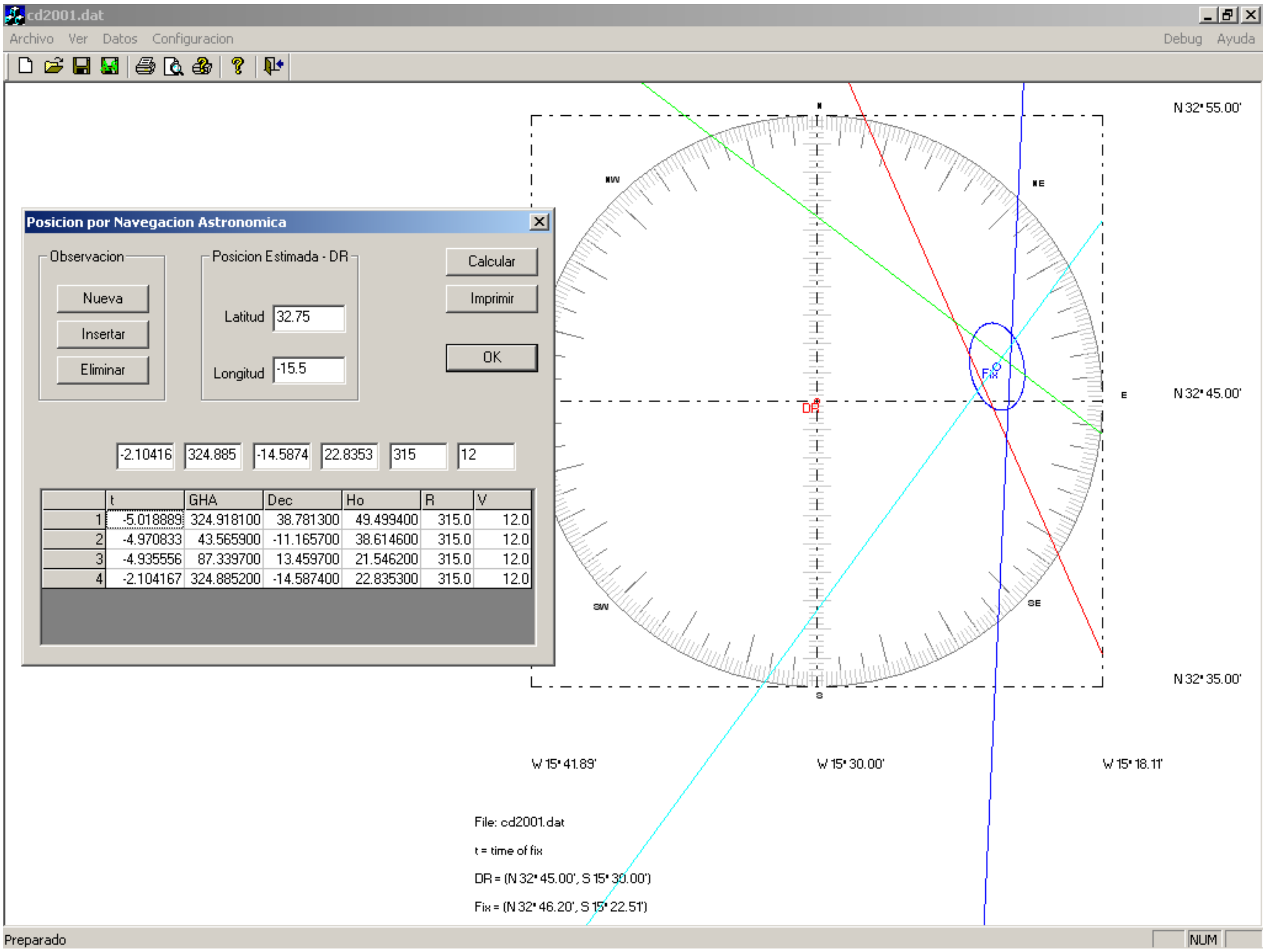
NAVIGATIONAL ALGORITHMS

Celestial Fix by Least squares Sight Reduction algorithm for n LoPs



© Andrés Ruiz
San Sebastián – Donostia
43º 19'N 002ºW
http://sites.google.com/site/navigationalalgorithms/

Index

Abstract

This paper describes an analytic method for calculating the position by the observation of celestial bodies as effective alternative to traditional graphic methods used in celestial navigation.

The algorithm is totally general, allowing the use of simultaneous sights, or observations taken at different times, including navigating a course at a certain speed. It uses the method of least squares to obtain the most probable position, and through successive iterations makes it possible to reduce the error to approximate the circle of equal altitude by the straight line of position. It also provides the error in the calculation of the position and the Confidence ellipse.

Navigational Algorithms
San Sebastián – Donostia
43º 19'N 002ºW
Current version: 201201

It describes the algorithm by C. De Wit *Optimal Estimation of a Multi-Star Fix*, and officially adopted by the *HM Royal Navy,* UK, to calculate the situation by observations of stars using the sextant. It was also included in The Nautical Almanac published annually by the USA Naval Observatory.

It is a robust and effective alternative to the traditional graphical methods to get the position by intersection of lines of position or bisectors.

This algorithm replaces plotting lines of position, **LoP**, for each observation, moving Lops if they are not simultaneous and obtaining of the position, by a calculation that systematizes this process and get the most probable position based on the method of least squares.

For the simplicity of its calculations, especially in matrix form, it can be used with a calculator or spreadsheet.

