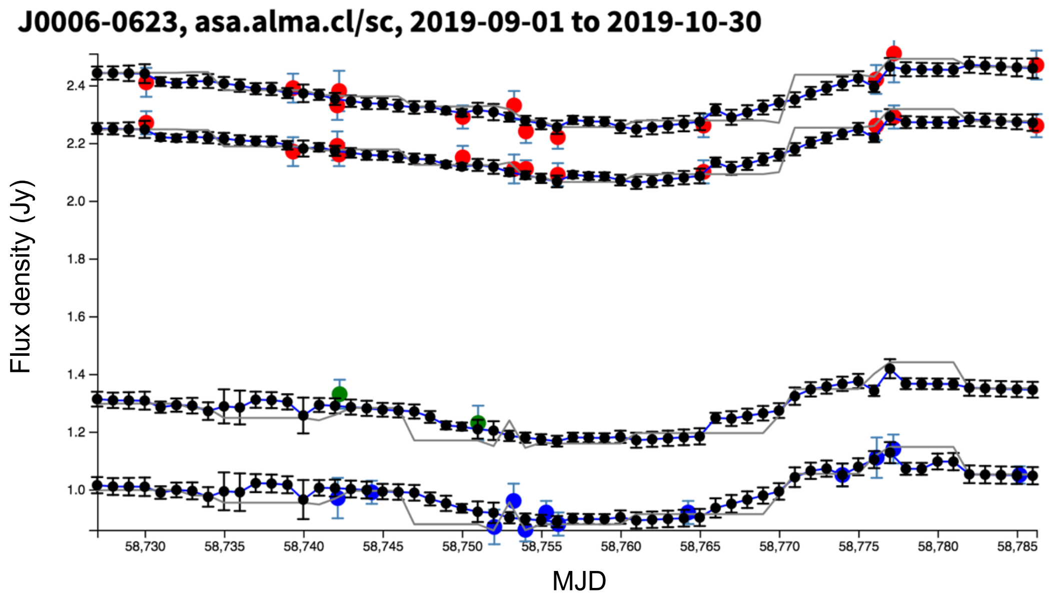
Flux Service of the ALMA Source Catalogue

Ruediger Kneissl, Kurt Plarre, Brian Mason, Alisdair Manning

31 August, 2022
Version 2.4



Time and frequency interpolated flux density estimates from the Flux Service fitting algorithm (black points) and getALMAFlux (grey line) are shown for Bands 3, 6 and 7 (data: red, green and blue) and correspond to the respective standard frequencies for the Calibrator Survey.

1 Description of the Flux Estimation Service in the ALMA Calibrator Source Catalogue

1.1 Introduction

The flux estimation service provides a REST API interface to the ALMA Calibrator Source Catalogue for a query of the best estimate of the flux density of a calibrator at a given frequency and on a given date. The algorithm uses the fit of a linear model for the flux density in time and log frequency space, and decides on which of the flux measurements closest to the query date can be included into the fit. The flux service using the fit algorithm is thus intended mainly for flux calibration with well-sampled sources, such as the Grid Monitoring of quasars, used as secondary flux calibrators. For more sparsely sampled calibrators, the heuristic originated in aU.getALMAFlux is used. The results on the estimated flux density and spectral index are returned in VO format, together with uncertainties and quality flags.

1.2 The method

All measurements in the SC are extracted for a time window +/- 2 months around the query time t_0, and sorted by distance in days from t_0. The Levenberg-Marquardt algorithm (JAVA code) is employed to provide a fit result, based on the below model for each set of measurements, increasing the number of data points starting from 3 to all. The result with the best goodness of fit, in order to avoid situations, where variability causes the flux curve to deviate from linear, is given as the query answer for the flux density and spectral index, together with their uncertainties.

