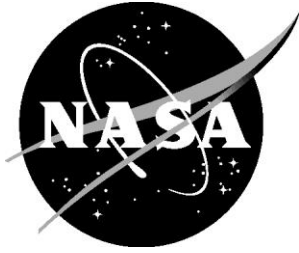


NASA/TP– 20220016787



Parameterization of Features on Spherical Surfaces

Douglas M. Trent
Langley Research Center, Hampton, Virginia
douglas.m.trent@nasa.gov

Zachary R. Yahn
University of Virginia, Charlottesville, Virginia
zry2yz@virginia.edu

Conor A. Nixon
NASA Goddard Space Flight Center, Greenbelt, Maryland
conor.a.nixon@nasa.gov

John W. Santerre
University of California, Berkeley, Berkeley, California
john.santerre@berkeley.edu

October 2023

NASA STI Program Report Series

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

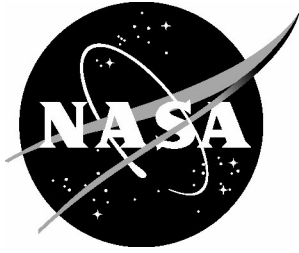
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- Help desk contact information: <https://www.sti.nasa.gov/sti-contact-form/> and select the "General" help request type.

NASA/TP–20220016787



Parameterization of Features on Spherical Surfaces

Douglas M. Trent
Langley Research Center, Hampton, Virginia
douglas.m.trent@nasa.gov

Zachary R. Yahn
University of Virginia, Charlottesville, Virginia
zry2yz@virginia.edu

Conor A. Nixon
NASA Goddard Space Flight Center, Greenbelt, Maryland
conor.a.nixon@nasa.gov

John W. Santerre
University of California, Berkeley, Berkeley, California
john.santerre@berkeley.edu

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

October 2023

Abstract

NASA's Planetary Systems Laboratory (PSL) at the Goddard Spaceflight Center in Greenbelt, Maryland is tasked with the study of solar system objects. This work is frequently based on analysis of imagery from visiting spacecraft, orbiting telescopes, and ground-based sensors.

A consequence of this mission is the need to characterize features on the spherical surface of planets and moons, such characterization including the overall area of the feature, its center, its moments of inertia, its perimeter, its compactness, its major axis, its bounding box, its aspect ratio and its North-South orientation. Accurate assessment of these measures from a "flat" two-dimensional image is a challenge, since the scale of distance observed at the center of a spherical object distorts in any direction towards the object's horizon.

PSL has developed a technique to guarantee accurate and automated measurement at any point on the visible surface, by overlaying an imaginary grid of equal-area cells onto the two-dimensional image of the sphere. The technique is named Grid-Oriented Normalization for Analysis of Spherical Areas (GONASA).

This paper discusses the construction of the normalization grid and describes specific algorithms for feature parameterization. The algorithms are implemented in an accompanying Excel spreadsheet, both for clarity and ease of adoption, and to emphasize their aptness for automation. Finally, a concrete use case is offered where the GONASA grid and algorithms are used to characterize methane clouds on the Saturnian moon Titan.

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Table of Contents

1.0 Introduction	4
1.1 Objective	4
1.2 Background	4
1.3 Proposal.....	4
1.4 Why Longitude/Latitude is not the solution	4
1.5 Thought Experiment - Why a grid of equal-area cells should be constructable.....	5
1.6 Properties of the GONASA Cartesian Iso-Long/Lat Grid	6
1.7 Literature Survey.....	7
2.0 Creating the Normalized Grid	12
2.1 Understanding the area of spherical rings.....	12
2.2 How to calculate latitude increments for equal-area rings	12
2.3 The nature of normalized latitude lines.....	14
2.4 How to position square grid cells to specific latitudes	15
2.5 How to choose the best grid dimensions.....	17
2.6 A note on the accompanying Excel spreadsheet	18
2.7 A note on grid cell resolution.....	18
3.0 GONASA Analysis Algorithms.....	19
3.1 Example Use Case	19
3.2 How to find the AREA of a surface feature	20
3.3 How to find the CENTROID of a feature.....	21
3.4 How to find the MOMENTS OF INERTIA of a feature.....	22
3.5 How to find the PERIMETER of a feature	24
3.6 How to find the COMPACTNESS of a feature.....	26
3.7 How to find the X-Y BOUNDING BOX & ASPECT RATIO of a feature.....	27
3.8 How to find the MAJOR AXIS and ORIENTATION of a feature	29
3.9 How to find the MAJOR AXIS BOUNDING BOX and ASPECT RATIO of a feature	31
4.0 Use Case – Methane Clouds on Titan	35
4.1 Motivation.....	35
4.2 Automated Cloud Analysis	35
4.3 GONASA Application	35
5.0 Summary	36
6.0 References.....	37

1.0 Introduction

1.1 Objective

The goal of this paper is to describe methods to enable the accurate and automated measurement of features on a spherical surface such as a planet or moon, given a two-dimensional representation of that surface in the form of a photograph or satellite image. The methods are collectively named Grid-Oriented Normalization for Analysis of Spherical Areas (GONASA). Algorithms that embody these methods are implemented in the accompanying GONASA.xlsx Excel spreadsheet.

1.2 Background

Viewed in a photograph, a sphere appears as a two-dimensional disk. The perceived area of a surface feature, observed in terms of two-dimensional image pixels, becomes increasingly distorted as the feature extends in any direction from the center of the disk towards the disk horizon. In particular, the perceived area of the feature shrinks towards the horizon becoming zero at the edge or limb of the disk. For large-scale features on the surface of a sphere that extend across such an expanse of distortion, computing area and other measurements can be difficult.

1.3 Proposal

If a grid of cells having equal surface area could be overlaid onto the two-dimensional image of the sphere, then any observed feature could be compared to the grid, the intersected grid cells counted, and the sum of the equal-area cells tallied to find the overall area of the feature. Other measures related to the dispersion of the area such as centroid, moments of inertia and perimeter could also be determined. With the perimeter of the feature defined, it would be straightforward to calculate the feature's major axis, orientation, bounding box, and aspect ratio.

If the grid were cartesian, consisting of rows with the same number of columns, the calculations would lend themselves to automated high-speed matrix computation. Automated feature analysis becomes increasingly important to planetary science as the number of missions and imaging capability grows, and as computation is moved to the spacecraft themselves to summarize results before transmission across limited bandwidth channels back to Earth.

1.4 Why Longitude/Latitude is not the solution

Longitude and latitude is a widely-used coordinate system for defining location on the surface of Earth and other astronomical bodies (Archinal, 2018). Equivalent coordinate systems (for example, right ascension and declination) are used from the inside of the sphere looking out, to define location on the celestial sphere (Capitaine, 1990). Longitude and latitude lines would seem to be a candidate for determining area, since they do form a cartesian grid of cells across the entirety of a sphere's surface. However, as lines of longitude converge at the poles, the area inscribed by their approaching lines becomes increasingly smaller. Cells formed by equally-spaced lines of latitude and longitude do not have equal area across the sphere.

1.5 Thought Experiment - Why a grid of equal-area cells should be constructable

- 1) Imagine a sphere marked with equally-spaced longitude lines. Choosing any two latitude boundaries will define a “ring” on the surface with the following characteristics:
 - a) The lines of latitude and longitude intersect locally at right angles on the surface.
 - b) The longitude-divided cells within the ring will all have equal area.
- 2) For subsequent rings, it would be a simple matter to choose a latitude boundary that yields a ring of equal area. Then the longitude-divided cells of the next ring are also guaranteed to have the same area as the cells of the first ring.
- 3) Some care must be taken to choose sets of latitude boundaries that cover the sphere contiguously and do not leave any overlapped or uncovered area.
- 4) Note that cells toward the pole will be stretched vertically more than cells near the equator. By choosing the number of longitude and latitude divisions, approximately square cells may be positioned at any desired latitude. In general, the latitude of square cells may be moved towards or away from the equator by increasing or decreasing, respectively, the number of longitude divisions.
- 5) Once defined, the lines of latitude and longitude are constant for all cells in a given row or column. The grid is considered to be *iso-longitudinal* and *iso-latitudinal*. The grid is also considered to be *cartesian* because the constructed grid is rectangular, with the same number of cells in each row.

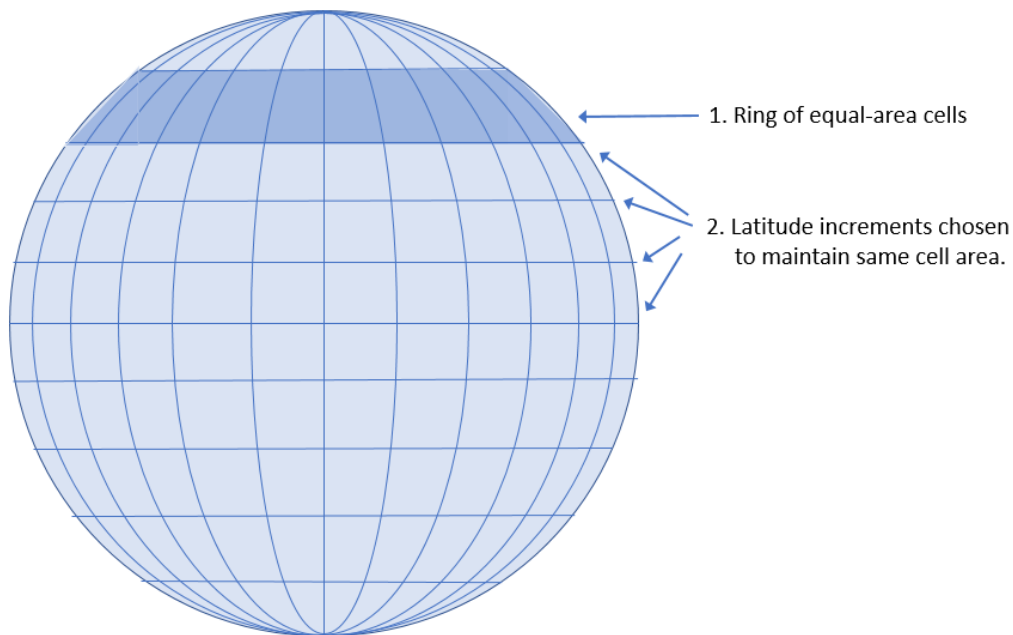


Figure 1. Latitude divisions chosen to create a grid of equal-area cells

1.6 Properties of the GONASA Cartesian Iso-Long/Lat Grid

The GONASA approach produces a grid of cells with the following properties:

Easy construction – The method is simple in concept and the grid is easy to construct without complicated formulae.

Cartesian – The grid of cells is rectangular and easily adapted to matrix computations, for example Pandas data frames in Python, or the rows and columns of an Excel spreadsheet.

Simple Cell indexing - Cells do not require a complicated numbering scheme and are simply indexed by their row and column. This reduces the complexity of heuristic analyses such as nearest-neighbor search.

Longitude/latitude basis: Cell boundaries are based entirely on longitude (right ascension) and latitude (declination), which are the coordinate systems commonly used in astronomy and geodesy.

Iso-longitudinal and iso-latitudinal – Once the longitude and latitude lines are defined, they are the same for all cells in a given row or column. Cell location values do not need to be computed on-the-fly or taken from a look-up table.

Single regime – The construction method is consistent across all points on the sphere, vs. for example, shifting methods between polar and equatorial regimes.

Equal-area cells – All cells are constructed to have exactly the same area, not approximately equal areas.

Variable grid resolution – Flexible construction allows easy rescaling in either the horizontal or vertical dimension (or both) to achieve varying cell granularity. This helps manage the tradeoff between model complexity and the desired level of measurement precision. Alternative methods that only allow successive doubling or halving of resolution are not considered flexible.

Square cells at any latitude – By adjusting the number of divisions of longitude and latitude, a ring of approximately square cells can be created at any latitude. Positioning these cells at the latitude of interest can be useful to preserve shapes and better measure distances of unknown direction. Square cells can also provide denser granularity for analyses related to area.

