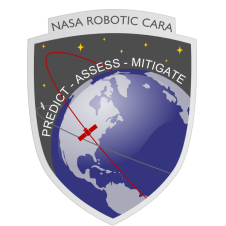


Software Development Kit: Multi Step Probability of Collision (Pc) Calculations

CONJUNCTION ASSESSMENT AND RISK ANALYSIS (CARA) PROGRAM



*Luis Baars*

*Omitron, Inc*

*Colorado Springs, CO*

|  |
| --- |
|  |

National Aeronautics and

Space Administration

Goddard Space Flight Center,

Greenbelt, MD 20771

**September 2025**

**Preface**

This document outlines the Multi Step Probability of Collision (PcMultiStep) Calculation submitted as part of the Software Development Kit (SDK). The SDK is intended to provide both industry and government customers with a code base with which to perform standard calculations inherent to the Collision Avoidance (CA) problem and as outlined in the CA Standard.

Contents

[1.0 Introduction 4](#_Toc208937288)

[1.1 Required Software 4](#_Toc208937289)

[2.0 Risk Assessment Algorithm 5](#_Toc208937290)

[2.1 PcMultiStep Calculation 5](#_Toc208937291)

[2.1.1 PcMultiStep Source Code Description 6](#_Toc208937292)

[2.1.2 PcMultiStep Test Cases 7](#_Toc208937293)

[2.1.3 PcMultiStep Plots 9](#_Toc208937294)

[3.0 Acronyms 12](#_Toc208937295)

[4.0 References 13](#_Toc208937296)

**List of Tables**

[Table 1: PcMultiStep Functions 6](#_Toc208937150)

[Table 2: PcMultiStep Function Input Parameters 7](#_Toc208937151)

[Table 3: PcMultiStep Function Output Parameters 7](#_Toc208937152)

[Table 4: ComparePcMultiStep\_to\_CARAValues Function Input Parameters 8](#_Toc208937153)

[Table 5: ComparePcMultiStep\_to\_CARAValues Function Output Parameters 9](#_Toc208937154)

# Introduction

The CARA Software Development Kit (SDK) contains entries and artifacts for each major algorithm needed to perform the required Collision Avoidance (CA) calculations outlined in the CA Standard. For a typical algorithm, the SDK will include a version of the algorithm, a driver program to take information from a text format CDM and execute the algorithm, producing the needed calculation or output, and a series of test cases that exercise the algorithm and produce validated results.

This document describes a specific algorithm, its associated inputs and outputs, the methodology used within the algorithm and examples of usage.

## Required Software

The following list is of software and hardware requirements for use of this SDK:

* Matlab 2019b
* Windows or Linux operating system for SDMC capabilities

# Risk Assessment Algorithm

## PcMultiStep Calculation

The preferred method to estimate the probability of collision (Pc) of a temporally isolated conjunction is the “PcMultiStep” algorithm formulated by Hall et. al. (2023)[[1]](#endnote-1) which uses a multi-step algorithm to apply increasingly accurate Pc estimation methods, but only as required for efficiency. The Pc estimation methods implemented within PcMultiStep are:

1. The 2D-Pc method: the two-dimensional collision probability method estimates Pc values for rectilinear motion conjunctions by calculating an integral over a circular region on the conjunction plane. Implemented via the PcCircle.m function.
2. The 2D-Nc method: the two-dimensional expected number of collisions method estimates Pc values for high velocity curvilinear motion conjunctions by calculating one integral over the surface of the collision sphere. Implemented via the Pc2D\_Hall.m function.
3. The 3D-Nc method: the three-dimensional expected number of collisions method estimates Pc values for high and low velocity curvilinear motion conjunctions by calculating a series of collision sphere integrals. Implemented via the Pc3D\_Hall.m function.
4. The SDMC method: the Simplified Dynamics Monte Carlo (SDMC) method estimates Pc values for curvilinear motion conjunctions by calculating a sampled series of two-body motion trajectory propagations. Implemented via the Pc\_SDMC.m function.

The first three methods represent semi-analytical approximations, and the last one a Monte Carlo approach. Because the computation required by each successive method increases significantly, the multistep algorithm by default invokes each only as required for efficiency. The determination of whether or not a more accurate Pc method is needed is enabled by usage violation checks that are made after each Pc calculation. For the 2D-Pc method, the usage violation checks are:

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.
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.

The 2D-Nc and 3D-Nc usage violation checks replace the “NPD” violation with “Convergence” violations indicating that iterative processes within each algorithm failed to converge. These methods also have “Offset”, “Extended”, and “Inaccuracy” violation checks, though the definitions of what constitutes each violation differ depending on the method.

The SDMC usage violation checks include an “Offset” check (which is equivalent to the 3D-Nc “Offset” check) and an “Extended” check.

See Hall et. al. (2023)i for further discussion on each Pc calculation and associated usage violation checks.

### PcMultiStep Source Code Description

Three separate PcMultiStep functions have been provided within the SDK. Table 1 summarizes the functions and their usages.