1.3 Model

$$\mathrm{FLUX} = (a \times (t - t_0) + b) \times (f / f_0)^c$$

The flux density is modeled by a linear term in time t, where t_0 denotes the query date, and a linear term in log frequency (f) space, where f_0 denotes the query frequency, with fit parameters a, b, and c. Note that this linear time term could be replaced in the future with a curve derived from an Ornstein-Uhlenbeck stochastic process model (Kelly+09, Kelly+11) developed within the EA ARC (Guzman et al. 2019), and the spectral index extended with curvature.

If no fit result of sufficient quality can be achieved (due to specific criteria developed to ensure good fit conditions and detailed in 1.8, such as sufficient number of sources in the time window, time and frequency bracketing, acceptable formal fit errors) the flux service calls as a fallback option the well-established aU.getALMAFlux algorithm and provides its result.

1.4 Example query format

An example of the query format can be found here:
https://almascience.org/sc/flux?DATE=15-Oct-2019&FREQUENCY=233.0E+09&NAME=J0006-0623

1.5 Detailed description of the data selection for the fitting algorithm

The Flux Estimation algorithm provides an estimate of the flux and spectral index of a given source, at a certain date and frequency, based on the available flux measurements for that source in the Source Catalogue. For this purpose, it combines two approaches to flux and spectral index estimation:

1. Algorithm implemented in getALMAFlux function as part of analysisUtils package ("getALMAFlux"). This algorithm requires at least two measurements in different bands to provide an estimate of the spectral index, and the flux. The spectral index is estimated first, and then used to interpolate/extrapolate to the desired frequency. It assumes that the flux is constant in time, during the period considered for the estimation.
2. This method considers a linear time variation of the flux in addition to the exponential variation in frequency, and performs a simultaneous fitting of the parameters involved, using numerical methods. This method provides a better approximation of the flux behavior, but requires stronger conditions on the data than getALMAFlux, to obtain a good fit.

The algorithm works by sequentially incorporating measurements, checking the goodness of fit in the following way:

1) With inputs SourceName, Query Date and Frequency
2) Get the measurements for SourceName and catalogue "ALMA" (ignoring a source band limit, in the query to DataBase)
3) Reduce the list of measurements, removing those where the observed date is more than 60 days from the Query Date, ordering the remainder by distance from Query Date
4) Check the conditions for all measurements in the range of 60 days from Query Date (these conditions are used as Best Conditions):
    a) **If there are at least two bands used in the measurements in the range of 7 days from the Query Date**
    b) **if Query Date is bracketed in one band (same or below as the Query Frequency) in the measurements in the range of 14 days from the Query Date**
5) Begin loop in measurements, from index = 4
    a) Take a subset from the measurements: 0..index, make a flux estimation using the subset of measurements, Check the Best Conditions for the subset of measurements, update best Conditions and best flux estimation when met.
    b) Check if the flux estimation is within the error allowed for the band of the Query Frequency and if the new estimation and Conditions fit better than previous, update best Conditions and best flux estimation when met
    c) Check spectral index bracketing (the higher band measurement has lower band measurements before / after following time bracketing), update best Conditions and best flux estimation when met
    d) if 7 and later 14 days are reached and Conditions 4) a) and respectively b) are not yet fulfilled, aborts.

Returns the flux estimation with the best goodness of fit. It can be null, if the measurements and flux estimations didn't match the criteria, in which case the result of getALMAFlux is used.

2 The Use of the Flux Estimation Service

2.1 Query Options

The URL format such as “%2B” in the source name for positive declination is not required, but the “+” is especially recognized by the server.

Required input

NAME: string in ascii format
DATE: in the format DD-MONTH-YEAR, DD-MON-YEAR
FREQUENCY: double precision float in Hz

Recommended query settings

MODEL = 0 (referring to the goodness-of-fit criteria, $\chi^2 = \sum_i^n [(D_i - M_i)^2 / \sigma_i^2] / (n-3)$, other values than “=0” are experimental).

WEIGHTED = true (since ALMA flux errors are considered reliable; “=false” allows two options: UNWEIGHTED = 0 uses averaged ALMA errors and “=1” uses the fit residuals).

