

Technical Note

Not peer-reviewed version

Beyond Plane Sailing: Solving the Range-Doppler Equations in a Reduced Geometry

[Tom Grydeland](#)^{*} and Yngvar Larsen

Posted Date: 8 December 2025

doi: 10.20944/preprints202512.0734.v1

Keywords: synthetic aperture radar; satellite orbits; geodesy



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Technical Note

Beyond Plane Sailing: Solving the Range-Doppler Equations in a Reduced Geometry

Tom Grydeland * and Yngvar Larsen 

NORCE Research AS, Norway

* Correspondence: tgry@norce-research.no

Abstract

An important part of geometry computations for synthetic aperture radar (SAR) carried on a satellite platform is to convert between the instrument-specific coordinate system and a geocentric coordinate system such as Cartesian Earth-centered, Earth-fixed (ECEF) coordinates, geodetic coordinates (latitude/longitude) or a projection of these. The solutions for points on or near ellipsoid height typically involves iteration over geodetic coordinates, which means performing the transformation from geodetic to ECEF and its 6 partial derivatives in every iteration step. We present a method for solving these equations in the satellite's zero Doppler plane, which is typically used as coordinate plane for SAR systems with small squint angles. Solving the system in this plane means one of the constraints is satisfied implicitly, and allows solution which satisfies the other constraint (correct range from satellite) to be solved using a one-dimensional Newton method. The method is simple to implement, fast and accurate. For targets on the ellipsoid, the solution can be made as accurate as machine precision allows. For high-precision applications with targets at non-zero height above the ellipsoid, a small correction step is necessary, and we describe how to arrive at this step.

Keywords: synthetic aperture radar; satellite orbits; geodesy

1. Background and Introduction

For a synthetic aperture radar (SAR) carried on a platform in motion, the natural coordinate axes for focused data are *range* (fast time: two-way pulse propagation time between instrument and target) and *azimuth* (slow time: a coordinate which gives the instrument's position while pulsing). These two coordinates in isolation describe a sphere centered on the satellite. When data are focused (energy from a single target is concentrated to a point), we choose for a monostatic instrument with a small squint angle to always focus data to the *zero-Doppler plane* i.e. every target is focused at its point of shortest range. With this restriction, the pair of SAR coordinates describe a circle in the chosen plane.

For almost every other use, an object on or near the Earth's surface will be placed in a coordinate system related to the Earth surface: Geodetic latitude and longitude, or any projection of these. The geodetic coordinates are related to a reference ellipsoid that approximates the Earth's shape to a first degree, and a third coordinate is then the height relative to this ellipsoid. A further refinement of the Earth's shape is the *geoid* which describes a surface everywhere normal to the local direction of gravity and coinciding with the mean sea level, and digital elevation models (DEMs) typically give height relative to the geoid. In this note we will only use height as relative to the ellipsoid.

For a surface described by a given ellipsoid and height, the circle described by the SAR coordinates will intersect this surface in zero, one or two points. Zero or one if the range is smaller than or exactly equal to the satellite's height above the surface. The two points in the final case are solutions for a left- or right-looking instrument. The solutions on the far side of the Earth are mathematically valid, but not relevant here.

2. Transforming Between Cartesian and Geodetic Coordinates

Let r_e, r_p = semimajor/semiminor axis of Earth's ellipsoid; define geodetic latitude and longitude φ and λ ; ellipticity $e^2 = (r_e^2 - r_p^2)/r_e^2$, and second ellipticity $e'^2 = (r_e^2 - r_p^2)/r_p^2$.

$$x = (\eta + h) \cos \varphi \cos \lambda \quad (1)$$

$$y = (\eta + h) \cos \varphi \sin \lambda \quad (2)$$

$$z = (\eta(1 - e^2) + h) \sin \varphi \quad (3)$$

where $\eta = r_e / \sqrt{1 - e^2 \sin^2 \varphi}$ is the local radius of curvature of the ellipsoid in the meridional plane.

