.

ODAS: Process description and collection of relevant mathematical algorithms

prepared by: S. Jordan, U. Bastian, W. Löffler

reference: GAIA-C3-TN-ARI-SJ-009-01
issue: 1
revision: 1
date: 26-Jan-2009
status: Issued

Abstract

The aim of this document is to provide a detailed description of the ODAS processes and the underlying algorithms. This document is not finished but will be updated over the next months in order to provide the necessary input for the ODAS software development in cycle 6 (and following cycles).
The current version of this document can be downloaded from http://gaia.esac.esa.int/dpacsvn/DPAC/CU3/docs/FL/nonECSS/GAIA-C3-TN-ARI-SJ-009/GAIA-C3-TN-ARI-SJ-009.pdf

Document History

Issue	Revision	Date	Author	Comment
1	1	26 Jan 2009	SJ	Correction of index k in Delta zeta
1	1	23 Jan 2009	SJ	Corrections according to UB. First issue for Livelink!
D	14	23 Jan 2009	SJ	Definition of zeta0 modified for mu dependence
D	13	22 Jan 2009	SJ	Definition of eta0 and zeta0 added
D	12	7-14 Jan 2009	SJ	Definition of B-splines changed+other smaller changes
D	11	16 Dec 2008	WL	General clean-up
D	10	11-25 Nov 2008	SJ	New chapters
D	9	11 Nov 2008	SJ	Smaller corrections
D	9	30 Oct 2008	SJ	Including corrections by UB
D	8	22-24 Oct 2008	SJ	Several new diagrams and chapters
D	7	22 Oct 2008	SJ	Constraint equations set up
D	6	21-22 Oct 2008	SJ	Design matrix included
D	5	20 Oct 2008	SJ	Corrections according to UB+other updates
D	4	19 Oct 2008	SJ	Correction of B-spline derivatives
D	3	14-17 Oct 2008	SJ	ICRS-RGC transformations added
D	2	11-14 Oct 2008	SJ	First updates
D	1	10 Oct 2008	SJ	First lines added

Contents

0.1 Applicable Documents

Jordan et al. (SJ-006)	First Look Software Requirements Specifications
Bastian (BAS-003)	Reference systems, conventions and notations for Gaia

0.2 Reference Documents

Bastian, U., 2006, *Extended geometric calibration model for Gaia's Astro instrument*, GAIA-ARI-BAS-011-4, URL http://www.rssd.esa.int/llink/livelink/open/420453

Bastian, U., 2007, *Reference systems, conventions and notations for Gaia*, GAIA-CA-SP-ARI-BAS-003, URL http://www.rssd.esa.int/llink/livelink/open/358698

Bernstein, H., Bastian, U., Hirte, S., 2005, *First look - description of the ring solution*, GAIA-ARI-BST-001-5, URL http://www.rssd.esa.int/llink/livelink/open/381198

Bombrun, A., Löffler, W., 2008, *Helmert Blocking Algebra in ODAS (not published in Livelink yet)*, GAIA-APB-001

ESA Gaia Project, 2006, *Mission Requirements Document (MRD)*, GAIA-EST-RD-00553, URL http://www.rssd.esa.int/llink/livelink/open/463164

Fabricius, C., Masana, E., 2007, *Nominal field angles in cycle 2 simulations*, GAIA-C2-TN-UB-CF-004, URL http://www.rssd.esa.int/llink/livelink/open/2757946

Hirte, S., Bernstein, H., Bastian, U., 2006, *Program description of the ring solution*, GAIA-C3-TN-ARI-SH-003, URL http://www.rssd.esa.int/llink/livelink/open/2698804

Hirte, S., Bernstein, H., Bastian, U., 2007, *Results of the Ring Solution*, GAIA-C3-TN-ARI-SH-004, URL http://www.rssd.esa.int/llink/livelink/open/2760797

Jordan, S., Biermann, M., Brown, A., et al., 2008, *First Look Software Requirements Specifications*, GAIA-C3-TN-ARI-SJ-006, URL http://www.rssd.esa.int/llink/livelink/open/2720764

Lindegren, L., 2000, *Attitude parameterization for GAIA*,
SAG-LL-030,
URL http://www.rssd.esa.int/llink/livelink/open/358607

Lindegren, L., 2001, *Proposed prototype processes for the GAIA Global Iterative Solution*,
GAIA-LL-034,
URL http://www.rssd.esa.int/llink/livelink/open/357825

Lindegren, L., 2003, *Algorithms for GDAAS Phase II - Definition*,
GAIA-LL-044,
URL http://www.rssd.esa.int/llink/livelink/open/357790

Lindegren, L., 2008, *Geometric calibration model for the 1M astrometric GIS demonstration*,
GAIA-C3-TN-LU-LL-063,
URL http://www.rssd.esa.int/llink/livelink/open/510843

Lindegren, L., Bastian, U., 2008, *Local plane coordinates for the detailed analysis of complex Gaia sources (in prep.)*,
GAIA-LL-061-3

0.3 Definitions, acronyms, and abbreviations

The following table has been generated from the on-line Gaia acronym list:

Acronym	Description
AC	Auto Collimation
AE	Astro Elementary
AF	Astrometric Field (in Astro)
AGIS	Astrometric Global Iterative Solution
AL	Acceptance Load
ASM	Automatic Storage Management
CCD	Charge-Coupled Device
CFS	Calibration Faint Stars
CU	Coordination Unit (in DPAC)
DFLM	Detailed First Look Monitor
DU	Development Unit (in DPAC)
FL	First Look
FPRS	Focal Plane Reference System
FoV	Field of View (also denoted FOV)
FoVRS	Field-of-View Reference System
GASS	GAia System-level Simulator
ICRS	International Celestial Reference System
IDT	Initial Data Treatment (Image Dissector Tube in Hipparcos scope)
ODAS	One-Day Astrometric Solution
ODC	One-Day Calibration
OGA1	First On-Ground Attitude determination (in IDT)
OGA2	Second (and improved) On-Ground Attitude determination (in ODAS/FL)
RGC	Reference Great Circle
RGCS	Reference Great-Circle System
SM	Sky Mapper
SNR	Signal-to-Noise Ratio (also denoted SN and S/N)
UB	University of Barcelona (Spain)

1 Introduction

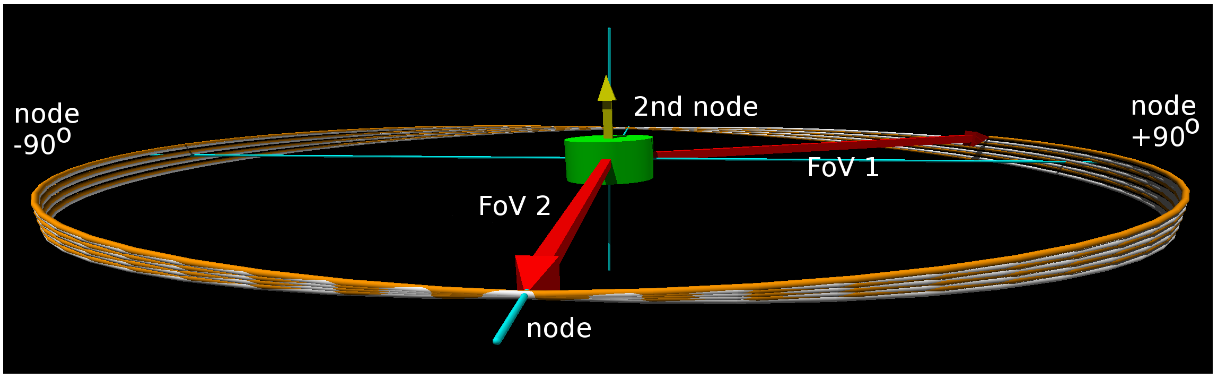

FIGURE 1: During one day the scanning plane of Gaia tilts by about 4° so that only regions around the two nodes are measured along slightly different direction.

The One-day Astrometric Solution (ODAS) is a part of the Gaia First Look (FL). The aim of the FL is to check on a daily basis, whether Gaia reaches the targeted level of precision. The final astrometric precision can only be obtained after many months of observational data have been incorporated in a global, coherent and interleaved data reduction. Neither the instrument state nor the data quality can be checked at the desired level of precision by standard procedures applied to typical space missions. It is desirable to know the measurement precision and instrument stability as soon as possible since unperceived, subtle effects can arise during the mission which could affect all data and potentially result in a loss of many months of data if not detected in a timely manner.

Like AGIS, the astrometric ODC, called ODAS, aims at an astrometric self-calibration of Gaia, but takes into account the special problems of the restriction in the time basis; the input data consist of a contiguous set of observations down to a magnitude of about $G = 16$ for a period of 14 to 24 hours each day. During one day the scanning plane of Gaia will vary only slightly ($\approx 4^\circ$) so that scan directions change only little. Therefore, only the along-scan coordinates along a Reference Great Circle (RGC) can be measured with high precision, across-scan information is only provided by the sky mappers, unbinned windows on the AF CCDs, and the finite change of scanning directions near the nodes of the scanning (Fig, 1). No proper motions, parallaxes, or global parameters will be determined by ODAS.

In addition to the determination of source positions, very precise values for the geometric calibration of the CCDs, as well as an attitude solution (OGA2) being orders of magnitude more accurate than the attitude solution provided by IDT (OGA1) will be available. Like AGIS, ODAS will be based on about 1-10% of the observed celestial objects called ‘primary sources’. For the case of ODAS, these are sources that have passed all tests for being astrometrically well-behaved, i.e., they are apparently single, they are constant over a day, they are, at the one hand, bright enough that a high SNR can be obtained for each source and, at the other hand, are faint enough that no saturation effects occur.

The primary sources used by ODAS are not necessarily the same as the primary sources of AGIS. The same celestial source can be judged as being primary source one day and then fail to meet the criteria another day. In order to distinguish them from the their AGIS counterparts we will call them one-day primary sources.

The input data for ODAS consist of the elementary observation data for the one-day primary sources, the OGA1 produced by IDT, a catalogue of the cross-matched sources with source positions calculated for the time of the observations, and a set of calibration parameters, copied from the Main Database.

The so-called Ring Solution (see Hirte et al. (SH-003); Bernstein et al. (GAIA-ARI-BST-001-5)) will be used as the basic module of ODAS. This is a direct (non-iterative) solution of the astrometric problem restricted to about 24 hours. Therefore, the correlations between source positions, attitude, and geometric calibration can be fully taken into account.

The output data of ODAS consist of the updated source positions for the one-day primary sources, the attitude, calibration parameters for the current observation time interval as well as residuals and normalised residuals for each elementary observation.

The investigation of the calibration parameters itself and the residuals of the ODAS solution is the tool to investigate whether the daily measurements are of the necessary precision to comply with the goals of Gaia as defined in the Mission Requirements Document ESA Gaia Project (GAIA-EST-RD-00553).

2 Definition of the most important indices used in this document

For the definition of indices I use the same convention as Lindegren (LL-063):

- ℓ identifies an observation (the transit of one source across a single CCD chip)
- i identifies the source
- f identifies the FoV (Astro-1 or Astro-2)
- n identifies the CCD chip. Actually, it may refer to the stitches of a CCD in future development cycles.
- m identifies the pixel column on the chip. μ is the continuous version of m allowing for fractions of pixels.

- j identifies a "short" calibration time interval (AGIS assumes constancy over months; in the case of ODAS the time interval is the current day).
- k identifies a "long" calibration time interval (AGIS assumes constancy of almost the whole mission;in the case of ODAS the time interval is the current day).

The indices i, f, n, m are known from the AstroElementary record for the observation ℓ.

Additionally, the index u is used for the knot index of the B-splines representing the attitude parameters with u defined by $t_u \leq t_\ell^{\rm obs} < t_{u+1}$ $t_\ell^{\rm obs}$ is the time of the observation ℓ (the time when the centroid of an image is crossing the nominal fiducial line; this time is directly provided in the AstroElementary records).

3 ODAS input and output data, auxiliary data

The input to ODAS consists of

- AstroElementary records of all ODAS primary stars.(+additional stars?), provided by IDT
- an input star catalogue which contains at least the astrometric parameters of all sources, for which AstroElementaries are read in
- a cross-match table which related AstroElementaries to the observed sources (provided by IDT)
- An initial attitude (OGA1, provided by IDT)
- Initial values for the 828 large-scale geometric parameters (coefficients of Legrange polynomials up to the second degree) (6 for all 14 SM CCDs, 12 for all 62 AF CCDs, see Fig. 6). If no reliable initial list exists, we can start by setting all these parameters to zero. At the start of the mission, nominal or pre-launch calibration values should be used. During the mission, updated values from previous ODAS runs or previous AGIS runs are to be used.
- Initial values for the xx small-scale geometric parameters (1966 $\times$ 2 for each CCD, 298832 for all 76 SM and AF CCDs). These values are only used and not updated! If no reliable initial list exists, we must set these parameters to zero. Intital values from previous analysis of the ODAS residuals or from previous AGIS runs.

The output of ODAS consists of

- An updated star catalogue in the RGCS, in which the source positions are updated. The other astrometric parameters (proper motion, parallaxes, and radial velocities are kept constant.
- The updated star catalogue converted to ICRS.
- An updated attitude (OGA2$^{\mathrm{RGCS}}$) defined in the RGCS.
- OGA2$^{\mathrm{RGCS}}$ converted to the ICRS.
- Updated values for the 828 large-scale geometric parameters (coefficients of Legrange polynomials up to the second degree) (6 for all 14 SM CCDs, 12 for all 62 AF CCDs, see Fig. 6). As long as simulation data do not take into account CCD distortions, we can restrict the updates to the zero-order and first-order large-scale geometric parameters.
- Residuals for all along and across-scan observations.
- The rms of the residuals.
- Supplementary information for the residuals which is used by the ODAS-related DFLM (see OdasResiduals definition in the Gaia Main Database Dictionary Tool).

We need auxiliary data for the

- Gaia Orbit
- Solar system ephemeris.
- OBT-TCB relation.

4 General ODAS data flow

As stated above, the ODAS performs an astrometric solution for the Gaia data of one day.

The adequate coordinate system for the ODAS is the so-called Reference Great Circle System (RGCS), where the Reference Great Circle (RGC) is defined to coincide closely with the mean scanning plane of Gaia during the (approximately) one-day time period for which the ODAS solution is performed.

The advantage of this choice is that the longitudinal RGCS coordinate v measured along the RGC is closely related to the along-scan measurements, while the latitudinal RGC coordinate r changes approximately in across-scan direction. The largest deviations of the along-scan

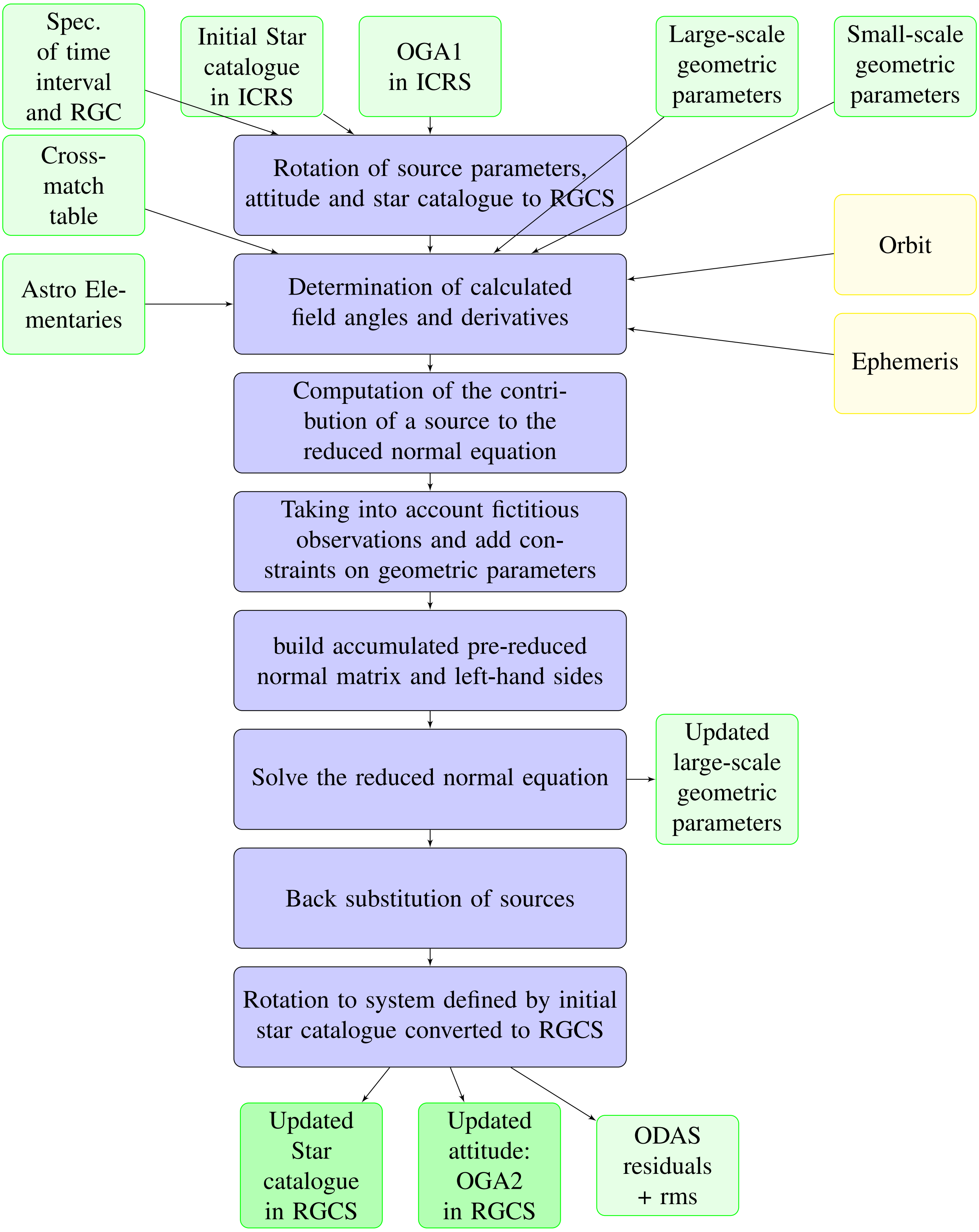

FIGURE 2: Data flow and major software blocks of ODAS.

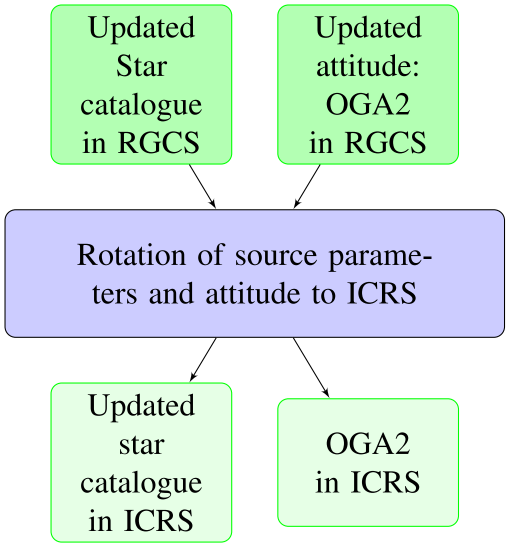

FIGURE 3: ODAS data flow (continued): Transformation of results to ICRS .

directions from the being parallel to the RGC are $\pm 2^\circ$ at the nodes of the scanning-law (see Fig, 1).

The block diagram of the ODAS data flow is shown in Fig. 2.

In the first step of the ODAS, the astrometric parameters (source positions α and δ, proper motions μ_α and μ_δ; the parallaxes and radial velocities are not affected by the choice of the coordinate system) of the objects in the initial input star catalogue must be converted to the RGCS (v, r, μ_v, μ_r). The relevant formula are listed in Sect. 10.1. Besides the astrometric parameters of the sources, the attitude (OGA1) must also be transformed to the RGCS (see Sect.9.1.

In order to determine the calculated field angles for a given observation (and their derivatives with respect to the source positions, the attitude parameters and the geometric calibration parameters), we need the observation time from the Astro Elementaries, the identification of the source (via the cross-match table), the source position and proper motion of this source (now in RGCS), the attitude OGA1 (converted to the RGCS), the large-scale and small-scale geometric calibration parameters, the position of Gaia at the time of the observation (via the Orbit File), and the Ephemeris of the solar-system objects which significantly contribute to the light bending (Sun, Earth, Jupiter, Saturn, . . .).

Using for each source i the vector l_{S_i} containing the differences between observed and calculated field angles (left-hand side of the observation equation, Eq. 21), the source block S_i (Eq. 46, the attitude/calibration block of the design matrix O_i (Eq. 47), and the weight matrix W_i, the contribution of the source i to the reduced normal equation can be computed (Eq. 53). Additionally, fictitious observation equations (in order to enforce normalisation of the attitude quaternions, see Sect. 5.8) and constraints for the geometric calibration parameters (in order to avoid degeneracy with the attitude parameters, see Eq. 15) must be taken into account.

After the normal matrix and the left-hand sides have been built up, the reduced normal equations are solved for the updates of the attitude unknowns (in a coordinate system which due to remaining degrees of freedom is rotated against the RGC) and the updates to the large-scale geometric geometric parameters (independent of the coordinate system).

With the attitude and geometric calibration parameters known, the source parameters can be determined by a back-substitution process (see Sect. 6.6).

Since our method still contains a degeneracy between the attitude and the source positions the results for these unknowns are provided in a coordinate system, which may be rotated against the RGCS. Therefore, the updated star catalogue is rotated by two angles, until the coordinate differences between the updated star catalogue and the input catalogue (converted into the RGCS) are minimised in a least-squares sense. Using these rotation angles, the attitude is also transformed into the RGCS.

The First Look can fully be performed in the RGCS, but if the source positions or the updated attitude (called OGA2) is used for other purposes, the source catalogue and the OGA2 should also be made available in the ICRS (Fig. 3) (formula given in Sect.10.1 and 9.1).

In order to consider outliers, a second ODAS run is necessary which, after deletion or down-weighting will use the updated star catalogue, attitude and large-scale geometric parameters from the first ODAS run. This will also minimise small non-linearities in the ODAS.

The output of the ODAS, in particular the ODAS residuals and errors of the ODAS residuals will be investigated with the ODAS-related Detailed First Look Monitor and Evaluator (see Jordan et al. (SJ-006)).

5 The observation equation

5.1 Abstract definition

The term “observation equation” is often used in astrometry and geodesy. In other fields of science, the same concept is usually called “condition equation”.

The actual observation can always be expressed as the sum of a modelled observation and an error term:

$$O^{\mathrm{obs}} = O^{\mathrm{model}}(x_1, \ldots, x_p) + \mathrm{error}$$

.

The model depends on the parameters $x_1, \ldots, x_p$.

If we apply Taylor's theorem, where we expand about the model computed using provisional parameter values $(x_1^0, \ldots, x_p^0)$ and ignore second and higher order terms,

$$O^{\text{model}} = O^{\text{model}}(x_1^0, \ldots, x_p^0) + \sum_{i=1}^{p} (x_i - x_i^0) \frac{\partial O^{\text{model}}}{\partial x_i}$$

.

