Polygon-Based Algorithms for N -Satellite Constellations Coverage Computing

Santiago M Henn, Juan A Fraire, Holger Hermanns

Polygon-Based Algorithms for *N*-Satellite Constellations Coverage Computing

Santiago M. Henn*, Juan A. Fraire*†, Holger Hermanns‡
*CONICET - Universidad Nacional de Córdoba, Argentina
†Univ Lyon, INSA Lyon, Inria, CITI, F-69621 Villeurbanne, France
‡Saarland University, Saarland Informatics Campus, Saarbrücken, Germany

***Abstract*—Satellite Coverage Analysis is a fundamental performance assessment element in remote sensing and communications services projects. Coverage is a key parameter in the constellation operation and design for missions relying on several satellites. Since coverage areas over the surface of the Earth change with time, intersecting and drifting apart, the dynamics of every satellite influence the constellation's behavior as a whole. For this reason, every configuration change, be it in the number of satellites or their relative positions, heavily impacts the cost/performance of the mission. This paper presents a constellation-to-ground coverage analysis model that enables the rapid evaluation of areas on the surface of the Earth. The method leverages geodetic projections and an oblate Earth model and uses dynamic transformation and anti-transformation techniques combined with polygon Boolean operations. Timestamped datasets are obtained to account for the dynamics of the scenario, which can be exploited in statistical coverage analysis of the constellation. Our empirical evaluations show that this approach is superior in accuracy and computation effort compared to traditional net-point techniques. While net-point approaches are at the core of state-of-the-art commercial software, they are approximate. We show that, for finer grid granularity, the net-point schemes converge to our polygon-based results.**

***Index Terms*—Coverage computation, Satellite Constellations, LEO satellite**

I. Introduction

Earth coverage is a fundamental performance measure in satellite constellations design and management. The number of spacecraft a mission needs largely depends on the constellation's ability to access well-defined regions on the Earth and the mission's requirements in terms of utility and/or data output [1]. Since Earth's topography is anything but homogeneous and political divisions are irregular, often following the landscape's morphology features, an Earth observation Mission Region(s) of Interest (ROIs) can take any shape or size [2], [3]. Time-dependent coverage to these areas from one or more satellite instruments requires a detailed analysis if precise analysis and predictions on the constellation's behavior are to be performed.

Coverage analysis is usually carried out using computer simulations, which get more demanding with increasing constellation size and complexity. However, decreasing launch costs and other economic factors (like the availability of COTS components) have induced a trend [4]–[7] towards distributed, small-satellite based missions [8], [9].

This brings back into discussion the so-called "constellation-to-ground coverage problem", which traditionally has been addressed using probabilistic statistical models [10], [11], judgment theorems [1], performance methods based on transformation groups [12], [13], numerical solving of differential equations [14], and interpolation algorithms [15]. All these approaches aim to find metrics for satellite access towards aiding mission analysis and design, but have some common shortcomings. First, each method restricts certain aspects of the general constellation-to-ground coverage problem, such as continuous global or region-restricted coverage. Second, in mathematical modeling, approximations are required; therefore, the obtained results are inexact and qualitative [16].

Analytical approximation methods have also been proposed. Ulybyshev's first proposed method [17] exploits a two-dimensional coverage-mapping plane for which the right ascension of the ascending node and time are considered dimensions of space to find revisit times, constrained to circular, far from the pole regions. The authors then propose a more general method [18], focused on obtaining maximum revisit times with several constraints, such as a spherical Earth model. However, secular variations of orbital elements are considered. These last two methods serve the purpose of determining access frequency. Still, no information on the coverage area to at least N-satellites, visibility constraints, or region of interest access is considered all at once. Zhang et al. propose a geometrical multi-satellite discontinuous-coverage analysis method [19] centered around obtaining performance metrics regarding revisit time. While its also constrained to a spherical Earth problem, it improves semi-analytical methods proposed by Crisp [20], which focuses on calculating the maximum revisit time of symmetric constellation for different altitudes, target latitudes, and constellation configurations, and Razoumny's approach [21], [22], who analyzed the discontinuous coverage of route satellite constellation using deterministic and stochastic approaches. However, the methods mentioned above have one or more restrictions commonly associated with the constellation-to-ground problem. These limitations include dependency on a spherical Earth, reliance on the symmetry of analyzed constellations, dependency on satellite propagators for the analytical approach, and the inability to access regions

This work has received support from the Project STARS STICAMSUD 21-STIC-12 Code STIC2020003, the ANR-21-CE25-0002-01 project, and the MISSION project from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101008233.

of interest for analysis. To overcome these limitations, we propose an alternative method in this work. It is important to note that while our approach does not consider circular FOVs based on intersections over a spherical Earth, we do consider near-circular FOVs based on other assumptions.

This paper proposes a new paradigm for calculating satellite coverage metrics for Low Earth Orbit (LEO) Satellite constellations of any ROI over the surface of the Earth that can be approximated by a piece-wise polygon comprised of segments between geographic coordinates. The ROI(s) and its intersections with the satellite's access areas are defined by N-sided precise, non-self-intersecting irregular polygons. These polygons capture surface areas more accurately than collections of grid points. Therefore, they are a better base for approximate calculations than methods based on net points. The core idea of the proposed methodology is to exploit Boolean operations algorithms for polygons together with fast orbital propagators, combined with a series of transformations between Euclidean and Non-Euclidean spaces. This allows us to avoid complex polygon operations in three-dimensional space as the ones required to deal with surfaces over the Earth, which have analytical solutions for special cases or are constricted to sphere operations [23], [24]. To account for the Earth's oblateness, we harvest and improve upon the method proposed by Nugnes et al. [25], [26] to obtain conical fields of view. In particular, we exploit their method to obtain a conical set of points from a given visibility elevation threshold, replacing the originally suggested iterative procedure to obtain the half-aperture angle through a binary search algorithm. Then, the essential difference with existing work relies precisely on exploiting fast Boolean operations and satellite-to-Earth vector-intersecting algorithms to translate the problem of access between points on a spheroid with constraints in elevation threshold to a 2D realm, where finding access to *N* satellites changes from a 3D space net-point grid to N-satellite(s) propagated positions in time problem, to a polygon intersection problem, where only access region boundaries are needed, which we demonstrate to be a fast alternative that yields accurate results. In terms of application, our research's main contributions can be divided into two main areas: mission phase and domain. For the mission phase, we contribute the following points of interest.

- Constellation design: defining the parameters of a LEO constellation to optimize mission performance when evaluating specific metrics.
- Constellation operations: scheduling specific tasks at the right moment to maximize a mission's data output, e.g., an adequate percentage of coverage is reached.

and for the mission domain:

- Communications: obtaining analytic throughput estimations and ensuring that *N* receiving/transmitting gateways cover specific areas.
- Earth observation: determining areas and timestamps to perform interferometric and stereographic acquisitions with N observation points and maximize data frequency (i.e., to reduce overlapping) for a specific region of interest served by N sensors.
- Navigation: supporting trilateration that requires at least N satellites in the served region.

The paper is organized as follows. Based on net point divisions, an overview of the traditional approach to the constellation to Earth coverage problem is presented and discussed in Section II. Section III-B provides helpful definitions used in the model, to then present the algorithm steps. Section IV presents three use cases purposefully designed to overlap and intersect a set of ROIs on the surface of the Earth. Conclusions and future perspectives are summarized in Section V.

II. Background

According to Wertz, one of the most important coverage characteristics is that Earth coverage is not a Gaussian parameter, and statistical data can give misleading results [2, page 459]. This leads to the recurring need for quick and precise analyses of many operating scenarios during mission design, be it for early considerations on possible configurations or for planning complex maneuvers during late phases of the mission (e.g., satellite re-positioning) [27], [28]. For Low Earth Orbits (LEOs), which are orbits that fall between 200-300 km to 1600 km in height, orbital speeds reach velocities up to $\sim$7.5 km/s, which means that the area that a satellite sees (access area) drifts rapidly [29]. For higher orbits such as Medium Earth Orbits (MEOs), between 3000 km and 35790 km, or Geosynchronous Earth Orbits (GEOs), precisely at 35790 km, orbital speeds are lower, and access areas are larger [2, table 2.7]. In the introduction, several algorithms are presented, and each has advantages and restrictions when addressing the constellation-to-ground coverage problem. Table I summarizes each of the relevant characteristics of this work, denoted by its corresponding reference, alongside Agi's STK (System Tool Kit) and our proposal. Since net-point division analysis algorithms are commonly used in coverage analysis [30], and they fulfill several of the relevant characteristics, in the following section, we will present how this method works with some of its advantages and downsides.

A. Net-point method

This method places a set of coordinates within a Region of Interest (ROI) in a uniform pattern, and relative positions between each point and each satellite are computed. Coverage to a delimited area surrounding a point is assumed to occur if accessible [16]. For the sake of simplicity, we will assume that access areas of instantaneous Field of View (FOV) are the areas over the surface of the Earth where a device has a line of sight of at least one satellite above a certain elevation over the horizon. Therefore, the resulting FOVs are approximately circular areas on the Earth's surface as depicted in Fig. 1. In practice, these footprints depend on the satellite's sensors, instruments, or antenna's radiation patterns and the Earth's shape [7]. Fig. 2 shows three access areas overlapping at a given instant. The ROI is also assumed to be sufficiently small not to evidence the Earth's curvature in this depiction. The ROI meshes into a grid of points or coordinates, and access areas are named **R**, **G**, and **B**. We can observe satellites' FOVs in Red, Green, and Blue. In the typical analysis, each coordinates

TABLE I: Comparison between Earth coverage analysis algorithms.

	[10]	[11]	[1]	[12], [13]	[14]	[18]	[16]	[21], [22]	[19]	[20]	STK	Proposed alg.
Allows different FOVs	✗	✗	✗	✗	✗	✗	✓[1]	✗	✓	✓	✓	✓
Allows any LEO satellite orbit	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗	✓	✓
Allows for analysis near the poles	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Allows for horizon visibility elevation threshold analysis	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
Allows for ROI percentage of access analysis in the time domain	✗	✗	✗	✗	✗	✓[1]	✓	✓	✓	✓[2]	✓	✓
Allows the usage of several orbital propagators	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
Independent of net-point grid granularity	✓[3]	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗	✓
Considers Earth's oblateness [4]	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✓	✓

[1] It is not implemented the published work.
[2] Restricted to target latitudes and longitudes.
[3] It does not rely on net-point area subdivisions.
[4] Excluding those that consider the Earth's shape as a source of gravitational perturbations.

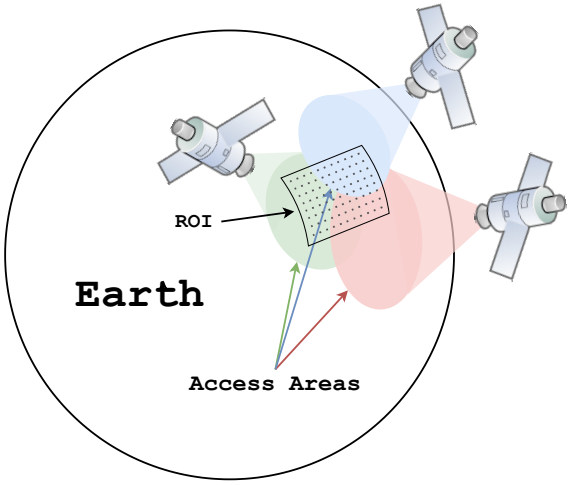


Fig. 1: Depiction of circular FOVs over the surface of the Earth, intersecting a Region Of Interest.