Variables

UT1 Universal Time
Approximately Greenwich Mean Time: GMT

In degrees:

B Latitude
N (+) / S (–)

L Longitude
E (+) / W (–)

GHA Greenwich Hour Angle
GHA = GHA(Aries) + SHA
W to E from 0° to 360°.

SHA Sidereal Hour Angle
SHA = 360° - Right Ascension.

DEC Declination
N (+) / S (–).

LHA Local Hour Angle
W to E from 0° to 360°.

Ho Observed Altitude
Is apparent altitude corrected for refraction and if appropriate corrected for parallax and semi-diameter, [2].

Hc Calculated or Computed Altitude.

Z Azimuth (true)
Measured clockwise around the horizon from 0° to 360°, is the arc of the horizon between the meridian of a place and the vertical circle passing through a celestial body.

p Intercept of a sight
p = Ho – Hc
Towards = + / Away = –

R Course or track
Measured as for azimuth from 0° to 360°.

V Speed
in knots.

n Number of observations

Sight Reduction

The procedure uses the classic **Marcq Saint Hilaire** method to reduce the sights as the mathematical link between the observer and the celestial body. If you know your estimated latitude and longitude, you can predict the true bearing and the height of the object above the horizon. This angle can then be compared to your corrected sextant angle to produce a position line. With several sights, the method plots a fix through the statistical intersection of these position lines.

The following sight reduction formulae are used:

$$LHA = GHA + L$$

$$Hc = \mathrm{asin}(\sin B \sin DEC + \cos B \cos Dec \cos LHA)$$

$$Z = \mathrm{acos}\left(\frac{\sin DEC - \sin Hc \sin B}{\cos Hc \cos B}\right)$$

if(0 < LHA < 180º) Z = 360 – Z

If the local hour angle is less than 180° then the azimuth is 360° less the product of the above expression:

Running Fix

An estimate can be made of the position at the adopted time of fix. The position at the time of the observations can then be easily calculated provided that the course and speed has been constant. Using speed (V) in knots and the track (R) the equations are:

$$t = UT1_{observation} - UT1_{fix}$$

$$B = B_e + \frac{V\,t}{60}\cos R$$

$$L = L_e + \frac{V\,t}{60}\,\frac{\sin R}{\cos B_e}$$

L_e and B_e are the estimated longitude and latitude at the time of fix and t is the time interval in hours.

Calculated Position

The position lines for one or more observations can be plotted using the azimuth Z and the intercept p:

$$p = Ho - Hc$$

- If p is positive the position line is drawn along the azimuth.
- If p is negative, the position line is away from the assumed position by adding 180° to the azimuth, (Z+180º).

Provided that there are no observation errors, the observer should be close to, or along the position line. Two or more position lines are required to determine a fix.

The procedure uses the -*Method of Least Squares*- to determine a fix from up to three observations.

If pi and Zi, (i=1,n), are the intercept and azimuth for the i observation:

$$A = \sum_{i=1}^{n} \cos^2 Z_i \qquad D = \sum_{i=1}^{n} p_i \cos Z_i$$

$$B = \sum_{i=1}^{n} \cos Z_i \cdot \sin Z_i \qquad E = \sum_{i=1}^{n} p_i \sin Z_i$$

$$C = \sum_{i=1}^{n} \sin^2 Z_i \qquad F = \sum_{i=1}^{n} p_i^2$$

$$G = A\,C - B^2$$

As a checking: A+C = n

An improved estimate of the fix is given by:

$$B_v = B_e + dB = B_e + \frac{C\,D - B\,E}{G}$$

$$L_v = L_e + dL = L_e + \frac{A\,E - B\,D}{G \cos B_e}$$

The distance between the assumed position and the improved estimated position in nautical miles is:

$$Do = 60\sqrt{dL^2 \cos^2 Be + dB^{\ 2}}$$

The method substitutes the DR position with the calculated fix in order to converge on a solution. Do < 20 nm.

$$B_e = B_v$$

$$L_e = L_v$$

Estimated Position Error

If three or more position lines are obtained an estimate of the error in position may be calculated. In general as the number of observation increases the error in the estimated position decreased.

The standard deviation of the estimated position, in nautical miles is:

$$\sigma = 60\sqrt{\frac{S}{n-2}}$$

$$S = F - D\,dB - E\,dL\,COS\,B_e$$

And the standard deviation in latitude and longitude is:

$$\sigma_B = \sigma\sqrt{\frac{C}{G}}$$

$$\sigma_L = \sigma\sqrt{\frac{A}{G}}$$

The **Confidence Ellipse** of axes (a,b) is:

$$\mathrm{a} = \frac{\sigma\,\mathrm{k}}{\sqrt{\frac{n}{2} + \frac{B}{\sin 2\theta}}}$$

$$\mathrm{b} = \frac{\sigma\,\mathrm{k}}{\sqrt{\frac{n}{2} - \frac{B}{\sin 2\theta}}}$$

$$\tan 2\theta = \frac{2B}{A - C}$$

The scale factor is:

$$\mathrm{k} = \sqrt{-2\mathrm{Ln}(1 - \mathrm{Prob})}$$

For a level of 95%, Prob = 0.95

Plot

Taking a system of Cartesian axes, it is possible to draw the various elements that define the celestial fix.

- Origin: estimated or assumed position.
- X axis: according to a parallel, positive eastwards.
- Y axis: according to a meridian, positive towards the North.