Then the linearised observation equations are given as

$$\Delta O = O^{\text{obs}} - O^{\text{model}}(x_1^0, \ldots, x_p^0) = \sum_{i=1}^{p} (x_i - x_i^0) \frac{\partial O^{\text{model}}}{\partial x_i} + \text{error} = \sum_{i=1}^{p} \Delta x_i \frac{\partial O^{\text{model}}}{\partial x_i} + \text{error}$$

.

If we have o observations $O_1^{\text{obs}}, \ldots, O_o^{\text{obs}}$ and write the equations in matrix form, we have

$$\underbrace{\begin{pmatrix} \Delta O_1 \\ \Delta O_2 \\ \ldots \\ \Delta O_o \end{pmatrix}}_{\Delta O} = \underbrace{\begin{pmatrix} \frac{\partial O_1^{\text{model}}}{\partial x_1} & \frac{O_1^{\text{model}}}{\partial x_2} & \ldots & \frac{\partial O_1^{\text{model}}}{\partial x_p} \\ \frac{\partial O_2^{\text{model}}}{\partial x_1} & \frac{O_2^{\text{model}}}{\partial x_2} & \ldots & \frac{\partial O_2^{\text{model}}}{\partial x_p} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\partial O_o^{\text{model}}}{\partial x_1} & \frac{O_o^{\text{model}}}{\partial x_2} & \ldots & \frac{\partial O_o^{\text{model}}}{\partial x_p} \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \ldots \\ \Delta x_p \end{pmatrix}}_{\Delta x} + \underbrace{\begin{pmatrix} \text{error}_1 \\ \text{error}_2 \\ \ldots \\ \text{error}_o \end{pmatrix}}_{error} \tag{1}$$

A is called Design Matrix (a special form of a Jacobian matrix). O^{model} is usually called the "predicted" or "calculated" observation.

5.2 The astrometric measurements

The astrometric measurements of Gaia consist of

- the observation times t_ℓ^{obs} when the centroid of an image is crossing the fiducial line of the relevant CCD (along-scan measurements)
- and the transverse pixel coordinate μ_ℓ (across-scan measurements, not always available on AF2-9).

In Lindegren (LL-063) it is explained that it is of great advantage to formulate the observation equations for both, along-scan and across-scan, in field angles η (along-scan) and ζ (across-scan), since we can deal with angles in both cases rather than with times and angles, and one avoids more complex calculations of predicted transit times.

In order to determine the observed across-scan field angles, the transverse pixel coordinate μ_ℓ must be transformed according to the current geometric across-scan calibration. In order to determine the "observed" along-scan field angles, the current geometric along scan calibration must be computed at the given across-scan pixel coordinate μ_ℓ.

5.3 The left-hand side of the observation equation

Let

- F be the nominal focal length (:Satellite:Telescope_FocalLength, currently $= 35$ m)
- f be the field of view (Fov 1=preceding field of view, representing light collected by telescope 1, FoV 2=the following field of view, representing light collected by telescope 2)
- $x_0(f)$ be the nominal FPRS across-scan coordinate of the intersection of the optical axis with the focal plane for the two FoVs (:Satellite:FoV_Centre_AC$\cdot 10^{-3}$ m/mm, currently -37.5 mm$\cdot 10^{-3}$m/mm$= -0.0375$ m for FoV 1, and $+37.5$ mm$\cdot 10^{-3}$ m/mm$=$ $+0.0375$ m for FoV 2)
- $y_0(f)$ be the nominal FPRS along-scan coordinate of the intersection of the optical axis with the focal plane for the two FoVs (:Satellite:FoV_Centre_AL$\cdot 10^{-3}$ m/mm, currently 0 m for FoV 1, and 0 m for FoV 2). Therefore, y_0 is currently not FoV dependent.
- x_{cn} be the nominal FPRS across-scan coordinates of the geometrical centres of the image section of the CCDs in the focal plane (array :Satellite:SM:CCD_Position_NominalX$\cdot 10^{-3}$ m/mm for the SM CCDs, array :Satellite:AF:CCD_Position_NominalX$\cdot 10^{-3}$ m/mm for the AF CCDs). The mapping of the CCD designation n to the array indices is described in the Gaia Parameter Data Base description of the array.
- y_{cn} be the nominal FPRS along-scan coordinate of the geometrical centres of the image section of the CCD n in the focal plane (array :Satellite:SM:CCD_Position_NominalY$\cdot 10^{-3}$ m/mm for the SM CCDs, array :Satellite:AF:CCD_Position_NominalY$\cdot 10^{-3}$ m/mm for the AF CCDs). The mapping of the CCD designation n to the array indices is described in the Gaia Parameter Data Base description of the array.
- μ_c be the nominal center of each CCD in across-scan CCD pixel coordinates (:Satellite.CCD_LIGHTSENSITIVEAREA_AC_ZEROPOINT + ((:Satellite.CCD_NUMBEROFCOLUMNS - 1.0d) * 0.5d), which is $= 14 + (1966 - 1)/2 = 996.5$)
- κ_c be the center of each CCD in along-scan CCD pixel coordinates, i.e. in CCD lines (1+((:Satellite.CCD_NUMBEROFTDILINES - 1)/2), which is $= 1 + (4500 - 1)/2 = 2250.5$.

- $\kappa_{\text{fiducial}}(g)$ is the position of the nominal fiducial line in along-scan pixels (:Satellite:CCD_TDILine_NominalFiducial), the index corresponds to the gate g with the first index corresponding to gate 1, and the 12^{th} index to gate 12. Index 13 means "no gating". The current (Cycle 6) values of κ_{fiducial} are listed in Tab. 1.
- s_x be the nominal AC pixel size (:Satellite.CCD_PIXELAREA_AC * 1.0E-6, currently $= 30 \cdot 10^{-6}$ m)
- s_y be the nominal AL pixel size (:Satellite.CCD_PIXELAREA_AL * 1.0E-6, currently $= 10 \cdot 10^{-6}$ m)

Due to the formulation of both the along-scan and the across-scan observation equation as field angles, the actual observation t_ℓ^{obs} enters the "calculated" (model) side of the along-scan observation equation rather than the "observed" side. On the other hand, the geometric calibration parameters enter the "observed" side.

The "observed" along-scan field angle is the along-scan field-angle position of the fiducial line (which depends on the gating) of the relevant CCD chip and pixel column, according to the current geometric calibration. This is modelled as:

$$\eta_\ell^{\text{obs}} = \eta_n^0 + \Delta\eta_{fnj} + \delta\eta_{nmk} + \dots . \tag{2}$$

The "observed" across-scan field angle is, analogously:

$$\zeta_\ell^{\text{obs}} = \zeta_{nf}^0 + (\mu - \mu_c)s_x/F + \Delta\zeta_{fnj\cdot} + \delta\zeta_{nmk} + \dots . \tag{3}$$

Note, that the ζ_{nf}^0 is depending on the FoV for the simulations performed in cycle 5 and later. Since ζ_{nf}^0 refers to the center of the CCD ($\mu = \mu_c$), the term $(\mu - \mu_c)s_x/F$ determines the field angle for other values of the across-scan pixel coordinate μ. This term was not mentioned in the current version of Bastian (GAIA-ARI-BAS-011-4) and Lindegren (LL-063) but was accounted for by the AGIS software (priv.comm. with Uwe Lammers). If this term would be neglected, the μ dependence would cause non-zero values of the across-scan geometrical parameters even for a nominal Gaia.

For a given gating (which defines the position of the fiducial line), the geometric calibration is given by the nominal along-scan coordinate η_n^0 associated with each CCD chip (both for ASM and AF), together with the nominal basic angle, and the nominal across-scan coordinate ζ_{nf}^0 of each ASM chip. These nominal values are never changed. The actual calibration is expressed as small corrections to these nominal angles. In the present model, all geometric calibration parameters are expressed in radians (Lindegren (LL-063)).

The η_n^0 and ζ_{nf}^0 are calculated according to Eqs. 3 to 6 in Fabricius & Masana (CF-004).

TABLE 1: The position of the nominal fiducial transit line for each gate. The values in this table are provided in the array :Satellite:CCD_TDILine_NominalFiducial=[3.5, 5.5,. . . , 2253.497329773031] in the Gaia Parameter Data Base. When converting to Java, note that the gates are counted from 1 to 12 and "none" corresponds to the last entry. See Fabricius & Masana (CF-004). The numbering of the gates is consistent with Fabricius & Masana (CF-004) but differs from the one used in the GaiaTools routine WindowCoordinateConversion in gaia.cu1.tools.satellite.coordinate.

Gate (g)	Length	$\kappa_{\rm fiducial}$
1	2	3.500000000000
2	4	5.500000000000
3	8	9.000000000000
4	16	13.750000000000
5	32	22.125000000000
6	64	38.312500000000
7	128	70.406250000000
8	256	134.453125000000
9	512	262.476562500000
10	1024	518.488281250000
11	2048	1030.494140625000
12	2900	1456.495862068966
none	4494	2253.497329773031

Then

$$\eta_n^0 = -((y_{cn} - (\kappa_{\text{fiducial}} - \kappa_c)s_y) - y_0)/F \quad (4)$$

$$\zeta_{nf}^0 = -(x_{cn} - x_0(f))/F \quad (5)$$

This result can be obtained by the following call to the relevant GaiaTool's method in the class gaia.cu1.tools.satellite.coordinate:

η_n^0 =fromWinCoordToFovRs(byte fov, byte ccdRow,byte ccdStrip, byte gate, double mu).get.X()

See the Javadocs for the definition of fov, ccdRow, ccdStrip, and gate.

The output is independent of the input value for mu=μ.

ζ_{nf}^0 =fromWinCoordToFovRs(byte fov, byte ccdRow,byte ccdStrip, byte gate, double mu).get.Y()

mu must be set to $\mu = \mu_c = 996.5$.

The FL has asked CU2 (JP) to deliver the η_n^0 and ζ_{nf}^0 as a routine deliveray of GASS.

$\Delta\eta_{fnj}$ and $\Delta\zeta_{fnj}$ are determined from the large-scale geometric parameters as defined in Sect. 5.3.1 [1] $\delta\eta_{nmk}$ and $\delta\zeta_{nmk}$ are the small-scale geometric parameters. The dots indicate, that magnitude and color dependent terms may have to be added in the future but are not taken into account in development cycle 6. The small-scale parameters will not be updated by ODAS.

The "computed" along-scan field angle is

$$\eta_\ell^{\text{calc}} = F_\eta(t_\ell^{\text{obs}}, \boldsymbol{s}_i, \boldsymbol{a}(t_\ell^{\text{obs}})) \quad (6)$$

The computed across-scan field angle is

$$\zeta_\ell^{\text{calc}} = F_\zeta(t_\ell^{\text{obs}}, \boldsymbol{s}_i, \boldsymbol{a}(t_\ell^{\text{obs}})) \quad (7)$$

The function $F = (F_\eta, F_\zeta)$ calculates the predicted field angles (η_ℓ, ζ_ℓ) from an assumed position of the source (from the input source catalogue) corresponding to the observation ℓ and the assumed attitude for the observation time t_ℓ^{obs}.

In AGIS the astrometric source parameters and the attitude are defined in the ICRS. The Java method getPredictedFieldAngles in the Java class AngleCalculatorImpl [2] calculates F and the

[1] Often, $\Delta\eta_{fnj}$ and $\Delta\zeta_{fnj}$ itself are called geometric parameters. However, the parameters used in the extended geometric calibration model are the coefficients of the Legendre polynomials

[2] gaia.cu3.agis.algo.gis.fieldangleimp.AngleCalculatorImpl.java

partial derivatives with respect to the source position and the attitude. For this the attitude and the proper direction (and its partial derivatives with respect to the attitude) are needed. Alternatively, we can use the Gaia Toolbox routine toProperSourceDirection and perform the conversion to field angles and derivatives of field angles with respect to source parameters and attitude parameters by using the method described in Sect. 11.

The function F involves the proper direction to the star at the observation time. In order to calculate this, we also need the barycentric ephemeris of the satellite, of the Sun, and of those major planets which are taken into account for the relativistic model.

Fig. 4 sketches how the calculation of the predicted field angles and their derivatives depend on input data and data models.

If we want to do the calculation in the ODAS RGCS, a couple of modifications are needed.

The left-hand side of the observation equation is defined as the difference between the observed and computed values:

$$\Delta\eta_\ell = \eta_\ell^{\mathrm{obs}} - \eta_\ell^{\mathrm{calc}} \tag{8}$$

and

$$\Delta\zeta_\ell = \zeta_\ell^{\mathrm{obs}} - \zeta_\ell^{\mathrm{calc}} \tag{9}$$

5.3.1 The extended geometric calibration Model by BAS-011

Bastian (GAIA-ARI-BAS-011-4) has extended the geometric calibration model of Lindegren (GAIA-LL-044) by introducing higher-order large-scale calibration parameters in the form of coefficients of Legendre polynomials, by across-scan calibration of all CCD strips, and by a specific form for the across-scan pixel coordinates. Additionally, Lindegren (LL-063) has modelled the magnitude-dependent (photometric) and colour-dependent (chromaticity) geometric parameters.

The geometric calibration models distinguish between "large-scale" and "small-scale" geometric parameters. The large-scale parameters provide a calibration for each CCD and, for the AF CCDs for each FoV separately. To the two SM CCD strips, only one FoV is connected (SM1 connected to FoV 1, SM2 connected to FoV 2); therefore, the large-scale parameters apply to the CCD only.

The large-scale shifts are called $\Delta\eta_{fnj}$ and $\Delta\zeta_{fnj}$.

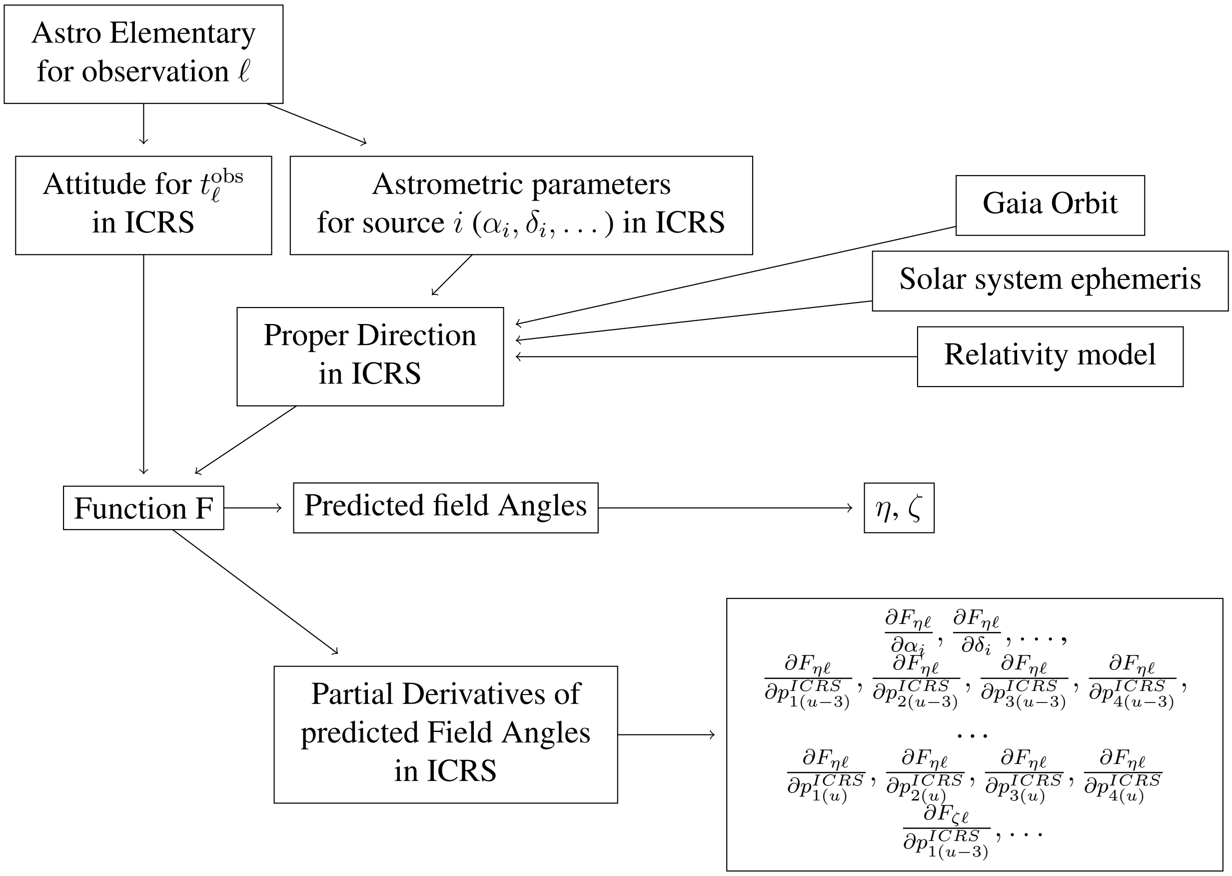

FIGURE 4: AGIS calculates the predicted field angles and its derivatives with respect to the astrometric parameters and attitude B-spline coefficients with the Java method getPredictedFieldAngles in the Java class AngleCalculatorImpl. This method can be re-used in ODAS, if the proper transformation from the ICRS results into the RGCS is made. Alternatively, we can use the Gaia Toolbox routine toProperSourceDirection and perform the conversion to field angles and derivatives of field angles with respect to source parameters and attitude parameters by using the method described in Sect. 11.

Additionally, small-scale parameters $\delta\eta_{nmk}$ and $\delta\zeta_{nmk}$ exist. ODAS itself will not update these values, but we must be able to apply these calibration parameters. From the ODAS residuals, updates to the small-scale parameters can in principle be determined which can be applied the following days.

The large-scale calibration model, according to Bastian (GAIA-ARI-BAS-011-4) is

$$\Delta\eta_{fnj}(\tilde{\mu}) = \sum_{r=0}^{2} \Delta\eta_{r\ fnj} L_r^*(\tilde{\mu}) \tag{10}$$

$\tilde{\mu}$ is the CCD-specific normalized across-scan pixel coordinate (following Bastian (BAS-003)):

$$\tilde{\mu} = \frac{\mu - \mu_0 + \frac{1}{2}}{1966} \tag{11}$$

Here μ_0 is the across-scan zero point of the CCD's light sensitive area. Its value is provided by the Gaia Parameter Database value :Satellite.CCD_LIGHTSENSITIVEAREA_AC_ZEROPOINT= 14. Eq. 11 ensures, that $\tilde{\mu} = 0.5$ for $\mu = \mu_c$ (with μ_c being the across-scan centre of the CCDs in pixel coordinates, see Fabricius & Masana (CF-004)) and that $0 \leq \tilde{\mu} \leq 1$ for the entire AC extent of a CCD.

Since GASS is using this definition of μ, the shift by μ_0 pixels is necessary in order to keep $\tilde{\mu}$ between 0 and 1 (as is necessary for the calibration model as defined by Bastian (GAIA-ARI-BAS-011-4).

1966=:Satellite:CCD_LightSensitiveArea_AC_Pixel is the total number of pixel columns in the light sensitive area of each CCD.

$L^*_r(\tilde{\mu})$ are shifted Legendre polynomials. They are normalized so that $L^*_r(1) = 1$. The three Legendre polynomials which are currently considered for ODAS are[3]

$$\begin{aligned} L^*_0(\tilde{\mu}) &= 1 \\ L^*_1(\tilde{\mu}) &= 2\left(\tilde{\mu} - \frac{1}{2}\right) \\ L^*_2(\tilde{\mu}) &= 6\left(\tilde{\mu} - \frac{1}{2}\right)^2 - \frac{1}{2} \end{aligned} \tag{12}$$

The actual along-scan calibration unknowns for ODAS are $\Delta\eta_{0\ fnj}$, $\Delta\eta_{1\ fnj}$, and $\Delta\eta_{2\ fnj}$.

$\Delta\eta_{0\ fnj}$ corresponds to a simple shift in η (because $L^*_0(\tilde{\mu}) = 1$), the $\Delta\eta_{1\ fnj} L^*_1(\tilde{\mu})$, and $\Delta\eta_{2\ fnj} L^*_2(\tilde{\mu})$ terms corespond to a shear and image distortion (see Fig. 5).

In the same way, the large-scale across-scan calibration is defined by

$$\Delta\zeta_{fnj}(\tilde{\mu}) = \sum_{r=0}^{2} \Delta\zeta_{r\ fnj} L^*_r(\tilde{\mu}) \tag{13}$$

Along-scan and across-scan calibration is necessary for all SM and AF CCDs. For AF2-9, across-scan calibration is only possible with the windows of bright stars and so called CFS (Calibration Faint Stars).

[3]For the cycle 6 development, the second-order term can be neglected. The software design needs to be flexible to fill-in additional geometric parameters.

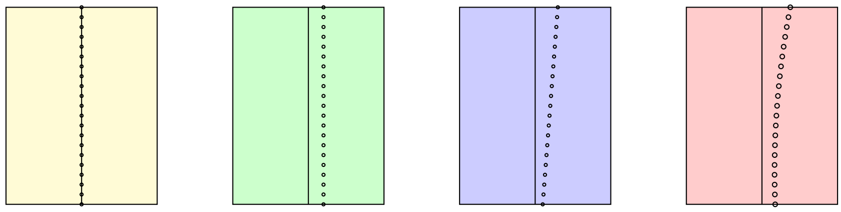

FIGURE 5: Fiducial line of a CCD with all geometric calibration parameters $= 0$ (left), with only a zero order term $\Delta\eta_{0\ fnj} \neq 0$ (shift, 2nd from left), with also a first order term $\Delta\eta_{1\ fnj} \neq 0$ (shift+shear, 3rd from left), and with also a second order term $\Delta\eta_{2\ fnj} \neq 0$ (shift+shear+distortion, right).

In order to avoid degeneracy between the attitude parameters and geometric calibration parameters, the zeroth order large-scale geometric parameters are subject to the constraints

$$\begin{aligned} \sum_{n=\mathrm{SM1\_1}}^{\mathrm{SM1\_7}} \Delta\eta_{0\ 1nj} + \sum_{n=\mathrm{AF1\_1}}^{\mathrm{AF9\_7}} \Delta\eta_{0\ 1nj} + \sum_{n=\mathrm{SM2\_1}}^{\mathrm{SM2\_7}} \Delta\eta_{0\ 2nj} + \sum_{n=\mathrm{AF1\_1}}^{\mathrm{AF9\_7}} \Delta\eta_{0\ 2nj} &= 0 \\ \sum_{n=\mathrm{SM1\_1}}^{\mathrm{SM1\_7}} \Delta\zeta_{0\ 1nk} + \sum_{n=\mathrm{AF1\_1}}^{\mathrm{AF9\_7}} \Delta\zeta_{0\ 1nk} &= 0 \\ \sum_{n=\mathrm{SM2\_1}}^{\mathrm{SM2\_7}} \Delta\zeta_{0\ 2nk} + \sum_{n=\mathrm{AF2\_1}}^{\mathrm{AF9\_7}} \Delta\zeta_{0\ 2nk} &= 0 \end{aligned} \tag{14}$$

j and k are assumed to be constant over one day. In the following, $j = k = 1$. For the across-scan large-scale geometric parameters, two separate sums for each FoV must be zero.

