News on Seeking Gaia's Astrometric Core Solution with AGIS

Uwe Lammers

Gaia Science Operations Centre
ESA/ESAC/SRE-ODO

Outline

0. AGIS : The name of the game
1. CG : Efficient pathfinder in 5000 Mio dimensions
2. GC : Calibration the easy way …
3. Amazon: AGIS in the Clouds

AGIS: The name of the game

- **Astrometric Global Iterative Solution**
  - DPAC's method of choice for constructing the astrometric part of the catalogue from all available relevant measurements (10^{12} for determining $5 \cdot 10^9$ unknowns)
- WP 320 – main players:
  - Lund Observatory, Lund (Lindegren + Hobbs + Holl)
  - ARI, Heidelberg (Bastian)
  - Lohrman Observatory, Dresden (Klioner + Butkevich)
  - ESAC (Lammers + team)
- Distributed, multi-threaded, all-in Java system using DPAC common resources + infrastructure
  - GaiaTools
  - Development tools: svn, Mantis, ant, ivy, …

What is AGIS solving?

Unknowns: source+attitude +calibration

Residual (O-C) of observation *l*

Downweighting function

$$\chi^2(x) = \sum_l \frac{R_l(x)^2}{\sigma_l^2 + \varepsilon_l^2} w\left(\frac{R_l(x)}{\sqrt{\sigma_l^2 + \varepsilon_l^2}}\right)$$