For the inverse relation, longitude is simple ($\tan \lambda = y/x$), but latitude is more involved. Closed-form solutions exist but don't lend themselves to numerical use. Many iteration methods have been described in the literature [1] differing in their speed of convergence and their use of transcendental functions (sines and cosines). We found that the following formula due to Bowring [2] gives sufficient accuracy for the ellipsoids of interest to us in a single iteration:

$$\tan \varphi = (z + e'^2 \cdot r_p \cdot \sin^3 \beta) / (\rho - e^2 \cdot r_e \cdot \cos^3 \beta) \quad (4)$$

where β is the reduced (or elliptic) latitude of the point, $\tan \beta \approx z \cdot r_e / \rho \cdot r_p$ (see Figure 1); and $\rho = \sqrt{x^2 + y^2}$ is the equatorial component of the position in the meridional plane, indicated by p in the figure. For a target on the ellipsoid, this formula gives latitude to within the representability of a double-precision floating-point number, 10^{-16} radians. For a target at an altitude of 9000 m, the latitude coordinate is still within a μm of the correct answer.

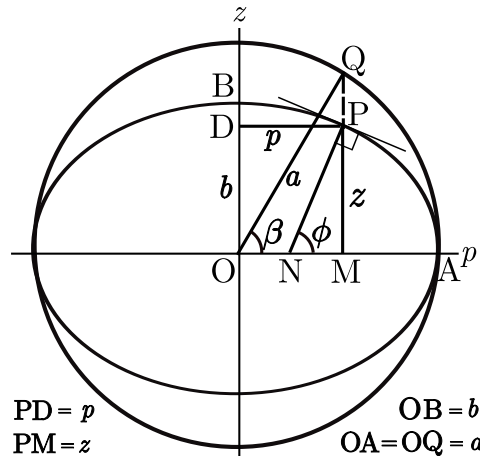


Figure 1. Figure showing the definition of geodetic latitude, φ , and reduced latitude, β , in the meridional plane. Illustration by Peter Mercator, placed in the public domain by the creator.

At satellite altitudes (1000 km) the error in latitude has increased to a few centimetres. If this is insufficient, a refined estimate of β can be obtained as

$$\tan \beta = r_p / r_e \tan \varphi \quad (5)$$

Using this β to compute a new value of φ gives answers that differ only in the last 1-2 bits of the numerical representation even at 1000 km.

Once geodetic latitude is known, height above the ellipsoid, when required, is given by

$$h = \rho \cdot \cos \varphi + z \cdot \sin \varphi - r_e^2 / \eta \quad (6)$$

This expression evaluated in double precision gives height accurate to a few nm for a target from 0 and up to 9000 m height.

The complete set of range-Doppler equations is then as follows:

$$\begin{aligned} \|\mathbf{P}_{\text{sat}} - \mathbf{P}_{\text{tar}}\| - r &= 0 \\ (\mathbf{P}_{\text{sat}} - \mathbf{P}_{\text{tar}}) \cdot \mathbf{V}_{\text{sat}} &= 0 \\ \frac{x^2}{r_e^2} + \frac{y^2}{r_e^2} + \frac{z^2}{r_p^2} &= 1 \end{aligned} \quad (7)$$

where r is range, \mathbf{P}_{sat} (\mathbf{V}_{sat}) is the satellite's position (velocity) at the given azimuth time, and $\mathbf{P}_{\text{tar}} = (x, y, z)$ is the solution (target position) being sought. The first equation states that the target must be at the correct range. The second equation states that the target must be in the correct plane. To use a different focus plane than the zero Doppler plane, replace the velocity vector \mathbf{V}_{sat} with a normal vector to the chosen plane. The final equation requires the target to be at ellipsoid height. For height not at the ellipsoid, it takes a more complex form, as can be understood from Equation (6), but the solution can be found in an approach where this equation will be satisfied by construction, as explained in the next section.

3. Transformation from SAR Coordinates to Geodetic lat/lon

The traditional way of finding geodetic coordinates for given SAR coordinates is a 2-D Newton-Raphson search, using the fact that for a target in the zero-Doppler plane, the vector describing the target's position relative to the satellite is perpendicular to the satellite's velocity vector:

$$(\mathbf{P}_{\text{sat}} - \mathbf{P}_{\text{tar}}) \cdot \mathbf{V}_{\text{sat}} = 0 \quad (8)$$

so the azimuth error is the value of this dot product. The error in range is the difference between the found and desired range. The Newton increment is then $-[dAz, dRg]^T \cdot [dX/d\Phi]^{-1}$ where $dX/d\Phi$ is the Jacobian for the transformation from Cartesian geocentric coordinates to geodetic lat/lon, described in Section 2, above.

