(Preprint) AAS XX-XXX

A MULTISTEP PROBABILITY OF COLLISION COMPUTATIONAL ALGORITHM

Doyle T. Hall,[*] Luis G. Baars,[†] and Stephen J. Casali[‡]

The two-dimensional probability of collision calculation method is both widely used and computationally efficient. However, Monte Carlo simulations show that this method sometimes fails to provide sufficiently accurate estimates for Earth-orbiting satellites. This study presents a multistep algorithm that calculates collision probabilities for all conjunctions, regardless if affected by curvilinear trajectory or time varying covariance dynamics. The algorithm sequentially applies increasingly accurate estimation methods, but only as required for efficiency. Evaluating usage violations for the two-dimensional probability of collision method represents one of the most important steps. Extensive testing demonstrates the efficiency and reliability of the multistep algorithm.

INTRODUCTION

The NASA Conjunction Assessment Risk Analysis (CARA) program is responsible for monitoring and providing collision avoidance to NASA's un-crewed spacecraft in orbit. The CARA team estimates collision probabilities for a specific set of high value Earth-orbiting satellites[1] based on the latest available satellite tracking data and orbit determination (OD) solutions.[2] Like CARA, many other conjunction analysis organizations use the probability of collision (P_c) to assess the risk of collisions between tracked Earth-orbiting satellites. The semi-analytical two-dimensional collision probability estimation method (2D-P_c), originally formulated in 1992,[3] represents the most widely used algorithm, in part because it has been improved in computational efficiency significantly over the past decades.[4-7] However, the 2D-P_c method relies on several assumptions that must be satisfied in order to produce accurate results, including using the approximations of locally rectilinear motion and time-invariant covariance matrices. Monte Carlo (MC) simulations, which do not invoke such limiting assumptions, provide a means to test the accuracy of semi-analytical methods, and allow accurate P_c estimation for low-velocity and multi-encounter interactions.[8] Extensive MC analysis of actual conjunction events shows that the 2D-P_c method sometimes fails to provide sufficiently accurate estimates for reliable conjunction risk assessments of satellites in low-Earth orbit (LEO); these relatively rare 2D-P_c method failures include some low relative velocity conjunctions, but also some high velocity encounters that involve long propagations or eccentric orbits.[9]

In these cases, the semi-analytical "3D-N_c" method – which evaluates the statistically expected number of collisions (N_c) as a means to estimate P_c values – provides a way of processing both

[*]Senior Conjunction Assessment Research Scientist, Omitron Inc., 555 E. Pikes Peak Ave, #205, Colorado Springs, CO 80903.
[†]Conjunction Assessment Research Analyst, Omitron Inc., 555 E. Pikes Peak Ave, #205, Colorado Springs, CO 80903.
[‡]Senior Astrodynamicist, Omitron Inc., 555 E. Pikes Peak Ave, #205, Colorado Springs, CO 80903.

single- and multi-encounter interactions.[10] The 3D-N_c formulation does this by accurately accounting for curvilinear trajectory and time varying covariance effects. Although usually much more efficient than MC methods, 3D-N_c requires significantly more computation than the 2D-P_c method. An optimized computation would only apply 3D-N_c to the limited subset of conjunctions for which 2D-P_c might fail. However, this hybrid approach requires a reliable method to identify such 2D-P_c method "usage violations." Increasing the speed of the 3D-N_c calculation itself also would provide further efficiency improvements.

This study presents a multistep algorithm designed to calculate P_c values both accurately and efficiently for all CARA conjunctions, regardless if affected by curvilinear trajectories, low velocities, or covariance variations. For each conjunction, the process starts with the 2D-P_c calculation, and identification of any associated usage violations. The algorithm then sequentially applies other, more accurate semi-analytical P_c estimation methods, but only as required. It next tries the "2D-N_c" method, which is significantly more efficient than the 3D-N_c method because it evaluates one of the required integrals analytically rather than numerically. If necessary, the algorithm then tries the more advanced 3D-N_c method itself, a step required relatively infrequently and primarily for long duration conjunctions caused by low relative velocities. Finally, in the very rare cases that all of these semi-analytical methods fail, or if the user desires validation of the results, the algorithm performs MC P_c estimation.

Evaluating 2D-P_c method usage violations represents one of the most important elements of the multistep algorithm. Detecting these violations entails comparing Mahalanobis distances[11] between the two satellites calculated using the rectilinear vs curvilinear trajectory approximations. Large differences between the Mahalanobis distance curves indicate likely 2D-P_c method failures. Usage violations correspond to cases for which the differences exceed a conservatively small threshold value. The analysis also provides an approximate correction factor for the 2D-P_c estimate, as well as an estimated duration for the conjunction. The multistep procedure follows up these calculations with more accurate estimation methods for conjunctions that have 2D-P_c method usage violations and corrected P_c estimates that are too large, or that have estimated durations that are too long.

Testing based on almost two million conjunctions between actual satellites indicates that the multistep algorithm provides reliably accurate P_c estimates, and is significantly more efficient than always using advanced semi-analytical or MC estimation methods.

OVERVIEW OF THE MULTISTEP COLLISION OF PROBABILITY ALGORITHM

This section describes the multistep collision probability algorithm, which uses the following five increasingly accurate computational methods:

1. The 2D-P_c method: the two-dimensional collision probability method estimates P_c values for rectilinear motion conjunctions by calculating an integral over a circular region on the conjunction plane.
2. The 2D-N_c method: the two-dimensional expected number of collisions method estimates P_c values for high velocity curvilinear motion conjunctions by calculating one integral over the collision sphere.
3. The 3D-N_c method: the three-dimensional expected number of collisions method estimates P_c values for high and low velocity curvilinear motion conjunctions by calculating a series of collision sphere integrals.
4. The SDMC method: the Simplified Dynamics Monte Carlo method, estimates P_c values for curvilinear motion conjunctions by calculating a sampled series of two-body motion trajectory propagations.

5. The BFMC method: the Brute Force Monte Carlo method, estimates P_c values for curvilinear motion single- or multi-conjunction interactions by calculating a sampled series of high fidelity trajectory propagations starting from the initial OD epochs.

The first three methods represent semi-analytical approximations, and the last two MC approaches. Because the computation required by each successive method increases significantly, the multistep algorithm by default invokes each only as required for efficiency. CARA's developmental software implementation executes the first four methods in an automated fashion. The final BFMC method represents CARA's most accurate, "gold standard" P_c estimation method, which is extremely computationally intensive[9] but rarely required for operational processing. The first four methods rely on assumptions or approximations that are not necessarily rigorously satisfied for all conjunctions, creating the possibility of P_c estimation inaccuracies. To address this, the multistep algorithm computes a set of conservative usage violation indicators that determine whether to invoke each successive P_c estimation method. The multistep algorithm aims to produce final "refined" P_c estimates that would all agree with the BFMC method, had it been actually executed.

Flowchart for the Multistep Calculation

Figure 1 shows a flowchart of the multistep P_c algorithm. The gray rectangles show the main computation modules, and the blue diamonds indicate decision points.

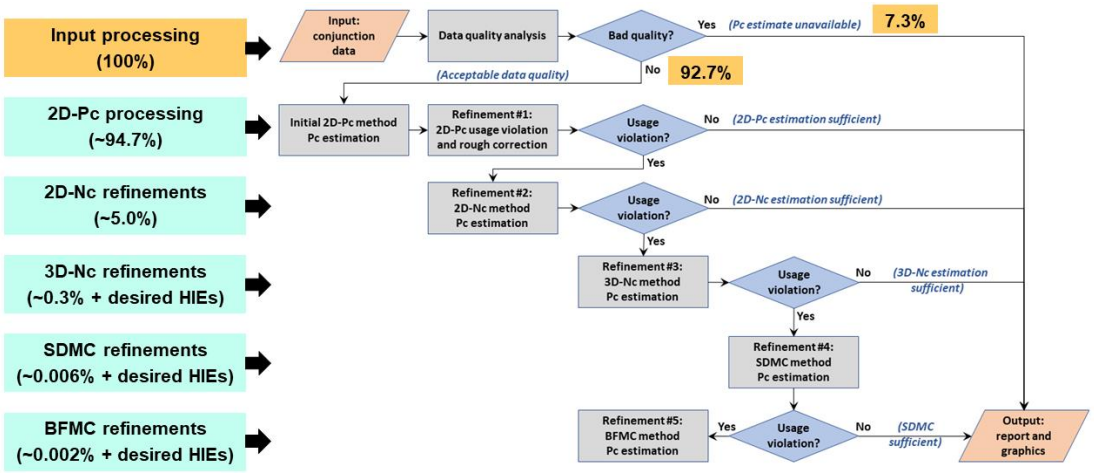

Figure 1. Schematic flowchart of the multistep algorithm for collision probability computation.

Fraction of CARA Conjunctions Processed at Each Step of P_c Refinement

The rectangles on the left side of Figure 1 show the fractions of conjunctions processed at each of the refinement steps, based on testing of 1.93 million conjunctions archived in the CARA database during the years 2021 and 2022. Among this original set, 7.3% of the conjunction data structures possess orbit determination (or filtering) conditions that prevent further processing. For the remaining 1.79 million conjunctions, the multistep algorithm reports refined estimates based on the 2D-P_c method for 94.7%, the 2D-N_c method for 5.0%, and the 3D-N_c method for 0.3%. Relatively few conjunctions ultimately require estimation using MC methods, 0.006% for SDMC and 0.002% for BFMC. By default, the algorithm invokes the successive estimation methods only as required, but optionally allows processing with any or all of the methods, useful for testing and development, as well as for complete analysis of "high interest events" (HIEs).

