

VieSOFT User Guide

Frédéric Jaron

Technical University of Vienna, Austria

e-mail: `Frederic.Jaron@tuwien.ac.at`

August 28, 2026

Abstract

This document describes the software package VieSOFT. The purpose of the included computer programs is the modeling and the removal of VLBI source structure from observational data. The installation procedure for UNIX-type operating systems is described. The usage of the programs is explained with the help of concrete examples.

Contents

1	Introduction	3
2	Installation	4
2.1	fitngauss.py	4
2.2	difxstruc	4
2.3	rundifxstruc.sh	4
3	Usage	5
3.1	fitngauss	5
3.1.1	Output of fitngauss.py	7
3.2	difxstruc	8
3.3	rundifxstruc.sh	8

1 Introduction

Geodetic Very Long Baseline Interferometry (VLBI, Sovers et al. 1998) is commonly carried out under the assumption that the observed target sources are perfect points on the sky. In reality, many of these sources, which mainly belong to the class of radio-loud active galactic nuclei (AGN, Blandford et al. 2019), have structure that is often resolved by VLBI. Neglecting the issue of source structure introduces an unmodeled systematic error in geodetic VLBI analysis (Anderson and Xu, 2018). This document describes a software package for the treatment of source structure in observational VLBI data. The application of the implemented algorithms has been demonstrated to be able to remove the impact of source structure (Jaron et al., 2024). The efficacy that the application of these programs has on the end results heavily depends on the quality and the reliability of information about the structure of the sources to be corrected. Since the appearance of AGN is subject to continuous changes, the software needs to be configured with images that typically must not be detached in time by more than about two weeks from the data to be corrected.

The algorithms implemented in this software package are based on the treatment of source structure as described in Charlot (1990), in particular the formalism in terms of Gaussian functions (his Section II c). The development of our solutions was the result of very fruitful discussions and close collaboration of researchers working at different institutes across Europe. The team consisted of Patrick Charlot and Arnaud Collioud (Laboratoire d’astrophysique de Bordeaux, France), Jakob Gruber (Bundesamt für Eich- und Vermessungswesen, Austria), Johannes Böhm, Frédéric Jaron, Hana Krásná, and Axel Nothnagel (Technische Universität Wien, Austria), Iván Martí-Vidal (University of Valencia, Spain), Victor Pérez-Díez (Yebes Observatory, Spain). The work was funded by the Austrian Science Fund (FWF), Grant-DOI 10.55776/P35920. Victor Pérez-Díez wrote the initial version of `fitngauss.py` and operated the VGOS imaging pipeline (as introduced in Pérez-Díez et al., 2024), generating the images used for the examples presented in this User Guide and in Jaron et al. (2024) and Jaron et al. (submitted).

2 Installation

Here we describe how the programs included in VieSOFT are set up for usage on your local UNIX-type computer.

2.1 `fitngauss.py`

`fitngauss.py` is a Python3 script.

Dependencies:

- NumPy
- AstroPy
- SciPy
- matplotlib

We suggest that you set up a conda environment with these Python packages installed. You may copy the `fitngauss.py` to a directory that is included in your `$PATH` environment variable. The following command should result in the creation of a conda environment, named `fitngauss`, that allows the execution of `fitngauss.py`:

```
conda create -n fitngauss numpy astropy scipy matplotlib
```

Activate this environment by

```
conda activate fitngauss
```

and you should be ready to go.

For our tests we have found the following versions to work: `astropy-8.0.1`, `scipy-1.18.0`, `numpy-2.5.2`, `matplotlib-3.11.0`. Other versions may or may not work.

2.2 `difxstruc`

`difxstruc` is written in pure ANSI C. Its only dependency is on the standard C libraries. We recommend compilation with the GNU C compiler `gcc`. Execution of the script `make.sh` in the `src` directory produces the executable `difxstruc`.

2.3 `rundifxstruc.sh`

`rundifxstruc.sh` is an automation script, written in BASH. You can copy this script to a directory that is contained in the `$PATH` environment variable, for convenient execution.

3 Usage

3.1 fitngauss

Syntax:

```
usage: fitngauss.py [-h] [--bandfile BANDFILE] [--srcfile SRCFILE]
                  [--srcoutfile SRCOUTFILE] [--fitsdir FITSDIR]
```

Fit *n* Gaussian components to FITS image

options:

-h, --help	show this help message and exit
--bandfile BANDFILE	Band configuration file
--srcfile SRCFILE	Input source configuration file
--srcoutfile SRCOUTFILE	
	Output source configuration file
--fitsdir FITSDIR	Directory where FITS files are stored

In the following we give explanations and examples for the four necessary configuration files.

bandfile

This text file contains the definition of the frequency bands. It consists of lines of the syntax: **BAND** <id> <low freq> <high freq>

Each line starts with the keyword **BAND**.<id> is the band identifier. For example, X or S. The floating point numbers <low freq> and <high freq> are the high and low frequency bounds of the band, both given in MHz.

