



Association *positionnelle*  
*probabiliste* de catalogues de sources

Version 1.0, 2012-9-17

Michel Fioc

Michel.Fioc@iap.fr

Institut d'astrophysique de Paris,  
université Pierre et Marie Curie,  
France

## I. Introduction

The code ASPECTS<sup>∗1</sup> is an implementation in Fortran 95 of the relations established in Fioc (2012).

“ASPECTS” is a French acronym for “*Association positionnelle/probabiliste de catalogues de sources*” (“Probabilistic/positional association of catalogs of sources” in English). The French word “*Aspect*” (pluralized in the acronym) has the same general meaning as the corresponding English word; interestingly, it signifies in particular the relative position of heavenly bodies. . .

## II. Access to the code

The source files of version 1.0 of ASPECTS are freely available at

<http://www2.iap.fr/users/fioc/Aspects/>

in tar file `<Aspects_v1.0.tar>`. Type “`tar xvf Aspects_v1.0.tar`” to extract them in `<Aspects_v1.0/>`.

### Warning on Numerical Recipes routines

To run the code, the user will need the following non-free Numerical Recipes routines in Fortran 90 (Press *et al.*, 1992):

<code>nrtype</code>	<code>locate</code>	<code>ran_state</code>
<code>nr</code>	<code>indexx</code>	<code>ran1</code>
<code>nrutil</code>	<code>sort_heap</code>	<code>gasdev</code>

We can unfortunately not provide them. . .

## III. Purpose of ASPECTS

Given two catalogs  $K$  and  $K'$  of  $n$  and  $n'$  astrophysical sources, respectively, ASPECTS computes, for any objects  $M_i \in K$  and  $M'_j \in K'$ , the probability that  $M'_j$  is a *counterpart* of  $M_i$ , i.e. that they are the same source. To determine this *probability of association*, denoted  $P(A_{i,j} | C \cap C')$  in Fioc (2012), the code takes into account the coordinates and the positional uncertainties of all the objects. ASPECTS also computes the probability  $P(A_{i,0} | C \cap C')$  that  $M_i$  has no counterpart.

Three different models of association are considered:

**Several-to-one** (“s:o”): a  $K$ -source has at most one counterpart in  $K'$ , but a  $K'$ -source may have several counterparts in  $K$ ;

**One-to-several** (“o:s”): the converse of several-to-one ( $K$  and  $K'$  are swapped);

**One-to-one** (“o:o”): a  $K$ -source has at most one counterpart in  $K'$  and vice versa.

While the last case is the most natural one, the other two may also occur. For instance, if the spatial resolution is much lower in  $K'$  than in  $K$ , several distinct  $K$ -sources may be merged in  $K'$ . On the opposite, if objects are more patchy at the wavelength of  $K'$  than at that of  $K$ , a single extended  $K$ -source may look as several objects in  $K'$ . Additional reasons for making several-to-one and one-to-several computations are that they are much

1. Pronounced [aspɛ] in International Phonetic Alphabet transcription, not [æspekts], and preferentially written “Aspects” if small capitals are not available.

simpler than one-to-one computations and that they may provide better estimates of unknown parameters for one-to-one associations. . .

ASPECTS also computes the *likelihood*  $L$  to observe the objects at their registered positions in these three cases, and a maximum-likelihood estimate of the *fraction of sources with a counterpart*.

## IV. Contents of the `<Aspects_v1.0/>` directory

A large variety of cases may be considered, depending on whether the positional uncertainties are known in both catalogs, only one or none. If not fully known, they may be modelled with some additional parameters (e.g.  $\hat{\sigma}$  and  $\hat{\nu}$ ; cf. Appendix A in Fioc (2012)), to be determined by likelihood maximization; galaxy positional uncertainties, for instance, might be expressed as a function of the object’s size.

It would be impractical to cover all the cases in a single general-purpose main program. Therefore, besides the major module, `<mod_Aspects.f90>`, and a series of minor modules, we provide two main codes as examples: `<example_simul.f90>` and `<example_read.f90>` (cf. Sect. VI). These programs assume that positional uncertainties are known, but extensive comments should make them easy to adapt to other cases (cf. Sect. VII).

