG
COMMON /SAT1/ M1, A1, E1, INC1, NU1, ARGPA1, LOMAN1, X1,
1 SINC1, CINC1, SNU1, CNU1, SARGP1, CARGP1,
2 SLMAN1, CLMAN1, BETA1
COMMON /SAT2/ M2, A2, E2, INC2, NU2, ARGPA2, LOMAN2, X2,
1 SINC2, CINC2, SNU2, CNU2, SARGP2, CARGP2,
2 SLMAN2, CLMAN2, BETA2
COMMON /TRANS/ R11, R12, R13, R21, R22, R23, R31, R32,
1 R33, S11, S12, S13, S21, S22, S23, S31, S32, S33
PARAMETER (ONE=1, TWO=2)
ER = 6378.145D0
DU = 6378.145D0
TU = 806.8118744D0
PI = DACOS(-1D0)
HALFPI = PI/2D0
TWOPI = 2D0*PI
DEGRAD = PI/180D0

```

```

RADDEG = 180D0/PI
C  *-----*
C  *   Input Orbital Elements of Sat 1   *
C  *-----*
WRITE(*,1000) ONE
READ (*,1010) A1
WRITE(*,1020) ONE
READ (*,1010) E1
WRITE(*,1030) ONE
READ (*,1010) INC1
WRITE(*,1040) ONE
READ (*,1010) ARGPA1
WRITE(*,1050) ONE
READ (*,1010) LONAN1
C  *-----*
C  *   Input Orbital Elements of Sat 2   *
C  *-----*
WRITE(*,1000) TWO
READ (*,1010) A2
WRITE(*,1020) TWO
READ (*,1010) E2
WRITE(*,1030) TWO
READ (*,1010) INC2
WRITE(*,1040) TWO
READ (*,1010) ARGPA2
WRITE(*,1050) TWO
READ (*,1010) LONAN2
C  *-----*
C  *   Input Simulation Limits           *
C  *-----*
WRITE(*,1060)
READ (*,1010) DTH
WRITE(*,1070)
READ (*,1080) ITER
INCR = TWOPI / DBLE(ITER)
WRITE(*,1090)
READ (*,1100) YESNO
IF (YESNO(1:1).EQ.'Y'.OR.YESNO(1:1).EQ.'y') THEN
  DEBUG = .TRUE.
ELSE
  DEBUG = .FALSE.
ENDIF
C  *-----*
C  *   Convert the input angles from deg *
C  *   to rad                           *
C  *-----*
INC1 = INC1 * DEGRAD

```

```

ARGPA1 = ARGPA1 * DEGRAD
LONAN1 = LONAN1 * DEGRAD
INC2 = INC2 * DEGRAD
ARGPA2 = ARGPA2 * DEGRAD
LONAN2 = LONAN2 * DEGRAD

C *-----*
C * Compute the elements of the *
C * transformation matrix to transform *
C * from the Perifocal Coordinate *
C * Frame to the Geocentric Equatorial *
C * Frame. *
C *-----*

SARGP1 = DSIN(ARGPA1)
CARGP1 = DCOS(ARGPA1)
SINC1 = DSIN(INC1)
CINC1 = DCOS(INC1)
SLNAN1 = DSIN(LONAN1)
CLNAN1 = DCOS(LONAN1)
BETA1 = DSQRT((1D0-E1)/(1D0+E1))

C *-----*
R11 = CLNAN1*CARGP1-SLNAN1*SARGP1*CINC1
R12 = -CLNAN1*SARGP1-SLNAN1*CARGP1*CINC1
R13 = 0D0
R21 = SLNAN1*CARGP1+CLNAN1*SARGP1*CINC1
R22 = -SLNAN1*SARGP1+CLNAN1*CARGP1*CINC1
R23 = 0D0
R31 = SARGP1*SINC1
R32 = CARGP1*SINC1
R33 = 0D0
IF (DEBUG) WRITE(*,1110) BETA1,R11,R12,R21,R22,R31,R32

C *-----*
C * Compute the elements of the *
C * transformation matrix to transform *
C * from the Perifocal Coordinate *
C * Frame to the Geocentric Equatorial *
C * Frame. *
C *-----*

SARGP2 = DSIN(ARGPA2)
CARGP2 = DCOS(ARGPA2)
SINC2 = DSIN(INC2)
CINC2 = DCOS(INC2)
SLNAN2 = DSIN(LONAN2)
CLNAN2 = DCOS(LONAN2)
BETA2 = DSQRT((1D0-E2)/(1D0+E2))

C *-----*
S11 = CLNAN2*CARGP2-SLNAN2*SARGP2*CINC2
S12 = SLNAN2*CARGP2+CLNAN2*SARGP2*CINC2

```

```

S13 = SARGP2*SINC2
S21 = -CLNAN2*SARGP2-SLNAN2*CARGP2*CINC2
S22 = -SLNAN2*SARGP2+CLNAN2*CARGP2*CINC2
S23 = CARGP2*SINC2
S31 = SLNAN2*SINC2
S32 = -CLNAN2*SINC2
S33 = CINC2
IF (DEBUG) WRITE(*,1120) BETA2,S11, S12, S13, S21, S22,
1  S23, S31, S32, S33
C  *-----*
C  * Initialize Probability of Close *
C  * Approach *
C  *-----*
PCA = 0D0
C  *-----*
C  * Start Numerical Integration *
C  *-----*
NU = -PI
LIMIT = PI - INCR/2D0
PDCAM1 = PDCA(NU)
100 PDCAM2 = PDCA(NU+.5D0*INCR)
PDCAM3 = PDCA(NU+INCR)
PCA = PCA + INCR/6D0 * (PDCAM1 + 4D0*PDCAM2 + PDCAM3)
NU = NU + INCR
IF (DEBUG) WRITE(*,1130) NU, PDCAM1, PDCAM2, PDCAM3, PCA
PDCAM1 = PDCAM3
DEBUG = .FALSE.
IF (NU .LT. LIMIT) GOTO 100
WRITE(*,1140) A1, A2, E1, E2, INC1*RADDEG, INC2*RADDEG,
1 ARGPA1*RADDEG, ARGPA2*RADDEG, LONAN1*RADDEG,
2 LONAN2*RADDEG, INCR*1D6, DTH, PCA
C  *-----*
1000 FORMAT (/ ' Input the semi-major axis of Sat ',I1,
1 ' in km: '\)
1010 FORMAT (F16.12)
1020 FORMAT ( ' Input the eccentricity of Sat ',I1,': '\)
1030 FORMAT ( ' Input the inclination of Sat ',I1,
1 ' in degrees: '\)
1040 FORMAT ( ' Input the argument of perapsis of Sat ',I1,
1 ' in degrees: '\)
1050 FORMAT ( ' Input the longitude of the ascending node ',
1 'of Sat ',I1,' in degrees: '\)
1060 FORMAT (/ ' Input distance threshold in km: '\)
1070 FORMAT ( ' Input the desired number of iterations per ',
1 'radian (7 digits max): '\)
1080 FORMAT (I7)
1090 FORMAT ( ' Run EOPCA in DEBUG mode (Y/N)? '\)

```