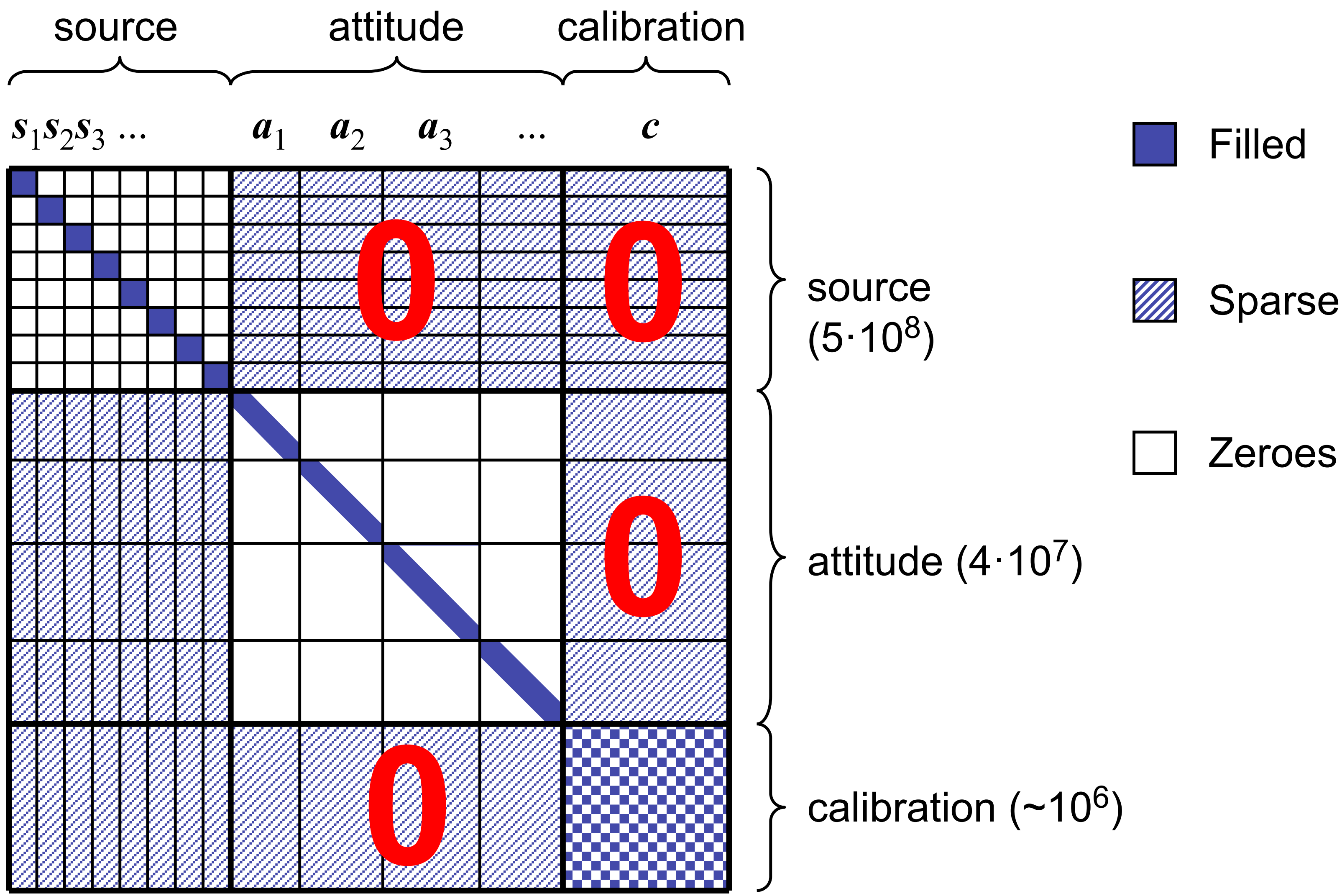
Sum over all observations

Observation noise

Excess noise

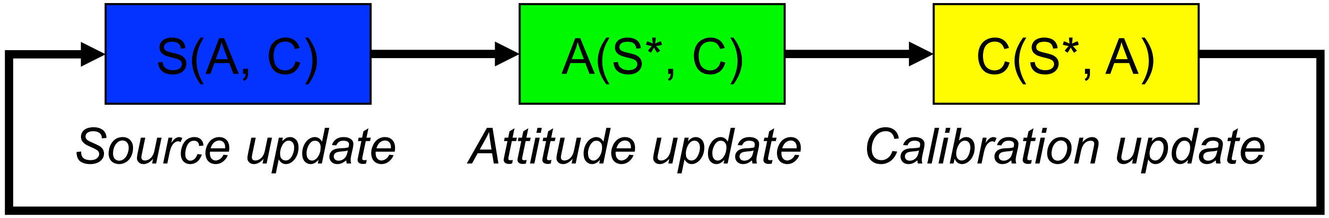
Merrit function

How to solve: Iterative with Normal Equations and