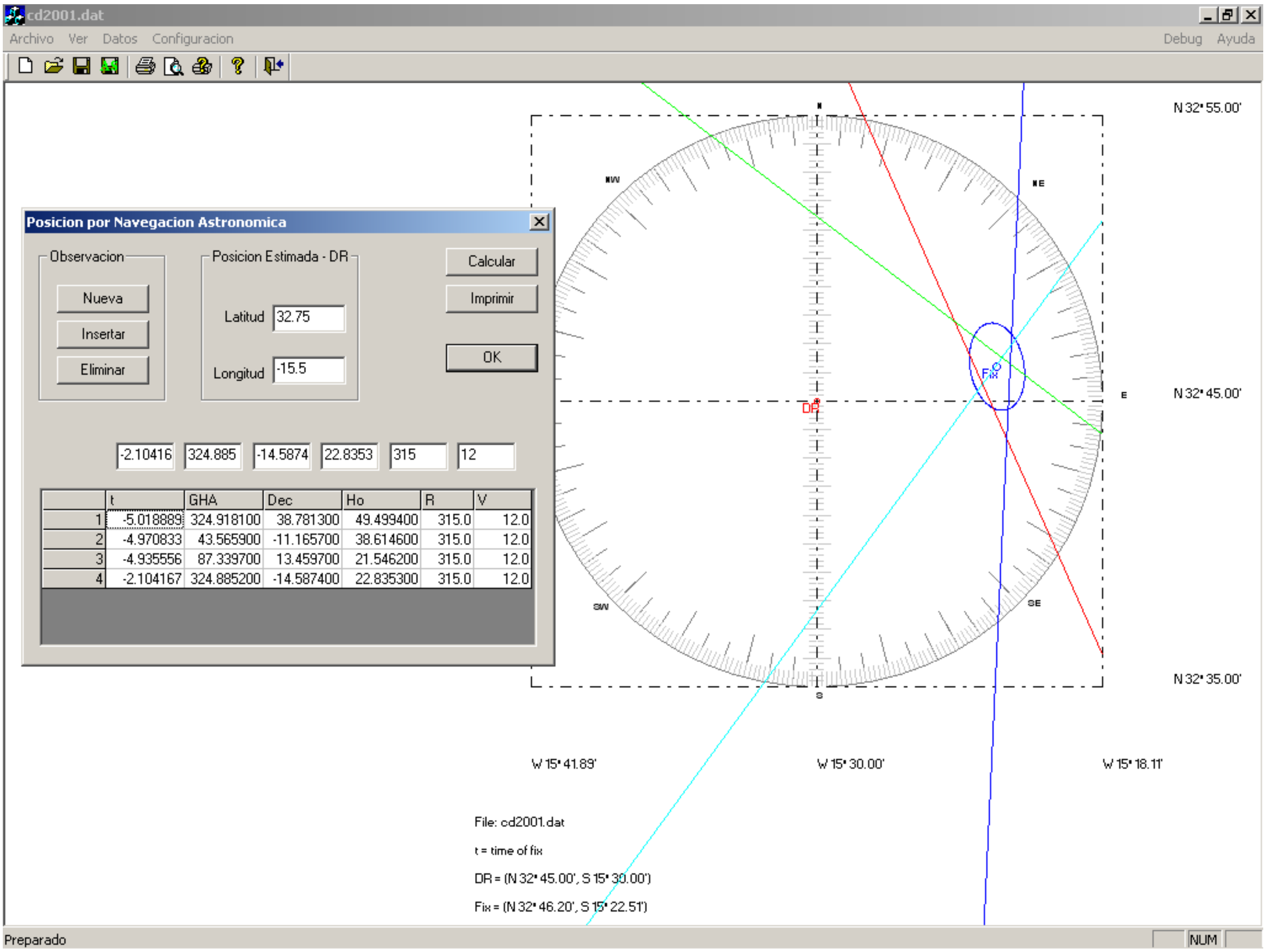
NAVIGATIONAL ALGORITHMS

Situación más probable por n Rectas de Altura

Método de los mínimos cuadrados



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

Indice

Resumen

El presente artículo describe un método analítico para calcular la situación por medio de la observación de astros como alternativa eficaz a los métodos gráficos tradicionalmente usados en navegación astronómica.