Lines of Position

Placed the plot within a square of 20 nautical miles, centred on the assumed position, the LoP defined by p and Z is determined by the intersection with the sides of this square:

- X = ±10
- Y = p ± 10 sin Z/cos Z
- The Lop intersects the vertical sides of the square if:

  -10<=Y<=+10

- Y = ±10
- X = p ± 10 cos Z/sin Z
- The Lop intersects the horizontal sides of the square if:

  -10<=X<=+10

Ellipse

Giving to α values between 0 ° and 360 °, points on the confidence ellipse focused on (Be, Le) are obtained:

$$x = a \cos\alpha \sin\theta - b \sin\alpha \cos\theta + 60\, dL \cos B_e$$

$$y = a \cos\alpha \cos\theta + b \sin\alpha \sin\theta + 60\, dB$$

Position

In each iteration, the origin is chosen in the position obtained in the previous step.

Mathematical basis

The equation on the Cartesian plane of the LoP, around the estimated position, is (see Two celestial LOPs Fix [1]):

$$p = x\sin z + y\cos z$$

For n observations, the most probable position, **MPP**, is the centre of gravity of the polygon formed with the intersection of the n LOPs. Mathematically this is obtained optimising the following equation, the distance between the fix and the LoP:

$$S = \sum_{i=1}^{n}\left[p_i - y\cos z_i - x\sin z_i\right]^2$$

Minimizing the function S:

$$\frac{\partial S}{\partial x} = 0$$

$$\frac{\partial S}{\partial y} = 0$$

And solving the resulting system of equations you can find the solution for the MPP:

$$\begin{bmatrix} \sum_{i=1}^{n}\sin^2 z_i & \sum_{i=1}^{n}\cos z_i \sin z_i \\ \sum_{i=1}^{n}\cos z_i \sin z_i & \sum_{i=1}^{n}\cos^2 z_i \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} p_i \sin z_i \\ \sum_{i=1}^{n} p_i \cos z_i \end{bmatrix}$$

[Z] {x} = {P}

$$\begin{bmatrix} C & B \\ B & A \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} E \\ D \end{bmatrix}$$

{x} = [Z]$^{-1}${P}

Its explicit solution is:

G = det([Z]) = Z_{11} Z_{22} - Z_{12}^2 = CA-B^2

$$x = \frac{1}{G} \begin{vmatrix} E & B \\ D & A \end{vmatrix}$$ = 1/G(EA-DB)

$$y = \frac{1}{G} \begin{vmatrix} C & E \\ B & D \end{vmatrix}$$ = 1/G(CD-BE)

B = B_e + y/60

L = L_e + x/60/COS B_e

Matrix solution

Using matrix notation was greatly simplified:

$$[A] = [\sin z_i \quad \cos z_i]$$

$$\{L\} = [p_i]$$

[A] {x} = {L}

An overdetermined system with 2 unknowns and n equations. It is shown that the least-squares solution comes from solving the following system:

[A]^T [A] {x} = [A]^T {L}

With the previous nomenclature:

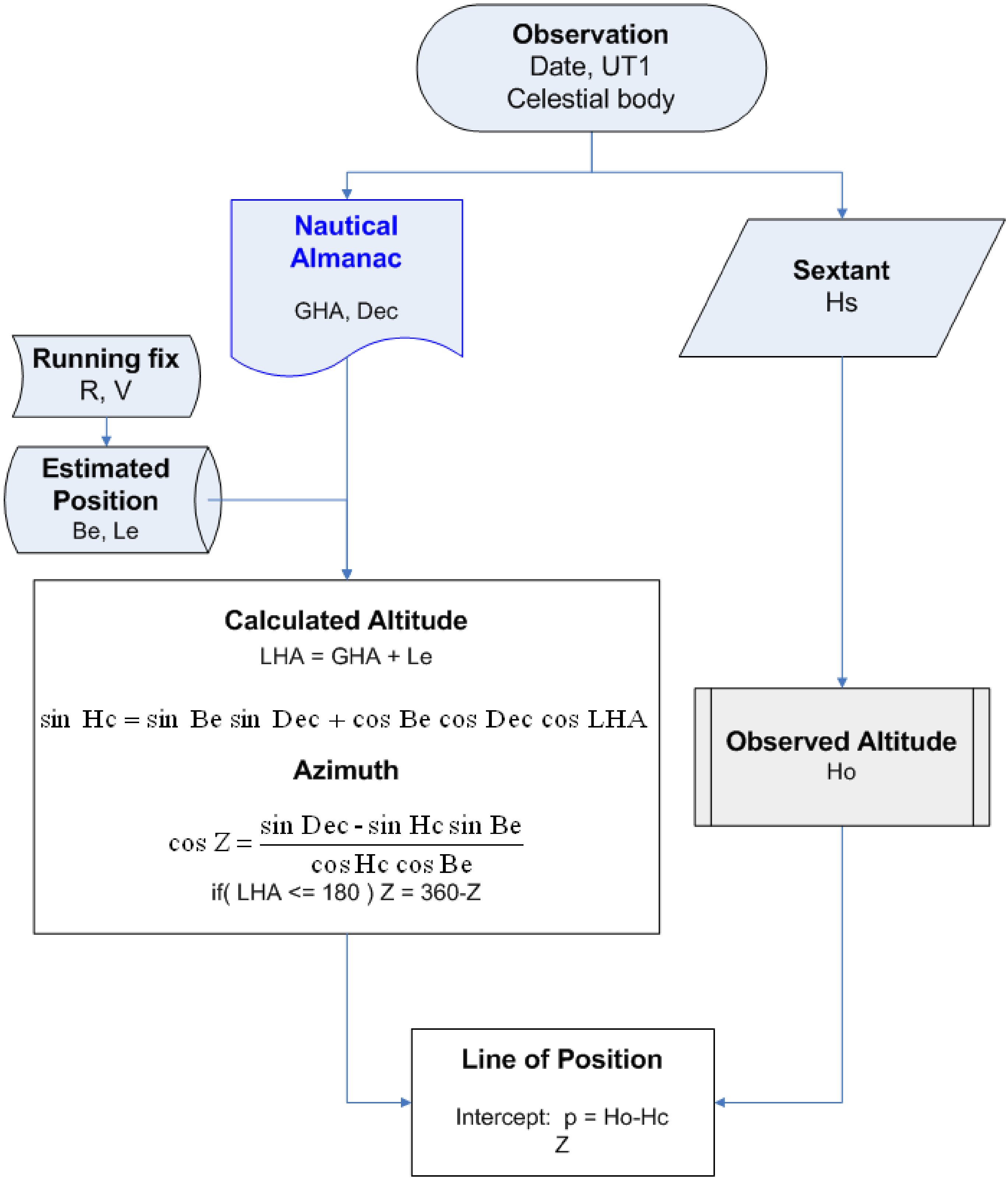
[Z] = [A]^T [A]

