Tolerance Estimation of Low-Frequency Marsquakes using Ellipsoidal Trilateration

Introduction

Given simulated and reported waveform data relating to Marsquakes [1] taken from an InSight seismometer, the task at hand is to find a way to accurately account for the allowable tolerance of reported errors in relation to their respective simulations. The data comes in form of ordered triples representing amplitudes of the P waves (1 input) and S waves (2 inputs), each with an uncertainty. Together, these form an ellipsoid of uncertainty centered at the endpoint of the vector of observed amplitudes, when represented graphically in 3 dimensions.

Since we care about the relative amplitudes of simulations and observations, an angle ϵ between the edge of the ellipsoid and the observed vector is used to estimate tolerance in comparison to the angle ζ that the simulation forms with the observation, which is a measure of "how good" the simulation is. Angles are used because they preserve the geometric property that is affected fundamentally by the ratios of amplitudes as opposed to their absolute values.

Current Solution

The ellipsoid is approximated to a sphere (all 3 wave amplitudes are assumed to have the same uncertainty), for which a tolerance angle ϵ can be calculated, represented by the maximum angle that can be formed between a line through the origin touching the surface of the sphere and the vector (denoted $\boldsymbol{A}_o$) of observed amplitudes.

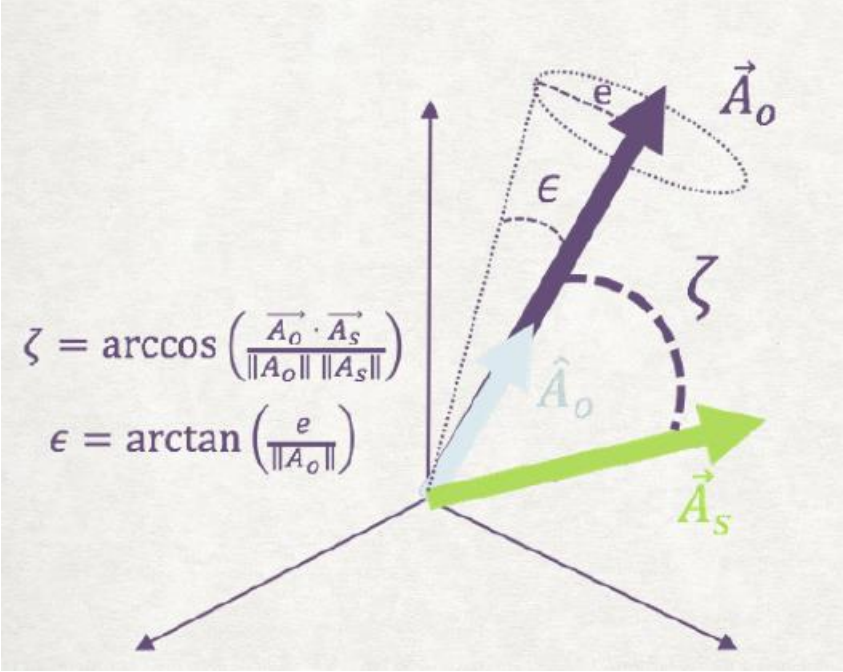

Figure 1: Graphical representation of tolerance angle ϵ

This solution's limitation is that ϵ is as a result of the approximation, constant for all possible orientations of the vector $\boldsymbol{A_s}$ of simulated amplitudes. However, it should be such that the tolerance is minimum when $\boldsymbol{A_s}$ is oriented across the thinnest part of the ellipsoid, and vice versa.

New Approach

The absolute values of the simulations do not matter, only the relative magnitudes of each input wave. Graphically, this means that for any simulated vector $\boldsymbol{A_s}$, any multiple of that vector should hold the same significance in relation to an observed vector $\boldsymbol{A_o}$ and its corresponding ellipsoid of uncertainty defined by the uncertainty vector $\boldsymbol{U}$. The common characteristic of all multiples of $\boldsymbol{A_s}$ is that they form the same angle ζ with $\boldsymbol{A_o}$, and together with $\boldsymbol{A_o}$ lie in the same plane.

The new approach borrows from the idea of trilateration which uses the intersection of spheres to locate optimal points. This time, we make use of the aforementioned common characteristic and the intersection of ellipsoids to locate the points in question. Consequently, the tolerance ϵ is defined to be the half of the maximum angle that can be formed between two lines passing though the origin, contained within the plane (denoted $Plane_1$) defined by $\boldsymbol{A_o}$ and $\boldsymbol{A_s}$ and touching the cross-section of the ellipsoid sliced by the plane.