Note, that for SM1 only the parameters $\Delta\eta_{0\ 1nj}$ and $\Delta\zeta_{0\ 1nj}$ exist, and no $\Delta\eta_{0\ 2nj}$ or $\Delta\zeta_{0\ 2nj}$. For SM2, only $\Delta\eta_{0\ 2nj}$ and $\Delta\zeta_{0\ 2nj}$ exist, and no $\Delta\eta_{0\ 1nj}$ or $\Delta\zeta_{0\ 1nj}$.

In the matrix form $Hh_0 = r_{\rm H}$ used by Bombrun & Löffler (GAIA-APB-001) these constraints are written as

$$\begin{pmatrix} 1 & \dots & 1 & 1 & \dots & 1 & 1 & \dots & 1 & | & 0 & \dots & 0 & 0 & \dots & 0 & | & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & | & 1 & \dots & 1 & 1 & \dots & 1 & | & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 & | & 0 & \dots & 0 & 0 & \dots & 0 & | & 1 & \dots & 1 & 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} \Delta\eta_{0\ 1,\mathrm{SM1\_1},1} \\ \vdots \\ \Delta\eta_{0\ 1,\mathrm{SM1\_7},1} \\ \Delta\eta_{0\ 1,\mathrm{AF1\_1},1} \\ \vdots \\ \Delta\eta_{0\ 1,\mathrm{AF9\_7},1} \\ \Delta\eta_{0\ 2,\mathrm{AF1\_1},1} \\ \vdots \\ \Delta\eta_{0\ 2,\mathrm{AF9\_7},1} \\ - - - - - \\ \Delta\zeta_{0\ 1,\mathrm{SM1\_1},1} \\ \vdots \\ \Delta\zeta_{0\ 1,\mathrm{SM1\_7},1} \\ \Delta\zeta_{0\ 1,\mathrm{AF1\_1},1} \\ \vdots \\ \Delta\zeta_{0\ 1,\mathrm{AF9\_7},1} \\ - - - - - \\ \Delta\zeta_{0\ 2,\mathrm{SM1\_1},1} \\ \vdots \\ \Delta\zeta_{0\ 2,\mathrm{SM1\_7},1} \\ \Delta\zeta_{0\ 2,\mathrm{AF1\_1},1} \\ \vdots \\ \Delta\zeta_{0\ 2,\mathrm{AF9\_7},1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{15}$$

Problem to be solved later: As we have found out during a discussion, we cannot simply assume that the geometric calibration is the same for all windowing schemes. We have to assume that the final calibration model will have different calibration parameters for different window classes (which depend on the magnitude and on the usage of the special calibration mode). This problem is of general importance for the astrometric calibration and goes beyond the question of the FL (ODAS) calibration alone. Discussion on CU3 level is needed before a decision what is applied within the ODAS is taken. For development cycle 6, no dependence of the geometric calibration on the window class is assumed.

Stitching: CCDs are not produced with a single electron etching run, but in segments called stitches. The different stitches of a CCD may therefore need different calibrations. Then the CCD index would refer to the stitches rather than the full CCD. For development cycle 6 we assume only one set of parameters per CCD, i.e. not per stitch. The software needs to be flexible to account for the geometric calibration on the level of stitches in the future.

Gating: We do not use gating in the development cycle 6 software. For ODAS, gating in essence extends the variety of window classes. For the proper FL purpose of ODAS, the usage of gated observations can probably be avoided even in the real mission. This must be taken into account in the filter process defining the ODAS primary stars.

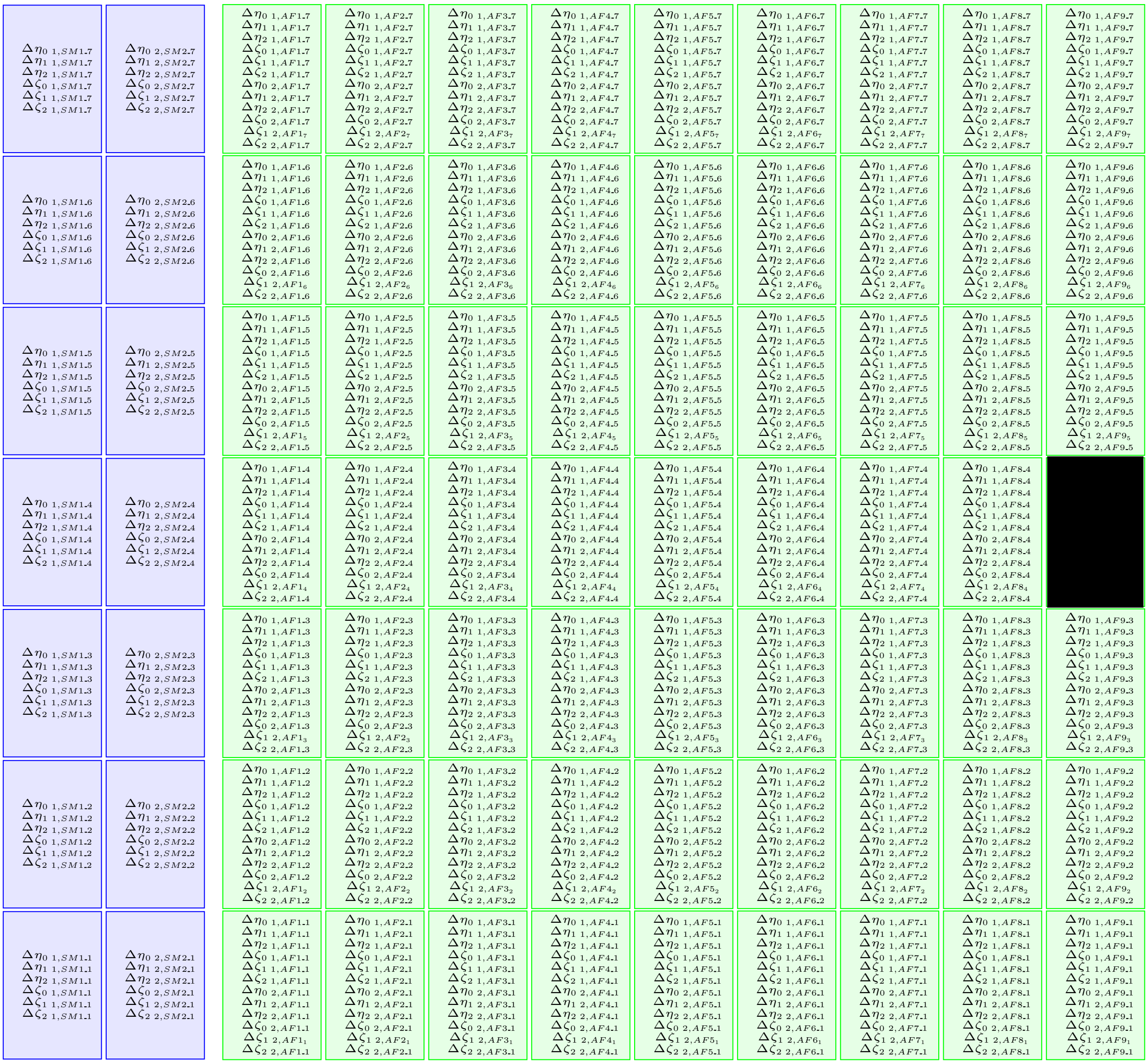

FIGURE 6: ODAS geometric calibration parameters for all CCDs if we use 2 as the highest degree of the Legendre polynomials.

5.3.2 Geometric calibration of magnitude and colour dependent terms

To be accounted for in future development cycles.

5.4 The observation equations for the ODAS

The model parameters for the *computed* field angles $\eta_\ell^{\rm calc}$ and $\zeta_\ell^{\rm calc}$ are the assumed source parameters $\boldsymbol{s}_i$ of the source which corresponds to the observation, and the attitude parameters $\boldsymbol{a}(t_\ell^{\rm obs})$ that correspond to the time of the observation. According to the adopted Gaia modelling, the geometric calibration parameters are formally added to the *observed* field angles. The consequence is that the relevant derivatives in the design matrix have a minus sign in front.

In ODAS, the source parameters and the attitude must be treated in the RGC system.

The source parameters in the RGC system for the Gaia observation l are $\boldsymbol{s}_i^{\rm RGCS} = (v_i, r_i, \mu_{vi}^*, \mu_{ri}, \varpi_i, v_{{\rm rad}i})$. Only v and r are updated by ODAS.

The attitude parameters are the 16 quaternion B-spline coefficients

$$\boldsymbol{p}^{\rm RGCS} = (p_{1(u-3)}, p_{2(u-3)}, p_{3(u-3)}, p_{4(u-3)}, \ldots, p_{1(u)}, p_{2(u)}, p_{3(u)}, p_{4(u)}).$$

The indices $1, 2, 3, 4$ correspond to the quaternion component index. The index u is the knot index of the B-splines with u defined by $t_u \le t_\ell^{\rm obs} < t_{u+1}$

The along-scan large-scale geometric calibration parameters are $\Delta\eta_{0\ fnj}$, $\Delta\eta_{1\ fnj}$, and $\Delta\eta_{2\ fnj}$.

The across-scan large-scale geometric calibration parameters are $\Delta\zeta_{0\ fnj}$, $\Delta\zeta_{1\ fnj}$, and $\Delta\eta_{2\ fnj}$.

According to Eq. 12 in Bastian (GAIA-ARI-BAS-011-4) the part of the observation equations for the large-scale calibration read

$$\eta_\ell^{\rm obs} - \eta_n^0 = -\sum_{r=0}^{2} \Delta\eta_{r\ fnj} L_r^*(\tilde{\mu}) \tag{16}$$

The minus sign is due to the fact that the geometric calibration unknowns appear on the "observed" rather than on the "calculated" side.

The derivatives thus read

$$\frac{\partial(\eta_\ell^{\rm obs} - \eta_n^0)}{\partial\Delta\eta_{r\ fnj}} = -L_r^*(\tilde{\mu}) \tag{17}$$

Since it is the "observed" η, which depends on the calibration parameters and η_n^0 are constants, it follows

$$\frac{\partial \eta_\ell^{\text{obs}}}{\partial \Delta\eta_{r\ fnj}} = -L_r^*(\tilde{\mu}) \tag{18}$$

$$\begin{pmatrix} \Delta\eta_\ell \\ \Delta\zeta_\ell \end{pmatrix} = \begin{pmatrix} \frac{\partial F_{\eta\ell}}{\partial \varpi_i} & \frac{\partial F_{\eta\ell}}{\partial r_i} & \frac{\partial F_{\eta\ell}}{\partial p_{1(u-3)}} & \frac{\partial F_{\eta\ell}}{\partial p_{2(u-3)}} & \dots & \frac{\partial F_{\eta\ell}}{\partial p_{3(u)}} & \frac{\partial F_{\eta\ell}}{\partial p_{4(u)}} & -L_0^*(\tilde{\mu}) & -L_1^*(\tilde{\mu}) & -L_2^*(\tilde{\mu}) \\ \frac{\partial F_{\zeta\ell}}{\partial \varpi_i} & \frac{\partial F_{\zeta\ell}}{\partial r_i} & \frac{\partial F_{\zeta\ell}}{\partial p_{1(u-3)}} & \frac{\partial F_{\zeta\ell}}{\partial p_{2(u-3)}} & \dots & \frac{\partial F_{\zeta\ell}}{\partial p_{3(u)}} & \frac{\partial F_{\zeta\ell}}{\partial p_{4(u)}} & -L_0^*(\tilde{\mu}) & -L_1^*(\tilde{\mu}) & -L_2^*(\tilde{\mu}) \end{pmatrix} \begin{pmatrix} \Delta\varpi_i \\ \Delta r_i \\ p_{1(u-3)} \\ p_{2(u-3)} \\ \dots \\ p_{3(u)} \\ p_{4(u)} \\ \Delta\eta_{0\ fnj} \\ \Delta\eta_{1\ fnj} \\ \Delta\eta_{2\ fnj} \\ \Delta\zeta_{0\ fnj} \\ \Delta\zeta_{1\ fnj} \\ \Delta\zeta_{2\ fnj} \end{pmatrix} + \begin{pmatrix} \text{error}(\eta_\ell) \\ \text{error}(\zeta_\ell) \end{pmatrix} \tag{19}$$

Note, that $-L_0^*(\tilde{\mu}) = 1$ per definition. According to Lindegren (GAIA-LL-034) (page 12, after Eq. 15), the partial derivatives with respect to the astrometric source parameters $\left(\frac{\partial F_{\eta\ell}}{\partial \varpi_i}, \frac{\partial F_{\eta\ell}}{\partial r_i}\right)$ and the attitude parameters $\left(\frac{\partial F_{\eta\ell}}{\partial p_{1(u-3)}}, \dots \frac{\partial F_{\eta\ell}}{\partial p_{4(u)}}\right)$ can be calculated by the routine which calculates F. The Java method is getPredictedFieldAngles in AngleCalculatorImpl [4].

5.5 Modifications needed in order to use the AGIS method for predicted field angles and derivatives

The AGIS calculator for the predicted field angles and its partial derivatives with respect to the astrometric parameters and attitude B-spline coefficients can be reused, if a couple of transformations are made:

1. The ODAS attitude is defined in the RGCS. Therefore, a transformation into the ICRS is necessary. The knot sequence is kept the same in both coordinate systems. Therefore, the B-spline coefficients can directly be transformed using the method described in Sect. 9.2.
2. The source parameters are defined in the RGCS. The transformation to the ICRS can be done with the method described in Sect. 10.1.
3. The field angles are independent of the coordinate system.

[4] gaia.cu3.agis.algo.gis.fieldangleimp.AngleCalculatorImpl.java

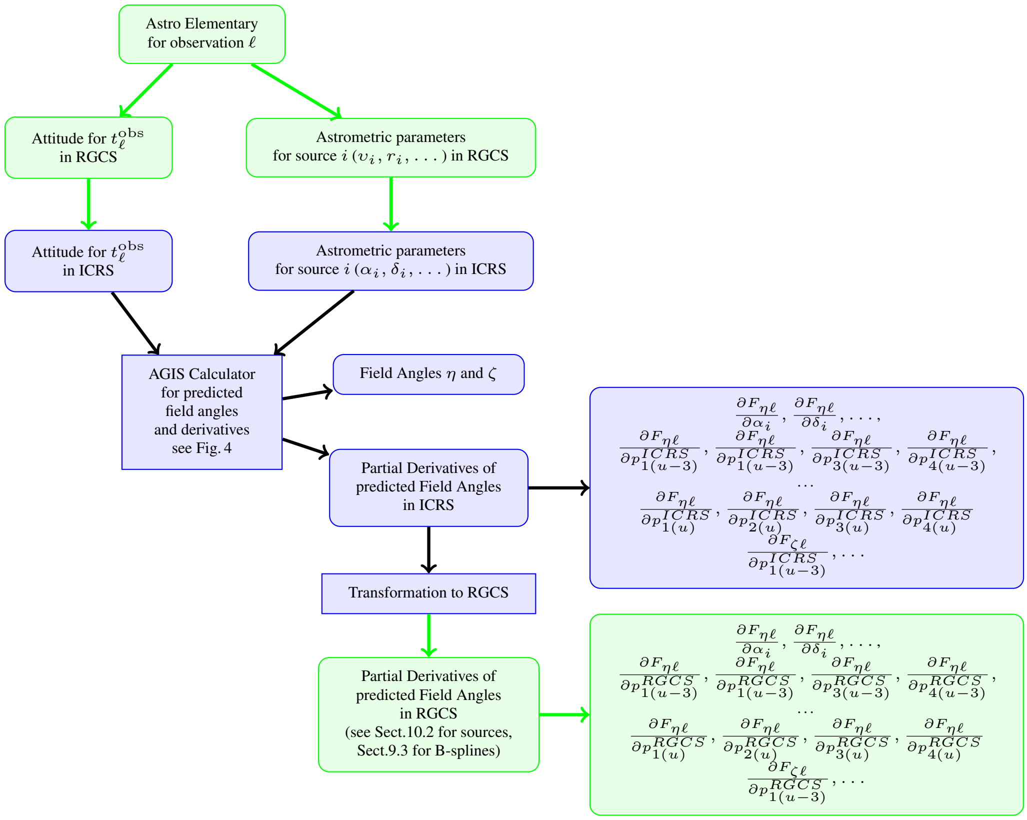

FIGURE 7: Before using the AGIS calculator for the predicted field angles and its partial derivatives with respect to the astrometric parameters and attitude B-spline coefficients, coordinate transformations from RGCS to ICRS are necessary for attitude and B-spline coefficients. The field angles, produced as an output are independent of the coordinate system, but the partial derivatives have to be transformed from ICRS to RGCS. The blue boxes in the figure correspond to the AGIS part (see Fig.4). The green boxes indicate those parts where transformations are necessary.

4. The partial derivatives of the predicted field angles with respect to the source parameters and the attitude parameters are defined in the ICRS. These have to be transformed with the methods described in Sect. 10.2 and Sect. 9.3, respectively.

This is schematically shown in Fig. 7.

5.6 Filling the design matrix

The arrangement of the unknowns is in principle arbitrary, but we want to choose a clearly laid out concept.

Let I be the number of sources, with the source parameters $\boldsymbol{s}_1^{\rm RGCS}, \ldots, \boldsymbol{s}_i^{\rm RGCS}, \ldots, \boldsymbol{s}_I^{\rm RGCS}$. The source parameters of the source i are $\boldsymbol{s}_i^{\mathrm{RGCS}} = (\upsilon_i, r_i, \mu^*_{\upsilon i}, \mu_{ri}, \varpi_i, \upsilon_{\mathrm{rad}i})$. Since proper motions, parallaxes and radial velocities are not updated, only partial derivatives with respect to source parameters υ_i and r_i occur in the design matrix.

For each observation ℓ we have the following parameters and indices relevant for parameter updates by ODAS:

- The two source parameters υ_i , and r_i.
- The 16 attitude parameters
  $p_{1(u-3)}, p_{1u-2}, p_{1(u-1)}, p_{1(u)},$
  $p_{2(u-3)}, p_{2u-2}, p_{2(u-1)}, p_{2(u)},$
  $p_{3(u-3)}, p_{3u-2}, p_{3(u-1)}, p_{3(u)},$
  $p_{4(u-3)}, p_{4u-2}, p_{4(u-1)}, p_{4(u)},$
- (Currently) the along-scan large geometric calibration parameters for the AF CCDs $\Delta\eta_{0\ fnj}$, $\Delta\eta_{1\ fnj}$, and $\Delta\eta_{2\ fnj}$
- (Currently) the across-scan large geometric calibration parameters for the AF CCDs $\Delta\zeta_{0\ fnj}$, $\Delta\zeta_{1\ fnj}$, and $\Delta z\eta_{2\ fnj}$. These parameters are only updated if across-scan information is available for CCD n.
- (Currently) the along-scan large geometric calibration parameters for the SM CCDs $\Delta\eta_{0\ nj}$, $\Delta\eta_{1\ nj}$, and $\Delta\eta_{2\ nj}$
- (Currently) the across-scan large geometric calibration parameters for the SM CCDs $\Delta\zeta_{0\ nk}$, $\Delta\zeta_{1\ nk}$, and $\Delta z\eta_{2\ nk}$ This parameters are only updated if across-scan information is available for CCD n.

For the cycle 6 development the second-order geometric calibration parameters $\Delta\eta_{2\ fnj}$, $\Delta\zeta_{2\ fnj}$, $\Delta\eta_{2\ fnj}$, $\Delta\zeta_{2\ fnj}$ can be omitted.

To every source i and its parameters, only nine FoV transits are possible in exactly one day (more, if the ODAS time interval is larger than one day). In every transit, one SM observation and up to nine AF observations (AF1-AF9) of the same source are possible. Therefore, up to 90 observations may contribute to the update of a source parameter in ODAS. Note, that each observation can give rise to one or two observation equations (along and across scan).

The attitude parameters at knot u are connected to all observations made in four B-spline intervals. Therefore, a much larger number of observations can contribute to one attitude parameter than to a source parameter.

The geometric parameters on one CCD n are connected to all observations performed on this CCD. This number of observations is much larger than the number of observations contributing to an attitude parameter.

At the other hand, the number of source parameters is much larger than the number of attitude parameters, which in turn is larger than the number of geometric parameters (in the future, however, the number of geometric parameters may grow).

Let L, I, U, and $N+1$ be the number of observations, sources, attitude knots, and CCDs (CCDs are numbered from 0 to N, so that we have $N+1$ CCDs), respectively. The indices for the calibration time intervals can be set to constant $k=1$ and $j=1$ CCD indices are counted from 0 to 75. The software developers can decide to omit the constants k and j completely. The introduction of them was only used to make a comparison with the def intions for AGIS easier (Lindegren (LL-063)).

We sort all observations $1 \leq \ell \leq L$ into blocks corresponding to the same source. The information which source corresponds to an observation ℓ is provided by the cross-match table. The lth observation of the source i is designated with the double index il ($\eta_{il}^{\rm obs}$, $\zeta_{il}^{\rm obs}$). The number of observations for source i is called E_i and Z_i for measurements in η and ζ, respectively. The sorting leads to the blocking structure for the design matrix shown in Fig.9. If we also want to have a blocking structure of the attitude parameters as also shown in that figure, we have to sort all observations $i1, \ldots iE_i$ (along-scan) and $i1, \ldots iZ_i$ within the source blocks in time order. This ordering is, however, not necessary for ODAS and provides only a “nice” structure.

In the same way, the computed field angles are designated with this extra index ($\eta_{il}^{\rm calc} = F_{\eta,il}$ $\zeta_{il}^{\rm calc} = F_{\zeta,il}$).

In order to be able to write down the equations in a compact form, we used an index $n = 0, \ldots N = 75$ for the CCDs. The correspondence between the n and the correct CCD designation is shown in Fig. 8.

Note, that parameter for SM1 only $\Delta\eta_{0\ 1nj}$ and $\Delta\zeta_{0\ 1nj}$ exists, and no $\Delta\eta_{0\ 2nj}$ or $\Delta\zeta_{0\ 2nj}$. For SM2, only $\Delta\eta_{0\ 2nj}$ and $\Delta\zeta_{0\ 2nj}$ exists, and no $\Delta\eta_{0\ 1nj}$ or $\Delta\zeta_{0\ 1nj}$.