This method converges to an accuracy of $1 \mu\text{m}$ in 5 or 6 iterations for a starting point within a hundred km or so. Such a starting point can be found by using the satellite's position e.g. at the midpoint of a SAR scene and adding an offset of a few hundred km to the left or right.

A better starting point can be obtained by taking the satellite position for the given azimuth time, and using an approximate Earth radius and height to compute a triangle, using the third vertex of this triangle as the start point for the iteration, as illustrated in Figure 2. In this way, the method usually converges in 2 or 3 iterations.

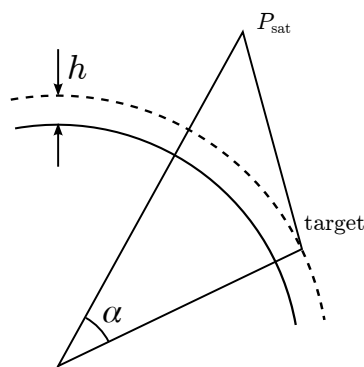


Figure 2. An improved initial guess for the target's position. The side lengths of the triangle are known: The satellite's position relative to the Earth centre is known, the range is specified, and the target's height is known to good accuracy with very rough knowledge of latitude. From these distances, the angle α can be computed. The heights of the satellite and target above the ellipsoid are exaggerated.

3.1. Solutions in the Plane

While the 2D Newton method gives an answer to the desired accuracy and converges reasonably fast, it suffers by having to do all computations in three dimensions. A recent paper by Klein [3] gave closed-form expressions for the ellipse that results when intersecting a plane with an ellipsoid. When the target we seek lies on the ellipsoid, all quantities can be restricted to this plane. We should point out that the axes of this ellipse are in general not aligned with the axes of the ellipsoid. Here, we will describe the method in more detail, and show the accuracy of the approach for targets at all heights and along an entire representative SAR satellite orbit from Sentinel-1A. We will also discuss computational efficiency in the final section.

Following Klein's description, we need a unit normal vector \mathbf{n} to the plane, for which we use the satellite's unit velocity vector. We also need a vector for an arbitrary point \mathbf{q} in the intersection plane, for which we choose the satellite's position. The unit vectors corresponding to the satellite's right ($\mathbf{A} \propto \mathbf{V}_{\text{sat}} \times \mathbf{P}_{\text{sat}}$) and up ($\mathbf{B} \propto \mathbf{A} \times \mathbf{V}_{\text{sat}}$) are rotated in the plane to form an orthonormal basis for the plane $\{\hat{\mathbf{a}}, \hat{\mathbf{b}}\}$ which also satisfies

$$(D_1 \hat{\mathbf{a}}, D_1 \hat{\mathbf{b}}) = 0 \quad (9)$$

where (\mathbf{u}, \mathbf{v}) is the ordinary inner product of vectors in \mathbf{R}^3 , and Klein's $D_1 = \text{diag}(1/a, 1/b, 1/c)$ and $D = \text{diag}(a, b, c)$ for the ellipsoid (used below) become $D_1 = \text{diag}(1/r_e, 1/r_e, 1/r_p)$ and $D = \text{diag}(r_e, r_e, r_p)$ in our case.

The distance between the plane and the origin is given by $\kappa = (\mathbf{q}, \mathbf{n})$, and we compute the centre of the ellipse by Klein's equation /40/:

$$\mathbf{m} = \frac{\kappa}{\kappa_t^2} D^2 \mathbf{n} \quad (10)$$

where

$$\kappa_t^2 = (D\mathbf{n}, D\mathbf{n}).$$

Using these results, and translating the coordinate system of the plane to place its origin at the centre of the intersection ellipse, the equation for the ellipse becomes

$$\frac{a^2}{A^2} + \frac{b^2}{B^2} = 1, \quad (11)$$

where the semiaxes are given by

$$A = \sqrt{\frac{1-d}{(D_1 \hat{\mathbf{a}}, D_1 \hat{\mathbf{a}})}} \quad \text{and} \quad B = \sqrt{\frac{1-d}{(D_1 \hat{\mathbf{b}}, D_1 \hat{\mathbf{b}})}}, \quad (12)$$

with d given by Klein's equation /25/ as $d = \kappa^2 / \kappa_t^2$.