With these properties, the GONASA grid may be deployed to measure the following characteristics of surface features in an accurate and automated way:

- Area
- Centroid
- Moments of Inertia
- Perimeter and Compactness
- Major Axis and Orientation
- Bounding Boxes and Aspect Ratios

1.7 Literature Survey

The field of cartography is a logical place to look for related ideas and techniques. Maps of the Earth's spherical surface have a long history of importance to navigation and global trade. Since the time of Eratosthenes in the third century BC (Bunbury, 1883) many great thinkers have devoted time to this subject.

Indeed, the concept of generating a cartesian grid from a spherical system of meridians and parallels is the central idea of a cylindrical projection [Britannica, 2007]. The spherical coordinate system is projected onto a circumscribing cylinder that is then unrolled as a rectangular grid (Figure 2).

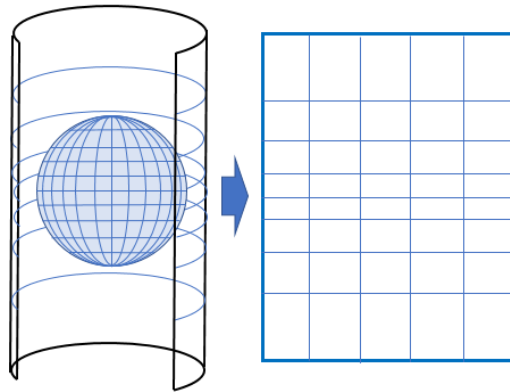


Figure 2. Cylindrical Projection

The Mercator Projection is a popular example of the cylindrical projection and is valuable to navigators precisely because its longitude and latitude intersect at right angles, just as they do locally on the Earth's sphere, thus preserving direction and orientation (Snyder, 1993). However, the Mercator Projection seriously distorts the size of objects near the poles (Figure 3). Greenland and Antarctica are not really that big!

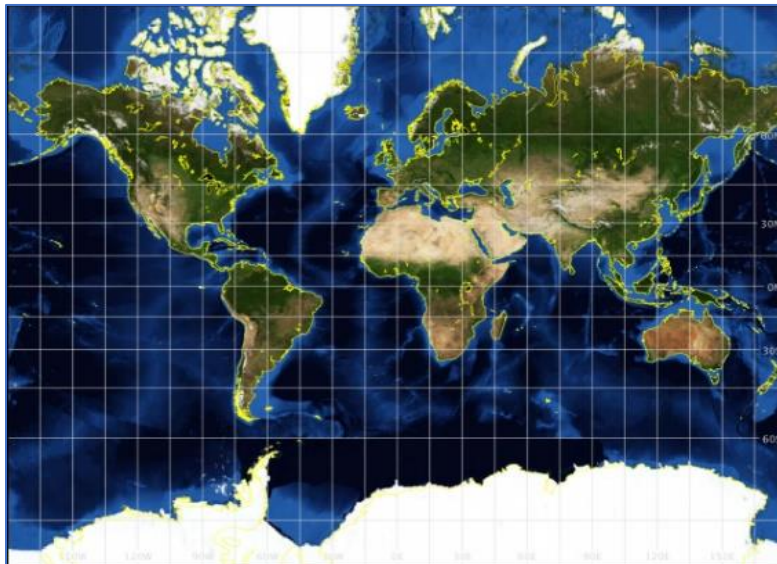


Figure 3. Mercator Projection (giss.nasa.gov)

The problem of size distortion is solved with the Lambert Cylindrical Equal-area Projection (Weisstein, 2020). Starting with equal longitude and latitude scale at the equator, the distance between latitude lines decreases towards the poles, in direct proportion to decreasing longitude width, such that the area of features is constant (Figure 4). This projection is analogous to the GONASA grid technique which increases latitude separation towards the poles to achieve a similar equal-area result.



Figure 4. Lambert Cylindrical Equal-Area Projection (giss.nasa.gov)

The ability to relocate square cells to other latitudes is captured in the Gall-Peters projection (Gall, 1885) which places longitude and latitude of equal scale at latitudes of 45° North and South (Figure 5). This is a singular instance of GONASA's general ability to position approximately square cells to any latitude.

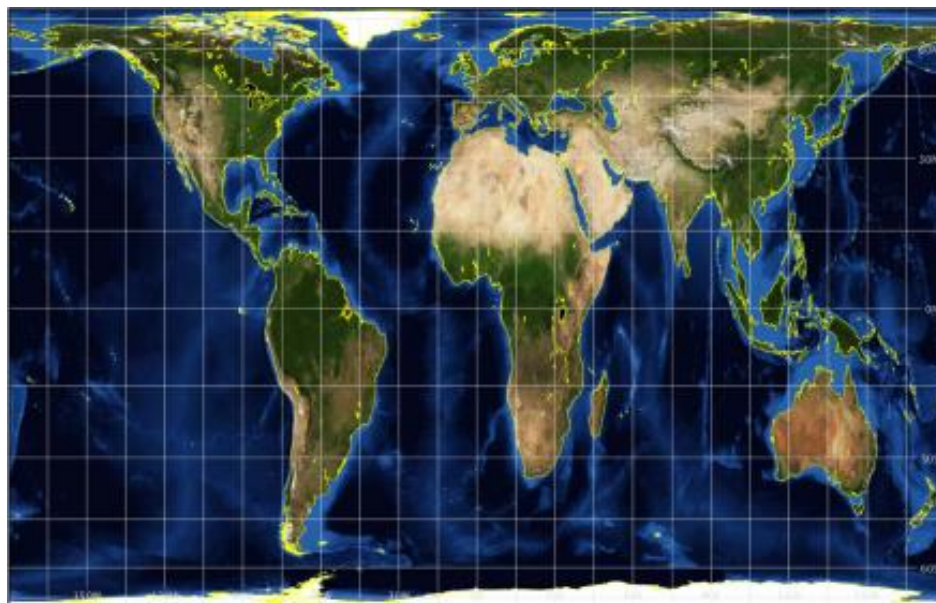


Figure 5. Gall-Peters Projection (giss.nasa.gov)

The idea of tessellating the sphere into a contiguous grid of approximately equal-area cells is well-established in theory (White, 1992) and in practice by the Quadrilateralized Spherical Cube (Chan, 1980), which projects the spherical surface inward onto the six faces of an inscribed cube (Figure 6). This technique was used with NASA's COBE Cosmic Background Explorer (Essinger-Hileman, 1989) and has the unique advantage of avoiding singularities at the poles. Cell areas are only approximately equal and are neither iso-longitudinal nor iso-latitudinal.

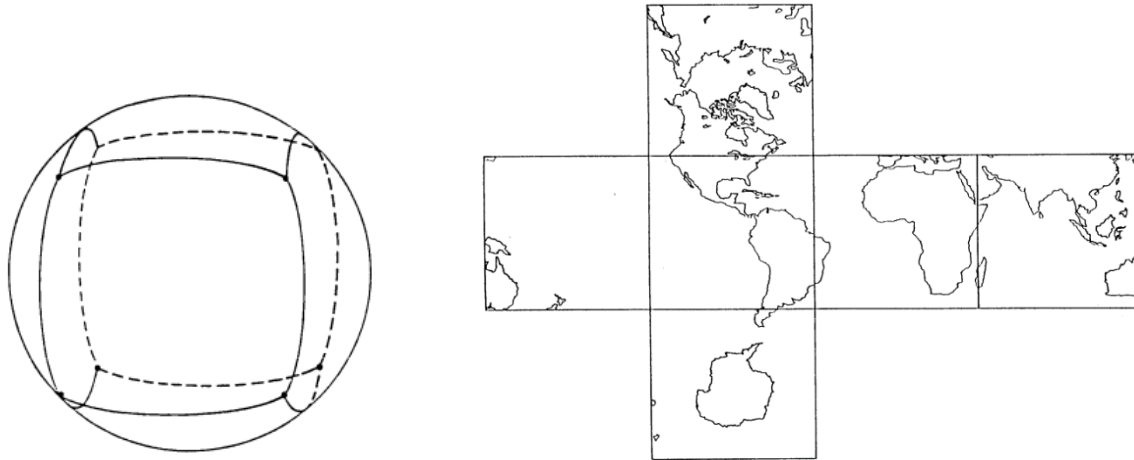


Figure 6. Quadrilateral Spherical Cube (Chan, 1980)

The HEALPix Hierarchical Equal Area isoLatitude Pixel framework (Gorski, 2005) also applies polyhedral faces to the sphere, but manages to place equal-area cells into rows of constant latitude to speed computation (Figure 7). The technique involves complex calculations for grid construction and cell indexing, and a different regime for polar vs. equatorial regions.

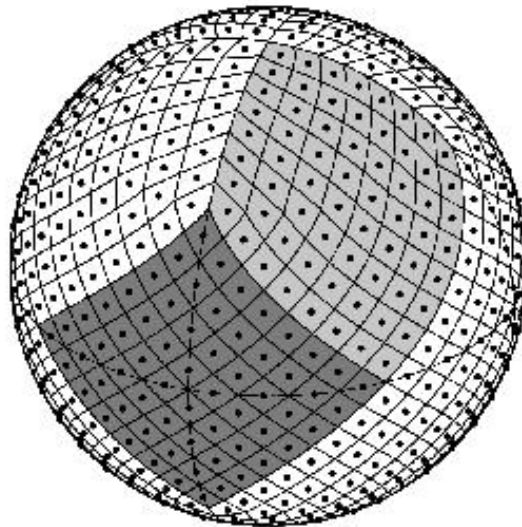


Figure 7. HEALPix Tessellation (healpix.jpl.nasa.gov - Courtesy NASA/JPL-Caltech)

(Roşca, 2010) achieves a single-regime grid of iso-latitudinal cells of truly equal area by applying a Lambert azimuthal projection (Figure 8). The construction calculations are simpler than HEALPix. The grid is not cartesian and the North-South curves have little relation to longitude.



Figure 8. Lambert Azimuthal Projection (Roşca, 2010)

Finally, the SREAG spherical rectangular equal-area grid (Malkin, 2019) uses iso-latitudinal rings of near-equal width to create a single regime of truly equal-area cells. The vertical divisions are based on longitude but their quantity is increased for each ring, progressing away from the poles (Figure 9). Therefore, the grid is not cartesian and the cell numbering/indexing is complex. SREAG construction is a bit heuristic, adjusting latitude divisions to accommodate an integer number of cell divisions per ring. It is based on the computational geometry of (Gringorten, 1992) and (Beckers, 2012).



90	1		2		3		
45	4	5	6	7	8	9	10
0	11	12	13	14	15	16	17
-45	18		19		20		
-90							
	0	120		240		360	

Figure 9. SREAG grid construction and non-Cartesian indexing scheme (Malkin, 2013)

A summary comparison of GONASA with the above tessellation techniques is shown in Figure 10.

Property	Quadrilateralized Spherical Cube (Chan)	HEALPix (Gorski)	Lambert Azimuthal Projection (Rosca)	SREAG (Malkin)	GONASA (Trent)
Easy Construction			X		X
Cartesian					X
Simple Cell Indexing					X
Long/lat basis				X	X
Iso-Latitudinal		X	X	X	X
Iso-Longitudinal					X
No singularity at poles	X				
Single Regime			X	X	X
Equal-Area cells		X	X	X	X
Variable Resolution					X
Square Cell Reposition					X

Figure 10. Comparison of Spherical Analysis Techniques

2.0 Creating the Normalized Grid

2.1 Understanding the area of spherical rings

The thought experiment of section 1.5 invites us to choose successive latitude boundaries that result in spherical rings of equal area. Thus we begin by examining the area of rings on a sphere and relating that to latitude. The area of the orange-colored spherical cap of height L in Figure 11 is equal to $2 \pi r L$ (Polyanin, 2006). A second spherical cap of height $2L$ will therefore have double the area and the difference of the two caps, the spherical ring shown in green, will have the same area as the orange cap. Subsequent rings of height L will also have equal areas.