El algoritmo es totalmente general, permitiendo utilizar observaciones simultáneas o tomadas en distintos tiempos, entre los cuales se navega a un rumbo y una velocidad determinada. Utiliza el método de los mínimos cuadrados para obtener la posición más probable, y mediante iteraciones sucesivas permite reducir el error cometido al aproximar el círculo de altura por la recta de altura. Proporciona así mismo el error cometido en el cálculo de la posición y la elipse de confianza.

Navigational Algorithms
San Sebastián – Donostia
43º 19’N 002ºW

Versión actual: 201201

Se describe el algoritmo de C. De Wit *Optimal Estimation of a Multi-Star Fix*, oficialmente adoptado por la *HM Royal Navy*, del Reino Unido, para calcular la situación a partir de las observaciones de astros mediante el sextante. Se incluye también en *The Nautical Almanac* publicado anualmente por el Observatorio Naval de Estados Unidos.

Es una alternativa eficaz y robusta a los métodos gráficos tradicionales para obtener la situación por intersección de rectas de altura o de bisectrices de altura.

Este algoritmo sustituye el trazado de la recta de altura, **RA**, correspondiente a cada observación, su traslado si no son simultáneas y el cálculo de la posición por un cálculo que sistematiza este proceso y obtiene la posición más probable en base al método de los mínimos cuadrados.

Por la simplicidad de sus cálculos, sobre todo en forma matricial, es susceptible de ser empleado con una calculadora o una hoja de cálculo electrónica.

Variables

UT1 *Tiempo Universal, (Universal Time)*. Se puede aproximar por el Tiempo medio en Greenwich, (*Greenwich Mean Time).*

B Latitud
N (+) y S (–)

L Longitud
E (+) y W (–)

RA Ascensión Recta, *Right Ascension*

SHA Angulo Sidéreo, *Sidereal Hour Angle*
SHA = 360° - RA

GHA Angulo Horario en Greenwich. *Greenwich Hour Angle*
GHA = GHA(Aries) + SHA
0º <= GHA <= 360º de W a E

DEC Declinación en grados N (+) y S (–).

LHA Angulo Horario Local. *Local Hour Angle*.
En grados de W a E, de 0° a 360°

H Altura aparente.
Es la altura medida con el sextante corregida del error de índice, y de la altura de ojo del observador:
H = Hs + EI – Dip

Ho Altura observada.
Es la altura aparente corregida de refracción, paralaje y semidiámetro [2].

Hc Altura calculada

Z Azimut verdadero, medido en sentido horario alrededor del horizonte de 0° a 360°. Se define como el arco de horizonte entre el meridiano del lugar y el círculo vertical que pasa por el astro observado.

p Diferencia de Alturas o intercepto.
p = Ho – Hc

R Rumbo circular. (0° a 360°.)

V Velocidad del buque en nudos

n Número de observaciones

Determinante de la Recta de Altura

Se utiliza el método de **Marcq Saint Hilaire** para calcular el determinante de la recta de altura, (también llamado *Determinante tangente punto aproximado*), que relaciona la posición del observador y la del astro. Esta recta tangente al círculo de altura es perpendicular al azimut del astro.

Conociendo, aunque sea de forma aproximada, la latitud y la longitud, se puede calcular el azimut verdadero y la altura del astro sobre el horizonte, que comparada con la altura observada da la recta de altura buscada.

Las formulas empleadas son:

LHA = GHA + L

Hc = asin(sin B sin DEC + cos B cos Dec cos LHA)

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

SI(0 < LHA < 180º) Z = 360 – Z

El movimiento del buque. Observaciones No Simultáneas

El equivalente matemático al traslado de una RA en la carta náutica debido al movimiento del buque es como sigue.

Adoptando una posición de estima a la hora en que se quiere hallar la situación verdadera astronómica: Be, Le

La posición a la hora de la observación se calcula para un rumbo y una velocidad constante, (o su equivalente si ha habido cambios de rumbo o velocidad).

Si el intervalo de tiempo es:

$$t = UT1_{observacion} - UT1_e$$

La posición a emplear para el cálculo de la altura y el azimut es:

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

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

Cálculo de la Posición

Para poder dibujar la recta de altura correspondiente a una observación, es necesario conocer su azimut y su diferencia de alturas asociada.

$$p = Ho - Hc$$

- Si p es positivo, la recta de altura se dibuja a lo largo del azimut desde la posición estimada.
- Si p es negativo, la recta de altura se dibuja en dirección opuesta al azimut, (Z+180º), desde la posición estimada

Se necesitan dos o más líneas de posición para obtener la situación verdadera observada.