A point with coordinates (a, b) in the translated plane have ECEF coordinates $\mathbf{m} + a \cdot \hat{\mathbf{a}} + b \cdot \hat{\mathbf{b}}$ in \mathbf{R}^3 .

The axes of the intersection ellipse are in general not aligned with the axes of the ellipsoid. In Figure 3 we present the parameters of the intersection ellipses along an entire orbit of the Sentinel-1 satellite. The ellipses have a semimajor axis very close to the equatorial radius of Earth (r_e). The semiminor axis varies between the equatorial and polar radii of Earth (panel 2). Hence, ellipticity and flattening of the intersection ellipse varies between those of the ellipsoid's meridional ellipse, and zero (panel 3). The ellipse centre's coordinates do not coincide with the centre of the ellipsoid, but the two are within a few km of each other (panel 4).

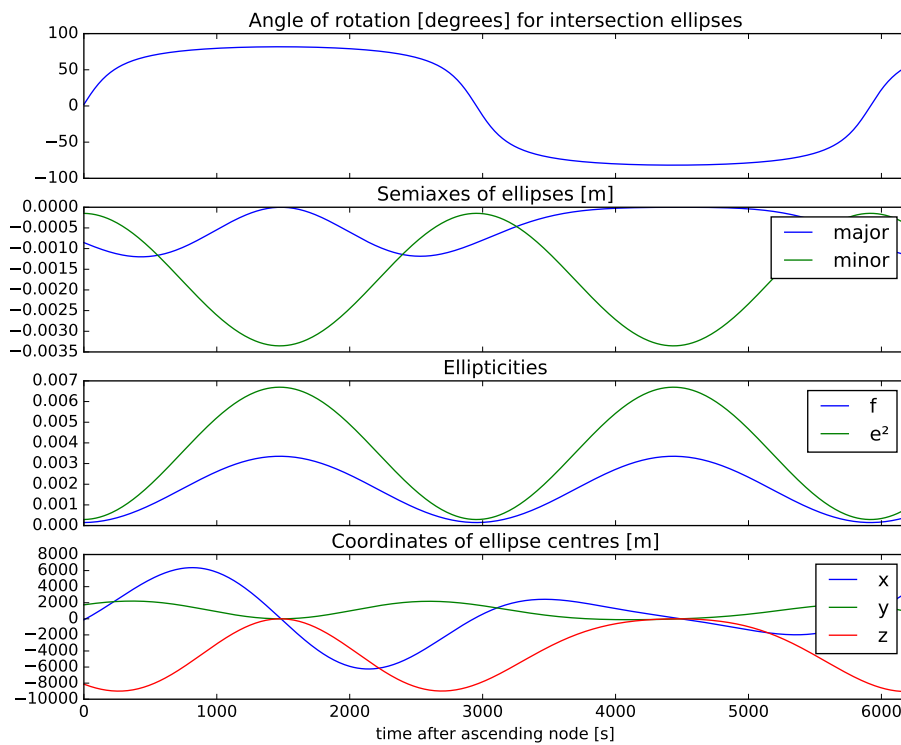


Figure 3. Parameters of the intersection ellipses for a full Sentinel-1 orbit. The x axis shows seconds since the ascending node in all panels. The first panel shows the angle through which the base had to be rotated to make the basis vectors line up with the axes of the ellipse. This figure mostly reflects the satellite's position in its orbit. The second panel shows how the semiaxes, as computed by Equations (12), relate to the Earth's equatorial radius. The deviations are exaggerated by a factor 1000 for the semi-major axis which only varies by a few metres. The semi-minor axis varies between the Earth's polar and equatorial radii in reflection of the angle of the zero-Doppler plane. The third panel shows the ellipse parameters flattening ($f = 1 - b/a$) and ellipticity squared ($e^2 = f(2 - f)$) for the intersection ellipses. As expected, and consistent with the semiaxes variations, the values range between those of the Earth's meridional ellipse and a circle ($e^2 = f = 0$). The final panel shows the coordinates of the intersection ellipse's centre point, as computed by Equation (10). The point is within a few kilometres of the Earth centre.