```

1100  FORMAT (A4)
1110  FORMAT (///' Beta1 = ',F15.12//
1      ' Sat 1 R-matrix to transform from Perifocal to ',
2      ' Geocentric Equatorial Frame.'//
3      (' ',F15.12,4X,F15.12,4X,' 0.0')/)
1120  FORMAT (///' Beta2 = ',F15.12//
1      ' Sat 2 S-matrix to transform from Geocentric ',
2      ' Equatorial Frame to Sat 2's Perifocal Frame.'//
3      (' ',F15.12,4X,F15.12,4X,F15.12)/)
1130  FORMAT (' NU      =',F16.10/' PDCAM1=',F16.10/' PDCAM2=',
1      F16.10/' PDCAM3=',F16.10/' PCA      =',F16.10///)
1140  FORMAT (/T21,'Sat 1',T41,'Sat 2'/T16,'-----',
1      T36,'-----'///' a (km)',T16,F16.10,4X,
2      F16.10///' e',T16,F16.10,4X,F16.10//
3      ' inc (deg)',T16,F16.10,4X,F16.10//
4      ' argument of',T16,F16.10,4X,F16.10/' perigee (deg)'//
5      ' long of asc',T16,F16.10,4X,F16.10/' node (deg)'///
6      ' True Anomaly Iteration Step Size      : ',
7      F16.10,' micro-radians'//
8      ' Distance Threshold for Close Approach: ',
9      F16.10,' km'//
A      ' Computed Probability of Close Approach : ',F9.6//)
      END
C      *-----*
C      *
C      *      Subroutine ORBBND      *
C      *
C      *-----*
C      SUBROUTINE ORBBND()
C      *-----*
      LOGICAL*4 DEBUG, INSIDE
      INTEGER*4 I, N, NORDER
      REAL*8 A, B, C, D, E, F, G, H, K, SAT2P
      REAL*8 P1, P2, P3, P4, P5, P(4)
      REAL*8 SOLUTION(4), RANK(4)
      REAL*8 ANGLE, DIST, TEMP1, TEMP2
      REAL*8 NU, X, Y, Z
      REAL*8 BOUND1, BOUND2, DTH, DSQR, PDTH, PR, NU11, NU12,
1      NU21, NU22
      REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1(4),
1      SINC1, CINC1, SNU1, CNU1, SARGP1, CARGP1,
2      SLNAN1, CLNAN1, BETA1
      REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2(4),
1      SINC2, CINC2, SNU2, CNU2, SARGP2, CARGP2,
2      SLNAN2, CLNAN2, BETA2
      REAL*8 R11, R12, R13, R21, R22, R23, R31, R32, R33,
1      S11, S12, S13, S21, S22, S23, S31, S32, S33

```

```

REAL*8 HALFPI,PI,TWOPI,DEGRAD,RADDEG
REAL*8 ER, DU, TU
COMPLEX*16 ANSWER(4)
COMMON /ADMIN/  DEBUG
COMMON /ASTRO/  ER, DU, TU
COMMON /BOUND/  BOUND1, BOUND2, DTH, DSQR, PDTH, PR, NU11,
1  NU12, NU21, NU22
COMMON /CONST/  HALFPI, PI, TWOPI, DEGRAD, RADDEG
COMMON /SAT1/   M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1,
1  SINC1, CINC1, SNU1, CNU1, SARGP1, CARGP1,
2  SLNAN1, CLNAN1, BETA1
COMMON /SAT2/   M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2,
1  SINC2, CINC2, SNU2, CNU2, SARGP2, CARGP2,
2  SLNAN2, CLNAN2, BETA2
COMMON /TRANS/  R11, R12, R13, R21, R22, R23, R31, R32,
1  R33, S11, S12, S13, S21, S22, S23, S31, S32, S33
C  *-----*
C  * Initialize Arrays to zero      *
C  *-----*
DO 100 I=1,4
  ANSWER(I)=(0D0,0D0)
  P(I)=0D0
  RANK(I)=0D0
  SOLUTION(I)=0D0
100 CONTINUE
C  *-----*
C  * Find the terms needed to compute *
C  * the polynomial coefficients of the *
C  * polynomial used to find NU11 thru *
C  * NU22. Use 2nd order polynomial *
C  * solution if X1(1) or X1(2) are *
C  * very small. *
C  *-----*
SAT2P = A2*(1D0 - E2*E2)
A = E2 * ((DSQR - PR*PR)*E2 + 2D0*X1(1)*SAT2P)
B = 2D0*((DSQR - PR*PR)*E2 + X1(1)*SAT2P)
C = DSQR - PR*PR - SAT2P*SAT2P

K = 4D0*X1(2)*X1(2)*SAT2P*SAT2P
IF (DEBUG) WRITE(*,900) DSQR, PR, E2, SAT2P, X1(1), A, B,
1  C, K
C  *-----*
IF (DABS(X1(2)).GT.1D-8 .AND. PR.NE.DABS(X1(1))) THEN
  P1 = A*A + K*E2*E2
  P2 = 2D0*(A*B + K*E2)
  P3 = B*B + 2D0*A*C + K*(1D0-E2*E2)
  P4 = 2D0*(B*C - K*E2)

```

```

P5 = C*C - K
C      *-----*
C      * Divide the polynomial by the      *
C      * highest order coefficient          *
C      *-----*
NORDER = 0
IF (P1.NE.0D0) THEN
C      *-----*
C      * The polynomial is 4th order      *
C      *-----*
      NORDER = 4
      P(1) = P2/P1
      P(2) = P3/P1
      P(3) = P4/P1
      P(4) = P5/P1
      CALL QUARTIC( P, ANSWER)
ELSE IF (P2.NE.0D0) THEN
C      *-----*
C      * The polynomial is 3rd order      *
C      *-----*
      NORDER = 3
      P(1) = P3/P2
      P(2) = P4/P2
      P(3) = P5/P2
      CALL CUBIC( P, ANSWER)
ELSE IF (P3.NE.0D0) THEN
C      *-----*
C      * The polynomial is 2nd order      *
C      *-----*
      NORDER = 2
      P(1) = P4/P3
      P(2) = P5/P3
      CALL QUADRATIC( P, ANSWER)
ENDIF
ELSE
  IF (A.NE.0D0) THEN
    NORDER = 4
    P(1) = B/A
    P(2) = C/A
    CALL QUADRATIC( P, ANSWER)
    ANSWER(3) = ANSWER(1)
    ANSWER(4) = ANSWER(2)
  ELSE
    NORDER = 2
    ANSWER(1) = -C / B
    ANSWER(2) = ANSWER(1)
  ENDIF
ENDIF

```