COLLISION PROBABILITY INPUT PARAMETERS AND RELATED QUANTITIES

The parameters required for P_c estimation include size estimates for the primary and secondary satellites, as well as orbital states and covariance matrices. This section describes these parameters and some related quantities.

The Combined Hard-body Radius for Known and Unknown Satellites

Because manufactured satellites generally have irregular shapes, P_c methods typically estimate probabilities for the spheres that circumscribe each of the two objects.[3-10,12] For known satellites, this "hard-body" radius (HBR) parameterization is both conservative (because it spans each satellite's fully deployed components) and convenient (because it is a single, time-invariant parameter). The collision probability for a conjunction between satellites that both have known sizes depends on the combined HBR, i.e., $R = R_1 + R_2$. Most CARA conjunctions, however, represent encounters involving unknown objects, typically debris created by on-orbit satellite explosions or collisions.[13] For many of these, radar cross section (RCS) measurements provide a means to estimate HBR values and associated uncertainties.[14,15] Analysis indicates that collision probabilities involving unknown secondary objects are approximated reasonably well by using the following effective combined HBR[15]

$$R = \sqrt{(R_1 + \bar{R}_2)^2 + \sigma_{R_2}^2} \tag{1}$$

with $(\bar{R}_2, \sigma_{R_2}^2)$ denoting the mean and variance of the RCS based secondary HBR estimate.

The Expected Number of Collisions (N_c) and Collision Probability (P_c)

The 2D-N_c and 3D-N_c methods evaluate the statistically expected number of collisions, N_c, as a means to estimate P_c values for single conjunctions, or to establish lower and upper P_c bounds for multi-encounter interactions. The relationship between N_c and P_c is straightforward. For an isolated conjunction, the expected number of collisions equals the collision probability, i.e., $N_c = P_c$. For a typical multi-encounter interaction, the total expected number of collisions is the linear sum of the individual conjunction collision probabilities, i.e., $N_c = \sum_k P_{c,k}$. The total collision probability for a multi-encounter interaction has the following bounds[10]

$$\max_k\left(P_{c,k}\right) \le P_c \le 1 - \prod_k (1 - P_{c,k}) \tag{2}$$

The upper bound corresponds to full statistical independence between all of the conjunctions, and the lower bound represents the opposite extreme. (Note: this assumes that the individual conjunctions in the sequence each have sufficiently short durations so that they are not blended[10] in time with one another; this temporally isolated assumption is satisfied for the vast majority of CARA conjunctions, as discussed later.) The upper limit from eq. (2) provides a conservative P_c estimate for multi-encounter interactions.

Trajectory Uncertainty Estimation and Covariance Realism

Realistic assessment of satellite collision risks requires accurate characterization of trajectory uncertainties. A review of previous studies[16] indicates that there are several sources of orbital state estimation errors, resulting in both aleatoric uncertainties (e.g., due to random errors arising from natural variability) and epistemic uncertainties (e.g., due to systematic errors arising from limited data or ignorance of physical processes). Risk assessment based on the probability of collision addresses random errors; addressing systematic errors requires using other metrics, such as the recently formulated *credibility of collision*[17] as discussed in more detail later.

Many P_c estimation methods approximate random trajectory uncertainties using a single Gaussian probability density function (PDF) for orbital states. This approach does not always provide an accurate assessment of uncertainties because the OD estimation process potentially suffers a lack of *covariance realism*.[16] In order to model trajectory uncertainties more accurately, some OD methods estimate each orbital state PDF as a Gaussian mixture model (GMM), i.e., a weighted sum of multiple multivariate normal (MVN) functions.[18,19] In these cases, calculating the collision probability of the k^{th} isolated conjunction requires a double summation

$$P_{c,k} = \sum_{g_1} \sum_{g_2} \left(P_{c,k,g_1,g_2} \right) \tag{3}$$

with the indices g_1 and g_2 spanning the GMM components of the primary and secondary orbital state PDFs, respectively. The algorithm illustrated in Figure 1 aims to calculate individual component probabilities, i.e., single P_{c,k,g_1,g_2} values. (Note: for brevity this analysis suppresses the k, g_1 and g_2 indices, unless otherwise stated.)

Instead of using the GMM approach of eq. (3), other methods apply multiplicative factors to state covariance matrices in an effort to correct for poor covariance realism, and estimate the associated effects on collision probabilities.[20-22] In these cases, given the scaled covariance matrices as input, the multistep P_c algorithm still can be used to account for the effects of both curvilinear trajectories and long duration encounters.

The Credibility of Collision

Recently, Delande *et al* [17] introduced the concept of the credibility of collision, which assesses risk based on orbital state possibility functions (rather than PDFs), in order to account for systematic trajectory uncertainties. Like PDFs, possibility functions also can take the form of single Gaussians or GMMs that incorporate OD-based mean state vectors and covariance matrices. Because of this, curvilinear trajectories and long duration encounters also potentially compromise the accuracy of credibility of collision calculations. Preliminary analysis indicates that the approach used by the 3D-N_c method of optimally re-centering the point used to linearize the equations of motion[10] also provides a means of increasing the accuracy of credibility of collision estimation. (Note: although not addressed further in this analysis, the CARA team is currently studying the credibility of collision as both a risk assessment metric and a tracking sensor priority metric.)

COLLISION PROBABILITY CALCULATION METHODS

This section summarizes each P_c method used in the multistep algorithm. For convenience, the 3D-N_c description precedes that of the 2D-N_c method, even though the algorithm invokes these methods in the opposite order.

The 2D-P_c Method

The 2D-P_c method of approximating collision probabilities, introduced in 1992 by Foster and Estes,[3] uses the approximations of rectilinear motion, constant velocities and constant relative position covariance matrices for each analyzed conjunction. Both the formulation and computational efficiency of this widely used method have improved significantly over the past decades.[4-7] Calculating 2D-P_c estimates requires two-dimensional integration over the HBR circle projection on the conjunction encounter plane. In 2005, Alfano[5] evaluated one of the integrals analytically by aligning the conjunction plane's axes with the principal axes of the marginalized joint PDF ellipse, yielding the following approximation

$$P_c \approx \frac{1}{\sqrt{8\pi}\,\sigma_x} \int_{-R}^{R} \{\text{erf}(Y_+) - \text{erf}(Y_-)\} \left[\exp\left(-\frac{(x + x_m)^2}{2\sigma_x^2} \right) \right] dx \tag{4}$$

with $Y_{\pm} = (y_m \pm \sqrt{R^2 - x^2})/(2\sigma_y)$. In this expression, (x_m, y_m) denote the close-approach miss location on the conjunction plane, and (σ_x, σ_y) the semi-major and semi-minor axes of the PDF's 1-sigma ellipse, respectively. The factor in the curly brackets represents the difference of error functions: $\Delta = \text{erf}(Y_+) - \text{erf}(Y_-)$. In many software environments (e.g., Matlab), this factor often can be computed significantly more accurately using complementary error functions, i.e., $\Delta = \text{erfc}(Y_-) - \text{erfc}(Y_+)$, especially when computing small P_c values.[7] For most conjunctions, Gauss-Chebyshev quadrature provides an efficient and accurate means to compute the integral in eq. (4), using an approach similar to that described in 2019 by Elrod.[7]

Monte Carlo studies show that 2D-P_c estimates can be inaccurate for some low relative velocity (i.e., extended duration) encounters,[8,9] and for a small fraction of high velocity encounters that are strongly affected by curvilinear trajectory effects – with the latter often associated with long propagations or highly eccentric orbits.[9,10] These discrepancies do not result from inaccurate calculation of the integral in eq. (4), but rather because one or more of the 2D-P_c assumptions (i.e., rectilinear motion, constant velocity or constant covariance) fails to be satisfied.

The 3D-N_c Method

The 3D-N_c method explicitly accounts for the effects of curvilinear trajectories along with associated velocity and covariance variations, as formulated in 2021 by Hall[10] (and briefly summarized in this section). To accomplish this, the 3D-N_c method approximates trajectories during individual conjunctions using two-body equations of motion[23] and assumes that the equinoctial element orbital states[24,25] of the primary and secondary objects each have a normally distributed uncertainty distribution at time of closest approach (TCA). More specifically, the method approximates each equinoctial state PDF at TCA using a single MVN function, an approach demonstrated by MC studies to mitigate P_c inaccuracies due to non-Gaussian covariance propagation effects.[9,26,27] For high velocity conjunctions negligibly affected by curvilinear trajectory effects, this 3D-N_c approach consistently produces P_c estimates that accurately match those calculated using the 2D-P_c method.[10]

The 3D-N_c method estimates each conjunction's P_c value by integrating the statistically expected collision rate over the duration of the encounter. The formulation results in a final expression that requires three dimensions of numerical integration, one in time, and two over the surface of the collision sphere

$$P_c \approx \int_{\tau_a}^{\tau_b} \left\{ R^2 \int_0^{2\pi} \int_0^{\pi} \mathcal{F}(\phi, \theta, R, t) \sin(\theta)\, d\theta\, d\phi \right\} dt \tag{5}$$

The integrand function represents the product of an MVN function and an averaged projected velocity[10]

$$\mathcal{F}(t, R, \phi, \theta) = \mathcal{N}\left(R\hat{\boldsymbol{r}}, \breve{\boldsymbol{r}}_t, \widetilde{\boldsymbol{A}}_t\right) \nu\left(R, \hat{\boldsymbol{r}}, \breve{\boldsymbol{X}}_t, \widetilde{\boldsymbol{P}}_t\right) \tag{6}$$