Pre-conditioner (Gauss-Seidel)



1: Conjugate Gradients

- The most simple scheme to tackle the problem: “Simple Iterations” (SI):



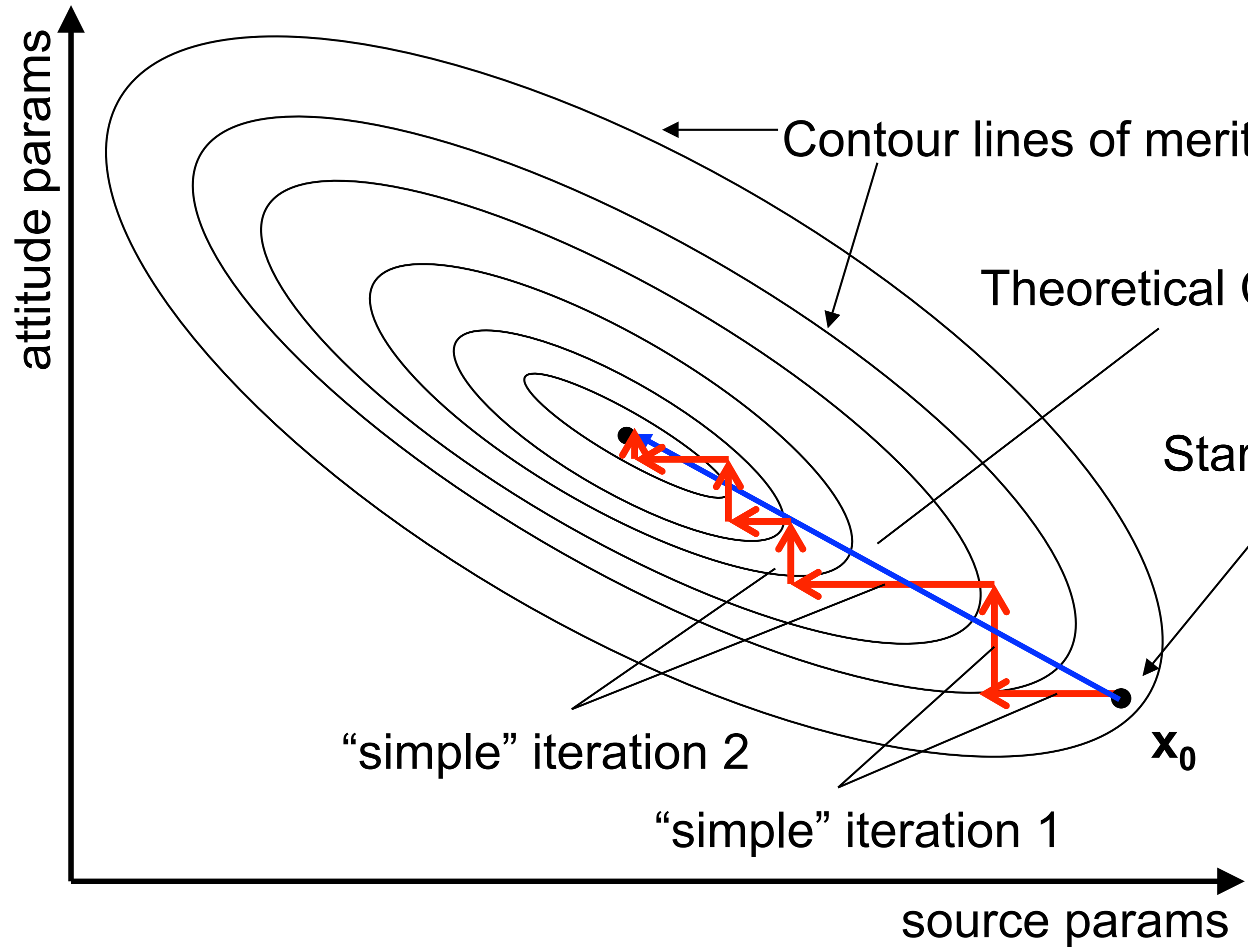
until convergence – very slow!

