THE ASTROMETRIC MODEL IMPLEMENTATION. SIMULATIONS AND DATA REDUCTION COMPATIBILITY TEST

G. Anglada-Escudé, J. Torra, E. Masana, X. Luri

Dept. of Astronomy and Meteorology, Univ. de Barcelona, 08028 Barcelona, Spain

ABSTRACT

The aim of this paper is to give a brief description of the astrometric model accuracy at the current stage of the implementation in GASS (simulator) and GDAAS2 (Data Reduction study). The astrometric model described is a set of algorithms which relate the astrometric parameters with the observed directions on the satellite quasi-intertial reference frame. This includes the kinematics of point sources, the relativistic light deflection due to Solar System gravitational field and the aberration. The description of this model was given by Klioner (2002), The form of these algorithms is slightly different in the telemetry simulations (S.A. Klioner, ANSI-C code) and in the data reduction scheme (Lindegren 2002, Fortran90). Both versions make use of the ephemeris for the Solar System by Observatoire de la Côte d'Azur (Mignard 2003, Fortran 90). All these algorithms have been wrapped or recoded since the simulations and data reduction both run in a Java environment.

All these manipulations required a strict verification since these algorithms constitute the core of the GIS (Global Iterative Solution). We present the compatibility tests performed during last year that helped us to make fully compatible the simulated data with the data reduction scheme.

Key words: Astrometry; Simulation; General Relativity; Data Reduction; Gaia.

1. DELIVERED CODE VERSIONS

1.1. Full PPN Model

This code is a set of ANSI-C modules delivered by S.A. Klioner. The ephemeris is internally managed by the ANSI-C modules. An ANSI-C version of the F. Mignard ephemeris was developed in the early stages of the implementation by the SWG[1] in order to keep the consistency of the astrometric model with other parts of the simulator. *Full PPN model* is going to be used by the simulator in the *predictor* mode. It computes the proper direction on the satellite quasi-intertial reference frame of a given source from its astrometric parameters. Some relevant features of this delivered code are:

- Monopolar deflection: Sun, Earth, Mars, Jupiter, Saturn, Uranus, Neptune (no moons nor minor bodies). This is only limited by the current version of the ephemeris.
- Quadrupolar deflection is considered in the case of the giant planets.
- Use of *accuracy* flags to improve numerical efficiency.
- High numerical efficiency.

1.2. Synthetic PPN Model

The original code was a set of Fortran90 routines delivered by L. Lindegren. This internally manages the calls to the ephemeris. It uses the F. Mignard Fortran90 routines in a slightly adapted version. It is going to be used in the data reduction scheme in, at least, two critical points: IDT[2] in *blind_corrector* submode, GIS[3] in *predictor* mode.

Some features of this delivered code are:

- Monopolar deflection: Sun, Earth, Mars, Jupiter, Saturn, Uranus, Neptune (no moons nor minor bodies).
- Quadrupolar deflection is not available in the delivered version.
- Accuracy flags to improve numerical efficiency.
- Very high numerical efficiency.
- It computes the partial derivatives of the proper direction w.r.t. the astrometric parameters.

[1] Simulations Working Group

[2] Initial Data Treatment

[3] Global Iterative Solution

1.3. RAMOD Analytic Model

The code is a set of ANSI-C routines delivered by Vecchiato (2003). It is planned to receive a more developed version to be applied in GASS in predictor mode. An improved version of these algorithms is expected to be used to assure the consistency of the hypothesis of the fundamental physics underlying the relativistic astrometric model.

2. FIRST TESTING CAMPAIGN

Just before the delivery of the described algorithms (July 2003) work started on testing the compatibility of *Full PPN model* and *Synthetic PPN model*. Another task in those first days was to unify the inputs/outputs of the delivered codes which led to a precise definition of the astrometric parameters, see Klioner (2003).

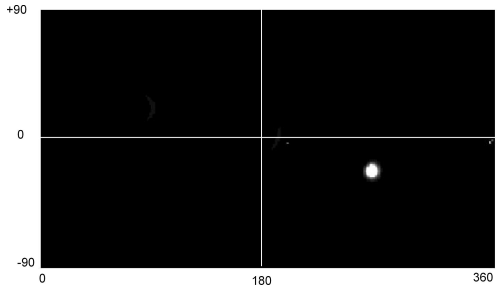

Figure 1. Black zones indicate discrepancies under 0.1 μas. The big white dot where the discrepancy clearly exceeds the 1 μas limit corresponds to the Sun's position. On the right you can see the discrepancy area around Jupiter (tiny white dot) which is detailed in Figure 2.