{P} = [A]^T {L}

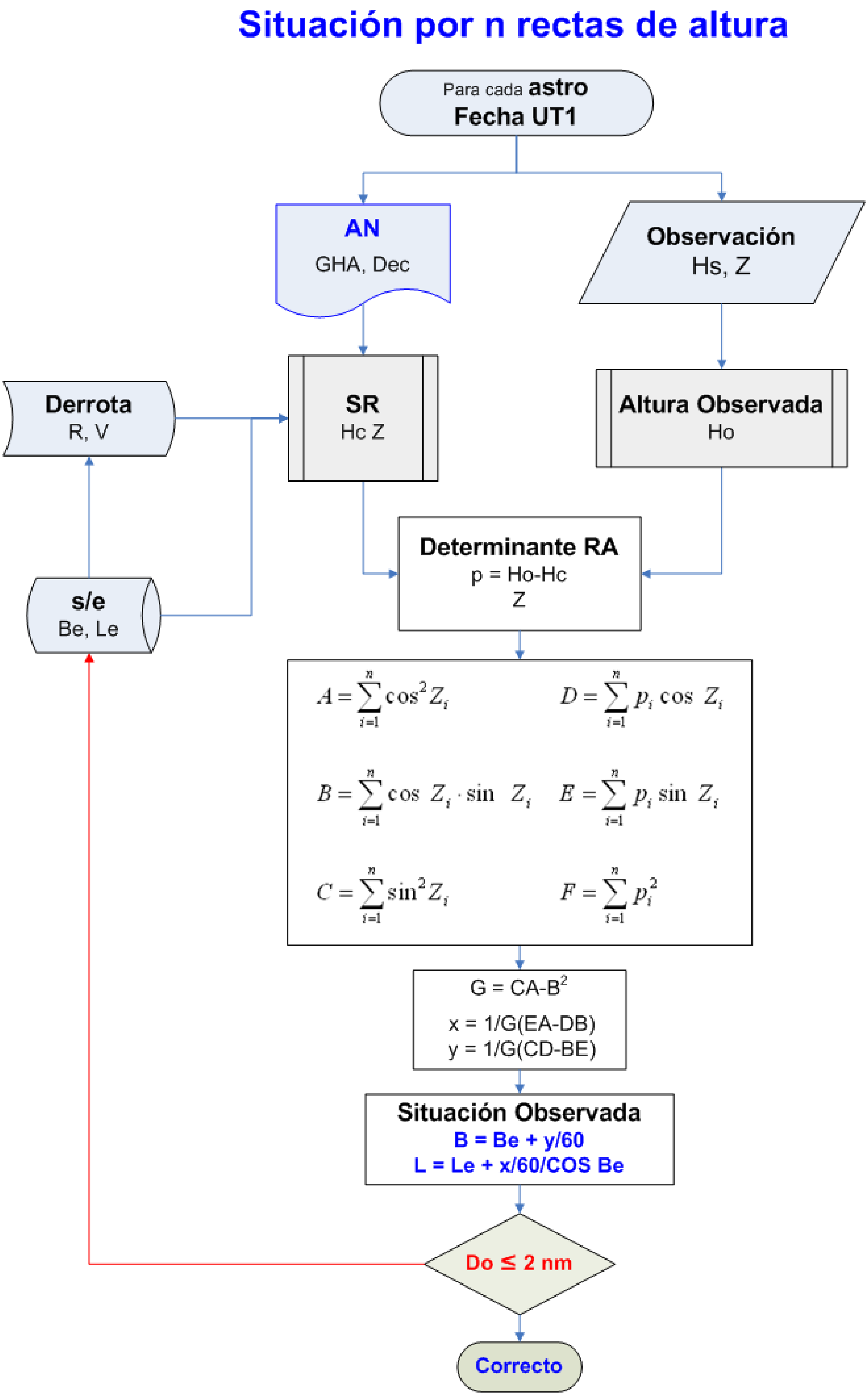
A1. Algorithms

Intercept method of sight reduction for the LoP

- Marcq Saint-Hilaire -

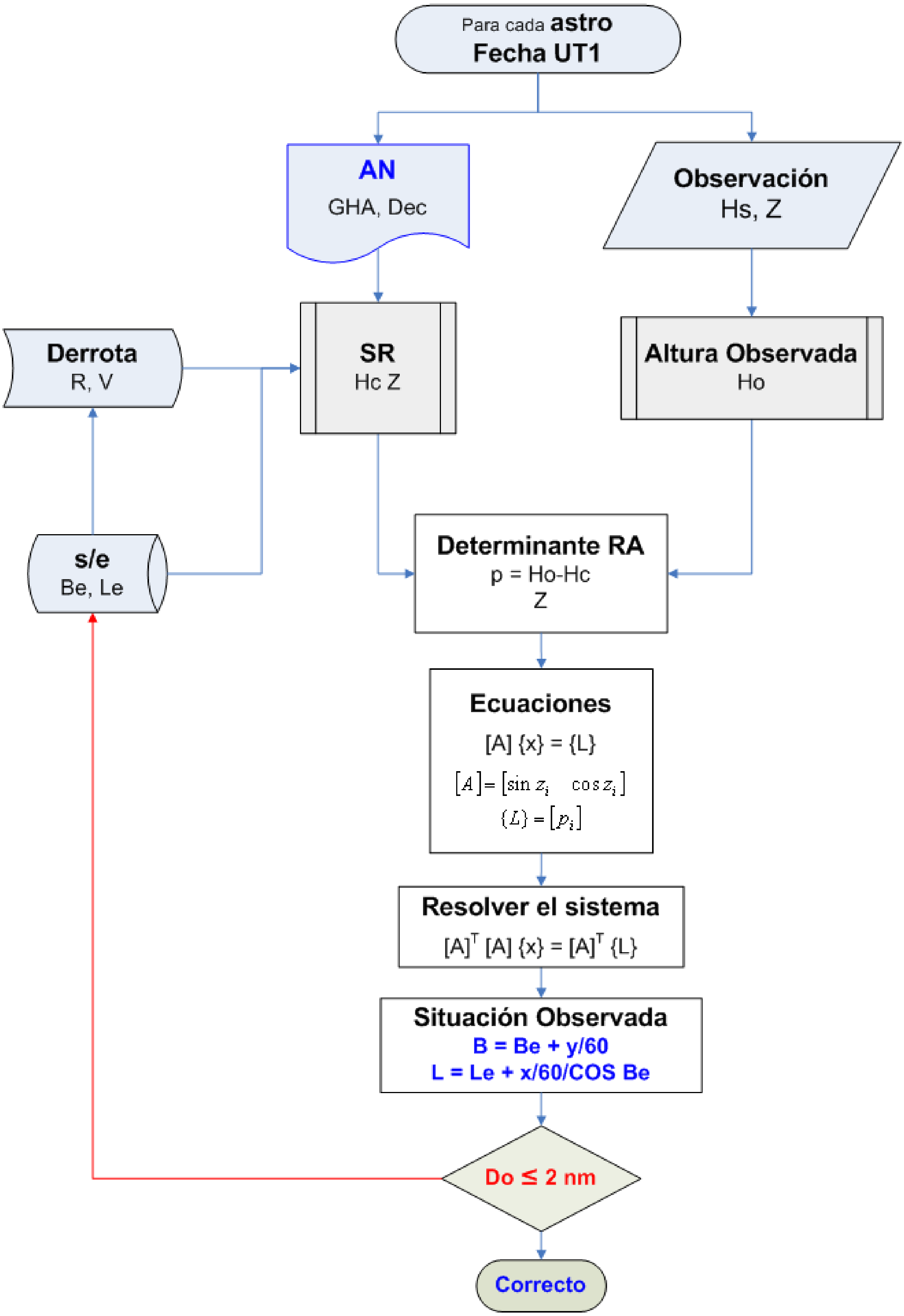


Position by n LoPs



Position by n LoPs - Matrix solution

Situación por n rectas de altura
Cálculo matricial



A2. Examples

Example in: Compact Data 2001-2005

UT	12:00:00	9-feb-2001
Be	32.75	
Le	-15.5	

Body	Time of Sight (UT)	Hs	GHA	DEC	HO	t = tObs - tv	R	V
Vega	6:58:52	49.585	324.9181	38.7813	49.4994	-5.0189	315	12
Spica	7:01:45	38.7067	43.5659	-11.1657	38.6146	-4.9708	315	12
Moon Lw	7:03:52	20.43	87.3397	13.4597	21.5462	-4.9356	315	12
Sun Lw	9:53:45	22.6733	324.8852	-14.5874	22.8353	-2.1042	315	12

1. Matrix solution by least squares method

Azimuth Z & intercept p

	Z	p = [L]
1	66.10	0.0923
2	217.41	-0.0855
3	272.69	-0.1117
4	126.19	0.0722

[A] $\begin{vmatrix} 0.9142 & 0.4052 \\ -0.6076 & -0.7943 \\ -0.9989 & 0.0468 \\ 0.8070 & -0.5905 \end{vmatrix}$

$[A]^T$ $\begin{vmatrix} 0.9142 & -0.6076 & -0.9989 & 0.8070 \\ 0.4052 & -0.7943 & 0.0468 & -0.5905 \end{vmatrix}$