Therefore, there are e.g. no calibration parameters $\Delta\eta_{r\ 21j}$ and $\Delta\zeta_{r\ 21j}$. This is the reason why $\Delta\eta_{1\ 111}$ follows $\Delta\eta_{2\ 101}$ and not $\Delta\eta_{1\ 211}$ in the vector of ODAS updates (and therefore in the respective columns of the design matrix). [5]

The ODAS software must take into account, that CCDs may be missing. This scheme will also change, as soon as stitching is taken into account.

[5] Wolfgang: You can, of course find a more reasonable way to take this into account.

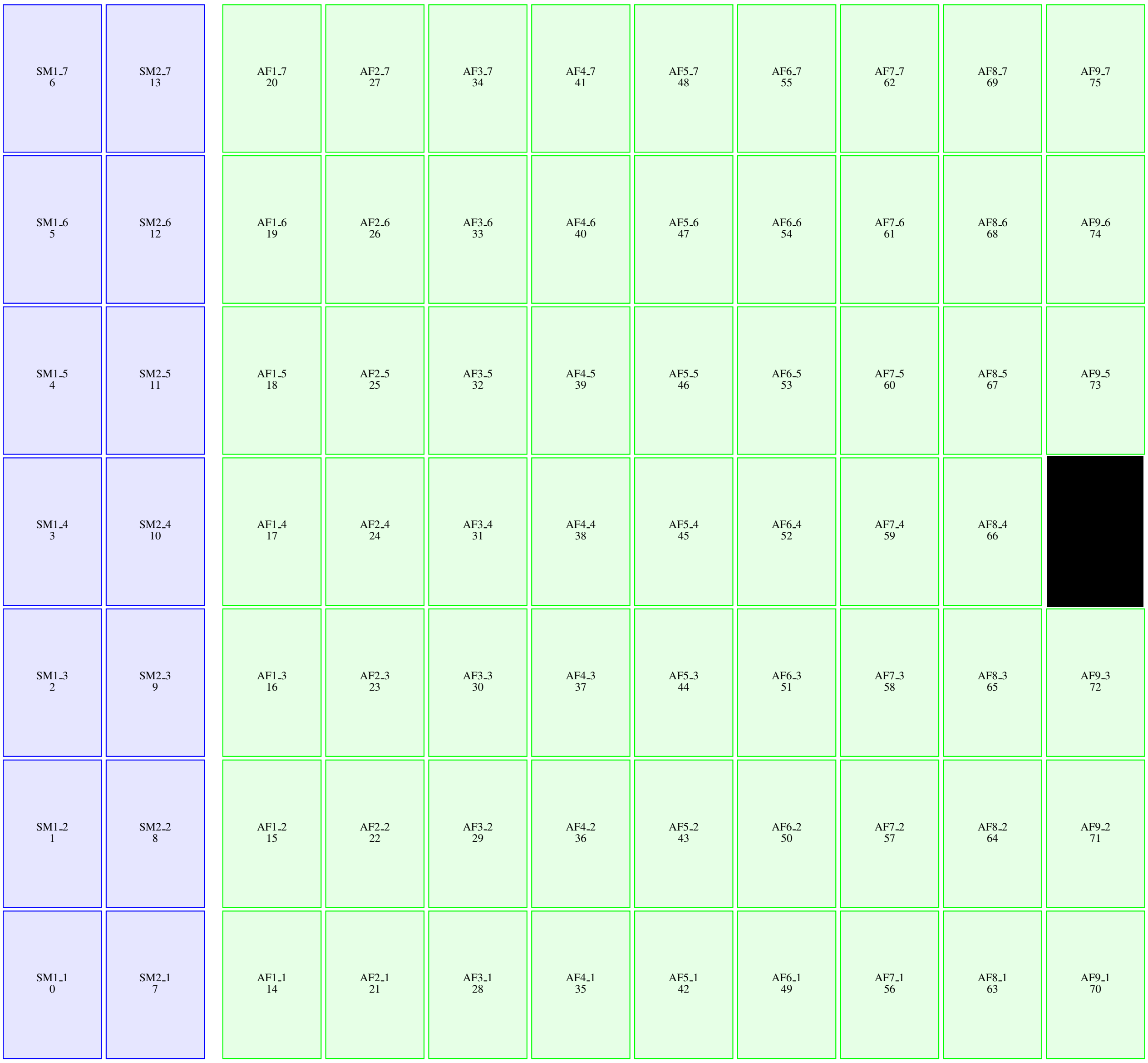

FIGURE 8: CCD indices used for writing down the geometric parameters in the observation equation. CCD indices start with 0!

The vector of the updates of the ODAS unknowns are:

$$\Delta x = \begin{pmatrix} \Delta\eta_1 \\ \Delta\zeta_1 \\ \vdots \\ \Delta\eta_I \\ \Delta\zeta_I \\ \hline \Delta p_{11} \\ \Delta p_{12} \\ \Delta p_{13} \\ \Delta p_{14} \\ \vdots \\ \Delta p_{U1} \\ \Delta p_{U2} \\ \Delta p_{U3} \\ \Delta p_{U4} \\ \hline \Delta(\Delta\eta_{0\ 101}) \\ \Delta(\Delta\eta_{1\ 101}) \\ \Delta(\Delta\eta_{2\ 101}) \\ \Delta(\Delta\eta_{0\ 111}) \\ \Delta(\Delta\eta_{1\ 111}) \\ \Delta(\Delta\eta_{2\ 111}) \\ \vdots \\ \Delta(\Delta\eta_{0\ 1N1}) \\ \Delta(\Delta\eta_{1\ 1N1}) \\ \Delta(\Delta\eta_{2\ 1N1}) \\ \Delta(\Delta\eta_{0\ 2N1}) \\ \Delta(\Delta\eta_{1\ 2N1}) \\ \Delta(\Delta\eta_{2\ 2N1}) \\ \hline \Delta(\Delta\zeta_{0\ 101}) \\ \Delta(\Delta\zeta_{1\ 101}) \\ \Delta(\Delta\zeta_{2\ 101}) \\ \Delta(\Delta\zeta_{0\ 111}) \\ \Delta(\Delta\zeta_{1\ 111}) \\ \Delta(\Delta\zeta_{2\ 111}) \\ \vdots \\ \Delta(\Delta\zeta_{0\ 1N1}) \\ \Delta(\Delta\zeta_{1\ 1N1}) \\ \Delta(\Delta\zeta_{2\ 1N1}) \\ \Delta(\Delta\zeta_{0\ 2N1}) \\ \Delta(\Delta\zeta_{1\ 2N1}) \\ \Delta(\Delta\zeta_{2\ 2N1}) \end{pmatrix} = \begin{pmatrix} \text{source parameter updates} \\ \hline \text{attitude parameter updates} \\ \hline \text{AL} - \text{scan geom. parameter updates} \\ \hline \text{AC} - \text{scan geom. parameter updates} \end{pmatrix} \tag{20}$$

The vector of ODAS residuals, i.e. the left-hand side of the observation equations, is[6]:

$$l = \begin{pmatrix} \eta^{\mathrm{obs}}_{11} - \eta^{\mathrm{calc}}_{11} \\ \vdots \\ \eta^{\mathrm{obs}}_{1E_1} - \eta^{\mathrm{calc}}_{1E_1} \\ \zeta^{\mathrm{obs}}_{11} - \zeta^{\mathrm{calc}}_{11} \\ \vdots \\ \zeta^{\mathrm{obs}}_{1Z_1} - \zeta^{\mathrm{calc}}_{1Z_1} \\ \vdots \\ \eta^{\mathrm{obs}}_{I1} - \eta^{\mathrm{calc}}_{I1} \\ \vdots \\ \eta^{\mathrm{obs}}_{IE_I} - \eta^{\mathrm{calc}}_{IE_I} \\ \zeta^{\mathrm{obs}}_{I1} - \zeta^{\mathrm{calc}}_{I1} \\ \vdots \\ \zeta^{\mathrm{obs}}_{IZ_I} - \zeta^{\mathrm{calc}}_{IZ_I} \end{pmatrix} \tag{21}$$

[6]A different sequence may be chosen for the actual software, e.g. having the two coordinates (along and across scan) for the source CCD transits next to each other. Then the order to the design matrix must be arranged accordingly.

Then the design matrix is given by

$$
A = \begin{pmatrix}
\frac{\partial F_{\eta,11}}{\partial v_1} & \frac{\partial F_{\eta,11}}{\partial r_1} & \cdots & \frac{\partial F_{\eta,11}}{\partial v_I} & \frac{\partial F_{\eta,11}}{\partial r_I} & \frac{\partial F_{\eta,11}}{\partial p_{11}} & \cdots & \frac{\partial F_{\eta,11}}{\partial p_{U4}} & \frac{\partial F_{\eta,11}}{\partial \Delta\eta_{0\,101}} & \cdots & \frac{\partial F_{\eta,11}}{\partial \Delta\eta_{2\,2N1}} & \frac{\partial F_{\eta,11}}{\partial \Delta\zeta_{0\,101}} & \cdots & \frac{\partial F_{\eta,11}}{\partial \Delta\zeta_{2\,2N1}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial F_{\eta,1E_1}}{\partial v_1} & \frac{\partial F_{\eta,1E_1}}{\partial r_1} & \cdots & \frac{\partial F_{\eta,1E_1}}{\partial v_I} & \frac{\partial F_{\eta,1E_1}}{\partial r_I} & \frac{\partial F_{\eta,1E_1}}{\partial p_{11}} & \cdots & \frac{\partial F_{\eta,1E_1}}{\partial p_{U4}} & \frac{\partial F_{\eta,1E_1}}{\partial \Delta\eta_{0\,101}} & \cdots & \frac{\partial F_{\eta,1E_1}}{\partial \Delta\eta_{2\,2N1}} & \frac{\partial F_{\eta,1E_1}}{\partial \Delta\zeta_{0\,101}} & \cdots & \frac{\partial F_{\eta,1E_1}}{\partial \Delta\zeta_{2\,2N1}} \\
\frac{\partial F_{\zeta,11}}{\partial v_1} & \frac{\partial F_{\zeta,11}}{\partial r_1} & \cdots & \frac{\partial F_{\zeta,11}}{\partial v_I} & \frac{\partial F_{\zeta,11}}{\partial r_I} & \frac{\partial F_{\zeta,11}}{\partial p_{11}} & \cdots & \frac{\partial F_{\zeta,11}}{\partial p_{U4}} & \frac{\partial F_{\zeta,11}}{\partial \Delta\eta_{0\,101}} & \cdots & \frac{\partial F_{\zeta,11}}{\partial \Delta\eta_{2\,2N1}} & \frac{\partial F_{\zeta,11}}{\partial \Delta\zeta_{0\,101}} & \cdots & \frac{\partial F_{\eta,11}}{\partial \Delta\zeta_{2\,2N1}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial F_{\zeta,1Z_1}}{\partial v_1} & \frac{\partial F_{\zeta,1Z_1}}{\partial r_1} & \cdots & \frac{\partial F_{\zeta,1Z_1}}{\partial v_I} & \frac{\partial F_{\zeta,1Z_1}}{\partial r_I} & \frac{\partial F_{\zeta,1Z_1}}{\partial p_{11}} & \cdots & \frac{\partial F_{\zeta,1Z_1}}{\partial p_{U4}} & \frac{\partial F_{\zeta,1Z_1}}{\partial \Delta\eta_{0\,101}} & \cdots & \frac{\partial F_{\zeta,1Z_1}}{\partial \Delta\eta_{2\,2N1}} & \frac{\partial F_{\zeta,1Z_1}}{\partial \Delta\zeta_{0\,101}} & \cdots & \frac{\partial F_{\eta,1Z_1}}{\partial \Delta\zeta_{2\,2N1}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial F_{\eta,I1}}{\partial v_1} & \frac{\partial F_{\eta,I1}}{\partial r_1} & \cdots & \frac{\partial F_{\eta,I1}}{\partial v_I} & \frac{\partial F_{\eta,I1}}{\partial r_I} & \frac{\partial F_{\eta,I1}}{\partial p_{11}} & \cdots & \frac{\partial F_{\eta,I1}}{\partial p_{U4}} & \frac{\partial F_{\eta,I1}}{\partial \Delta\eta_{0\,101}} & \cdots & \frac{\partial F_{\eta,I1}}{\partial \Delta\eta_{2\,2N1}} & \frac{\partial F_{\eta,I1}}{\partial \Delta\zeta_{0\,101}} & \cdots & \frac{\partial F_{\eta,I1}}{\partial \Delta\zeta_{2\,2N1}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial F_{\eta,IE_I}}{\partial v_1} & \frac{\partial F_{\eta,IE_I}}{\partial r_1} & \cdots & \frac{\partial F_{\eta,IE_I}}{\partial v_I} & \frac{\partial F_{\eta,IE_I}}{\partial r_I} & \frac{\partial F_{\eta,IE_I}}{\partial p_{11}} & \cdots & \frac{\partial F_{\eta,IE_I}}{\partial p_{U4}} & \frac{\partial F_{\eta,IE_I}}{\partial \Delta\eta_{0\,101}} & \cdots & \frac{\partial F_{\eta,IE_I}}{\partial \Delta\eta_{2\,2N1}} & \frac{\partial F_{\eta,IE_I}}{\partial \Delta\zeta_{0\,101}} & \cdots & \frac{\partial F_{\eta,IZ_I}}{\partial \Delta\zeta_{2\,2N1}} \\
\frac{\partial F_{\zeta,I1}}{\partial v_1} & \frac{\partial F_{\zeta,I1}}{\partial r_1} & \cdots & \frac{\partial F_{\zeta,I1}}{\partial v_I} & \frac{\partial F_{\zeta,I1}}{\partial r_I} & \frac{\partial F_{\zeta,I1}}{\partial p_{11}} & \cdots & \frac{\partial F_{\zeta,I1}}{\partial p_{U4}} & \frac{\partial F_{\zeta,I1}}{\partial \Delta\eta_{0\,101}} & \cdots & \frac{\partial F_{\zeta,I1}}{\partial \Delta\eta_{2\,2N1}} & \frac{\partial F_{\zeta,I1}}{\partial \Delta\zeta_{0\,101}} & \cdots & \frac{\partial F_{\eta,I1}}{\partial \Delta\zeta_{2\,2N1}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial F_{\zeta,1Z_1}}{\partial v_1} & \frac{\partial F_{\zeta,IZ_I}}{\partial r_1} & \cdots & \frac{\partial F_{\zeta,IZ_I}}{\partial v_I} & \frac{\partial F_{\zeta,IZ_I}}{\partial r_I} & \frac{\partial F_{\zeta,IZ_I}}{\partial p_{11}} & \cdots & \frac{\partial F_{\zeta,IZ_I}}{\partial p_{U4}} & \frac{\partial F_{\zeta,IZ_I}}{\partial \Delta\eta_{0\,101}} & \cdots & \frac{\partial F_{\zeta,IZ_I}}{\partial \Delta\eta_{2\,2N1}} & \frac{\partial F_{\zeta,IZ_I}}{\partial \Delta\zeta_{0\,101}} & \cdots & \frac{\partial F_{\eta,IZ_I}}{\partial \Delta\zeta_{2\,2N1}}
\end{pmatrix} \quad (22)
$$

Note, that $\frac{\partial F_{\eta,i1}}{\partial v_s} = 0$, $\frac{\partial F_{\eta,i1}}{\partial r_s} = 0$, $\frac{\partial F_{\zeta,i1}}{\partial v_s} = 0$ and $\frac{\partial F_{\zeta,i1}}{\partial r_s} = 0$ if $i \neq s$, i.e. derivatives of $F_{\eta,i1}$ with respect to a source other than i are zero.

Moreover, all derivatives with respect to the large-scale geometric parameters are zero, if the observations do not correspond to the correct CCD or FoV. Finally, all derivatives $F_{\eta,i1}$ with respect to $\Delta\zeta_{r\ fik}$ and all derivatives $F_{\zeta,i1}$ with respect to $\Delta\eta_{r\ fij}$ are zero. r is the degree of the Legendre polynomial, f the FoV, i the source index and $j = k = 1 =$or any other number.

This leads to the blocking structure of the design matrix, indicated by Fig.9. For the source blocks and the geometric calibration blocks, the non-zero elements are indicated by the dark-gray rectangles, the zero elements are marked by light-gray rectangles.

The actual order of the observations for a given source can be chosen arbitrarily.

The full observation equation reads

$$l = A\Delta x \tag{23}$$

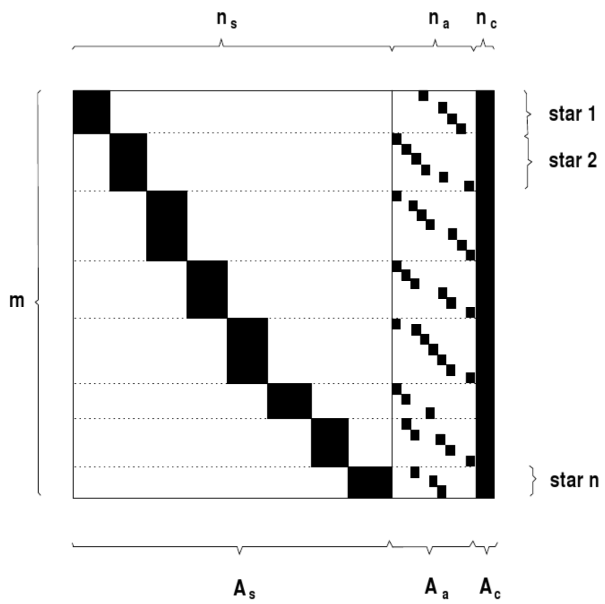

FIGURE 9: Blocking structure of the design matrix (this is Fig. 1 in Bernstein et al. (GAIA-ARI-BST-001-5). In our nomenclature, m corresponds to our L, $n_s = n$ corresponds to I, n_a corresponds to U, and n_c corresponds to $N \times 2 \times R$, where R is the highest degree of Legendre polynomials used to describe the large-scale geometric parameters per CCD).

5.7 The weight matrices for each source

Let $\sigma_{\eta_{il}}$ be the error of $\eta_{il}^{\rm obs}$, and $\sigma_{\zeta_{il}}$ be the error of $\zeta_{il}^{\rm obs}$. The weight matrix W_i associated with the source i is a diagonal matrix whose length is set by the number of observations available for the source i (see Bombrun & Löffler (GAIA-APB-001)).

Each element on the diagonal is a positive real number equal to $\sigma_{\eta_{il}}^2/\sigma_0^2$, and $\sigma_{\zeta_{il}}^2/\sigma_0^2$, respectively. σ_0 is a the unit weight error, a positive global (i.e. valid for all sources) parameter, for example 10 µas.

For the source i with E_i along-scan and Z_i across-scan measurements, W_i looks like this:

$$
W_i = \begin{pmatrix}
\sigma^2_{\eta i1}/\sigma^2_0 & 0 & 0 & 0 & 0 & 0 \\
0 & \sigma^2_{\eta i2}/\sigma^2_0 & 0 & 0 & 0 & 0 \\
\vdots & & & & & \\
0 & 0 & \sigma^2_{\eta i E_i}/\sigma^2_0 & 0 & 0 & \\
0 & 0 & 0 & \sigma^2_{\zeta i1}/\sigma^2_0 & 0 & 0 \\
0 & 0 & 0 & 0 & \sigma^2_{\zeta i2}/\sigma^2_0 & 0 \\
\vdots & & & & & \\
0 & 0 & 0 & 0 & 0 & \sigma^2_{\zeta i Z_i}/\sigma^2_0
\end{pmatrix} \tag{24}
$$

5.8 The fictitious observation equation for the quaternion normalisation

The attitude is represented by normalised quaternions (q_1, q_2, q_3, q_4) calculated from a sequence of B-spline coefficients for each quaternion component $p_{1,u}$, $p_{2,u}$, $p_{3,u}$ $p_{4,u}$, defined at the knots $t_1 \ldots t_u \ldots t_U$. (see Sect. 8).

If the B-spline coefficients are updated by ODAS without any additional constraints, there is no guarantee that the updated B-spline coefficients lead to normalised quaternions.

In ODAS, “weak constraints” are used in the form of fictitious observation equations.

For this purpose, we have to evaluate the norm of the quaternions at each B-spline knot t_u. This means that we have to determine the quaternions

$$
q_u = (q_{1u}, q_{2u}, q_{3u}, q_{4u}) = (q_1(t_u), q_2(t_u), q_3(t_u), q_4(t_u)) \tag{25}
$$

for every knot t_u.

Let

$$
N(q_u)^{\text{calc}} = \sqrt{(q^2_{1u} + q^2_{2u} + q^2_{3u} + q^2_{4u})} \tag{26}
$$

be the norm of the quaternion at t_u.

Our “fictitious observation” is:

$$
N(q_u)^{\text{obs}} = 1 \tag{27}
$$

Then the linearised observation equations in quaternions are:

$$
N(q_u)^{\text{obs}} - N(q_u)^{\text{calc}} = \sum_{d=1}^{4} \frac{\partial N(q_u)^{\text{calc}}}{\partial q_{du}} \Delta q_{du} = \sum_{d=1}^{4} 2 q_{du} \Delta q_{du}, \tag{28}
$$

where $N(q_u)^{\text{obs}}$ has to be computed using the starting values for the q_{iu} in Eq. 26.