```

ENDIF
IF (DEBUG) THEN
  WRITE(*,1000) NORDER, P1, P2, P3, P4, P5
  DO 200 I=1,4
    WRITE(*,1010) I, P(I), I, DREAL(ANSWER(I)),
1    DIMAG(ANSWER(I))
200  CONTINUE
ENDIF
N = 0
DO 300 I=1,NORDER
  IF (DIMAG(ANSWER(I)).EQ.0D0) THEN
    IF ((DABS(DREAL(ANSWER(I)))-1D0).LE.1D-10) THEN
      N = N + 1
      SOLUTION(N) = DREAL(ANSWER(I))
      IF (DABS(SOLUTION(N)).GT.1D0) THEN
        IF (SOLUTION(N).GT.0D0) THEN
          SOLUTION(N) = 0D0
        ELSE
          SOLUTION(N) = PI
        ENDIF
      ELSE
        SOLUTION(N) = DACOS(SOLUTION(N))
      ENDIF
    ENDIF
  ENDIF
300 CONTINUE
IF (DEBUG) WRITE(*,1020) N, (I,RADDEG*SOLUTION(I),I=1,N)
IF (N.GE.2) THEN
  RANK(1) = SOLUTION(1)
  DO 500 I=2,N
    TEMP2 = SOLUTION(I)
    DO 400 J=1,I-1
      IF (TEMP2.LT.RANK(J)) THEN
        TEMP1 = RANK(J)
        RANK(J) = TEMP2
        TEMP2 = TEMP1
      ENDIF
    CONTINUE
400  RANK(I) = TEMP2
500  CONTINUE
ENDIF
IF (DEBUG) WRITE(*,1030) N, (I,RADDEG*RANK(I),I=1,N)
C  *-----*
C  * Find out if the projection of the *
C  * Sat 1 vector into Sat 2's *
C  * perifocal plane is within Sat 2's *
C  * orbit. *

```

```

C      *-----*
      ANGLE = DATAN2(X1(2),X1(1))
      DIST  = A2*(1D0-E2*E2)/(1D0+E2*DCOS(ANGLE))
      IF (DIST.GT.PR) THEN
        INSIDE = .TRUE.
      ELSE
        INSIDE = .FALSE.
      ENDIF
      IF (DEBUG) WRITE(*,1040) RADDEG*ANGLE, DIST, INSIDE
C      *-----*
C      * If the projection of Sat 1's      *
C      * vector into Sat 2's perifocal    *
C      * plane is within Sat 2's orbit,    *
C      * then use the first set of quadrant *
C      * checks. If outside of Sat 2's     *
C      * orbit, then use the second set of *
C      * quadrant checks.                  *
C      *-----*
      NU11 = 0D0
      NU12 = 0D0
      NU21 = 0D0
      NU22 = 0D0
      IF (INSIDE) THEN
        IF (X1(2).LT.0D0) THEN
          IF (PDTH.GT.BOUND1) THEN
            IF (PDTH.GT.BOUND2) THEN
              IF (N.EQ.0) THEN
                NU11 = 0D0
                NU12 = TWOPI
              ELSE IF (N.EQ.2) THEN
                NU11 = RANK(2) - TWOPI
                NU12 = RANK(1)
              ELSE IF (N.EQ.4) THEN
                NU11 = RANK(4) - TWOPI
                NU12 = RANK(1)
                NU21 = RANK(2)
                NU22 = RANK(3)
              ELSE
                WRITE(*,1050) N
              ENDIF
            ELSE
              IF (N.EQ.2) THEN
                NU11 = -RANK(2)
                NU12 = RANK(1)
              ELSE IF (N.EQ.4) THEN
                NU11 = -RANK(4)
                NU12 = RANK(1)
              ENDIF
            ENDIF
          ELSE
            IF (N.EQ.2) THEN
              NU11 = -RANK(2)
              NU12 = RANK(1)
            ELSE IF (N.EQ.4) THEN
              NU11 = -RANK(4)
              NU12 = RANK(1)
            ENDIF
          ENDIF
        ELSE
          IF (N.EQ.2) THEN
            NU11 = -RANK(2)
            NU12 = RANK(1)
          ELSE IF (N.EQ.4) THEN
            NU11 = -RANK(4)
            NU12 = RANK(1)
          ENDIF
        ENDIF
      ELSE
        IF (N.EQ.2) THEN
          NU11 = -RANK(2)
          NU12 = RANK(1)
        ELSE IF (N.EQ.4) THEN
          NU11 = -RANK(4)
          NU12 = RANK(1)
        ENDIF
      ENDIF

```

```

        NU21 = RANK(2)
        NU22 = RANK(3)
    ELSE
        WRITE(*,1060) N
    ENDIF
ENDIF
ELSE
    IF (PDTH.GT.BOUND2) THEN
        IF (N.EQ.2) THEN
            NU11 = RANK(2)
            NU12 = TWOPI - RANK(1)
        ELSE IF (N.EQ.4) THEN
            NU11 = RANK(2)
            NU12 = RANK(3)
            NU21 = RANK(4)
            NU22 = TWOPI - RANK(1)
        ELSE
            WRITE(*,1070) N
        ENDIF
    ELSE
        IF (N.EQ.2) THEN
            NU11 = TWOPI - RANK(2)
            NU12 = TWOPI - RANK(1)
        ELSE IF (N.EQ.4) THEN
            NU11 = RANK(2)
            NU12 = RANK(3)
            NU21 = TWOPI - RANK(4)
            NU22 = TWOPI - RANK(1)
        ELSE IF (N.NE.0) THEN
            WRITE(*,1080) N
        ENDIF
    ENDIF
ENDIF
ELSE
    IF (PDTH.GT.BOUND1) THEN
        IF (PDTH.GT.BOUND2) THEN
            IF (N.EQ.0) THEN
                NU11 = 0D0
                NU12 = TWOPI
            ELSE IF (N.EQ.2) THEN
                NU11 = - RANK(1)
                NU12 = TWOPI - RANK(2)
            ELSE IF (N.EQ.4) THEN
                NU11 = - RANK(1)
                NU12 = TWOPI - RANK(4)
                NU21 = TWOPI - RANK(3)
                NU22 = TWOPI - RANK(2)
            