The complete list of files in `<Aspects_v1.0/>` is the following:

Makefile	<code>mod_Aspects.f90</code>	<code>K_u.dat</code>
<code>example_simul.f90</code>	<code>mod_simul_catalogs.f90</code>	<code>K_p.dat</code>
<code>example_read.f90</code>	<code>mod_types.f90</code>	<code>mod_files.f90</code>
<code>Read_me.pdf</code>	<code>mod_constants.f90</code>	
<code>1209.5361v1.pdf</code>	<code>mod_variables.f90</code>	

## V. Compilation

You first need to edit file `<Makefile>`: set variable `<NR_dir>` on line 6 to the location of the directory containing the Numerical Recipes routines. If the Numerical Recipes library is installed on your computer, you may remove all the lines containing `"$(NR_dir)/*.f90 \"` and call instead the library.

The default compiler is `<ifort>`. To choose instead `<gfortran>`, comment line 8 (with a `"#"` before `"f95_compiler"`) and uncomment line 9. The code is written in standard Fortran 95, so other compilers should also work. Note that none of the options of compilation (set by variable `<f95_options>`) is mandatory: they provide various checks during the compilation and the execution, but might actually slow down the latter.

To compile the code, just type `"make"` on the keyboard. This will produce two executables: `<example_simul>` and `<example_read>`. To recompile from scratch, type first `"make clean"`: this will delete the executables and the `<*.mod>` files.

## VI. Executables

**Program `<example_simul>`:** This executable calls subroutine `<simul_catalogs>` and creates all-sky mock catalogs for a one-to-one association model and a given value of the fraction  $f$  of  $K$ -sources with a counterpart. It then determines estimates of  $f$  and of the fraction  $f'$  of  $K'$ -sources with a counterpart for various models of association, and computes the maximum likelihood in each case.

It also provides the following recommendations on model selection, the best estimate of  $f$  or  $f'$  and the computation of association probabilities:

- if  $\hat{f}_{s:o}/n' \approx \hat{f}'_{o:s}/n$ , use one-to-one probabilities (subroutine `<comput_oto_prob>`) with  $f$  ( $= \langle f_1 \rangle = \hat{f}_{o:o}^*$  (not  $\hat{f}_{o:o}$ ) and  $f'$  ( $= \langle f_2 \rangle = \hat{f}_{o:o}^* := \hat{f}_{o:o}^* n/n'$ );
- else,
  - if  $\hat{f}_{s:o}/n' > \hat{f}'_{o:s}/n$ , use several-to-one probabilities (subroutine `<comput_sto_prob>`) with  $f$  ( $= \langle f_1 \rangle = \hat{f}_{s:o}$ );
  - if  $\hat{f}_{s:o}/n' < \hat{f}'_{o:s}/n$ , use one-to-several probabilities (subroutine `<comput_ots_prob>`) with  $f'$  ( $= \langle f_2 \rangle = \hat{f}'_{o:s}$ ).

**Program `<example_read>`:** Instead of simulating catalogs, this executable reads  $K$  and  $K'$  in files `<K_u.dat>` and `<K_p.dat>`, respectively (they have actually been produced with `<example_simul>`). It then makes the same computations as `<example_simul>` and outputs the several-to-one, one-to-several and one-to-one association probabilities in files `<sto_prob.dat>`, `<ots_prob.dat>` and `<oto_prob.dat>`, respectively.

## VII. Understanding and adapting the code

### VII.1. Notations

A brief summary of the notations used in the code is given in the header of file `<mod_variables.f90>` (the module it contains provides access to the variables used by both `<example_simul>` and `<example_read>`). We recall them here:

- Quantities related to catalog  $K$  are written with a suffix `<_u>` (for “unprimed”) in most files; those related to  $K'$ , with a suffix `<_p>` (for “prime”). In file `<mod_Aspects.f90>`, subscripts `<_1>` and `<_2>` are used instead because  $K$  and  $K'$  may be swapped;
- Most notations explicitly refer to those used in Fioc (2012): for instance, `<alpha_u(i)>` is the right ascension  $\alpha_i$  of  $M_i$ , `<beta_p(j)>` is the position angle of the positional uncertainty ellipse (called “positional uncertainty angle” hereafter)  $\beta'_j$  of  $M'_j$ , etc.