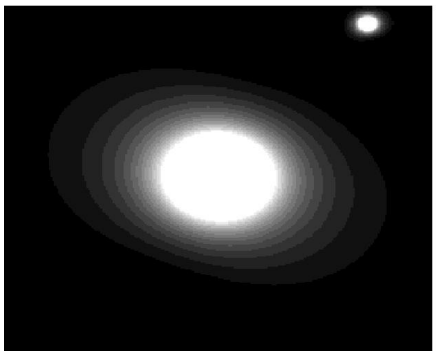

Figure 2. A detailed discrepancy map around Jupiter. In the same image there is the discrepancy zone around Neptune.

When the parameters were standarized the proper test campaigns started. A test consists of the generation of a grid of stellar positions on the surface of a sphere at 10 parsecs. Each simulated star has a velocity of 100 km s^{-1} in each spatial direction (high proper motion regime).

When the global compatibility of the two versions is tested, the stars are generated at each R.A. and declination integer degree; i.e., a grid of $360 \times 181 = 65\,160$ stars. For each synthetic star, the observed direction is calculated using two different *Test configurations* (see below the description of a *Test configuration*) and the angular difference among these vectors is calculated. We call this quantity the *angular discrepancy*.

Figures 1 and 2 are the result of the first successful tests obtained when all the issues involving the input parameters were solved in the technical note *'Proposal for the representation of the astrometric parameters'*, Klioner (2003).

Globally the agreement was very good (*angular discrepancies* were almost everywhere under 0.01 μas), but some relevant discrepancies were detected around the position of the light deflectors (Sun and planets listed in the code description).

To obtain an objective criteria of significance of the discrepancies found we defined the *Error thresholds* (*warning*, *severe* and *fatal*) and the *Error radius*. A *warning*, a *severe* or a *fatal* is generated when the *angular discrepancy* in an observed direction exceeds 0.1 μas, 0.5 μas or 1.0 μas respectively. A smoother grid of N synthetic stars around a deflector is generated. The rate of the number of *warnings*, *severes* or *fatals* is proportional to the square of the radius of an effective circle on the celestial sphere. These radii are the *Error radius* and a test consists essentially in computing them. Some discrepancy maps are generated in order to ensure that the deflector is centred in the window (see Figure 2).

The results of the preliminary test campaigns are given in Table 1.

Table 1. Table showing the first error radius found in the first preliminary compatibility test.

Object	Window Size	R_w	R_s	R_f
Sun	30^0	7.89^0	4.50^0	3.50^0
Venus	5^0	$10.0'$	$4.50'$	$3.20'$
Earth	30^0	7.89^0	4.50^0	3.50^0
Mars	5^0	$2.90'$	$1.20'$	$1.00'$
Jupiter	5^0	1.67^0	0.79^0	0.52^0
Saturn	5^0	$21.0'$	$9.60'$	$6.70'$
Uranus	5^0	$10.7'$	$4.70'$	$3.40'$
Neptune	5^0	$5.70'$	$2.50'$	$1.70'$

3. INTEGRATION OF THE ALGORITHMS

Once the found discrepancies were reported to the RRFWG[4] we started the real integration of the algorithms within the simulator/data reduction scheme.

[4]Relativity and Reference Frame Working Group

The *Full PPN model* was wrapped into Java. The ephemeris module is still managed from inside the ANSI-C code to avoid complicated Java-C interfaces. A wrapper consists of a *dynamic library* properly compiled and a Java Native Interface Class. This approach caused considerable loss of portability of the GASS, but was chosen because the complexity of such codes could have caused severe problems during the Java translation. The Quadrupolar deflection was switched of since the *Synthetic PPN model* had not the capability of compute it.

The *Synthetic PPN model* was translated entirely since the code was relatively simpler than the *Full PPN model* and it was considered critical to run the data reduction scheme avoiding the use of wrappers as much as possible. The translation of these algorithms also forced the translation of the ephemeris Fortran90 code to Java.

A campaign of compatibility tests to compare the Java version and the Fortran90 one was performed during December 2003. The results of this tests are shown in Figure 3.

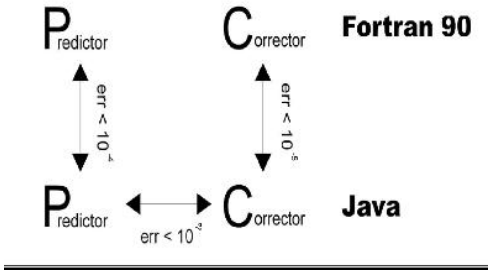

Figure 3. The scheme of the tests carried out during Winter 2003–2004. Each double sense arrow represents a test. The 'err' quantities are the average discrepancies found in μas.

4. FINAL INTEGRATED VERSIONS AND COMPATIBILITY TESTS

From the results of the first test, a discusion with the algorithms providers helped to identify some bugs in the code distributions which were amended in the final integrated codes.