Here is an example for the conventional VGOS frequency setup (that was introduced by Niell et al. 2018):

```
BAND A 3000.40 3480.40
BAND B 5240.40 5720.40
BAND C 6360.40 6840.40
BAND D 10200.40 10680.40
```

There are no restrictions concerning the file name, a typical choice would be 'bands.conf', for example.

srcfile

This text file contains the *a priori* configuration of Gaussian components, which eventually will describe the brightness distribution of the source. This means that the values given in this file are the input for the optimization algorithm which adjusts the components to match the source image. The file is organized in lines. It should contain one line of the form

DEC <dec>

where <dec> is the declination of the source in degrees. For example,

DEC 58.1

configures the declination of the source (here 0059+581) to be 58.1°.

The file should also contain one line of the form:

REF <x> <y>

where <x> and <y> are the coordinates (in mas) of a source reference point, relative to its a priori position. This line should be set to:

REF 0 0

This means that the reference point should be set to the origin of the coordinate system of the image plane. With this option it is possible to shift the reference point, but this should not be done in operational usage of the software.

Finally, the file can contain an arbitrary number of lines of the form:

COMP <f_low> <f_high> <flux> <x> <y> <dx> <dy> <psi> <alpha> <beta>

These lines contain the configuration of Gaussian components describing the radio brightness distribution of the source. The columns <f_low> and <f_high> contain the low and high ends of the frequency range which the component entry refers to. If that range is contained in the range of any of the frequency bands defined in the **bandfile** (see above), then this component will be processed for this band, i.e., the corresponding fits file (see below) will be used. The column <flux> contains the peak flux density of the component, given in Jy. The position of the component is given by the <x> and <y> coordinates in mas, relative to the origin of the image. The columns <dx> and <dy> contain the extension of the Gaussian components, in terms of their standard deviations, given in mas. The column <psi> contains the angle ψ , describing the rotation of the elliptical Gaussian component, in degrees, in the positively mathematical sense (i.e., anti-clockwise). The values <alpha> and <beta> contain the radio spectral index α and a curvature β . These values are, however, currently not yet taken into into by the program `fitnngauss.py` and can be set to zero, or any other value.

Example **srcfile** for the source 0059+581

```
DEC      58.1
REF      0 0
COMP 3000.40 3480.40 0.000273 0.169 0.091 0.132 0.140 66.98 0 0
COMP 3000.40 3480.40 0.000020 0.080 0.044 0.334 0.282 0.00 0 0
COMP 5240.40 5720.40 0.000310 0.169 0.095 0.159 0.173 36.10 0 0
COMP 5240.40 5720.40 0.000011 -0.047 -0.066 0.822 0.661 24.13 0 0
COMP 6360.40 6840.40 0.000322 0.171 0.098 0.166 0.188 29.88 0 0
COMP 6360.40 6840.40 0.000013 -0.070 -0.096 0.948 0.750 30.33 0 0
COMP 10200.40 10680.40 0.000178 0.097 0.273 0.141 0.146 76.22 0 0
COMP 10200.40 10680.40 0.000035 -0.062 -0.004 1.023 0.873 23.41 0 0
COMP 10200.40 10680.40 0.000032 -1.824 -1.806 0.300 0.357 16.67 0 0
COMP 10200.40 10680.40 0.000247 0.271 0.056 0.159 0.206 0.33 0 0
COMP 10200.40 10680.40 0.000017 0.451 -0.386 0.215 0.157 42.90 0 0
COMP 10200.40 10680.40 0.000117 0.051 0.019 0.118 0.188 22.99 0 0
COMP 10200.40 10680.40 0.000017 -0.163 -0.817 0.346 0.155 0.00 0 0
```

The name of the **srcfile** can in principle be freely chosen. However, the program `difxstruc` (see Sect. 3.2) expects these files to follow the naming convention `name.src`,

where **name** is the identifier of the source. A reasonable naming strategy could be to name this **srcfile**, which has the meaning of an *input* file for the adjustment algorithm, **name.in**. The output file, i.e., **srcoutfile** (see below), can then have the suffix **.src**.