Semi-major and semi-minor axes of the positional uncertainty ellipse are an exception: to make text searches easier,  $a$  is noted `<sigma_major_u>`,  $b'$  is noted `<sigma_minor_p>`, etc. Another exception is the surface area covered by  $K$  and  $K'$ : it is noted  $S$  in the paper, but `<area>` in the code;

- In comments and outputs to the terminal, code variables and Fortran expressions are written between grave accents (`<`>`); mathematical expressions and variables defined in the article are noted between dollars in pseudo- $\text{\TeX}$ ;
- The word “neighbor” (`<ngb>`) designates a nearby object in the same catalog; “counterparts” (`<ctp>`), a nearby object in the other catalog;
- The following notations are used to distinguish the association models:

`<sto>` := “several-to-one”;    `<ots>` := “one-to-several”;    `<oto>` := “one-to-one”.

### VII.2. Angles

- All plane angles (right ascensions, declinations, semi-major and semi-minor axes and positional uncertainty angles) are in radians.
- Right ascensions are in  $[0, 2\pi[$ . Sources must be ordered by increasing right ascension; this is done automatically when calling subroutine `<simul_catalogs>` (as in `<example_simul>`), but you have to do it yourself if you read catalogs (as in `<example_read>`).
- The distribution of the observed position of a source relative to its true position is described by a 2-dimensional normal (i.e. Gaussian) law<sup>\*2</sup>. The semi-major (resp. semi-minor) axis is the  $1\text{-}\sigma$  positional uncertainty along the major (resp. minor) axis, independently from the position along the minor (resp. major) axis. So, if  $a, b, X$  and  $Y$  are, respectively, the semi-major and semi-minor axes and the positional errors along each axis, then

$$P((X, Y) \in [x, x + dx[ \times [y, y + dy[) = \frac{1}{2\pi ab} \exp\left(-\frac{1}{2} \left[ \frac{x^2}{a^2} + \frac{y^2}{b^2} \right]\right) dx dy.$$

- The positional uncertainty angle from the North to the semi-major axis is counted eastward and is in  $[0, \pi[$ .

### VII.3. Surface area

Both catalogs must cover exactly the same surface. The area of this surface is in steradians; it is automatically set to  $4\pi$  (all sky) by `<simul_catalogs>`, but you have to give its value otherwise. Parameter `<area>` is used to compute `<xi0_u(i)>` :=  $\xi_{i,0}$  and `<xi0_p(j)>` :=  $\xi_{0,j}$  for all sources  $M_i$  and  $M'_j$ ; in the case of uniform random distributions for unrelated  $K$ - and  $K'$ -sources, `<xi0_u(i)>` = `<xi0_p(j)>` =  $1/\text{<area>}$ .

### VII.4. Simulations

Simulations are run with subroutine `<simul_catalogs>` (file `<mod_simul_catalogs.f90>`). Besides the number of objects in  $K$  and  $K'$  (variables `<n_u>` :=  $n$  and `<n_p>` :=  $n'$ ) and the semi-major and semi-minor axes, two parameters are required:

---

2. The sphere is locally assimilated to its tangent plane or to a tangent cylinder. Note that this would not be appropriate if the semi-major axis were not negligible with respect to  $\pi$  radians. A Kent distribution (Kent, 1982), for instance, would then seem a better choice, as it is one possible generalization of the normal law to the sphere, but, contrary to a Gaussian, it is not stable: the sum of independent Kent-distributed random variables does not follow a distribution of the same type.