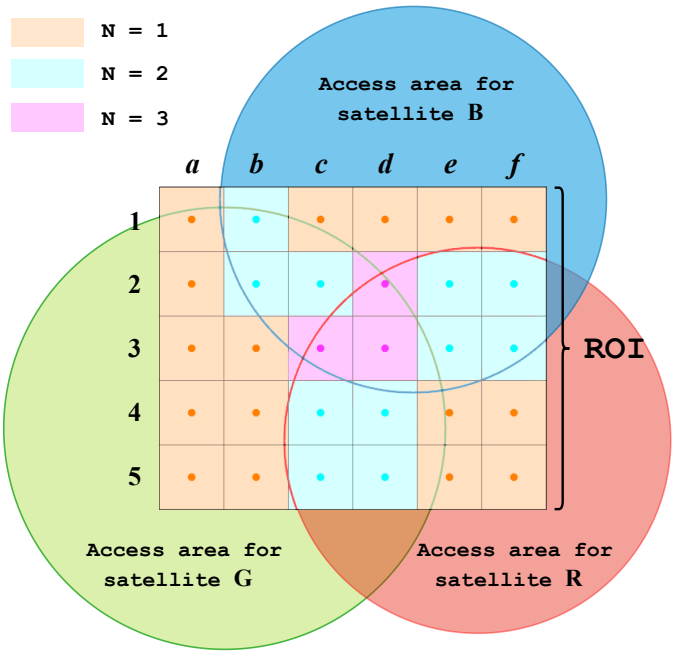


Fig. 2: Detail of a meshed ROI, access areas for three intersecting FOVs, and grid access to N Satellites. Points $\{a, b : 1, 5\}$ and $\{c, d : 2, 5\}$ have access to Satellite **G**, points $\{b, f : 1, 1\}$ and $\{c, f : 1, 3\}$ have access to Satellite **B** and finally, points $\{c, f : 3, 5\}$ and $\{d, f : 2, 5\}$ have access to Satellite **R**. With only this information, it is possible to derive that some points will have access to one satellite or N multiple satellites, as it is the case for points $\{b, 1\}$,$\{b, 2\}$,$\{c, 2\}$ having access to 2 Satellites (**G** and **B**) and points $\{c, 3\}$,$\{d, 2\}$,$\{d, 3\}$ to all 3 Satellites.

$\{a, f : 1, 5\}$ will be computed alongside the satellite's position and an Earth's geometry model to determine if access is possible; albeit in this case, we assume positive access as having a line of sight from a satellite to a point and vice versa. A set of data comprising timestamped N satellite(s) access points can be constructed using the net point information. For example, if we assume that each square in Fig. 2 represents 1 unit of area, at the depicted instant in time, 16 area units have access to $N = 1$ satellite, 11 units reached by 2 satellites, and finally, 3 units with 3 satellites in view. Some characteristics of this method are discussed in the following paragraphs.

a) Grid precision: The precision depends entirely on the ROI's grid granularity or how dense and uniformly distributed the point population is. This measure is given in degrees and establishes the separation amongst the net points. For each point, computations must be performed to assess whether a satellite's access is possible. As the grid becomes more precise, the computation time needed to perform the analysis increases, which works against the ability to iterate over a scenario quickly. Furthermore, if the ROI size increases, but grid granularity is kept constant, so does the effect that a close-to-edge point has: if a coordinate falls just near the FOV of a satellite, it will induce an error proportional to the size of the area surrounding said coordinate.

b) ROI size: In the classical net-point division method, the coverage state of the net point is regarded as that of the corresponding grid cell. That is, if a satellite constellation can cover the net point, it is considered that the constellation can cover the entire corresponding grid cell; otherwise, it is considered that the constellation has no coverage for the corresponding grid cell. This is called the 0-1 judgment strategy [31]. If the ROI size increases with its grid granularity remaining constant, the number of net points will increase proportionally to this increment. According to the algorithm's principle, if the results' precision increases N-fold, the computing time will increase N^2-fold. Thus, this method is usually unsuitable for large regions under high accuracy requirements. Overall, the net point division method exhibits a low computational efficiency and little reliability of the results [30], [32].

c) Orbit propagation: Orbit propagation accounts for much of the computation time needed for coverage analysis. This is: predicting every satellite's position over time while considering all necessary orbital perturbations. The net point approach requires a precise notion of the rise and set times of the satellites, a measure that converges towards an instant in time using sub-sampling algorithms. Re-propagating or approximating a given satellite's position is required, and thus, some knowledge of the orbit dynamics. These procedures increase processing time, and not using them significantly decreases the net point method's accuracy, as shown in section IV.

B. Polygon-based methods

We identified three publications that propose polygon-based approaches for addressing the constellation-to-ground coverage problem. Shaojun et al. present an area-based method to compute gaps in satellite availability on the surface of the Earth, using a statistical analysis based on the ratio of the total area of the *holes* (regions without connectivity) to the Region of Interest [32]. They assess coverage gaps over time using interpolation techniques, obtaining the first polygon intersection points over a spherical model of the Earth, which are subsequently adjusted to fit an ellipsoidal model. Finally, the area is obtained using numerical integration based on Green's Theorem for a polygonal area on a sphere's surface. Dai et al. use a geometric subdivision approach, decomposing larger regions into sub-regions and interpreting the multi-satellite coverage problem as a one-satellite coverage problem [31]. They obtain a set of adjacent convex triangular polygons for a given scenario time-step, consisting of several arc segments representing the path amongst the polygon vertices. Using their analytical approach, they can determine the Earth's surface area based on the assumption of a spherical shape.. Song et al. propose a method called *Cell Area Analytical Method* [16]. This method is based on the net point approach but defines a *cell* around each grid point that is progressively partitioned until it converges to a desired precision. The algorithm outputs a spherical polygon that defines the Region of Interest reached by N-satellites, and the coverage rate is computed afterward.

III. Model

Performing analysis involving the Earth's surface is a well-known issue in Geography and Satellite Applications [33], [34]. Involving astrodynamics such as Constellation-to-ground coverage analysis and non-spherical models of the Earth further increases the complexity of the problem [35], [36]. Since dealing with any non-euclidean plane is challenging, the issue is usually approached by making compromises, simplifications, and assumptions. We identify three main items in the case of the study of the Earth's surface: i) assuming a spherical Earth model, ii) restricting analysis on small regions, and iii) restricting analysis far from the poles. Some provide the bases for comprehensible and easy-to-implement methods, which are precise enough for specific applications [36], [37], but fall short when pushed outside their limits.

A. Definitions

a) Euclidean and Non-Euclidean spaces: In the context of the presented work, we will refer to the Euclidean plane as a two-dimensional plane that answers to the modern definition of Euclidean space. A Non-Euclidean mathematical space is every space that does not answer to the definition of Euclidean space in at least one of its postulates. The main difference that characterizes a core concept in each one, and that we care the most for our method, is how the shortest path between two points is defined in one or the other, given that we will deal with closed polygons. Those can be thought of as a series of consecutive segments. We will focus on spheres and ellipsoids among the several examples that exist for non-Euclidean planes. This is because we need to deal with points over the surface of the Earth, which is approximated by a sphere or an ellipsoid for all intents and purposes of this explanation. The shortest path between points in any Riemannian manifold is called a Geodesic, where a Riemannian manifold is a smooth topological space that allows the construction of metrics such as distance, angles, and area. In a Cartesian Plane, a Geodesic is a simple straight line; in spheres, they are curves referred to as Great Circles. An essential part of this work deals with mapping regions delimited by points in different types of spaces and/or determining whether the enclosed regions by these points intersect or not. Transformations between different geometric spaces is a process where compromises need to be made since it is impossible to preserve all the properties of a shape after applying a transformation to the points or segments that define it. The most common example of a spheroid-to-plane transformation is the widely used Mercator Projection, used to map the surface of the Earth, whose shortcomings are wildly known and criticized, mainly the extreme deformation that shapes exhibit near the Earth's poles [38].

b) Geographic Coordinate System (GCS): The simplest, oldest, and most widely used coordinate system for positions on the surface of the Earth is the Geographic Coordinate System (GCS), which measures and communicates positions using two pairs of angles: Latitude and Longitude. The usual notation for this coordinate is the tuple (Lat, Lon). Although this is similar to a Cartesian coordinate system, the tuple does not represent a coordinate on a planar surface. Instead,

Latitude and Longitude are angle measurements from the equator and the Greenwich meridian, respectively. In this work the assumed possible ranges are $(-90°, 90°)$ for the Latitude and $(-180°, 180°)$ for the Longitude.

c) Access Area Polygon (AAP): We will now define an Access Area Polygon (AAP) as a set of coordinates enclosing a region over the surface of the Earth. The term Area is coined as part of the definition, although we are dealing with non-planar surfaces. As stated in the introduction, coverage areas, and therefore AAPs, can take the form of any irregular polygon when viewed from a zenithal position, provided it is not self-intersecting. Fig. 3 shows three examples of irregular polygons, where **a** and **b** are valid inputs for the algorithm presented in this work, and **c** should be treated as two separate polygons if such type of shape is needed. This means

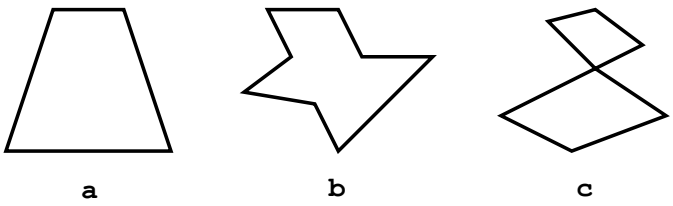


Fig. 3: Example for three irregular polygons, where **a** and **b** are non-self-intersecting shapes and **c** is a self-intersecting quadrilateral

that coverage areas can be any region on Earth that can be approximated as a piece-wise non-self-intersecting polygon, such as latitude-longitude bounded regions, spherical polygons (provided a geodesic is drawn in-between coordinates) or irregular polygons representing a country's boundaries. A classic usage of coverage areas in satellite operations is drawing the instantaneous access areas of Earth-to-satellite visibility at a given time for one or more satellites. These areas are approximately spherical caps, as the ones depicted in Fig. 4, that, when viewed from above, take the appearance of circular AAPs, as shown in Fig. 5. These AAPs comprise collections of

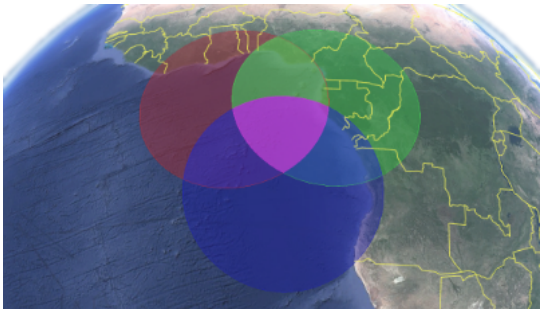


Fig. 4: Example of three valid AAPs intersecting over the surface of the Earth

coordinates enclosing a solid portion of the Earth. A polygon segment is defined between each successive coordinate, and the number of segments that form a polygon is a key parameter that determines the level of precision when approximating a smooth, continuous access area. Intersection points between AAPs are critical to understanding what is needed to obtain coverage areas. In the Euclidean plane, these points always correspond to the intersection between the 2D shapes. The area contained in the intersection can be understood as the enclosed region by the set of coordinates that fall between the intersection points of any two shapes. Fig. 5 depicts what

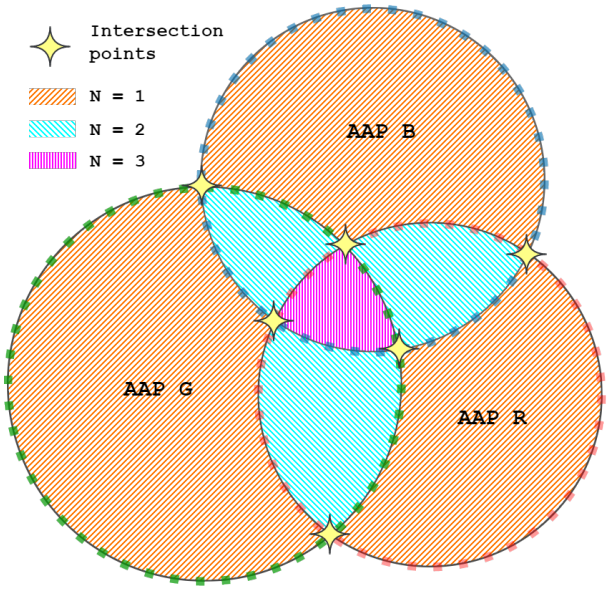


Fig. 5: Top view of AAPs for N = 1,2,3 satellites, depicting intersection points and intersection regions.

we can think of as a top view of the intersection points and intersection areas for the three access areas that served as an example in Fig. 2.