Este algoritmo usa el *método de los mínimos cuadrados* para determinar la situación a partir de tres o más observaciones.

Si p_i y Z_i, (i=1,n), son la diferencia de alturas y el azimut para la observación i:

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

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

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

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

Como comprobación, se debe cumplir que A+C = n

Calculados los coeficientes anteriores, la situación estimada queda mejorada en dB y dL, obteniéndose así la situación verdadera:

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

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

La distancia entre la posición estimada y la posición mejorada, en millas náuticas, es:

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

Si Do < 20 mn se necesita otra iteración que permita volver a mejorar la posición obtenida hasta que la solución converja, sustituyendo la posición estimada por esta ultima obtenida:

$$B_e = B_v$$

$$L_e = L_v$$

Es posible, pero no aconsejable, empezar el proceso iterativo con una posición situada en un hemisferio diferente. Manteniendo B en el intervalo (-90º, +90º) y L dentro de (-180º, +180º), la solución converge después de unas pocas iteraciones.

Error Estimado en la Posición

Si se obtienen tres o más RA, se puede calcular una estimación del error en posición.

La desviación standard de la posición calculada en millas náuticas viene dada por:

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

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

Y las desviaciones standard en latitud y longitud son:

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

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

En general a medida que el número de observaciones aumenta, decrece el error en la posición obtenida.

La elipse de error proporciona un medio de expresar el nivel de confianza de un conjunto de datos ajustados. Por ejemplo, una elipse de error de un nivel del 95% de confianza define un límite alrededor de una posición calculada en la que existe un 95% de posibilidades de que contenga la situación real.

La elipse de confianza de ejes (a,b) es:

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

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

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

Donde el factor de escala es:

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

Para un nivel del 95%, Prob = 0.95

La forma de la elipse de confianza depende únicamente del número n de observaciones y de la distribución de estas en azimut, mientras que su tamaño es proporcional además de a k, a los errores en la observación. El método asume que *todas las observaciones tienen igual peso*. La situación ideal se da al tener una distribución circular de errores, A=C, B=0, en donde el error es el mismo en todas las direcciones, situación que se alcanza cuando las observaciones están igualmente espaciadas en azimut. Si además los astros observados tienen alturas similares, se minimiza los errores sistemáticos en la obtención de la posición.

Dibujo de las Rectas de Altura y de la elipse de confianza.

Tomando un sistema de ejes cartesianos, es posible dibujar los distintos elementos que definen la situación astronómica.

- Origen: la posición estimada.
- Eje X: según un paralelo, positivo hacia el Este.
- Eje Y: según un meridiano, positivo hacia el Norte.

Rectas de Altura

Situando el dibujo dentro de un cuadrado de lado 20 millas centrado en la posición de estima, la RA definida por p y Z queda determinada por la intersección con los lados de dicho cuadrado:

- X = ±10
- Y = p ± 10 sin Z/cos Z
- La RA cruza los lados del cuadrado si -10<=Y<=+10

- Y = ±10
- X = p ± 10 cos Z/sin Z
- La RA cruza las bases del cuadrado si -10<=X<=+10

Elipse

Dando valores a α entre 0º y 360º se obtienen los puntos de la elipse de confianza centrado en B_e, L_e:

x = a cos α sin θ – b sin α cos θ + 60 dL cos B_e

y = a cos α cos θ + b sin α sin θ + 60 dB

Situación

A cada iteración el origen se escoge en la posición obtenida en el paso anterior.

Fundamento matemático

La ecuación en el plano cartesiano de la recta de altura, alrededor de la posición verdadera es [1]:

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

Para n observaciones la posición más probable se obtiene de aplicar el método de los mínimos cuadrados a la intersección de las n líneas de posición. Es decir optimizando la distancia entre la situación verdadera y las rectas de altura.

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

Minimizando la función S:

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

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

Resulta el siguiente sistema de ecuaciones, de cuya resolución se obtiene la posición más probable.