- the type of simulation, `<one_to_one>`. Set it to `<.true.>` to simulate one-to-one associations and to `<.false.>` for several-to-one associations;
- the simulated fraction of  $K$ -sources with a counterpart in  $K'$ , `<input_f_u>`. Note that because the number of sources with a counterpart must be an integer, the effective fraction, `<eff_f_u>`, may be slightly different. If  $n > n'$ , `<input_f_u>` must be in  $[0, n'/n]$ ; else, `<input_f_u>`  $\in [0, 1]$ .

Subroutine `<simul_catalogs>` randomly and uniformly draws in  $[0, \pi[$  the true positional uncertainty angles (i.e. with respect to the direction of the North at the true position) and convert them to observed angles <sup>\*3</sup> in subroutine `<position>` (cf. Appendix A).

## VII.5. Code constants

The default type used for reals (except in Numerical Recipes routines) is given by parameter `<real_type>`, which is defined in `<mod_types.f90>`.

Following constants, which control the accuracy of results, are defined and commented in `<mod_constants.f90>`:

- `<n_max_sigma>`**: Value of  $\langle R\_ctp \rangle / \max_{M_i \in K, M'_j \in K'}^{1/2} (a_i^2 + a'_j{}^2)$ , where `<R_ctp>` is the radius within which counterparts are searched. Reasonable values of `<n_max_sigma>` should be in  $[5, 10]$ .  
Default value: 5;
- `<R_ngb_ctp_ratio>`**: Value of  $\langle R\_ngb \rangle / \langle R\_ctp \rangle$ , where `<R_ngb>` is the radius within which neighbors are searched in one-to-one calculations. An `<R_ngb>` equal to a few times `<R_ctp>` seems reasonable.  
Default value: 2;
- `<n_max_ngb>`**: Maximal number of neighbors (including the main source) considered for one-to-one calculations <sup>\*4</sup>. This number must be  $\geq 1$ ; a value of 1 is equivalent to several-to-one calculations (no other neighbor than the main source itself).  
Default value: 10;
- `<min_xi_ratio>`**: Minimum value of  $\xi_{i,j} / \xi_{i,0}$ , where  $\xi_{i,j}$  is the spatial probability density to observe source  $M'_j$  at some position if it is a true counterpart of object  $M_i$ , and  $\xi_{i,0}$  is the same quantity if they are unrelated. This ratio should be  $\ll 1$ : if  $\xi_{i,j} / \xi_{i,0} < \langle \text{min\_xi\_ratio} \rangle$ ,  $\xi_{i,j}$  is set to 0 and  $M'_j$  is not considered as a possible counterpart of  $M_i$ ; if `<min_xi_ratio>`  $\leq 0$ , this never happens.  
Default value:  $10^{-10}$ ;
- `<epsilon_f>`**: Accuracy of maximum-likelihood estimates of  $f$  and  $f'$ : an estimate  $\hat{f}$  of  $f$  is the limit of a sequence  $f_{k+1} = g(f_k)$ , where  $g$  is some function; when  $|f_{k+1} - f_k| < \langle \text{epsilon\_f} \rangle$ , the iteration procedure stops and  $\hat{f}$  is set to  $f_{k+1}$ .  
Default value:  $10^{-5}$ .

## VII.6. Allocation of arrays

Allocatable arrays are declared in `<mod_variables.f90>` and `<example_read.f90>`. They must be allocated in the main program (see examples in `<example_simul.f90>` and `<example_read.f90>`) before use. It would be cleaner to do this in a subroutine, but this is not possible in standard Fortran 95.