VERBOSE = 0 (“=1” gives the iterative fitting results for an increasing number of measurements used and allows to review the selected number of measurement result).

TEST = false (“=true” ignores all measurements of the request date, which can be used for testing purposes, i.e. estimate the flux density of a specific measurement by other measurements of the source in the database).

FALLBACK = true (“=false” will force reporting the primary algorithm fit result, which in combination with VERBOSE=1,2 is useful to diagnose cases for which FALLBACK=true reports the fallback result).

2.2 Output parameters

The output is given as a VOTable in XML format. It contains the following fields

- Status Code: result quality (lowest order bit indicating primary or fallback estimation algorithm; next higher order bit indicating success or failure of the query; thus (00_2=) 0: successful query (primary); (01_2=) 1: successful query (fallback); ($1X_2$) >=2: query failed)
- Source Name: input as string
- Frequency: input in Hz as double precision float
- Date: input as string
- Flux Density: in Jansky as double precision float
- Flux Density Error: in Jansky as double precision float
- Spectral Index: as double precision float
- Spectral Index Error: as double precision float
- Warning: as integer, code in three digits
  - 1st digit: number of flux measurements used for the result ("9" indicates the use of ≥9 measurements)
  - 2nd digit: "1" if there is a measurement in the band containing the queried frequency or two measurements bracketing in frequency within +/- one week, "0" otherwise
  - 3rd digit: "1" if there are measurements on the queried day or bracketed by two measurements within +/- one week, "0" otherwise
  - The warning flags are intended to give some information about coverage with measurements in the database, which are possibly relevant for the quality of the intended fit, but the individual measurement errors are not considered.
- Age of Nearest Monitor Point: Quality measure in terms of quasar variability, time separation to the closest measurement given in days ("+/positive" if the measurement is "before" the query date, and "-/negative" if the measurement is "after" the query date)
- Verbose Output: as string (if specified)
- Software Version: as string

2.3 Output Example

```
<TABLE>
<FIELD datatype="int" width="1" name="StatusCode">
<DESCRIPTION> 0: a valid flux density was able to be calculated with the
default algorithm 1: the fallback algorithm was used 2: no valid flux density
could be calculated </DESCRIPTION>
</FIELD>
<FIELD datatype="char" name="SourceName" arraysize="16"/>
<FIELD unit="Hz" datatype="double" width="10" name="Frequency"/>
<FIELD datatype="char" name="Date" arraysize="32"/>
<FIELD unit="Jansky" datatype="double" width="10" name="FluxDensity"/>
<FIELD unit="Jansky" datatype="double" width="10" name="FluxDensityError"/>
<FIELD unit="Unitless" datatype="double" width="10" name="SpectralIndex"/>
<FIELD unit="Unitless" datatype="double" width="10" name="SpectralIndexError"
/>
<FIELD datatype="int" width="10" name="DataConditions">
<DESCRIPTION>3 numerical codes: value[0]: number of available measurements
used, where 9 means 9 or more value[1]: 1 if measurements exist from at least
two distinct bands, 0 otherwise value[2]: 1 if there is at least one
measurement on either side of the selected date, 0 otherwise </DESCRIPTION>
</FIELD>
<FIELD datatype="float" width="10" name="Nearest Measurement Date">
<DESCRIPTION>number of days from the given date to the nearest measurement
used </DESCRIPTION>
</FIELD>
<FIELD datatype="char" name="Verbose" arraysize="256000"/>
<FIELD datatype="char" name="Version" arraysize="20">
<DESCRIPTION>version of flux estimation software </DESCRIPTION>
</FIELD>
<DATA>
<TABLEDATA>
<TR>
<TD>0</TD>
<TD>J0854+2006</TD>
<TD>2.33E11</TD>
<TD>01-Jun-2015</TD>
<TD>1.7683921991208924</TD>
<TD>0.028968543068228363</TD>
<TD>-0.5601663408057809</TD>
<TD>0.02118062686373895</TD>
<TD>911</TD>
<TD>1.0</TD>
<TD/>
<TD>ARCHIVE-2020.06.01</TD>
</TR>
</TABLEDATA>
</DATA>
</TABLE>
```