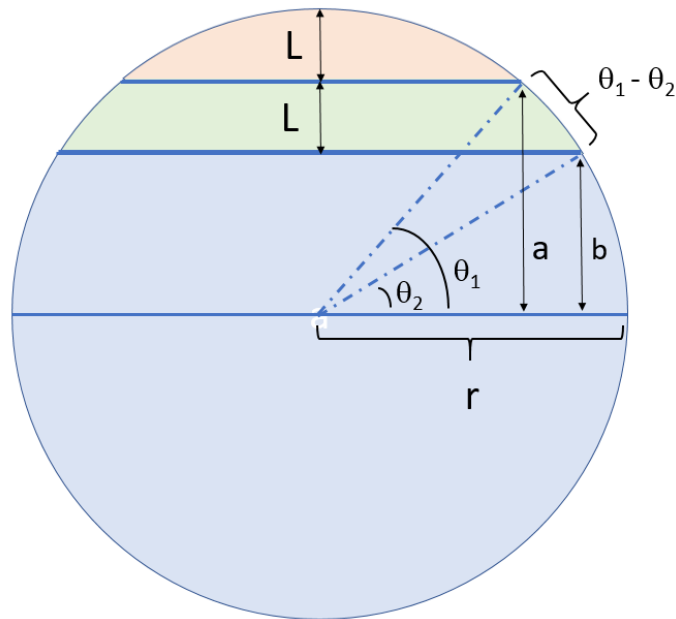


Figure 11. Area of spherical rings

Intuitively, we could slice the full sphere into an integer number (n) of divisions each having height $L = 2r/n$ and these divisions would have equal area. More formally, we can calculate L using Archimedes' formula for the area of a sphere = $4 \pi r^2$ (Beyer, 1987), dividing that into n equal-area rings, and solving for L :

$$4 \pi r^2 / n = 2 \pi r L$$

$$2 r / n = L$$

2.2 How to calculate latitude increments for equal-area rings

Referring again to Figure 11, the opposite sides of angles θ_1 and θ_2 will have lengths:

$$a = r \sin(\theta_1) \quad b = r \sin(\theta_2)$$

The ring height L is the difference of lengths a and b :

$$L = a - b = r [\sin(\theta_1) - \sin(\theta_2)]$$

Given θ_1 and L we can solve for θ_2 :

$$L = r [\sin(\theta_1) - \sin(\theta_2)]$$

$$L / r = \sin(\theta_1) - \sin(\theta_2)$$

$$\sin(\theta_2) = \sin(\theta_1) - (L / r)$$

$$\theta_2 = \arcsin [\sin(\theta_1) - (L / r)] \quad \text{and since } L = 2 r / n$$

$$\theta_2 = \arcsin [\sin(\theta_1) - (2 / n)]$$

We construct the grid by choosing an appropriate value for n (discussed in later sections) and a starting latitude θ_1 somewhere North of the feature. Then we calculate successive latitude boundaries using the formula above.

As an example, let's choose $n = 10$ and start at the North pole, noting that the value of n should be an even integer to include the equator as a latitude boundary, if desired.

$$\text{lat}_1 = 90^\circ \text{ (north latitude)}$$

$$\text{lat}_2 = \arcsin [\sin(90^\circ) - 2/10] = 53.13^\circ$$

$$\text{lat}_3 = \arcsin [\sin(53.13^\circ) - 2/10] = 36.87^\circ$$

$$\text{lat}_4 = \arcsin [\sin(36.87^\circ) - 2/10] = 23.58^\circ$$

$$\text{lat}_5 = \arcsin [\sin(23.58^\circ) - 2/10] = 11.537^\circ$$

$$\text{lat}_6 = \arcsin [\sin(11.537^\circ) - 2/10] = 0^\circ \text{ (arriving exactly at the equator)}$$

Now, simply invert for southern hemisphere (south latitude):

$$\text{lat}_7 = 11.537^\circ, \text{lat}_8 = 23.58^\circ, \text{lat}_9 = 36.87^\circ, \text{lat}_{10} = 53.13^\circ, \text{lat}_{11} = 90^\circ$$

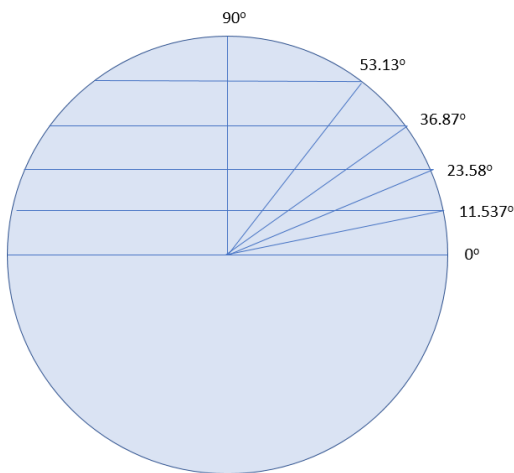


Figure 12. Latitude values for equal-area rings

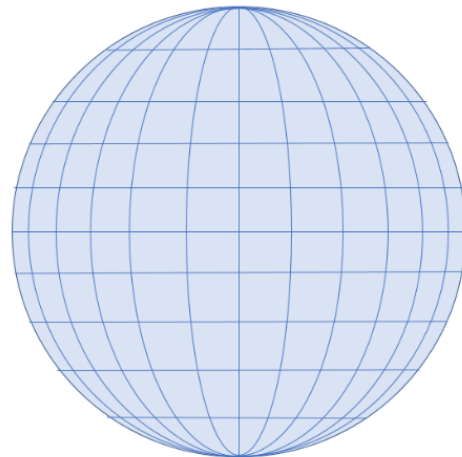


Figure 13. Resulting grid of equal-area cells

Automatic calculation of the latitude divisions is provided on the *Grid Construction* tab of the GONASA.xlsx spreadsheet.

2.3 The nature of normalized latitude lines

It's interesting to note that the latitude decrements are calculated by repeatedly subtracting a constant ($2/n$) which represents the opposite (vertical) side of the right triangle's latitude angle. Therefore, the calculated latitude increments will appear to be evenly spaced when viewed from a distance (Figure 14).

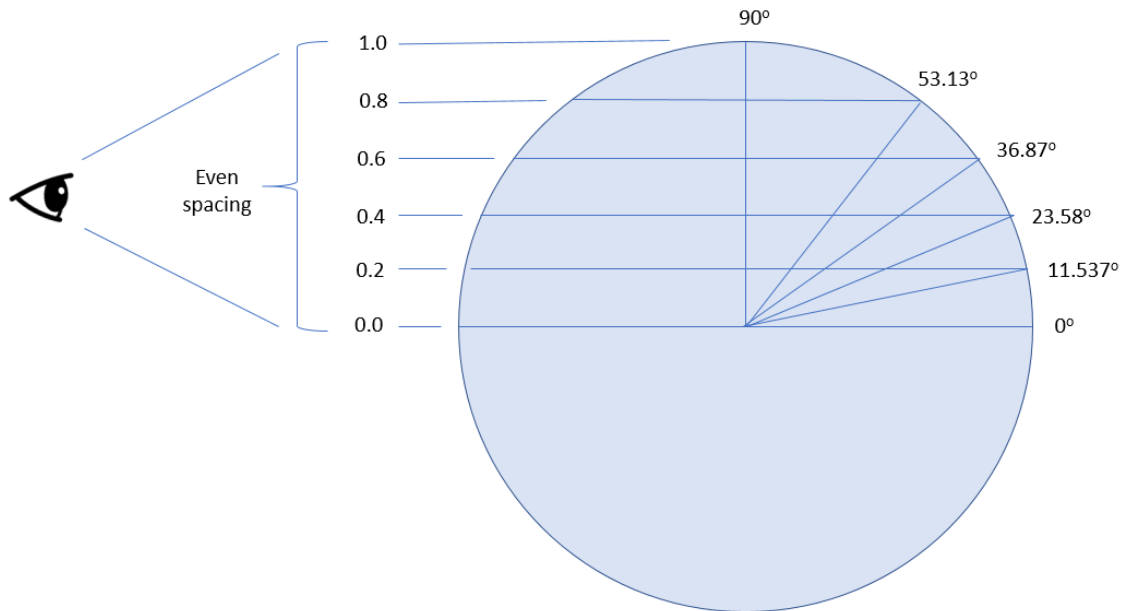


Figure 14. Equal-area rings appear evenly spaced from a distance

GONASA's constantly-adjusted latitude spacing simultaneously accounts for two phenomena:

- 1) Moving toward the poles, the increasing increments of latitude (which increase cell area) exactly compensate for the converging lines of longitude (which decrease cell area), leaving all cell areas equal.
- 2) The increasing latitude increments exactly compensate for the spherical foreshortening of latitude line spacing when viewed at a distance.

The latter effect of perceived even-spacing lends itself well to simple manual construction techniques. For example, when viewing a two-dimensional image of a sphere from a distance, horizontal latitude lines may be drawn at constant vertical pixel intervals to create rings of equal area.

When longitude and latitude are available for an image, more precise calculations are possible with the GONASA technique. Note that the GONASA grid cell boundaries are defined by unchanging longitude and latitude values. Thus the GONASA method continues to work well with different projections of a sphere's image. For instance, on a Mercator projection of the sphere the GONASA equal-area grid cells will appear as a well-ordered set of contiguous non-overlapping rectangles.

2.4 How to position square grid cells to specific latitudes

In general, overlaying a feature onto a mosaic of grid cells will incur less partial-cell error when the grid cells are roughly square. Also, the assumptions of the GONASA algorithms will be more accurate when approximately square cells have been positioned to the latitude of the feature. In particular, the centroid, moments of inertia, perimeter, compactness, bounding box and major axis algorithms assume square cell dimensions when taking the cell diameter as the square root of the cell area.

The construction technique of section 2.2 creates latitude separations that become increasingly larger towards the poles, to compensate for converging lines of longitude. Therefore, relatively speaking, the aspect ratio of the grid cells will be elongated vertically towards the poles and stretched horizontally nearer the equator. Somewhere in between the cells are approximately square, the width (w) and height (h) of a cell being equal as shown in Figure 15. The latitude of this occurrence can be manipulated by adjusting the number of longitude divisions (m) and latitude divisions (n).

Note that grid cells defined by longitude are never exactly square. The vertical edges of the cells are curved and not precisely straight. The width of their horizontal edges will be slightly less on the polar side than on the equator side. Multiplying cell width and height does not provide the exact cell area. Nonetheless, the cell area can be calculated precisely by dividing the total surface area of the visible hemisphere by the product of the longitude and latitude divisions:

$$\text{True cell area (identical for all cells)} = 2 \pi r^2 / (m * n)$$

To position approximately square cells to the latitude of the feature, we first need to establish the cell size, chosen to be consistent with the scale of the feature. The grid of cells “resolution” should be granular enough to fit irregularities along the feature’s perimeter. The cell size may be specified by the length of a side (e.g., 10km) or by the cell area (e.g., 100 km²). If the specification is by area, then the assumed side lengths of the approximately square cell will be the square root of the area.

Then, knowing the target latitude and the lengths of h and w , the necessary number of longitude divisions (m) can be calculated by observing that cell width decreases as the cosine of the latitude. Figure 16 shows two circles of latitude on the surface of a sphere, one at the equator and one at latitude α . The radius of the two circles is $r \cos(\alpha)$. Therefore the circumference of the two circles is: $2 \pi r \cos(\alpha)$.