The arguments for these functions include the combined HBR, R; the radial unit vector measured from the center position of the primary satellite, $\hat{\boldsymbol{r}} = [\cos(\phi)\sin(\theta),\ \sin(\phi)\sin(\theta), \cos(\theta)]^T$; and an inertial frame primary-to-secondary position/velocity vector and associated covariance matrix

$$\breve{\boldsymbol{X}}_t = \begin{bmatrix} \breve{\boldsymbol{r}}_t \\ \breve{\boldsymbol{v}}_t \end{bmatrix} \quad \text{and} \quad \widetilde{\boldsymbol{P}}_t = \begin{bmatrix} \widetilde{\boldsymbol{A}}_t & \widetilde{\boldsymbol{B}}_t^T \\ \widetilde{\boldsymbol{B}}_t & \widetilde{\boldsymbol{C}}_t \end{bmatrix} \tag{7}$$

(Note: for brevity the dependence on time, t, is listed as a subscript for these quantities.) The 6×1 vector $\breve{\boldsymbol{X}}_t$ and associated 6×6 matrix $\widetilde{\boldsymbol{P}}_t$ represent the effective mean and covariance of the relative position/velocity vector, adjusted to account for curvilinear effects with optimal accuracy, and derived by centering the linearization of the motion on the peak PDF overlap point.[10,28] Calculating

$\breve{\boldsymbol{X}}_t$ and $\widetilde{\boldsymbol{P}}_t$ requires an iterative procedure which uses as input the TCA mean state vectors and covariance matrices of the two objects.[10] An earlier approach formulated in 2012 by Coppola[29] uses the mean relative state and covariance in eqs. (5) and (6) – i.e., $\overline{\boldsymbol{X}}_t$ and $\overline{\boldsymbol{P}}_t$ instead of $\breve{\boldsymbol{X}}_t$ and $\widetilde{\boldsymbol{P}}_t$; however, MC studies indicate that this simpler method produces significantly less accurate P_c estimates for many conjunctions than either the 2D-P_c or 3D-N_c methods.[10] (Note: preliminary analysis indicates that using $\breve{\boldsymbol{X}}_t$ and $\widetilde{\boldsymbol{P}}_t$ instead of $\overline{\boldsymbol{X}}_t$ and $\overline{\boldsymbol{P}}_t$ also improves credibility of collision[17] estimation accuracy.)

The quantity in the curly brackets of eq. (5) represents the time-dependent statistically expected collision rate, calculated as an integral over the surface of the collision sphere, which has radius R and is centered on the position of the primary object. (Chan[30,31] also discusses the concept of the collision sphere.) Calculating the collision rate entails evaluating a two-dimensional unit sphere integral over the azimuthal and axial angles (ϕ, θ), respectively, computed efficiently as a single sum using Lebedev quadrature.[32] (By default, all unit sphere integrations in this analysis use the maximum available 5,810 Lebedev quadrature points.) To calculate the time integral, CARA's 3D-N_c function computes a time series of Lebedev sums, using a dynamic algorithm to refine the number and spacing of the temporal integration steps in order to achieve a desired convergence tolerance. The limits for the integration, $\tau_a \leq t < \tau_b$, span the duration of the conjunction, i.e., the interval that the collision rate is large enough to contribute appreciably to the total P_c value.[10] Most LEO conjunctions have very short durations,[33] meaning that $\Delta\tau = \tau_b - \tau_a$ is small compared to the minimum orbital period of the two satellites. However, for some low velocity multi-encounter interactions, $\Delta\tau$ durations become so extended that sequential conjunctions potentially "blend" in time with one another; these rare blended interactions sometimes require BFMC estimation to ensure accurate P_c estimation.

Extensive testing indicates that the 3D-N_c method's three-dimensional integration approach accounts accurately for the curvilinear trajectory effects and time varying covariance dynamics that occur in both high and low velocity LEO interactions, as long as the latter are not blended in time.

The 2D-N_c Method

The 2D-N_c method is a computationally efficient version of the 3D-N_c method. The efficiency results from evaluating the time integral analytically by approximating the integrand using a 2nd order series expansion method. The derivation proceeds as follows: The first step shifts the time integration in eq. (5) to be the innermost integral. The second step consolidates the integrand's dependence on time with the following approximation (which adopts subscript notation for the variables R and t):

$$\mathcal{F}(\phi, \theta, R, t) = \mathcal{F}_{R,t}(\hat{\boldsymbol{r}}) \approx \left(\frac{\nu_{R,T}(\hat{\boldsymbol{r}})}{(2\pi)^{3/2} \left| \overline{\boldsymbol{A}}_{T_c} \right|^{1/2}} \right) e^{-\mathcal{M}_{R,t}(\hat{\boldsymbol{r}})/2} \tag{8}$$

The modified Mahalanobis distance (MMD) function is given by

$$\mathcal{M}_{R,t}(\hat{\boldsymbol{r}}) = (R\hat{\boldsymbol{r}} - \breve{\boldsymbol{r}}_t)^T \left[\widetilde{\boldsymbol{A}}_t^{-1} \right] (R\hat{\boldsymbol{r}} - \breve{\boldsymbol{r}}_t) + \ln\left(\left| \widetilde{\boldsymbol{A}}_t \right| / \left| \overline{\boldsymbol{A}}_{T_c} \right| \right) \tag{9}$$

The first term of $\mathcal{M}_{R,t}(\hat{\boldsymbol{r}})$ represents the squared effective Mahalanobis distance for a point on the collision sphere, and the second accounts for the time dependence of the relative position covariance matrix $\widetilde{\boldsymbol{A}}_t$. Eq. (8) introduces two reference times, T_c and T. The first denotes the TCA, i.e., the instant during the conjunction that the mean relative distance achieves minimum value, given by $T_c = \text{argmin}(|\bar{\boldsymbol{r}}_t|)$. The second time, T, indicates the instant of minimum MMD at the center

of the collision sphere, i.e., $T = \text{argmin}(\mathcal{M}_{0,t})$. For most conjunctions, these two times are somewhat offset from one another, and T can be determined using a bisection search or a similar numerical method.[34] Eq. (8) approximates the integrand's slowly varying velocity factor by neglecting changes from the value at time T, i.e., $\nu\left(R, \hat{\boldsymbol{r}}, \widetilde{\boldsymbol{X}}_t, \widetilde{\boldsymbol{P}}_t\right) \approx \nu\left(R, \hat{\boldsymbol{r}}, \widetilde{\boldsymbol{X}}_T, \widetilde{\boldsymbol{P}}_T\right) = \nu_{R,T}(\hat{\boldsymbol{r}})$.

The third derivation step performs a 2nd order series expansion of the $\mathcal{M}_{R,t}(\hat{\boldsymbol{r}})$ function, centered on the time T, yielding the following approximation (which suppresses the dependence on $\hat{\boldsymbol{r}}$ for brevity)

$$\mathcal{M}_{R,t} \approx \mathcal{M}_{R,T_*} + \left(\frac{t - T_*}{\omega_{R,T}}\right)^2 \quad \text{with} \quad T_* = T - \frac{\dot{\mathcal{M}}_{R,T}}{\ddot{\mathcal{M}}_{R,T}} \tag{10}$$

and

$$\mathcal{M}_{R,T_*} = \mathcal{M}_{R,T} - \frac{\dot{\mathcal{M}}_{R,T}^2}{2\ddot{\mathcal{M}}_{R,T}} \quad \text{and} \quad \omega_{R,T}^2 = \frac{2}{\ddot{\mathcal{M}}_{R,T}} \tag{11}$$

The time derivatives $\dot{\mathcal{M}}_{R,T}$ and $\ddot{\mathcal{M}}_{R,T}$ can be estimated numerically by applying the chain rule to eq. (9) and using finite differencing[34] to approximate the derivatives in the resulting expressions.

The fourth and final derivation step combines eqs. (5) and (8)-(11), allowing the time integral to be evaluated analytically, which yields the final 2D-N_c method approximation for the collision probability

$$P_c \approx \frac{R^2}{2\pi\left|\overline{\boldsymbol{A}}_{T_c}\right|^{1/2}} \int_0^{2\pi} \int_0^{\pi} \left\{\left(\nu_{R,T}\right)\left(\omega_{R,T}\right)\left[\exp\left(-\frac{\mathcal{M}_{R,T_*}}{2}\right)\right]\right\} \sin(\theta)\, d\theta\, d\phi \tag{12}$$

Note that each factor in the curly brackets depends on $\hat{\boldsymbol{r}}$, and in turn on ϕ and θ. Calculating 2D-N_c estimates requires only one unit sphere integration, again computed efficiently with Lebedev quadrature. Testing indicates that eq. (12) accurately accounts for curvilinear trajectory and covariance variation effects for the vast majority of CARA conjunctions, except for some with very low relative velocities. Specifically, for some low velocity (but unblended) conjunctions, the 2nd order approximation used in eqs. (10) and (11) potentially becomes inaccurate; these relatively rare events require using the 3D-N_c method. However, as before, the even rarer blended low velocity interactions further require BFMC estimation.

For sufficiently small HBR values, the time T closely matches the instant that the collision rate attains its peak value during the conjunction. This small-HBR limit corresponds to $R \ll \min_i\left(\sqrt{\lambda_i}\right)$, with $\{\lambda_1, \lambda_2, \lambda_3\}$ representing the eigenvalues of the $\widetilde{\boldsymbol{A}}_T$ covariance matrix. Further neglecting velocity uncertainties and position-velocity correlations (i.e., approximating $\widetilde{\boldsymbol{C}}_T$ and $\widetilde{\boldsymbol{B}}_T$ in eq. (7) as 3×3 zero matrices, respectively) allows the unit sphere integrals in eq. (12) to be evaluated analytically, yielding the following "small-HBR" approximation for the collision probability

$$P_c \approx \left(\frac{R^2 |\breve{\boldsymbol{v}}_T| \omega_{0,T}}{2\left|\overline{\boldsymbol{A}}_{T_c}\right|^{1/2}}\right) \exp\left(-\frac{\mathcal{M}_{0,T}}{2}\right) \qquad \text{for small } R, \widetilde{\boldsymbol{B}}_T, \text{and } \widetilde{\boldsymbol{C}}_T \tag{13}$$

This equation provides a convenient means of testing software implementations of the 2D-N_c method, as well as a basis to estimate a rough correction factor for 2D-P_c method estimates as described later.