2.4 ALMA Pipeline Utilization of the Flux Service

When pipeline use of the source catalog flux service is enabled by setting dbservice=True in hifa_importdata(), the pipeline will query the flux service to obtain flux estimates for all calibrators and use the resulting fluxes. Queries will be directed to the URL set by the FLUX_SERVICE_URL environment variable (nominally, this should be the source catalog hosted by the local ARC archive). A backup URL is set by the FLUX_SERVICE_URL_BACKUP environment variable. Results are written to a flux.csv file in the pipeline 'working' directory. In the event that the primary query fails, the hifa_importdata() stage in the weblog will show a slightly reduced QA score (0.9) with a corresponding message. If both queries fail (for any source/SPW/EB), the score will be lower (0.6), again with a corresponding message, and the flux values in the ASDM will be used. ***In this case the problem should be reported, resolved, and the pipeline re-run***. One valid solution is to generate a flux.csv file using au.getALMAFluxcsv(), or use the version of this file which may already exist (see "ADAPT usage of flux service" below).

The recommended configuration of the environment variables is
FLUX_SERVICE_URL = https://almascience.org/sc/flux
FLUX_SERVICE_BACKUP_URL = https://asa.alma.cl/sc/flux

Diagnostics & Troubleshooting

- If the flux service was successfully used by the pipeline, there will be a message in the hifa_importdata() "Pipeline QA" messages (at the bottom of the page) stating "Flux catalog service used."
- There will be a flux.csv file with lines that look like the following:

uid___A002_Xc7111c_X25ff.ms,0,17,4.288079181984935,0.0,0.0,0.0,-0.161,0.0,0.0,"# field=J0522-3627 intents=AMPLITUDE,ATMOSPHERE,BANDPASS,POINTING,WVR origin=DB age=-4.0 queried_at=2021-02-05 21:44:15 UTC"

If the lines contain additional content as illustrated below, it indicates that the old method of flux estimation was used (AU.getALMAfluxcsv()) – emphasis is added to the "additional content":

uid___A002_Xc7111c_X25ff.ms,0,17,4.1064,0.0,0.0,0.0,-0.20401381428145646,0.0,0.0,"# field=J0522-3627 intents=AMPLITUDE,ATMOSPHERE,BANDPASS,POINTING,WVR origin=DB age=-4.0 queried_at=2021-02-05 21:44:15 UTC ***# +-0.1151Jy, freq=680.912GHz, spec_index=-0.204+-0.011, Band3/7_separation=2 days, spixAge=-6 days, Band3age=-6 days, setjy parameters for field 0 (J0522-3627): spix=-0.2040, reffreq='680.9116GHz', fluxdensity=[4.106367,0,0,0], au.getALMAFluxcsv v1.5036 executed on 2021-02-05 22:01:40 UT http://asa.alma.cl/sourcecat/xmlrpc"***

For amplitude calibrators in particular the AU heuristics are not as accurate, but are sufficiently accurate to be ingested to the archive and delivered to PI's.

If on the other hand the flux.csv has lines that look like the following, the ASDM flux estimates were used. ***These are the estimates that existed at the time of the observation and can be significantly less accurately; they need to be carefully checked, and it is recommended to re-run with either FS or AU flux estimates instead.*** As above, the content which indicates this case is emphasized:

uid___A002_Xc7111c_X25ff.ms,0,17,4.176755318134163,0.0,0.0,0.0,0.0,0.0,0.0,"# field=J0522-3627 intents=AMPLITUDE,ATMOSPHERE,BANDPASS,POINTING,WVR ***origin=Source.xml age=N/A queried_at=N/A"***