For the ODAS, however, we need to update B-spline coefficients. The linearised observation equations in B-spline coefficients are:

$$1 - N(q_u)^{\text{calc}} = \sum_{b=u-3}^{u} \sum_{d=1}^{4} \frac{\partial N(q_b)^{\text{calc}}}{\partial p_{d,b}} \Delta p_{db} = \sum_{b=u-3}^{u} \sum_{d=1}^{4} \frac{\partial N(q_b)^{\text{calc}}}{\partial q_{du}} \frac{\partial q_{du}}{\partial p_{d,b}} \Delta p_{db} \tag{29}$$

Since (see Sect. 8)

$$\begin{pmatrix} q_1(t_u) \\ q_2(t_u) \\ q_3(t_u) \\ q_4(t_u) \end{pmatrix} = \begin{pmatrix} q_{1u} \\ q_{2u} \\ q_{3u} \\ q_{4u} \end{pmatrix} = \frac{1}{6} \begin{pmatrix} p_{1,u-3} + 4p_{1,u-2} + p_{1,u-1} \\ p_{2,u-3} + 4p_{2,u-2} + p_{2,u-1} \\ p_{3,u-3} + 4p_{3,u-2} + p_{3,u-1} \\ p_{4,u-3} + 4p_{4,u-2} + p_{4,u-1} \end{pmatrix} \tag{30}$$

the derivatives with respect to the B-spline coefficients are then:

$$\begin{pmatrix} \frac{\partial q_{1u}}{\partial p_{1,u-3}} \\ \frac{\partial q_{2u}}{\partial p_{2,u-3}} \\ \frac{\partial q_{3u}}{\partial p_{3,u-3}} \\ \frac{\partial q_{4u}}{\partial p_{4,u-3}} \end{pmatrix} = \begin{pmatrix} 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \end{pmatrix} \tag{31}$$

$$\begin{pmatrix} \frac{\partial q_{1u}}{\partial p_{1,u-2}} \\ \frac{\partial q_{2u}}{\partial p_{2,u-2}} \\ \frac{\partial q_{3u}}{\partial p_{3,u-2}} \\ \frac{\partial q_{4u}}{\partial p_{4,u-2}} \end{pmatrix} = \begin{pmatrix} 2/3 \\ 2/3 \\ 2/3 \\ 2/3 \end{pmatrix} \tag{32}$$

$$\begin{pmatrix} \frac{\partial q_{1u}}{\partial p_{1,u-1}} \\ \frac{\partial q_{2u}}{\partial p_{2,u-1}} \\ \frac{\partial q_{3u}}{\partial p_{3,u-1}} \\ \frac{\partial q_{4u}}{\partial p_{4,u-1}} \end{pmatrix} = \begin{pmatrix} 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \end{pmatrix} \tag{33}$$

$$\begin{pmatrix} \frac{\partial q_{1u}}{\partial p_{1,u}} \\ \frac{\partial q_{2u}}{\partial p_{2,u}} \\ \frac{\partial q_{3u}}{\partial p_{3,u}} \\ \frac{\partial q_{4u}}{\partial p_{4,u}} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{34}$$

Therefore the linearised observation equations in B-spline coefficients are:

$$1 - N(q_u)^{\mathrm{calc}} = \sum_{d=1}^{4} \frac{\partial N(q_b)^{\mathrm{calc}}}{\partial q_{du}} (1/6 + 2/3 + 1/6) \Delta p_{du} = \sum_{d=1}^{4} \frac{\partial N(q_b)^{\mathrm{calc}}}{\partial q_{du}} \Delta p_{du} \tag{35}$$

Using Eq. 28, the fictitious observation equation thus simply reads

$$N(q_u)^{\mathrm{obs}} - N(q_u)^{\mathrm{calc}} = \sum_{d=1}^{4} 2 q_{du} \Delta p_{du}. \tag{36}$$

In the matrix form used for constraint equations in Eq. 10 of Bombrun & Löffler (GAIA-APB-001)) this reads

$$H h_0 = r_{\mathrm{H}} \tag{37}$$

with

$$r_{\mathrm{H}} = \begin{pmatrix} 1 - N(q_1) \\ \vdots \\ 1 - N(q_u) \\ \vdots \\ 1 - N(q_U) \end{pmatrix} \tag{38}$$

and

$$H = \begin{pmatrix} 2q_{1,1} & 2q_{2,1} & 2q_{3,1} & 2q_{4,u} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2q_{1,2} & 2q_{2,2} & 2q_{3,2} & 2q_{4,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2q_{1,u} & 2q_{2,u} & 2q_{3,u} & 2q_{4,u} & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2q_{1,U} & 2q_{2,U} & 2q_{3,U} & 2q_{4,U} \end{pmatrix} \tag{39}$$

and

$$h_0 = \begin{pmatrix} p_{1,1} \\ p_{2,1} \\ p_{3,1} \\ p_{4,1} \\ p_{1,2} \\ p_{2,2} \\ p_{3,2} \\ p_{4,2} \\ \vdots \\ p_{1,u} \\ p_{2,u} \\ p_{3,u} \\ p_{4,u} \\ \vdots \\ p_{1,U} \\ p_{2,U} \\ p_{3,U} \\ p_{4,U} \end{pmatrix} \tag{40}$$

The corresponding weight matrix is

$$W = w_{\text{normalisation}} \underbrace{\begin{pmatrix} 1 & \dots & & \dots & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & & 1 & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \vdots & 0 & 1 \end{pmatrix}}_{\text{unit matrix with size } U \times U.} \tag{41}$$

According Hirte et al. (SH-004) the weight $w_{\text{normalisation}}$ can be set to about $1 \cdot 10^6$.

However, ODAS must be able to change the weight[7]. We have to test during an ODAS run, how big the largest deviation of the norm of the quaternions from 1 is.

This test can be performed after the ODAS run, when the attitude parameters (B-spline coefficients) have been updated. We have to transform the sequence of B-spline coefficients to a sequence of quaternions at the knots of the B-splines, and calculate the norm $\sqrt{(q_{1u}^2 + q_{2u}^2 + q_{3u}^2 + q_{4u}^2)}$ for all $u = 1, \dots U$. If for any of the quaternions at the knot times t_u the deviation from unity is larger than 10^{-14} the weight $w_{\text{normalisation}}$ has to be enlarged (see Fig. 10).

6 The reduced normal matrix

6.1 Definition of the normal matrix

A square matrix A is called normal, if

$$AA^{\text{H}} - A^{\text{H}}A = 0 \tag{42}$$

A^{H} denotes the conjugate transpose [8]. If A is real, $A^{\text{H}} = A^{\text{T}}$, with A^{T} being the transposed matrix.

Normal matrices arise, for example, from a normal equation [9].

[7] in property file?
[8] see e.g. http://mathworld.wolfram.com/NormalMatrix.html
[9] see e.g. http://mathworld.wolfram.com/NormalEquation.html

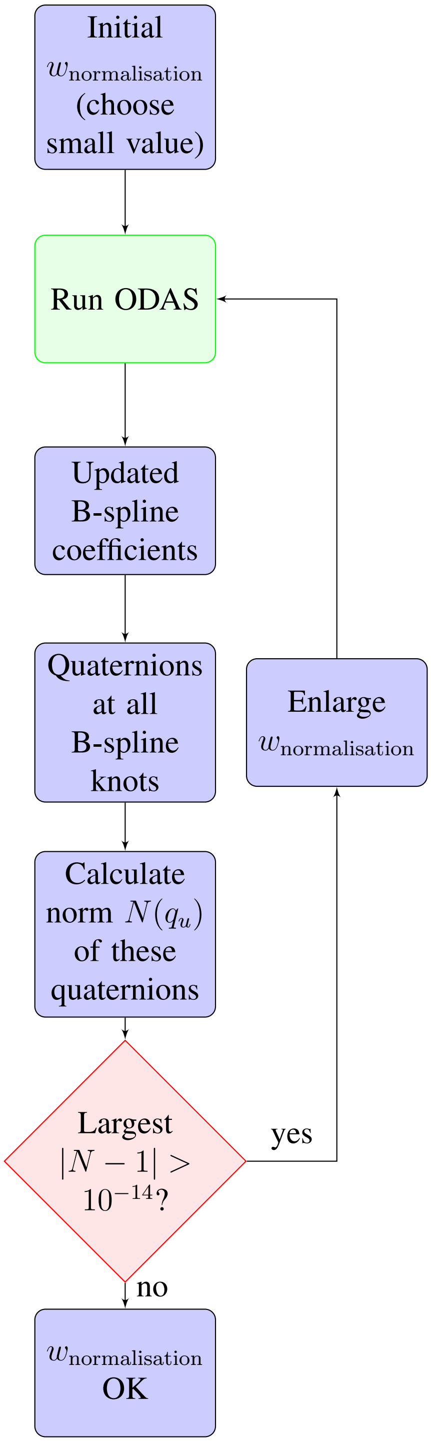

FIGURE 10: The weight $w_{\text{normalisation}}$ has to be set once. The value should be large enough to guarantee the normalisation of the quaternions, but not too large to influence the updates due to the (real) observations. After an ODAS run, we can check, whether is large enough $w_{\text{normalisation}}$ to ensure the normalisation at the level of the numerical accuracy. $w_{\text{normalisation}}$ is of the order of $3 \cdot 10^6$.

Given an overdetermined matrix equation

$$Ax = b \tag{43}$$

the normal equation is that which minimises the sum of the square differences between left and right sides:

$$A^{\mathrm{T}}Ax = A^{\mathrm{T}}b \tag{44}$$

$A^{\mathrm{T}}A$ is a normal matrix.

6.2 The block structure of the observation equation

In the case of the ODAS,, the observation equation reads (Eq. 23)

$$l = A\Delta x$$

In Sect. 5.7 we have introduced the diagonal weight matrix W_i associated with each source. If W is the matrix whose diagonal consists of all W_i for all sources, the normal matrix of our ODAS problem reads

$$A^{\mathrm{T}}Wl = A^{\mathrm{T}}WA\Delta x \tag{45}$$

Let us call the non-zero block for source i in the design matrix A (see Eq. 22)

$$S_i = \begin{pmatrix} \frac{\partial F_{\eta,i1}}{\partial v_i} & \frac{\partial F_{\eta,i1}}{\partial r_i} \\ \vdots & \vdots \\ \frac{\partial F_{\eta,iE_i}}{\partial v_i} & \frac{\partial F_{\eta,iE_i}}{\partial r_i} \\ \frac{\partial F_{\zeta,i1}}{\partial v_i} & \frac{\partial F_{\zeta,i1}}{\partial r_i} \\ \vdots & \vdots \\ \frac{\partial F_{\zeta,iZ_i}}{\partial v_i} & \frac{\partial F_{\zeta,iZ_i}}{\partial r_i} \end{pmatrix} \tag{46}$$

Let us further call the attitude and geometric calibration part of the design matrix (Eq. 22) connected to source i

$$O_i = \begin{pmatrix} \frac{\partial F_{\eta,i1}}{\partial p_{11}} & \cdots & \frac{\partial F_{\eta,i1}}{\partial p_{U4}} & \frac{\partial F_{\eta,i1}}{\partial \Delta\eta_{0\ 101}} & \cdots & \frac{\partial F_{\eta,i1}}{\partial \Delta\eta_{2\ 2N1}} & \frac{\partial F_{\eta,i1}}{\partial \Delta\zeta_{0\ 101}} & \cdots & \frac{\partial F_{\eta,i1}}{\partial \Delta\zeta_{2\ 2N1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_{\eta,iE_i}}{\partial p_{11}} & \cdots & \frac{\partial F_{\eta,iE_i}}{\partial p_{U4}} & \frac{\partial F_{\eta,iE_i}}{\partial \Delta\eta_{0\ 101}} & \cdots & \frac{\partial F_{\eta,iE_i}}{\partial \Delta\eta_{2\ 2N1}} & \frac{\partial F_{\eta,iE_i}}{\partial \Delta\zeta_{0\ 101}} & \cdots & \frac{\partial F_{\eta,iE_i}}{\partial \Delta\zeta_{2\ 2N1}} \\ \frac{\partial F_{\zeta,i1}}{\partial p_{11}} & \cdots & \frac{\partial F_{\zeta,i1}}{\partial p_{U4}} & \frac{\partial F_{\zeta,i1}}{\partial \Delta\eta_{0\ 101}} & \cdots & \frac{\partial F_{\zeta,i1}}{\partial \Delta\eta_{2\ 2N1}} & \frac{\partial F_{\zeta,i1}}{\partial \Delta\zeta_{0\ 101}} & \cdots & \frac{\partial F_{\eta,i1}}{\partial \Delta\zeta_{2\ 2N1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_{\zeta,iZ_i}}{\partial p_{11}} & \cdots & \frac{\partial F_{\zeta,iZ_i}}{\partial p_{U4}} & \frac{\partial F_{\zeta,iZ_i}}{\partial \Delta\eta_{0\ 101}} & \cdots & \frac{\partial F_{\zeta,iZ_i}}{\partial \Delta\eta_{2\ 2N1}} & \frac{\partial F_{\zeta,iZ_i}}{\partial \Delta\zeta_{0\ 101}} & \cdots & \frac{\partial F_{\eta,iZ_i}}{\partial \Delta\zeta_{2\ 2N1}} \end{pmatrix} \tag{47}$$

The left part with the derivatives of the calculated field angles with respect to the attitude parameters is very sparse, because each observation contributes to four B-spline knots only. The right part with the derivatives of the calculated along-scan and across-scan field angles with respect to the geometric calibration parameters is also sparse, because all derivatives are zero, which do not correspond to the correct CCD or FoV. Moreover, all derivatives of the along-scan field angles $F_{\eta,i1}$ with respect to the the across-scan geometric parameters $\Delta\zeta_{r\ fik}$ and all derivatives of the across-scan field angles $F_{\zeta,i1}$ with respect to the the across-scan geometric parameters $\Delta\eta_{r\ fij}$ are zero.

With these submatrices defined, the design matrix can be written as (see REFERENCE??

$$A = \begin{pmatrix} S_1 & 0 & \dots & 0 & O_1 \\ 0 & S_2 & \dots & 0 & O_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & S_n & O_I \end{pmatrix} \tag{48}$$

The vector of the ODAS updates (see Eq. 20) can be written as (Bombrun & Löffler (GAIA-APB-001))

$$\Delta x = \begin{pmatrix} \Delta\eta_1 \\ \Delta\zeta_1 \\ \vdots \\ \Delta\eta_I \\ \Delta\zeta_I \\ \hline \Delta p_{11} \\ \Delta p_{12} \\ \Delta p_{13} \\ \Delta p_{14} \\ \vdots \\ \Delta p_{U1} \\ \Delta p_{U2} \\ \Delta p_{U3} \\ \Delta p_{U4} \\ \hline \Delta(\Delta\eta_{0\ 101}) \\ \Delta(\Delta\eta_{1\ 101}) \\ \Delta(\Delta\eta_{2\ 101}) \\ \Delta(\Delta\eta_{0\ 111}) \\ \Delta(\Delta\eta_{1\ 111}) \\ \Delta(\Delta\eta_{2\ 111}) \\ \vdots \\ \Delta(\Delta\eta_{0\ 1N1}) \\ \Delta(\Delta\eta_{1\ 1N1}) \\ \Delta(\Delta\eta_{2\ 1N1}) \\ \Delta(\Delta\eta_{0\ 2N1}) \\ \Delta(\Delta\eta_{1\ 2N1}) \\ \Delta(\Delta\eta_{2\ 2N1}) \\ \hline \Delta(\Delta\zeta_{0\ 101}) \\ \Delta(\Delta\zeta_{1\ 101}) \\ \Delta(\Delta\zeta_{2\ 101}) \\ \Delta(\Delta\zeta_{0\ 111}) \\ \Delta(\Delta\zeta_{1\ 111}) \\ \Delta(\Delta\zeta_{2\ 111}) \\ \vdots \\ \Delta(\Delta\zeta_{0\ 1N1}) \\ \Delta(\Delta\zeta_{1\ 1N1}) \\ \Delta(\Delta\zeta_{2\ 1N1}) \\ \Delta(\Delta\zeta_{0\ 2N1}) \\ \Delta(\Delta\zeta_{1\ 2N1}) \\ \Delta(\Delta\zeta_{2\ 2N1}) \end{pmatrix} = \begin{pmatrix} \Delta\eta_1 \\ \Delta\zeta_1 \\ \vdots \\ \Delta\eta_I \\ \Delta\zeta_I \\ \hline \Delta h_O \end{pmatrix} = \begin{pmatrix} \Delta h_{S_1} \\ \Delta h_{S_2} \\ \vdots \\ \Delta h_{S_n} \\ \hline \Delta h_O \end{pmatrix} \tag{49}$$

Δh_O is the subvector containing all updates of the attitude and the geometric calibration parameters, Δh_{S_i} are subvectors containing both coordinate updates $\Delta\eta_i$ and $\Delta\zeta_i$.

6.3 The normal matrix of ODAS

From the observation equation of ODAS $l = A\Delta x$ we obtain the normal equation

$$\begin{aligned}
&A^{\mathrm{T}}Wl = A^{\mathrm{T}}WA\Delta x = \\
&\left(\begin{array}{cccc} S_1^{\mathrm{T}} & 0 & \dots & 0 \\ 0 & S_2^{\mathrm{T}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & S_n^{\mathrm{T}} \\ \hline O_1^{\mathrm{T}} & O_2^{\mathrm{T}} & \dots & O_I^{\mathrm{T}} \end{array}\right)
\left(\begin{array}{cccc} W_1 & 0 & \dots & 0 \\ 0 & W_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & W_I \end{array}\right)
\left(\begin{array}{c} \eta_{11}^{\mathrm{obs}} - \eta_{11}^{\mathrm{calc}} \\ \vdots \\ \eta_{1E_1}^{\mathrm{obs}} - \eta_{1E_1}^{\mathrm{calc}} \\ \zeta_{11}^{\mathrm{obs}} - \zeta_{11}^{\mathrm{calc}} \\ \vdots \\ \zeta_{1Z_1}^{\mathrm{obs}} - \zeta_{1Z_1}^{\mathrm{calc}} \\ \vdots \\ \eta_{I1}^{\mathrm{obs}} - \eta_{I1}^{\mathrm{calc}} \\ \vdots \\ \eta_{IE_I}^{\mathrm{obs}} - \eta_{IE_I}^{\mathrm{calc}} \\ \zeta_{I1}^{\mathrm{obs}} - \zeta_{I1}^{\mathrm{calc}} \\ \vdots \\ \zeta_{IZ_I}^{\mathrm{obs}} - \zeta_{IZ_I}^{\mathrm{calc}} \end{array}\right) = \\
&\left(\begin{array}{cccc} S_1^{\mathrm{T}} & 0 & \dots & 0 \\ 0 & S_2^{\mathrm{T}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & S_n^{\mathrm{T}} \\ \hline O_1^{\mathrm{T}} & O_2^{\mathrm{T}} & \dots & O_I^{\mathrm{T}} \end{array}\right)
\left(\begin{array}{cccc} W_1 & 0 & \dots & 0 \\ 0 & W_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & W_I \end{array}\right)
\left(\begin{array}{c} l_{S_1} \\ l_{S_2} \\ \vdots \\ l_{S_I} \end{array}\right) = \\
&\left(\begin{array}{cccc} S_1^{\mathrm{T}} & 0 & \dots & 0 \\ 0 & S_2^{\mathrm{T}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & S_n^{\mathrm{T}} \\ \hline O_1^{\mathrm{T}} & O_2^{\mathrm{T}} & \dots & O_I^{\mathrm{T}} \end{array}\right)
\left(\begin{array}{cccc} W_1 & 0 & \dots & 0 \\ 0 & W_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & W_I \end{array}\right)
\left(\begin{array}{cccc|c} S_1 & 0 & \dots & 0 & O_1 \\ 0 & S_2 & \dots & 0 & O_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & S_n & O_I \end{array}\right)
\left(\begin{array}{c} \Delta h_{S_1} \\ \Delta h_{S_2} \\ \vdots \\ \Delta h_{S_n} \\ \hline \Delta h_O \end{array}\right)
\end{aligned} \tag{50}$$

The subvectors l_{S_i} consist of all residuals $\eta_{il}^{\mathrm{obs}} - \eta_{il}^{\mathrm{calc}}$ and $\zeta_{il}^{\mathrm{obs}} - \zeta_{il}^{\mathrm{calc}}$ of one particular source.

The multiplications result in

$$\left(\begin{array}{c} S_1^{\mathrm{T}}W_1 l_{S_1} \\ S_2^{\mathrm{T}}W_2 l_{S_2} \\ \vdots \\ S_I^{\mathrm{T}}W_I l_{S_I} \\ \hline \sum_{i=1}^{I} O_i^{\mathrm{T}}W_i l_{S_i} \end{array}\right) =
\left(\begin{array}{cccc|c} S_1^{\mathrm{T}}W_1S_1 & 0 & \dots & 0 & S_1^{\mathrm{T}}W_1O_1 \\ 0 & S_2^{\mathrm{T}}W_2S_2 & \dots & 0 & S_2^{\mathrm{T}}W_2O_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & S_I^{\mathrm{T}}W_IS_I & S_I^{\mathrm{T}}W_IO_I \\ \hline O_1^{\mathrm{T}}W_1S_1 & O_2^{\mathrm{T}}W_2S_2 & \dots & O_I^{\mathrm{T}}W_IS_I & \sum_{i=1}^{I} O_i^{\mathrm{T}}W_iO_i \end{array}\right)
\left(\begin{array}{c} \Delta h_{S_1} \\ \Delta h_{S_2} \\ \vdots \\ \Delta h_{S_n} \\ \hline \Delta h_O \end{array}\right) \tag{51}$$

6.4 The reduced normal matrix of ODAS

With the exception of the lowest row of submatrices, the matrix $K = A^{\mathrm{T}}WA$ in Eq. 51 is already in row echelon form. This enables us to easily eliminate the source unknowns from the normal matrix. For this purpose, a Gauss elimination scheme is used, which further reduces the entire matrix to row echelon form (on the level of submatrix blocks, not on the level of single entries within the source or attitude/geometric calibration blocks).

This is done by elementary row operations: If we multiply the first row of K by $O_1^{\mathrm{T}}W_1S_1(S_1^{\mathrm{T}}W_1S_1)^{-1}$ and subtract the result from the last row, the first column of the last row is eliminated. If we further multiply the first row by $(S_1^{\mathrm{T}}W_1S_1)^{-1}$, we obtain a unity matrix 1 in the upper left corner.

If the same procedure is performed with the other rows, we obtain the following result:

$$\left(\begin{array}{c} (S_1^{\mathrm{T}}W_1S_1)^{-1}S_1^{\mathrm{T}}W_1l_{S_1} \\ (S_2^{\mathrm{T}}W_2S_2)^{-1}S_2^{\mathrm{T}}W_2l_{S_2} \\ \vdots \\ (S_I^{\mathrm{T}}W_IS_I)^{-1}S_I^{\mathrm{T}}W_Il_{S_I} \\ \hline \sum_{i=1}^{I}(O_i^{\mathrm{T}} - O_i^{\mathrm{T}}W_iS_i(S_i^{\mathrm{T}}W_iS_i)^{-1}S_i^{\mathrm{T}})W_il_{S_i} \end{array}\right) = \tag{52}$$

$$\left(\begin{array}{cccc|c} 1 & 0 & \dots & 0 & (S_1^{\mathrm{T}}W_1S_1)^{-1}S_1^{\mathrm{T}}W_1O_1 \\ 0 & 1 & \dots & 0 & (S_2^{\mathrm{T}}W_2S_2)^{-1}S_2^{\mathrm{T}}W_2O_2 \\ 0 & 0 & \ddots & 0 & \vdots \\ 0 & 0 & \dots & 1 & (S_I^{\mathrm{T}}W_IS_I)^{-1}S_I^{\mathrm{T}}W_IO_I \\ -- & -- & -- & -- & ----------------- \\ 0 & 0 & \dots & 0 & \sum_{i=1}^{I} O_i^{\mathrm{T}}W_iO_i - O_i^{\mathrm{T}}W_iS_i(S_i^{\mathrm{T}}W_iS_i)^{-1}S_i^{\mathrm{T}}W_iO_i \end{array}\right) \left(\begin{array}{c} \Delta h_{S_1} \\ \Delta h_{S_2} \\ \vdots \\ \Delta h_{S_n} \\ \hline \Delta h_O \end{array}\right)$$

The lower right corner of the matrix is the reduced normal matrix M (reduced, because the astrometric source parameters are eliminated).