The Simplified Dynamics Monte Carlo Method

The SDMC method estimates collision probabilities using a multi-trajectory simulation based on equinoctial orbital states and two body equations of motion. Specifically, SDMC (also known as two-body Monte Carlo[9,10,27]) uses a "from-TCA" approach that repetitively samples equinoctial orbital states from the Gaussian PDFs estimated for the primary and secondary objects at TCA. Propagating each sampled state forward and backward in time for short durations using the equations of two-body motion[23-25] allows estimation of the fraction that represent collisions (i.e., hits).[9,10] Testing indicates that the SDMC and 3D-N_c methods produce P_c estimates that consistently agree to one another to within expected MC statistical variations. However, SDMC usually requires significantly more computation than 3D-N_c, especially when estimating small probabilities. (For typical laptops, SDMC estimation becomes prohibitive roughly for $P_c < 10^{-7}$; for the 24-core processing system available to CARA analysts, the limit is roughly $P_c < 10^{-8}$.) For conjunctions with sufficiently large P_c values, the SDMC method provides a compelling means to illustrate how curvilinear trajectories affect close approach (CA) distributions, as shown later.

The Brute Force Monte Carlo Method

The BFMC method estimates collision probabilities using high fidelity trajectory propagations, based on full special perturbations (SP) orbital states.[9,10,27] (SP orbital states typically comprise seven- or eight-dimensional vectors that include satellite ballistic coefficients and/or solar radiation pressure parameters.[23]) Specifically, BFMC uses a "from-epoch" MC approach that repetitively samples equinoctial element-based SP orbital states from Gaussian PDFs estimated at the OD epochs of both the primary and secondary objects. Propagating each sampled SP state using high accuracy numerical integration allows estimation of the fraction that represent hits.[9,27] This high fidelity from-epoch approach allows BFMC to estimate P_c values for extended interactions that potentially involve many close encounters, such as between two closely orbiting objects over a multi-day period. For unblended conjunctions, testing indicates that the SDMC and BFMC methods consistently agree with one another to within expected statistical variations.[9,27] However, BFMC requires significantly more computation than SDMC. (For the 24-core processing system currently available to CARA analysts, BFMC estimations become prohibitive roughly for $P_c < 10^{-5}$.) BFMC also uses a relatively large set of metadata[9] (e.g., time-dependent atmospheric density parameters) that, if accessible, requires considerable effort to assemble in a timely fashion. For these reasons, the multistep algorithm recommends BFMC only for the extremely rare occurrences of blended interactions that pose sufficiently high risk to warrant the effort.

USAGE VIOLATIONS FOR COLLISION PROBABILITY CALCULATIONS

As mentioned previously, the first four P_c computation methods used by the multistep algorithm each potentially provide inaccurate estimates under certain conditions. This section summarizes usage violation criteria that indicate the potential for the P_c estimates to be insufficiently inaccurate for reliable risk assessment.

2D-P_c Method Usage Violation Analysis

The multistep algorithm detects potential 2D-P_c inaccuracies using four types of usage violations:

1. "NPD" violations indicate if any of the position covariance matrices are not positive definite.
2. "Offset" violations indicate if the time between the TCA and the time of minimum Mahalanobis distance for the rectilinear encounter is potentially too long (i.e., exceeds a conservative threshold value, as described below).

3. "Extended" violations indicate if the rectilinear encounter duration is potentially too extended.
4. "Inaccuracy" violations indicate if the Mahalanobis distances for the rectilinear vs curvilinear encounters potentially differ by too much.

To check for non-positive definite (NPD) covariance matrix violations, the algorithm calculates eigenvalues for the inertial position covariance matrices of both the primary and secondary objects at TCA. It also calculates eigenvalues for the relative position (or joint) covariance matrix at TCA, which may have been corrected for the effects of orbital error cross correlation.[10,35] The algorithm registers an NPD usage violation indicator if any of these nine eigenvalues are not positive.

To check for relative velocities that are too low to satisfy the 2D-P_c method's assumptions of rectilinear motion, constant velocity or constant covariance, the algorithm calculates two scalars: one to indicate if the peak collision rate occurs too far in time from the TCA, and another to indicate if the conjunction extends too much in duration. During a rectilinear encounter with constant velocity, the mean relative position varies linearly in time: $\bar{\boldsymbol{r}}_t = \bar{\boldsymbol{r}}_{T_c} + \bar{\boldsymbol{v}}_{T_c}(t - T_c) = \bar{\boldsymbol{r}}_{T_c} + \bar{\boldsymbol{v}}_{T_c}(\delta t)$. Assuming constant covariance, the squared Mahalanobis distance at the center of collision sphere varies quadratically in time

$$M'_t = \mathcal{M}'_{0,t} = \bar{\boldsymbol{r}}_t^T \left[\bar{\boldsymbol{A}}_{T_c}^{-1}\right] \bar{\boldsymbol{r}}_t = a(\delta t)^2 + b(\delta t) + c \tag{14}$$

with $a = \bar{\boldsymbol{v}}_{T_c}^T \left[\bar{\boldsymbol{A}}_{T_c}^{-1}\right] \bar{\boldsymbol{v}}_{T_c}$, $b = 2\bar{\boldsymbol{r}}_{T_c}^T \left[\bar{\boldsymbol{A}}_{T_c}^{-1}\right] \bar{\boldsymbol{v}}_{T_c}$, and $c = \bar{\boldsymbol{r}}_{T_c}^T \left[\bar{\boldsymbol{A}}_{T_c}^{-1}\right] \bar{\boldsymbol{r}}_{T_c}$. (Note: primes distinguish quantities derived using the rectilinear motion approximation from those derived for curvilinear motion.) The minimum occurs at the time $T' = T_c - b/(2a)$. For small HBR values, T' closely corresponds to the instant that the collision rate attains its peak value during a rectilinear encounter, which is analogous to the time T for a curvilinear encounter used in eqs. (9)-(12). To check if T' is too far from TCA for the 2D-P_c assumptions to hold reliably, the algorithm compares $\delta T' = T' - T_c$ to the minimum orbital period of the two objects, p_{min}. Specifically, if the offset violation indicator $V_o = \delta T'/p_{min}$ is greater than a threshold of 0.01, the algorithm registers a 2D-P_c offset violation. This threshold has been selected conservatively so that, in combination with the other thresholds discussed below, the analysis produces no failed detections of actual usage violations among the 1.79 million conjunctions tested in this study. Since orbital periods for LEO objects are ~100 minutes, this threshold registers offset violations for $\delta T'$ values longer than ~60 s, which occur relatively rarely.[33]

Expressing the rectilinear encounter Mahalanobis distance as $M'_t = M'_{T'} + [(t - T')/w']^2$ indicates a 1-sigma width in time of $w' = (\bar{\boldsymbol{v}}_{T_c}^T \bar{\boldsymbol{A}}_{T_c}^{-1} \bar{\boldsymbol{v}}_{T_c})^{-1/2}$. This width corresponds to a conjunction duration[33] of $\Delta\tau' = \left[2\sqrt{2}\ \mathrm{erfc}^{-1}(\gamma)\right] w'$, with γ corresponding to the small fraction of P_c that accumulates outside the duration bounds. (This analysis defines durations using $\gamma = 10^{-6}$, corresponding to $\Delta\tau' \approx 9.8w'$.) The rectilinear conjunction bounds are $\tau'_a = T' - \Delta\tau'/2$ and $\tau'_b = T' + \Delta\tau'/2$. To check if the 2D-$P_c$ assumptions hold reliably, the algorithm compares the extended violation indicator $V_e = \Delta\tau'/p_{min}$ to a threshold of 0.02. If V_e is greater than this limit, then the algorithm registers a 2D-P_c extended duration usage violation. Again, this selected threshold produces no failed detections among the test conjunctions, and registers violations for $\Delta\tau'$ values longer than ~120 s, which rarely occur for LEO satellites.[33]

The fourth and final usage violation check for the 2D-P_c method compares Mahalanobis distances for the rectilinear and curvilinear trajectories. More specifically, the algorithm aims to detect any significant differences between two temporal MMD curves: one calculated for the constant-velocity and constant-covariance rectilinear trajectory approximation, given by eq. (14); and the other for the two-body motion curvilinear trajectory approximation, for which the velocity and

covariance vary. Setting $R = 0$ in eq. (9) yields the MMD at the center of collision sphere during a curvilinear encounter

$$M_t = \mathcal{M}_{0,t} = \breve{\boldsymbol{r}}_t^T \left[\widetilde{\boldsymbol{A}}_t^{-1}\right] \breve{\boldsymbol{r}}_t + \ln\left(\left|\widetilde{\boldsymbol{A}}_t\right| / \left|\overline{\boldsymbol{A}}_{T_c}\right|\right) \tag{15}$$

If the M_t' and M_t functions given by eqs. (14) and (15) differ significantly during the encounter, then the 2D-P_c method could potentially provide an inaccurate estimate. Figure 2 schematically illustrates the two functions, showing M_t' as the dotted blue curve and M_t as the solid black curve. To determine if the two curves differ significantly, the algorithm calculates the following quantities for the curvilinear encounter: the minimum, $M_{min} = M_T = \mathcal{M}_{0,T}$, and the corresponding 1-sigma width in time, $w = \mathcal{w}_{0,T}$. Calculating these requires an iterative numerical process, but less computation than executing the 2D-N_c method. Calculating the analogous quantities for the rectilinear encounter, M'_{min} and w', requires relatively little computation.