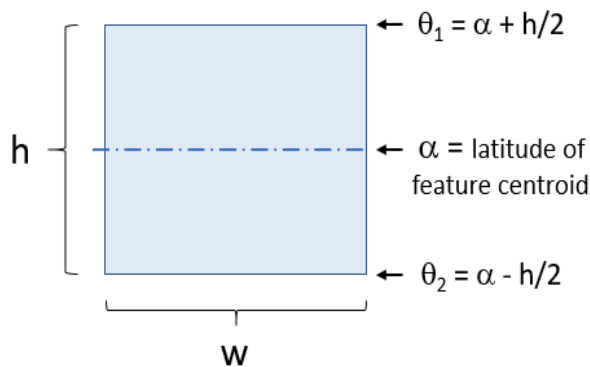


Figure 15. Cell dimensions at latitude α

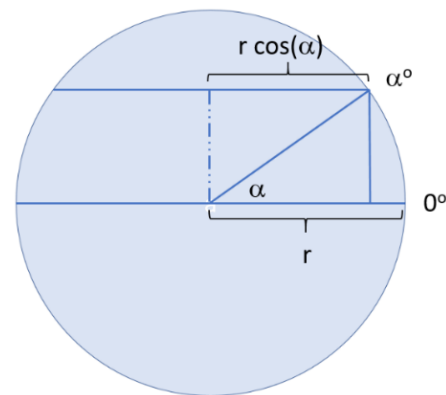


Figure 16. Cell width variation by latitude

Given m longitude divisions in the visible hemisphere ($2m$ in both hemispheres), the width of each cell at latitude α is:

$$w = 2 \pi r \cos(\alpha) / 2m$$

Rearranging, we have the desired number of longitude divisions to achieve width w at latitude α :

$$m = \pi r \cos(\alpha) / w$$

Next, to find the desired number of latitude divisions (n), we observe that the height of the cell is the arc between latitudes θ_1 and θ_2 shown in Figure 11. For convenience with the trig functions, we will express the cell height in terms of radians, by dividing h by the sphere radius r .

$$h/r = \theta_1 - \theta_2$$

By positioning the target latitude α at the center of the cell (see Figure 15), we have:

$$\theta_1 = \alpha + (\frac{1}{2})h/r \quad \theta_2 = \alpha - (\frac{1}{2})h/r$$

From section 2.2, we know how to find θ_2 given θ_1 and the latitude divisions n :

$$\theta_2 = \arcsin [\sin(\theta_1) - (2 / n)]$$

Then:

$$h/r = \theta_1 - \theta_2$$

$$h/r = (\alpha + h/2r) - \arcsin[\sin(\alpha + h/2r) - 2/n]$$

$$\alpha - h/2r = \arcsin[\sin(\alpha + h/2r) - 2/n]$$

$$\sin(\alpha - h/2r) = \sin(\alpha + h/2r) - 2/n$$

$$n = 2 / [\sin(\alpha + h/2r) - \sin(\alpha - h/2r)]$$

This result is consistent with the calculations of Section 2.1:

$$L / r = \sin(\theta_1) - \sin(\theta_2) \quad \text{and since } L = 2r/n$$

$$2/n = \sin(\theta_1) - \sin(\theta_2)$$

$$n = 2 / [\sin(\theta_1) - \sin(\theta_2)] \quad \text{where: } \theta_1 = \alpha + h/2r ; \theta_2 = \alpha - h/2r$$

Note that the number of divisions m and n may well be fractional to achieve a specific cell size. If the desired cell size can vary a little, m and n may be rounded to integers. Furthermore, n may be set to an even integer to include the equator as a latitude division, and m may be selected to simplify the longitude widths.

The next section offers advice for choosing suitable values of m and n .

2.5 How to choose the best grid dimensions

Choosing the ideal number of latitude and longitude divisions can involve a tradeoff among optimizing various objectives:

- An even number of latitude divisions to span the hemisphere and include the equator
- An integer number of longitude divisions to neatly divide the visible hemisphere West to East
- A number of longitude divisions to result in single decimal longitude widths (e.g., 0.7°, 1.2°, 3.8°)
- A particular combination of longitude to latitude divisions to position square cells to the feature's latitude
- A fractional number of divisions to yield very square cells or a precise area per cell

To facilitate the task, the *Setup Guide* tab of the GONASA.xlsx spreadsheet accepts inputs of the sphere radius, desired cell area, and latitude of the feature (as any decimal from 90° to -90°). The spreadsheet then calculates pairs of longitude/latitude divisions to prioritize various objectives. Users may select the recommendation that best suits their needs (Figure 17).

Setup Guide for Constructing the Desired Grid	
Enter values in green cells, and read division info from red cells, depending on priorities	
2,615 = Radius of spherical object (km)	0.038240918 = Height, width of square cell (radians)
10,000 = Desired equal area of cells (km ²)	58.6492 = Ideal latitude divisions for square cells at feature latitude:
26.9 = Latitude of the feature (+90 to -90)	73.2635 = Ideal longitude divisions for square cells at feature latitude:
42,965,835 = Surface area of visible hemisphere (km ²): $2 \pi r^2$	4,296.85 = # cells given these long/lat divisions
Prioritize square cells and even latitude divisions	Prioritize square cells and width x height cell area
58 = Even latitude divisions (so equator is included)	58.6492 = Integer latitude divisions
72.4603 = Longitude divisions (this hemisphere)	73.2635 = Longitude divisions (this hemisphere)
2.4841 = Longitude width (degrees)	2.4569 = Longitude width (degrees)
101.1085 = Cell height (km) at feature latitude	100.0000 = Cell height (km) at feature latitude
101.1085 = Cell width (km) at feature latitude	100.0000 = Cell width (km) at feature latitude
10222.9348 = Width x Height cell area (km ²)	10000.0000 = Width x Height cell area (km ²)
10223.3970 = True Cell Area (km ²)	9999.3907 = True Cell Area (km ²)
Prioritize true cell area and even latitude divisions	Prioritize integer longitude/latitude divisions
58 = Even latitude divisions (so equator is included)	59 = Integer latitude divisions
74.0790 = Longitude divisions (this hemisphere)	73 = Longitude divisions (this hemisphere)
2.4298 = Longitude width (degrees)	2.4658 = Longitude width (degrees)
101.1085 = Cell height (km) at feature latitude	99.4110 = Cell height (km) at feature latitude
98.8992 = Cell width (km) at feature latitude	100.3610 = Cell width (km) at feature latitude
9999.5479 = Width x Height cell area (km ²)	9976.9858 = Width x Height cell area (km ²)
10000.0000 = True Cell Area (km ²)	9975.8149 = True Cell Area (km ²)
Prioritize single decimal longitude widths and true cell area	Prioritize single decimal longitude widths and square cells
59.6748 = Latitude divisions (for true cell area)	57.6378 = Latitude divisions (for square cells)
72.0000 = Longitude divisions (for decimal longitude widths)	72.0000 = Longitude divisions (for decimal longitude widths)
2.4569 = Ideal longitude width (degrees)	2.4569 = Ideal longitude width (degrees)
2.5 = Constrained longitude width (degrees)	2.5 = Constrained longitude width (degrees)
98.2974 = Cell height (km) at feature latitude	101.7378 = Cell height (km) at feature latitude
101.7549 = Cell width (km) at feature latitude	101.7549 = Cell width (km) at feature latitude
10002.2413 = Width x Height cell area (km ²)	10352.3151 = Width x Height cell area (km ²)
10000.0000 = True Cell Area (km ²)	10353.4066 = True Cell Area (km ²)

Figure 17. Guide for choosing longitude/latitude divisions, depending on priorities

2.6 A note on the accompanying Excel spreadsheet

As this report was prepared for publication, it was noted that the accompanying Excel spreadsheet could be considered to be NASA-released software and would need to undergo a rigorous certification cycle. The singular purpose of the spreadsheet is to provide additional explanation and visualization of the algorithms described in this report. Therefore, in the interest of timely publication, the spreadsheet has been rendered read-only and inoperable as a software program. The suggested methods of algorithm implementation remain clearly visible. The authors apologize for any inconvenience.

2.7 A note on grid cell resolution

One of the user inputs to the Setup Guide described in section 2.5 is the desired area for each grid cell. Generally speaking, the smaller the grid cell size, the more closely cells can approximate the uneven contours of a surface feature. This assumes the cells are counted in a binary manner, as containing the feature or not. However, the problem with smaller cell size and greater resolution is having to manage and count many more cells.

One approach is to keep the cell size fairly large for convenience, but to estimate the coverage percentage of cells that are partially covered by a feature's perimeter. The coverage estimation may be performed either manually, by a human looking at the grid overlay on an image of the feature, or programmatically by dividing the pixel count of the segmented feature by the pixel count of the partially-covered cell.

This approach is used in the next section to calculate feature area.

3.0 GONASA Analysis Algorithms

3.1 Example Use Case

To better illustrate GONASA’s automated algorithms, an imaginary take on a real NASA project will be used. The Planetary Systems Lab is currently characterizing methane clouds on the surface of the Saturnian moon Titan from images captured by the Cassini mission. The radius of Titan is approximately 2575 km and cloud altitude is estimated at 40 km, so using a radius to the clouds of 2615 km yields a spherical surface area of **85,931,670 km²** as shown in Figure 18.

Note that in cases involving the analysis of features on the celestial sphere, the radius is infinite and area measures in units of distance² are not appropriate. Instead, we can specify the radius in terms of degrees as one radian = $180/\pi = 57.29577951^\circ$. The area of the celestial sphere is then **41,253 degrees²**.

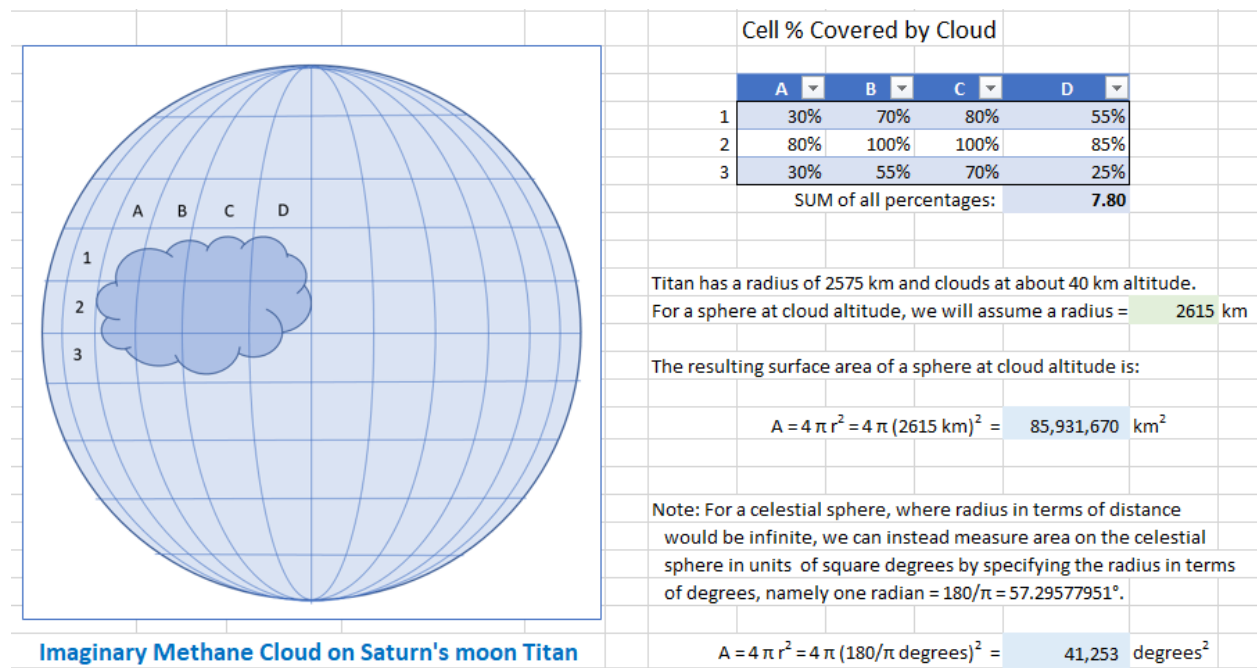


Figure 18. Calculating total surface area and loading the feature data.

A normalized grid of longitude and calculated latitude lines having equal area cells is prepared as described in the previous section and overlaid onto the two-dimensional image of the moon and its imaginary methane cloud. The percentage overlap of the cloud on each cell is loaded into a cartesian table. This is shown under the *Use Case* tab of the accompanying GONASA.xlsx Excel spreadsheet.