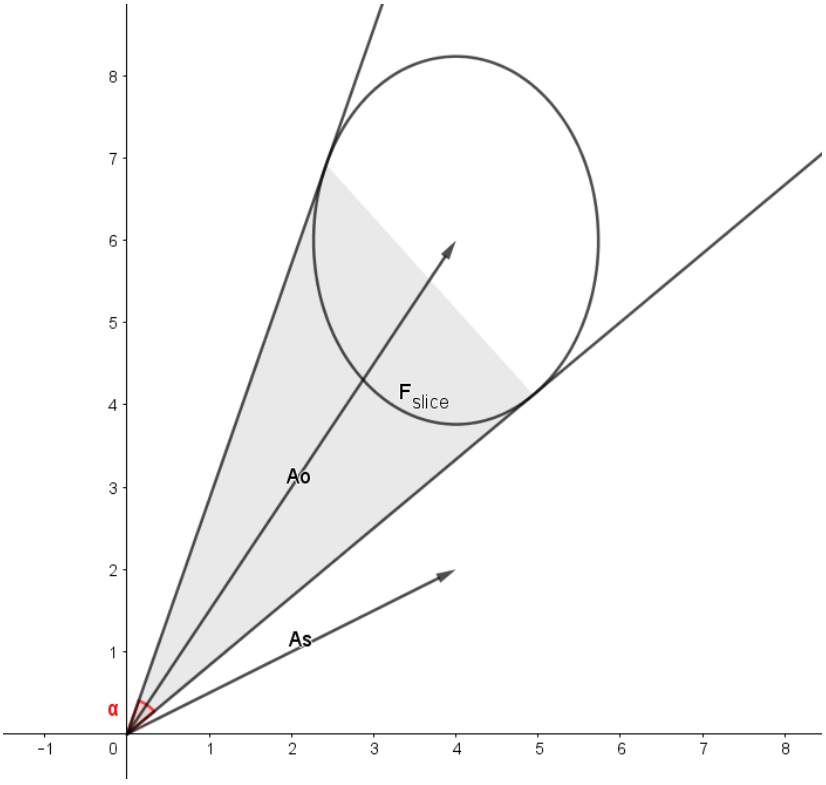

Figure 2: Definition of tolerance $\epsilon = \frac{1}{2}\alpha$ given a cross-sectional slice of the ellipsoid for arbitrary parameters

When a plane intersects with an ellipsoid, the resultant shape is always an ellipse whose associated angles can be calculated and optimized given a parametric representation of this ellipse. However, while a closed form parametrization exists, it is generally difficult to derive [2]. So, the idea is to use knowledge of tangential properties of a line on a surface to locate the optimal points of this ellipse without having to parametrize it. These points, for any ellipse of intersection, must exist on the surface of the elliptic cone tangential to the ellipsoid, whose vertex is at the origin.

Analytic Solution

Considering the arbitrary vectors:

$$\boldsymbol{A}_s = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 3 \end{bmatrix}, \boldsymbol{A}_o = \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}, \boldsymbol{U} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \sqrt{2} \\ \sqrt{3} \\ \sqrt{5} \end{bmatrix}$$

Note: The numbers assigned are examples for easy visualization WLOG.

For conciseness, let the following notation apply for all scalar functions f:

$$\sum_{cyc} f(x) = f(x) + f(y) + f(z)$$

The general form of the ellipsoid of uncertainty is thus given by the level surface of function $F(x, y, z)$ at $F = 1$, defined such that

$$F(x, y, z) = \sum_{cyc} \left(\frac{x - x_o}{u_x}\right)^2 = 1 \qquad \text{eq (1)}$$

The elliptic cone intersects with the ellipsoid at every point $P_o = (x, y, z)$ for which the following equation holds:

$$\nabla F \cdot \boldsymbol{P}_o = 0 \qquad \text{eq (2)}$$

That is to say, every vector extending from the origin to a point on the intersection is orthogonal to the gradient vector field for which the ellipsoid is a level set.