## VII.7. Routines in `<mod_Aspects.f90>`

- `<ctp_preliminaries>`**: preliminaries required for all computations;
- `<oto_preliminaries>`**: additional preliminaries required for one-to-one computations only;
- `<sto_analysis>`**: iteration procedure converging to the maximum-likelihood estimate  $\hat{f}_{s;o}$  of  $f$  (`<sto_f_u>` in `<example_simul>` and `<example_read>`) for a several-to-one association model; it also computes  $\hat{f}'_{s;o}$  (`<sto_f_p>`) and the standard deviation of  $\hat{f}_{s;o}$  (`<sigma_sto_f_u>`);
- `<ots_analysis>`**: iteration procedure converging to the maximum-likelihood estimate  $\hat{f}'_{o;s}$  of  $f'$  (`<ots_f_p>`) for a one-to-several association model; it also computes  $\hat{f}_{o;s}$  (`<ots_f_u>`) and the standard deviation of  $\hat{f}'_{o;s}$  (`<sigma_ots_f_p>`);
- `<oto_analysis>`**: iteration procedure converging to the maximum-likelihood estimate  $\hat{f}_{o;o}$  of  $f$  (`<oto_f_u>`) for a one-to-one association model <sup>\*5</sup>;

3. Except near the poles, the difference between the true and the observed angles is negligible.

4. One might instead define a maximal number of possible associations between local neighbors and counterparts.

5. If  $K$  and  $K'$  are swapped, one obtains  $\hat{f}'_{o;o} = \hat{f}_{o;o} n/n'$ .

$\hat{f}_{o;o}$  is not the recommended estimator of  $f$  in that case: use rather  $\hat{f}_{o;o}^* := (\hat{f}_{s;o} \hat{f}'_{o;s} n/n)^{1/2}$  (`<oto_f_u_star>`) and  $\hat{f}'_{o;o} := \hat{f}_{o;o} n/n'$  (`<oto_f_p_star>`) for  $f'$ .

- ◀**func\_ln\_sto\_Lh**◀: log-likelihood ( $\ln L_{s:o}$ ) for a several-to-one association model and a given  $f$ ;
- ◀**func\_ln\_ots\_Lh**◀: log-likelihood ( $\ln L_{o:s}$ ) for a one-to-several association model and a given  $f'$ ;
- ◀**func\_ln\_oto\_Lh**◀: log-likelihood ( $\ln L_{o:o}$ ) for a one-to-one association model and a given  $f^{**6}$ ;
- ◀**comput\_sto\_prob**◀: computes several-to-one association probabilities of the  $M'_j$ ,  $P_{s:o}(A_{i,j} | C \cap C')$ , for a given  $f$  and  $M_i$ ;
- ◀**comput\_ots\_prob**◀: computes one-to-several association probabilities,  $P_{o:s}(A_{i,j} | C \cap C')$ , for a given  $f'$ . The sum may be different from 1;
- ◀**comput\_oto\_prob**◀: computes one-to-one association probabilities,  $P_{o:o}(A_{i,j} | C \cap C')$ , for given values of both  $f$  and  $f'$ . One must have  $f/n' = f'/n$ ;
- ◀**write\_prob**◀: outputs to a file the possible counterparts  $M'_j$  of  $M_i$  and corresponding probabilities of association (just computed by one of the ◀comput\_\*t\*\_prob▶ routines).

## A. Notes on subroutine ◀position▶ (file ◀mod\_simul\_catalogs.f90▶)

We provide here some explanations on how subroutine ◀position▶ simulates the observed position of a source from its true position and its positional uncertainty parameters.

Let  $M_0(\alpha_0, \delta_0)$  be the true position of a source;  $a$  ( $\ll \pi$  radians) and  $b$  be the semi-major and semi-minor axes of the positional uncertainty ellipse; and  $\vec{u}_\alpha$  and  $\vec{u}_b$  be unit vectors along the major and minor axes, such that  $(\vec{u}_a, \vec{u}_b, \vec{u}_{r_0})$  is a direct orthonormal basis.

For any vectors  $\vec{v}$  and  $\vec{w}$  perpendicular to  $\vec{u}_{r_0}$ , we note  $\angle(\vec{v}, \vec{w})^+$  the angle from  $\vec{v}$  to  $\vec{w}$  oriented counterclockwise around  $\vec{u}_{r_0}$ , and  $\angle(\vec{v}, \vec{w})^-$  the same angle oriented clockwise (i.e.,  $\angle(\vec{v}, \vec{w})^- = -\angle(\vec{v}, \vec{w})^+ = \angle(\vec{w}, \vec{v})^+$ ).