3.2 How to find the AREA of a surface feature

The first order of business is to measure area, which is the zeroth-order spatial moment of the feature (Wirth, 2004). Given the table of covered cell percentages from the previous section, those percentages are summed and multiplied by the area per cell to find the overall feature area. The area per cell is easily determined by dividing the total surface area of the sphere by the number of cells in the grid. (Figure 19)

Area		EAST LONGITUDE - Left Cell Edge							
NORTH LATITUDE		0	15	30	45	60	75	90	105
Top Cell Edge	Cell Center:	7.5	22.5	37.5	52.5	67.5	82.5	97.5	112.5
90	71.565								
53.13	45								
36.87	30.225								
23.58	17.5585			30%	70%	80%	55%		
11.537	5.7685			80%	100%	100%	85%		
0	-5.7685			30%	55%	70%	25%		
-11.537	-17.5585								
-23.58	-30.225								
Number of Latitude divisions:		10							
Number of Longitude divisions:		24							
adius of sphere at cloud altitude:		2615 km		Corresponding measurements on the celestial sphere					
Sphere surface area:		85,931,670 km ²		41,252.961		degrees ²			
Number of grid cells:		240		240					
Area per cell:		358,049 km ²		171.89		degrees ²			
Cells covered by cloud:		7.80 (cells)		7.80 (cells)					
Cloud area:		2,792,779 km ²		1,341		degrees ²			
% Sphere covered by cloud:		3.25%		3.25%					

Figure 19. Calculation of feature area

This example uses a grid of 10 rows by 24 columns, producing 240 cells, each of constant area = 358,049 km². The feature covers 7.8 cells and therefore has an area of **2,792,779 km²** which is 3.25% of the total surface area. A parallel calculation shows how an analogous feature on the celestial sphere would cover 1,341 degrees².

This example is shown under the *Area* tab of the GONASA.xlsx spreadsheet.

3.3 How to find the CENTROID of a feature

Next we measure the first-order spatial moment of the feature, normalized by the area calculated above. This is the center of mass, or centroid of the feature (Wirth, 2004). Because the sphere's surface is sliced and diced neatly into contiguous quadrilateral cells, we may continue to employ the rows and columns structure of an Excel spreadsheet to calculate the centroid in terms of longitude and latitude.

Mathematically, we are finding the first moments of area about the X-axis (Q_x) and Y-axis (Q_y) as:

$$Q_x = \text{SUM}(\text{cell latitude centers} * dA) \quad Q_y = \text{SUM}(\text{cell longitude centers} * dA)$$

where dA is the cell area coverage (treating a full single cell area as unit 1 for convenience)

The centroid coordinates are then: Longitude = Q_y/A Latitude = Q_x/A (where A = # Total area cells)

More intuitively, the feature's percentage coverage of each cell can be thought of as a weighted vote for the latitude and longitude coordinate at the cell's center. Summing the latitude and longitude voting separately and then dividing by the total cell coverage yields the average weighted longitude and latitude, which define the centroid of the feature's area (Figure 20).

Centroid		EAST LONGITUDE - Left Cell Edge								
NORTH LATITUDE		0	15	30	45	60	75	90	105	
Top Cell Edge	Cell Center:	7.5	22.5	37.5	52.5	67.5	82.5	97.5	112.5	SumProduct:
90	71.565									0
53.13	45									0
36.87	30.225									0
23.58	17.5585			30%	70%	80%	55%			147.375
11.537	5.7685			80%	100%*	100%	85%			220.125
0	-5.7685			30%	55%	70%	25%			108
-11.537	-17.5585									0
-23.58	-30.225									0
	SumProduct:	0	0	8.1518	14.88678	15.77735	13.11828	0	0	
First moment of area about X (Q_x):	51.934	(Sum of the SumProduct in Row 14)							$Q_x = \sum y dA$	
First moment of area about Y (Q_y):	475.50	(Sum of the SumProduct in Column L)							$Q_y = \sum x dA$	
Cell coverage:	7.8	(cells)								
Average weighted Longitude:	60.962	(degrees East)				Longitude center = Q_y/A				
Average weighted Latitude:	6.658	(degrees North)				Latitude center = Q_x/A				

Figure 20. Calculation of feature center of mass (centroid)

This example is shown under the *Centroid* tab of the GONASA.xlsx spreadsheet. The centroid of the cloud feature is located at **6.66° North Latitude** and **60.96° East Longitude** (where 0° longitude was arbitrarily set at the left edge of the visible hemisphere for this example).

Note that this technique accommodates features with irregular perimeters, and even multiple features whose overall centroid needs to be determined. Further, if the feature contains holes or even varying densities, this technique will continue to allow calculation of a center of mass.

3.4 How to find the MOMENTS OF INERTIA of a feature

Next we measure the second-order spatial moment of the feature, normalized by the centroid calculated above. This is the Area Moment of Inertia and describes how the area of the feature is dispersed about the centroid in the X and Y directions.

We use an approach similar to that taken to find the Centroid. We are finding the second moments of area about an X-axis (I_x) and Y-axis (I_y) that pass through the centroid:

$$I_x = \sum (\text{Cell latitude} - \text{Centroid latitude})^2 * dA$$

$$I_y = \sum (\text{Cell longitude} - \text{Centroid longitude})^2 * dA$$

where dA is the full or partial area of each cell

The desired units for the moments of inertia will be kilometers to the fourth power (km^4). To simplify the transition from coordinates to meters, we will assume that square cells have been positioned to the feature's latitude and thus each cell's diameter in both the X and Y directions may be taken as the square root of the constant cell area.

We give each cell center a row and column index representing the center of the cell, and rewrite the centroid longitude and latitude in terms of the cell indices. As shown on the *Moment of Inertia* tab of the GONASA.xlsx spreadsheet, the centroid latitude and longitude of 60.962° N, 9.925° E equates to a cell index of (column 7.564, row 9.925).

The equation for the second moment of area about X may be written as:

$$\begin{aligned} I_x &= \sum (\text{Cell center vertical distance from Centroid})^2 * dA \\ &= \sum ((\text{Cell row} - \text{Centroid row}) * \text{Cell diameter})^2 * \text{Cell area} * \% \text{ Cell coverage} \\ &= \sum (\text{Cell row} - 9.925)^2 * \% \text{ Cell coverage} * (\text{Cell diameter})^2 * \text{Cell area} \end{aligned}$$

Since Cell diameter and cell area are constant across all cells, and $(\text{Cell diameter})^2 = \text{Cell area}$, we can bring $(\text{Cell Area})^2$ outside the summation:

$$I_x = (\text{Cell area})^2 * \sum [(\text{Cell row} - 9.925)^2 * \% \text{ Cell coverage}]$$

The equation for the second moment of area about Y is derived similarly as:

$$I_y = (\text{Cell area})^2 * \sum [(\text{Cell column} - 7.564)^2 * \% \text{ Cell coverage}]$$

Values for $I_x = 5.27\text{E}+11 \text{ km}^4$ and $I_y = 1.03\text{E}+12 \text{ km}^4$ are calculated in Figure 21.

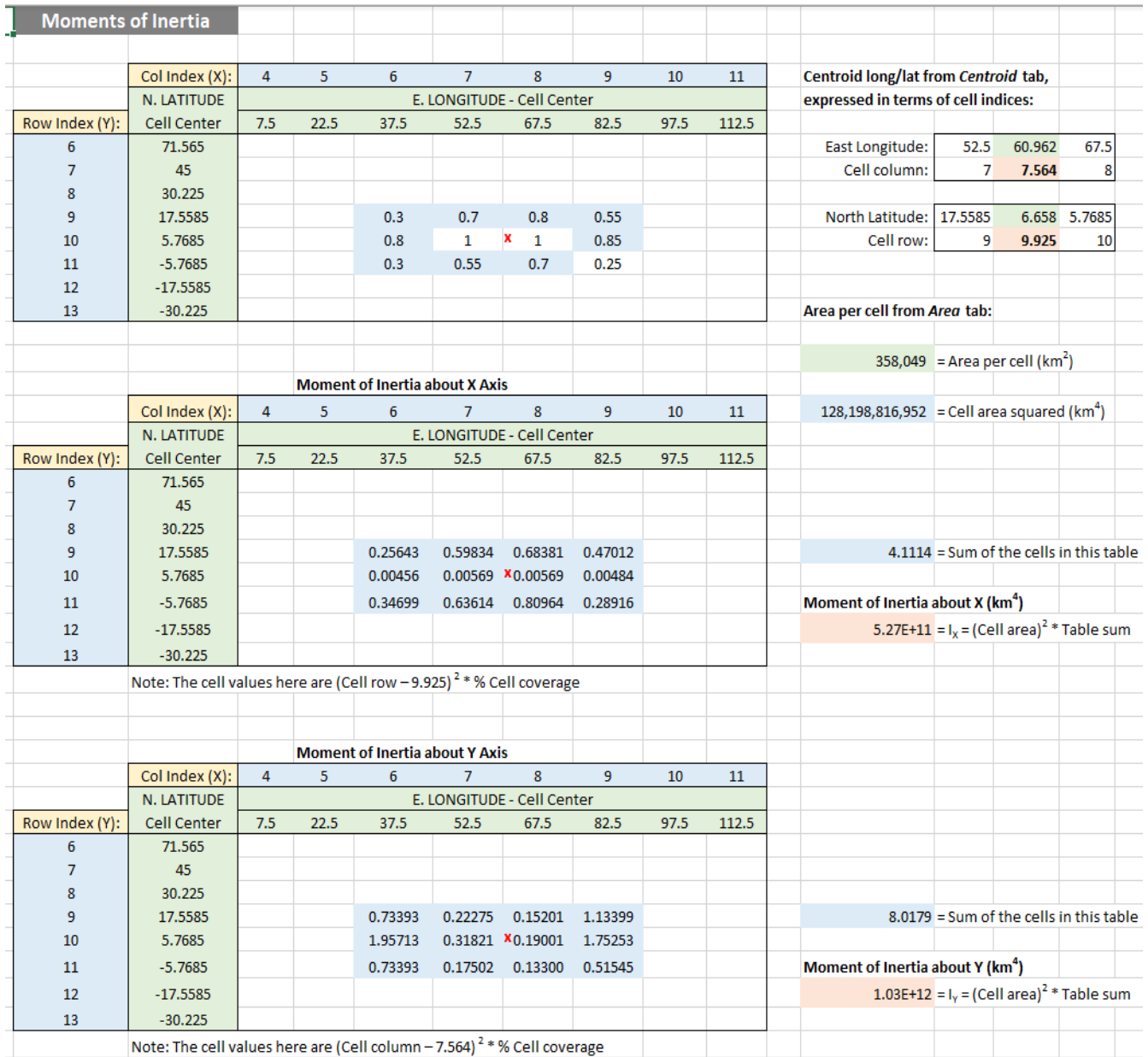


Figure 21. Calculation of feature moments of inertia.

3.5 How to find the PERIMETER of a feature

Once the cloud feature is loaded into the spreadsheet table, we have a visualization of the feature on equal-area cells, including the feature's perimeter. In a few steps, we can isolate the perimeter and count the externally-facing sides of the perimeter cells. Taking the diameter of the cells to be the unchanging square root of the cell areas, we can multiply by the external cell side count to determine the perimeter distance.

Since vertical and horizontal cell sides are assumed to have equal distance, the technique of Section 2.4 should first be used to position square cells to the latitude of the feature being analyzed.

Note that the detected perimeter is formed by a continuous boundary of cells that connect at either their faces or their corners. This is a so-called eight-connected boundary. Where boundary cells connect at their faces, the perimeter distance will be the diameter of the cells. Where boundary cells connect at their corners, we have a choice. The perimeter distance may be calculated either as the Manhattan distance of two cell diameters, or the Euclidean distance of $\sqrt{2}$ cell diameters (Figure 22). This is a choice between smoothing with the diagonal or using the staircase.

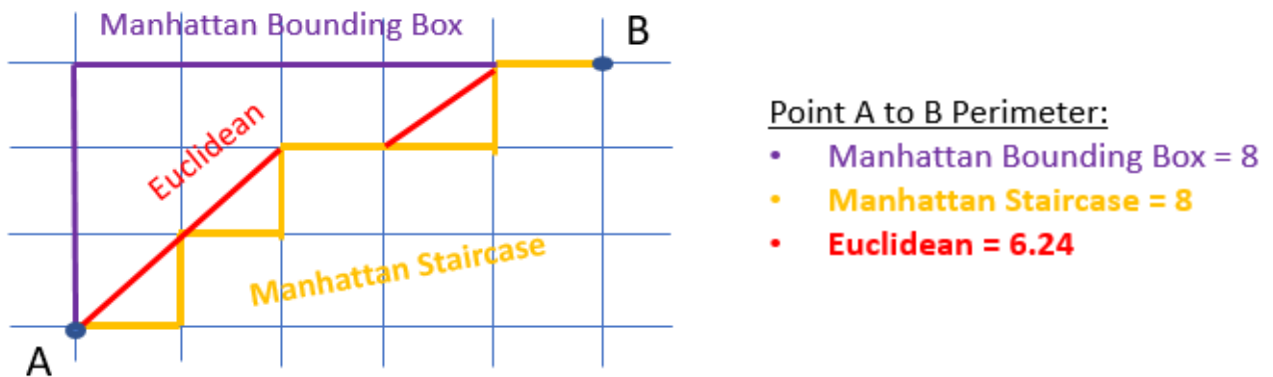


Figure 22. Manhattan vs. Euclidean distance

It will be noted that for the Manhattan distance method, the perimeter of a convex feature will be the same as the perimeter of the feature's bounding box, which would be a far simpler although less accurate way to estimate perimeter. On convex features, the accuracy of the Manhattan method does not improve with increasing cell resolution. Instead, we will use a Euclidean distance algorithm to obtain a more accurate perimeter estimate, and one which improves with increasing cell resolution.