$$\nabla F = \begin{bmatrix} \frac{2(x - x_o)}{{u_x}^2} \\ \frac{2(y - y_o)}{{u_y}^2} \\ \frac{2(z - z)}{{u_z}^2} \end{bmatrix} \qquad \text{eq (3)}$$

Combining (2) and (3):

$$\sum_{cyc} \frac{2x(x-x_o)}{{u_x}^2} = 0 \qquad \text{eq (4)}$$

Define $G(x, y, z)$ such that

$$G(x, y, z) = \sum_{cyc} \left(\frac{x^2 - x_o x}{{u_x}^2} \right) = 0 \qquad \text{eq (5)}$$

The level surface of function G at $G = 0$ describes a new ellipsoid which always intersects with the original one.

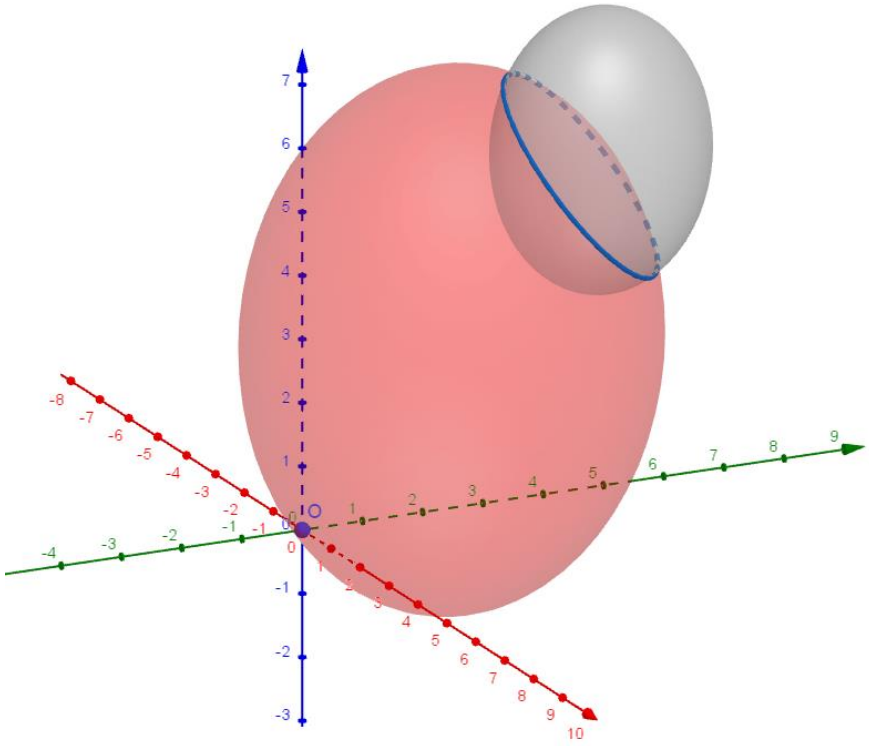

Figure 3: Intersection of ellipsoids defined by F (grey) and G (red) for arbitrary parameters

Combining (1) and (5):

$$F - G = \sum_{cyc} \left(\frac{x^2 - 2x_o x + {x_o}^2}{{u_x}^2} \right) - \sum_{cyc} \left(\frac{x^2 - x_o x}{{u_x}^2} \right) = 1 - 0 \qquad \text{eq (6)}$$

$$\sum_{cyc} \left(\frac{{x_o}^2 - x_o x}{{u_x}^2} \right) = 1 \qquad \text{eq (7)}$$

$$\sum_{cyc} \left(\frac{x_o}{u_x} \right)^2 - \sum_{cyc} \left(\frac{x_o}{{u_x}^2} x \right) = 1 \qquad \text{eq (8)}$$

This is the equation of a plane (denoted $Plane_2$) whose normal vector $\boldsymbol{n}_2$ is

$$\boldsymbol{n}_2 = \begin{bmatrix} \frac{x_o}{{u_x}^2} \\ \frac{y_o}{{u_y}^2} \\ \frac{z_o}{{u_z}^2} \end{bmatrix} \qquad \text{eq (9)}$$

In standard form, $Plane_2$ has the equation

$$\frac{x_o}{{u_x}^2}x + \frac{y_o}{{u_y}^2}y + \frac{z_o}{{u_z}^2}z = \boldsymbol{n}_2 \cdot \boldsymbol{A}_o - 1 \qquad \text{eq (10)}$$