$[Z]=[A]^T[A]$ $\begin{vmatrix} C & B \\ B & A \end{vmatrix} \quad \begin{vmatrix} 2.8540 & 0.3297 \\ 0.3297 & 1.1460 \end{vmatrix}$

$([A]^T[A])$-1 $\begin{vmatrix} 0.3624 & -0.1043 \\ -0.1043 & 0.9026 \end{vmatrix}$

$\{p\}=[A]^T$ [L] $\begin{vmatrix} E \\ D \end{vmatrix} \quad \begin{vmatrix} 0.3062 \\ 0.0574 \end{vmatrix}$

$[X] = ([A]^T[A])$-1 $[A]^T$ [L] $\begin{vmatrix} 0.1050 \\ 0.0199 \end{vmatrix}$

dB = y 0.0199
dL = x/cos(Bf) 0.1248

B = 32.7699
L = -15.3752

2. Solution by the program CelestialFix.exe

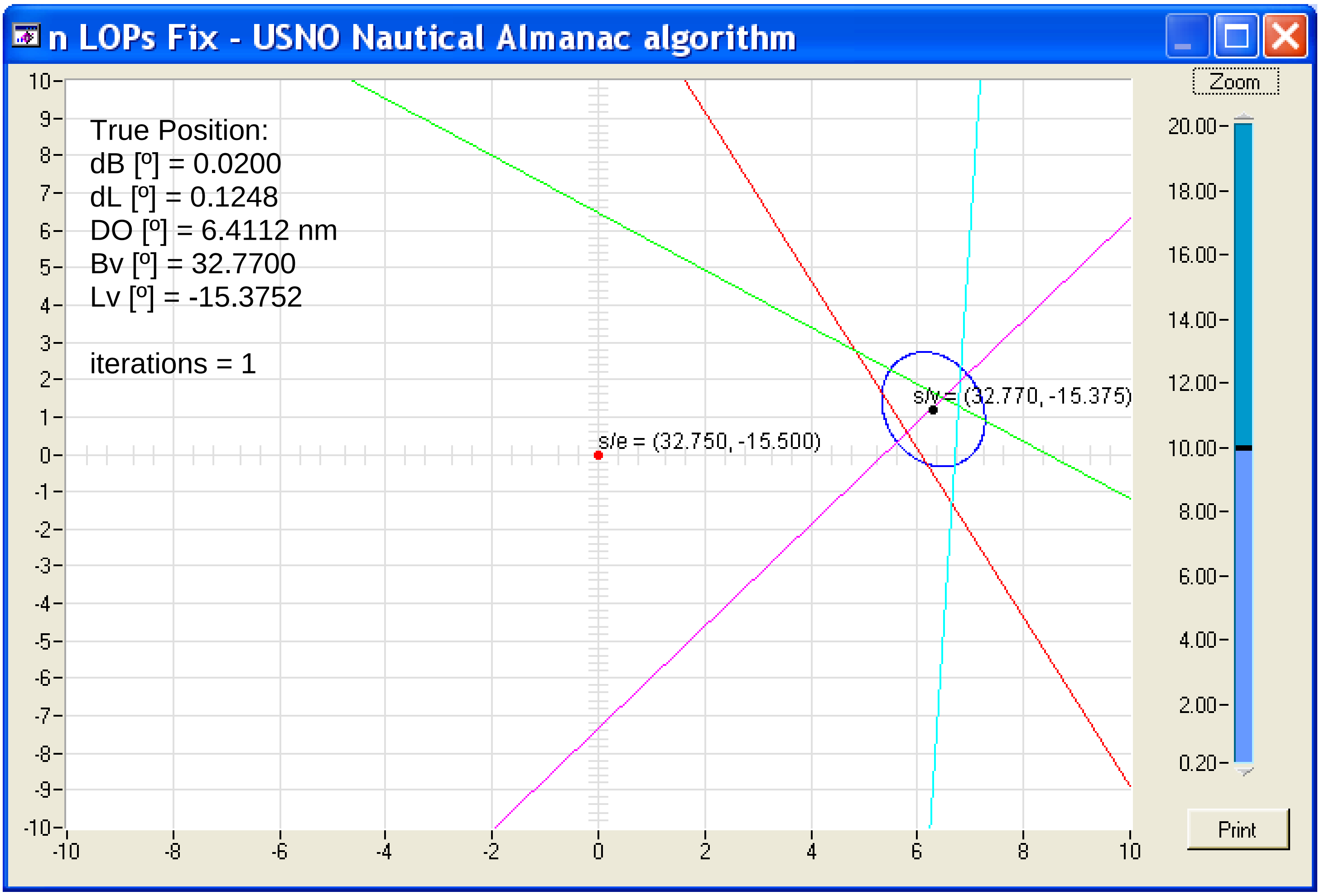
GHA	DEC	HO	BO	LO	LHA	HC	Z	p
324.9181	38.7813	49.4994	32.0402	-14.6561	310.2620	49.4071	66.0955	0.0923
43.5659	-11.1657	38.6146	32.0470	-14.6642	28.9017	38.7001	217.4136	-0.0855
87.3397	13.4597	21.5462	32.0520	-14.6701	72.6696	21.6579	272.6852	-0.1117
324.8852	-14.5874	22.8353	32.4524	-15.1462	309.7390	22.7631	126.1928	0.0722

Estimate position at time of fix:
Befix [deg] = 32.7500
Lefix [deg] = -15.5000

Least squares:
n = 4
A = 1.1460
B = 0.3297
C = 2.8540
D = 0.0575
E = 0.3062
F = 0.0335
G = 3.1619