The intersection between an ellipse and a circle gives a quartic equation, and while there are closed-form solutions to these equations, iterative methods will again give answers to the required accuracy much more quickly and more easily. We use the reduced latitude β in $(a, b) = (A \cos \beta, B \sin \beta)$ as iteration variable, and the difference from the desired range as the function whose root we're trying to find:

$$f = da^2 + db^2 - r^2 \quad (13)$$

$$df = -2 \cdot da \cdot A \cdot \sin \beta + 2 \cdot db \cdot B \cdot \cos \beta \quad (14)$$

where

$$da = A \cdot \cos \beta - a_c \quad (15)$$

$$db = B \cdot \sin \beta - b_c \quad (16)$$

and (a_c, b_c) is the satellite's coordinates in the (a, b) system.

We can use the same kind of triangle equation to find a good starting point for this iteration. In practical tests, we converge to less than a μ m in three iterations, and three iterations are completed in roughly half the time of a single iteration of the 2D Newton search.

3.2. Beyond the Plane: Nonzero Height

When we geolocate a target at nonzero height, we are still looking for a point strictly within the zero Doppler plane, but we have added two complexities that were not present in the zero-height case:

1. the point is no longer on the intersection ellipse, and
2. the distance from the point to the ellipse is not the same as the distance from the point to the ellipsoid (the height which we're interested in).

For an arbitrary ellipsoid and plane configuration, these problems are not easily tractable. In the context of a satellite in low Earth orbit, the following properties make our problem more tractable:

1. The ellipsoid is close to spherical (for WGS-84, $f = 1/298.257223563$)
2. The distance from the origin (centre of the Earth) to the zero Doppler plane is typically small relative to the Earth radius, i.e. a few km.

Hence, the intersection ellipse is almost exactly circular, and the zero Doppler plane and the local tangent plane are very nearly normal to each other.

In the 1D iteration described above, we use the ellipsoid with half-axes $A + h$ and $B + h$ instead of A and B . This is not exactly the curve where we're looking for a solution, but the difference is nowhere greater than 1–2 centimetres for terrestrial targets, see Figure 4 and caption.

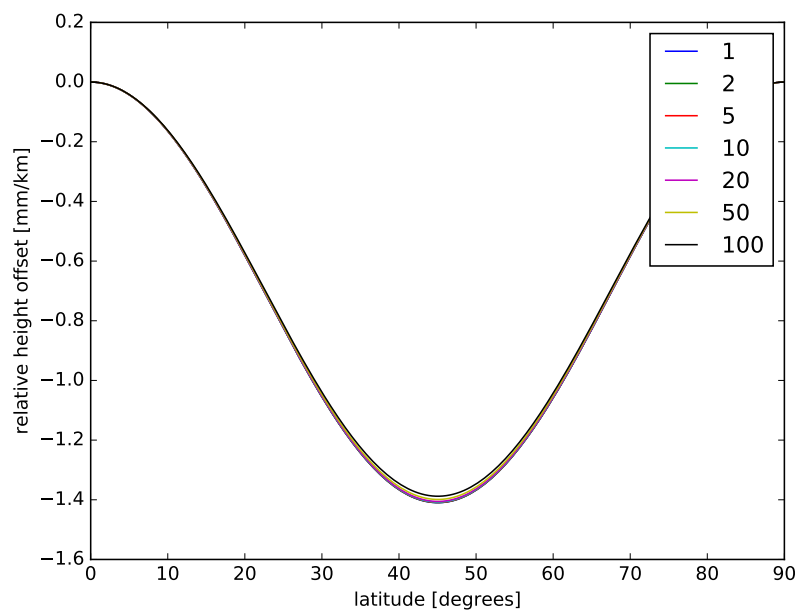


Figure 4. Increasing the radius of a circle by h results in a new curve where every point is at a distance h from the original circle. For an ellipse, increasing both semi-axes with h results in a new curve which is at distance h for points on the axes, but slightly closer at intermediate latitudes. Plotted here is the size of this offset relative to the nominal height, in units of millimetres per kilometer of height, for a meridional ellipse of WGS-84, and for various heights h from 1 to 100 km. As can be seen from Figure 3, the intersection ellipse is nowhere more eccentric than the meridional ellipse, so these curves represent an upper bound on the contribution to height misestimation from this effect.