Note that $Plane_1$, defined by $\boldsymbol{A}_o$ and $\boldsymbol{A}_s$, is described by its normal vector $\boldsymbol{n}_1$ such that

$$\boldsymbol{n}_1 = \boldsymbol{A}_o \times \boldsymbol{A}_s = \begin{bmatrix} y_o z_s - y_s z_o \\ x_s z_o - x_o z_s \\ x_o y_s - x_s y_o \end{bmatrix} \qquad \text{eq (11)}$$

$Plane_1$ and $Plane_2$ both intersect with the ellipsoid, each forming an ellipse with it. The points of intersection of these ellipses are of interest, and they lie along the line of intersection of the planes, whose directional vector $\boldsymbol{m}$ is orthogonal to $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$.

$$\boldsymbol{m} = \boldsymbol{n}_1 \times \boldsymbol{n}_2 = \begin{bmatrix} \frac{z_o}{{u_z}^2}(x_s z_o - x_o z_s) - \frac{y_o}{{u_y}^2}(x_o y_s - x_s y_o) \\ \frac{x_o}{{u_x}^2}(x_o y_s - x_s y_o) - \frac{z_o}{{u_z}^2}(y_o z_s - y_s z_o) \\ \frac{y_o}{{u_y}^2}(y_o z_s - y_s z_o) - \frac{x_o}{{u_x}^2}(x_s z_o - x_o z_s) \end{bmatrix} \qquad \text{eq (12)}$$

This line can be parametrized in $\boldsymbol{r}(t)$ as

$$\boldsymbol{r}(t) = \boldsymbol{C} + \boldsymbol{m}t \qquad \text{eq (13)}$$

where $\boldsymbol{C}$ is the point of intersection of $\boldsymbol{A}_o$ and $Plane_2$ (which by definition must lie on the line).

$$\boldsymbol{C} = k\boldsymbol{A}_o \qquad \text{eq (14)}$$

$$\boldsymbol{n}_2 \cdot \boldsymbol{C} = \boldsymbol{n}_2 \cdot \boldsymbol{A}_o - 1 \qquad \text{eq (15)}$$

Combining equations (14) and (15):

$$k\boldsymbol{n}_2 \cdot \boldsymbol{A}_o = \boldsymbol{n}_2 \cdot \boldsymbol{A}_o - 1 \qquad \text{eq (16)}$$

$$k = 1 - \frac{1}{\boldsymbol{n}_2 \cdot \boldsymbol{A}_o} \qquad \text{eq (17)}$$

Combining equations (13), (14) and (17):

$$\boldsymbol{r}(t) \;=\; \left(1-\frac{1}{\boldsymbol{n_2}\cdot\boldsymbol{A_o}}\right)\boldsymbol{A_o} \;+\; \boldsymbol{m}t \qquad \text{eq (18)}$$

The optimal points are such that $\boldsymbol{r}(t)$ intersects with the ellipsoid defined by F:

$$\sum_{cyc}\left(\frac{x-x_o}{u_x}\right)^2 = \sum_{cyc}\left(\frac{\left(1-\frac{1}{\boldsymbol{n_2}\cdot\boldsymbol{A_o}}\right)x_o + m_x t - x_o}{u_x}\right)^2 = 1 \qquad \text{eq (19)}$$

$$\sum_{cyc}\left(\frac{m_x}{u_x}t - \frac{x_o}{u_x(\boldsymbol{n_2}\cdot\boldsymbol{A_o})}\right)^2 = 1 \qquad \text{eq (20)}$$

$$\left(\sum_{cyc}\left(\frac{m_x}{u_x}\right)^2\right)t^2 - 2\left(\sum_{cyc}\left(\frac{x_o}{{u_x}^2}\frac{m_x}{(\boldsymbol{n_2}\cdot\boldsymbol{A_o})}\right)\right)t + \left(\sum_{cyc}\left(\frac{x_o}{{u_x}^2}\frac{x_o}{(\boldsymbol{n_2}\cdot\boldsymbol{A_o})^2}\right)\right) - 1 = 0 \qquad \text{eq (21)}$$

Defining a vector $\boldsymbol{v}$ for utility:

$$\boldsymbol{v} = \begin{bmatrix} \frac{m_x}{u_x} \\ \frac{m_y}{u_y} \\ \frac{m_z}{u_z} \end{bmatrix} \qquad \text{eq (22)}$$