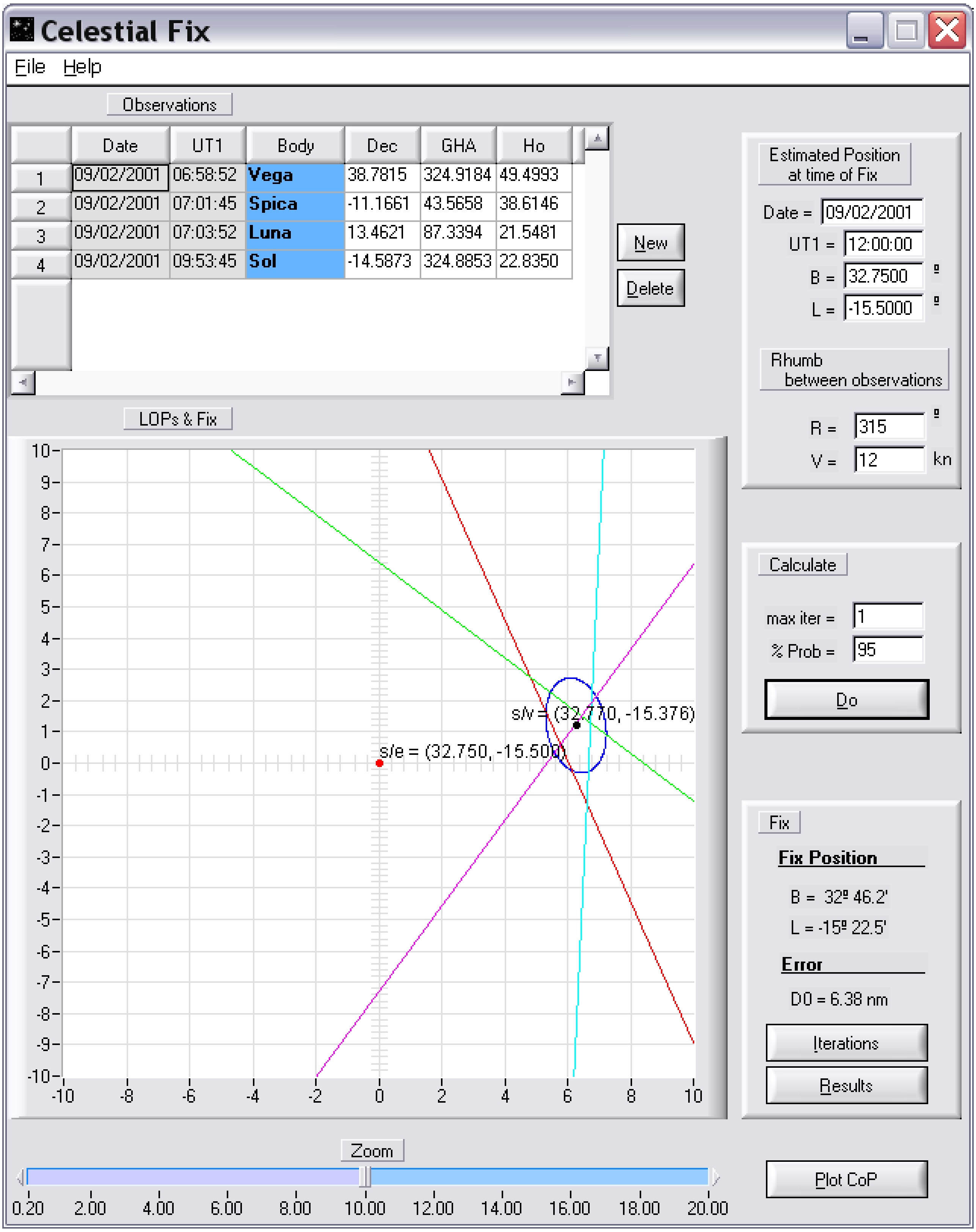
Error:
S = 0.0002
sigma = 0.6577 nm
sigmaB = 0.3959
sigmaL = 0.6248

Ellipse:
Prob = 0.9500
k = 2.4477
theta = -10.5536
a = 1.5458
b = 0.9428



A3. Software

Available at the Navigational Algorithms web site.