B. Model

a) Obtaining AAPs: The first step is to obtain the AAPs corresponding to each satellite at a given time. To achieve this, the satellite's positions fixed with respect to the Earth must be obtained through an orbital propagator such as a simple two-body computing procedure, SGP4 propagator, or High Precision Orbit Propagators (HPOP) [2]. Once obtained, the Earth-intersecting conical field of view for each satellite is computed using a method proposed by Nugnes et al. [25], [26], [39] to obtain conical fields of view. They propose two procedures: either modeling a nadir-pointing conical sensor of a certain aperture ($\boldsymbol{\eta}_{hor}$ in Fig. 6) mounted on the satellite or setting a certain elevation threshold over the horizon ($\boldsymbol{\varepsilon}_{1,2}$ in Fig. 6) as a constraint for access. This last procedure requires numerical methods, as there is no analytical solution to obtain the coordinates for the conical intersection. The published algorithm proposes a step-by-step successive approximations algorithm to evaluate different values for aperture angles until the desired elevation threshold value, $\boldsymbol{\varepsilon}_{target}$ is found, within a certain margin of tolerance [39].The number of iterations in this approach requires finding a single conic intersection coordinate depending on factors such as the target angle and

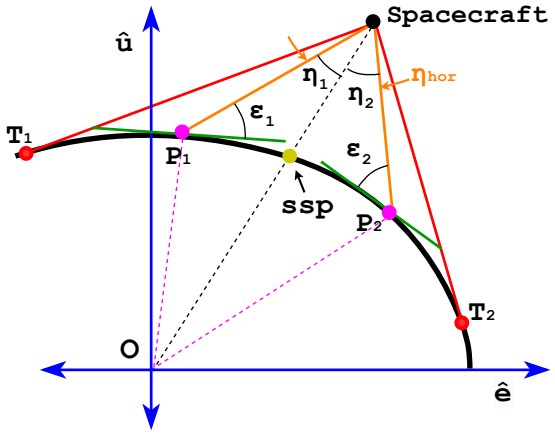


Fig. 6: Diagram of a Spacecraft, Earth-intersecting points, and defined angles, where $\hat{\mathbf{e}}$ is the unit direction vector along the major semi-axis and $\hat{\mathbf{u}}$ is the unit direction vector along the semi-minor axis.

the step size used to reach it. This means it can take thousands of iterations, which can be impractical for some scenarios. We propose a binary search algorithm that has the potential to significantly decrease the number of iterations required to achieve the same goal, thus offering a potentially more efficient alternative to the cited approach. The algorithm is analogous for finding both $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, as seen in Fig. 6, and works as follows:

1) Obtain the first endpoint for each of the search intervals, $\eta_{1_{sup}}$, and $\eta_{2_{sup}}$, as the values for $\eta_{1,2}$ that correspond to the points $\boldsymbol{T}_{1,2}$ respectively. There cannot be half-aperture angles η greater than those values since they will be associated with points over the surface of the Earth.
2) Obtain the second endpoint for each of the search intervals, $\eta_{1_{inf}}$, and $\eta_{2_{inf}}$, as the values of $\eta_{1,2}$ that correspond to one degree less than the computed value for a spherical Earth model [2]:

$$\eta_{1,2\,inf} = \arcsin \frac{R_{\oplus} \cos \varepsilon_{1,2}}{\|\boldsymbol{P}_{1,2}\|} - 1\,deg. \tag{1}$$

where $R_{\oplus}$ is the average Earth radius. A value of 1 degree is used since it is the maximum error between the oblate and spherical Earth models [25], unless this operation results in a negative elevation, where a value of 0 degrees is used instead.
3) Compute the initial guess for $\eta_{1,2}$ as the half interval between endpoints for each one:

$$\eta_{1,2} = \frac{\eta_{1,2\,sup} + \eta_{1,2\,inf}}{2} \tag{2}$$

and obtain the values for $\varepsilon_{1,2}$.
4) Compute the differences between $\varepsilon_{1,2}$ and ε_{target}. If the differences are within the defined tolerance, end the search.
5) Update the endpoints according to the obtained $\varepsilon_{1,2}$, reducing the search interval, and repeat from step 3).

We will present an example in which a point $\boldsymbol{P}_1$ is to be found, at different elevation threshold target values, for a spacecraft located at the fixed coordinates $[x, y, z] = [-990.945, -5817.571, 3334.217]$. A single point is obtained with a rotation angle $\psi = 0$ for the rotation matrix A_{321} described in Eq. 43 of the generic pointing case Coverage Region Determination method [25]. The number of iterations needed to converge at the ε_{target} with a 0.001-degree tolerance for the successive approximations (**A**) [39] and the binary search[1] (**B**) algorithms are presented in Table II.

TABLE II: Obtained positions and number of iterations for algorithms **A** (successive approximations) and **B** (binary search) at different ε_{target} values

ε_{target}	Alg.	Obtained position [km]			ε_1	N° Iter.
		x	y	z		
5°	**A**	781.945	-5557.115	3020.954	5.0091	105
	B	782.671	-5557.065	3020.858	5.0005	10
15°	**A**	145.495	-5572.148	3089.708	14.9995	845
	B	145.477	-5572.147	3089.710	14.9999	12
25°	**A**	-208.277	-5556.276	3114.431	25.0046	2035
	B	-208.140	-5556.285	3114.423	24.9995	11
35°	**A**	-421.511	-5538.366	3124.686	35.0006	3444
	B	-421.507	-5538.366	3124.686	35.0005	13

b) Transformation: The second step is to define a bijective transformation capable of mapping the AAPs defined in a GCS into a continuous Euclidean plane. This means that, due to the injective-surjective behavior of the transformation needed, we shall be able to retrieve coordinates again into the non-euclidean plane. This transformation shall also preserve the intersection points between AAPs, such that any intersection point or region mapped into the Euclidean plane can be retrieved again into GCS coordinates. The core strategy is the following: to map AAPs into a Euclidean plane and there, find the AAPs intersections and anti-transform them back again into GCS coordinates, which can subsequently be stored or fed into a Geographical Information System (GIS), a type of database containing geographic data [40], to obtain metrics such as surface. Let $\{\mathcal{L}^{NE}\}$ be a set of the AAPs GCS coordinates in the non-euclidean (NE) plane and $\mathcal{T}$, a transformation that maps the coordinates into a euclidean (E) plane, we need to verify that:

$$\mathcal{T}\{\{\mathcal{L}^{NE}\}\} = \{\mathcal{L}^{E}\} \tag{3}$$

and that:

$$\mathcal{T}^{-1}\left\{\{\mathcal{L}^{E}\}\right\} = \{\mathcal{L}^{NE}\} \tag{4}$$

with bijective behavior. This is, we need to verify that:

$$\mathcal{T}^{-1}\left\{\mathcal{T}\left\{\{\mathcal{L}^{NE}\}\right\}\right\} = \{\mathcal{L}^{NE}\} \tag{5}$$

Our aim is to find an intersection in the euclidean plane such that, if we have two intersecting AAPs $\{\mathcal{L}_1^{NE}\}$ and $\{\mathcal{L}_2^{NE}\}$, we can verify that:

$$\{\mathcal{L}_1^{NE}\} \cap \{\mathcal{L}_2^{NE}\} = \mathcal{T}^{-1}\{\{\mathcal{L}_1^{E}\} \cap \{\mathcal{L}_2^{E}\}\} \tag{6}$$

Where $\mathcal{T}^{-1}$ is the anti-transformation that remaps coordinates from the Euclidean plane back into the non-euclidean space. The process is depicted for two AAPs in Fig. 7.

[1]Available at https://github.com/santiagohenn/oblate-earth-coverage-java

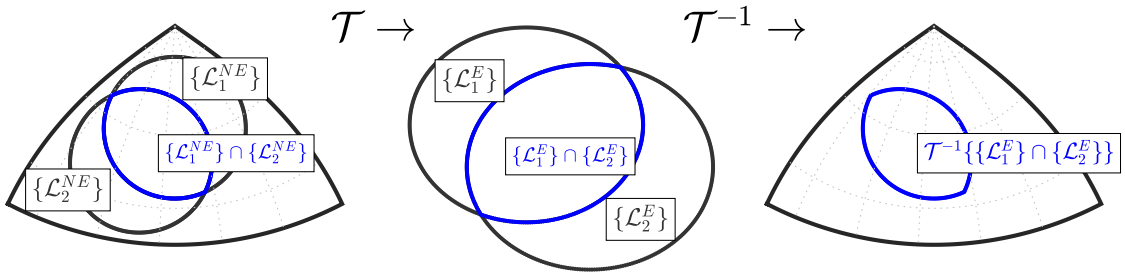


Fig. 7: Two AAPs intersecting in the Non-Euclidean plane (left), transformed to the Euclidean space (center) and the retrieved intersection (blue) in the Non-Euclidean space again (right)

Since intersections are easy to solve in the 2D Euclidean plane, having a transformation $\mathcal{T}$ that verifies the requirements in the scope of LEO Satellites AAPs allows us to use a recursive technique that can be applied to k-combinations from a given set $\mathcal{S}$ of N satellites' AAPs to obtain every intersection in a specific moment in time. Due to our historical need to use flat maps, several transformations are used to map Earth's (and other celestial bodies) surface into a Euclidean plane. These transformations are called Cartographic Transformations, and although projections are considered in several fields of pure mathematics, map projections specifically refer to those in Cartography. All projections of a spheroid on a plane necessarily distort the surface somehow and to some extent. Finding the right one demands a compromise where not all transformed regions' properties are preserved [41]. More generally, a map projection and any method of flattening a continuous curved surface onto a plane can be considered. Since we are especially interested in intersections between AAPs (which are curved continuous surfaces), a conformal map projection will be used as our transformation $\mathcal{T}$, as described in Eq. 3. Specifically, we will use stereographic projection.

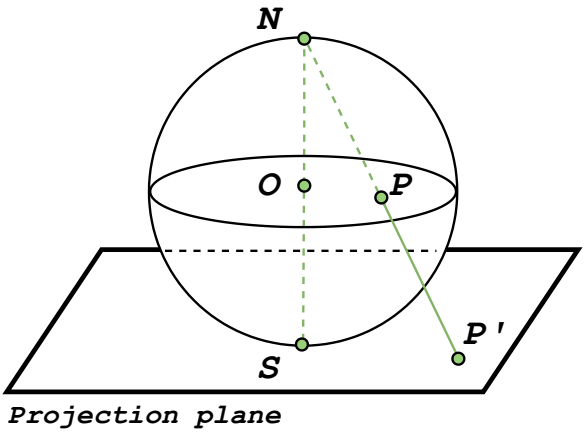


Fig. 8: Diagram of a stereographic projection with its reference point at the North Pole.

The stereographic projection is a smooth, bijective transformation that projects a sphere or spheroid (non-euclidean surface) onto a Euclidean plane. The projection is defined on the entire spheroid, except at the projection point. It is conformal, meaning it preserves angles at which curves meet, but is neither isometric nor equivalent (area-preserving). Still, we can compromise on this projection since shapes and areas are subjects to be addressed in the non-euclidean plane. Stereographic projections are obtained by projecting points $\mathcal{P}$ on the surface of the sphere from a sphere's pole to a point $\mathcal{P}'$ in a plane tangent to the other pole, as it is depicted in Fig. 8, exemplifying a projection from the North pole onto a plane tangent to the South pole. Transformation equations for a pair of geographic coordinates x and y are:

$$x = k \cos\phi \sin(\lambda - \lambda_0) \tag{7}$$

$$y = k\left[\cos\phi_0 \sin\phi - \sin\phi_0 \cos\phi \sin(\lambda - \lambda_0)\right] \tag{8}$$

Where λ_0 is the central longitude, ϕ_0 is the central latitude, and:

$$k = \frac{2R}{1 + \sin\phi_0 \sin\phi + \sin\phi_0 \cos\phi \cos(\lambda - \lambda_0)} \tag{9}$$

Inverse formulas to retrieve the latitude ϕ and longitude λ are:

$$\phi = \sin^{-1}\left(\cos c \sin\phi_0 + \frac{y \sin c \cos\phi_0}{\rho}\right) \tag{10}$$

$$\lambda = \lambda_0 + \tan^{-1}\left(\frac{x \sin c}{\rho \cos\phi_0 \cos c - y \sin\phi_0 \sin c}\right) \tag{11}$$

Where $\rho = \sqrt{x^2 + y^2}$ and $c = 2\tan^{-1}\left(\frac{\rho}{2R}\right)$ and the two-argument form of the tangent function should be used for this calculation. We can finally define our transformation $\mathcal{T}$ for a pair of angles (ϕ, λ) as a stereographic mapping function that applies Eqs. 8 and 9 to ϕ and λ respectively, such that $\mathcal{T}\{(\phi, \lambda)\} = (x, y)$ and the inverse stereographic mapping $\mathcal{T}^{-1}$ as a function that applies Eqs. 10 and 11 to the coordinates x and y respectively, such that $\mathcal{T}^{-1}\{(x, y)\} = (\phi, \lambda)$. Since the stereographic projection is bijective, it is verified that $\mathcal{T}^{-1}\{\mathcal{T}\{(\phi, \lambda)\}\} = (\phi, \lambda)$, which grants us the kind of symmetry we were looking for, as portrayed by Eq. 5.