The iteration converges to within 2 nm in range in three iterations, just as for the zero-height case.

The next approximation is we are now close enough to the sought coordinates that the surface normal to the ellipsoid which passes through the target is indistinguishable¹ from the one computed in

¹ The dot product between this normal vector and the one computed at the foot point of the most carefully corrected process described herein differs from 1 in the 11th digit.

P_0 , the point we arrived at using the 1D iteration over the enlarged ellipse. It differs from the normal to the intersection ellipse through this point in the intersection plane, though, and the dot product between these two normal vectors gives the factor by which the height above the ellipsoid we have arrived at, h_0 , differs from the desired height, h :

$$h_0 = h \cdot (\mathbf{n}_{2D} \cdot \mathbf{n}_{3D}), \quad h_0 < h. \quad (17)$$

While we could easily compute a new height h' and repeat the 1D Newton method, it turns out we are now close enough to the true solution that we can compute directly the 3D direction and step to arrive at the desired solution. The geometry is shown in Figure 5.

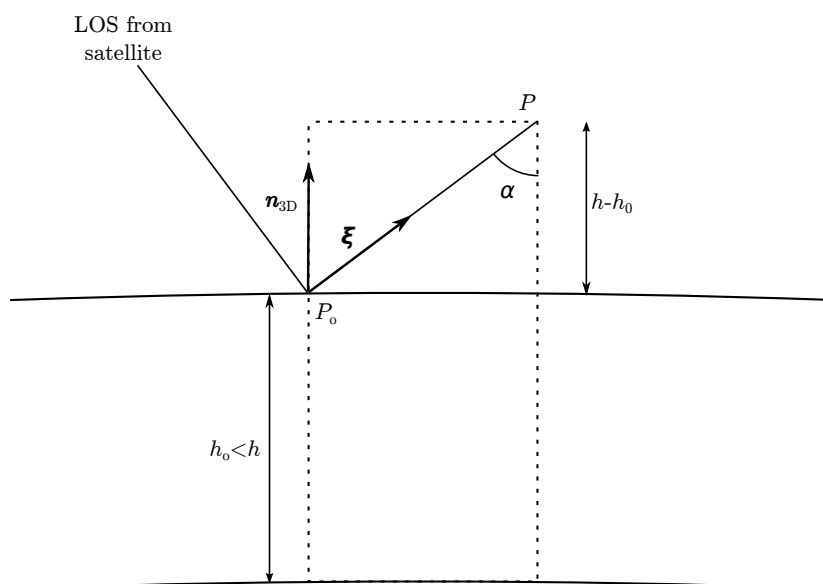


Figure 5. How we arrive at the step size. The plane of the illustration is not exactly the zero Doppler plane, but the points P and P_0 and surface normal vector \mathbf{n}_{3D} all lie in the illustration plane. The upper and lower nearly-horizontal curves represent the enlarged ellipse used to find the approximate solution P_0 and the line drawn by its foot points on the ellipsoid projected back into the illustration plane, respectively. In practical terms the distances in this illustration are so small that this is a line with constant offset h_0 from the enlarged ellipse, and both ellipses are well represented as straight lines within the figure. The curvature of the ellipses is highly exaggerated in the illustration.

The point we arrived at, P_0 , lies in the correct plane, has correct range to the satellite, it has a height very close to h above the ellipse, and height $h_0 < h$ above the ellipsoid. Since $r \gg h - h_0$, the step direction is perpendicular to the relative position between satellite and P_0 , and inside the zero Doppler plane, which means it can be expressed by way of the unit vector $\boldsymbol{\zeta} \propto (P_0 - P_{\text{sat}}) \times V_{\text{sat}}$. In the plane containing the vector $\boldsymbol{\zeta}$ and the surface normal to the ellipsoid that passes through the desired point P , we can easily see that the step size is $(h - h_0) / \cos \alpha$, and $\cos \alpha = \boldsymbol{\zeta} \cdot \mathbf{n}_{3D}$, i.e.