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

[Z] {x} = {P}

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

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

Cuya solución explicita es:

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

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

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

$B = B_e + y/60$

$L = L_e + x/60/\text{COS } B_e$

Solución matricial

Utilizando el cálculo matricial, se simplifica notablemente:

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

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

[A] {x} = {L}

Sistema sobredeterminado con 2 incógnitas y n ecuaciones. Se demuestra que la solución por mínimos cuadrados resulta de resolver el siguiente sistema:

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

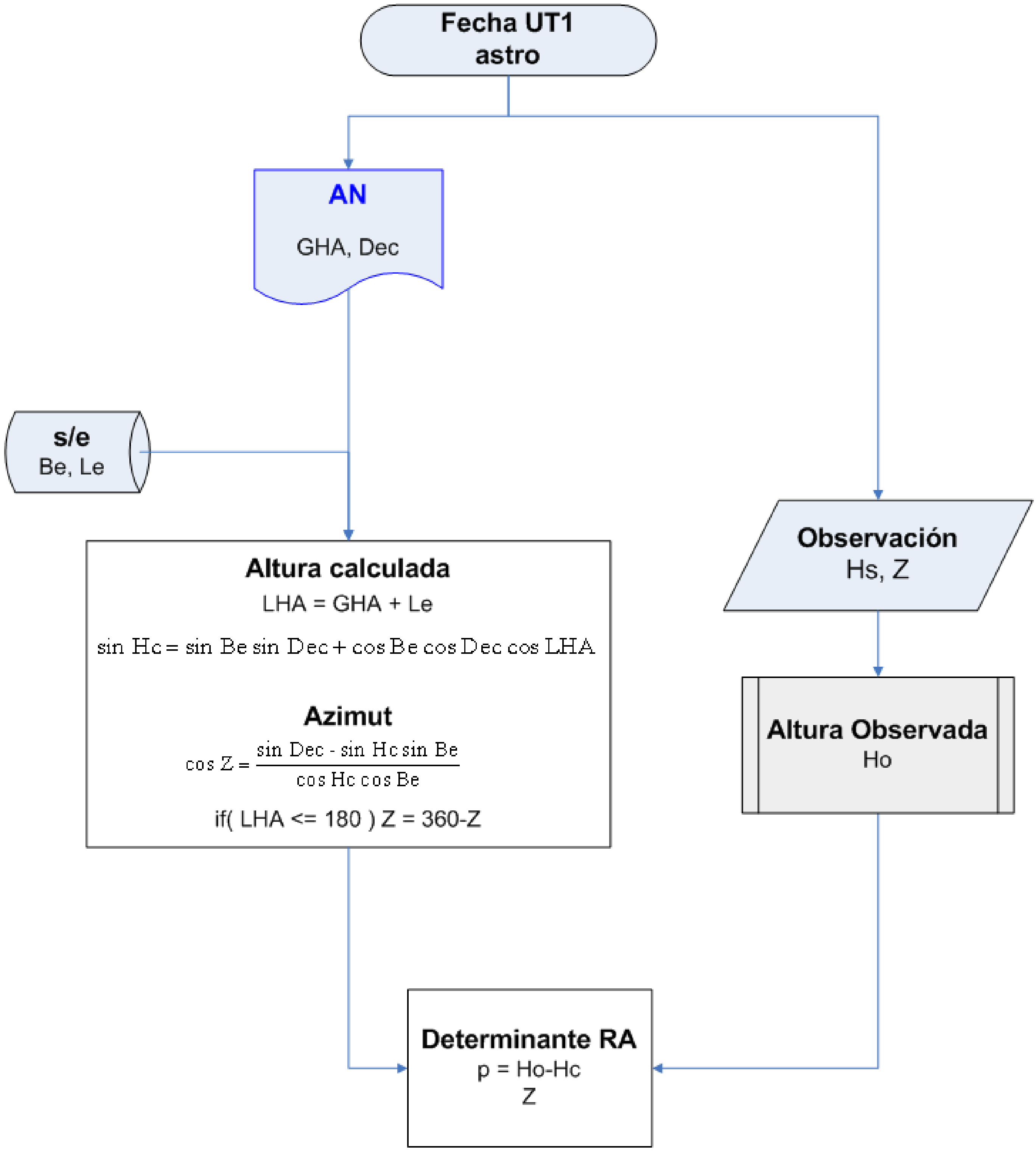
Donde con la nomenclatura anterior:

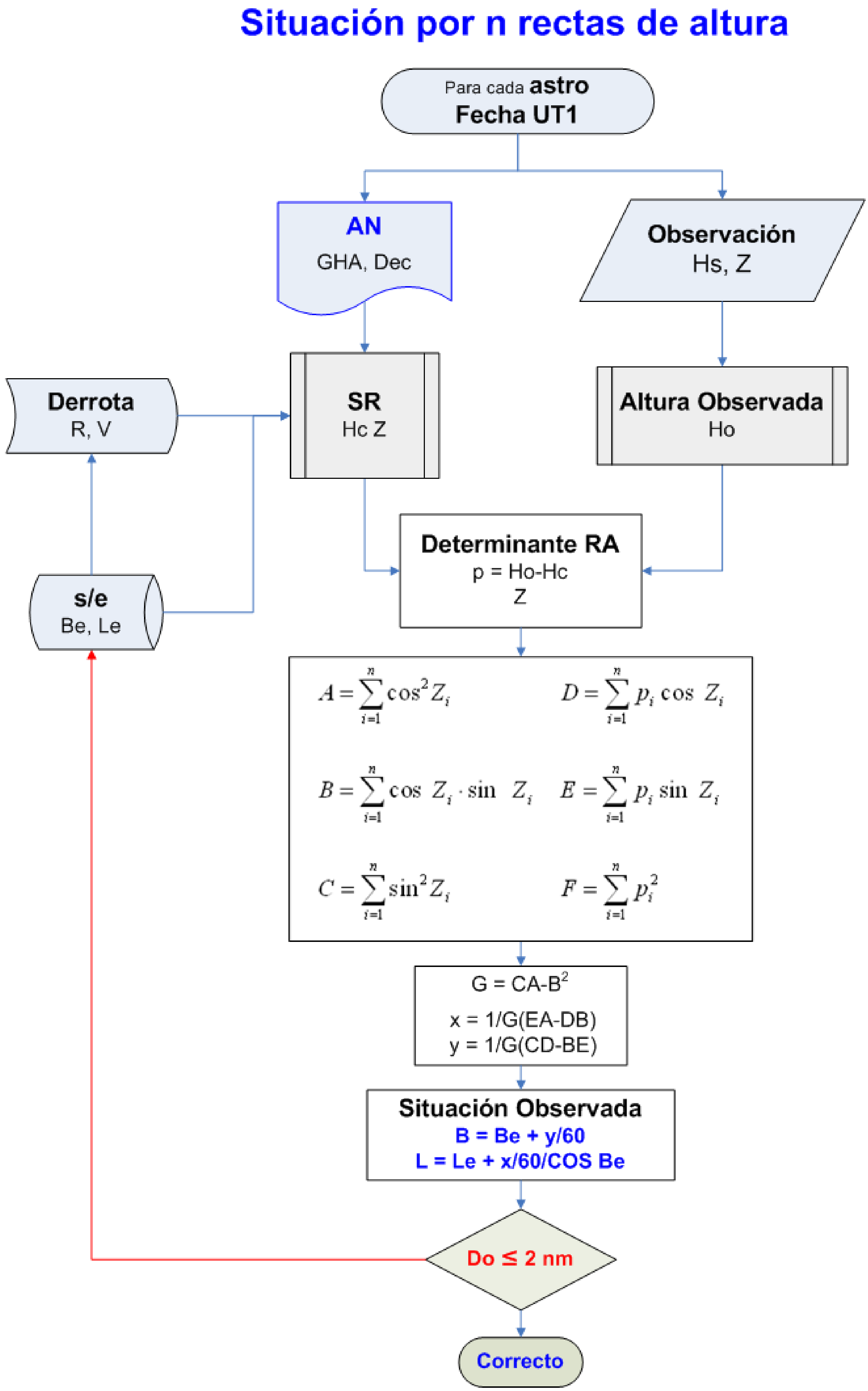
$[Z] = [A]^T [A]$

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

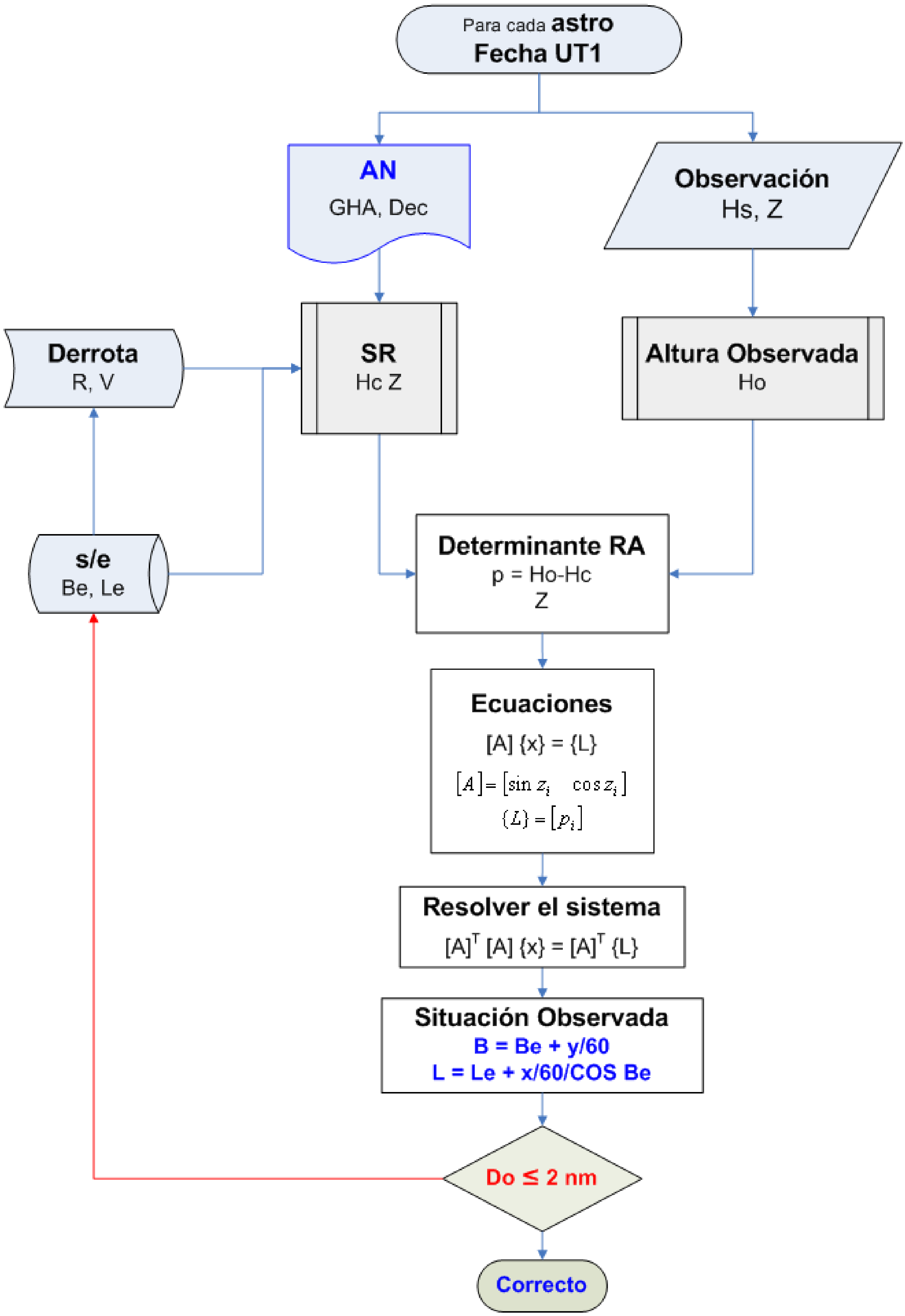
A1. Algoritmo

Determinante Marcq Saint-Hilaire de la recta de altura




Situación por n rectas de altura
Para cada astro
Fecha UT1
AN
GHA, Dec
Observación
Hs, Z
Derrota
R, V
SR
Hc Z
Altura Observada
Ho
Determinante RA
p = Ho-Hc
Z
s/e
Be, Le
$A=\sum_{i=1}^{n}\cos^2 Z_i$
$D=\sum_{i=1}^{n}p_i\cos Z_i$
$B=\sum_{i=1}^{n}\cos Z_i\cdot\sin Z_i$
$E=\sum_{i=1}^{n}p_i\sin Z_i$
$C=\sum_{i=1}^{n}\sin^2 Z_i$
$F=\sum_{i=1}^{n}p_i^2$
$G = CA-B^2$
x = 1/G(EA-DB)
y = 1/G(CD-BE)
Situación Observada
B = Be + y/60
L = Le + x/60/COS Be
Do ≤ 2 nm
Correcto

Situación por n rectas de altura
Cálculo matricial



A2. Ejemplos

Ejemplo 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. Solución Matricial por Mínimos Cuadrados

Azimut Z y diferencia de alturas 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]
$$\begin{vmatrix} 0.9142 & 0.4052 \\ -0.6076 & -0.7943 \\ -0.9989 & 0.0468 \\ 0.8070 & -0.5905 \end{vmatrix}$$