We will now present an example to clarify the method further and provide the reader with an intermediate step to

validate the equation and methods. Suppose two polygons with 3 vertices in GCS coordinates, in degrees:

$$\mathcal{L}_1^{NE} = \{(-80, -20), (-80, -150), (-80, 100)\}$$

and

$$\mathcal{L}_2^{NE} = \{(-85, 150), (-70, -70), (-70, 0)\}$$

Applying Eqs. 7 and 8 to $\mathcal{L}_1^{NE}$ and $\mathcal{L}_2^{NE}$ with $(\phi_0, \lambda_o) = (-90, 0)$ yields the following sets of pairs, truncated to three decimal places:

$$\mathcal{T}\{\mathcal{L}_1^{NE}\} = \mathcal{L}_1^E = \{(-381.277, -1047.552), (-557.391, -965.429), (1097.846, -193.579)\}$$

and

$$\mathcal{T}\{\mathcal{L}_2^{NE}\} = \mathcal{L}_2^E = \{(278.164, -481.794), (-2111.265, 768.437), (0.0, 2246.761)\}$$

Both sets of coordinates are depicted in Fig. 9, where curves following a geodesic have been traced within the polygon coordinates. Eqs. 10 and 11 can be applied to the coordinates

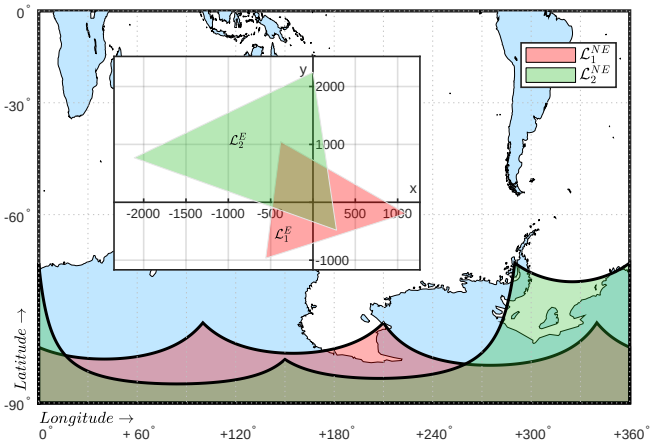


Fig. 9: $\mathcal{L}_1^{NE}$ and $\mathcal{L}_2^{NE}$ sets depicted over a portion of the Earth in a Miller cylindrical projection and the transformed (non-projected) sets $\mathcal{L}_1^E$ and $\mathcal{L}_2^E$ focused on Cartesian axes within the Figure

contained in $\mathcal{L}_1^E$ and $\mathcal{L}_2^E$ to obtain the original sets $\mathcal{L}_1^{NE}$ and $\mathcal{L}_2^{NE}$.

c) Method: Having the transformation, we can describe a recursive method to obtain AAP intersections, which also define our coverage areas served by N multiple satellites. Multiple options exist for performing polygon intersections, which are part of the family of the so-called *Polygon Boolean operations*. These operations are part of the Computational Geometry field, where applying Boolean algebra to polygons is a key aspect in computer graphics [42]. We will use the Martinez-Rueda Algorithm [43]. This algorithm extends the plane sweep technique [44] and works in three steps:

1) Subdivide the edges of the polygons at their intersection points.
2) Select those subdivided edges that lie inside the other polygon or not, depending on the operation.
3) Join the edges selected in step 2 to form the resulting polygon.

Taking the sets $\mathcal{L}_1^E$ and $\mathcal{L}_2^E$ from the example in paragraph III-B0b, applying the intersection algorithm, and truncating to three decimal places, we obtain:

$$\mathcal{L}_1^E \cap \mathcal{L}_2^E = \{(278.164, -481.794), (-480.355, -84.910), (-381.277, 1047.552), (169.356, 585.51)\}$$

at which point we can apply the anti-transformation $\mathcal{T}^{-1}$ to retrieve the geographic coordinates in the non-Euclidean space. In degrees:

$$\mathcal{T}^{-1}\{\mathcal{L}_1^E \cap \mathcal{L}_2^E\} = \{(-85.0, 150.0), (-85.615, -100.024), (-80, -20), (-84.522, 16.132)\}$$

Fig. 11 details the intersection $\mathcal{L}_1^E \cap \mathcal{L}_2^E$ in Cartesian axes, used to obtain $\mathcal{L}_1^{NE} \cap \mathcal{L}_2^{NE} = \mathcal{T}^{-1}\{\mathcal{L}_1^E \cap \mathcal{L}_2^E\}$ which corresponds to the required symmetry, according to Eq. 6. Fig. 10 shows examples of the transformation-intersection-inverse transformation procedures on semi-circular AAPs in the northern polar region.

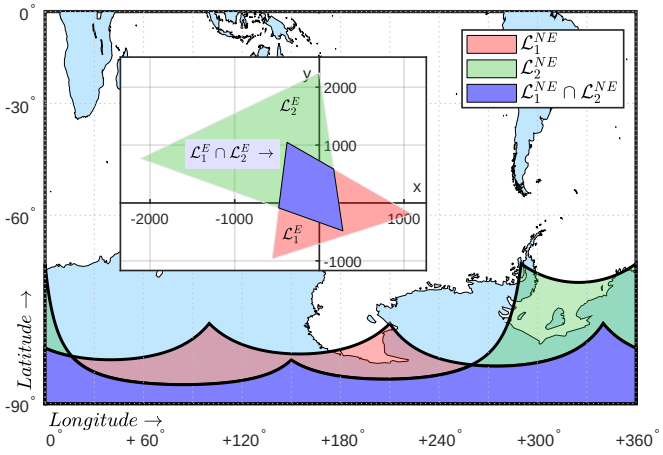
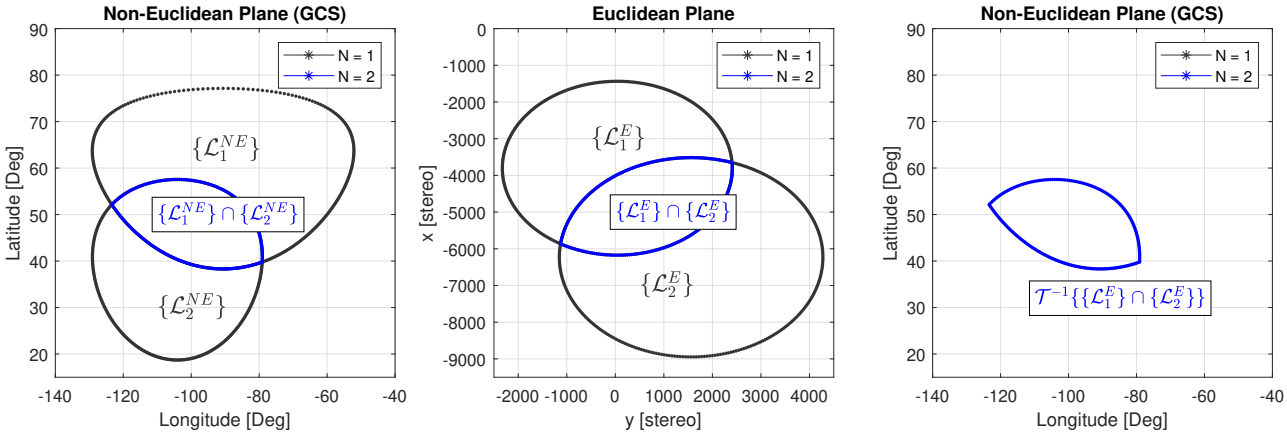
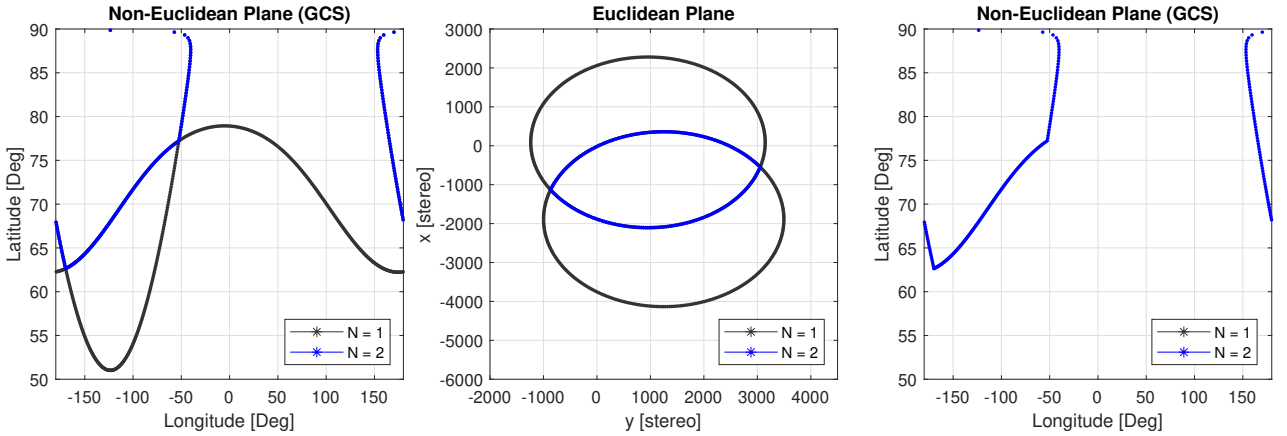


Fig. 11: $\mathcal{L}_1^{NE}$ and $\mathcal{L}_2^{NE}$ and their intersection over a portion of the Earth in Miller's cylindrical projection and the transformed (non-projected) sets $\mathcal{L}_1^E$, $\mathcal{L}_2^E$ and their intersection, focused on Cartesian axes within the Figure

If we have a constellation comprised of N satellites, we will have N starting AAPs over the surface of the Earth and, at most, all of them intersecting in a given instant in time. Nevertheless, this might be impossible due to design restrictions since constellations are usually designed to be spread apart, particularly in satellite communications, where sparse constellations show promising future [27], [45]. Furthermore, some constellations won't allow some AAPs intersections to occur. If we posed the case of two LEO satellites sharing



(a)



(b)

Fig. 10: (a) Intersection of two AAPs on the Non-Euclidean GCS (left) mapped to a stereographic projection (center) where the intersection is performed and then remapped onto the GCS (right). (b) Depiction of two AAPs intersecting over an Earth's Pole, showing the discontinuity that results from flattening the Non-Euclidean plane (left). Retrieval of the intersection (right) is possible since the stereographic projection (center) preserves information about the polygons' intersection points.

the same orbital plane and phased $180°$ in anomaly from each other, their AAPs will never intersect. It is possible to analyze the maximum number of intersecting AAPs that any N-sized constellation can allow, restricting the problem of finding all viable intersections. This number is given by the mathematical combination of the maximum $\mathcal{K}$ satellites capable of projecting overlapped areas, at any given moment in time, from a given constellation of size N. The possible k-combinations from a given set $\mathcal{S}$ of N elements are denoted in elementary combinatorics texts by $C(N,k)$ or by a variation such as C_N^k. The following formula gives this number:

$$\mathrm{C}_N^k = \frac{N!}{k!\,(N-k)!} \tag{12}$$

If k is such that $k <= \mathcal{K}$, then the number of operations $\mathcal{O}(\mathcal{K}, N)$ needed to analyze every possible AAPs intersection at any given point in time is:

$$^{2}\mathcal{O}(\mathcal{K}, N) = \sum_{k=2}^{\mathcal{K}} \mathrm{C}_N^k \tag{13}$$

which is the sum of all possible combinations given by equation 12 for all possible values of k. If we define a *snapshot* as a particular point in time within the scenario timespan t_{sc}, the dynamics of the analysis will be approximated by the discretization of said scenario in several *snapshots*, separated by a time step value ts. The number of *snapshots* needed for the complete scenario simulation is $N_{ts} = t_{sc}/ts$. Therefore the maximum required AAP operations N_{op} per simulation is:

$$N_{op} = N_{ts} \sum_{k=1}^{\mathcal{K}} \mathrm{C}_N^k, \tag{14}$$

where k is the number of satellites visible from our computed coverage area and starts at 1 because starting AAPs need to

[2]if starting AAPs are to be excluded from the numbers, then $k = 2$, otherwise $k = 1$.

be computed. Equation 14 gives us a notion of complexity for the computations needed, which is directly proportional to the constellation size and the simulation time and inversely proportional to the maximum possible amount of AAP intersections permitted by the constellation design.