We want to simulate an observed position,  $M(\alpha, \delta)$ . First, define  $\vec{n} := \vec{u}_r \times \vec{u}_r / \|\vec{u}_r \times \vec{u}_r\|$ ,  $\vec{t} := \vec{n} \times \vec{u}_{r_0}$  (basis  $(\vec{t}, \vec{n}, \vec{u}_{r_0})$  is therefore direct and orthonormal) and  $\psi := \angle(\vec{u}_{r_0}, \vec{u}_r)$ .

Let  $\beta_0 := \angle(\vec{u}_{\delta_0}, \vec{u}_a)^-$  and  $\beta := \angle(\vec{u}_\delta, \vec{u}_a)^-$  be the true and observed positional uncertainty angles,  $\gamma_0 := \angle(\vec{n}, \vec{u}_{\delta_0})^-$ ,  $\gamma := \angle(\vec{n}, \vec{u}_\delta)^-$  and  $\epsilon := \angle(\vec{u}_a, \vec{t})^+$  ( $\gamma$  and  $\beta$  are in the plane perpendicular to  $\vec{u}_r$ , not to  $\vec{u}_{r_0}$ , but one may consider here that  $\vec{u}_r \approx \vec{u}_{r_0}$ ; more rigorously, one may project the sphere on the tangent cylinder containing  $M_0$  and  $M$  and unroll this cylinder on a plane).

The following relations hold:

$$\gamma_0 + \beta_0 = \gamma + \beta = \angle(\vec{n}, \vec{u}_a)^-,$$

so

$$\begin{aligned} \gamma_0 + \beta_0 - \epsilon &= \gamma + \beta - \epsilon \\ &= \angle(\vec{n}, \vec{u}_a)^- - \angle(\vec{u}_a, \vec{t})^+ = \angle(\vec{n}, \vec{u}_a)^- + \angle(\vec{u}_a, \vec{t})^- = \angle(\vec{t}, \vec{n})^+ \\ &= \pi/2 \pmod{2\pi}; \end{aligned} \tag{1}$$

$$\cos \psi = \cos \delta_0 \cos \delta \cos(\alpha - \alpha_0) + \sin \delta_0 \sin \delta; \tag{2}$$

$$\cos \gamma_0 = \frac{\cos \delta \sin(\alpha - \alpha_0)}{\sin \psi}; \tag{3}$$

$$\sin \gamma_0 = \frac{\cos \delta_0 \sin \delta - \sin \delta_0 \cos \delta \cos(\alpha - \alpha_0)}{\sin \psi}. \tag{4}$$

(Cf. Appendix A of Fioc (2012) for Eqs. (2) to (4).)

One has

$$\overrightarrow{M_0 M} \approx \psi \vec{t} = \psi (\cos \epsilon \vec{u}_a + \sin \epsilon \vec{u}_b) = a r_1 \vec{u}_a + b r_2 \vec{u}_b,$$

where  $r_1$  and  $r_2$  are independent random numbers drawn from a normal distribution with mean 0 and standard deviation 1. Therefore,  $\psi \approx ([a r_1]^2 + [b r_2]^2)^{1/2}$  and, using Fortran ◀atan2▶ function to obtain a unique result in  $]-\pi, \pi]$ ,  $\epsilon \approx \text{atan2}(b r_2, a r_1)$ .

One may either draw  $\beta_0$  randomly and derive  $\alpha, \delta$  and  $\beta$  (as in the code), or draw  $\beta$  randomly and derive  $\alpha, \delta$  and  $\beta_0$  (not implemented). Note that in the second case, two positions may be valid solutions of the set of equations.

---

6. Returns the same value if  $K$  and  $K'$  are swapped and  $f/n' = f'/n$ .

### A.1. Determination of $\alpha$ , $\delta$ and $\beta$ from $\beta_0$ (and from $\alpha_0$ , $\delta_0$ , $\psi$ from $\epsilon$ )

Angle  $\gamma_0$  is first derived from Eq. (1),  $\beta_0$  and  $\epsilon$ .