The remaining system of equations for the attitude and geometric parameters is

$$b := \sum_{i=1}^{I}(O_i^{\mathrm{T}} - O_i^{\mathrm{T}}W_iS_i(S_i^{\mathrm{T}}W_iS_i)^{-1}S_i^{\mathrm{T}})W_il_{S_i} = \\ (\Delta h_O \sum_{i=1}^{I} O_i^{\mathrm{T}}W_iO_i - O_i^{\mathrm{T}}W_iS_i(S_i^{\mathrm{T}}W_iS_i)^{-1}S_i^{\mathrm{T}}W_iO_i)\Delta h_O =: M\Delta h_0 \tag{53}$$

After computing the submatrices S_i and O_i of the design matrix and after determining W_i, the reduced normal matrix can be filled source by source by adding a summand $(O_i^{\mathrm{T}}W_iO_i - O_i^{\mathrm{T}}W_iS_i(S_i^{\mathrm{T}}W_iS_i)^{-1}S_i^{\mathrm{T}}W_iO_i)$ to M. In the same way, the left-hand side is obtained by adding $((O_i^{\mathrm{T}} - O_i^{\mathrm{T}}W_iS_i(S_i^{\mathrm{T}}W_iS_i)^{-1}S_i^{\mathrm{T}})W_il_{S_i})$.

6.5 Inversion of the reduce normal matrix

Any kind of pseudo inversion can be used. A description of the actual method will follow in a future version of the document.

6.6 Backsubstitution

After the subvector Δh_0 has been obtained by solving the reduced system of equations 53, the source parameters can be obtained source-by-source (backsubstitution):

$$h_{S_i} = (S_I^{\mathrm{T}} W_I S_I)^{-1} S_I^{\mathrm{T}} W_I (l_{S_I} - O_i h_0) \tag{54}$$

7 Backrotation

The algorithmic description of the backrotation will follow in a future version.

8 Attitude representation by cubic B-splines

For Gaia, the attitude parameters are represented by B-spline coefficients.

In order to determine the four attitude quaternion components (q_1, q_2, q_3, q_4) for a given time t_ℓ^{obs}, the attitude spline coefficients from four attitude knots $u-3$, $u-2$, $u-1$, and u are needed for each of the four attitude components: $p_{1,u-3}$, $p_{2,u-3}$, $p_{3,u-3}$ $p_{4,u-3}$,$\dots p_{1,u}$ $p_{2,u}$, $p_{3,u}$ $p_{4,u}$. In total every observation at t_ℓ^{obs} contributes to 16 attitude parameters.

The sequence of time knots are t_{u-3}, t_{u-2}, t_{u-1}, and t_u, with u defined by $t_u \leq t_\ell^{\mathrm{obs}} < t_{u+1}$

Let $t = \frac{t_\ell^{\mathrm{obs}} - t_u}{t_{u+1} - t_u}$.

Then the attitude quaternion components can be determined from

$$q_1(t) = \frac{1}{6} \begin{pmatrix} t^3. & t^2 & t & 1 \end{pmatrix} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} p_{1,u-3} \\ p_{1,u-2} \\ p_{1,u-1} \\ p_{1,u} \end{pmatrix}$$
$$q_2(t) = \ldots$$
$$q_3(t) = \ldots$$
$$q_4(t) = \frac{1}{6} \begin{pmatrix} t^3. & t^2 & t & 1 \end{pmatrix} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix} \begin{pmatrix} p_{4,u-3} \\ p_{4,u-2} \\ p_{4,u-1} \\ p_{4,u} \end{pmatrix} \tag{55}$$

In non-matrix representation the interpolation formulae for the quaternions for an equidistant distribution of the B-spline knots is given by

$$\begin{aligned} q_1(t) = \frac{1}{6}[(-p_{1,u-3} + 3p_{1,u-2} - 3p_{1,u-1} + p_{1,u})t^3 + (3p_{1,u-3} - 6p_{1,u-2} + 3p_{1,u-1})t^2 + \\ (-3p_{1,u-3} + 3p_{1,u-1})t + p_{1,u-3} + 4p_{1,u-2} + p_{1,u-1}] \\ q_2(t) = \frac{1}{6}[(-p_{2,u-3} + 3p_{2,u-2} - 3p_{2,u-1} + p_{2,u})t^3 + (3p_{2,u-3} - 6p_{2,u-2} + 3p_{2,u-1})t^2 + \\ (-3p_{2,u-3} + 3p_{2,u-1})t + p_{2,u-3} + 4p_{2,u-2} + p_{2,u-1}] \\ q_3(t) = \frac{1}{6}([-p_{3,u-3} + 3p_{3,u-2} - 3p_{3,u-1} + p_{3,u})t^3 + (3p_{3,u-3} - 6p_{3,u-2} + 3p_{3,u-1})t^2 + \\ (-3p_{3,u-3} + 3p_{3,u-1})t + p_{3,u-3} + 4p_{3,u-2} + p_{3,u-1}] \\ q_4(t) = \frac{1}{6}[(-p_{4,u-3} + 3p_{4,u-2} - 3p_{4,u-1} + p_{4,u})t^3 + (3p_{4,u-3} - 6p_{4,u-2} + 3p_{4,u-1})t^2 + \\ (-3p_{4,u-3} + 3p_{4,u-1})t + p_{4,u-3} + 4p_{4,u-2} + p_{4,u-1}] \end{aligned} \tag{56}$$

This formula should give (with numerical accuaracy) the same result as the method interpolateAttitude in gaia.cu1.tools.satellite.attitude (as long as we use equidistant knots!). This has not been verified yet!

After the interpolation, the quaternions should be normalised. With the norm

$$\sqrt{q_1(t)^2 + q_2(t)^2 + q_3(t)^2 + q_4(t)^2} \tag{57}$$

it follows:

$$\begin{aligned} q_{1,\text{norm}}(t) &= q_1(t)/\sqrt{q_1(t)^2+q_2(t)^2+q_3(t)^2+q_4(t)^2} \\ q_{2,\text{norm}}(t) &= q_2(t)/\sqrt{q_1(t)^2+q_2(t)^2+q_3(t)^2+q_4(t)^2} \\ q_{3,\text{norm}}(t) &= q_3(t)/\sqrt{q_1(t)^2+q_2(t)^2+q_3(t)^2+q_4(t)^2} \\ q_{4,\text{norm}}(t) &= q_4(t)/\sqrt{q_1(t)^2+q_2(t)^2+q_3(t)^2+q_4(t)^2} \end{aligned} \tag{58}$$

In order to perform the interpolation, we need three supporting knots with spline coefficients before the actual interpolation interval. Therefore, all subintervals which are read in or written out should contain these three extra spline coefficients in front of the interval defined by $[t_{\text{begin}}, t_{\text{end}})$. Note, that the IOGA delivered by IDT 5.0 is not consistent with this definition.

9 Transforming the attitude B-spline coefficients and derivatives with respect to B-spline coefficients from ICRS to RGCS and vice versa

9.1 Transformation of quaternions

The transformation between the ICRS and the RGCS is done with the help of the RGCS-defining quaternion $q^{\text{mid ICRS}} = q^{\text{ICRS}}(t = (T_{\text{end}} - T_{\text{beg}})/2)$, the attitude quaternion in ICRS approximately in the middle of an ODAS day. The actual choice of the time for which the RGCS-defining quaternion is evaluated is not important. Any time that is reasonably close to the middle of the ODAS time interval $[T_{\text{beg}}, T_{\text{end}}]$ is OK.

An attitude quaternion is transformed from ICRS to RGCS by

$$q^{\text{RGCS}}(t) = q^{\text{mid ICRS}*} \cdot q^{\text{ICRS}}(t) \tag{59}$$

The $q^* = (-q_1, -q_2, -q_3, q_4)$ denotes the conjugate of the quaternion $q = (q_1, q_2, q_3, q_4)$ and the dot "·" the quaternion product.

If we write down the quaternion components explicitly, we obtain

$$\begin{pmatrix} q_1^{\text{RGCS}} \\ q_2^{\text{RGCS}} \\ q_3^{\text{RGCS}} \\ q_4^{\text{RGCS}} \end{pmatrix} = \begin{pmatrix} -q_1^{\text{mid ICRS}} \\ -q_2^{\text{mid ICRS}} \\ -q_3^{\text{mid ICRS}} \\ q_4^{\text{mid ICRS}} \end{pmatrix} \cdot \begin{pmatrix} q_1^{\text{ICRS}} \\ q_2^{\text{ICRS}} \\ q_3^{\text{ICRS}} \\ q_4^{\text{ICRS}} \end{pmatrix} = \begin{pmatrix} -q_1^{\text{mid ICRS}} q_4^{\text{ICRS}} - q_2^{\text{mid ICRS}} q_3^{\text{ICRS}} + q_3^{\text{mid ICRS}} q_2^{\text{ICRS}} + q_4^{\text{mid ICRS}} q_1^{\text{ICRS}} \\ q_1^{\text{mid ICRS}} q_3^{\text{ICRS}} - q_2^{\text{mid ICRS}} q_4^{\text{ICRS}} - q_3^{\text{mid ICRS}} q_1^{\text{ICRS}} + q_4^{\text{mid ICRS}} q_2^{\text{ICRS}} \\ -q_1^{\text{mid ICRS}} q_2^{\text{ICRS}} + q_2^{\text{mid ICRS}} q_1^{\text{ICRS}} - q_3^{\text{mid ICRS}} q_4^{\text{ICRS}} + q_4^{\text{mid ICRS}} q_3^{\text{ICRS}} \\ q_1^{\text{mid ICRS}} q_1^{\text{ICRS}} + q_2^{\text{mid ICRS}} q_2^{\text{ICRS}} + q_3^{\text{mid ICRS}} q_3^{\text{ICRS}} + q_4^{\text{mid ICRS}} q_4^{\text{ICRS}} \end{pmatrix} \tag{60}$$

In matrix form this reads

$$\begin{pmatrix} q_1^{\mathrm{RGCS}} \\ q_2^{\mathrm{RGCS}} \\ q_3^{\mathrm{RGCS}} \\ q_4^{\mathrm{RGCS}} \end{pmatrix} = \begin{pmatrix} -q_1^{\mathrm{mid\ ICRS}} \\ -q_2^{\mathrm{mid\ ICRS}} \\ -q_3^{\mathrm{mid\ ICRS}} \\ q_4^{\mathrm{mid\ ICRS}} \end{pmatrix} \cdot \begin{pmatrix} q_1^{\mathrm{ICRS}} \\ q_2^{\mathrm{ICRS}} \\ q_3^{\mathrm{ICRS}} \\ q_4^{\mathrm{ICRS}} \end{pmatrix} = \begin{pmatrix} q_4^{\mathrm{ICRS}} & q_3^{\mathrm{ICRS}} & -q_2^{\mathrm{ICRS}} & q_1^{\mathrm{ICRS}} \\ -q_3^{\mathrm{ICRS}} & q_4^{\mathrm{ICRS}} & q_1^{\mathrm{ICRS}} & q_2^{\mathrm{ICRS}} \\ q_2^{\mathrm{ICRS}} & -q_1^{\mathrm{ICRS}} & q_4^{\mathrm{ICRS}} & q_3^{\mathrm{ICRS}} \\ -q_1^{\mathrm{ICRS}} & -q_2^{\mathrm{ICRS}} & -q_3^{\mathrm{ICRS}} & q_4^{\mathrm{ICRS}} \end{pmatrix} \begin{pmatrix} -q_1^{\mathrm{mid\ ICRS}} \\ -q_2^{\mathrm{mid\ ICRS}} \\ -q_3^{\mathrm{mid\ ICRS}} \\ q_4^{\mathrm{mid\ ICRS}} \end{pmatrix} \tag{61}$$

Analogously, an attitude quaternion is transformed from RGCS to ICRS by

$$q^{\mathrm{ICRS}}(t) = q^{\mathrm{mid\ ICRS}} \cdot q^{\mathrm{RGCS}}(t) \tag{62}$$

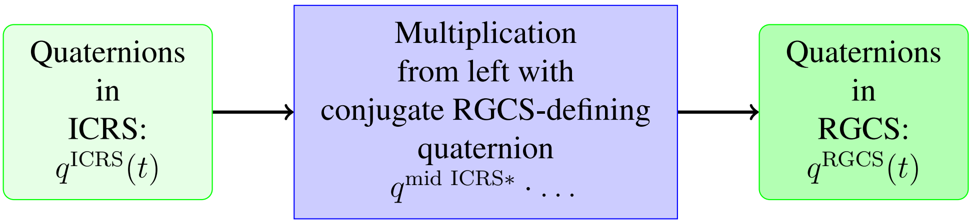

FIGURE 11: Method to transform quaternions from ICRS to RGCS: $q^{\mathrm{RGCS}}(t) = q^{\mathrm{mid\ ICRS}*} \cdot q^{\mathrm{ICRS}}(t)$.

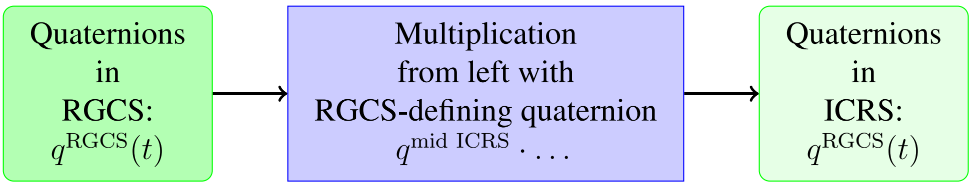

FIGURE 12: Method to transform quaternions from RGCS to ICRS: $q^{\mathrm{ICRS}}(t) = q^{\mathrm{trans}} \cdot q^{\mathrm{RGCS}}(t)$.

9.2 Transformation of B-spline coefficients

In the prototype versions of the ODAS software, the transformation between the B-spline coefficients from the ICRS to RGCS was performed by first calculating a grid of quaternions from the B-spline coefficients defined in the ICRS, then transforming the quaternions into the RGCS with the method described in Sect. 9.1, and then fitting B-splines to the RGCS quaternions.

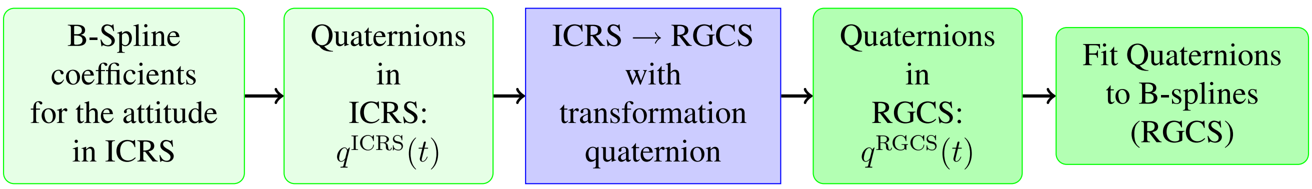

FIGURE 13: Method to transform attitude B-Spline coefficients defined in the ICRS to attitude B-Spline coefficients defined in the RGCS by using quaternions as intermediate values.

This method is the only useful method if the B-Spline coefficients in the two systems are defined on a different sequence of knots.

At the time the first versions of the ODAS were written, we did not recognize that one can directly transform B-spline coefficients with the same method as quaternions.

The reason is the linearity of the B-splines in terms of the B-spline coefficients.

The set of B-spline coefficients $p_u = (p_{1u}, p_{2u}, p_{3u}, p_{4u})$ with the knot index u is transformed from ICRS to RGCS by

$$p_u^{\mathrm{RGCS}} = q^{\mathrm{mid\ ICRS}*} \cdot p_u^{\mathrm{ICRS}}, \tag{63}$$

where $q^{\mathrm{mid\ ICRS}*}$ is the conjugate RGCS-defining quaternion and the $\cdot$ the quaternion product.

Analogously, the set of B-spline coefficients is transformed from RGCS to ICRS by

$$p_u^{\mathrm{ICRS}} = q^{\mathrm{mid\ ICRS}} \cdot p_u^{\mathrm{RGCS}} \tag{64}$$

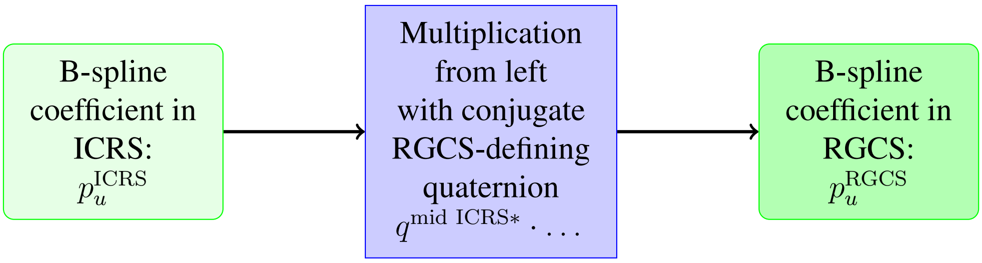

FIGURE 14: Method to transform attitude B-Spline coefficients from ICRS to RGCS by directly rotating with the transformation quaternion (conjugate RGCS-defining quaternion).

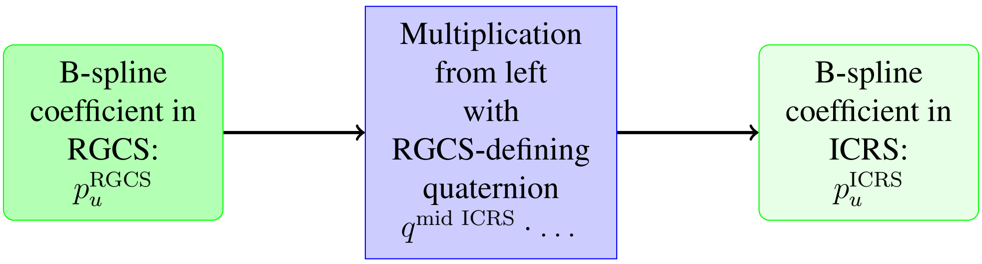

FIGURE 15: Transformation of attitude B-Spline coefficients from RGCS to ICRS.

In Appendix A it is shown that both methods are also numerically equivalent down to the level of computational accuracy.

9.3 Transformation of derivatives with respect to B-spline coefficients from ICRS to RGCS

Let f be a function, for which the partial derivatives are provided with respect to the B-spline coefficients $p_u^{\rm ICRS} = (p_{1u}^{\rm ICRS}, p_{2u}^{\rm ICRS}, p_{3u}^{\rm ICRS}, p_{4u}^{\rm ICRS})$:

$$\frac{\partial f}{\partial p_u^{\rm ICRS}} = \begin{pmatrix} \frac{\partial f}{\partial p_{1u}^{\rm ICRS}} \\ \frac{\partial f}{\partial p_{2u}^{\rm ICRS}} \\ \frac{\partial f}{\partial p_{3u}^{\rm ICRS}} \\ \frac{\partial f}{\partial p_{4u}^{\rm ICRS}} \end{pmatrix} \tag{65}$$

where u is the index of the knot.

The derivatives with respect to the B-spline coefficients in RGCS are

$$\frac{\partial f}{\partial p_u^{\rm RGCS}} = \begin{pmatrix} \frac{\partial f}{\partial p_{1u}^{\rm RGCS}} \\ \frac{\partial f}{\partial p_{2u}^{\rm RGCS}} \\ \frac{\partial f}{\partial p_{3u}^{\rm RGCS}} \\ \frac{\partial f}{\partial p_{4u}^{\rm RGCS}} \end{pmatrix} = \frac{\partial f}{\partial p_u^{\rm ICRS}} \frac{\partial p_u^{\rm ICRS}}{\partial p_u^{\rm RGCS}} = \begin{pmatrix} \frac{\partial f}{\partial p_{1u}^{\rm ICRS}} \frac{\partial p_{1u}^{\rm ICRS}}{\partial p_{1u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{2u}^{\rm ICRS}} \frac{\partial p_{2u}^{\rm ICRS}}{\partial p_{1u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{3u}^{\rm ICRS}} \frac{\partial p_{3u}^{\rm ICRS}}{\partial p_{1u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{4u}^{\rm ICRS}} \frac{\partial p_{4u}^{\rm ICRS}}{\partial p_{1u}^{\rm RGCS}} \\ \frac{\partial f}{\partial p_{1u}^{\rm ICRS}} \frac{\partial p_{1u}^{\rm ICRS}}{\partial p_{2u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{2u}^{\rm ICRS}} \frac{\partial p_{2u}^{\rm ICRS}}{\partial p_{2u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{3u}^{\rm ICRS}} \frac{\partial p_{3u}^{\rm ICRS}}{\partial p_{2u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{4u}^{\rm ICRS}} \frac{\partial p_{4u}^{\rm ICRS}}{\partial p_{2u}^{\rm RGCS}} \\ \frac{\partial f}{\partial p_{1u}^{\rm ICRS}} \frac{\partial p_{1u}^{\rm ICRS}}{\partial p_{3u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{2u}^{\rm ICRS}} \frac{\partial p_{2u}^{\rm ICRS}}{\partial p_{3u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{3u}^{\rm ICRS}} \frac{\partial p_{3u}^{\rm ICRS}}{\partial p_{3u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{4u}^{\rm ICRS}} \frac{\partial p_{4u}^{\rm ICRS}}{\partial p_{3u}^{\rm RGCS}} \\ \frac{\partial f}{\partial p_{1u}^{\rm ICRS}} \frac{\partial p_{1u}^{\rm ICRS}}{\partial p_{4u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{2u}^{\rm ICRS}} \frac{\partial p_{2u}^{\rm ICRS}}{\partial p_{4u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{3u}^{\rm ICRS}} \frac{\partial p_{3u}^{\rm ICRS}}{\partial p_{4u}^{\rm RGCS}} + \frac{\partial f}{\partial p_{4u}^{\rm ICRS}} \frac{\partial p_{4u}^{\rm ICRS}}{\partial p_{4u}^{\rm RGCS}} \end{pmatrix} \tag{66}$$

Since

$$p_u^{\rm ICRS} = q^{\rm mid\ ICRS} \cdot p_u^{\rm RGCS} \tag{67}$$

$$\partial \begin{pmatrix} p_{1u}^{\rm ICRS} \\ p_{2u}^{\rm ICRS} \\ p_{3u}^{\rm ICRS} \\ p_{4u}^{\rm ICRS} \end{pmatrix} / \partial p_u^{\rm RGCS} = \partial \begin{pmatrix} +q_1^{\rm mid\ ICRS} p_{4u}^{\rm RGCS} + q_2^{\rm mid\ ICRS} p_{3u}^{\rm RGCS} - q_3^{\rm mid\ ICRS} p_{2u}^{\rm RGCS} + q_4^{\rm mid\ ICRS} p_{1u}^{\rm RGCS} \\ -q_1^{\rm mid\ ICRS} p_{3u}^{\rm RGCS} + q_2^{\rm mid\ ICRS} p_{4u}^{\rm RGCS} + q_3^{\rm mid\ ICRS} p_{1u}^{\rm RGCS} + q_4^{\rm mid\ ICRS} p_{2u}^{\rm RGCS} \\ +q_1^{\rm mid\ ICRS} p_{2u}^{\rm RGCS} - q_2^{\rm mid\ ICRS} p_{1u}^{\rm RGCS} + q_3^{\rm mid\ ICRS} p_{4u}^{\rm RGCS} + q_4^{\rm mid\ ICRS} p_{3u}^{\rm RGCS} \\ -q_1^{\rm mid\ ICRS} p_{1u}^{\rm RGCS} - q_2^{\rm mid\ ICRS} p_{2u}^{\rm RGCS} - q_3^{\rm mid\ ICRS} p_{3u}^{\rm RGCS} + q_4^{\rm mid\ ICRS} p_{4u}^{\rm RGCS} \end{pmatrix} / \partial p_u^{\rm RGCS} \tag{68}$$

$$\Rightarrow \begin{pmatrix} \frac{\partial p_{1u}^{\rm ICRS}}{\partial p_{1u}^{\rm RGCS}} \\ \frac{\partial p_{1u}^{\rm ICRS}}{\partial p_{2u}^{\rm RGCS}} \\ \frac{\partial p_{1u}^{\rm ICRS}}{\partial p_{3u}^{\rm RGCS}} \\ \frac{\partial p_{1u}^{\rm ICRS}}{\partial p_{4u}^{\rm RGCS}} \end{pmatrix} = \begin{pmatrix} +q_4^{\rm mid\ ICRS} \\ +q_3^{\rm mid\ ICRS} \\ -q_2^{\rm mid\ ICRS} \\ -q_1^{\rm mid\ ICRS} \end{pmatrix} \tag{69}$$