C. The algorithm

Given a satellite set $\mathcal{S}$ of size N, and assuming we have the starting AAPs $\{\mathcal{L}_n^{NE}\}$ for $n = 1, 2, ..., N$, we need to generate all the subsets of possible combinations. These AAPs can depend on sensors, cameras FOVs, antenna radiation patterns, etc., which can be drawn about the Sub Satellite Point (SSP) provided by an orbital propagator. In this work, we will assume that a line of sight at TH degrees over the horizon from a GCS coordinates to a satellite is sufficient to consider favorable access. As an example, if we have 4 satellites, we need to compute the starting AAPs $\{\mathcal{L}_1^{NE}\}, \{\mathcal{L}_2^{NE}\}, \{\mathcal{L}_3^{NE}\}$, and $\{\mathcal{L}_4^{NE}\}$. If all four of them are capable of intersecting (due to the constellation design), then the amount of subsets needed is given by the equation 13:

$$\mathcal{O}(4,4) = \sum_{k=2}^{4} \mathrm{C}_4^4 = \sum_{k=2}^{4} \frac{4!}{k!\,(4-k)!} = 11 \tag{15}$$

If we refer to each AAP via their index n to simplify notation, as given by the satellites 1 through 4, all 11 possible subsets are $\{1,2\}$, $\{1,3\}$, $\{1,4\}$, $\{2,3\}$, $\{2,4\}$, $\{3,4\}$, $\{1,2,3\}$, $\{1,2,4\}$, $\{1,3,4\}$, $\{2,3,4\}$, $\{1,2,3,4\}$. Where $\{1,2\}$ is the pair of AAPs from satellites 1 and 2, $\{1,3\}$ is the pair from satellites 1 and 3, and so on. An algorithm that obtains all possible subsets from the constellation parameters is designed. This algorithm takes N and $\mathcal{K}$ as parameters. It gives back a collection of index sets $\mathcal{C}$, which can be used to obtain the necessary parameters of a satellite from a dictionary-like type of data structure. The next step is to map all AAPs into the Euclidean plane, applying the stereographic transformation $\mathcal{T}$ to perform the intersections. However, optimization is possible at this step. Suppose geodesics are traced between AAPs, depending on the footprint conditions. In that case, intersections can be quickly discarded if conditions that allow them are impossible: e.g., two satellites sufficiently far apart so that twice the Maximum Earth Central Angle of each line of sight FOV is less than the geodesic distance between the SSPs. After every impossible intersection has been filtered out, transformation and the subsequent intersection are performed on every subset.

$$\{\mathcal{L}_i^E\}_{\bigcap_{i=1}^n} = \bigcap_{i=1}^{n} \mathcal{T}\{\mathcal{L}_i^{NE}\} \tag{16}$$

and from this, we can obtain and store our desired intersection:

$$\mathcal{I} = \{\mathcal{L}_i^{NE}\}_{\bigcap_{i=1}^n} = \mathcal{T}^{-1}\{\mathcal{T}\{\mathcal{L}_i^{NE}\}\} \tag{17}$$

Intersection $\mathcal{I}$ is a collection of pairs of angles (latitudes and longitudes) in the Geographic Coordinate System. These collections can be fed to state-of-the-art GIS databases to obtain metrics like precise area values. When obtaining $\mathcal{I}$, information about the number of satellites with access to that particular intersection is preserved, as it is the size of the evaluated subset $\mathcal{C}$. Information about the specific satellites can also be preserved through their unique index, contained in $\mathcal{C}$. Every obtained intersection $\mathcal{I}$ at a given point in time (t) for the scenario is, in turn, stored in a set that represents every AAP occurring at the surface of the Earth for every combination of satellites involved. We will call this set $\mathcal{A}_t$ such that:

$$\mathcal{A}_t = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_{\mathcal{K}}\} \tag{18}$$

The collection of every set $\mathcal{A}_t$ such that $0 < t < t_{sc}$ where t_{sc} is the scenario time-span, and each t is spaced ts units of time between each other, as stated in section III-B, gives us a discrete analysis that models the coverage of the constellation over time. This dataset can be stored and used to perform analysis over several ROIs on the Earth without the need for re-propagating orbits. Thus, the obtained information can be re-used if the constellation's initial parameters do not change.

D. Percentage of Coverage

So far, the model outputs a set of intersections $\mathcal{A}_t$ that provides information about coverage regions and the number of satellites in sight for each one of them. This number can be between one and $\mathcal{K}$ as given in equation 13. Any number of accessed satellites k greater than one means an intersection occurs between AAPs with $k-1$ satellites in sight. If we intersect a ROI over the surface of the Earth with each $\mathcal{I}_i \in \mathcal{A}_t$ for each time-step in our given scenario, we can obtain the metric of Percentage of Coverage for at least k satellites for any k that we need. This is a crucial metric for designing constellations, giving a precise sense of shared access for a particular region. The visibility scale ranges from zero (or zero percent) when no satellites are in sight to one (or one hundred percent) when the entire ROI has visibility over at least k satellites. To obtain the Percentage of Coverage, a recursive intersection is performed between every member of $\mathcal{A}_t$, with a polygon $\mathcal{R}$ that defines the ROI, such that $\mathcal{R}$ is a set of GSC coordinates. The intersection is performed using the algorithm described in section III-C. At each time step, the resulting set of intersections $\mathcal{I}_R$ is subjected to a union operation, also called a boolean OR procedure, for each k possible satellite in sight. For each k the union $\mathcal{U}$ that will result from m obtained intersections is given by:

$$\mathcal{U} = \bigcup_{i=1}^{m} \mathcal{I}_{R_i} \tag{19}$$

The procedure described by equation 19 can yield one or more polygons. This depends on whether the polygons $\mathcal{I}_{R_i}$ intersect or not. If we define S as a function that takes a polygon $\mathcal{X}$ and computes the Coverage Area $(C.A.)$ for that particular polygon on the surface of the Earth, such that $C.A. = S(\mathcal{X})$, the Percentage of Coverage (PoC) for each k number of satellites is given by:

$$PoC_k = \frac{\sum_{i=1}^{n} S(\mathcal{U}_i)}{S(\mathcal{R})} \tag{20}$$

where equation 19 is assumed to yield n polygons.

E. Considerations and limitations

The stereographic transformation depends on the projection point, which needs to be chosen with criteria for a given distribution of satellites. Since the transformation is defined for every point but the reference, if a given polygon includes the reference point, algorithmic procedures have to be put in place to prevent indetermination in the equations: e.g., changing the reference or moving the problematic coordinate ever so slightly, since just making the numbers different would suffice, the introduced change should be within the error margins considered for the case study. Another consideration to take into account is the size of AAPs. If these areas cover more than 90 degrees over the surface of the Earth, certain projection points can distort the polygon's shape and cause unexpected results. Given their proximity to the Earth, LEO orbits won't exhibit this behavior. However, putting the projection reference inside these big polygons is a possible workaround, as shown in section IV-C. The amount of polygons that need to be analyzed at each time-step affects the computation time and depends on the size of the constellation and the number of satellite's AAPs that could feasibly intersect, meaning that a great number of satellites with a great number of possible interactions among them ramps up the algorithm's processing necessities. Then, the maximum possible number of combinations to be analyzed per time-step is given by Eq. 13 and the maximum number of operations per simulation by equation 14, both when $\mathcal{K} = N$. In these conditions, the number of operations grows exponentially with N, the constellation size. It quickly shoots up for values greater than $N = 25$, when operations are in the order of $33 \cdot 10^6$, compared to $2 \cdot 10^6$ if $\mathcal{K} = 8$ or $15 \cdot 10^3$ if $\mathcal{K} = 4$, at the same constellation size. As explained in Section III-B, the tendency towards sparse constellations and the desire to maximize access frequency over regions of interest makes situations where lots of satellite's AAPs overlap both unlikely and undesirable.

IV. Results

We present three case studies to test the model and compare our results with AGI's STK software. Percentage of coverage from at least N-satellites for three scenarios comprising a constellation and a ROI are used as the only constellation performance metric in this work, albeit measures of frequency such as coverage intervals or gaps can be obtained directly from the outputs of the algorithm, increasing the propagation time. The intent and purpose of these case studies are to evaluate how the proposed model performs concerning state-of-the-art software. Constellations, albeit feasible, are fictional. Nonetheless, the cases are inspired by real-life remote sensing applications. STK combines *coverage definition* and *area target* objects to define a grid used in a net-point procedure to obtain a Percentage of Access. Three granularity values (**g**) are used for the net-point methods in each case study. In all cases, the reference projection point is set to $(\phi_0, \lambda_o) = (0, 0)$, and 50 coordinates were used to draw the AAPs conic intersections, positioned at the point on Earth that complies with a visibility threshold over the horizon of 5 degrees. Points are placed at equal intervals considering a sweep angle from the SSP, starting in the directions of $\hat{\mathbf{e}}$ and $\hat{\mathbf{u}}$ (see Fig. 6), which means that coordinates are at $\frac{360^\circ}{50}$ degrees apart. The Binary-Search approach obtains the Earth's intersecting points with a convergence tolerance of 0.001 degrees.

All orbital elements used are dated to December 1^{st}, 2022, at precisely 19:00:00 UTCG, and true anomalies are used in the tables. Propagation is performed using a Simplified Perturbations Model SGP4 propagator with time-steps of 60 seconds, part of the Orekit project [46], based on David Vallado's work [47], [48]. This model calculates orbital state vectors of satellites relative to the Earth-centered inertial coordinate system with a simplified drag model and secular and periodic orbital perturbations caused by Earth's geometry, with errors of $\sim$1 km that grow at a rate of $\sim$1-3 km/day. The WGS84 system is used to model the Earth's shape: 6378.137 km of major semi-axis, 6356.75231424 km of minor semi-axis, and a flattening of 1/298.257223563. Statistical differences are presented in Tables V, VIII and X where avg, max, and min are the average, maximum, and minimum absolute differences between models, σ and σ^2 are the standard deviation and variance, respectively. Minimum differences not informed in the tables are zero. A fourth row is added to each table corresponding to simulations performed without sub-sampling algorithms in STK for a granularity density of 0.5 degrees. Statistical values are truncated to three decimal places. The average of 10 simulations[3] is used for the computational times, both for STK (**STK**) and our model (**Mod**). Said times are truncated to the first decimal place.

A. First case

In this first case, we analyze three satellites whose AAPs intersect a ROI that encloses the South-American territory, known to have wide isolated areas such as the Amazonas rainforest, Patagonia, or the Andes mountain range, of great interest in satellite applications and remote sensing missions [49]–[51]. The coordinates that define the ROI polygon are presented in Table III. The satellite's elements are shown in Table IV.

TABLE IV: Orbital elements of the three satellites for the first case

Satellite	a [km]	e	i [deg.]	Ω [deg.]	ω [deg.]	ν [deg.]
1	6978.135	0	80	290	0	280
2	6978.135	0	80	290	0	300
3	6978.135	0	80	300	0	290

Propagation is performed for 30 minutes, starting at 18:50:00 UTCG. A snapshot of the three AAPs corresponding to the satellites can be observed in Fig. 12. The model is applied to the scenario, and the Percentage of Coverage is obtained for every k possible satellite in sight: 1, 2, or 3. Results are shown in Fig. 13. The satellites traverse their orbit from South to North, so an increase in PoC is expected until they reach a maximum, after which coverage starts to decrease

[3]A 6^{th} Gen. i5 computer with 16GB of RAM is used to perform the simulations.