$$\delta \mathbf{P} = \frac{h - h_0}{\boldsymbol{\zeta} \cdot \mathbf{n}_{3D}} \boldsymbol{\zeta}. \quad (18)$$

The unit normal vectors \mathbf{n}_{2D} and \mathbf{n}_{3D} are given directly by their foot point coordinates in the (t, u) and (x, y, z) coordinates, respectively:

$$\mathbf{n}_{2D} = a/A^2 \hat{\mathbf{a}} + b/B^2 \hat{\mathbf{b}} \quad (19)$$

$$\mathbf{n}_{3D} = [x/r_e^2, y/r_e^2, z/r_p^2] \quad (20)$$

We have now obtained the 3D coordinates of the target in geocentric Cartesian coordinates. For some applications, this is the desired result. If geodetic coordinates are required, the transformation described in Section 2 above can be applied.

4. Accuracy of the Results

For targets at ellipsoid height, the described method (including choice of iteration start point) gives answers limited by floating-point accuracy in three iterations.

For targets at any interesting nonzero height, we arrive at a solution which is at correct range to within a few tens of micrometers, and similarly not out of the zero-Doppler plane by an amount larger than this. The correction (18) does not change the range or out-of-plane errors appreciably.

The error that could potentially remain affects the precise vertical and horizontal coordinates of the target, and ranges from less than a millimetre to a few millimetres, as seen in the third panel of Figure 6. This error is much smaller than the absolute accuracy of orbital information, and also of the digital elevation models. Therefore, it has no practical consequence.

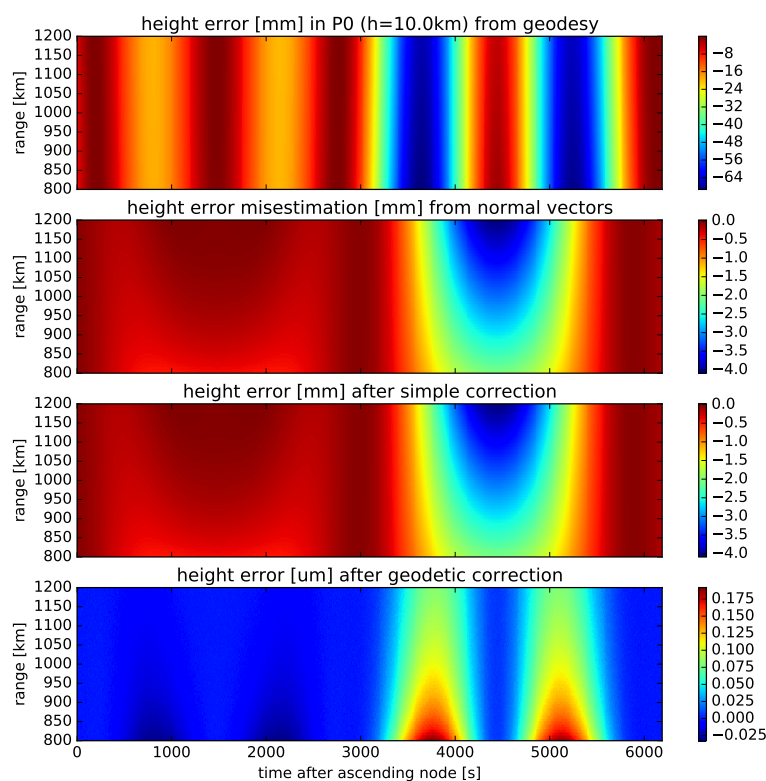


Figure 6. Height errors for targets at 10 km altitude for the same orbit shown in Figure 3 as functions of seconds after ascending node (x) and range in kilometres (y). The first panel shows the difference between the height h_0 of the point P_0 above the ellipsoid as given by Equation (6) and the 10 km height sought, in units of millimetres. The second panel shows by how much the height error described by Equation (17) misestimates the height error in the first panel. The height error is mostly determined by latitude. It is near zero near the equator and the poles, and largest at intermediate latitudes. The misestimation is only noticeable when height error and intersection ellipse eccentricity interact. The third panel shows the height error that remains after the correction step illustrated in Figure 5 has been applied. The fact that this error is indistinguishable from the error misestimation in the previous panel indicates that the step is correctly computed and applied. This is also demonstrated by the final panel, where the height error after correction step is also shown, but when the step was scaled by $h - h_0$ and h_0 was computed using (6). Notice that in this panel, the error is in units of micrometres.