- Extension is “Accelerated Simple Iterations” (ASI): Updates in successive iterations are not uncorrelated
  - Compute “trial” updates for a small number of sources, compute extrapolation factor and use this for all the others
  - Baseline in AGIS since ~2008
  - Convergence rate ~2 times better than SI

Conjugate Gradients

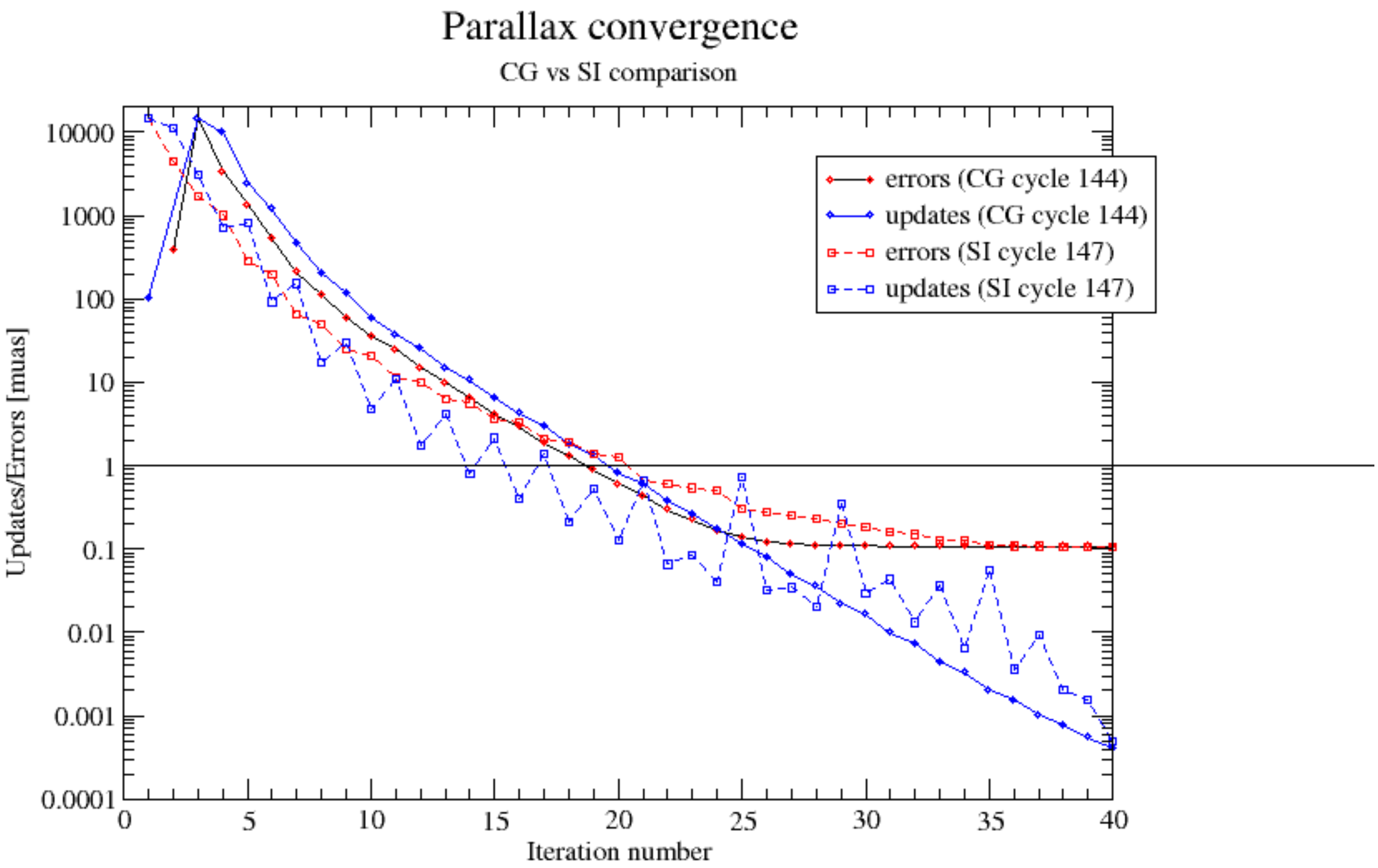
- CG standard method in Linear Algebra knows since decades
- History for AGIS
  - “Discovered” for AGIS by LL in 2008
  - Then prototyped in AGISLab by AB
  - Implemented in AGIS during summer 2009
- Different blocks are solved like in SI but
  - Updates are only “provisional” and not applied
  - “Kernel results” from different blocks and auxiliary calculated quantities from the previous iteration are combined to compute the final update
  - This effectively constructs a basis of Conjugate vectors in the space of the unknowns
  - Faster convergence!