Simplifying equation (21):

$$(\boldsymbol{v}\cdot\boldsymbol{v})t^2 - 2\left(\frac{\boldsymbol{n_2}\cdot\boldsymbol{m}}{\boldsymbol{n_2}\cdot\boldsymbol{A_o}}\right)t + \frac{1}{\boldsymbol{n_2}\cdot\boldsymbol{A_o}} - 1 = 0 \qquad \text{eq (23)}$$

This is a quadratic equation whose roots are given by

$$(t_1, t_2) = \frac{\left(\frac{\boldsymbol{n_2}\cdot\boldsymbol{m}}{\boldsymbol{n_2}\cdot\boldsymbol{A_o}}\right) \pm \sqrt{\left(\frac{\boldsymbol{n_2}\cdot\boldsymbol{m}}{\boldsymbol{n_2}\cdot\boldsymbol{A_o}}\right)^2 + (\boldsymbol{v}\cdot\boldsymbol{v})\left(1-\frac{1}{\boldsymbol{n_2}\cdot\boldsymbol{A_o}}\right)}}{\boldsymbol{v}\cdot\boldsymbol{v}} \qquad \text{eq (24)}$$

The points defined by these roots are used to calculate the final tolerance angle:

$$\epsilon = \frac{1}{2}\cos^{-1}\left(\frac{\boldsymbol{r}(t_1)\cdot\boldsymbol{r}(t_2)}{\|\boldsymbol{r}(t_1)\|\|\boldsymbol{r}(t_2)\|}\right) \blacksquare \qquad \text{eq (25)}$$

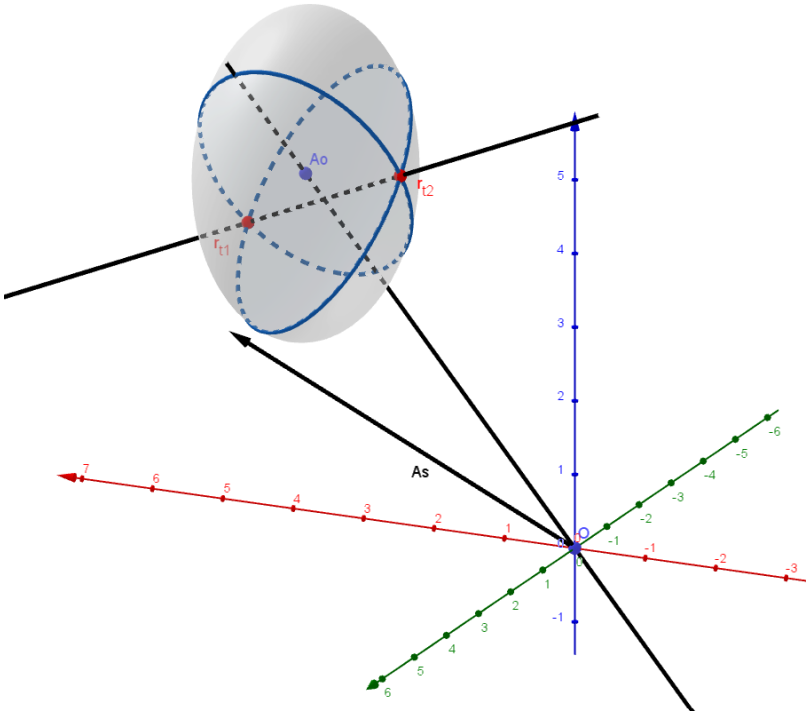

Figure 4: Geometric representation of optimal points $\boldsymbol{r}(t_1)$ and $\boldsymbol{r}(t_2)$ for arbitrary parameters

Numerical Solution

Considering the arbitrary vectors $\boldsymbol{A_s}$, $\boldsymbol{A_o}$ and $\boldsymbol{U}$, let $\boldsymbol{A_v}$ denote the vector of variables (x, y, z).