Adapt/Pipeline Driver Usage of Flux Service

As of the 2020AUG offline release the flux service has been used by Adapt, and specifically the PipelineDriver, for production pipeline runs*. All of the diagnostics above apply. Additional points to note are the following:

- If there was an improper configuration—for example use of an incorrect recipe which did not specify dbservice=True, or the FLUX_SERVICE_URL was not correctly set—then the most likely behavior was for the ASDM fluxes to be used. These should be re-run manually using FS or AU fluxes.
- The “working” directory will contain a flux.csv file. If everything worked as intended, this file will have been generated from flux service queries. If not, it will contain information from the ASDM. This flux.csv file will be copied to the “products” directory in the process of ingestion.
- There will also be a file called “flux.csv.getalmaflux”. This file is for diagnostic purposes, and can be used as a backup for manual runs of the pipeline if desired.

**: use of the flux service was not uniformly adopted across all executives due to an irregularity of release & configuration procedures. Adapt runs which were not properly configured would most likely have used the ASDM flux values; they should be examined and re-run if warranted.*

PI Use of Flux Service for pipeline re-calibration

If PI’s wish to re-create the pipeline calibration from scratch, they can do so with the delivered flux.csv file; this will use the same flux densities assumed in the calibration of their delivered data.

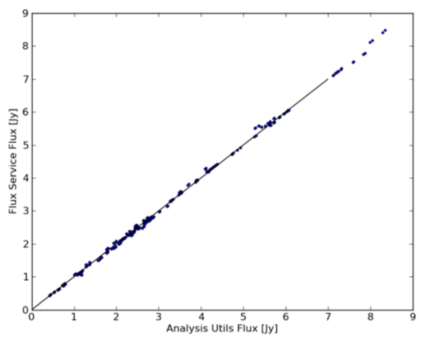
If desired the PI can also set the FLUX_SERVICE_URL environment variables and modify the hifa_importdata() call to have dbservice=True.

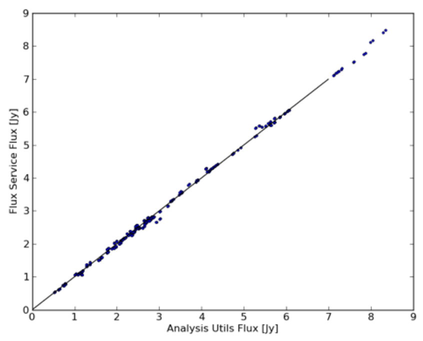
3 Tests of the flux density estimation

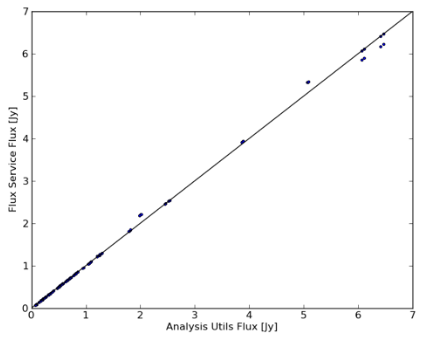
3.1 Pipeline tests of the flux service

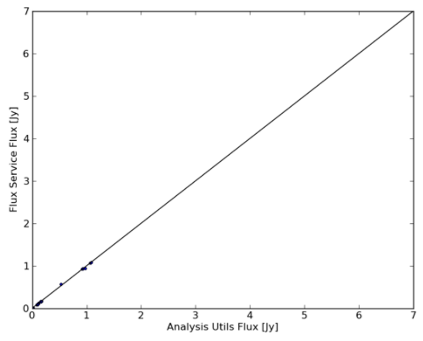
This is reporting AU vs pipeline-queried flux service results for the new pipeline (6.1.1.10) for the bench-marking 36 MOUS sets, and quantitative summary stats below. This was set up to locally query almascience.nrao.edu/sc/flux .. as an ARC example, there were no failed queries or other apparent issues. Results are broken apart by target intent.