CG vs SI/ASI

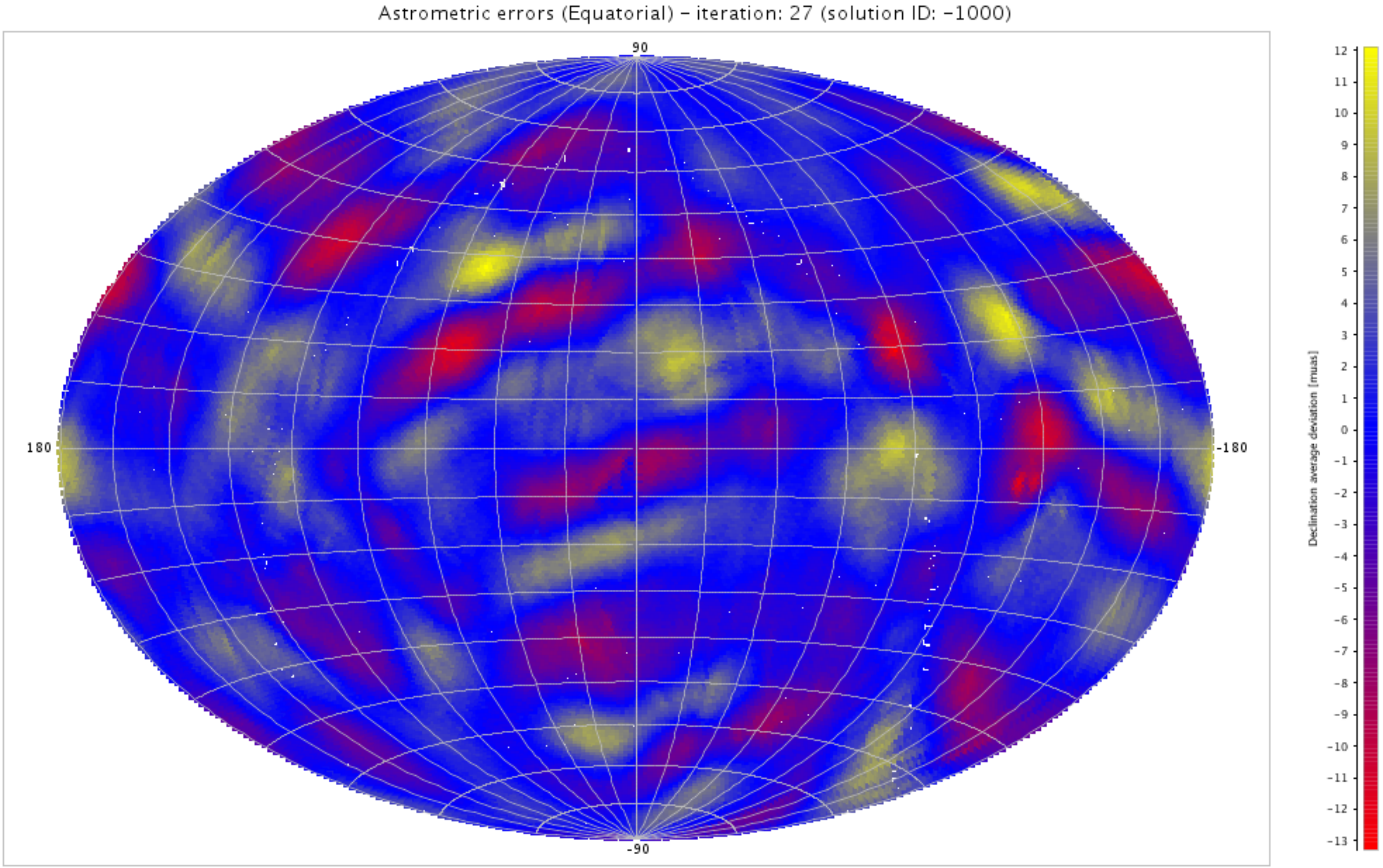


attitude params
Contour lines of merit function
Theoretical CG step
Starting point
x_0
"simple" iteration 2
"simple" iteration 1
source params

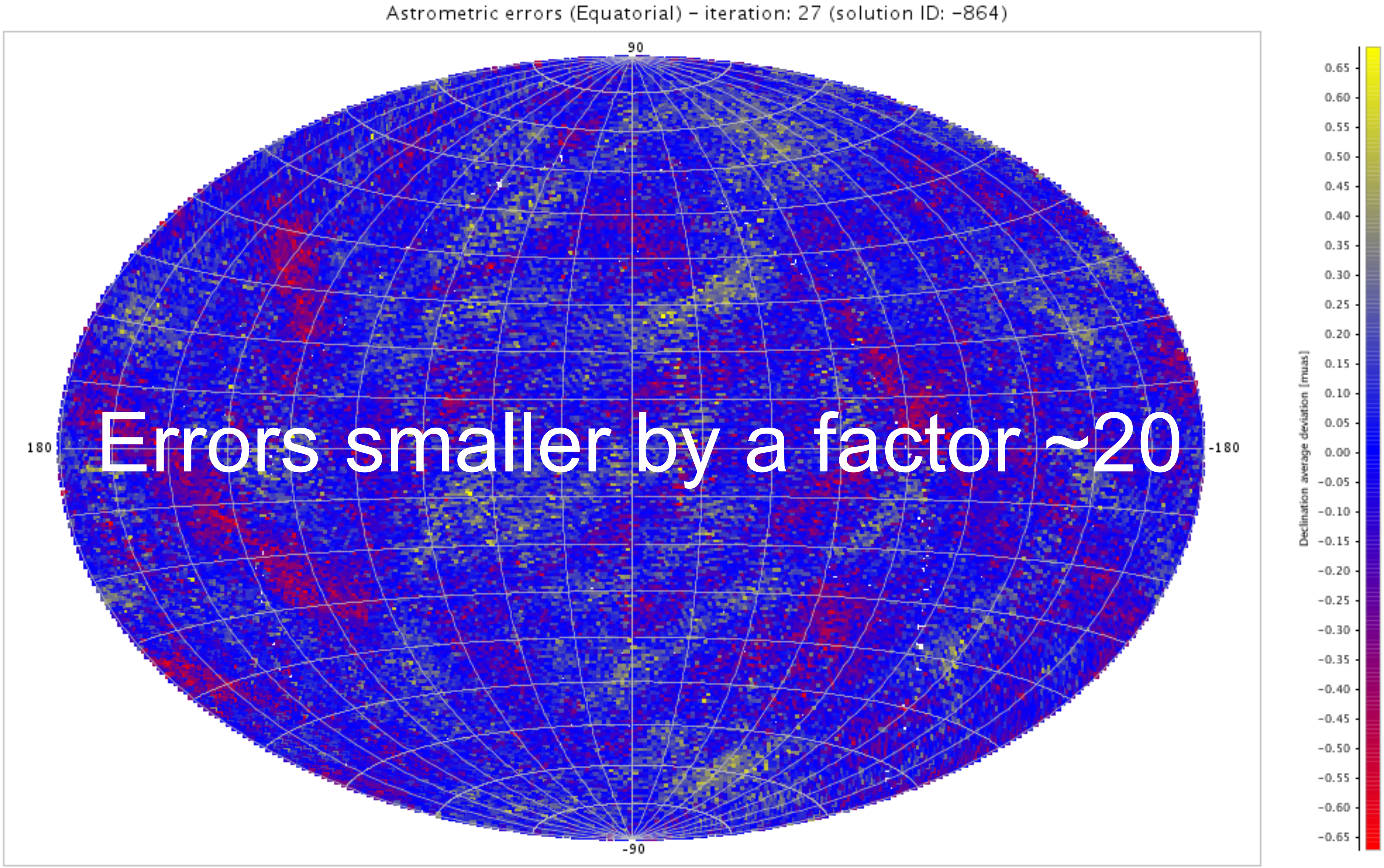
Convergence rate CG vs ASI



Declination error after I27 (ASI)