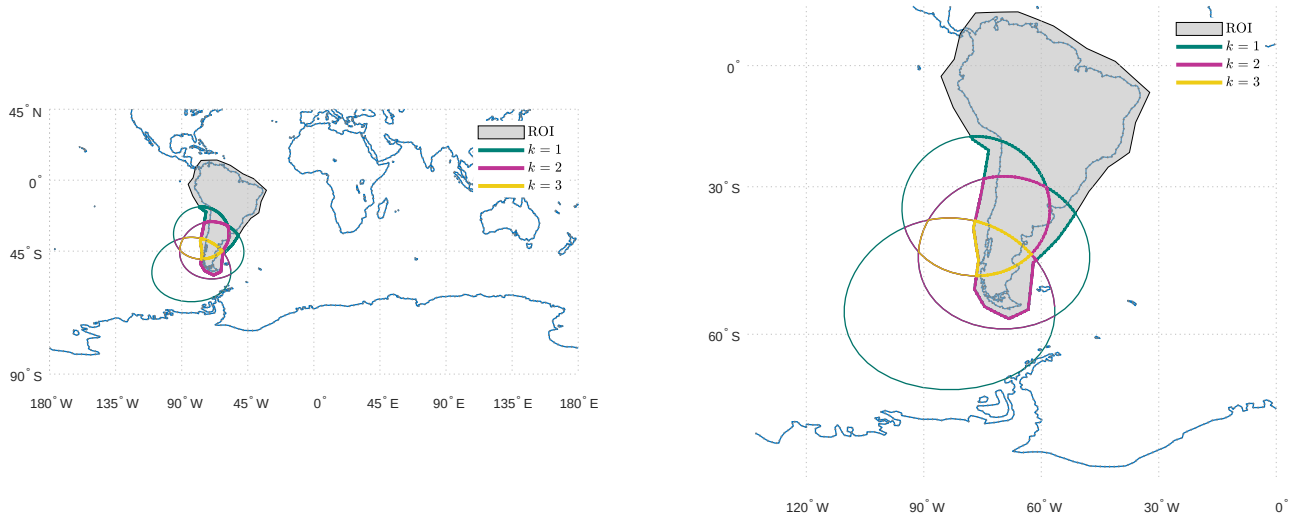


Fig. 12: Snapshot of three satellite AAPs intersecting a ROI over South America (first case study).

TABLE III: Defining coordinates for the ROI in the first case

Latitude [deg.]	Longitude [deg.]
-55.634522	-63.34879
13.353532	-76.798171
-46.63608	-62.055581
8.172611	-80.677799
-41.455159	-55.589533
1.628289	-82.229651
-31.093317	-47.830275
-2.734591	-85.591996
-25.367036	-42.916078
-10.642313	-82.488293
-21.822195	-37.484598
-18.550035	-77.574096
-14.459834	-35.932746
-21.276835	-73.435825
-6.824792	-32.311759
-31.093317	-75.246319
1.082929	-41.105585
-39.273719	-77.315454
4.35509	-48.347559
-46.09072	-76.022245
10.081371	-56.882742
-51.817001	-77.056812
13.626212	-65.93521
-55.089162	-74.470393
-57.270602	-68.262987

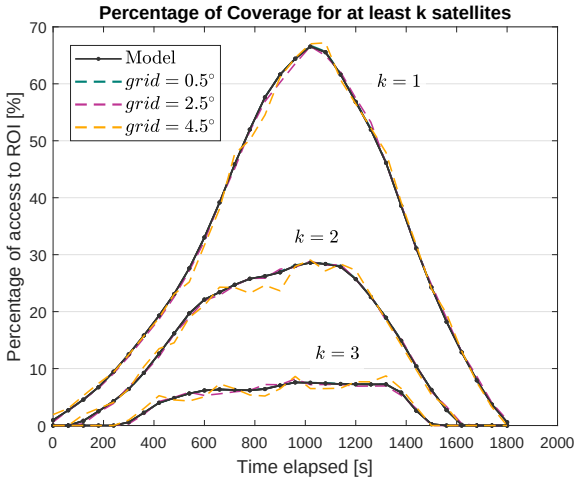


Fig. 13: Percentage of Coverage obtained for $k = 1, 2, 3$ satellites for the first case study.

as satellites leave the sight of the ROI. Statistical analysis of the differences between the proposed model and STK's outputs can be observed in Table V. These differences diminish as the grid granularity decreases, and the ROI is accessed by at least 1, 2, and 3 satellites at some point in the considered scenario.

B. Second case

For the second case, a constellation of 15 satellites is designed. The ROI, in this case, is a polygon enclosing Greenland, an arctic region of particular interest in space missions, due in part to its isolation and the fast changes it is experiencing due to global warming [52], [53]. The ROI coordinates are depicted in Table VI. The test constellation is comprised of three planes of 5 satellites each. The satellites are separated 20 degrees from each other in each plane, and each plane is 20 degrees apart from the next one in the Right Ascension of the Ascending Node (RAAN) parameter. Given this constellation, no more than 6 AAPs can intersect each other at the same time. Table VII displays the orbital elements of the first satellite in each plane. To obtain the elements of the remaining satellites, we increment the anomaly of each element by 20 degrees relative to the previous one. Propagation is performed for 60 minutes, starting at 19:00:00 UTCG. A snapshot of the constellation can be observed in Fig. 14. The model is applied to the scenario, and the percentage of Coverage is obtained for every k possible satellite in sight: 1 to 15. Results are shown in Fig. 15, and a statistical

TABLE V: Statistical differences between the proposed model (**Mod**) and STK (AGI's System Tool Kit) for the first case study.

			Number of satellites in sight (k)												
	Time [s]		1					2				3			
g [deg.]	**STK**	**Mod**	avg	max	min	σ	σ^2	avg	max	σ	σ^2	avg	max	σ	σ^2
0.5	23.8	0.5	0.092	0.210	0	0.102	0.010	0.054	0.177	0.075	0.005	0.025	0.112	0.037	0.001
2.5	2.1		0.449	1.378	0.007	0.556	0.309	0.264	0.728	0.333	0.111	0.231	0.876	0.336	0.113
4.5	0.8		0.829	3.118	0.017	1.133	1.285	0.750	3.248	1.046	1.095	0.537	1.483	0.729	0.532
0.5 (1)	13.8		4.891	12.288	5.10^{-4}	3.027	9.163	4.413	11.186	3.573	12.771	2.746	7.063	2.482	6.164

(1) Without using sub-sampling algorithms

TABLE VI: Coordinates that define the polygon around Greenland for the second case study.

Latitude [deg.]	Longitude [deg.]
59.639134	-46.905107
82.562197	-20.073398
58.907547	-44.245067
84.756959	-27.475249
60.126859	-42.163297
84.87889	-37.768448
62.565482	-41.122411
83.903441	-48.061647
64.39445	-38.924987
83.293785	-59.280077
65.004106	-36.264947
82.196404	-64.600157
65.857624	-33.373599
81.099023	-68.532391
67.198868	-31.291828
79.026193	-73.968125
67.198868	-28.284826
77.197225	-73.505509
68.296248	-24.468247
75.855982	-71.192431
69.881354	-20.882975
74.63667	-67.260197
72.198046	-20.536013
75.002464	-63.906234
73.905083	-18.338589
74.514739	-58.933115
76.221776	-17.297704
72.198046	-56.735691
79.635849	-15.909857
69.637491	-55.579151
80.855161	-11.168046
65.857624	-54.306958
83.049922	-11.861969
63.175138	-52.687803
60.858446	-50.374725

TABLE VII: Orbital elements for the first satellite on each plane

Satellite	a [km]	e	i [deg.]	Ω [deg.]	ω [deg.]	ν [deg.]
1	6778.135	0	80	250	0	220
2	6978.135	0	80	270	0	220
3	6978.135	0	80	290	0	220

analysis comparing the results with STK's outputs is shown in Table VIII. Statistical differences between our model and STK diminish as the grid granularity increases. Our model and STK show that overlapped access to at least 7 satellites is never achieved throughout the analyzed period, with zero percentage of access for $k = 7, 8$. This indicates that coverage can be achieved with at most 6 satellites in this scenario.

C. Third case

In this third case, we analyze seven satellites. Five are in LEO, while the remaining are in Medium and Geosynchronous Earth Orbits. The ROI is defined as a region over the Caribbean, delimited between -90 and -60 degrees of longitude and 0 to 30 degrees of latitude. In this case, the square is drawn using straight lines rather than great circles.. The Caribbean region is of great interest in satellite applications, particularly in disaster management, given the number of tropical storms endured by the region each year [54], [55]. This region is also continuously monitored with Geostationary satellites, given its proximity to the equator. It is continuously watched, for example, by the Geostationary Operational Environmental Satellites (GOES) operated by the National Oceanic and Atmospheric Administration (NOAA). Table IX shows the satellite's elements. The maximum allowed number of overlapping satellites is set to 5. Propagation is performed for 30 minutes, starting at 18:50:00 UTCG. A snapshot of the seven AAPs corresponding to the satellites can be observed in Fig. 16 and results both in Fig. 17 and Table X. As with the previous case studies, our polygon-based approach and the STK results converge as the grid granularity increases. Our analysis shows that the ROI is always accessible by at least one satellite, as expected with the deployed geostationary satellite. Additionally, our results indicate that no more than four satellites can access the ROI simultaneously, which aligns with the orbital plane phasing of the five LEO satellites.

V. Conclusions

In this work, we have proposed and tested a methodology to obtain coverage areas over the surface of the Earth. The presented model permits the systematization of procedures that obtain precise coverage metrics for Satellite Constellations. We designed and implemented an alternative to traditional net-point-based techniques to obtain coverage metrics by leveraging key polygon-based Boolean operations algorithms and oblate-Earth models. By revising the Coverage Area Determination for Conical Fields of View Considering an Oblate Earth algorithm and proposing an alternative way to determine Earth intersections, we increased the convergence speed potentially. This could result in a reduction of the number of iterations needed for each point, from possibly hundreds or thousands to tens, depending on the step and limits used. Results show that Access Area Polygons can be obtained through recursive algorithms at each time step in a desired scenario simulation. These procedures can be automatized

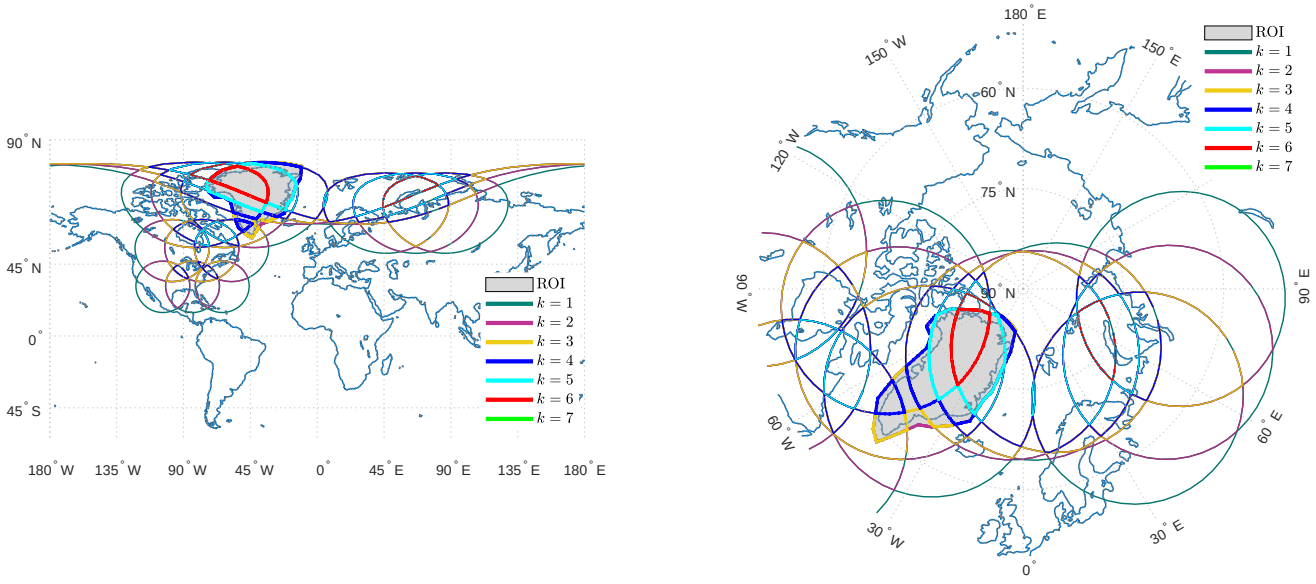


Fig. 14: Snapshot of 15 satellite's AAPs intersecting a ROI over Greenland (second case study).