Combining Eq. (4) and Eq. (2), one gets first

$$\sin \gamma_0 \sin \psi = \cos \delta_0 \sin \delta - \sin \delta_0 \frac{\cos \psi - \sin \delta_0 \sin \delta}{\cos \delta_0},$$

then

$$\delta = \arcsin(\sin \gamma_0 \sin \psi \cos \delta_0 + \sin \delta_0 \cos \psi).$$

From Eq. (4) and Eq. (3), one obtains

$$\sin(\alpha - \alpha_0) = \frac{\cos \gamma_0 \sin \psi}{\cos \delta}$$

and

$$\cos(\alpha - \alpha_0) = \frac{\cos \delta_0 \sin \delta - \sin \gamma_0 \sin \psi}{\sin \delta_0 \cos \delta}.$$

Thus,

$$\alpha = \alpha_0 + \text{atan2}(\sin[\alpha - \alpha_0], \cos[\alpha - \alpha_0]) \pmod{2\pi}.$$

Using  $\langle \text{atan2} \rangle$ , one derives  $\gamma$  from

$$\cos \gamma = \frac{\cos \delta_0 \sin(\alpha - \alpha_0)}{\sin \psi} \quad (5)$$

and

$$\sin \gamma = \frac{\cos \delta_0 \sin \delta \cos(\alpha - \alpha_0) - \sin \delta_0 \cos \delta}{\sin \psi}, \quad (6)$$

and then  $\beta$  from Eq. (1) and  $\epsilon$ .

### A.2. Determination of $\alpha$ , $\delta$ from $\beta$ (and from $\alpha_0$ , $\delta_0$ , $\psi$ and $\epsilon$ )

Angle  $\gamma$  is first derived from Eq. (1),  $\beta$  and  $\epsilon$ .

From Eq. (5), one obtains

$$\alpha = \alpha_{(1)} = \alpha_0 + \arcsin \frac{\cos \gamma \sin \psi}{\cos \delta_0} \pmod{2\pi}$$

or

$$\alpha = \alpha_{(2)} = \alpha_0 + \pi - \arcsin \frac{\cos \gamma \sin \psi}{\cos \delta_0} \pmod{2\pi}.$$

Combination  $\sin \delta_0 \times \text{Eq. (2)} + \cos \delta_0 \cos(\alpha_{(i)} - \alpha_0) \sin \psi \times \text{Eq. (6)}$  leads to

$$\sin \delta_{(i)} = \frac{\sin \delta_0 \cos \psi + \cos \delta_0 \cos(\alpha_{(i)} - \alpha_0) \sin \gamma \sin \psi}{1 - \cos^2 \gamma \sin^2 \psi},$$

and  $\cos \delta_0 \cos(\alpha_{(i)} - \alpha_0) \times \text{Eq. (2)} - \sin \delta_0 \sin \psi \times \text{Eq. (6)}$  to

$$\cos \delta_{(i)} = \frac{\cos \psi \cos \delta_0 \cos(\alpha_{(i)} - \alpha_0) - \sin \gamma \sin \psi \sin \delta_0}{1 - \cos^2 \gamma \sin^2 \psi}.$$

Declinations  $\delta_{(1)}$  and  $\delta_{(2)}$  are then derived using  $\langle \text{atan2} \rangle$ . Values of  $\delta_{(i)}$  out of  $[-\pi/2, \pi/2]$  are excluded, but both may fall in this interval: one must then arbitrarily select one of the positions  $(\alpha_{(1)}, \delta_{(1)})$  and  $(\alpha_{(2)}, \delta_{(2)})$ .

## References

- Fioc, M. (2012): *Probabilistic positional association of catalogs of astrophysical sources: the ASPECTS code*. arXiv:1209.5361.
- Kent, J. T. (1982): *The Fisher-Bingham distribution on the sphere*. Journal of the Royal Statistical Society. Series B (Methodological), 44(1), 71.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1992): *Numerical recipes in Fortran. The art of scientific computing*.