In the same way it follows

$$\begin{pmatrix} \frac{\partial p_{2u}^{\mathrm{ICRS}}}{\partial p_{1u}^{\mathrm{RGCS}}} \\ \frac{\partial p_{2u}^{\mathrm{ICRS}}}{\partial p_{2u}^{\mathrm{RGCS}}} \\ \frac{\partial p_{2u}^{\mathrm{ICRS}}}{\partial p_{3u}^{\mathrm{RGCS}}} \\ \frac{\partial p_{2u}^{\mathrm{ICRS}}}{\partial p_{4u}^{\mathrm{RGCS}}} \end{pmatrix} = \begin{pmatrix} -q_3^{\mathrm{mid\ ICRS}} \\ +q_4^{\mathrm{mid\ ICRS}} \\ +q_1^{\mathrm{mid\ ICRS}} \\ -q_2^{\mathrm{mid\ ICRS}} \end{pmatrix} \tag{70}$$

$$\begin{pmatrix} \frac{\partial p_{3u}^{\mathrm{ICRS}}}{\partial p_{1u}^{\mathrm{RGCS}}} \\ \frac{\partial p_{3u}^{\mathrm{ICRS}}}{\partial p_{2u}^{\mathrm{RGCS}}} \\ \frac{\partial p_{3u}^{\mathrm{ICRS}}}{\partial p_{3u}^{\mathrm{RGCS}}} \\ \frac{\partial p_{3u}^{\mathrm{ICRS}}}{\partial p_{4u}^{\mathrm{RGCS}}} \end{pmatrix} = \begin{pmatrix} +q_2^{\mathrm{mid\ ICRS}} \\ -q_1^{\mathrm{mid\ ICRS}} \\ +q_4^{\mathrm{mid\ ICRS}} \\ -q_3^{\mathrm{mid\ ICRS}} \end{pmatrix} \tag{71}$$

$$\begin{pmatrix} \frac{\partial p_{4u}^{\mathrm{ICRS}}}{\partial p_{1u}^{\mathrm{RGCS}}} \\ \frac{\partial p_{4u}^{\mathrm{ICRS}}}{\partial p_{2u}^{\mathrm{RGCS}}} \\ \frac{\partial p_{4u}^{\mathrm{ICRS}}}{\partial p_{3u}^{\mathrm{RGCS}}} \\ \frac{\partial p_{4u}^{\mathrm{ICRS}}}{\partial p_{4u}^{\mathrm{RGCS}}} \end{pmatrix} = \begin{pmatrix} +q_1^{\mathrm{mid\ ICRS}} \\ +q_2^{\mathrm{mid\ ICRS}} \\ +q_3^{\mathrm{mid\ ICRS}} \\ q_4^{\mathrm{mid\ ICRS}} \end{pmatrix} \tag{72}$$

Therefore, the derivatives with respect to the B-spline coefficients in RGCS are

$$\begin{pmatrix} \frac{\partial f}{\partial p_{1u}^{\mathrm{RGCS}}} \\ \frac{\partial f}{\partial p_{2u}^{\mathrm{RGCS}}} \\ \frac{\partial f}{\partial p_{3u}^{\mathrm{RGCS}}} \\ \frac{\partial f}{\partial p_{4u}^{\mathrm{RGCS}}} \end{pmatrix} = \begin{pmatrix} +\frac{\partial f}{\partial p_{1u}^{\mathrm{ICRS}}} q_4^{\mathrm{mid\ ICRS}} - \frac{\partial f}{\partial p_{2u}^{\mathrm{ICRS}}} q_3^{\mathrm{mid\ ICRS}} + \frac{\partial f}{\partial p_{3u}^{\mathrm{ICRS}}} q_2^{\mathrm{mid\ ICRS}} + \frac{\partial f}{\partial p_{4u}^{\mathrm{ICRS}}} q_1^{\mathrm{mid\ ICRS}} \\ +\frac{\partial f}{\partial p_{1u}^{\mathrm{ICRS}}} q_3^{\mathrm{mid\ ICRS}} + \frac{\partial f}{\partial p_{2u}^{\mathrm{ICRS}}} q_4^{\mathrm{mid\ ICRS}} - \frac{\partial f}{\partial p_{3u}^{\mathrm{ICRS}}} q_1^{\mathrm{mid\ ICRS}} + \frac{\partial f}{\partial p_{4u}^{\mathrm{ICRS}}} q_2^{\mathrm{mid\ ICRS}} \\ -\frac{\partial f}{\partial p_{1u}^{\mathrm{ICRS}}} q_2^{\mathrm{mid\ ICRS}} + \frac{\partial f}{\partial p_{2u}^{\mathrm{ICRS}}} q_1^{\mathrm{mid\ ICRS}} + \frac{\partial f}{\partial p_{3u}^{\mathrm{ICRS}}} q_4^{\mathrm{mid\ ICRS}} + \frac{\partial f}{\partial p_{4u}^{\mathrm{ICRS}}} q_3^{\mathrm{mid\ ICRS}} \\ -\frac{\partial f}{\partial p_{1u}^{\mathrm{ICRS}}} q_1^{\mathrm{mid\ ICRS}} - \frac{\partial f}{\partial p_{2u}^{\mathrm{ICRS}}} q_2^{\mathrm{mid\ ICRS}} - \frac{\partial f}{\partial p_{3u}^{\mathrm{ICRS}}} q_3^{\mathrm{mid\ ICRS}} + \frac{\partial f}{\partial p_{4u}^{\mathrm{ICRS}}} q_4^{\mathrm{mid\ ICRS}} \end{pmatrix} \tag{73}$$

10 Transforming the astrometric source parameters and derivatives with respect to the source parameters from ICRS to RGCS

10.1 Transformation of source parameters

In order to perform the transformation with the help of the transformation quaternion (conjugate RGCS-defining quaternion), the source positions α and δ in the ICRS must first be transformed into a quaternion:

$$q_{\rm source}^{\rm ICRS} = \begin{pmatrix} \cos\delta\cos\alpha \\ \cos\delta\sin\alpha \\ \sin\delta \\ 0 \end{pmatrix} \tag{74}$$

The corresponding quaternion in the RGCS is then

$$q_{\rm source}^{\rm RGCS} = \left[\begin{pmatrix} -q_1^{\rm mid\ ICRS} \\ -q_2^{\rm mid\ ICRS} \\ -q_3^{\rm mid\ ICRS} \\ q_4^{\rm mid\ ICRS} \end{pmatrix} \cdot \begin{pmatrix} \cos\delta\cos\alpha \\ \cos\delta\sin\alpha \\ \sin\delta \\ 0 \end{pmatrix} \right] \cdot \begin{pmatrix} q_1^{\rm mid\ ICRS} \\ q_2^{\rm mid\ ICRS} \\ q_3^{\rm mid\ ICRS} \\ q_4^{\rm mid\ ICRS} \end{pmatrix} = \begin{pmatrix} \cos r\cos\upsilon \\ \cos r\sin\upsilon \\ \sin r \\ 0 \end{pmatrix} \tag{75}$$

Again, the dot "·" denotes the quaternion product.

From this, υ and r can be calculated:

$$\upsilon = \mathrm{atan2}(\mathrm{q}_2^{\rm RGCS}, \mathrm{q}_1^{\rm RGCS}) \tag{76}$$

and

$$r = \arcsin q_3^{\rm RGCS} \tag{77}$$

In the same way the proper motion μ_α^* and μ_δ in the ICRS can be transformed into a quaternion:

$$q_\mu^{\rm ICRS} = \begin{pmatrix} -\mu_\delta\sin\delta\cos\alpha - (\mu_\alpha^*/\cos\delta)\sin\alpha\cos\delta \\ -\mu_\delta\sin\delta\sin\alpha + (\mu_\alpha^*/\cos\delta)\cos\alpha\cos\delta \\ +\mu_\delta\cos\delta \\ 0 \end{pmatrix} = \begin{pmatrix} -\mu_\delta\sin\delta\cos\alpha - \mu_\alpha^*\sin\alpha \\ -\mu_\delta\sin\delta\sin\alpha + \mu_\alpha^*\cos\alpha \\ +\mu_\delta\cos\delta \\ 0 \end{pmatrix} \tag{78}$$

since

$$\begin{aligned} \frac{d(\cos\delta\cos\alpha)}{dt} &= -\dot\delta\sin\delta\cos\alpha - \dot\alpha\sin\alpha\cos\delta = -\mu_\delta\sin\delta\cos\alpha - \mu_\alpha^*\sin\alpha \\ \frac{d(\cos\delta\sin\alpha)}{dt} &= -\dot\delta\sin\delta\sin\alpha + \dot\alpha\cos\alpha\cos\delta = -\mu_\delta\sin\delta\sin\alpha + \mu_\alpha^*\cos\alpha \\ \frac{d(\sin\delta)}{dt} &= \dot\delta\cos\delta = \mu_\delta\cos\delta \end{aligned} \tag{79}$$

The quaternion of the proper motion in the RGCS is then

$$q_\mu^{\mathrm{RGCS}} = \left[\begin{pmatrix} -q_1^{\mathrm{mid\ ICRS}} \\ -q_2^{\mathrm{mid\ ICRS}} \\ -q_3^{\mathrm{mid\ ICRS}} \\ q_4^{\mathrm{mid\ ICRS}} \end{pmatrix} \cdot \begin{pmatrix} -\sin\delta\cos\alpha\mu_\delta - \sin\alpha\mu_\alpha^* \\ -\sin\delta\sin\alpha\mu_\delta + \cos\alpha\mu_\alpha^* \\ +\cos\delta\mu_\delta \\ 0 \end{pmatrix} \right] \cdot \begin{pmatrix} q_1^{\mathrm{mid\ ICRS}} \\ q_2^{\mathrm{mid\ ICRS}} \\ q_3^{\mathrm{mid\ ICRS}} \\ q_4^{\mathrm{mid\ ICRS}} \end{pmatrix} \tag{80}$$

From this we obtain

$$\mu_r = q_3^{\mathrm{RGCS}} / \cos r \tag{81}$$

and

$$\mu_\upsilon^* = \frac{q_1^{\mathrm{RGCS}} - \mu_r \sin r \cos\upsilon}{\sin\upsilon} = \frac{q_1^{\mathrm{RGCS}} \cos r - q_3^{\mathrm{RGCS}} \sin r \cos\upsilon}{\sin\upsilon\cos r} \tag{82}$$

The parallaxes ϖ are the same in ICRS and RGCS.

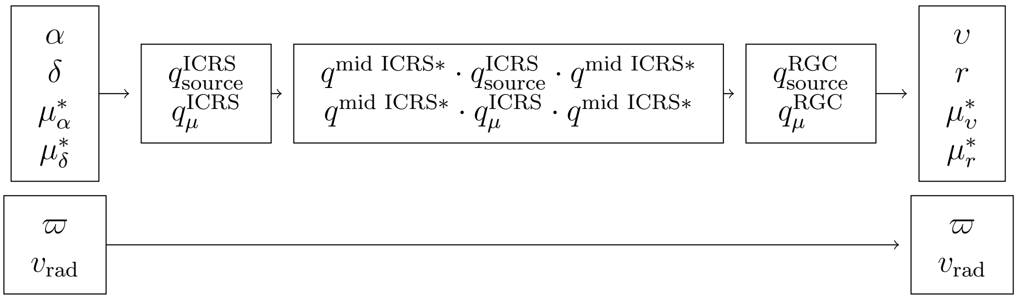

FIGURE 16: Transformation of source parameters in ICRS to RGCS.

10.2 Transformation of derivatives with respect to source parameters

In ODAS, only the source positions are updated. Therefore, only the derivatives to the two positional source parameters are regarded here.

The transformation between coordinate differences in ICRS and RGCS is mathematically the same as the transformation between local celestial coordinates (a, d) and local scan coordinates (w, z) which is given by the position angle of the scan θ (see Fig. 1 in Lindegren & Bastian (GAIA-LL-061-3)[10]).

Let $\boldsymbol{z}^{\mathrm{RGCS}} = (0, 0, 1)$ be the unit vector, pointing in the polar direction of the RGCS ($r = \pi/2$) and let $\boldsymbol{z}^{\mathrm{ICRS}} = (z_1, z_2, z_3)$ be this vector transformed into the ICRS.

Written in quaternions it follows that

[10] Forthcoming version, so that the numbers of the equations are not finally known yet.

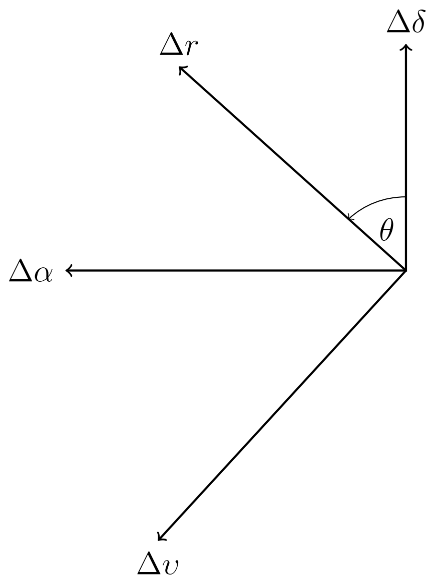

FIGURE 17: Rotation of coordinate differences in RGCS with respect to ICRS.

$$q_z^{\mathrm{RGCS}} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ 0 \end{pmatrix} = \left[\begin{pmatrix} q_1^{\text{mid ICRS}} \\ q_2^{\text{mid ICRS}} \\ q_3^{\text{mid ICRS}} \\ q_4^{\text{mid ICRS}} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \right] \cdot \begin{pmatrix} -q_1^{\text{mid ICRS}} \\ -q_2^{\text{mid ICRS}} \\ -q_3^{\text{mid ICRS}} \\ q_4^{\text{mid ICRS}} \end{pmatrix} \tag{83}$$

Analogous to Eq. xx to yy in Lindegren & Bastian (GAIA-LL-061-3)), the rotation angle θ between Δr and $\Delta\delta$ can be determined from

$$\theta = \text{atan2}(\boldsymbol{q}_0 \cdot \boldsymbol{z}^{\mathrm{ICRS}}, -\boldsymbol{p}_0 \cdot \boldsymbol{z}^{\mathrm{ICRS}}) \tag{84}$$

In this equation, the dot "·" is the scalar product between the vectors, and $\boldsymbol{q_0}$ and $\boldsymbol{p_0}$ are the local tangential unit vectors on the celestial sphere, pointing in the direction of increasing α and increasing δ, respectively, defined by

$$\boldsymbol{q_0} = \begin{pmatrix} -\sin\delta\cos\alpha \\ -\sin\delta\sin\alpha \\ \cos\delta \end{pmatrix} \tag{85}$$

and

$$\boldsymbol{p_0} = \begin{pmatrix} -\sin\alpha \\ -\cos\alpha \\ 0 \end{pmatrix} \tag{86}$$

From Eq. zz from Lindegren & Bastian (GAIA-LL-061-3):

$$\Delta\upsilon = \Delta\alpha\cos\theta + \Delta\delta\sin\theta$$
$$\Delta r = \Delta\delta\cos\theta - \Delta\alpha\sin\theta \tag{87}$$

we obtain the inverse transformation

$$\Delta\alpha = \Delta\upsilon\cos\theta - \Delta r\sin\theta$$
$$\Delta\delta = \Delta r\cos\theta + \Delta\upsilon\sin\theta \tag{88}$$

and it follows

$$\frac{\partial\alpha}{\partial\upsilon} = +\cos\theta$$
$$\frac{\partial\alpha}{\partial r} = -\sin\theta$$
$$\frac{\partial\delta}{\partial\upsilon} = +\sin\theta$$
$$\frac{\partial\delta}{\partial r} = +\cos\theta \tag{89}$$

Now, derivatives of a function f with respect to the ICRS coordinates α and δ can be converted to derivatives with respect to the RGCS coordinates υ and r:

$$\frac{\partial f}{\partial\upsilon} = \frac{\partial f}{\partial\alpha}\frac{\partial\alpha}{\partial\upsilon} + \frac{\partial f}{\partial\delta}\frac{\partial\delta}{\partial\upsilon} = \frac{\partial f}{\partial\alpha}\cos\theta - \frac{\partial f}{\partial\delta}\sin\theta$$
$$\frac{\partial f}{\partial r} = \frac{\partial f}{\partial\alpha}\frac{\partial\alpha}{\partial r} + \frac{\partial f}{\partial\delta}\frac{\partial\delta}{\partial r} = +\frac{\partial f}{\partial\alpha}\sin\theta + \frac{\partial f}{\partial\delta}\cos\theta \tag{90}$$

11 Converting proper directions and their derivatives into field angles and their derivatives

The Gaia Toolbox Routine toProperSourceDirection(AstrometricSource stellarObject, boolean withDerivatives) [11] calculates proper directions and and their derivatives with respect to the

[11] public gaia.cu1.tools.astro.refsystem.ExtendedVector gaia.cu1.tools.astro.refsystem.BarycentricReferenceFrame.toProperSourc
gaia.cu1.mdb.cu1.dm.AstrometricSource stellarObject, boolean withDerivatives) throws GaiaException

source parameters for a given time $t_\ell^{\rm obs}$ and given astrometric parameters ($\alpha^{\rm ICRS}$,$\delta^{\rm ICRS}$, $\mu_\alpha^{\rm ICRS}$, $\mu_\delta^{\rm ICRS}$, ϖ).

For the ODAS we need to determine the calculated field angles $F_\eta(t_\ell^{\rm obs}, \boldsymbol{s}_i, \boldsymbol{a}(t_\ell^{\rm obs}))$ and $F_\zeta(t_\ell^{\rm obs}, \boldsymbol{s}_i, \boldsymbol{a}(t_\ell^{\rm obs}))$ and their derivatives with respect to the source positions and the attitude parameters.

The first step is to calculate the attitude quaternion $q_\ell = q(t_\ell^{\rm obs}) = (q_{\ell 1}, q_{\ell 2}, q_{\ell 3}, q_{\ell 4})$.

From this we determine the attitude matrix (see Eq. 13 in Lindegren (LL-063)):

$$A(q_\ell) = \begin{pmatrix} q_{\ell 1}^2 - q_{\ell 2}^2 - q_{\ell 3}^2 - q_{\ell 4}^2 & 2(q_{\ell 1}q_{\ell 2} + q_{\ell 3}q_{\ell 4}) & 2(q_{\ell 1}q_{\ell 3} - q_{\ell 2}q_{\ell 4}) \\ 2(q_{\ell 1}q_{\ell 2} - q_{\ell 3}q_{\ell 4}) & -q_{\ell 1}^2 + q_{\ell 2}^2 - q_{\ell 3}^2 + q_{\ell 4}^2 & 2(q_{\ell 2}q_{\ell 3} + q_{\ell 1}q_{\ell 4}) \\ 2(q_{\ell 1}q_{\ell 3} + q_{\ell 2}q_{\ell 4}) & 2(q_{\ell 2}q_{\ell 3} - q_{\ell 1}q_{\ell 4}) & -q_{\ell 1}^2 - q_{\ell 2}^2 + q_{\ell 3}^2 + q_{\ell 4}^2 \end{pmatrix} \tag{91}$$

The calculated pseudo field angles $\phi^{\rm calc}$ and $\zeta^{\rm calc}$ (in the Satellite Reference System) are then (Eq. 12 in Lindegren (LL-063), see also Fig.18):

$$\begin{pmatrix} \cos\zeta^{\rm calc}\cos\phi^{\rm calc} \\ \cos\zeta^{\rm calc}\sin\phi^{\rm calc} \\ \sin\zeta^{\rm calc} \end{pmatrix} = A(q_\ell)\begin{pmatrix} u_l \\ u_m \\ u_n \end{pmatrix} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} \tag{92}$$

The pseudo field angles are connected to the Field of View Reference System (FoVRS) by $\eta^{\rm calc} = \phi^{\rm calc} \pm \gamma/2$ with γ being the basic angle.

$u = (u_l, u_m, u_n)$ is the direction to the source given in the ICRS (l is a unit vector towards $(\alpha, \delta) = (0, 0)$, n is a unit vector towards $\delta = 90°$ and $m = n \times l$ completes the right-handed triad.

From this we obtain

$$\phi^{\rm calc} = {\rm atan2}(u_y, u_x) \tag{93}$$

and

$$\zeta^{\rm calc} = {\rm asin}(u_z) \tag{94}$$

With A_{ij} being the components of the attitude matrix 91, the derivatives of Eq. 94 with respect to u_l, u_m, and u_n are

$$\frac{\partial\zeta^{\rm calc}}{\partial u_l} = \frac{A_{31}}{\sqrt{1-u_z^2}} \tag{95}$$

$$\frac{\partial \zeta^{\text{calc}}}{\partial u_m} = \frac{A_{32}}{\sqrt{1-u_z^2}} \tag{96}$$

$$\frac{\partial \zeta^{\text{calc}}}{\partial u_n} = \frac{A_{33}}{\sqrt{1-u_z^2}} \tag{97}$$

In the same way a little algebra leads to the derivatives of Eq. 93 with respect to u_l, u_m, and u_n (using the fact that the derivatives of η^{calc} are the same):

$$\frac{\partial \eta^{\text{calc}}}{\partial u_l} = \frac{\partial \phi^{\text{calc}}}{\partial u_l} = \frac{\frac{A_{21}}{u_x} - A_{11}\frac{u_y}{u_x}}{1+\frac{u_y^2}{u_x^2}} = \frac{A_{21}u_x - A_{11}u_x u_y}{u_x^2+u_y^2} \tag{98}$$

$$\frac{\partial \eta^{\text{calc}}}{\partial u_m} = \frac{\partial \phi^{\text{calc}}}{\partial u_m} = \frac{\frac{A_{22}}{u_x} - A_{12}\frac{u_y}{u_x}}{1+\frac{u_y^2}{u_x^2}} = \frac{A_{22}u_x - A_{12}u_x u_y}{u_x^2+u_y^2} \tag{99}$$

$$\frac{\partial \eta^{\text{calc}}}{\partial u_n} = \frac{\partial \phi^{\text{calc}}}{\partial u_n} = \frac{\frac{A_{23}}{u_x} - A_{13}\frac{u_y}{u_x}}{1+\frac{u_y^2}{u_x^2}} = \frac{A_{23}u_x - A_{13}u_x u_y}{u_x^2+u_y^2} \tag{100}$$

The denominator can never become zero, because $\zeta \ll \pi/2$.