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

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

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

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

$[X] = ([A]^T[A])$-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. Solución empleando el programa SR_LS.exe

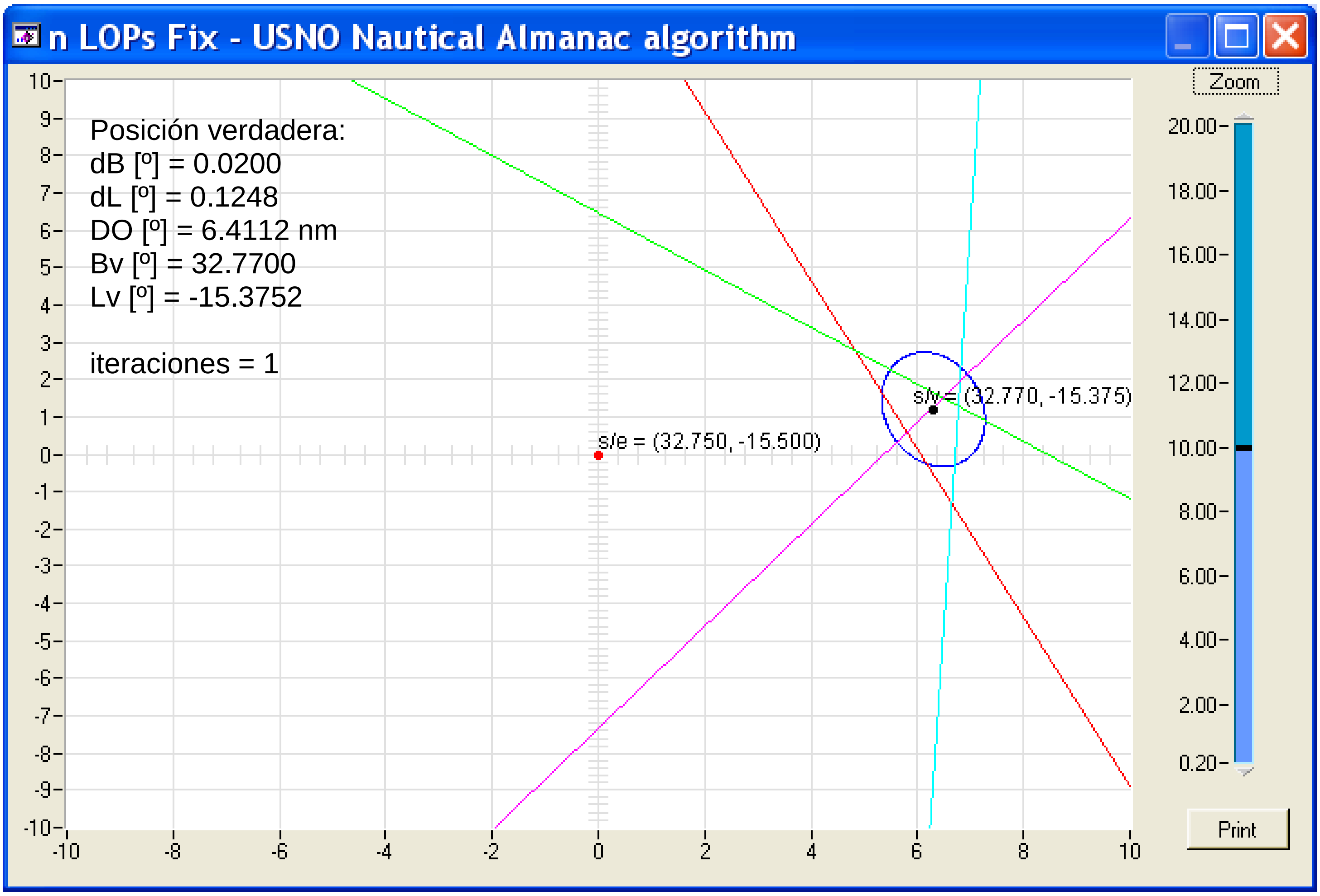
GHA	DEC	HO	BO	LO	LHA	HC	Z	p
324.9181	38.7813	49.4994	32.0402	-14.6561	310.2620	49.4071	66.0955	0.0923
43.5659	-11.1657	38.6146	32.0470	-14.6642	28.9017	38.7001	217.4136	-0.0855
87.3397	13.4597	21.5462	32.0520	-14.6701	72.6696	21.6579	272.6852	-0.1117
324.8852	-14.5874	22.8353	32.4524	-15.1462	309.7390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