```

```

ELSE
  WRITE(*,1090) N
ENDIF
ELSE
  IF (N.EQ.2) THEN
    NU11 = - RANK(1)
    NU12 = RANK(2)
  ELSE IF (N.EQ.4) THEN
    NU11 = - RANK(1)
    NU12 = RANK(4)
    NU21 = TWOPI - RANK(3)
    NU22 = TWOPI - RANK(2)
  ELSE
    WRITE(*,1100) N
  ENDIF
ENDIF
ELSE
  IF (PDTH.GT.BOUND2) THEN
    IF (N.EQ.2) THEN
      NU11 = RANK(1)
      NU12 = TWOPI - RANK(2)
    ELSE IF (N.EQ.4) THEN
      NU11 = RANK(1)
      NU12 = TWOPI - RANK(4)
      NU21 = TWOPI - RANK(3)
      NU22 = TWOPI - RANK(2)
    ELSE
      WRITE(*,1110) N
    ENDIF
  ELSE
    IF (N.EQ.2) THEN
      NU11 = RANK(1)
      NU12 = RANK(2)
    ELSE IF (N.EQ.4) THEN
      NU11 = RANK(1)
      NU12 = RANK(4)
      NU21 = TWOPI - RANK(3)
      NU22 = TWOPI - RANK(2)
    ELSE IF (N.NE.0) THEN
      WRITE(*,1120) N
    ENDIF
  ENDIF
ENDIF
ENDIF
ELSE
  IF (X1(2).LT.0D0) THEN
    IF (PDTH.GT.BOUND1) THEN

```

```

IF (PDTH.GT.BOUND2) THEN
  IF (N.EQ.0) THEN
    NU11 = 0D0
    NU12 = TWOPI
  ELSE IF (N.EQ.2) THEN
    NU11 = RANK(2) - TWOPI
    NU12 = RANK(1)
  ELSE IF (N.EQ.4) THEN
    NU11 = RANK(4) - TWOPI
    NU12 = RANK(1)
    NU21 = RANK(2)
    NU22 = RANK(3)
  ELSE
    WRITE(*,1130) N
  ENDIF
ELSE
  IF (N.EQ.2) THEN
    NU11 = - RANK(2)
    NU12 = RANK(1)
  ELSE IF (N.EQ.4) THEN
    NU11 = - RANK(4)
    NU12 = RANK(1)
    NU21 = RANK(2)
    NU22 = RANK(3)
  ELSE
    WRITE(*,1140) N
  ENDIF
ENDIF
ELSE
  IF (PDTH.GT.BOUND2) THEN
    IF (N.EQ.2) THEN
      NU11 = RANK(2)
      NU12 = TWOPI - RANK(1)
    ELSE IF (N.EQ.4) THEN
      NU11 = RANK(2)
      NU12 = RANK(3)
      NU21 = RANK(4)
      NU22 = TWOPI - RANK(1)
    ELSE
      WRITE(*,1150) N
    ENDIF
  ELSE
    IF (N.EQ.2) THEN
      NU11 = TWOPI - RANK(2)
      NU12 = TWOPI - RANK(1)
    ELSE IF (N.EQ.4) THEN
      NU11 = RANK(2)

```

```

        NU12 = RANK(3)
        NU21 = TWOPI - RANK(4)
        NU22 = TWOPI - RANK(1)
    ELSE IF (N.NE.0) THEN
        WRITE(*,1160) N
    ENDIF
ENDIF
ENDIF
ELSE
    IF (PDTH.GT.BOUND1) THEN
        IF (PDTH.GT.BOUND2) THEN
            IF (N.EQ.0) THEN
                NU11 = 0D0
                NU12 = TWOPI
            ELSE IF (N.EQ.2) THEN
                NU11 = - RANK(1)
                NU12 = TWOPI - RANK(2)
            ELSE IF (N.EQ.4) THEN
                NU11 = - RANK(1)
                NU12 = TWOPI - RANK(4)
                NU21 = TWOPI - RANK(3)
                NU22 = TWOPI - RANK(2)
            ELSE
                WRITE(*,1170) N
            ENDIF
        ELSE
            IF (N.EQ.2) THEN
                NU11 = - RANK(1)
                NU12 = RANK(2)
            ELSE IF (N.EQ.4) THEN
                NU11 = - RANK(1)
                NU12 = RANK(4)
                NU21 = TWOPI - RANK(3)
                NU22 = TWOPI - RANK(2)
            ELSE
                WRITE(*,1180) N
            ENDIF
        ENDIF
    ELSE
        IF (PDTH.GT.BOUND2) THEN
            IF (N.EQ.2) THEN
                NU11 = RANK(1)
                NU12 = TWOPI - RANK(2)
            ELSE IF (N.EQ.4) THEN
                NU11 = RANK(1)
                NU12 = TWOPI - RANK(4)
                NU21 = TWOPI - RANK(3)
            ENDIF
        ENDIF
    ENDIF

```

```

        NU22 = TWOPI - RANK(2)
    ELSE
        WRITE(*,1190) N
    ENDIF
ELSE
    IF (N.EQ.2) THEN
        NU11 = RANK(1)
        NU12 = RANK(2)
    ELSE IF (N.EQ.4) THEN
        NU11 = RANK(1)
        NU12 = RANK(4)
        NU21 = TWOPI - RANK(3)
        NU22 = TWOPI - RANK(2)
    ELSE IF (N.NE.0) THEN
        WRITE(*,1200) N
    ENDIF
ENDIF
ENDIF
ENDIF
RETURN
900  FORMAT (// ' DSQR = ',D17.10,4X,' PR  = ',D17.10,4X,' E2= ',
1    D17.10/' SAT2P= ',D17.10,4X,' X(1)= ',D17.10,4X//
2    ' A = ',D17.10,4X,' B = ',D17.10,4X,' C = ',D17.10/
3    ' K = ',D17.10//)
1000 FORMAT (// ' Norder = ',I1,/' P1= ',D17.10,4X,' P2= ',
1    D17.10/' P3= ',D17.10,4X,' P4= ',D17.10/
2    ' P5= ',D17.10//)
1010 FORMAT (// ' P(',I1,')= ',D17.10,4X,' ANSWER(',I1,')= ',
1    '(',F16.10,' ',F16.10,')')
1020 FORMAT (// ' N= ',I1,/'(' Solution(',I1,')= ',F16.10,' deg'//)
1030 FORMAT (// ' N= ',I1,/'(' Rank(',I1,')= ',F16.10,' deg'//)
1040 FORMAT (// ' Sat 1 Projection occurs at a Sat 2 NU of ',
1    F16.10,' deg where Sat 2 radius is ',F16.10,' km'//
2    ' INSIDE = ',L1//)
1050 FORMAT (// ' ERROR! N= ',I1,' X1(2) < 0 PDTH > BOUND1 ',
1    ' PDTH > BOUND2 INSIDE'//)
1060 FORMAT (// ' ERROR! N= ',I1,' X1(2) < 0 PDTH > BOUND1 ',
1    ' PDTH < BOUND2 INSIDE'//)
1070 FORMAT (// ' ERROR! N= ',I1,' X1(2) < 0 PDTH < BOUND1 ',
1    ' PDTH > BOUND2 INSIDE'//)
1080 FORMAT (// ' ERROR! N= ',I1,' X1(2) < 0 PDTH < BOUND1 ',
1    ' PDTH < BOUND2 INSIDE'//)
1090 FORMAT (// ' ERROR! N= ',I1,' X1(2) > 0 PDTH > BOUND1 ',
1    ' PDTH > BOUND2 INSIDE'//)
1100 FORMAT (// ' ERROR! N= ',I1,' X1(2) > 0 PDTH > BOUND1 ',
1    ' PDTH < BOUND2 INSIDE'//)

```