The first step of the algorithm is to create a binary mask of the feature that codes a "0" for presence of the feature in a cell and "1" for cells where the feature is not present. A threshold may be considered for presence, such as requiring more than 25% occupancy of the cell.

The second step fills in any single interior cells that are not showing the feature's presence. This is done by changing any 1 to 0 when the cell is surrounded on four sides by cells that contain a 0. In other words, we fill in any one-cell holes. Note that we are only interested in the perimeter as the external edge of the feature. Internal edges from holes are a distraction.

The third step finds perimeter cells that are connected diagonally, and modifies the nearest exterior cell to contain a value less than one. A value of $\text{SQRT}(2)/2$ is recommended, such that when the two sides of the diagonally-connected cells are summed, their value will equal the Euclidean diagonal of $\text{SQRT}(2)$.

In the fourth step, each cell on the perimeter counts its externally-facing cells, including the discounted Euclidean value for diagonals. The cell-side total multiplied by the cell diameter yields the perimeter distance of the feature (Figure 23).

Perimeter		EAST LONGITUDE - Left Cell Edge								Perimeter =	13.41421	cells
NORTH LATITUDE	Cell Center:	0	15	30	45	60	75	90	105	Area per cell =	358,049	km ²
Top Cell Edge		7.5	22.5	37.5	52.5	67.5	82.5	97.5	112.5	Diameter per cell =	598.4	km
90	71.565									Cloud perimeter =	8027	km
53.13	45											
36.87	30.225											
23.58	17.5585				30%	70%	80%	55%				
11.537	5.7685				80%	100%	100%	85%				
0	-5.7685				30%	55%	70%	25%				
-11.537	-17.5585											
-23.58	-30.225											
1 - Reverse Binary Mask												
		1	1	1	1	1	1	1	1	Mark a cell = 0 if cloud occupation is above a certain threshold (say, 25%)		
		1	1	1	1	1	1	1	1	Mark unoccupied cells = 1		
		1	1	1	1	1	1	1	1			
		1	1	0	0	0	0	1	1	0.25 = Occupancy threshold		
		1	1	0	0	0	0	1	1			
		1	1	0	0	0	1	1	1			
		1	1	1	1	1	1	1	1			
		1	1	1	1	1	1	1	1			
2 - Fill Any Single-Cell Holes												
		1	1	1	1	1	1	1	1	If any cell is surrounded on four sides by occupied cells, mark it as occupied		
		1	1	1	1	1	1	1	1			
		1	1	1	1	1	1	1	1			
		1	1	0	0	0	0	1	1			
		1	1	0	0	0	0	1	1			
		1	1	0	0	0	1	1	1			
		1	1	1	1	1	1	1	1			
		1	1	1	1	1	1	1	1			
3 - Adjust cell-side values for any diagonal connections												
		1	1	1	1	1	1	1	1	If an exterior cell connects two occupied cells diagonally, use Euclidean distance		
		1	1	1	1	1	1	1	1	0.70711 = Euclidean distance value to use		
		1	1	1	1	1	1	1	1	Note: Corners of the four sides are checked individually (vs. checking for sum >=2)		
		1	1	0	0	0	0	1	1	to allow single cell intrusions into the occupied area		
		1	1	0	0	0	0.70711	1	1			
		1	1	1	1	1	1	1	1			
		1	1	1	1	1	1	1	1			
4 - Sum the exterior cell sides of the perimeter cells (and flip exterior to zeroes)												
		0	0	0	0	0	0	0	0	For each occupied cell, count the unoccupied cells on the four sides		
		0	0	0	0	0	0	0	0	For perimeter cells, the cell counts are the externally-facing sides		
		0	0	0	0	0	0	0	0	The internal sides of perimeter cells always face 0 cells, so those not counted		
		0	0	2	1	1	2	0	0	Internal cells face 0 cells on four sides, so their count = 0		
		0	0	1	0	0	1.70711	0	0			
		0	0	2	1	1.70711	0	0	0			
		0	0	0	0	0	0	0	0			
		0	0	0	0	0	0	0	0			
		0	0	0	0	0	0	0	0			
		0	0	0	0	0	0	0	0			
Perimeter =		13.4142 Cell diameters								Sum the counts of all cells		

Figure 23. Calculation of feature perimeter.

The perimeter calculation is shown under the *Perimeter* tab of the GONASA.xlsx spreadsheet. The perimeter is estimated as **8027 kilometers**.

3.6 How to find the COMPACTNESS of a feature

The compactness of a feature is a measure of how well its perimeter is minimized to contain maximum area. The most compact object then is a circle, and the measurement is normalized to a maximum value of 1 for a circle. Compactness is defined as the ratio of the area of the feature to the area of a circle having the same perimeter (Wirth, 2004).

Given a feature of perimeter P , we calculate the area of a circle with the same perimeter:

$$\text{Circle of circumference } P = 2 \pi r \text{ will have an area of } \pi r^2 = \pi (P/2 \pi)^2 = P^2 / 4 \pi$$

Then the compactness measure is defined, using Area (A) and Perimeter (P) determined in the previous sections:

$$\text{Feature Area / Circle Area} = A / (P^2 / 4 \pi) = 4 \pi A / P^2$$

Using $A = 2,792,779 \text{ km}^2$ and $P = 8,027 \text{ km}$, the compactness is directly calculated:

$$\text{Compactness} = 4 \pi (2,792,779 \text{ km}^2) / (8,027 \text{ km})^2 = \mathbf{0.545}$$

For comparison, a square will have a compactness of:

$$4 \pi A / P^2 = 4 \pi (P/4)^2 / P^2 = \pi / 4 = 0.785$$

The example feature is fairly compact, but its rectangular shape is more elongated and less compact than a square.

The compactness calculation is shown under the *Compactness* tab of the GONASA.xlsx spreadsheet.

3.7 How to find the X-Y BOUNDING BOX & ASPECT RATIO of a feature

The X-Y bounding box is the smallest longitude- and latitude-aligned rectangle that completely encloses a feature. The bounding box will be defined by the range of its longitude and latitude, as well as by the distance of its sides. The ratio of the bounding boxes horizontal width to its vertical height is the X:Y aspect ratio of the feature.

The first step in determining the bounding box is similar to finding the perimeter. A binary mask of the feature is coded with cell values of 1 where the feature is present and 0 where it is not. Again, a minimum threshold may be chosen for feature presence (in this example, 25%). See Figure 24.

NORTH LATITUDE Top Cell Edge	EAST LONGITUDE - Left Cell Edge								120
	0	15	30	45	60	75	90	105	
90									
53.13									
36.87									
23.58				30%	70%	80%	55%		
11.537				80%	100%	100%	85%		
0				30%	55%	70%	25%		
-11.537									
-23.58									
-36.87									

Mark a cell = 1 if cloud occupation is above a certain threshold (say, 25%)
 Mark unoccupied cells = 0

0.25 = Occupancy threshold

1 - Binary Mask									Row Sum:	Latitudes:
0	0	0	0	0	0	0	0	0	0	FALSE
0	0	0	0	0	0	0	0	0	0	FALSE
0	0	0	0	0	0	0	0	0	0	FALSE
0	0	1	1	1	1	0	0	0	4	23.58
0	0	1	1	1	1	0	0	0	4	11.537
0	0	1	1	1	0	0	0	0	3	0
0	0	0	0	0	0	0	0	0	0	-11.537
0	0	0	0	0	0	0	0	0	0	FALSE
Column Sum:	0	0	3	3	3	2	0	0		FALSE
Longitudes:	FALSE	FALSE	30	45	60	75	90	FALSE	FALSE	

Given area per cell = 358,049 km² Then diameter per cell = 598.4 km

		Left Longitude	Right Longitude	Width (cells)	Width (km)
Top Latitude	23.58	30	90	4	2,393
Bottom Latitude	-11.537				
Height (cells)	3				
Height (km)	1,795				
Aspect Ratio (X:Y)	4:3				

Figure 24. Calculation of X-Y bounding box and aspect ratio

The columns and rows of the mask are summed and non-zero sums indicate which columns and rows are occupied. Next a test is performed – for longitude row, to see if either the sum immediately above or to the left is zero; for the latitude column, to see if either the sum immediately to the left or above is zero. If so, then the longitude/latitude of that cell boundary are recorded.

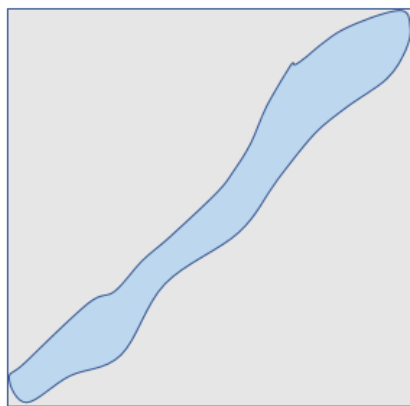
Finding the corners of the bounding box is then a simple matter of finding the Max and Min values within the recorded longitude and latitude range. The width and height of the bounding box are determined from the count of non-zero cells in the Column Sum row and the Row Sum column. Taking the constant diameter of the cells as the square root of the area, the width and height is obtained from multiplying that by the cell counts.

The width to height aspect ratio of the bounding box is the ratio of width and height cell counts

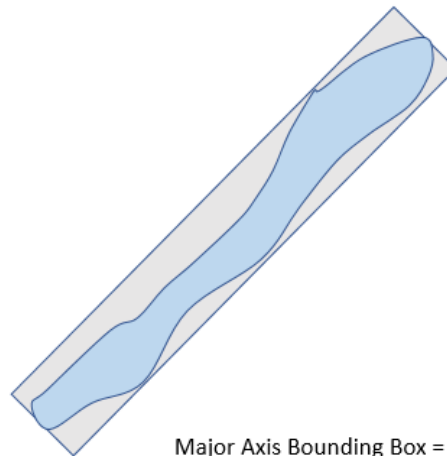
The bounding box is seen to range from **-11.537° to 23.58° North latitude**, and from **30° to 90° East longitude**. The bounding box is **2393 km wide by 1795 km high** and its X-Y aspect ratio is **4:3**.

The bounding box and aspect ratio calculations are shown under the *Bounding Box* tab of the GONASA.xlsx spreadsheet.

Although the X-Y bounding box and X-Y aspect ratio are fairly easy to calculate, they are misleading measures when a feature is elongated in a diagonal direction. For example, if a very thin and very long feature is oriented on a 45° angle, its X-Y bounding box will be square and the X-Y aspect ratio will be 1:1, when in actuality the thin and long feature has a pronounced aspect ratio along the 45° diagonal (Figure 25).



X-Y Bounding Box =1:1 Aspect Ratio



Major Axis Bounding Box =
8:1 Aspect Ratio

Figure 25. X-Y vs. Major Axis bounding boxes and their aspect ratios

In the next section we will find the major axis of the feature which will lead to a more accurate measure of aspect ratio.

3.8 How to find the MAJOR AXIS and ORIENTATION of a feature

Features that are stretched (for example, not square or circular) will tend to have a discernable orientation, for example West-to-East or Northwest-to-Southeast. A common technique for describing the orientation of stretched features is to identify the *major axis* of the feature. The major axis is the longest line segment that connects two points on the feature's perimeter (Figure 26). The minor axis is then the longest line segment perpendicular to the major axis that connects two perimeter cells.