Declination error after I27 (CG)



2: Generic calibration

Astrometric calibration

All calibration effects with relevance for the astrometric data processing that can be modelled as shifts to observed field angles

Old AGIS calibration model:

$$\eta_l^{obs} = \eta^0_n + \Delta\eta_x + \delta\eta_y + C_0(G - G_{ref}) + C_1(W - W_{ref})$$

$$\zeta_l^{obs} = \zeta^0_n + \Delta\zeta_y$$

- AL/AC large scale
- AL small scale
- Linear flux-dependent term
- Linear spectrum-dependent term

Problem with hard-coded calibration

- This is much too inflexible!
- Geometric calibrations may not become more complex in the future but in the end we want to find residual calibration (CTI) effects not considered in IDT/IDU …
- Each change in the model entails software changes
- We want a scheme that is
  - Flexible
  - Extensible
  - efficient

Generic Calibration: Outline

- Extension of base scheme
  - $\eta_l^{obs} = \eta_n^0 + \Sigma\, E_i^{AL}(l)$
  - $E_i(l) = \Sigma\, c_{i,j} \cdot f_{i,j}(l)$

  $f_{i,j}$: elemental calibration functions
  $c_{i,j}$: calibration unit values = to be determined
- The c values are depended on
  - time
  - CCD/gate numbers
  - pixel column
  - telescope number
  - ….

  The entirety of all c constitutes the astrometric calibration.

XML configuration

```xml
?xml version="1.0" encoding="ISO-8859-1"?>
<AstroCal xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="astrocal.xsd">
 <DataSpace>
    <Axis id="0" label="CcdRowNo" origin="0" min="1" max="7" delta="1" discrete="true" distributed="true"/>
    <Axis id="1" label="AFCcdStripNo" origin="0" min="3" max="11" delta="1" discrete="true" distributed="true"/>
    <Axis id="2" label="Time" origin="J2012.0" min="0d" max="1840d" delta="115d" discrete="false" distributed="false">
        <Discretization id="1" uniform="true" scale="4"/>
        <Discretization id="2" uniform="true" scale="2"/>
    </Axis>
    <Axis id="3" label="Telescope" origin="0" min="1" max="2" delta="1" discrete="true" distributed="false"/>
    <Axis id="4" label="PixelCol" origin="0" min="0" max="11" delta="1" discrete="true" distributed="false">
        <Discretization id="1" uniform="true" scale="2"/>
        <Discretization id="2" uniform="true" scale="3"/>
 </Axis>
 </DataSpace>
 <FuncsCollection>
    <Func id="0" classcal="gaia.cu3.agis.algo.gis.calibration.model.L0" description="1"/>
 </FuncsCollection>
 <ALEffectsCollection>
    <Effect id="0" use="true" update="true" description="AL large-scale" dep="*,*,0,*,-" funcs="0"
constraints="gaia.cu3.agis.algo.gis.calibration.model.AlLargeScaleConstraints"/>
    <Effect id="1" use="true" update="true" description="AL small-scale" dep="*,*,-,-,0" funcs="0"
constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
    <Effect id="2" use="true" update="true" description="Radiation damage" dep="*,*,1,-,1" funcs="0"
constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
    <Effect id="3" use="true" update="true" description="Chromaticity" dep="*,*,2,*,-" funcs="0"
constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
    <Effect id="4" use="true" update="true" description="Effect X" dep="*,*,-,*,2" funcs="0"
constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
    <Effect id="5" use="true" update="true" description="Effect Y" dep="*,*,2,*,2" funcs="0"
constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
 </ALEffectsCollection>
 <ACEffectsCollection>
    <Effect id="0" use="true" update="true" description="AC large-scale" dep="*,-,-,*,-"
 funcs="0" constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
 </ACEffectsCollection>
</AstroCal>
```