```

1110 FORMAT (// 'ERROR! N=','I1.', X1(2) > 0 PDTH < BOUND1 ',
1 'PDTH > BOUND2 INSIDE'//)
1120 FORMAT (// 'ERROR! N=','I1.', X1(2) > 0 PDTH < BOUND1 ',
1 'PDTH < BOUND2 INSIDE'//)
1130 FORMAT (// 'ERROR! N=','I1.', X1(2) < 0 PDTH > BOUND1 ',
1 'PDTH > BOUND2 OUTSIDE'//)
1140 FORMAT (// 'ERROR! N=','I1.', X1(2) < 0 PDTH > BOUND1 ',
1 'PDTH < BOUND2 OUTSIDE'//)
1150 FORMAT (// 'ERROR! N=','I1.', X1(2) < 0 PDTH < BOUND1 ',
1 'PDTH > BOUND2 OUTSIDE'//)
1160 FORMAT (// 'ERROR! N=','I1.', X1(2) < 0 PDTH < BOUND1 ',
1 'PDTH < BOUND2 OUTSIDE'//)
1170 FORMAT (// 'ERROR! N=','I1.', X1(2) > 0 PDTH > BOUND1 ',
1 'PDTH > BOUND2 OUTSIDE'//)
1180 FORMAT (// 'ERROR! N=','I1.', X1(2) > 0 PDTH > BOUND1 ',
1 'PDTH < BOUND2 OUTSIDE'//)
1190 FORMAT (// 'ERROR! N=','I1.', X1(2) > 0 PDTH < BOUND1 ',
1 'PDTH > BOUND2 OUTSIDE'//)
1200 FORMAT (// 'ERROR! N=','I1.', X1(2) > 0 PDTH < BOUND1 ',
1 'PDTH < BOUND2 OUTSIDE'//)
END

```

```

C *-----*
C *
C *          Function M()
C *
C *-----*
C REAL*8 FUNCTION M(NU)
C *-----*
LOGICAL*4 DEBUG
REAL*8 NU, SHNU, CHNU, X, Y, Z
REAL*8 BOUND1, BOUND2, DTH, DSQR, PDTH, PR, NU11, NU12,
1 NU21, NU22
REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1(4),
1 SINC1, CINC1, SNU1, CNU1, SARGP1, CARGP1,
2 SLNAN1, CLNAN1, BETA1
REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2(4),
1 SINC2, CINC2, SNU2, CNU2, SARGP2, CARGP2,
2 SLNAN2, CLNAN2, BETA2
REAL*8 R11, R12, R13, R21, R22, R23, R31, R32, R33,
1 S11, S12, S13, S21, S22, S23, S31, S32, S33
REAL*8 HALFPI, PI, TWOPI, DEGRAD, RADDEG
REAL*8 ER, DU, TU
COMMON /ADMIN/ DEBUG
COMMON /ASTRO/ ER, DU, TU
COMMON /SOUND/ BOUND1, BOUND2, DTH, DSQR, PDTH, PR, NU11,
1 NU12, NU21, NU22
COMMON /CONST/ HALFPI, PI, TWOPI, DEGRAD, RADDEG

```

```

COMMON /SAT1/  M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1,
1 SINC1, CINC1, SNU1, CNU1, SARGP1, CARGP1,
2 SLNAN1, CLNAN1, BETA1
COMMON /SAT2/  M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2,
1 SINC2, CINC2, SNU2, CNU2, SARGP2, CARGP2,
2 SLNAN2, CLNAN2, BETA2
COMMON /TRANS/ R11, R12, R13, R21, R22, R23, R31, R32,
1 R33, S11, S12, S13, S21, S22, S23, S31, S32, S33
C  *-----*
  IF (NU.EQ.0D0) THEN
    M = 0D0
    RETURN
  ELSE IF (DABS(NU).EQ.PI) THEN
    M = DSIGN(PI,NU)
    RETURN
  ELSE IF (NU.EQ.TWOPI) THEN
    M = TWOPI
    RETURN
  ENDIF
  SHNU = DSIN(5D-1*NU)
  CHNU = DCOS(5D-1*NU)
  IF (CHNU.EQ.0D0) THEN
    M = DSIGN(PI,NU)
    IF (DEBUG) WRITE(*,1000) RADDEG*NU, RADDEG*M
    RETURN
  ENDIF
  M = 2D0*( DATAN( BETA2*SHNU/CHNU ) - E2*BETA2*SHNU*CHNU/
1  (CHNU*CHNU + BETA2*BETA2 * SHNU*SHNU ))
  IF (M.LT.0D0) M = TWOPI + M
  IF (NU.LT.0D0) M = M - TWOPI
  IF (DEBUG) WRITE(*,1000) RADDEG*NU, RADDEG*M
  RETURN
1000 FORMAT (' NU=' ,F16.10, ' deg' ,4X, ' M=' ,F16.10, ' deg' )
  END
C  *-----*
C  *
C  *          Function PDCA()
C  *
C  *-----*
C  REAL*8 FUNCTION PDCA(NU)
C  *-----*
  LOGICAL*4 DEBUG
  INTEGER*4 I
  REAL*8 M, SHNU1, CHNU1, DMPDNU
  REAL*8 NU, DX, DY, X, Y, Z
  REAL*8 BOUND1, BOUND2, DTH, DSQR, PDTH, PR, NU11, NU12,
1  NU21, NU22

```

```

REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1(4),
1 SINC1, CINC1, SNU1, CNU1, SARGP1, CARGP1,
2 SLNAN1, CLNAN1, BETA1
REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2(4),
1 SINC2, CINC2, SNU2, CNU2, SARGP2, CARGP2,
2 SLNAN2, CLNAN2, BETA2
REAL*8 R11, R12, R13, R21, R22, R23, R31, R32, R33,
1 S11, S12, S13, S21, S22, S23, S31, S32, S33
REAL*8 HALFPI, PI, TWOPI, DEGRAD, RADDEG
REAL*8 ER, DU, TU
COMMON /ADMIN/  DEBUG
COMMON /ASTRO/  ER, DU, TU
COMMON /BOUND/  BOUND1, BOUND2, DTH, DSQR, PDTH, PR, NU11,
1 NU12, NU21, NU22
COMMON /CONST/  HALFPI, PI, TWOPI, DEGRAD, RADDEG
COMMON /SAT1/    M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1,
1 SINC1, CINC1, SNU1, CNU1, SARGP1, CARGP1,
2 SLNAN1, CLNAN1, BETA1
COMMON /SAT2/    M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2,
1 SINC2, CINC2, SNU2, CNU2, SARGP2, CARGP2,
2 SLNAN2, CLNAN2, BETA2
COMMON /TRANS/  R11, R12, R13, R21, R22, R23, R31, R32,
1 R33, S11, S12, S13, S21, S22, S23, S31, S32, S33
C      *-----*
C      NU1 = NU
C      SNU1 = DSIN(NU1)
C      CNU1 = DCOS(NU1)
C      *-----*
C      *      Compute the Radius of Sat 1      *
C      *-----*
C      X1(4) = A1 * (1D0 - E1*E1) / (1D0 + E1*CNU1)
C      *-----*
C      *      Compute the position of Sat 1 in      *
C      *      Perifocal Coordinate Frame      *
C      *-----*
C      X1(1) = X1(4)*CNU1
C      X1(2) = X1(4)*SNU1
C      X1(3) = 0D0
C      IF (DEBUG) WRITE(*,1000) RADDEG*NU1, (X1(I),I=1,4)
C      *-----*
C      *      Compute the position of Sat 1 in      *
C      *      the Geocentric Equatorial Frame.      *
C      *-----*
C      X = R11*X1(1) + R12*X1(2)
C      Y = R21*X1(1) + R22*X1(2)
C      Z = R31*X1(1) + R32*X1(2)
C      IF (DEBUG) WRITE(*,1010) X,Y,Z,DSQRT(X*X+Y*Y+Z*Z)

```