TABLE VIII: Statistical differences between the proposed model (**Mod**) and STK (AGI's System Tool Kit) for the second case study.

			Number of satellites in sight (k) (1)																							
	Time [s]		1				2				3				4				5				6			
g [deg.]	STK	Mod	avg	max	σ	σ^2	avg	max	σ	σ^2	avg	max	σ	σ^2	avg	max	σ	σ^2	avg	max	σ	σ^2	avg	max	σ	σ^2
0.5	49.7	13.2	0.584	8.539	1.6997	2.8891	0.634	6.556	1.597	2.552	0.799	5.838	1.693	2.869	0.584	6.439	1.281	1.642	0.619	5.643	1.325	1.756	0.262	3.225	0.682	0.465
2.5	4.9		0.735	8.698	1.905	3.630	0.527	6.535	1.511	2.284	1.341	6.416	2.265	5.130	1.158	7.887	2.205	4.862	1.173	9.091	2.295	5.268	0.594	6.186	1.402	1.966
4.5	3.6		1.223	13.631	3.159	9.980	1.054	11.059	2.965	8.793	2.078	12.231	3.935	15.488	2.209	15.643	4.278	18.305	2.344	14.113	4.425	19.581	1.210	10.729	2.785	7.757
0.5 (2)	31.2		2.729	34.380	6.369	40.570	2.601	27.168	5.916	35.008	6.205	42.501	8.877	78.805	16.277	68.372	22.803	520.001	11.504	55.864	17.001	289.044	3.473	19.890	6.597	43.532

(1) Results for $k = 7, 8$ are zero.
(2) Without using sub-sampling algorithms.

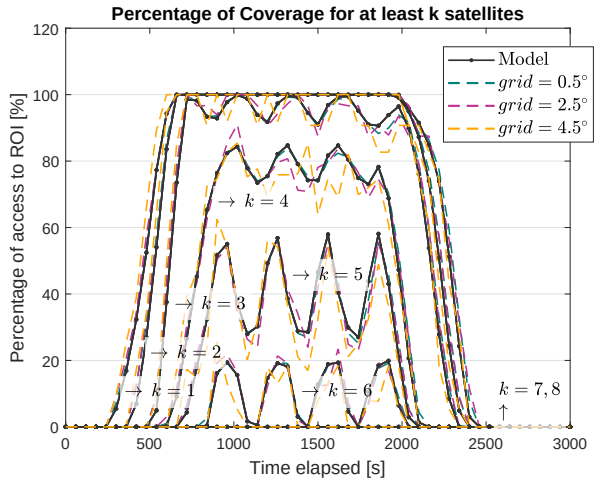


Fig. 15: Percentage of Coverage obtained for $k = 1 - 8$ satellites for the second case study. Results for $k = 7, 8$ are zero.

to yield results for any part of the Earth. In particular, we showed that increasing the precision of traditional net-point algorithms (increasing the grid's granularity) in trusted state-of-the-art software delivers results that approach the ones from our proposed model at lower speeds. For the case studies

TABLE IX: Orbital elements of the seven satellites for the third case study

Satellite	a [km]	e	i [deg.]	Ω [deg.]	ω [deg.]	ν [deg.]
1	42166.2587	0	80	290	0	285.6
2 (1)	26560.721	0.009092	53.266	135.682	33.188	157.627
3	6978.135	0	60	300	30	-180
4	6978.135	0	60	300	30	-160
5	6978.135	0	60	300	30	-140
6	6978.135	0	60	300	30	-120
7	6978.135	0	60	300	30	-100

(1) A GPS satellite elements are used

presented, we obtained increments between 3 and 73 times in computation speed compared to the net-point method at 0.5 degrees of granularity, as shown in Tables V, VIII and X. Although a trade-off between speed and granularity can be made for the net point approach, this increases the differences with the proposed model and does not always result in better time performance, as shown in the third case study results, in Table X. There, even with a 4.5-degree granularity, our method is shown to be faster. We also demonstrated that depriving STK of its convergence algorithms significantly increases the differences obtained, going as high as 68%, as shown in the second case study, hinting that time-dependent propagation procedures based on numerical integration, such as High Precision Orbital Propagators, would further increase the computation time needed to obtain precise rise and set

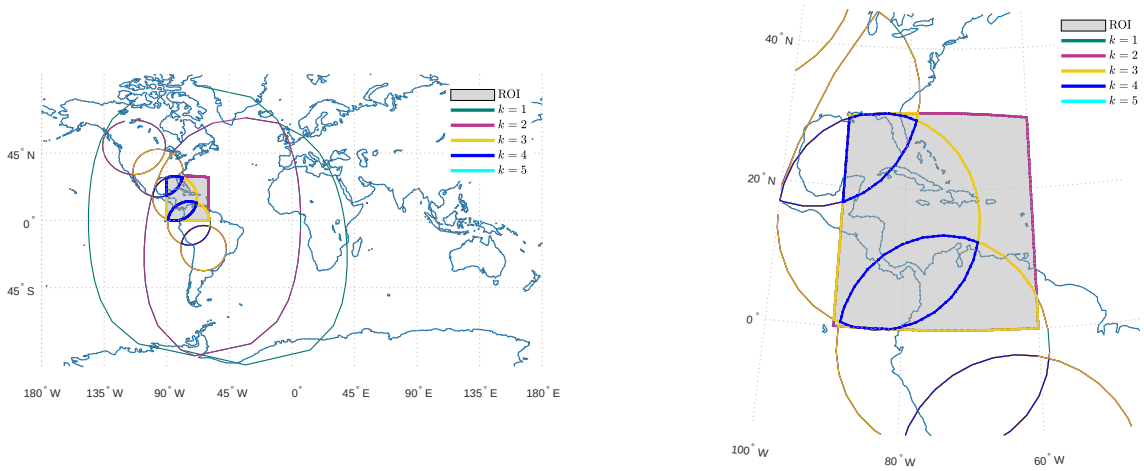


Fig. 16: Snapshot of five LEO, a MEO, and a GEO satellite's AAPs intersecting a ROI over the Caribbean (third case study).

TABLE X: Statistical differences between the proposed model (**Mod**) and STK (AGI's System Tool Kit) for the third case study.

			Number of satellites in sight (k) (1)											
	Time [s]		2				3				4			
g [deg.]	**STK**	**Mod**	avg	max	σ	σ^2	avg	max	σ	σ^2	avg	max	σ	σ^2
0.5	27.3	1.4	0.096	0.866	0.230	0.053	0.525	1.934	0.519	0.270	1.171	2.501	0.985	0.971
2.5	3.9		0.132	1.293	0.329	0.108	2.126	4.485	1.319	1.742	1.756	5.632	1.688	2.851
4.5	2.3		0.132	1.293	0.329	0.108	2.149	6.953	2.105	4.434	2.959	9.950	2.956	8.739
0.5 (2)	18.1		0.077	0.866	0.209	0.044	5.208	8.488	2.500	6.253	10.550	24.087	8.317	69.184

(1) Differences for $k = 1$ and $k = 5$ where 0.
(2) Without using sub-sampling algorithms.

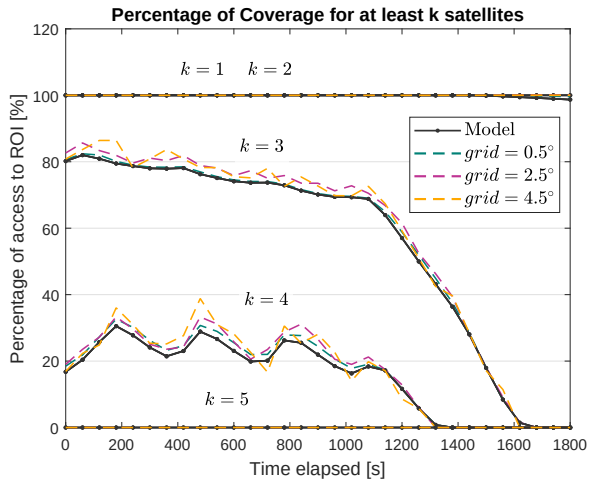


Fig. 17: Percentage of Coverage obtained for $k = 1 - 5$ satellites, third case study.

times for points in the surface of the Earth. Our model also avoids problems with polar discontinuities arising from using GCS coordinates as Euclidean pairs. As stated in the introduction, the presented techniques significantly impact space mission design and analysis, particularly those comprising LEO Satellite Constellations. We have shown throughout our results that a significant increase in speed for evaluating design and operation scenarios can be achieved without sacrificing the model's precision for state-of-the-art software. Many satellite applications and satellite-based services need information about overlapping coverage access. They can benefit from an automatized, open-source, fast, and precise model, especially those in the field of telecommunications, where the modeling of communication traffic and network throughput represents the cutting-edge technologies being studied. For future work, we identify four main lines of development:

1) Optimizing and improving the model: procedures such as ban lists or satellite availability windows can be implemented to optimize the algorithms further. e.g., instead of comparing satellite positions, a ban list can be consulted to determine if two satellites can intersect. This can also be applied to establish constraints with time-dependant availability windows, which bans communication and therefore overlapping, using metrics such as energy or sun exposure.
2) Using different kinds of sensors: apart from the line of sight access areas or simple conic sensors, some other type of sensors should be explored, such as rectangular or even an antenna's radiation pattern. Procedures for constructing these AAPs over an oblate surface constitute an interesting line of research.
3) Constellation optimization: applying this model as part of an optimization algorithm that takes action on constellation parameters (such as orbital elements) based on

an output metric that reflects any multi-asset operation aspect and iterates over the system to find optimal scenarios.

4) Network models: the proposed techniques can be used in synergy with ground-based telecommunication models to adapt or evolve them so that they can be used to predict a network's behavior based on satellite constellations [56].
5) Network optimization: as a final goal, we aim to optimize the operation of a satellite network considering both network and constellation models so that communication parameters and protocols can be dynamically optimized to achieve optimality in terms of its capabilities.

ACKNOWLEDGEMENT

This research has received support from Project STICAMSUD 21-STIC-12, the European Union's Horizon 2020 R&D program under the Marie Skłodowska-Curie grant agreement No 101008233 (MISSION project), and the French National Research Agency (ANR) under the project ANR-22-CE25-0014-01.

REFERENCES

[1] Z. Song, X. Hu, M. Wang, and G. Dai, "Judgement theorems and an approach for solving the constellation-to-ground coverage problem," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–10, 02 2018.

[2] J. Wertz, *Orbit Constellation Design and Management*, 1st ed., ser. Space Technology Library. Dordrecht: Springer Dordrecht, 2001.

[3] X. Chen, Z. Song, G. Dai, M. Wang, E. Ortore, and C. Circi, "A general formal method for manifold coverage analysis of satellite constellations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 2, pp. 1462–1479, 2022.

[4] N. Saeed, A. Elzanaty, H. Almorad, H. Dahrouj, T. Y. Al-Naffouri, and M.-S. Alouini, "Cubesat communications: Recent advances and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1839–1862, 2020.

[5] H. Jones, "The recent large reduction in space launch cost," in *The Recent Large Reduction in Space Launch Cost*. 48th International Conference on Environmental Systems, 2018, pp. 1–10.

[6] L. A. Singh, W. R. Whittecar, M. D. DiPrinzio, J. D. Herman, M. P. Ferringer, and P. M. Reed, "Low cost satellite constellations for nearly continuous global coverage," *Nature Communications*, vol. 11, no. 1, p. 200, Jan 2020.

[7] Y. Lee and J. P. Choi, "Connectivity analysis of mega-constellation satellite networks with optical intersatellite links," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 4213–4226, 2021.

[8] M. N. Sweeting, "Modern small satellites-changing the economics of space," *Proceedings of the IEEE*, vol. 106, no. 3, pp. 343–361, 2018.

[9] B. Di, H. Zhang, L. Song, Y. Li, and G. Y. Li, "Ultra-dense leo: Integrating terrestrial-satellite networks into 5g and beyond for data offloading," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 47–62, 2019.