The following is true of any angle θ within ϵ, for θ measured between an acceptable simulation and the original observation:

$$\forall x: (x - x_o)^2 \leq {u_x}^2 \qquad \text{eq (26)}$$

$$\forall y: (y - y_o)^2 \leq {u_y}^2 \left(1 - \left(\frac{x - x_o}{u_x}\right)^2\right) \qquad \text{eq (27)}$$

$$\forall z: (z - z_o)^2 \leq {u_z}^2 \left(1 - \left(\frac{x - x_o}{u_x}\right)^2 - \left(\frac{y - y_o}{u_y}\right)^2\right) \qquad \text{eq (28)}$$

$$\forall \boldsymbol{A_v}: \boldsymbol{A_v} \cdot (\boldsymbol{A_s} \times \boldsymbol{A_o}) = 0 \qquad \text{eq (29)}$$

$$\theta = \cos^{-1}\left(\frac{\boldsymbol{A_o} \cdot \boldsymbol{A_v}}{\|\boldsymbol{A_o}\| \|\boldsymbol{A_v}\|}\right) \qquad \text{eq (30)}$$

If a weighted distribution of θ is plotted, it reflects low likely it is to find an acceptable angle within the range of acceptable angles for any cross-sectional ellipse defined by the plane parallel to $\boldsymbol{A_s}$ and $\boldsymbol{A_o}$.