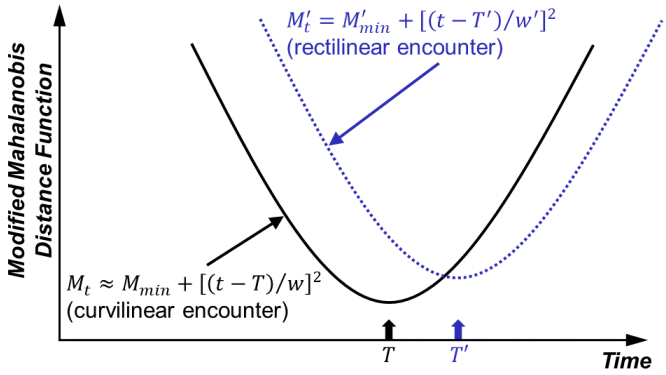

Figure 2. Comparison of modified Mahalanobis distance functions during a conjunction.

Combining the Mahalanobis curve parameters with the small-HBR approximation of eq. (13), provides a rough correction scale factor for 2D-P_c estimate, F_c, expressed as a natural logarithm as follows

$$\ln(F_c) = \ln\left(\frac{P_c}{P_c'}\right) \approx \ln\left(\frac{\breve{v}_T}{\bar{v}_{T_c}}\right) + \ln\left(\frac{w}{w'}\right) - \left[\frac{M_{min} - M'_{min}}{2}\right] \tag{16}$$

with P_c denoting the probability for the actual curvilinear encounter, and P_c' the probability for the 2D-Pc approximation based on rectilinear trajectories. To check for potential 2D-P_c inaccuracies, the algorithm calculates an inaccuracy usage violation indicator, $V_i = 1 - \exp[-|\ln(F_c)|]$, and registers a violation if V_i exceeds a threshold of 0.02. In the absence of any other usage violations, testing confirms that for $V_i \leq 0.02$ the 2D-P_c method provides estimates within ~2% of the refined P_c values, as calculated using the subsequent more accurate estimation methods (i.e., 2D-N_c, 3D-N_c, etc.). The correction factor F_c can be less than or greater than one, depending on the specific conjunction. Specifically, $F_c < 1$ indicates that the rectilinear 2D-P_c approximation overestimates the actual P_c value that accounts for curvilinear effects. Similarly, $F_c > 1$ indicates 2D-P_c underestimation.

Testing also indicates that, in general, most screened CARA conjunctions ultimately are found to have collision probability estimates that are negligibly small, i.e., $P_c < 10^{-10}$. In order to prevent needless processing for the majority of these low risk conjunctions, the multistep algorithm calculates an initial approximation by applying the scale factor, i.e., $P_c'' = F_c\, P_c'$. This scaled estimate combines the 2D-P_c method estimate P_c' given by eq. (4), and the correction factor F_c given by eq. (16) – which together require significantly less computation than simply executing the 2D-N_c

method. Testing verifies that, among the set of test conjunctions with no NPD, offset or extended usage violations, P_c'' values closely match the fully refined estimates for cases that satisfy the assumptions used for the small-HBR approximation of eq. (13). However, for those that do not rigorously satisfy these assumptions, P_c'' still provides an approximation that, although rough, allows safe and efficient processing of very low risk conjunctions. Specifically, among the set of test conjunctions that have $P_c'' < 10^{-15}$ and only have inaccuracy usage violations, testing indicates no instances with actual refined estimates $P_c > 2\times10^{-14}$, which is well below the negligibly small level of $P_c = 10^{-10}$. On this basis, in the absence of any other 2D-P_c usage violations, the multistep algorithm conservatively reports the quantity $\max(P_c', P_c'')$ as the refined estimate for conjunctions with $V_i > 0.02$ and $P_c'' < 10^{-15}$. For conjunctions with $V_i > 0.02$ and $P_c'' \geq 10^{-15}$ (or that have any other 2D-P_c method violations), the algorithm proceeds on to the 2D-N_c method, and then to the other methods as required, to calculate refined P_c estimates.

In summary, the multistep algorithm advances past 2D-P_c only for conjunctions that register an NPD, offset or extended usage violation, or that have an inaccuracy violation and a scaled 2D-P_c estimate that exceeds a conservative threshold, i.e., $P_c'' \geq 10^{-15}$. This leads to the algorithm to report 2D-P_c based estimates for 94.7% of the test conjunctions.

2D-N_c Method Usage Violation Analysis

The algorithm detects potential 2D-N_c inaccuracies using four types of violations:

1. "Convergence" violations indicate if the iterative process failed to converge on the minimum Mahalanobis distance time, T, or failed to find the associated $(\breve{\boldsymbol{X}}_T, \widetilde{\boldsymbol{P}}_T)$ values.
2. "Offset" violations indicate if the offset between the TCA and T is potentially too long.
3. "Extended" violations indicate if the curvilinear conjunction duration $\Delta\tau$ is potentially too long.
4. "Inaccuracy" violations indicate if the 2nd order series used for eqs. (10) and (11) potentially becomes inaccurate.

The specific indicators and associated (conservatively selected) threshold values for these 2D-N_c usage violations are similar in character to those described previously the for the 2D-P_c method, but not described in further detail here for brevity.

3D-N_c Method Usage Violation Analysis

The algorithm also detects potential 3D-N_c inaccuracies using four types of violations:

1. "Convergence" violations indicate if the iterative process failed to calculated the required $(\breve{\boldsymbol{X}}_t, \widetilde{\boldsymbol{P}}_t)$ values at any point during the duration of the conjunction.
2. "Offset" violations indicate if the offset between the TCA and the time of peak collision rate (numerically estimated from the temporal collision rate curve) is potentially too long.
3. "Extended" violations indicate if the conjunction duration is potentially too long.
4. "Inaccuracy" violations indicate if the difference between the 2D-N_c and 3D-N_c estimates is unexpectedly large.

Again, these violation indicators are similar in character to those described previously.

SDMC Method Usage Violation Analysis

The multistep algorithm detects potential SDMC inaccuracies using two types of usage violations. Specifically, "offset" SDMC violations use exactly the same criteria as used by the 3D-Nc method's offset violations. In addition, "extended" SDMC violations indicate if any MC hits occur at or near the edges of the conjunction's full encounter segment, as defined by Hall.[10] These edge-

case hits indicate potentially blended conjunctions, which require BFMC to ensure accurate P_c estimation.

COMPARISON OF COLLISION PROBABILITIES AND USAGE VIOLATIONS

As mentioned previously, the multistep algorithm optionally allows automated processing using any or all of the first four P_c estimation methods, 2D-P_c, 2D-N_c, 3D-N_c and SDMC, respectively. In order to compare estimates from these methods, this study performed a two-pass testing analysis. The first pass processed all 1.79 million CARA conjunctions using each of the first three methods. The second pass executed SDMC for the subset found in the first pass to have refined P_c estimates of 10^{-6} or above. (This study conducted no systematic BFMC method testing, but previous analysis indicates that, for non-blended conjunctions, the SDMC and BFMC methods consistently agree with one another to within expected statistical variations.[9,27])

Comparing the 2D-P_c and 2D-N_c Methods

The left panel of Figure 3 compares the 2D-N_c and 2D-P_c methods. Specifically, the vertical axis plots 2D-N_c estimates calculated using eq. (12) and the horizontal axis plots 2D-P_c estimates from eq. (4). Gray circles show conjunctions for which neither method registered a usage violation; the other colors indicate violations registered by one or both of the methods. For clarity, the axes only span P_c values above 10^{-20} (leftward- or downward-pointing triangles show values below this cutoff). Because all of the gray symbols line up along the diagonal, the plot indicates that the 2D-N_c and 2D-P_c estimates accurately match one another in the absence of any usage violations.

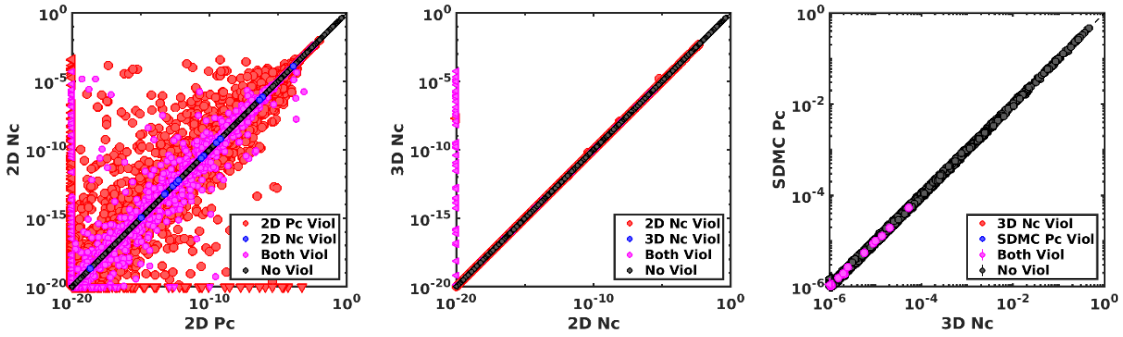

Figure 3. Comparison of 2D-N_c vs 2D-P_c method P_c estimates with usage violations (left), 3D-N_c vs 2D-N_c method estimates with usage violations (middle), and SDMC vs 3D-N_c method P_c estimates with usage violations (right).