AMPLITUDE Median fractional difference: -1.195e-05
mean fractional difference: 0.000125363937734 Nquery: 408
Max and Min fractional difference: 0.0616 -0.1036
BANDPASS Median fractional difference: -0.00033
mean fractional difference: -0.00124 Nquery: 408
Max and Min fractional difference: 0.0616 -0.1036
PHASE Median fractional difference: 7.3127e-06
mean fractional difference: 0.00260 Nquery: 408
Max and Min fractional difference: 0.09484 -0.03833
CHECK Median fractional difference: 2.673e-05
mean fractional difference: 0.00179 Nquery: 68
Max and Min fractional difference: 0.06417 -0.0324









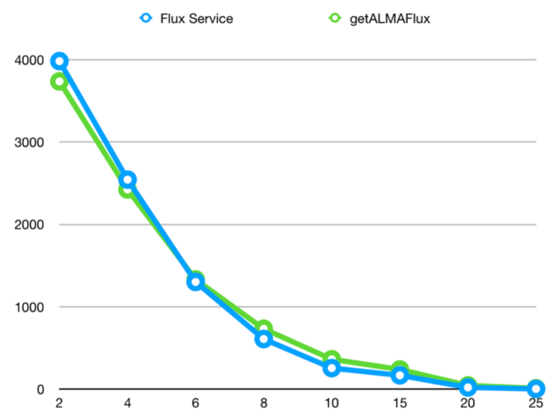
No problem was reported from the Pipeline tests for 2020AUG (PLdriver) validation.

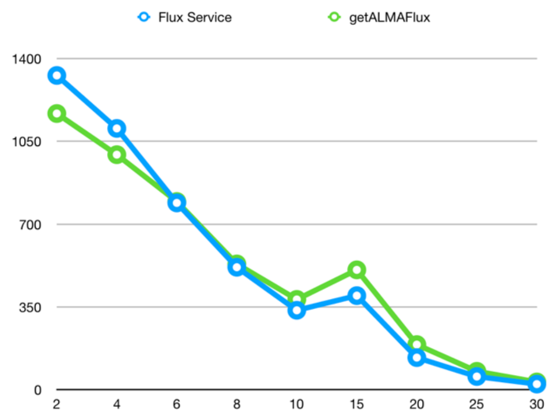
3.2 Statistical Tests of the Flux Density Estimation

In order to test the quality of the flux service fitting algorithm (ARCHIVE-2020JUN) estimating the flux densities grid source measurements between 2015-01-01 and 2019-12-31 are invalided in a test database (all measurements of the respective day), and the estimate and the measurement flux densities are compared.

The run consisted of 48190 queries (24095 for SC flux service and getALMAFlux each, queried separately) in total, 12999 queries resulted in fit cases (i.e. 11096 cases resorted to the fallback), which were investigated further. This fraction of fit cases is reasonable, because the fit criteria are tuned to the grid monitoring cadence, and the statistical method used here, which eliminates measurements from the estimate in order to compare the two, effectively doubles the monitoring cadence (estimating a measurement eliminated from a 14-day cadence requires interpolating measurements separated by 28 days). Note also, that comparing estimate and measurement involves two errors. Both circumstances substantially increase the observed scatter over the uncertainty of a normal query at the grid monitoring cadence (see examples in Section 4). Therefore the method does not directly inform about the typical query statistics in operations, but allows for a useful comparison between the two algorithms.

There was one spontaneous script failure, presumably due to a connectivity issue. The run was also interrupted because of maintenance of the test database. The two flux queries followed the previous ones on average every 6sec interspersed with several in-/validation queries.





Histograms of percentage differences between flux density estimates and measurements for Bands 3 (left) and Bands 6/7 (right). The blue curve shows the statistics for the fitting results and the green curve for the getALMAFlux results with the exact same queries. Note, that the absolute quality is not representative of the flux normalisation for science projects, because the omission of measurements doubles the cadence and the method includes the error of the individual measurements.