A4. Source code

As an illustrative example on how to implement the algorithms.

```
#include <stdio.h>
#include <math.h>
#include "mathlib.hpp"

FILE *fpOut = stdout;
FILE *fpIn = stdin;


inline void dato( const char *texto, double *variable )
{
 fprintf( fpOut, texto );
 fscanf( fpIn, "%lf", variable );
}


void main(void)
{
double Lat, Lon;
double GHA, D, LHA;
double Z, HC;
double X, G1, G2, SHA, D1, D2;
int body;
int limbo;
double Hs, IE, hOjo, HO;
double TC = 10;
double Pmb = 1010;
double sd, HP;
double AA, BB, CC, DD, EE, G;

 AA = 0; BB = 0; CC = 0; DD = 0; EE = 0;
otro: body = 0;

 fprintf( fpOut, "[0] Estrella \n" );
 fprintf( fpOut, "[1] Sol \n" );
 fprintf( fpOut, "[2] Luna \n" );
 fprintf( fpOut, "[3] Venus/Marte \n" );
 fprintf( fpOut, "[4] Jupiter/Saturno \n" );
 fprintf( fpOut, "cuerpo celeste = " );
 fscanf( fpIn, "%d", &body );
 fprintf( fpOut, "\n" );

 if( body == 1  ||  body == 2 ) {
                           fprintf( fpOut, "Limbo  [1]Inferior / [2]Superior: " );
                           fscanf( fpIn, "%d", &limbo );
                           }

 dato( "Lat [deg] = ", &Lat );
 dato( "Lon [deg] = ", &Lon );

 dato( "GMT [h] = ", &X );
 dato( "GHA en h   [deg] = ", &G1 );
 dato( "GHA en h+1 [deg] = ", &G2 );

 if( G2 < G1 ) G2 = G2+360.0;
 GHA = G1+X*(G2-G1);

 if( body == 0 ) {
                           dato( "SHA [deg] = ", &SHA );
                           GHA = GHA+SHA;
                           }

 while( GHA > 360.0 ) GHA = GHA-360.0;

 if( body == 0 ) {
                           dato( "Declinacion [deg] = ", &D );
                           }
 else {
                           dato( "Declinacion en h   [deg] = ", &D1 );
                           dato( "Declinacion en h+1 [deg] = ", &D2 );
                           D = D1+X*(D2-D1);
                           }
 fprintf( fpOut, "\n" );


 LHA = GHA+Lon;
 if( LHA > 360.0 ) LHA = LHA-360.0;
```

```
 if( LHA < 0.0 )   LHA = LHA+360.0;

 HC = ASIN( SIN( Lat )*SIN( D )+COS( Lat )*COS( D )*COS( LHA ) );

 Z  = ACOS( (SIN( D )-SIN( Lat )*SIN( HC ))/(COS( Lat )*COS( HC )) );
 if( LHA <= 180.0 ) Z = 360.0-Z;


 fprintf( fpOut, "LHA [deg] = %lf \n", LHA );
 fprintf( fpOut, "\n" );
 fprintf( fpOut, "HC [deg] = %lf \n", HC );
 fprintf( fpOut, "Azimut [deg] = %lf \n", Z );
 fprintf( fpOut, "\n" );

 dato( "Hs [deg] = ", &Hs );
 dato( "EI [deg] = ", &IE );
 dato( "hOjo [m] = ", &hOjo );
 dato( "T [Celsius] = ", &TC );
 dato( "P [mb] = ", &Pmb );

double dip, H, RO, F, R, PA, OB;

 dip = .0293*sqrt( hOjo );
 H  = Hs+IE- dip;
 RO = 0.0167/TAN( H+7.31/(H+4.4) );
 F  = 0.28*Pmb/(TC+273);
 R  = F*RO;

 PA = OB = sd = HP = 0;

 if( body == 1 ) {
                           HP = 0.0024;
                           dato( "SD [deg] = ", &sd );
                           }
 else if( body == 2 ) {
                           OB = -0.0017*COS( H );
                           dato( "HP [deg] = ", &HP );
                           sd = 0.2724*HP;
                           }

 if( body == 1  ||  body == 2 ) {
                           PA = HP*COS( H )+OB;
                           if( limbo == 2 ) sd = -sd;
                           }

 HO = H-R+PA+sd;

 fprintf( fpOut, "HO [deg] = %lf \n", HO );
 fprintf( fpOut, "\n" );

double P, PO;
double BI, LI, DO;

 P = HO-HC;
fprintf( fpOut, "mn ['] = %lf  +Towars/-Away\n", 60.0*P );

 AA = AA+SQ( COS( Z ) );
 BB = BB+COS( Z )*SIN( Z );
 CC = CC+SQ( SIN( Z ) );
 DD = DD+P*COS( Z );
 EE = EE+P*SIN( Z );

int YN;
 fprintf( fpOut, "Otro FIX [1/0]: " );
 fscanf( fpIn, "%d", &YN );
 if( YN == 1 ) goto otro;

 G = AA*CC-SQ( BB );
 BI = Lat+(CC*DD-BB*EE)/G;
 LI = Lon+(AA*EE-BB*DD)/(G*COS( Lat ));
 DO = 60.0*sqrt( SQ(LI-Lon)*SQ( COS( Lat ) )+SQ(BI-Lat) );
 DO =  int(DO*10.0)/10.0;

 fprintf( fpOut, "Lat [deg] = %lf \n", BI );
 fprintf( fpOut, "Lon [deg] = %lf \n", LI );
 fprintf( fpOut, "Dist [deg] = %lf \n", DO );
 fprintf( fpOut, "\n" );
}
```

A5. References

- Optimal Estimation of a Multi-Star Fix, C. De Wit. NAVIGATION, Vol.21 , No. 4, Winter 1974-75, pp 320-325
- Compact Data for Navigation and Astronomy for the Years 1991-1995, Cambridge University Press ISBN 0-521-38731-0. Yallop B. D. and Hohenkerk C. Y. (1991).
- Compact Data for Navigation and Astronomy for 1996 to 2000. ISBN 0-11-772467-X. Yallop B. D. and Hohenkerk C. Y. (1995), The Stationery Office.
- NavPac and Compact Data 2001–2005. ISBN 011-887311-3. HM Nautical Almanac Office. Stationery Office
- NavPac and Compact Data 2006–2010. ISBN 011-887331-8. HM Nautical Almanac Office. Stationery Office
- The Nautical Almanac. USNO
- Two celestial LOPs Fix. Andrés Ruiz. Navigational Algorithms
- Corrections for Sextant Altitude. Andrés Ruiz. Navigational Algorithms