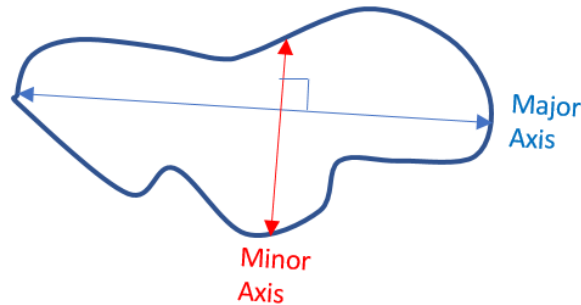


Figure 26. Major and minor axes of a feature.

The perimeter cells of the feature have already been determined in section 3.5. For convenience, the row and column indices are used to identify the perimeter cells, as was done for the Moments of Inertia calculations in section 3.5. Here however, the rows are numbered in decreasing order, which moves the origin to bottom left vs. Excel's placement of the origin at top left. This "first quadrant" coordinate system will be important to define the orientation of the feature as the slope of the major axis.

Assuming the cells are square, we find the two perimeter cells at greatest distance by arraying all $n(n-1)/2$ combinations of cells and taking the Euclidean distance of their cell indices. For the nine perimeter cells of this example, there are $9(9-1)/2 = 36$ unique pairs. The pair of cells with X,Y indices (2,2) and (5,4) are separated by the greatest Euclidean distance of $\text{SQRT}[(2-5)^2 + (2-4)^2] = 3.606$ cell diameters. Using a cell diameter of 598.4 km, the square root of the assumed square cell area, the length of the major axis is **2,157 km**.

Note that the major axis is measured between the *centers* of the most distant cells, and not their outer edges. This is a reasonable estimate because the perimeter cells might only be partially covered by the feature. If greater accuracy is needed, or if more than one pair of cells shares the greatest distance, then additional longitude/latitude divisions can be used to increase the cell count resolution.

The slope of the major axis is simply the difference of the cell pair y values divided by the difference of their x values = $(2-4)/(2-5) = 2/3$. An equation for the major axis may be written in the form of $y = mx + b$ where the slope $m = 2/3$ and the y-intercept $b = y - mx$ for either of the perimeter cell coordinates, for example $b = 4 - (5 * 2/3) = 2/3$. The major axis falls on the line with equation $y = (2/3)x + 2/3$.

The orientation of the feature is the arctangent of the slope = $\text{DEGREES}(\text{ATAN}(0.667)) = 33.7^\circ$. These calculations are shown in Figure 27 below and under the *Major Axis & Orientation* tab of the GONASA.xlsx spreadsheet.

Major Axis & Orientation		Perimeter Cells (Copied from Perimeter Tab)										Area per cell =	358,049	km ²	
Col Index (X):	0	1	2	3	4	5	6	7							
N. LATITUDE	E. LONGITUDE - Cell Center														
Row Index (Y):	Cell Center	7.5	22.5	37.5	52.5	67.5	82.5	97.5	112.5			Diameter per cell =	598.4	km	
7	71.565	0	0	0	0	0	0	0	0						
6	45	0	0	0	0	0	0	0	0						
5	30.225	0	0	0	0	0	0	0	0						
4	17.5585	0	0	2	1	1	2	0	0						
3	5.7685	0	0	1	0	0	1.70711	0	0						
2	-5.7685	0	0	2	1	1.70711	0	0	0						
1	-17.5585	0	0	0	0	0	0	0	0						
0	-30.225	0	0	0	0	0	0	0	0						
Distance between X/Y Indexed Perimeter Cells															
X, Y indices of perimeter cells		X (Column #):	2	2	2	3	3	4	4	5	5				
		Y (Row #):	4	3	2	4	2	4	2	4	3				
7	7	2	4												
6	6	2	3	1											
5	5	2	2	2	1										
4	4	3	4	1	1.41421	2.23607									
3	3	3	2	2.23607	1.41421	1	2								
2	2	4	4	2	2.23607	2.82843	1	2.2361							
1	1	4	2	2.82843	2.23607	2	2.23607	1	2						
0	0	5	4	3	3.16228	3.60555	2	2.8284	1	2.236068					
7	7	5	3	3.16228	3	3.16228	2.23607	2.2361	1.414214	1.414214	1				
6	6														
5	5	Largest separation distance: 3.60555 (cell diameters)				Length of Major Axis =						2,157	km		
4	4														
3	3														
2	2	Identification of perimer cell pair having largest separation distance													
1	1	X (Column #):	2	2	2	3	3	4	4	5	5				
0	0	X (Column #) Y (Row #):	4	3	2	4	2	4	2	4	3				
7	7	2	4												
6	6	2	3	FALSE											
5	5	2	2	FALSE	FALSE										
2	4	3	4	FALSE	FALSE	FALSE									
2	3	3	2	FALSE	FALSE	FALSE	FALSE								
2	2	4	4	FALSE	FALSE	FALSE	FALSE	FALSE							
1	1	4	2	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE						
0	0	5	4	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE				
7	7	5	3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE				
6	6														
5	5	More than one maximum distance pair? No (If yes, then greater cell resolution may be required)													
3	4														
3	3	Coordinates of Perimeter Cells on Major Axis:													
3	2		Cell 1	Cell 2											
1	1	X (Col #):	2	5	Major Axis Slope = $m = (y_2 - y_1) / (x_2 - x_1) =$										0.666667
0	0	Y (Row #):	2	4											
7	7	Longitude:	37.5	82.5	Major Axis y-intercept = $b = y_1 - (m * x_1) =$										0.667
6	6	Latitude:	-5.7685	17.5585											
5	5	Orientation (1st quadrant) =										33.7	Degrees		

Figure 27. Calculating the major axis and feature orientation.

3.9 How to find the MAJOR AXIS BOUNDING BOX and ASPECT RATIO of a feature

The goal at this point is to find the bounding box of smallest dimension that completely encloses the feature and then to take the width/height ratio of that bounding box as the most accurate measure of the feature's aspect ratio. The major axis, shown in dark blue in Figure 28, sets the width of the gray bounding box and its orientation. The top and bottom of the bounding box are set by the two line segments (in orange) that extend perpendicularly above and below the major axis and have the greatest distance to the perimeter.

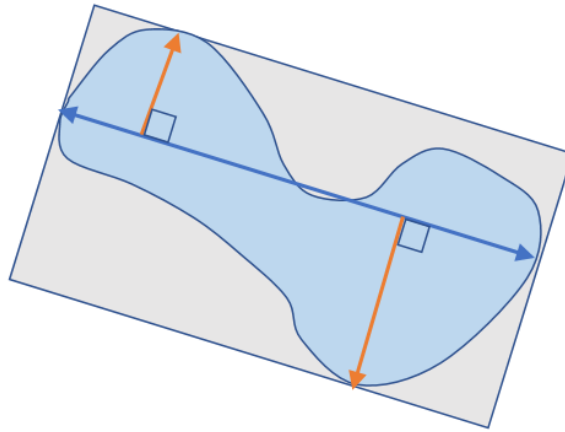


Figure 28. Using the major axis to define the feature's minimum bounding box.

The equation for the major axis was determined in the previous section as $y = (2/3)x + 2/3$. We will measure the distance from that line to each of the perimeter cells to find maxima above and below the line. First, let's derive an equation for the perpendicular distance from a point D to the major axis line (Figure 29).

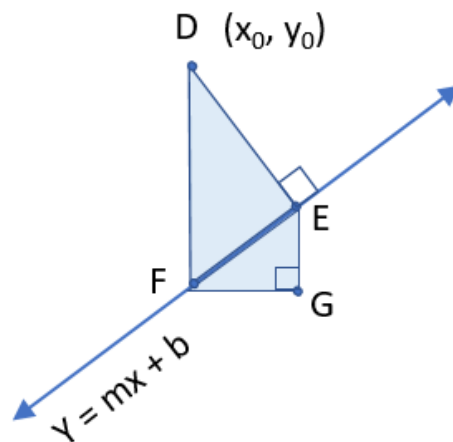


Figure 29. Calculating the perpendicular distance from point D to the major axis.

Referring to Figure 29, the distance DF has no X component and is simply $Y_D - Y_F$ where $Y_F = m * X_D + b$:

$$DF = y_0 - m x_0 - b$$

Distances EG / FG define the slope (m) of the line. Triangles DEF and FGE are seen to be similar triangles and so $EG / FG = EF / DE = m$ and thus:

$$EF = m * DE$$

By the Pythagorean theorem:

$$DF = \text{SQRT}[DE^2 + EF^2]$$

$$DF = \text{SQRT}[DE^2 + (m * DE)^2]$$

$$DF = \text{SQRT}[DE^2(1+m^2)]$$

$$DF = DE * \text{SQRT}[(1+m^2)]$$

Combining the equations for DF:

$$DE = DF / \text{SQRT}[(1 + m^2)]$$

$$DE = (y_0 - m x_0 - b) / \text{SQRT}[(1 + m^2)]$$

When this distance is calculated for all nine perimeter cells of the example, cell (2,4) has the maximum distance of 1.664 above the major axis, and cell (4,2) has the greatest distance of 1.1094 below the major axis. The sum of their distances is 2.7735. These cell index distances are converted to kilometers by multiplying by cell diameter of 598.4 km, resulting in a bounding box width of **2157 km** and height of **1660 km**.

The aspect ratio is the major axis width divided by the sum of the two perpendicular segment distances, or **1.3 : 1**.

A related measurement is *elongation* which is defined as the shortest dimension divided by the longest dimension. Elongation has a maximum value of 1 when the feature is circular or square. In this case, the elongation is the inverse of the aspect ratio, or **0.769**.

The calculation of the Major Axis bounding box dimensions and feature aspect ratio are shown in Figure 30 and under the *MA Box & Aspect Ratio* tab of the GONASA.xlsx spreadsheet.

Major Axis Bounding Box & Aspect Ratio

		Perimeter Cells (Copied from Perimeter Tab)								
		Col Index (X):	0	1	2	3	4	5	6	7
		N. LATITUDE	E. LONGITUDE - Cell Center							
Row Index (Y):	Cell Center	7.5	22.5	37.5	52.5	67.5	82.5	97.5	112.5	
7	71.565	0	0	0	0	0	0	0	0	0
6	45	0	0	0	0	0	0	0	0	0
5	30.225	0	0	0	0	0	0	0	0	0
4	17.5585	0	0	2	1	1	2	0	0	0
3	5.7685	0	0	1	0	0	1.70711	0	0	0
2	-5.7685	0	0	2	1	1.70711	0	0	0	0
1	-17.5585	0	0	0	0	0	0	0	0	0
0	-30.225	0	0	0	0	0	0	0	0	0

Major Axis cells (Copied from Major Axis Tab)

	Cell 1	Cell 2
X (Col #):	2	5
Y (Row #):	2	4

Major Axis Distance = 3.60555 (Cell Diameters)
(This is the denominator of the equation below)

Equation for major axis (from Major Axis tab)

slope (m)	y-intercept (b)
0.66667	0.66667

From the text, we have derived the perpendicular distance from a point (x_0, y_0) to the line $y = mx + b$ as:
Distance = $(y_0 - m x_0 - b) / \text{SQRT}([1 + m^2])$

Perpendicular Distance from each cell to the Major Axis line

X,Y indices of perimeter cells	Distance	X (Col #)	Y (Row #)
7	1.6641	2	4
6	0.83205	2	3
5	0	2	2
4	1.1094	3	4
3	-0.5547	3	2
2	0.5547	4	4
1	-1.1094	4	2
0	1.8E-16	5	4
7	-0.8321	5	3
6			
5	1.6641 = Furthest cell distance above Major Axis		
4	-1.1094 = Furthest cell distance below Major Axis		
3			
2			
1			
0			
7	3.60555 = Width (Major Axis distance)		
6	2.7735 = Height (Sum of two cell distances from MA)		
5	598.4 = Cell diameter (square root of cell area)		
2	4	2157 = Width (km)	
2	3	1660 = Height (km)	
2	2		
1			
0			
7			
6			
5			
3	4		
3			
3	2		
4			

Bounding Box Dimensions

- 3.60555 = Width (Major Axis distance)
- 2.7735 = Height (Sum of two cell distances from MA)
- 598.4 = Cell diameter (square root of cell area)
- 2157 = Width (km)
- 1660 = Height (km)

Aspect Ratio (based on Major Axis bounding box)

1.3 : 1 = Aspect Ratio of Feature

Elongation (inverse of aspect ratio)

0.769 = Height/Width (where circular or square = 1)

Perimeter Cells of greatest distance from the Major Axis line

	Cell 3	Cell 4
X (Col #):	2	4
Y (Row #):	4	2

Bounding Box Side Equations (y = mx + b)

	m	b
Top	0.667	2.667
Left	-1.500	5.000
Right	-1.500	11.500
Bottom	0.667	-0.667

Bounding Box Corners (Cell Indices)

	x	y
Top-Left	1.077	3.385
Top-Right	4.077	5.385
Bottom-Right	5.615	3.077
Bottom-Left	2.615	1.077

Bounding Box Corners (Long/Lat)

	E. Longitude	N. Latitude
Top-Left	23.65	10.30
Top-Right	68.65	35.91
Bottom-Right	91.73	6.68
Bottom-Left	46.73	-16.65

Figure 30. Calculation of major axis bounding box and feature aspect ratio.