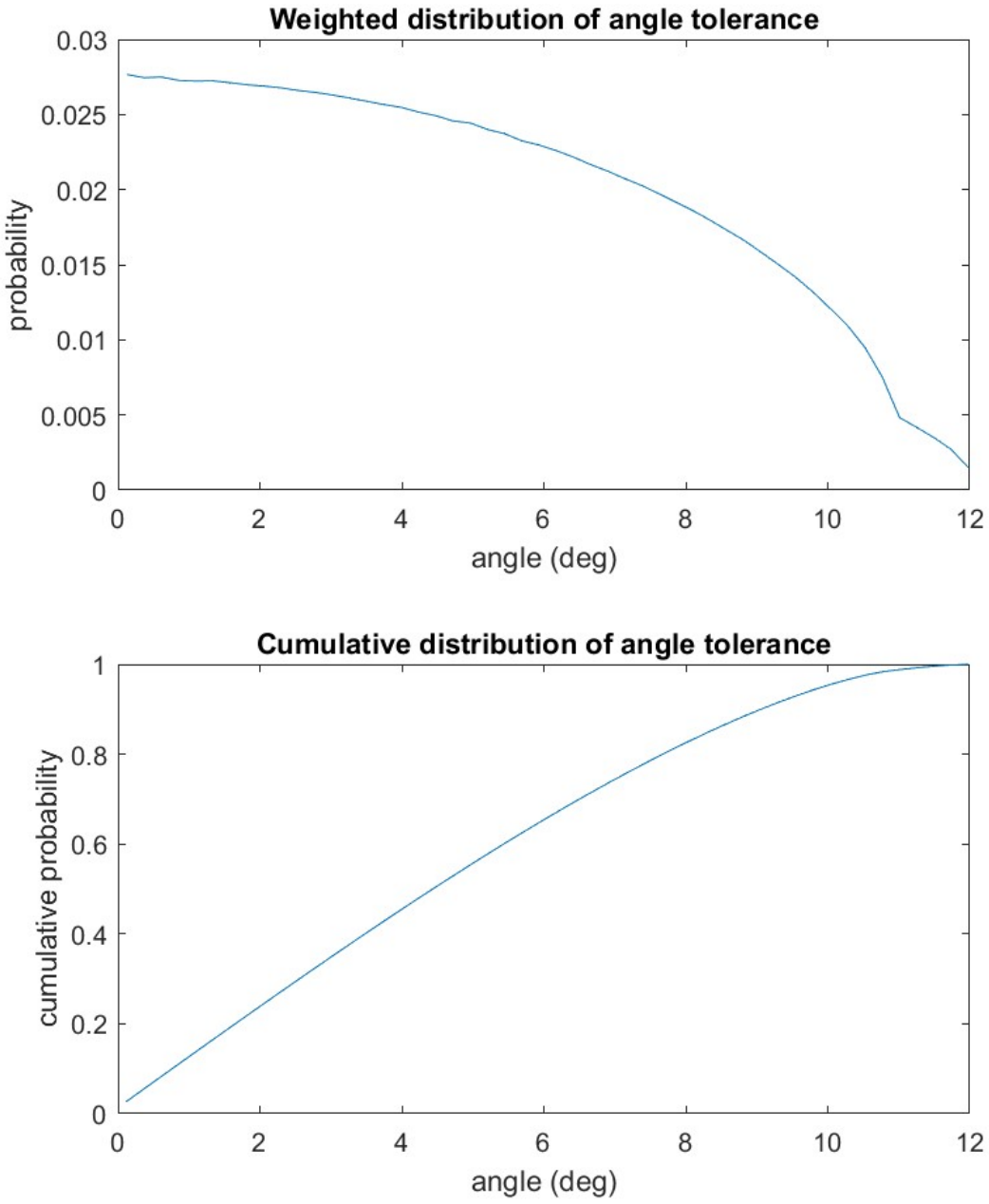

Figure 5: Probability distributions of allowable angles $\theta \leq \epsilon$ for arbitrary parameters

Results and Discussion

The calculated values of tolerance from analytic (appendix 1) vs numerical (appendix 2) solutions for the example parameters are as follows:

Analytic: $\epsilon = 11.52°$

Numerical: $\epsilon = 12.10°$

The discrepancy is because vector $\boldsymbol{A}_o$ does not actually bisect the angle α formed between the lines connecting the points of interest to the origin, so halving that angle in the analytic computation is an approximation. Specifically, the approximation averages out the values after the slope discontinuity in the graph of weighted distribution of angle tolerance (figure 5) which is a way of extrapolating that graph to maintain its smoothness.

Geometrically, making this approximation is significant because assuming $\boldsymbol{A}_o$ divides α into angles θ_1 and θ_2 where $\theta_1 \leq \theta_2$; then bisecting α distributes the weights equally onto both sides of $\boldsymbol{A}_o$ for the remaining angle $\theta_2 - \theta_1$. In other words, instead of having every angle between θ_1 and θ_2 come from the larger side, it is assumed that half of it comes from the smaller side, just as is the case with every other $\theta \in [0, \theta_1]$. This similarly shifts the estimated final error by $\frac{1}{2}(\theta_2 - \theta_1)$, which is justified because it eliminates the positive bias on the tolerance angle created by θ_2.

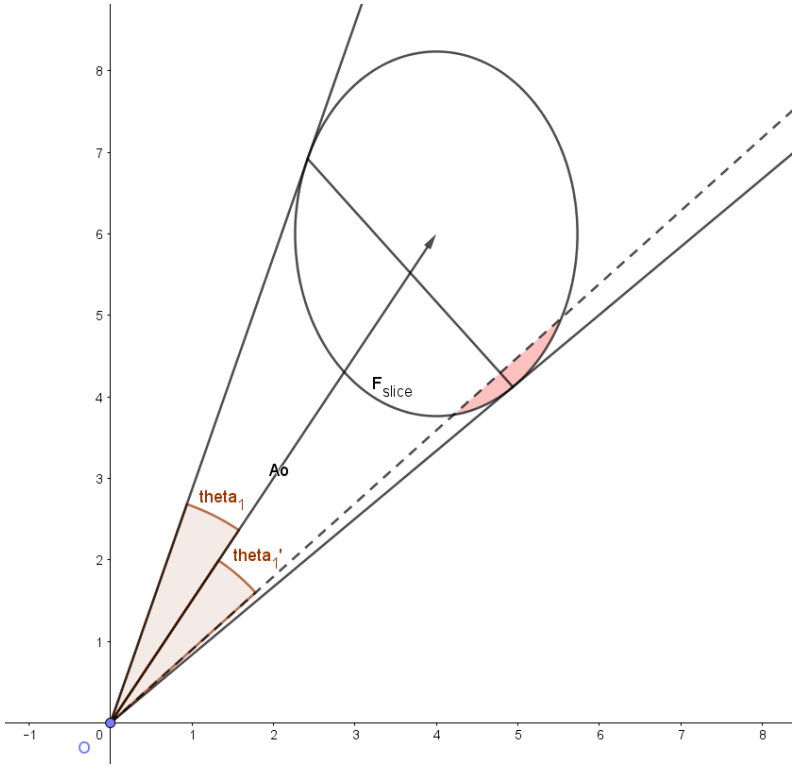

Figure 6: The approximation adjusts ϵ such that the ellipse region right of the dotted line is equally distributed both sides of $\boldsymbol{A}_o$

References

[1] M. Sita and S. van der Lee, “Origins and potential focal mechanisms of three low-frequency Marsquakes detected by a single InSight seismometer,” *JGR: Planets*. [Abstract]. [Accessed May 15, 2022].