This is a very compact way to calculate the derivatives in Eqs. 95-100. If (AGIS) Java routines are used which obtain the derivatives via tangential vectors in the direction of increasing ϕ and ζ (by calculating vector cross products) the equivalence of both approaches can be verified by writing unit tests which compare the results from Eqs. 95-100 with the respective derivatives in the (AGIS) Java routine.

Now, since the derivatives $\frac{\partial u}{\partial \alpha}$ and $\frac{\partial u}{\partial \delta}$ are known from the Java Toolbox routine toProper-SourceDirection, we can calculate

$$\frac{\partial F_\eta}{\partial \alpha} = \frac{\partial F_\eta}{\partial u_l}\frac{\partial u_l}{\partial \alpha} + \frac{\partial F_\eta}{\partial u_m}\frac{\partial u_m}{\partial \alpha} + \frac{\partial F_\eta}{\partial u_n}\frac{\partial u_n}{\partial \alpha}, \tag{101}$$

$$\frac{\partial F_\zeta}{\partial \alpha} = \frac{\partial F_\zeta}{\partial u_l}\frac{\partial u_l}{\partial \alpha} + \frac{\partial F_\zeta}{\partial u_m}\frac{\partial u_m}{\partial \alpha} + \frac{\partial F_\zeta}{\partial u_n}\frac{\partial u_n}{\partial \alpha}, \tag{102}$$

$$\frac{\partial F_\eta}{\partial \delta} = \frac{\partial F_\eta}{\partial u_l}\frac{\partial u_l}{\partial \delta} + \frac{\partial F_\eta}{\partial u_m}\frac{\partial u_m}{\partial \delta} + \frac{\partial F_\eta}{\partial u_n}\frac{\partial u_n}{\partial \delta}, \tag{103}$$

and

$$\frac{\partial F_\zeta}{\partial \delta} = \frac{\partial F_\zeta}{\partial u_l}\frac{\partial u_l}{\partial \delta} + \frac{\partial F_\zeta}{\partial u_m}\frac{\partial u_m}{\partial \delta} + \frac{\partial F_\zeta}{\partial u_n}\frac{\partial u_n}{\partial \delta}, \tag{104}$$

The derivatives with respect to the attitude quaternions can be obtained in the same way from Eqs. 93 and ,94:

$$\frac{\partial\eta^{\text{calc}}}{\partial q_1} = \frac{\partial\phi^{\text{calc}}}{\partial q_1} = \frac{\frac{\partial u_y}{\partial q_1}u_x - \frac{\partial u_x}{\partial q_1}u_y}{u_x^2+u_y^2} = 2\frac{(q_2u_l - q_1u_m + q_4u_n)u_x - (q_1u_l + q_2u_m + q_3u_n)u_y}{u_x^2+u_y^2} \tag{105}$$

$$\frac{\partial\eta^{\text{calc}}}{\partial q_2} = \frac{\frac{\partial u_y}{\partial q_2}u_x - \frac{\partial u_x}{\partial q_2}u_y}{u_x^2+u_y^2} = 2\frac{(q_1u_l + q_2u_m + q_3u_n)u_x + (q_2u_l - q_1u_m + q_4u_n)u_y}{u_x^2+u_y^2} \tag{106}$$

$$\frac{\partial\eta^{\text{calc}}}{\partial q_3} = \frac{\frac{\partial u_y}{\partial q_3}u_x - \frac{\partial u_x}{\partial q_3}u_y}{u_x^2+u_y^2} = 2\frac{(-q_4u_l - q_3u_m + q_2u_n)u_x + (q_3u_l - q_4u_m - q_1u_n)u_y}{u_x^2+u_y^2} \tag{107}$$

$$\frac{\partial\eta^{\text{calc}}}{\partial q_4} = \frac{\frac{\partial u_y}{\partial q_4}u_x - \frac{\partial u_x}{\partial q_4}u_y}{u_x^2+u_y^2} = 2\frac{(-q_3u_l + q_4u_m + q_1u_n)u_x + (q_4u_l - q_3u_m + q_2u_n)u_y}{u_x^2+u_y^2} \tag{108}$$

$$\frac{\partial\zeta^{\text{calc}}}{\partial q_1} = \frac{\frac{\partial u_z}{\partial q_1}}{\sqrt{1-u_z^2}} = 2\frac{q_3u_l - q_4u_m - q_1u_n}{\sqrt{1-u_z^2}} \tag{109}$$

$$\frac{\partial\zeta^{\text{calc}}}{\partial q_2} = \frac{\frac{\partial u_z}{\partial q_2}}{\sqrt{1-u_z^2}} = 2\frac{q_4u_l + q_3u_m - q_2u_n}{\sqrt{1-u_z^2}} \tag{110}$$

$$\frac{\partial\zeta^{\text{calc}}}{\partial q_3} = \frac{\frac{\partial u_z}{\partial q_3}}{\sqrt{1-u_z^2}} = 2\frac{q_1u_l + q_2u_m + q_3u_n}{\sqrt{1-u_z^2}} \tag{111}$$

$$\frac{\partial\zeta^{\text{calc}}}{\partial q_4} = \frac{\frac{\partial u_z}{\partial q_4}}{\sqrt{1-u_z^2}} = 2\frac{q_2u_l - q_1u_m + q_4u_n}{\sqrt{1-u_z^2}} \tag{112}$$

Again, these formula can be used to verify the AGIS routine.

MISSING JPG FILE: direction.jpg

FIGURE 18: At a certain instant t the celestial direction $u = (u_x, u_y, u_z)$ is observed at spherical coordinates (η, ζ) in the instrument frame (this is Fig. 2 from Lindegren (SAG-LL-030).

A Numerical test of the two methods to convert the attitude from ICRS to RGCS

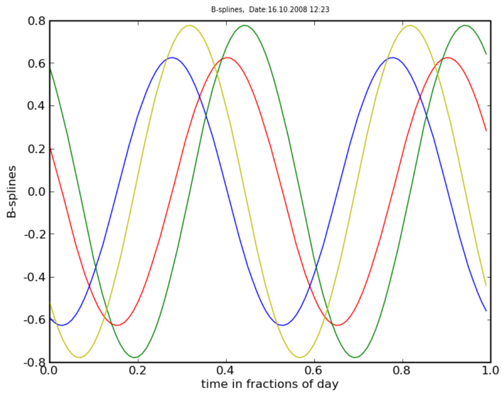

FIGURE 19: This is a plot of the assumed four B-spline coefficients in the ICRS. The equidistant knot separation is $\Delta t_{\rm knots} = 0.01$.. Each B-spline component (corresponding to the quaternion components) is drawn in a different colour.

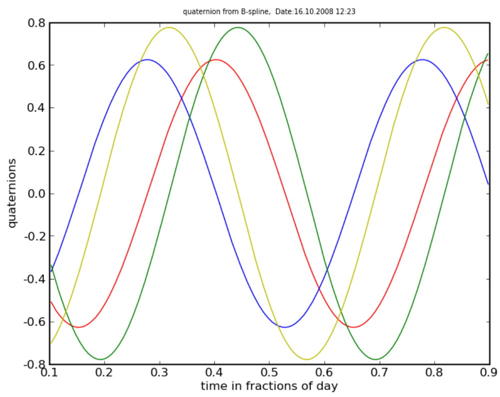

FIGURE 20: This is a plot of the four quaternion components calculated from the B-spline coefficients of Fig. 19 evaluated every $\Delta t = 0.001$.

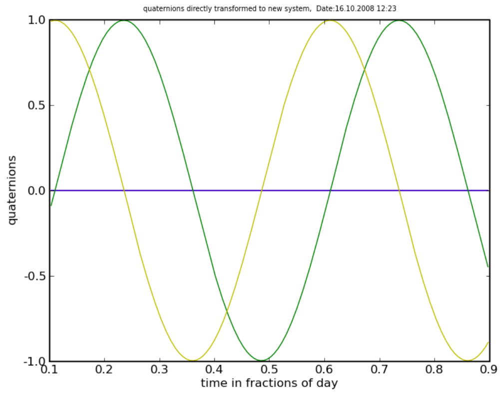

FIGURE 21: The quaternion components from Fig. 20 were converted to the RGCS by using a transformation quaternion.

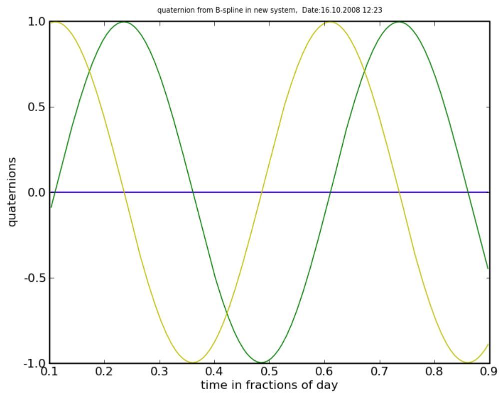

FIGURE 22: This is a plot of the four quaternion components calculated from the B-spline coefficients of Fig. 19 after conversion into the RGCS with the same transformation quaternion as used in Fig. 21.

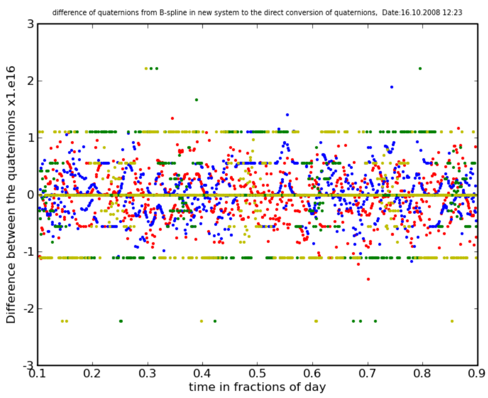

FIGURE 23: Difference between the quaternions shown in Fig. 20 and Fig. 22 The result is that the differences are of the order of 10^{-16} and therefore at the level of the numerical noise.

A.1 Python programs used for the test of the two methods to convert the attitude from ICRS to RGCS

Note, that this a quick and dirty program just to verify that both methods provide the same result up to the level of numerical accuracy and to produce the plots Fig. 19 to Fig. 23.

Main program B-splinetest.py:

```
#!/usr/bin/env python
from Quaternionen import *
from Plot import *
from pylab import *
from numpy import *
from time import *
#from quaternion_functions import *
def conjugate_quaternion(q1,q2,q3,q4):
  return -q1,-q2,-q3,q4


def multiply_quaternions(q1a,q2a,q3a,q4a,q1b,q2b,q3b,q4b):
  return +q4b*q1a+q3b*q2a-q2b*q3a+q1b*q4a,-q3b*q1a+q4b*q2a+q1b*q3a+q2b*q4a,+q2b*q1a-q1b*q2a+q4b*q3a+q3b*q4a,-q1b*q1a-q2b*q2a-q3b*q3a+q4b*q4a

def simple_rgc_scanning_law(omega,t):
#Evaluation of the quaternions for an angular speed omega and for any time t
# For the RGCS
        e1,e2,e3=0.0,0.0,1.0
        return Quaternion(e1*sin(omega*t/2),e2*sin(omega*t/2),e3*sin(omega*t/2),cos(omega*t/2))


def simple_icrs_scanning_law(qr,omega,t):
#Evaluation of the quaternions:
        q=simple_rgc_scanning_law(omega,t)
# The ICRS is tilted relative to the RGCS according to the quaternion
# (qr1,qr2,qr3,qr4):
        qi=qr*q
        return qi



# This python script needs the file quaternion_functions.py in
# which quaternion operations and the scanning laws are defined.


# Initialize t array from tbegin to tend in steps of tstep
tbegin=0.0
tend=1.0
tstep=0.01
tb=arange(tbegin,tend,tstep)

# Angular speed omega=2*pi/T=2*pi/0.25 days for the scanning law.
rotationperiod=0.25
omega=2*pi/rotationperiod

#choose a rotation quaternion (which defines the difference between RGCS and ICRS):
qr=Quaternion(0.213146290330655,-0.590189353578809,0.583066205072733,-0.516022273107071)

#Evaluation of the B-spline coefficients  all knots in t:
#Actually this calculates quaternions, but we assume they are
# B-spline coefficients.
b=simple_icrs_scanning_law(qr,omega,tb)


# Plot the B-splines:
figure(1)
plot_quaternions(tb,b,"B-splines, ","time in fractions of day","B-Splines","B-splines.png")


tbegin=0.1
tend=0.9
tstep=0.001
t=arange(tbegin,tend,tstep)
q1=arange(tbegin,tend,tstep)
q2=arange(tbegin,tend,tstep)
q3=arange(tbegin,tend,tstep)
q4=arange(tbegin,tend,tstep)

spline_basis_matrix=array([[-1,3,-3,1],[3,-6,3,0],[-3,0,3,0],[1,4,1,0]])/6.0

#bb=b.asArray()
```

```
b1=b.c1
b2=b.c2
b3=b.c3
b4=b.c4

print b1[0]


for k in arange(3,len(t)-2):
   smaller= array(where(tb<t[k]))
   smallerarray=smaller[0]
   leftindex= len(smallerarray)-1
#   print leftindex,tb[leftindex],t[k],tb[leftindex+1]
   bspline_vec=array([b1[leftindex-1],b1[leftindex],b1[leftindex+1],b1[leftindex+2]]).transpose()
#   print bspline_vec
   tt=(t[k]-tb[leftindex])/(tb[leftindex+1]-tb[leftindex])
   tarray=array([tt**3,tt**2,tt,1])
   bspline_vec=array([b1[leftindex-1],b1[leftindex],b1[leftindex+1],b1[leftindex+2]]).transpose()
   q1[k]=dot(dot(tarray,spline_basis_matrix),bspline_vec)
   bspline_vec=array([b2[leftindex-1],b2[leftindex],b2[leftindex+1],b2[leftindex+2]]).transpose()
   q2[k]=dot(dot(tarray,spline_basis_matrix),bspline_vec)
   bspline_vec=array([b3[leftindex-1],b3[leftindex],b3[leftindex+1],b3[leftindex+2]]).transpose()
   q3[k]=dot(dot(tarray,spline_basis_matrix),bspline_vec)
   bspline_vec=array([b4[leftindex-1],b4[leftindex],b4[leftindex+1],b4[leftindex+2]]).transpose()
   q4[k]=dot(dot(tarray,spline_basis_matrix),bspline_vec)

q=Quaternion(q1,q2,q3,q4)

#Omit limits of the interval for plotting
qq=Quaternion(q.c1[3:len(t)-2],q.c2[3:len(t)-2],q.c3[3:len(t)-2],q.c4[3:len(t)-2])
# Plot the B-splines:
figure(2)
plot_quaternions(t[3:len(t)-2],qq,"quaternions from B-splines, ","time in fractions of day","quaternions","qfromB-splines.png")


# Now convert q1,q2,q3,q4 into another system with the qq-th quaternion
qq=10
#qmidcon=q.conjugate()
#qmidcon=qmidcon.asArray()
#qmidcon=qmidcon[0:4,qq]
#print qmidcon
qmidcon=Quaternion(q.c1[qq],q.c2[qq],q.c3[qq],q.c4[qq]).conjugate()


#qmidcon=Quaternion(qmidcon[0],qmidcon[1],qmidcon[2],qmidcon[3])


#Transformation to new system:
qn=qmidcon*q
print qn

#Omit limits of the interval for plotting
qn=Quaternion(qn.c1[3:len(t)-2],qn.c2[3:len(t)-2],qn.c3[3:len(t)-2],qn.c4[3:len(t)-2])
figure(3)
plot_quaternions(t[3:len(t)-2],qn,"quaternions directly transformed to new system, ","time in fractions of day","quaternions","qnewfromquaternions


# Now convert first the B-spline coefficients to the new system:
bn=qmidcon*b
b1n=bn.c1
b2n=bn.c2
b3n=bn.c3
b4n=bn.c4
q1nn=arange(tbegin,tend,tstep)
q2nn=arange(tbegin,tend,tstep)
q3nn=arange(tbegin,tend,tstep)
q4nn=arange(tbegin,tend,tstep)

for k in arange(3,len(t)-2):
   smaller= array(where(tb<t[k]))
   smallerarray=smaller[0]
   leftindex= len(smallerarray)-1
   print leftindex,tb[leftindex],t[k],tb[leftindex+1]
   bspline_vec=array([b1n[leftindex-1],b1[leftindex],b1[leftindex+1],b1[leftindex+2]]).transpose()
#   print bspline_vec
   tt=(t[k]-tb[leftindex])/(tb[leftindex+1]-tb[leftindex])
   tarray=array([tt**3,tt**2,tt,1])
   bspline_vec=array([b1n[leftindex-1],b1n[leftindex],b1n[leftindex+1],b1n[leftindex+2]]).transpose()
   q1nn[k]=dot(dot(tarray,spline_basis_matrix),bspline_vec)
   bspline_vec=array([b2n[leftindex-1],b2n[leftindex],b2n[leftindex+1],b2n[leftindex+2]]).transpose()
   q2nn[k]=dot(dot(tarray,spline_basis_matrix),bspline_vec)
   bspline_vec=array([b3n[leftindex-1],b3n[leftindex],b3n[leftindex+1],b3n[leftindex+2]]).transpose()
   q3nn[k]=dot(dot(tarray,spline_basis_matrix),bspline_vec)
   bspline_vec=array([b4n[leftindex-1],b4n[leftindex],b4n[leftindex+1],b4n[leftindex+2]]).transpose()
   q4nn[k]=dot(dot(tarray,spline_basis_matrix),bspline_vec)


figure(4)
```

```
qnn=Quaternion(q1nn,q2nn,q3nn,q4nn)
qqnn=Quaternion(qnn.c1[3:len(t)-2],qnn.c2[3:len(t)-2],qnn.c3[3:len(t)-2],qnn.c4[3:len(t)-2])
figure(3)
plot_quaternions(t[3:len(t)-2],qqnn,"quaternion from B-spline in new system, ","time in fractions of day","quaternions","qnewfromBsplines.png")


figure(5)
print "qn0",qn.c1
print "qqnn=",qqnn.c1
qdif=Quaternion((qqnn.c1-qn.c1)*1.e16,(qqnn.c2-qn.c2)*1.e16,(qqnn.c3-qn.c3)*1.e16,(qqnn.c4-qn.c4)*1.e16)
plot_quaternions(t[3:len(t)-2],qdif,"Difference of quaternions from B-spline in new system to the direct conversion of quaternions,  ","time in fr

show()
```

Quaternionen.py:

```
#!/usr/bin/env python
import unittest
import numpy
from numpy import array

class Quaternion:
    """
    Quaternion class
    """
    def __init__(self, q1=0, q2=0, q3=0, q4=1):
       self.c1=q1
       self.c2=q2
       self.c3=q3
       self.c4=q4
    def norm(self):
        return (self.c1**2+self.c2**2+self.c3**2+self.c4**2)**0.5
    def normalise(self):
        self.c1=self.c1/self.norm()
        self.c2=self.c2/self.norm()
        self.c3=self.c3/self.norm()
        self.c4=self.c4/self.norm()
        return self
    def conjugate(self):
        return self.__class__(-self.c1,-self.c2,-self.c3,self.c4)
    def __mul__( self, other ):
        """Multiply this quaternion by another quaternion,
        generating a new quaternion
        """
        qq4= self.c4*other.c4 - self.c1*other.c1 - self.c2*other.c2 -self.c3*other.c3
        qq1= self.c4*other.c1 + self.c1*other.c4 + self.c2*other.c3 -self.c3*other.c2
        qq2= self.c4*other.c2 + self.c2*other.c4 + self.c3*other.c1 -self.c1*other.c3
        qq3= self.c4*other.c3 + self.c3*other.c4 + self.c1*other.c2 -self.c2*other.c1
        return self.__class__(qq1,qq2,qq3,qq4)
    def __add__( self, other ):
        """Add this quaternion by another quaternion,
        generating a new quaternion
        """
        return self.__class__(self.c1+other.c1,self.c2+other.c2,self.c3+other.c3,self.c4+other.c4)
    def __sub__( self, other ):
        """Subtract another quaternion from  this quaternion,
        generating a new quaternion
        """
        qq1=self.c1-other.c1
        qq2=self.c2-other.c2
        qq3=self.c3-other.c3
        qq4=self.c4-other.c4
        return self.__class__(qq1,qq2,qq3,qq4)

    def asArray(self):
        return array([self.c1,self.c2,self.c3,self.c4])


class QuaternionTests(unittest.TestCase):

    def testConjugate(self):
        a=Quaternion(1,2,3,4).conjugate()
        self.assertEquals(a.c1,-1)
        self.assertEquals(a.c2,-2)
        self.assertEquals(a.c3,-3)
        self.assertEquals(a.c4,4)
        b=a.conjugate()
        b=b.conjugate()
        self.assertEqual(b.asArray().all(),a.asArray().all())
```

```
    def testMultiply(self):
        a=Quaternion(1,2,3,4)
        p=a*a.conjugate()
        print a.norm()**2,p.c4
        self.assertAlmostEqual(a.norm()**2,p.c4,13)

    def testAdd(self):
        a=Quaternion(1,2,3,4)
        b=Quaternion(-1,-2,-3,-4)
        c=a+b
        self.assertEqual(c.asArray().all(),array([0,0,0,0]).all())

if __name__ == '__main__':
    unittest.main()
```

Plot.py:

```
#!/usr/bin/env python
from Quaternionen import *
from pylab import *
from time import *

def plot_quaternions(t,q,tit,xlab,ylab,plotname):
# Plot the B-splines:
    lt=localtime()
    title('B-splines,  Date:'+str(lt[2])+'.'+str(lt[1])+'.'+str(lt[0])+' '+str(lt[3])+':'+str(lt[4]),size=7)
    plot(t,q.c1,'r-')
    plot(t,q.c2,'b-')
    plot(t,q.c3,'g-')
    plot(t,q.c4,'y-')
#    axis([0, 1, -1, 1])
    xlabel(xlab)
    ylabel(ylab)
    savefig(plotname)
```

B Documents relevant for the ODAS/Ring Solution

Bernstein et al. (GAIA-ARI-BST-001-5)	Description of the Ring Solution
Hirte et al. (SH-003)	Program Description of the Ring Solution (FORTRAN version)
Hirte et al. (SH-004)	Results of the Ring Solution