In section 3.7, it was sufficient to define the X-Y bounding box by the longitude and latitude lines of the bounding box's sides. When the bounding box is aligned to the major axis and is potentially rotated away from the X-Y axis, its location needs to be defined by the latitude/longitude coordinates of the four corners. Those corners are derived as follows.

In general, we'll write equations for each of the four sides of the bounding box and then solve for their intersection, starting with cell indices and then converting to longitude and latitude. The equations for the top and bottom sides, which are parallel to the major axis, will have the same slope m but a different y-intercept b that places the parallel line through cells 3 and 4.

$$\text{Top cell 3} = (2,4): \quad y = mx + b \Rightarrow 4 = (2/3)*2 + b \Rightarrow b_3 = 8/3$$

$$\text{Bottom cell 4} = (4,2): \quad y = mx + b \Rightarrow 2 = (2/3)*4 + b \Rightarrow b_4 = -2/3$$

The left and right sides of the bounding box, being perpendicular to the major axis, will have the negative reciprocal of the major axis slope, or $m = -1.5$.

$$\text{Left cell 1} = (2,2): \quad y = mx + b \Rightarrow 2 = (-1.5)*2 + b \Rightarrow b_1 = 5$$

$$\text{Right cell 2} = (5,4): \quad y = mx + b \Rightarrow 4 = (-1.5)*5 + b \Rightarrow b_2 = 11.5$$

We now have the four equations for the sides of the bounding box:

$$\begin{array}{lll} \text{Top: } & y = (2/3)x + 8/3 & \\ \text{Left: } & y = -1.5x + 5 & \text{Right: } y = -1.5x + 11.5 \\ & \text{Bottom: } y = (2/3)x - 2/3 & \end{array}$$

The intersection of each adjacent pair of equations will give us the corners of the bounding box. The intersection is found by setting $m_1x + b_1 = m_2x + b_2$, solving for x , and then plugging x to solve for y . In general terms, the intersection occurs at:

$$x_0 = (b_1 - b_2) / (m_2 - m_1) \quad y_0 = (m_2 b_1 - m_1 b_2) / (m_2 - m_1)$$

These calculations are performed in Figure 29 above. Once the cell indices are obtained, we can use the longitude/latitude of the cell centers to interpolate coordinate values. The resulting corners of the bounding box are:

Top-Left: **10.30° N. Lat, 23.65° E. Long**
 Top-Right: **35.91° N. Lat, 68.65° E. Long**
 Bottom-Right: **6.68° N. Lat, 91.73° E. Long**
 Bottom-Left: **-16.65° N. Lat, 46.73° E. Long**

All calculations are shown on the *MA Box & Aspect Ratio* tab of the GONASA.xlsx spreadsheet.

4.0 Use Case – Methane Clouds on Titan

4.1 Motivation

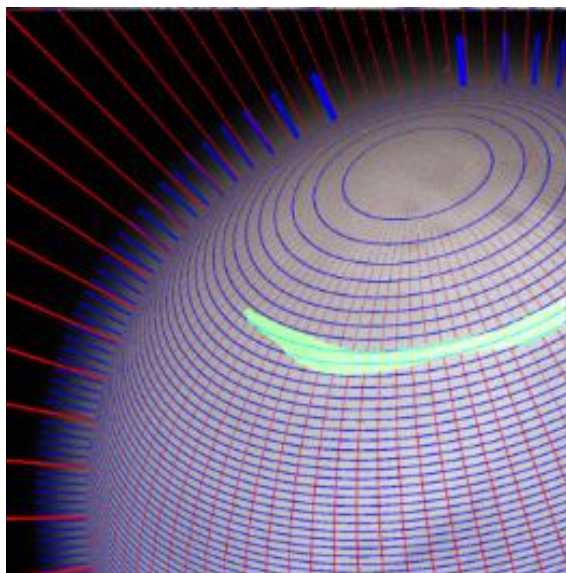
Saturn's moon Titan is the only other body in the solar system that has liquid lakes, rivers and clouds and an evaporation/precipitation cycle like Earth's (Lorenz, 1993). On Titan, the liquid and vapor are methane, but the cycle is entirely analogous to Earth's water cycle. Studying the seasonal and long-term weather patterns of Titan's methane clouds will inform our understanding of atmospheric mechanisms on other solar system worlds as well as our own (Mitchell, 2011).

4.2 Automated Cloud Analysis

Whereas previous analysis of Titan's methane clouds has relied upon manual detection and characterization by humans (Griffith, 2009), the Planetary Systems Laboratory (PSL) at NASA's Goddard Spaceflight Center is automating this analysis using machine learning to automatically detect methane clouds in images of Titan's atmosphere. The images analyzed were taken by the Cassini spacecraft mission from June 2004 to October 2017. In addition to automatically detecting image pixels where methane clouds exist, the PSL is interested in automating the calculation of cloud area, centroid, perimeter, orientation and aspect ratio. The GONASA algorithms of this paper were developed specifically to accomplish this task.

4.3 GONASA Application

The GONASA algorithms described here have been coded into Python routines and executed against images of Titan taken by the Cassini mission. To match the size and shape of methane clouds detected in Titan's northern latitudes, a granularity of 40 longitudinal divisions of the visible hemisphere (4.5° each) and 90 latitudinal divisions was chosen for the grid. In the image below, that grid is superimposed on Titan with the detected methane cloud highlighted in green (Figure 31).



A cloud area of **558,556 km²** was calculated by counting the number of cells covered by the cloud.

A cloud centroid of **51.05° North, 93.18° East** was calculated by finding the weighted center of mass of the covered cells.

A cloud perimeter of **4423 km** was measured by counting the outside-facing sides of the perimeter cells, which corresponds to a compactness ratio of **0.359**.

A major axis of **1591 km** length was detected, with an orientation of **-15.9 degrees**.

A minimum bounding box aligned to the major axis revealed a cloud width of **1591km** and height of **390 km**, for an aspect ratio of **4.077 : 1**.

Figure 31. GONASA grid on Titan (Goddard PSL)

5.0 Summary

This paper has described a novel approach for the parameterization of features on spherical surfaces, such as planets and moons. The GONASA technique is both iso-latitudinal and iso-longitudinal, creating a cartesian grid of equal-area cells with fixed longitude and latitude boundaries. The technique is compared to other approaches in the literature and is seen to have advantages in simplicity of construction and use, flexibility of resolution and square cell positioning, and integration with standard coordinate systems of longitude/latitude and right ascension/declination.

The GONASA toolkit includes algorithms to calculate feature area, centroid, moments of inertia, perimeter, compactness, bounding boxes, aspect ratio and orientation. The algorithms are carried out in the accompanying GONASA.xlsx spreadsheet for clarity of implementation and as evidence of their suitability for automation.

Finally, the practical value of the GONASA method is demonstrated with the real use case of characterizing methane clouds in the atmosphere of Titan.

6.0 References

- Archinal, B., *et al* (2018). Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements: 2015. *Celest Mech Dyn Astr* (2018) 130:22.
<https://doi.org/10.1007/s10569-017-9805-5>
- Beckers, B., & Beckers, P. (2012). A general rule for disk and hemisphere partition into equal-area cells. *Computational Geometry*, 45(7), 275–283.
<https://doi.org/10.1016/j.comgeo.2012.01.011>
- Beyer, W. H. (1987). CRC Standard Mathematical Tables, 28th ed. Boca Raton, FL: CRC Press, p. 227.
- Britannica, T. Editors of Encyclopaedia (2007). Cylindrical projection. Encyclopedia Britannica.
<https://www.britannica.com/science/cylindrical-projection>
- Bunbury, E.H. (1883). A History of Ancient Geography among the Greeks and Romans from the Earliest Ages till the Fall of the Roman Empire, page 667. London: John Murray.
- Capitaine, N. (1990). The celestial pole coordinates. *Celestial Mech Dyn Astr* 48, 127–143.
<https://doi.org/10.1007/BF00049510>
- Chan, F. K. (1980). A quadrilateralized spherical cube Earth Data Base - NASA Technical Reports Server (NTRS). NASA. *Goddard Space Flight Center Fifth Ann. Flight Mech. Estimation Theory Symp.* <https://ntrs.nasa.gov/citations/19810002572>
- Essinger-Hileman, T. (1989). *Lambda - Cosmic background explorer*. NASA. Retrieved October 30, 2022, from https://lambda.gsfc.nasa.gov/product/cobe/skymap_info_new.html
- Gall, J. (1885). Use of cylindrical projections for geographical astronomical, and scientific purposes. *Scottish Geographical Magazine*, 1(4), 119–123.
<https://doi.org/10.1080/14702548508553829>
- Gorski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., & Bartelmann, M. (2005). HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, 622(2), 759–771.
<https://doi.org/10.1086/427976>
- Griffith, C. A., *et al* (2009). Characterization of Clouds in Titan’s Tropical Atmosphere. *Astrophysical Journal Letters*, 702, L105–L109.
<https://doi.org/10.1088/0004-637X/702/2/L105>

- Gringorten, I. I., & Yepez, P. J. (1992). The Division of a Circle or Spherical Surface into Equal-area Cells or Pixels, Tech. Rep. 92-27740, Phillips Laboratory, Hanscom AFB, MA, <https://apps.dtic.mil/dtic/>
- Lorenz, R. D. (1993). The life, death and afterlife of a raindrop on titan. *Planetary and Space Science*, 41(9), 647–655. <https://doi.org/10.1016/0032-0633>
- Mitchell, J., Ádámkóvics, M., Caballero, R. et al (2011). Locally enhanced precipitation organized by planetary-scale waves on Titan. *Nature Geosci* 4, 589–592. <https://doi.org/10.1038/ngeo1219>
- Malkin, Z. (2019). A new equal-area isolatitudinal grid on a spherical surface. *The Astronomical Journal*, 158(4), 158. <https://doi.org/10.3847/1538-3881/ab3a44>
- Polyanin, Andrei D; Manzhirov, Alexander V. (2006), Handbook of Mathematics for Engineers and Scientists, CRC Press, p. 69, ISBN 9781584885023.
- Roşca, D. (2010). New uniform grids on the sphere. *Astronomy and Astrophysics*, 520. <https://doi.org/10.1051/0004-6361/201015278>
- Snyder, John P (1993). Flattening the Earth: Two Thousand Years of Map Projections. University of Chicago Press. p. 48. ISBN 0-226-76747-7.
- Weisstein, Eric W. (2020). Lambert Cylindrical Equal-Area Projection. From MathWorld--A Wolfram Web Resource. <https://mathworld.wolfram.com/LambertCylindricalEqual-AreaProjection.html>
- White, D., Kimerling, J. A., & Overton, S. W. (1992). Cartographic and geometric components of a global sampling design for environmental monitoring. *Cartography and Geographic Information Systems*, 19(1), 5–22. <https://doi.org/10.1559/152304092783786636>
- Wirth, M. (2004). Shape Analysis & Measurement. Retrieved November 2, 2022, from <http://www.cyto.purdue.edu/cdroms/micro2/content/education/wirth10.pdf>