[2] P. P. Klein, “On the Ellipsoid and Plane Intersection Equation,” November 2012. Available: https://www.scirp.org/pdf/AM20121100009_89014420.pdf. [Accessed April 10, 2022].

Appendices

Appendix 1: MATLAB Code for Analytic Solution

```
As = input('Vector of simulated amplitudes: ');
Ao = input('Vector of observed amplitudes: ');
U = input('Vector of uncertainties: ');

n1 = cross(Ao,As);
n2 = Ao ./ (U.^2);
m = cross(n1,n2);
v = m ./ U;
k = 1-(1/dot(n2,Ao));
B = dot(n2,m)/dot(n2,Ao);

t1 = (B - sqrt(B^2 + k*dot(v,v)))/dot(v,v);
t2 = (B + sqrt(B^2 + k*dot(v,v)))/dot(v,v);
r1 = r(t1, k, Ao, n1, n2);
r2 = r(t2, k, Ao, n1, n2);
tol = (1/2)*acosd(dot(r1,r2)/(norm(r1)*norm(r2)));

fprintf('Tolerance angle: %.2f\n', tol)

function rt = r(t, k, Vo, m, n)
rt = k*Vo + t*cross(m,n);
end
```

Appendix 2: MATLAB Code for Numerical Solution

```
As = input('Vector of simulated amplitudes: ');
Ao = input('Vector of observed amplitudes: ');
U = input('Vector of uncertainties: ');
bin = 50;
acc = 500;

xo = Ao(1); yo = Ao(2); zo = Ao(3);
ux = U(1); uy = U(2); uz = U(3);
x = linspace(xo-ux,xo+ux,acc);
y = linspace(yo-uy,yo+uy,acc);
z = linspace(zo-uz,zo+uz,acc);

theta = NaN(length(x),length(y),length(z));
n1 = cross(Ao,As);

for ii = 1:length(x)
    for jj = 1:length(y)
        for kk = 1:length(z)
            if abs((y(jj)-yo)/uy) <= sqrt(1-((x(ii)-xo)/ux)^2)
                if abs((z(kk)-zo)/uz) <= sqrt(1-((x(ii)-xo)/ux)^2 ...
                    -((y(jj)-yo)/uy)^2)
                    if abs(dot(n1,[x(ii),y(jj),z(kk)])) <= 0.1
                        theta(ii,jj,kk) = acosd(dot([x(ii),y(jj),z(kk)],Ao) ...
                        /(norm([x(ii),y(jj),z(kk)])*norm(Ao)));
                    end
                end
            end
        end
    end
end

theta_vec = theta(:);
theta_vec = theta_vec(~isnan(theta_vec));

theta_max = max(theta_vec);
fprintf('Max angle: %.2f\n', theta_max)
theta_min = min(theta_vec);
bin_size = (theta_max - theta_min)/bin;
scale = linspace(theta_min,theta_max,bin+1);
real_scale = zeros(1,bin);
for jj = 1:length(real_scale)
    real_scale(jj) = mean([scale(jj),scale(jj+1)]);
end
theta_posn = ceil((theta_vec-theta_min)/bin_size);
theta_posn(theta_posn==0) = 1;
theta_posn(theta_posn==bin+1) = bin;

freq = zeros(1,bin);
for jj = 1:length(theta_posn)
    freq(theta_posn(jj)) = freq(theta_posn(jj))+1;
end

prob_freq = zeros(size(freq));
```

```
for jj = 1:length(prob_freq)
    prob_freq(jj) = freq(jj)/sum(freq);
end

theta_av = sum(real_scale.*prob_freq);
fprintf('Average angle: %.2f\n', theta_av)

cum_freq = zeros(size(prob_freq));
cum_freq(1) = prob_freq(1);
for jj = 2:length(cum_freq)
    cum_freq(jj) = cum_freq(jj-1)+prob_freq(jj);
end

figure
subplot(2,1,1)
plot(real_scale,prob_freq)
xlabel('angle (deg)'),ylabel('probability')
title('Weighted distribution of angle tolerance')
subplot(2,1,2)
plot(real_scale,cum_freq)
xlabel('angle (deg)'),ylabel('cumulative probability')
title('Cumulative distribution of angle tolerance')
```