[10] D. R. Luders, "Satellite networks for continuous zonal coverage," *ARS Journal*, vol. 31, no. 2, pp. 179–184, 1961. [Online]. Available: https://doi.org/10.2514/8.5422

[11] Y. Seyedi and S. M. Safavi, "On the analysis of random coverage time in mobile leo satellite communications," *IEEE Communications Letters*, vol. 16, no. 5, pp. 612–615, 2012.

[12] G. V. Mozhaev, "The problem of the continuous earth coverage and the kinematically regular satellite networks," *Cosmic Research*, vol. 11, p. 755, nov 1973.

[13] ——, "Capabilities of kinematically regular satellite systems with symmetry groups of the second type in the problem of continuous single coverage of the earth," *Cosmic Research*, vol. 43, no. 3, pp. 205–212, May 2005.

[14] P. Sengupta, S. Vadali, and K. Alfriend, "Satellite orbit design and maintenance for terrestrial coverage," *Journal of Spacecraft and Rockets*, vol. 47, pp. 177–187, 01 2010.

[15] H. Wang, C. Han, S. Liu, Y. Sun, and H. Zhang, "Adaptive algorithm to determine the coverage belt for agile satellite with attitude maneuvers," in *2019 IEEE 10th International Conference on Mechanical and Aerospace Engineering (ICMAE)*, 2019, pp. 241–246.

[16] Z. Song, H. Liu, D. Guangming, W. Maocai, and C. Xiaoyu, "Cell area-based method for analyzing the coverage capacity of satellite constellations," *International Journal of Aerospace Engineering*, vol. 2021, no. 20, p. 10, 2021.

[17] Y. Ulybyshev, "Satellite constellation design for complex coverage," *Journal of Spacecraft and Rockets*, vol. 45, no. 4, pp. 843–849, 2008. [Online]. Available: https://doi.org/10.2514/1.35369

[18] ——, "General analysis method for discontinuous coverage satellite constellations," *Journal of Guidance Control and Dynamics*, vol. 38, pp. 2475–2483, 2015.

[19] Y. Zhang, S. Bai, and C. Han, "Geometric analysis of a constellation with a ground target," *Acta Astronautica*, vol. 191, 11 2021.

[20] N. Crisp, S. Livadiotti, and P. Roberts, "A semi-analytical method for calculating revisit time for satellite constellations with discontinuous coverage," Cornell University, WorkingPaper, Jul. 2018.

[21] Y. N. Razoumny, "Fundamentals of the route theory for satellite constellation design for earth discontinuous coverage. part 1: Analytic emulation of the earth coverage," *Acta Astronautica*, vol. 128, pp. 722–740, 2016.

[22] ——, "Fundamentals of the route theory for satellite constellation design for earth discontinuous coverage. part 2: Synthesis of satellite orbits and constellations," *Acta Astronautica*, vol. 128, pp. 741–758, 2016.

[23] H. Maehara, "On the intersection graph of random caps on a sphere," *European Journal of Combinatorics*, vol. 25, no. 5, pp. 707–718, 2004.

[24] ——, "On a condition for the union of spherical caps to be connected," *Journal of Combinatorial Theory, Series A*, vol. 101, no. 2, pp. 264–270, 2003.

[25] M. Nugnes, C. Colombo, and M. Tipaldi, "Coverage area determination for conical fields of view considering an oblate earth," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 10, pp. 2233–2245, oct 2019. [Online]. Available: https://doi.org/10.2514%2F1.g004156

[26] "Compass project," accessed: 2022-09-01. [Online]. Available: https://www.compass.polimi.it/

[27] G. M. Capez, S. Henn, J. A. Fraire, and R. Garello, "Sparse satellite constellation design for global and regional direct-to-satellite iot services," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–16, 2022.

[28] H. Li, D. Li, and Y. Li, "A multi-index assessment method for evaluating coverage effectiveness of remote sensing satellite," *Chinese Journal of Aeronautics*, vol. 31, no. 20, pp. 2023–2033, 2018.

[29] C. S. Allen, M. Giraudo, C. Moratto, and N. Yamaguchi, "Chapter 4 - spaceflight environment," in *Space Safety and Human Performance*, T. Sgobba, B. Kanki, J.-F. Clervoy, and G. M. Sandal, Eds. Butterworth-Heinemann, 2018, pp. 87–138.

[30] Z. Song, G. Dai, M. Wang, and X. Chen, "A novel grid point approach for efficiently solving the constellation-to-ground regional coverage problem," *IEEE Access*, vol. 6, p. 44445–44458, 2018.

[31] G. Dai, X. Chen, M. Wang, E. Fernández, T. N. Nguyen, and G. Reinelt, "Analysis of satellite constellations for the continuous coverage of ground regions," *Journal of Spacecraft and Rockets*, vol. 54, no. 6, pp. 1294–1303, 2017.

[32] W. Y. O. Shaojun Feng and R. Mautz, "An area computation based method for raim holes assessment," *Journal of Global Positioning Systems*, vol. 5, no. 1, pp. 11–16, 2006.

[33] P. Chiberre, E. Meinhardt-Llopis, C. d. Franchis, and G. Facciolo, "3d modeling of earth's surface: Study of the antarctica," *International Geoscience and Remote Sensing Symposium*, pp. 1088–1091, 2019.

[34] D. G. Post and M. B. Eisen, "How long is the coastline of law? thoughts on the fractal nature of legal systems," *Journal of Legal Studies*, vol. 29, p. 545, 2000.

[35] A. Talgat, M. A. Kishk, and M. S. Alouini, "Stochastic geometry-based analysis of leo satellite communication systems," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2458–2462, 2021.

[36] H. Chen, H. Wu, Z.-L. Li, and J. Tu, "An improved computational geometry method for obtaining accurate remotely sensed products via convex hulls with dynamic weights: A case study with leaf area index," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2308–2319, 2019.

[37] Y. Lee, J. Jeong, J. Yun, W. Cho, and K. J. Yoon, "Spherephd: Applying cnns on 360 degrees images with non-euclidean spherical polyhedron representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 834–847, 2022.

[38] J. P. Snyder, *Flattening the Earth: Two Thousand Years of Map Projections*. The University of Chicago Press, 1993.

[39] M. Nugnes, C. Colombo, and M. Tipaldi. Coverage area determination considering an oblate earth. Accessed on 31/01/2023. [Online]. Available: https://it.mathworks.com/matlabcentral/fileexchange/81848-coverage-area-determination-considering-an-oblate-earth

[40] K.-T. Chang, *Introduction to Geographic Information Systems*. McGraw-Hill, 01 2008.

[41] M. Nyrtsov, M. Fleis, M. Borisov, and P. Stooke, "Conic projections of the triaxial ellipsoid: The projections for regional mapping of celestial bodies," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 52, no. 4, p. 322, 2017.

[42] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer Berlin, Heidelberg, 2008. [Online]. Available: https://doi.org/10.1007/978-3-540-77974-2

[43] F. Martínez, C. Ogayar, J. R. Jiménez, and A. J. Rueda, "A simple algorithm for boolean operations on polygons," *Advances in Engineering Software*, vol. 64, pp. 11–19, 2013.

[44] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, 1st ed. Springer New York, NY, 01 2008.

[45] J. A. Fraire, S. Henn, F. Dovis, R. Garello, and G. Taricco, "Sparse satellite constellation design for LoRa-based direct-to-satellite Internet of things," in *IEEE GLOBECOM*. IEEE, 2020, pp. 1–6.

[46] L. Maisonobe, V. Pommier, and P. Parraud, "Orekit: An open source library for operational flight dynamics applications," in *4th International Conference on Astrodynamics Tools and Techniques*, 2010, pp. 3–6.

[47] D. A. Vallado and W. D. McClain, *Fundamentals of astrodynamics and applications*, 4th ed. Hawthorne, California: Microcosm Press, 2013, vol. The space technology library.

[48] D. Vallado, P. Crawford, R. Hujsak, and T. Kelso, *Revisiting Spacetrack Report #3*. AIAA, 2006. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2006-6753

[49] M. E. Pritchard, J. Biggs, C. Wauthier, E. Sansosti, D. W. D. Arnold, F. Delgado, S. K. Ebmeier, S. T. Henderson, K. Stephens, C. Cooper, K. Wnuk, F. Amelung, V. Aguilar, P. Mothes, O. Macedo, L. E. Lara, M. P. Poland, and S. Zoffoli, "Towards coordinated regional multi-satellite insar volcano observations: results from the latin america pilot project," *Journal of Applied Volcanology*, vol. 7, no. 1, p. 5, Jun 2018. [Online]. Available: https://doi.org/10.1186/s13617-018-0074-0

[50] J. C. D. M. Esquerdo, J. F. G. Antunes, A. C. Coutinho, E. A. Speranza, A. A. Kondo, and J. L. dos Santos, "Satveg: A web-based tool for visualization of modis vegetation indices in south america," *Computers and Electronics in Agriculture*, vol. 175, p. 105516, 2020.

[51] J. D. Restrepo A, A. J. Kettner, and G. Robert Brakenridge, "Monitoring water discharge and floodplain connectivity for the northern andes utilizing satellite data: A tool for river planning and science-based decision-making," *Journal of Hydrology*, vol. 590, p. 125123, 2020.

[52] A. Beiranvand Pour, T.-Y. S. Park, Y. Park, J. K. Hong, B. Zoheir, B. Pradhan, I. Ayoobi, and M. Hashim, "Application of multi-sensor satellite data for exploration of zn–pb sulfide mineralization in the franklinian basin, north greenland," *Remote Sensing*, vol. 10, no. 8, 2018. [Online]. Available: https://www.mdpi.com/2072-4292/10/8/1186

[53] J. An, B. Zhang, S. Ai, Z. Wang, and Y. Feng, "Evaluation of vertical crustal movements and sea level changes around greenland from gps and tide gauge observations," *Acta Oceanologica Sinica*, vol. 40, no. 1, pp. 4–12, Jan 2021. [Online]. Available: https://doi.org/10.1007/s13131-021-1719-0

[54] D. Stettner, C. Velden, R. Rabin, S. Wanzong, J. Daniels, and W. Bresky, "Development of enhanced vortex-scale atmospheric motion vectors for hurricane applications," *Remote Sensing*, vol. 11, no. 17, 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/17/1981

[55] C. Gang, S. Pan, H. Tian, Z. Wang, R. Xu, Z. Bian, N. Pan, Y. Yao, and H. Shi, "Satellite observations of forest resilience to hurricanes along the northern gulf of mexico," *Forest Ecology and Management*, vol. 472, p. 118243, 2020.

[56] B. Finley and A. Vesselkov, "Cellular iot traffic characterization and evolution," *CoRR*, vol. abs/1911.02877, 2019. [Online]. Available: http://arxiv.org/abs/1911.02877

Santiago Henn is currently pursuing a Ph.D. in Engineering Sciences at the Universidad Nacional de Córdoba (UNC/CONICET). He completed his Bachelor's degree in Electronic Engineering in 2017 and obtained a Master's degree in Satellite Technology in 2020. Santiago's research revolves around the application of informatics in the analysis and optimization of space systems, with a particular emphasis on developing innovative algorithms and design techniques for satellite constellation missions.

Juan A. Fraire is a researcher at the National Institute for Research in Digital Science and Technology (INRIA) in France and CONICET in Argentina, and an associate professor at Universidad Nacional de Córdoba (UNC) and Saarland University in Germany. His research focuses on spaceborne networking and distributed applications enabled by state-of-the-art informatics techniques. Juan is the founder of the annual Space-Terrestrial Internetworking Workshop (STINT) and has been its chair since 2014. He has co-authored over 75 papers published in international journals, conference proceedings, and the "Delay-Tolerant Satellite Network" book. Juan participates in joint projects with NASA's Jet Propulsion Laboratory (JPL) and the Argentinian Space Agency (CONAE) and collaborates with world-renowned space companies and start-ups.

Holger Hermanns is Professor of Computer Science at Saarland University, Saarbrücken, Germany, holding the Chair of Dependable Systems and Software on Saarland Informatics Campus. He has authored or co-authored more than 250 peer-reviewed scientific papers. His research interests include modeling and verification of concurrent systems, resource-aware embedded systems, compositional performance and dependability evaluation, and their applications to space and energy informatics. He is a member of Academia Europaea, ERC Advanced Grantee, and spokesperson of the Center for Perspicuous Computing – TRR 248 of the German Research Foundation DFG.