The left panel of Figure 3 shows that the 2D-P_c usage violation algorithm correctly detects all conjunctions with significantly different 2D-N_c and 2D-P_c estimates, with no missed detections. This reflects both the comprehensive nature of the 2D-P_c usage violation criteria, and the conservative selection of the associated usage violation thresholds. However, a significant portion of estimates with 2D-P_c usage violations also line up very nearly along the diagonal in the left panel of Figure 3. These represent "false alarms" elicited by the 2D-P_c usage violation algorithm, which also result from the conservatively selected thresholds. Specifically, among the entire test data set, 86% of conjunctions with 2D-P_c usage violations also have 2D-P_c estimates found to be within 30% of the refined estimates – an accuracy level found to provide reasonably reliable risk assessments. The false alarm rate is even larger among high-risk events with refined estimates of $P_c \geq 10^{-4}$, for which 94% of the usage violation cases have 2D-P_c estimates accurate to within 30%. So overall, the 2D-P_c usage violation algorithm has a false alarm rate of ~90%. In other words, of the

original 5.3% of test conjunctions flagged with 2D-P_c usage violations, most (~4.8%) represent false alarms, and the remainder (~0.5%) represent actual 2D-P_c method usage violations.

The highly scattered off-diagonal population in the left panel of Figure 3 represents the most discrepant 2D-N_c and 2D-P_c estimates. Notably, scaling these values using the rough correction factor F_c from eq. (16) reduces the level of scatter considerably, but does not eliminate it altogether. In addition, the points significantly above the diagonal line on the 2D-N_c vs 2D-P_c plot tend to correspond to conjunctions with $F_c \gg 1$, and those well below the diagonal tend to have $F_c \ll 1$.

Comparing the 3D-N_c and 2D-N_c Methods

The middle panel of Figure 3 compares 3D-N_c and 2D-N_c, with the vertical axis plotting 3D-N_c estimates calculated using eq. (5), and the horizontal axis 2D-N_c estimates from eq. (12). Again, all of the gray symbols line up along the diagonal, indicating that the 3D-N_c and 2D-N_c estimates accurately match one another in the absence of any usage violations. The 2D-N_c usage violation analysis correctly identifies all off-diagonal points, again reflecting conservatively selected 2D-N_c usage violation criteria. In this case, however, a much larger fraction of 2D-N_c estimates with usage violations (plotted as the red circles, which are slightly larger than the gray circles) also line up along the diagonal. As before, this high false alarm level exists because the multistep algorithm uses conservative usage violation criteria for the 2D-N_c method, with thresholds adjusted to eliminate any missed detections among the test conjunctions. The middle panel of Figure 3 has a notably smaller number of colored points overall than does the left panel, indicating that 2D-N_c produces significantly fewer usage violations than 2D-P_c among the test conjunctions. Finally, most of the off-diagonal points in middle panel of Figure 3 represent low velocity conjunctions.

Comparing the SDMC and 3D-N_c Methods

The right panel of Figure 3 compares the SDMC and the 3D-N_c methods. Specifically, the vertical axis plots SDMC P_c estimates, and the horizontal axis plots 3D-N_c estimates calculated using eq. (5). Again, gray circles show conjunctions for which neither method registered a usage violation, and the other colors indicate violations registered by one or both of the methods. (The plot includes error bars that show 95% confidence intervals[9] for the SDMC estimates, but these are usually smaller than the circle symbols themselves.) The right panel of Figure 3 indicates that the 3D-N_c and SDMC methods consistently produce results that agree to within expected statistical limits, as has been reported previously.[10]

EXAMPLE CONJUNCTION COLLISION PROBABILITY CALCULATIONS

This section shows several example output plots created by the multistep algorithm. The software optionally produces two types of graphs for each conjunction: a temporal plot that shows collision rate and cumulative probability variations; and a close approach distribution plot that shows the spatial pattern of CA points using a primary-centered format, similar to the "B-plane" plots used for planetary encounters.[9] Figures 4-8 show these plots for a selected set of conjunctions, some of which have no P_c estimation usage violations whatsoever, as well as for others that register 2D-P_c, 2D-N_c, and 3D-N_c method violations.

Figure 4 shows temporal and CA distribution plots for a conjunction with a high collision probability ($P_c \approx 0.01$) that registers no usage violations for any of the P_c estimation methods. The left panel plots the collision rate (bottom) and cumulative P_c (top) estimated using the 3D-N_c and SDMC methods, showing the expected offsets and durations, and illustrating how the final total P_c values agree with those estimated using the 2D-P_c and 2D-N_c methods. The right panel plots different views of the 3-sigma (3σ) covariance ellipse predicted using the 2D-P_c method, along with

the CA points calculated in the SDMC simulation – with the red dots representing MC hits, and the blue dots representing misses. Specifically, the top-left plot in the right panel shows a zoomed-out view of the entire CA distribution; the top-right plot shows a zoomed-in view that plots the HBR circle along with the distribution of hits; and the bottom plot shows a view zoomed in by unequal magnifications on the horizontal and vertical axes.

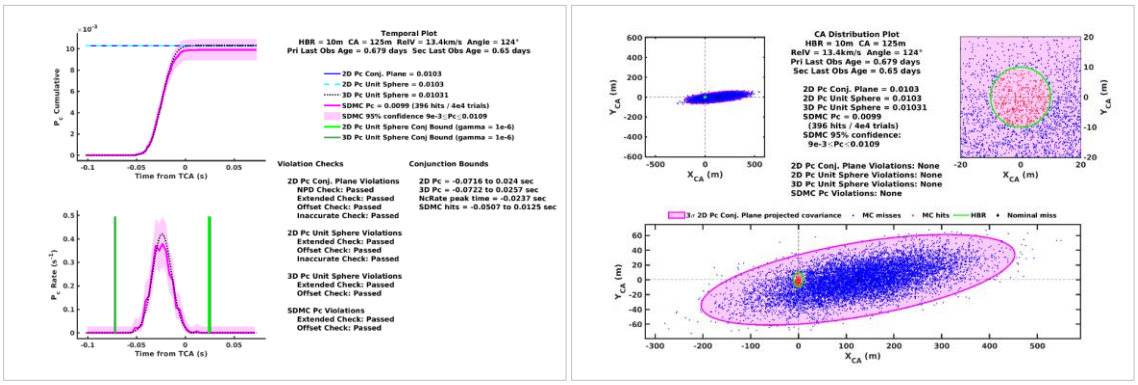

Figure 4. Temporal plot (left) and close approach distribution plot (right) for a high-P_c conjunction with no usage violations for any of the P_c estimation methods.

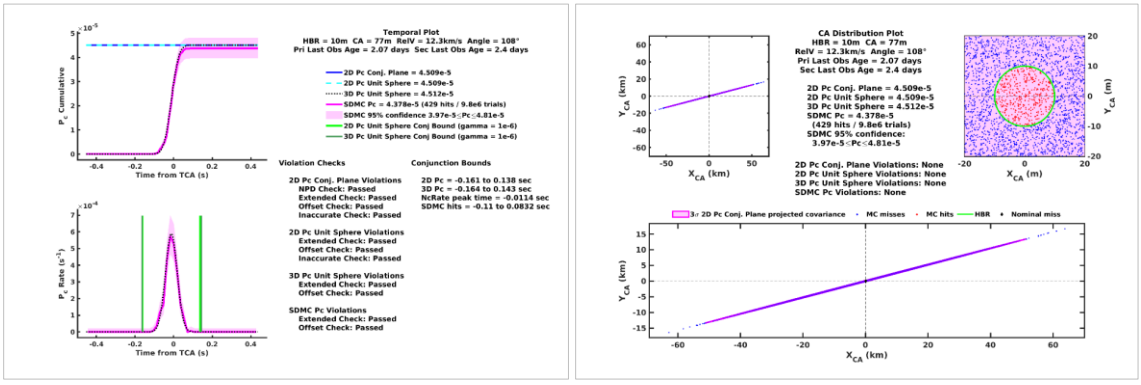

Figure 5. Temporal plot (left) and close approach distribution plot (right) for a moderate-P_c conjunction with no usage violations for any of the P_c estimation methods.

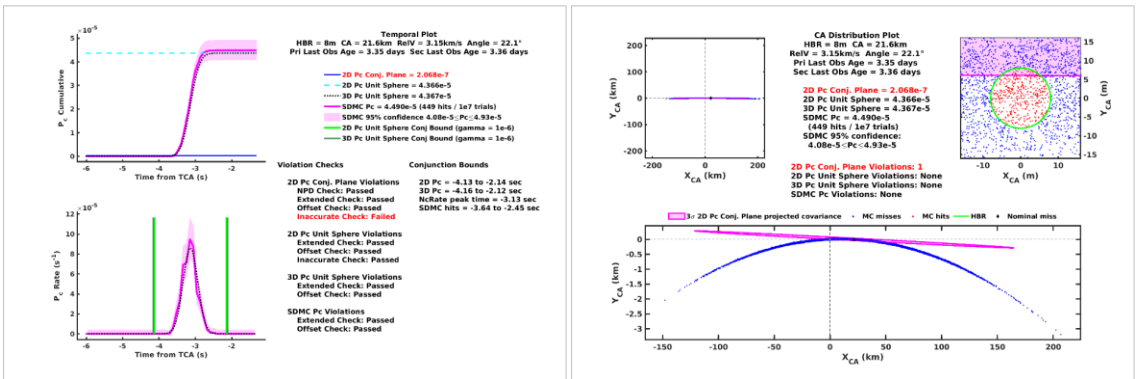

Figure 6. Temporal plot (left) and close approach distribution plot (right) for a conjunction with a 2D-P_c method usage violation.