Table 1: PcMultiStep Functions

| Function Name | Description |
| --- | --- |
| PcMultiStep.m | Includes only the 2D-Pc, 2D-Nc, and 3D-Nc methods as well as associated usage violation checks. This method is intended for an operational setting where Monte Carlo runs may not be appropriate when timely algorithm completion is necessary. |
| PcMultiStepWithSDMC.m | Extends PcMultiStep.m by adding the SDMC method (and usage violation checks) to the set of calculations used. This method is appropriate for more complete offline analyses where verification of the other Pc methods may be warranted. |
| PcMultiStepWithPlots.m | Extends PcMultiStepWithSDMC.m by generating Close Approach Distribution and Pc Temporal plots using outputs from all four Pc methods. See section 2.1.3 for more information on the plots. This method is intended for comprehensive offline analyses of individual conjunctions. |

All three functions share similar input and output parameter lists. In fact, these parameter lists are shared across all Pc calculation subfunctions (e.g., PcCircle.m, Pc2D\_Hall.m, etc.) so that the PcMultiStep functions can be easily integrated where previously released functions may have been used[[2]](#footnote-1).

As inputs, the PcMultiStep functions accept the following:

Table 2: PcMultiStep Function Input Parameters

| Input Variable | Definition |
| --- | --- |
| r1 | [3X1] TCA ECI Position Vector of the Primary Object (meters) |
| v1 | [3X1] TCA ECI Velocity Vector of the Primary Object (meters/second) |
| cov1 | [6X6] TCA Primary State covariance matrix corresponding to input primary object reference frame |
| r2 | [3X1] TCA ECI Position Vector of the Secondary Object (meters) |
| v2 | [3X1] TCA ECI Velocity Vector of the Secondary Object (meters/second) |
| cov2 | [6X6] TCA Secondary State covariance matrix corresponding to input primary object reference frame |
| HBR | Combined hard body radius or exclusion zone of the two objects (m) |
| params | (Optional) A Matlab structure holding parameters which can customize the calculation. See individual function documentation for a listing of the parameters available. |

The PcMultiStep functions output the following:

Table 3: PcMultiStep Function Output Parameters

| Output Variable | Definition |
| --- | --- |
| Pc | Probability of Collision calculated using the PcMultiStep method approximation as applied to a single conjunction. |
| out | A Matlab structure holding a large number of auxiliary output variables. See the headers of the PcMultiStep functions for a detailed description of these auxiliary outputs. For users that only need conjunction Pc estimates, this output can be neglected. |

### PcMultiStep Test Cases

Validation cases for the algorithms are contained within the PcTestCaseCDMs directory of the SDK. This directory contains a set of 53 actual CARA conjunctions that have been approved for public release. This test set includes test cases curated by CARA to exercise various typical conjunctions encountered within CARA operations, including many types of usage violations that can occur. The CDMs and test software can be found at:

SDK/DataFiles/PcTestCaseCDMs

The spreadsheet named “CARA\_PcMethod\_Test\_Conjunctions.xlsx” contains a listing of each test case, a description of the expected PcMultiStep results (in the “Comment” column), metadata about the objects involved, and various quantities calculated by CARA using the official software baseline. A “README.md” document (in Markdown format) details the nuances of comparing test outputs across Matlab versions and hardware architectures. A detailed read of the README is suggested before running the comparisons on one’s own system.

The function “ComparePcMultiStep\_to\_CARAValues.m” provides a means of running every CDM provided within the directory, and comparing the local outputs against CARA computed values.

The inputs to ComparePcMultiStep\_to\_CARAValues.m are:

Table 4: ComparePcMultiStep\_to\_CARAValues Function Input Parameters

| Input Variable | Definition |
| --- | --- |
| PcMethod | (Optional) The PcMultiStep method to compare, valid values are: ‘Pc2D’, ‘Nc2D’, ‘Nc3D’, or ‘SDMCPc’.  Defaults to ‘Pc2D’. |
| highAccuracySDMC | (Optional) If the PcMethod is ‘SDMCPc’ and this value is set to true, then the SDMC method will be run in a high-accuracy mode that is designed to exactly replicate CARA SDMC Pc values. If the PcMethod is not ‘SDMCPc’, then this option has no effect.  Defaults to false.  Warning: This mode will take a very long time to complete. |
| numToRun | (Optional) If the PcMethod is ‘SDMCPc’ this value will specify the number of test cases to run. This will save some time if a user only wants to run a subset of conjunctions to make sure SDMC is running correctly without going through the whole set. When empty, all conjunctions will be run. If the PcMethod is not ‘SDMCPc’, then this option has no effect.  Defaults to [ ]. |

The ComparePcMultiStep\_to\_CARAValues function outputs the following:

Table 5: ComparePcMultiStep\_to\_CARAValues Function Output Parameters

| Output Variable | Definition |
| --- | --- |
| compareData | A table containing comparison information for each CDM analyzed. See the function description for more information on the comparison outputs supplied, especially for SDMC. If the function is called without this variable, then a text output of the comparison is printed to the Matlab console window. |

### PcMultiStep Plots

The PcMultiStepWithPlots.m function will generate two types of plots. The following sections provide an explanation of the plots. The sample conjunction ‘000035946\_conj\_000030648\_20221210\_140311\_20221206\_003234.cdm’ is used in both examples.