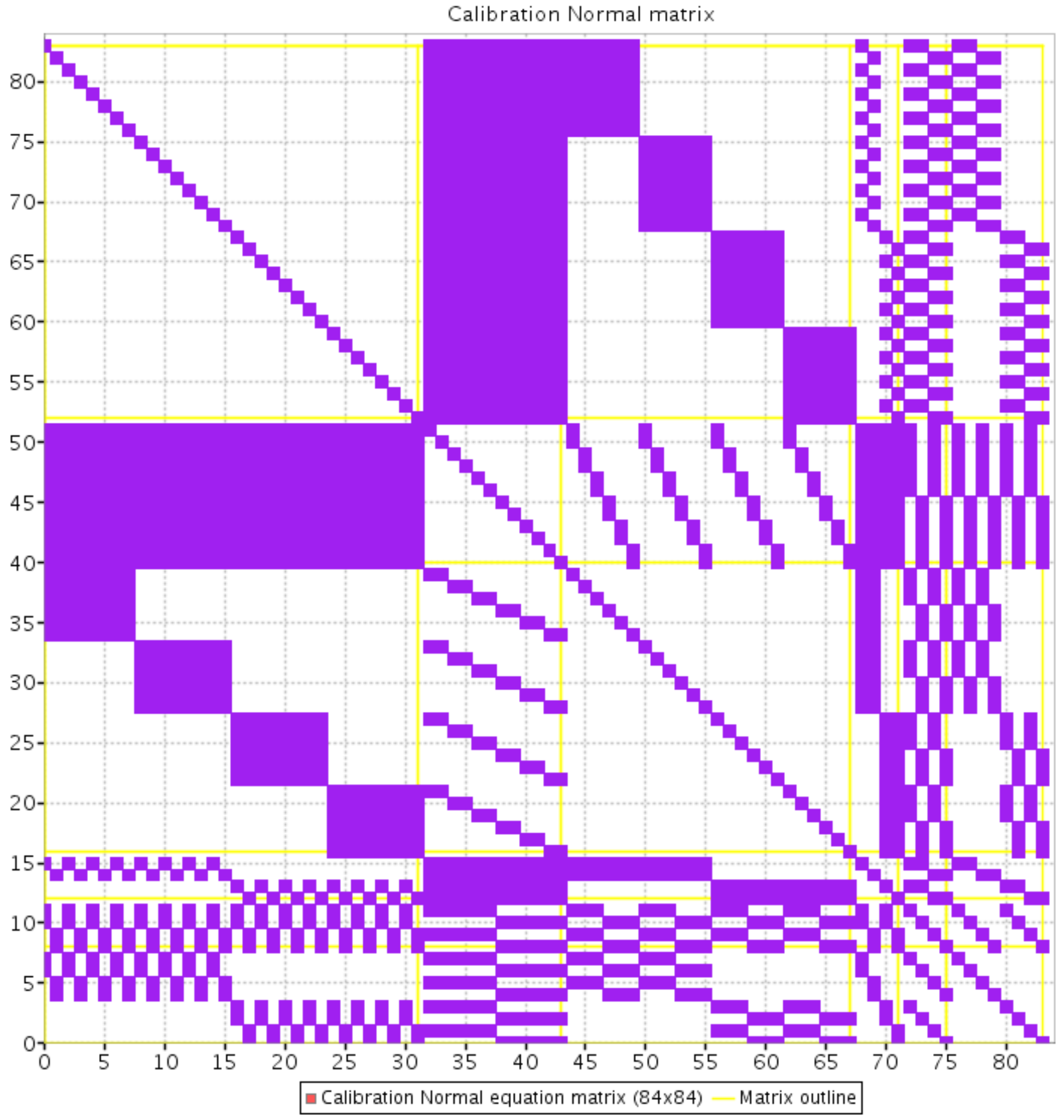
Calibration is entirely defined at runtime through this single configuration file!

Hyper-cube definition: What parameters does calibration depend on?