```

C      *-----*
C      * Transform Sat 1's position from      *
C      * the Geocentric Equatorial Frame      *
C      * to Sat 2's perifocal frame          *
C      *-----*
X1(1) = S11*X + S12*Y + S13*Z
X1(2) = S21*X + S22*Y + S23*Z
X1(3) = S31*X + S32*Y + S33*Z
PR = DSQRT(X1(1)*X1(1)+X1(2)*X1(2))
IF (DEBUG) WRITE(*,1020) (X1(I),I=1,3),
1 DSQRT(X1(1)*X1(1)+X1(2)*X1(2)+X1(3)*X1(3)), PR
C      *-----*
C      * Compute the distance between X1      *
C      * projected onto Sat 2's perifocal      *
C      * plane and Sat 2's perigee (BOUND1) *
C      * and apogee (BOUND2)                  *
C      *-----*
DX = X1(1) - A2*(1D0-E2)
DY = X1(2)
BOUND1 = DSQRT(DX*DX + DY*DY)
DX = X1(1) + A2*(1D0+E2)
BOUND2 = DSQRT(DX*DX + DY*DY)
C      *-----*
C      * Find out if there are any points      *
C      * on Sat 2's orbit that are exactly      *
C      * DTH away from the endpoint of          *
C      * vector X1. If there are, then          *
C      * there are either two or four          *
C      * points.                                *
C      *-----*
DSQR = DTH*DTH - X1(3)*X1(3)
PDTH = OD0
IF (DSQR .GT. OD0) THEN
  PDTH = DSQRT(DSQR)
  IF (DEBUG) WRITE(*,1030) PDTH, BOUND1, BOUND2
  CALL ORBBND()
  IF (DEBUG) CALL PRINTR()
ELSE
  IF (DEBUG) WRITE(*,1030) PDTH, BOUND1, BOUND2
  PDCA = OD0
  RETURN
ENDIF
C      *-----*
C      * Compute Probability Density of        *
C      * Close Approach                        *
C      *-----*
PDCA = (M(NU12)-M(NU11) + M(NU22)-M(NU21)) / (TWOPI*TWOPI)

```

```

      IF (DEBUG) WRITE(*,1040) PDCA
      SHNU1 = DSIN(5D-1*NU1)
      CHNU1 = DCOS(5D-1*NU1)
      DMPDNU = CHNU1*CHNU1 + BETA1*BETA1*SHNU1*SHNU1
      DMPDNU = (1D0-E1)*BETA1/(DMPDNU*DMPDNU)
      PDCA = PDCA*DMPDNU
      IF (DEBUG) WRITE(*,1050) PDCA, DMPDNU
      RETURN
1000  FORMAT (// ' Input NU=',F16.10//
1      ' Sat 1 Position in Sat 1's perifocal frame://'
2      '      X Axis: ',F17.10,' km'/
3      '      Y Axis: ',F17.10,' km'/
4      '      Z Axis: ',F17.10,' km'/
5      ' Sat 1 Radius: ',F17.10,' km'//)
1010  FORMAT (' Sat 1 Position in the inertial reference frame:'
1      '//'      X Axis: ',F17.10,' km'/
2      '      Y Axis: ',F17.10,' km'/
3      '      Z Axis: ',F17.10,' km'/
4      ' Sat 1 Radius: ',F17.10,' km'//)
1020  FORMAT (' Sat 1 Position in Sat 2's perifocal frame://'
1      '      X Axis: ',F17.10,' km'/
2      '      Y Axis: ',F17.10,' km'/
3      '      Z Axis: ',F17.10,' km'/
4      ' Sat 1 Radius: ',F17.10,' km'/
5      '      Projection: ',F17.10,' km'/
6      '      Radius'//)
1030  FORMAT (// ' PDTH=',F17.10/ ' BOUND1=',F17.10.
1      4X,' BOUND2=',F17.10/)
1040  FORMAT (' Unscaled PDCA=',F17.10/)
1050  FORMAT (' PDCA=',F16.10,4X,' DMPDNU=',F16.10//)
      END
C      *-----*
C      *
C      *      Subroutine PRINTR()
C      *
C      *-----*
C      SUBROUTINE PRINTR()
C      *-----*
      LOGICAL*4 DEBUG
      INTEGER*4 I, J, K
      REAL*8 DX, DY, DZ, DR1, DR2, DR3, DR4
      REAL*8 BOUND1, BOUND2, DTH, DSQR, PDTH, PR, NU11, NU12,
1      NU21, NU22
      REAL*8 M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1(4)
      REAL*8 M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2(4)
      REAL*8 SINC1, CINC1, SNU1, CNU1, SARGP1, CARGP1,
1      SLNAN1, CLNAN1, BETA1

```

```

REAL*8 SINC2, CINC2, SNU2, CNU2, SARGP2, CARGP2,
1 SLNAN2, CLNAN2, BETA2
REAL*8 HALFPI, PI, TWOPI, DEGRAD, RADDEG
REAL*8 ER, DU, TU
COMMON /ADMIN/  DEBUG
COMMON /ASTRO/  ER, DU, TU
COMMON /BOUND/  BOUND1, BOUND2, DTH, DSQR, PDTH, PR, NU11,
1 NU12, NU21, NU22
COMMON /CONST/  HALFPI, PI, TWOPI, DEGRAD, RADDEG
COMMON /SAT1/   M1, A1, E1, INC1, NU1, ARGPA1, LONAN1, X1,
1 SINC1, CINC1, SNU1, CNU1, SARGP1, CARGP1,
2 SLNAN1, CLNAN1, BETA1
COMMON /SAT2/   M2, A2, E2, INC2, NU2, ARGPA2, LONAN2, X2,
1 SINC2, CINC2, SNU2, CNU2, SARGP2, CARGP2,
2 SLNAN2, CLNAN2, BETA2

```

C