Figure 5 shows temporal and CA plots for a conjunction with $P_c \approx 4.5\times10^{-5}$ and no usage violations for any of the estimation methods. Again, both plots illustrate how the P_c values estimated using the 2D-P_c, 2D-N_c, 3D-N_c, and SDMC methods agree with one another.

Figure 6 shows temporal variation and CA distribution plots for a conjunction with a relative velocity of ~3.1 km/s, which registers one 2D-P_c method usage violation, but no 2D-N_c, 3D-N_c, or SDMC violations. Specifically, the algorithm indicates a 2D-P_c method inaccuracy usage violation for this conjunction (i.e., with $V_i > 0.02$). Notably, for this event the 3σ ellipse for the rectilinear encounter deviates significantly from the distribution of MC hits for the curvilinear encounter. This difference explains why the 2D-P_c method erroneously reports a P_c estimate more than two orders of magnitude smaller than the more accurate, refined estimate of $P_c \approx 4.4\times10^{-5}$ (as reported by the 2D-N_c, 3D-N_c and SDMC methods).

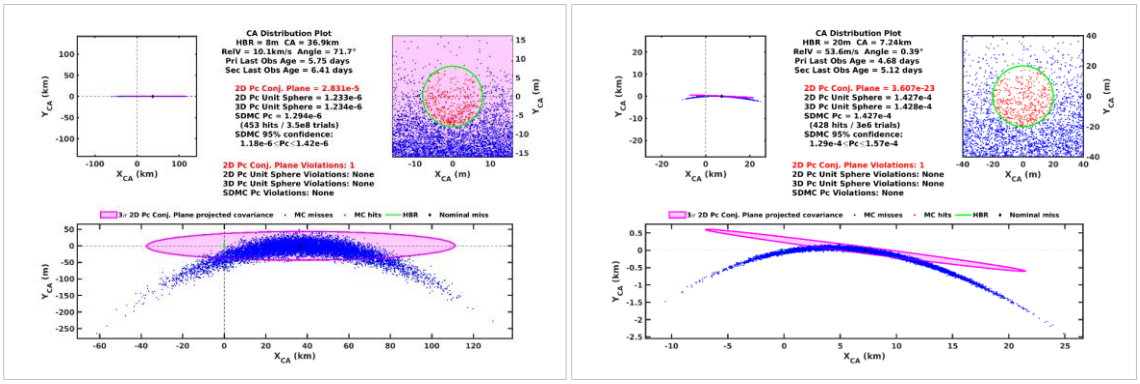

Figure 7. Close approach distributions for a high velocity conjunction with a 2D-P_c method usage violation (left), and a low velocity conjunction with a 2D-P_c method usage violation (right).

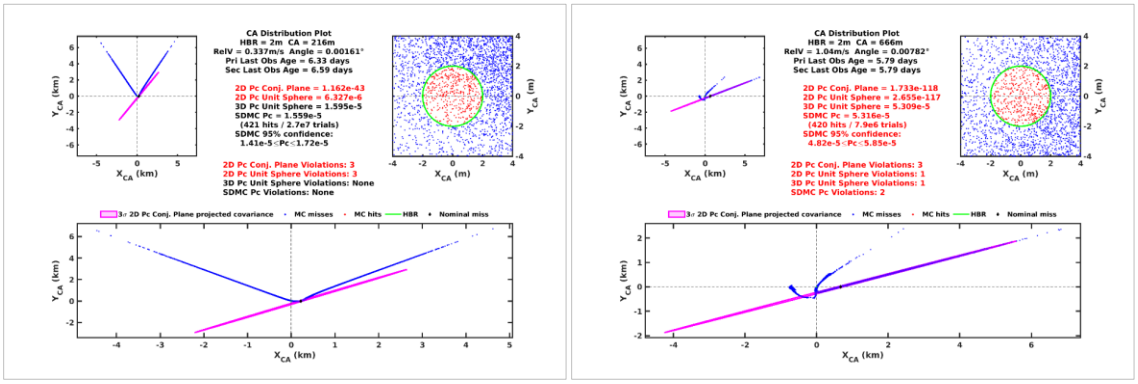

Figure 8. Close approach distributions for very low velocity conjunctions with 2D-N_c method usage violations (left) and 3D-N_c method usage violations (right).

Figure 7 shows CA distribution plots for two conjunctions that each have a 2D-P_c method usage violation, but no 2D-N_c, 3D-N_c, or SDMC violations. The left panel shows a high velocity conjunction for which the 2D-P_c method reports an erroneously high P_c estimate. The right panel shows a low velocity conjunction for which 2D-P_c reports an erroneously low estimate.

Figure 8 shows CA distribution plots for two conjunctions that each have multiple usage violations. The left panel shows an example of a conjunction with usage violations for both the 2D-P_c and 2D-N_c methods, but not for the 3D-N_c or SDMC methods, which has a very low relative ve-

locity of ~0.4 m/s. The right panel shows a conjunction with usage violations for all of the estimation methods (except BFMC), which has a comparably low velocity of ~1.0 m/s. As mentioned previously, usage violations for the more advanced 3D-N_c and SDMC methods occur almost exclusively for such very low velocity conjunctions.

CONCLUSIONS

The multistep algorithm provides an efficient and accurate means to estimate collision probabilities for all types of interactions experienced by Earth-orbiting satellites. The flexibility arises from an approach of applying several increasingly advanced P_c estimation methods. The efficiency arises by invoking these methods only as required, based on a set of usage violation criteria that detect potential P_c estimation inaccuracies. Extensive testing using ~2 million actual conjunctions demonstrates the reliability of the usage violation criteria. The algorithm processes 94.7% of CARA conjunctions using estimates based on the most efficient 2D-P_c method. It processes most of the remainder using 2D-N_c (5.0%) and 3D-N_c (0.3%), which are also semi-analytical methods. The algorithm resorts to MC methods for <0.01% of the test conjunctions, most of which represent very low relative velocity interactions.

Detecting potential 2D-P_c method usage violations represents one of the most important elements of the multistep algorithm. The comprehensive nature of the 2D-P_c usage violation criteria combined with the conservative selection of the associated violation thresholds allows the algorithm to identify all conjunctions with significant 2D-P_c estimation inaccuracies, with no missed detections. However, in order to achieve this zero missed detection rate, the resulting 2D-P_c usage violation algorithm has a false alarm rate of ~90%. In other words, of the original 5.3% of test conjunctions flagged with 2D-P_c usage violations, most (~4.8%) represent false alarms, and the remainder (~0.5%) represent actual 2D-P_c method usage violations.

In summary, when provided realistic orbital states and covariance matrices, the multistep algorithm computes accurate collision probabilities for both high and low velocity interactions, accounting for curvilinear trajectory effects and time varying covariance dynamics as required.

ACKNOWLEDGMENTS

The authors would like to thank Russell Carpenter, Joe Frisbee, Matt Hejduk, Dolan Highsmith, and Dan Snow for several helpful discussions and analyses.

NOTATION

a,b,c	coefficients for the quadratic time variation of the M_t' function – see eq. (14)
$\overline{\boldsymbol{A}}_t, \overline{\boldsymbol{B}}_t, \overline{\boldsymbol{C}}_t$	the 3×3 submatrices of the 6×6 covariance matrix $\overline{\boldsymbol{P}}_t$ – similar to the form given in eq. (7)
$\widetilde{\boldsymbol{A}}_t, \widetilde{\boldsymbol{B}}_t, \widetilde{\boldsymbol{C}}_t$	the 3×3 submatrices of the 6×6 covariance matrix $\widetilde{\boldsymbol{P}}_t$ – see eq. (7)
$\mathcal{F}$	integrand function of the three-dimensional P_c integral used in the 3D-N_c method
F_c	the rough multiplicative correction factor for 2D-P_c method estimates, see eq. (16)
g_1, g_2	indices for Gaussian mixture model components for the primary and secondary satellites
k	index for one isolated conjunction within a multi-encounter interaction
$\mathcal{M}_{R,t}(\hat{\boldsymbol{r}})$	the MMD for a point on the collision sphere for a curvilinear encounter – see eq. (9)
$\mathcal{M}'_{R,t}(\hat{\boldsymbol{r}})$	the MMD for a point on the collision sphere for a rectilinear encounter