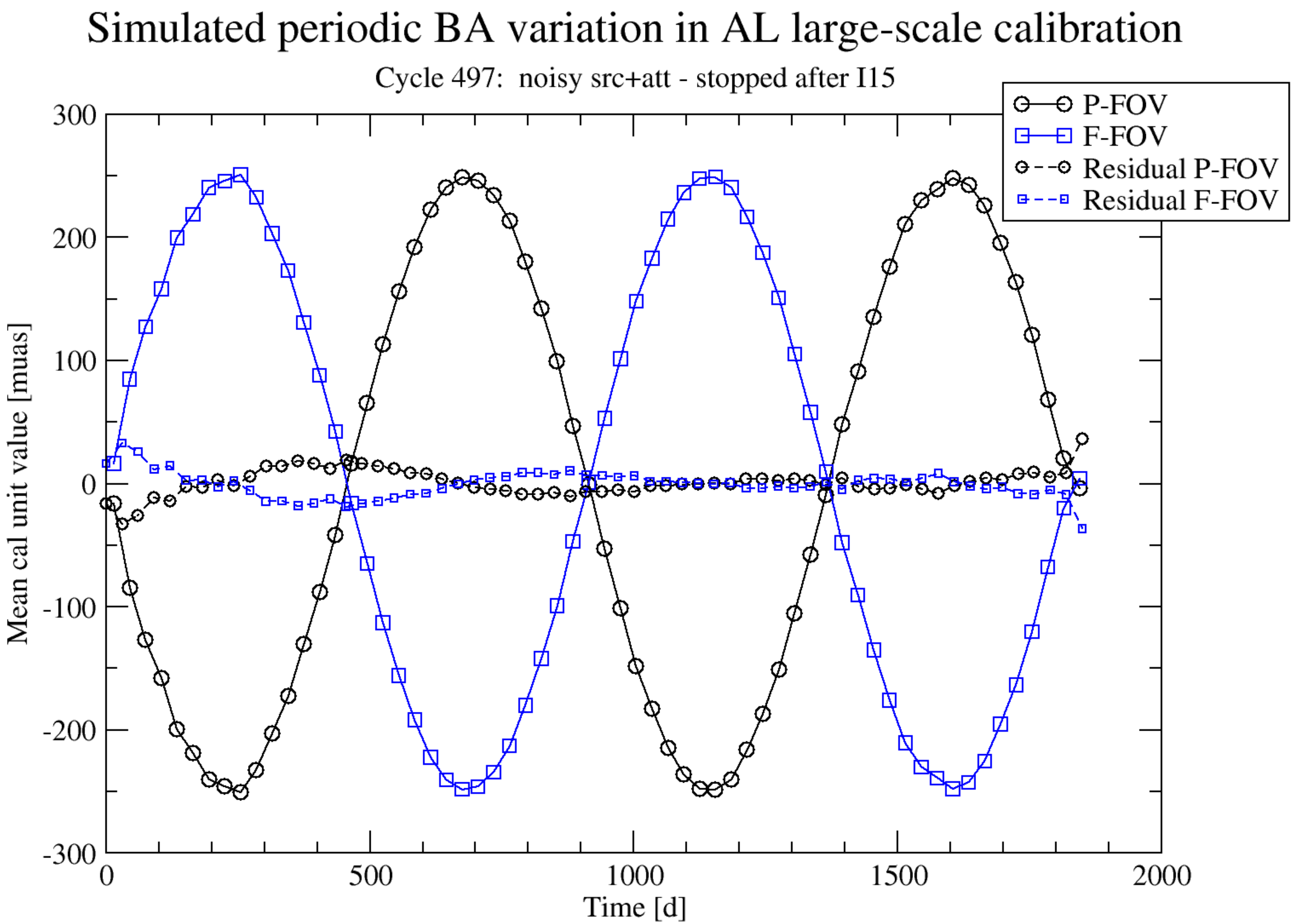
Functions definition: What fundamental functional forms does calibration consist of?

Effects definition: What distinct effects constitute the calibration?

Test of gen cal update



Generic calibration: It works!



3: Running AGIS in the Cloud

- How do we run AGIS during operations?
  - Big computational task (~10^{20} FLOPS)
  - Baseline so far: On a big machine at ESAC!
- Since about 2 years, Cloud computing is a big hype
  - Really nothing more than a collection of virtual machines (“instances”) with strictly defined profiles
  - Do not know where they are or what they are physically
  - Service providers: Several but Amazon is becoming a dominant player …
- Advantages
  - 100% availability – no worries about maintenance, hardware failures, network etc.
  - Elasticity: My “virtual cluster” can grow or shrink as I need it at every moment in time
  - Cost-effectiveness: Only pay for CPUs and disks when I need them

In practice …

- Have run AGIS successfully in the Amazon Cloud last year
  - 2 Mio sources
  - ASI cycle with ~40 iterations
- Trying to step up to 50 Mio sources last couple of months
  - Current ESAC machine to small for this data set
  - Performance problems – likely issues in AGIS
  - No worries, this is normal work, will fix it
- Conclusions so far:
  - Running AGIS in the Cloud works
  - Remains option for operations
  - Cost-effectiveness: Yes, clearly now. During operations it will depend on how often we run AGIS!

Conclusion

AGIS development is on track

I am convinced we will have a good system that will give us the best possible astrometric catalogue for Gaia!