```

*-----*
X2(4) = A2*(1D0-E2*E2)/(1D0+E2*DCOS(NU11))
X2(1) = X2(4)*DCOS(NU11)
X2(2) = X2(4)*DSIN(NU11)
X2(3) = 0D0
DX = X2(1) - X1(1)
DY = X2(2) - X1(2)
DZ = X2(3) - X1(3)
DR1 = DSQRT(DX*DX + DY*DY + DZ*DZ)
X2(4) = A2*(1D0-E2*E2)/(1D0+E2*DCOS(NU12))
X2(1) = X2(4)*DCOS(NU12)
X2(2) = X2(4)*DSIN(NU12)
X2(3) = 0D0
DX = X2(1) - X1(1)
DY = X2(2) - X1(2)
DZ = X2(3) - X1(3)
DR2 = DSQRT(DX*DX + DY*DY + DZ*DZ)
WRITE(*,1000) RADDEG*NU11, DR1, RADDEG*NU12, DR2
X2(4) = A2*(1D0-E2*E2)/(1D0+E2*DCOS(NU21))
X2(1) = X2(4)*DCOS(NU21)
X2(2) = X2(4)*DSIN(NU21)
X2(3) = 0D0
DX = X2(1) - X1(1)
DY = X2(2) - X1(2)
DZ = X2(3) - X1(3)
DR3 = DSQRT(DX*DX + DY*DY + DZ*DZ)
X2(4) = A2*(1D0-E2*E2)/(1D0+E2*DCOS(NU22))
X2(1) = X2(4)*DCOS(NU22)
X2(2) = X2(4)*DSIN(NU22)
X2(3) = 0D0
DX = X2(1) - X1(1)
DY = X2(2) - X1(2)

```

```

DZ = X2(3) - X1(3)
DR4 = DSQRT(DX*DX + DY*DY + DZ*DZ)
WRITE(*,1100) RADDEG*NU21, DR3, RADDEG*NU22, DR4
RETURN
1000 FORMAT (// ' NU11=',F16.10,' deg',4X,
1      'Range to NU11 = ',F16.10,' km'/
2      ' NU12=',F16.10,' deg',4X,
3      'Range to NU12 = ',F16.10,' km')
1100 FORMAT (' NU21=',F16.10,' deg',4X,
1      'Range to NU21 = ',F16.10,' km'/
2      ' NU22=',F16.10,' deg',4X,
3      'Range to NU22 = ',F16.10,' km'//)
END
C      *-----*
C      *
C      *          Function CCUBRT
C      *
C      *-----*
COMPLEX*16 FUNCTION CCUBRT(X)
COMPLEX*16 X
C      *-----*
REAL*8 A, ANGLE, B, MAG
REAL*8 HALFPI,PI,TWOPI,DEGRAD,RADDEG
COMMON /CONST/ HALFPI, PI, TWOPI, DEGRAD, RADDEG
C      *-----*
A = DREAL(X)
B = DIMAG(X)
IF ( B .EQ. 0D0) THEN
    MAG = DABS(A)
    MAG = DSIGN( MAG**(1D0/3D0), A)
    CCUBRT = DCMPLX( MAG, 0D0)
ELSE
    MAG = CDABS(X)
    ANGLE = DATAN2( B, A)
    IF ( A .LT. 0D0) THEN
        ANGLE = (DSIGN(TWOPI,ANGLE)+ANGLE)/3D0
    ELSE
        ANGLE = ANGLE/3D0
    ENDIF
    MAG = MAG**(1D0/3D0)
    CCUBRT = DCMPLX(MAG*DCOS(ANGLE),MAG*DSIN(ANGLE))
ENDIF
RETURN
END
C      *-----*
C      *
C      *          Subroutine QUADRATIC
C      *

```

```

C      *
C      *-----*
SUBROUTINE QUADRATIC(P,X)
REAL*8 P(2)
COMPLEX*16 X(2)
C      *-----*
REAL*8 A
COMPLEX*16 B
C      *-----*
A = -P(1)
B = CDSQRT(DCMPLX(A*A - 4D0*P(2)))
X(1) = (A - B)/2D0
X(2) = (A + B)/2D0
RETURN
END
C      *-----*
C      *
C      *          Subroutine CUBIC
C      *
C      *-----*
SUBROUTINE CUBIC(P,X)
REAL*8 P(3)
COMPLEX*16 X(3)
C      *-----*
REAL*8 A, B, C
COMPLEX*16 CCUBRT, CX, D, E, SQRT3XJ
C      *-----*
SQRT3XJ = (0D0, 1.73205080756887729D0)
A = (3D0*P(2)-P(1)*P(1))/3D0
B = (2D0*P(1)*P(1)*P(1) - 9D0*P(1)*P(2) + 27D0*P(3))/27D0
C = B*B/4D0 + A*A*A/27D0
IF (C.LT.0D0) THEN
  CX = (0D0,1D0)*DSQRT(DABS(C))
ELSE
  CX = (1D0,0D0)*DSQRT(C)
ENDIF
D = CCUBRT(-B/2D0 + CX)
E = CCUBRT(-B/2D0 - CX)
X(1) = D + E - P(1)/3D0
X(2) = -(D + E)/2D0 + (D - E)/2D0*SQRT3XJ - P(1)/3D0
X(3) = -(D + E)/2D0 - (D - E)/2D0*SQRT3XJ - P(1)/3D0
RETURN
END
C      *-----*
C      *
C      *          Subroutine QUARTIC
C      *

```

```

C      *-----*
      SUBROUTINE QUARTIC(P,X)
      REAL*8 P(4)
      COMPLEX*16 X(4)
C      *-----*
      INTEGER I, IS
      REAL*8 C(3), Y
      COMPLEX*16 D, E, R, S(3)
C      *-----*
      C(1) = -P(2)
      C(2) = P(1)*P(3)-4D0*P(4)
      C(3) = -P(1)*P(1)*P(4) + 4D0*P(2)*P(4) - P(3)*P(3)
      CALL CUBIC(C,S)
      IS = 0
      DO 100 I=1,3
        IF (DIMAG(S(I)).EQ.0D0) THEN
          IF (IS.EQ.0D0) THEN
            IS = I
            Y = DREAL(S(I))
          ELSE
            IF (DREAL(S(I)).GT.Y) THEN
              IS = I
              Y = DREAL(S(I))
            ENDIF
          ENDIF
        ENDIF
      CONTINUE
100    IF (IS.EQ.0) THEN
      Y = 0D0
      WRITE(*,1000)
      DO 200 I=1,3
200    WRITE(*,1010) DREAL(S(I)), DIMAG(S(I))
      ENDIF
      R = CDSQRT(DCMPLX(P(1)*P(1)/4D0 - P(2) + Y))
      IF (R.EQ.(0D0,0D0)) THEN
        D = 2D0*CDSQRT(DCMPLX(Y*Y - 4D0*P(4)))
        E = -D
        D = CDSQRT(.75D0*P(1)*P(1) - 2D0*P(2) + D)
        E = CDSQRT(.75D0*P(1)*P(1) - 2D0*P(2) + E)
      ELSE
        D = (4D0*P(1)*P(2)-8D0*P(3)-P(1)*P(1)*P(1))/4D0/R
        E = -D
        D = CDSQRT(.75D0*P(1)*P(1) - R*R - 2D0*P(2) + D)
        E = CDSQRT(.75D0*P(1)*P(1) - R*R - 2D0*P(2) + E)
      ENDIF
      X(1) = -P(1)/4D0 + R/2D0 + D/2D0
      X(2) = -P(1)/4D0 + R/2D0 - D/2D0

```