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

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

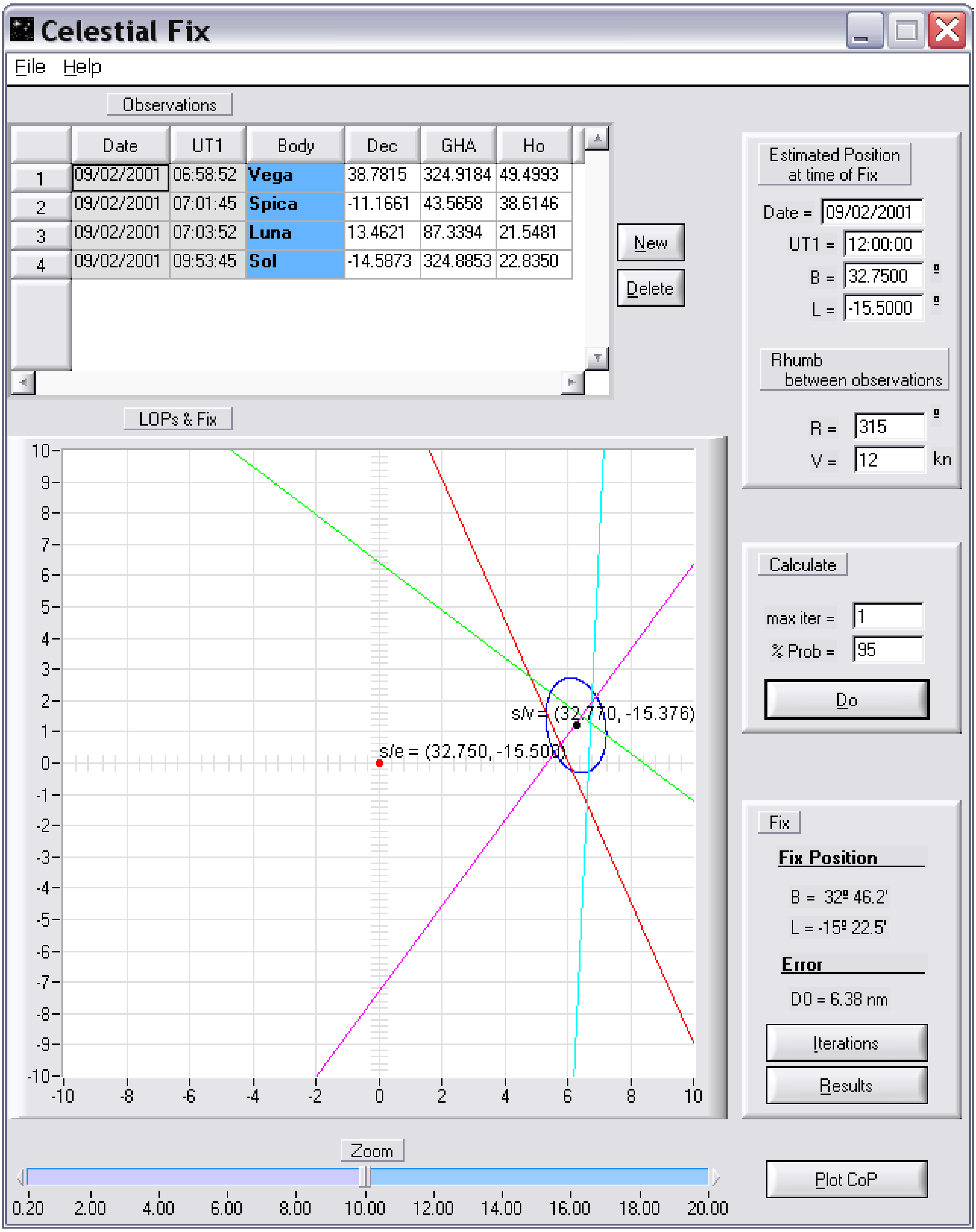
Error:
S = 0.0002
sigma = 0.6577 nm
sigmaB = 0.3959
sigmaL = 0.6248

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



A3. Software

Disponible en la Web de Navigational Algorithms.



A4. Código fuente

Se incluye este programa a modo ilustrativo de cómo implementar el método:

```
#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. Referencias

- 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