#### CA Distribution Plots

The close approach distribution plots show an encounter plane representation that includes the hard-body circle, an ellipse representing the marginalized PDF for the 2D-Pc rectilinear trajectory approximation, and (when available) the distribution of close approach points from an MC simulation that incorporates curvilinear effects. Figure 1 shows a classic example of a CA distribution plot when a 2D-Pc usage violation occurs.

The three panels show different views of the encounter plane. Specifically, the top-left panel shows a zoomed-out view of the entire CA distribution; the top-right panel shows a zoomed-in view that plots the HBR circle along with the distribution of MC hits; and the bottom panel shows a view zoomed in by unequal magnifications on the horizontal and vertical axes. The pink oval plotted on each panel shows the 3-sigma covariance ellipse predicted using the 2D-Pc method. The dots show the CA points calculated in an SDMC method simulation – with red dots representing MC hits, and blue dots representing misses. The text between the panels provides some details about the conjunction, including the combined HBR (20 m, in this case), the nominal miss distance (7.24 km), the relative velocity (53.6 m/s), and the collision probabilities estimated with 2D-Pc, 2D-Nc, 3D-Nc, and SDMC methods. In addition, this text provides a count of the number of usage violations encountered for each Pc method.

In this case, the bottom panel shows that the distribution of MC points does not align well with the 2D-Pc method covariance ellipse. This misalignment graphically illustrates that the 2D-Pc method has at least one usage violation, and therefore may not provide an accurate probability estimate for this conjunction. The 2D-Pc method yields an erroneous estimate of Pc ≈ 4.5e-23, but the 2D-Nc, 3D-Nc, and SDMC methods all yield consistent and accurate estimates of Pc ≈ 1.5e-4 (note the 95% confidence region for SDMC). This occurs because the latter three methods all accurately account for curvilinear trajectory effects.

Chart

AI-generated content may be incorrect.

Figure : CA Distribution Plot

#### Pc Time Plots

Hall et. al. (2018)[[3]](#endnote-2) introduced Pc time plots, which are a temporal representation of the evolution of Pc near TCA. Figure 2 shows the Pc time plot for the same conjunction depicted in Figure 1.

The top panel plots the cumulative Pc against the time (in seconds) from TCA. The 2D-Pc (solid blue line) and 2D-Nc (cyan dashed line) methods will always show straight lines, indicating that the Pc values calculated are invariant over time. On the other hand, since the 3D-Nc (black dotted line) does integrate over time, we can estimate how the Pc changes. The SDMC results (pink solid line and pink shaded region) indicate the cumulative Pc and 95% confidence, respectively, based on the number of hits that have been found. Note that 2D-Nc and 3D-Nc both fall within the 95% confidence limits once the end of the simulation is reached.

The bottom panel plots the Pc rate against the same time axis from the top panel. Here only the 3D-Nc and SDMC values are plotted. In addition, green vertical bars indicate the conjunction duration bounds for 2D-Nc (light green) and 3D-Nc (dark green) methods.

The text on the top right shows similar metadata and Pc results as were displayed within the CA Distribution plot. Below this text, the usage violation checks section has been expanded to show which specific usage violations were encountered for each Pc method. To the right of the usage violations, specific times for the conjunction duration bounds, peak Pc rate, and time range for SDMC hits are provided.

Chart

AI-generated content may be incorrect.

Figure : Pc Time Plot

### 

# Acronyms

|  |  |
| --- | --- |
| 2D | Two Dimensional |
| 3D | Three Dimensional |
| CA | Collision Avoidance or Close Approach, depending on context |
| CARA | Conjunction Assessment Risk Analysis |

|  |  |
| --- | --- |
| CDM | Conjunction Data Message |

|  |  |
| --- | --- |
| ECI | Earth Centered Inertial |
| HBR | Hard Body Radius |
| MC | Monte Carlo |
| Nc | Expected Number of Collisions |
| NPD | Non-Positive Definite |
| Pc | Probability of Collision |
| SDK | Software Development Kit |
| SDMC | Simple Dynamics Monte Carlo |
| TCA | Time of Closest Approach |

# References

1. Hall, D. T., Baars, L. G., Casali, S. J. *A Multistep Probability of Collision  
    Computational Algorithm.* AAS 23-398. 2023. [↑](#endnote-ref-1)
2. While the basic inputs and outputs remain the same, additional input and output parameters (“params” and “out” structures, respectively) are different. Integration of PcMultiStep functions will likely require some work to resolve differences within these structures. [↑](#footnote-ref-1)
3. Hall, D. T., Casali, S. J., Johnson, L. C., Skrehart, B. B., Baars, L. G. *High Fidelity   
    Collision Probabilities Estimated Using Brute Force Monte Carlo Simulations.*   
    AAS 18-244. 2018. [↑](#endnote-ref-2)