M_t	the MMD at the center of the collision sphere for a curvilinear encounter, i.e., $M_t = \mathcal{M}_{0,t}$
M_t'	the MMD at the center of the collision sphere for a rectilinear encounter, i.e., $M_t' = \mathcal{M}_{0,t}'$
N_c	the statistically expected number of collisions
$\mathcal{N}(\boldsymbol{r}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{A}})$	the MVN function, i.e., $\mathcal{N}(\boldsymbol{r}, \bar{\boldsymbol{r}}, \bar{\boldsymbol{A}}) = \exp[-(\boldsymbol{r}-\bar{\boldsymbol{r}})^T \bar{\boldsymbol{A}}^{-1}(\boldsymbol{r}-\bar{\boldsymbol{r}})/2]/\sqrt{(2\pi)^3\lvert\bar{\boldsymbol{A}}\rvert}$
P_c	the probability of collision
P_c'	the P_c calculated for a rectilinear encounter, i.e. the estimate calculated with the 2D-Pc method
P_c''	the scaled 2D-P_c estimate, roughly corrected for curvilinear trajectory effects, $P_c'' = F_c\, P_c'$
$\hat{\boldsymbol{r}}$	unit vector pointing to one location on the unit sphere
$\bar{\boldsymbol{r}}_t, \bar{\boldsymbol{v}}_t$	the time-dependent OD based mean primary-to-secondary (i.e., relative) position and velocity vectors
$\breve{\boldsymbol{r}}_t, \breve{\boldsymbol{v}}_t$	the time-dependent <u>effective</u> mean relative position and velocity vectors for a curvilinear encounter, calculated using the iterative method described in detail by Hall[10]
R	the combined primary+secondary hard-body radius, i.e., $R = R_1 + R_2$
$\bar{R}_2, \sigma_{R_2}^2$	the mean and variance of an RCS-based hard-body radius of the secondary object
t	time
T	the time that the MMD function attains its minimum value for a curvilinear encounter
T'	the time of minimum MMD for a rectilinear encounter, as indicated by the prime
T_c	the nominal time of closest approach for a conjunction
T_*	the time of minimum MMD for a point on the collision sphere
ν	the averaged projected velocity function at a point on the unit sphere – see eq. (6)
$w_{R,T}$	the 1-sigma time width of the MMD function at a point on the collision sphere for a curvilinear encounter
$w_{R,T}'$	the 1-sigma MMD time width at a point on the collision sphere for a rectilinear encounter
w	the 1-sigma MMD time width at the center of the collision sphere for a curvilinear encounter, i.e., $w = w_{0,T}$
w'	the 1-sigma MMD time width at collision sphere center for a rectilinear encounter
x_m, y_m	conjunction plane miss distance coordinates for a rectilinear encounter
$\bar{\boldsymbol{X}}_t, \bar{\boldsymbol{P}}_t$	the time-dependent mean relative position/velocity state vector and covariance matrix
$\breve{\boldsymbol{X}}_t, \breve{\boldsymbol{P}}_t$	the time-dependent effective mean relative pos./vel. state and covariance for a curvilinear encounter
λ_i	the i^{th} eigenvalue of the $\widetilde{\boldsymbol{A}}_T$ covariance matrix
σ_x, σ_y	conjunction plane 1-sigma values of the marginalized joint relative position PDF
τ_a, τ_b	begin and end times for the conjunction duration
ϕ, θ	azimuthal and axial angles that span the unit sphere
CARA	Conjunction Assessment Risk Analysis
CA	close approach

HBR	hard-body radius
LEO	low Earth orbit
MMD	modified Mahalanobis distance
MVN	multi-variate normal
PDF	probability density function
RCS	radar cross section
TCA	time of closest approach

REFERENCES

[1] Newman, L. K., "The NASA Robotic Conjunction Assessment Process: Overview and Operational Experiences," *Acta Astronautica*, Vol. 66, 2010, pp. 1253-1261.
[2] Tapley, B. D., Schutz, B. E., and Born, G. H., *Statistical Orbit Determination*, Elsevier Academic Press, Burlington, MA, 2004.
[3] J. L. Foster and H. S. Estes, "A Parametric Analysis of Orbital Debris Collision Probability and Maneuver Rate for Space Vehicles," NASA/JSC-25898, Aug. 1992.
[4] M. R. Akella and K. T. Alfriend, "The Probability of Collision Between Space Objects," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 5, pp. 769-772, 2000.
[5] S. Alfano, "A Numerical Implementation of Spherical Object Collision Probability." *Journal of the Astronautical Sciences*, Vol. 53 No. 1, pp. 103-109, 2005.
[6] K. Chan, *Spacecraft Collision Probability*, El Segundo, CA, The AeroSpace Corporation, 2008.
[7] C. Elrod, "Computational Bayesian Methods Applied to Complex Problems in Bio and Astro Statistics." Doctoral Dissertation, Baylor University, July 2019.
[8] S. Alfano, "Satellite Conjunction Monte Carlo Analysis," *AAS SpaceFlight Mechanics Meeting*, Pittsburgh, PA, Paper 09-233, Feb. 2009.
[9] D. T. Hall , Casali, S. J., Johnson, L.C., Skrehart, B. B., and Baars, L. G., "High Fidelity Collision Probabilities Estimated Using Brute Force Monte Carlo Simulations," *AAS Astrodynamics Specialist Conference*, Snowbird, UT, Paper 18-244, Aug. 2018.
[10] D. T. Hall, "Expected Collision Rates for Tracked Satellites," *Journal of Spacecraft and Rockets*, Vol. 58, No. 3, pp. 715-728, May-June, 2021.
[11] Mahalanobis, P. C., "On the Generalised Distance in Statistics," *Proc. of the National Institute of Sciences in India*, Vol. 2, No. 1, pp. 49-55, 1936.
[12] Mashiku, A. K. and Hejduk, M. D., "Recommended Methods for Setting Mission Conjunction Analysis Hard Body Radii," *AAS Astrodynamics Specialist Conference*, Portland, ME, Paper AAS 19-702, 2019.
[13] "Top Ten Satellite Breakups", Orbital Debris Quarterly News, NASA's Orbital Debris Program Office, July 2010.
[14] Baars, L. G., and Hall. D. T., "Processing Space Fence Radar Cross-Section Data to Produce Size and Mass Estimates," *AAS Astrodynamics Specialist Conference*, Charlotte, NC, Paper 22-586, Aug. 2022.
[15] Hall, D. T., and Baars, L. G., "Satellite Collision and Fragmentation Probabilities Using Radar Based Size and Mass Estimates," *Journal of Spacecraft and Rockets*, in press, May 2023.
[16] Poore, A. B., Aristoff, J. M., and Horwood, J. T., "Covariance and Uncertainty Realism in Space Surveillance and Tracking," Defense Technical Information Center, Technical Report, Accession No. 1020892, June 2016.

[17] Delande, E. D., Jones, B. A., and Jah, M. K., "Exploring and Alternative Approach to the Assessment of Collision Risk," *J. of Guidance, Control and Dynamics*, Vol.46, No.3, pp.467-482, 2023.
[18] Horwood, J. T., Aragon, N. D., and Poore, A. B., "Gaussian Sum Filters for Space Surveillance: Theory and Simulations," *J. of Guidance, Control and Dynamics*, Vol.34, No.6, pp.1839-1851, 2011.
[19] DeMars, K. J., Cheng, Y., and Jah, M. K., "Collision Probability with Gaussian Mixture Orbit Uncertainty," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 979–985.
[20] Cerven, W.T. "Improved Empirical Covariance Estimation." AAS/AIAA Paper No. 13-762, AAS/AIAA Astrody-namics Specialists Conference, Hilton Head SC, August 2013.
[21] Laporte, F. "JAC Software, Solving Conjunction Assessment Issues." 2014 AMOS Technical Conference, Maui HI, Sep. 2014.
[22] Hejduk, M. D., and Johnson, L. C., "Approaches to Evaluating Probability of Collision Uncertainty," *AAS Space Flight Mechanics Meeting*, Napa, CA, Paper AAS 16-241, 2016.
[23] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, 2nd ed., Microcosm Press, El Segundo CA, 2001.
[24] Broucke, D. T., and Cefola, P. J., "On the Equinoctial Orbit Elements," *Celestial Mechanics*, Vol. 5, pp. 303-310, 1972.
[25] Vallado, D., and Alfano, S., "Updated Analytical Partials for Covariance Transformations and Optimization," *AAS/AIAA Space Flight Mechanics Meeting*, Williamsburg, VA, Paper 15-537, 2015.
[26] Sabol, C., Sukut, T., Hill, K., Alfriend, K., Write, B., Li, Y., and Schumacher, P., "Linearized Orbit Covariance Generation and Propagation Analysis via Simple Monte Carlo Simulations," *AAS/AIAA Space Flight Mechanics Meeting*, San Diego, CA, Paper 10-134, Feb. 2010.
[27] Sabol, C., Binz, C., Segerman, A., Roe, K., and Schumacher, P. W., Jr., "Probability of Collisions with Special Perturbations using the Monte Carlo Method," *AIAA/AAS Astrodynamics Specialist Conference*, Girdwood, AK, Paper 11-435, Aug. 2011.
[28] Shelton, C. T., and Junkins, J. L., "Probability of Collision between Space Objects Including Model Uncertainty," *Acta Astronautica*, Vol. 155, 2019, pp. 462-471.
[29] Coppola, V. T., "Including Velocity Uncertainty in the Probability of Collision between Space Objects," *AAS/AIAA Spaceflight Mechanics Meeting*, Charleston SC, Paper 12-247, Feb. 2012.
[30] Chan, K., "Formulation of Collision Probability with Time-Dependent Probability Distribution Functions," *AAS/AIAA Space Flight Mechanics Meeting*, Williamsburg, VA, Paper 15-233, Jan. 2015.
[31] Chan, K., "Hovering Collision Probability," *AAS/AIAA Space Flight Mechanics Meeting*, Williamsburg, VA, Paper 15-234, Jan. 2015.
[32] Lebedev, V., and Laikov, D., "A Quadrature Formula for the Sphere of the 131st Algebraic Order of Accuracy," *Doklady Mathematics*, Vol. 59, No. 3, 1999, pp. 477-481.
[33] Coppola, V. T., "Evaluating the Short Encounter Assumption of the Probability of Collision Formula," *AAS/AIAA Spaceflight Mechanics Meeting*, Charleston SC, Paper 12-248, Feb. 2012.
[34] Press, W. H., Teukolsky, S. A., Vettering, W. T., and Flannery, B. B., *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed., Cambridge University Press, New York, NY, 1992.
[35] Casali, S., Hall, D., Snow, D., Hejduk, M., Johnson, L., Skrehart, B., and Baars, L., "Effect of Cross-Correlation of Orbital Error on Probability of Collision Determination," *AAS Astrodynamics Specialist Conference*, Snowbird, UT, Paper 18-272, Aug. 2018.