With the experience acquired in the previous test campaigns we defined a set of test protocols which helped us to perform the final comparisons and quantify the final compatibility level. A test consists of evaluating the *Error Radius* defined above when two *Test Configurations* are confronted for a given observation time.

A *Test Configuration* is :

- Title
- Code: Code version going to be used (Lindegren Java, Klioner Java-C, Vecchiato Java-C,...)
- Effects: Relevant relativistic corrections taken into account
- Observation time: This parameter will help us to test the different contributions to the light deflection effects in different colar system spatial configurations. The reference epoch will be JD2010.0 in all test configurations.

Figure 4 contains the results of a test using the final configurations integrated in GASS (simulator) and GDAAS2 (data reduction).

The one used in the Simulator:

- Reference Monopolar Configuration
- Code: Klioner Java-C
- Effects: only mass monopoles
- Observation time: JD2013.0

and the one used in GDAAS2 (IDT and GIS):

- GDAAS monopolar configuration
- Code: L.Lindegren Java
- Effects: only mass monopoles
- Observation time: JD2013.0

With such configurations and the codes debugged we obtained the *Error radius* for this last test as shown in Table 2.

Table 2. If you compare the numbers is this table with the results shown in Table 1 you will notice that they are about 2 orders of magnitude smaller.

Object	Window Size	R_w	R_s	R_f
Sun	20^0	7.50^0	4.70^0	3.50^0
Venus	$5'$	$31.0'$	$12.1'$	$9.00'$
Earth	$30''$	–	–	$372''$
Mars	$2'$	$6''$	$3''$	$2''$
Jupiter	$10'$	$97''$	$55''$	$43''$
Saturn	$5'$	$72''$	$33''$	$23''$
Uranus	$1'$	$16''$	$7'$	$5''$
Neptune	$1'$	$12''$	$6''$	$4''$

ACKNOWLEDGMENTS

This work has been possible through the effort of the people of the Barcelona group that have introduced me to the 'world' of Gaia. Special thanks for the patience shown by S.A. Klioner, F. Mignard, L. Lindegren, M. Lattanzi and all the other algorithm providers whom we have intensively contacted during the past year.

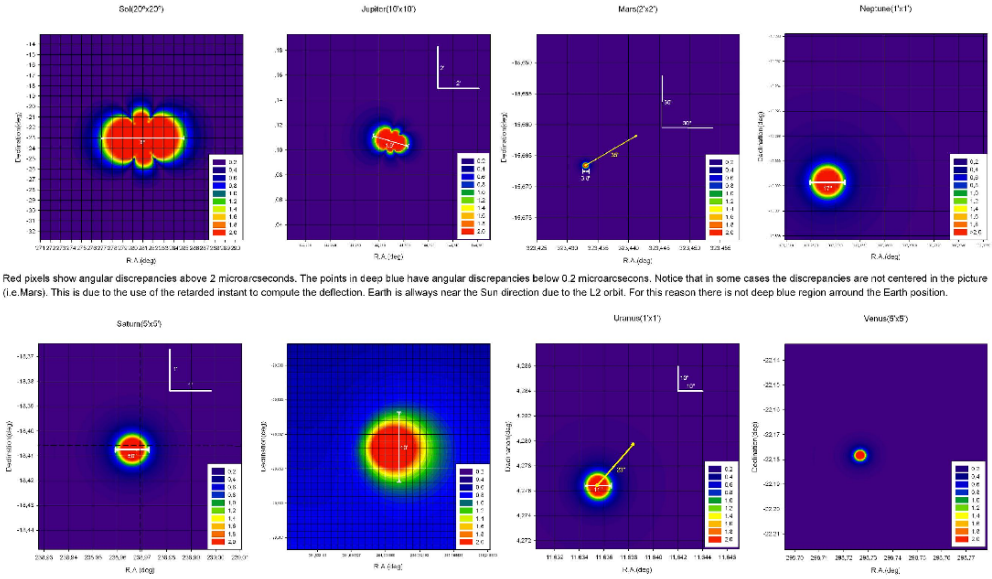

Figure 4. Test results using final configurations.

REFERENCES

Klioner, S.A., 2002, *A practical model for microarcsecond astrometry from space*, Astronomical Journal 125,1580-1587; (preprint available from arXiV as astro-ph/0107457)

Klioner, S.A., 2003, *Proposal for the representationof the astrometric parameters*, Gaia SWG technical report (Available in livelink on 25 sept 2003)

Lindegren, L., 2002, Gaia technical report GAIA-LL-044

Mignard, F., Gaia technical report GAIA-FM-013

Vecchiato, A., et al., Gaia technical report RRFWG-TOPD-001