```

X(3) = -P(1)/4D0 - R/2D0 + E/2D0
X(4) = -P(1)/4D0 - R/2D0 - E/2D0
RETURN
1000 FORMAT (' Cubic Error! All 3 roots were complex!'/)
1010 FORMAT (' Root: ',I1,' (' ,F17.10,' ',F17.10,' ')')
END

```

Bibliography

1. Bate, R.R., Mueller, D.D., and White, J.E., Fundamentals of Astrodynamics, New York: Dover Publications, Inc., 1971.
2. Brouwer D., and Clemence G.M., Methods of Celestial Mechanics, New York and London: Academic Press, 1961.
3. Chobotov, V.A., "The Collision Hazard in Space," Journal of the Astronautical Sciences, Vol. XXX, No. 3, July-Sept. 1982, pp. 191-212.
4. Gradshteyn, I.S., and Ryzhik, I.M., Table of Integrals, Series, and Products, New York and London: Academic Press, 1965.
5. Helstrom, C.W., Probability and Stochastic Processes for Engineers, New York: Macmillan Publishing Company, 1984.
6. Kessler, D.J. and Cour-Palais, B.G., "Collision Frequency of Artificial Satellites: The Creation of a Debris Belt," Journal of Geophysical Research, Vol. 83, No. A6, June 1, 1978, pp. 2637-2646.
7. Selby, S.M., Standard Mathematical Tables, Cleveland, Ohio: The Chemical Rubber Co., 1971,
8. Takahashi, K., "Collision Between Satellites in Stationary Orbits," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-17, No. 4, July 1981, pp. 591-596.
9. Zazworsky, R.M., "Intercept Probability of Satellites in Circular Orbits," Journal of Guidance and Control, Vol. 8, No. 1, Feb 1985, pp. 118-124.

VITA

Captain Randal L. Richey was [REDACTED] [REDACTED]

[REDACTED] PII Redacted

[REDACTED] PII Redacted

[REDACTED] He graduated from high school [REDACTED] in 1972, and then was accepted into the USAF Academy. In June 1976, he received a Bachelor of Science in Astronautical Engineering, along with a commission in the USAF. His first assignment was as Squadron Astronautical Engineer in the Det 1, 4602 CPUSS at Lowry AFB, Colorado. In June 1980, he received the Air Force Association's highest honor in the field of science and engineering, the Theodore Von Karman Award, for his work at Lowry AFB. In 1980, he was assigned to SAFSP/OD-4 at Sunnyvale AFS, California, where he remained until entering the School of Engineering, Air Force Institute of Technology, in June 1984.

Permanent address: [REDACTED]

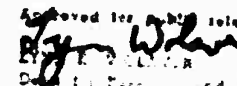
[REDACTED] PII Redacted

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

A179385

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS						
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.						
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE									
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GA/AA/85D-8			5. MONITORING ORGANIZATION REPORT NUMBER(S)						
6a. NAME OF PERFORMING ORGANIZATION School of Engineering		6b. OFFICE SYMBOL (If applicable) AFIT/EN		7a. NAME OF MONITORING ORGANIZATION					
6c. ADDRESS (City, State and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433			7b. ADDRESS (City, State and ZIP Code)						
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER					
8c. ADDRESS (City, State and ZIP Code)			10. SOURCE OF FUNDING NOS.						
			<table border="1"> <tr> <td>PROGRAM ELEMENT NO.</td> <td>PROJECT NO.</td> <td>TASK NO.</td> <td>WORK UNIT NO.</td> </tr> </table>			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.						
11. TITLE (Include Security Classification) See Box 19									
12. PERSONAL AUTHOR(S) Randal L. Richey, B.S., Capt, USAF									
13a. TYPE OF REPORT MS Thesis		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Yr. Mo., Day) 1986 December					
15. PAGE COUNT 134									
16. SUPPLEMENTARY NOTATION									
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)						
FIELD	GROUP	SUB GR	Close Approach, Collision, Intercept, Probability, Satellite						
22	3								
19. ABSTRACT (Continue on reverse if necessary and identify by block number)									
Title: DETERMINING THE PROBABILITY OF CLOSE APPROACH BETWEEN TWO SATELLITES									
Thesis Chairman: William E. Wiesel, Ph.D. Associate Professor of Astronautics									
Approved for public release; IAW AFR 100-1  12 March 87 Director, Research and Professional Development Air Force Institute of Technology (AFIT) Wright-Patterson AFB, OH 45433									
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS ☐			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED						
22a. NAME OF RESPONSIBLE INDIVIDUAL William E. Wiesel, Ph.D.		22b. TELEPHONE NUMBER (Include Area Code) 513-255-2362		22c. OFFICE SYMBOL AFIT/ENY					

This theory developed into

Probability of close approach is the probability that two satellites will be within some specified distance threshold of each other at a random time within a specified time interval. In this paper, methods were developed to calculate probability of close approach between two satellites. To simplify the analysis, the investigation was restricted to satellite orbits and time intervals where the mean anomaly of both satellites can be treated as independent, uniformly distributed random variables. In addition, all orbital parameters, except for mean anomaly, were assumed to be constant over time. This means that all the methods developed in this paper to calculate the probability of close approach will only be valid over very long time intervals where the ratio of the orbital periods of the two satellites can be approximated as an irrational number. Likewise, there can be no perturbations in the orbital parameters of both satellites.

The first method developed was a general method for calculating the probability of close approach between two satellites in elliptical orbits. The method requires numerical integration and direct solution of the roots of a 4th order polynomial during each numerical integration step.

Another method was developed for calculating the probability of close approach between two satellites in circular orbits. This method still requires numerical integration to obtain a solution, but in this case a direct solution was found for the limits of integration. Furthermore, the calculations required during each numerical integration step are much simpler than those required to calculate the probability of close approach with elliptical orbits.

Finally, a direct solution for approximate probability of close approach between two satellites in circular orbits was developed for the case where the angle between the orbital planes of both satellites is not small and the probability of close approach is small.

Both the elliptical orbit and the circular orbit methods of computing probability of close approach yielded results that compare favorably with estimates of probability of close approach derived from statistical simulations.