It is of academic interest to remove even this tiny residual error. The only source of the error is the misestimation of the height h_0 of point P_0 above the ellipsoid. Using the geodetic formula for height

(6) removes the error completely, as seen in the final panel of Figure 6, at the cost of one conversion from Cartesian to geodetic coordinates.

A less computationally expensive possibility is to model this misestimation as a function of ellipticity and target height. All terrestrial targets ($h < 10$ km) of interest to SAR are close enough to the ellipsoid that errors scale linearly with altitude, as shown by Figure 4. This means that curves such as those in the figure can be precomputed for a single height and for ellipticities between the limits seen in Figure 3. A correction to h_0 can be interpolated from these curves by latitude and ellipticity, and scaled by the height.

5. Efficiency Considerations

Most end products of SAR data for Earth observation will end up being geolocalized ([5] Section 1.5). Especially for full-resolution products this is a computationally expensive operation, and improving the efficiency of this conversion is valuable to reduce this cost. To see the expected impact of this method we must try to count fundamental operations, since benchmarks in a high-level programming language can be distorted by factors having little to do with the properties of the algorithms themselves.

The performance baseline is the 2D Newton-Raphson iteration described in Section 3, which we will call geodetic iteration in the following. We will not count the time spent finding a candidate position, since that is comparable in the two algorithms. We will also disregard time spent propagating or interpolating statevectors for the platform's trajectory, since these costs are equal for the two algorithms.

Each geodetic iteration involves the evaluation of the dot product (8), which is done in ECEF, so the coordinates must be converted according to Equations (1)–(3), along with their partial derivatives with regard to latitude and longitude, and some other bookkeeping operations. If care is taken to compute sines, cosines and other common terms only once in each iteration, this involves a total of 4 trigonometric functions, 2 square root, 51 multiplications, 7 divisions, and 46 additions or subtractions. The cost of comparing the metric to a threshold is not counted,

The in-plane solution has some setup costs associated with the computation of the intersection ellipse and its basis vectors. We shall count up these costs, but we will point out that these results are common to all points with a common azimuth coordinate, so the cost can be divided over hundreds or thousands of points in typical cases. Computing the intersection ellipse, its basis and parameters (up to but not including Equation (13)) involves 3 trigonometric functions, 4 square roots, 116 multiplications, 11 divisions, and 45 additions or subtractions.

The computation necessary for each iteration and each point (Equations (13)–(16)) now consist of 2 trigonometric functions 11 multiplications, 1 division and 6 additions/subtractions.

When height is not zero, the extra step to correct for out-of-plane height error involves 28 multiplications, 15 divisions, 17 additions and 3 square roots per point

The relative costs of the different types of operations depend on the details of the computer hardware used, but for a current high-performance computer the cost of operations as multiples of the time spent on a single addition or subtraction we will use the factors 3 (multiplications), 5 (divisions), 7 (square roots) and 13 (trigonometric functions).

With these relative costs, the cost per geodetic iteration is roughly 300 units, while the in-plane method costs 70 units per iteration, roughly a factor of 4 improvement.

References

1. Burtch, R. (2006), A Comparison of Methods used in Rectangular to Geodetic Coordinate Transformations, *ACSM Annual Conference and Technology Exhibition*, pp. 1–25.
2. Bowring, B. R. (1976), TRANSFORMATION FROM SPATIAL TO GEOGRAPHICAL COORDINATES, *Survey Review*, 23(181), 323–327, doi:10.1179/sre.1976.23.181.323.
3. Klein, P. P. (2012). On the Ellipsoid and Plane Intersection Equation. *Applied Mathematics*, 2012, 1634–1640. doi:10.4236/am.2012.311226.

4. Grydeland, T., & Larsen, Y. (2018). Beyond Plane Sailing : Solving the Range-Doppler equations in a reduced geometry. Proceedings of EUSAR 2018: 12th European Conference on Synthetic Aperture Radar, 4pp.
5. Franceschetti, G., & Lanari, R. (1999). Synthetic Aperture Radar Processing. CRC Press.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.