The program **fitngauss.py** takes the values provided in this file and the number of components per frequency range as a priori configuration for the source structure adjustment algorithm. To be precise, it will leave the number of components unchanged, but it will adjust (almost) all of the parameters in a least-squares manner to closely match the brightness distribution provided in the image file. The parameters which are not being adjusted are the frequency range and the spectral parameters α and β .

srcoutfile

The **srcoutfile** has the same syntax as the **srcfile** (described above). It contains the results of the optimization procedure carried out by **fitngauss.py**, based on the **srcfile**. This means that it has the same number of components per frequency range, but their parameters have been adjusted to match the brightness distribution, provided in the image file, as closely as possible. The results can be checked in the output panels. If the results are not yet satisfactory, then fine tuning of the input parameters and/or the addition of another component may lead to improvement. A typical procedure would be to start with a minimal number of components (e.g., only one) and increase the number step-by-step in each iteration until the results do not improve any more. It is highly discouraged to increase the number of components to an unnecessary amount. In practical terms, one may copy the **srcoutfile** to the **srcfile**, overwriting it in the process, add another component and run the program again.

The name of the **srcoutfile** can be freely chosen, but since **difxstruc** expects the format **name.src**, it is recommended to follow this convention.

fitsdir

The **fitsdir** is a directory that contains the images of the source structure in the fits (Pence et al., 2010) file format. The directory itself can have any name and be located at any position. The fits files within that directory, however, have to be named after the following convention: **name_B.fits**, where **name** is the source name, and **B** is the band identifier. The band identifier has to be identical to the one given in the **bandfile** for the frequency range that correspond to the data contained in the fits file.

Fits files related to the source 0059+581

```
0059+581_A.fits
0059+581_B.fits
0059+581_C.fits
0059+581_D.fits
```

3.1.1 Output of fitngauss.py

After a successful completion, the program **fitngauss.py** outputs two files. One file is the above mentioned **srcoutfile**, which contains an adjusted version of the input model. The other output file is a PDF containing a panel of the original image, the modeled image, and the residuals. An example of the output panel is shown in Fig. 1, where the results for the source 0059+851 and VGOS Band A are shown. Similar panels are produced for all other bands that are in listed the **bandfile**. During the iterative adjustment procedure, the user will monitor these panels as a direct feedback of the quality of the current model.

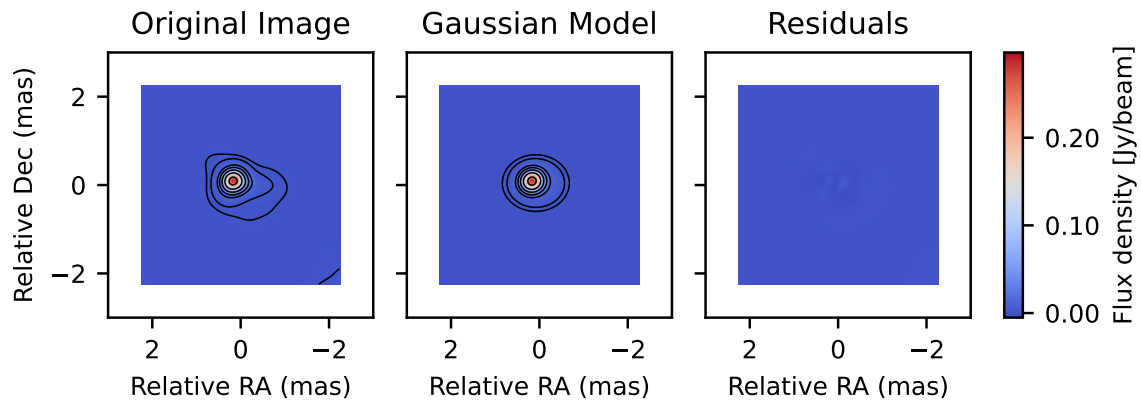


Figure 1: Example output file of the program `fitngauss.py`. The panel on the right-hand side shows the original image. The panel in the middle shows a visualization of the optimized representation of the original image by a set of Gaussian components. The panel on the left-hand side shows the residuals in terms of observed minus computed.

3.2 difxstruc

The program `difxstruc` carries out the actual removal of the source structure from a DiFX visibility file, i.e., the so-called SWIN file.

```
difxstruc -i <DiFX .input file> -s <src file> -d <input SWIN file>
          -o <output SWIN file>
```

All of the command line options are mandatory, and the program processes one SWIN file at a time. (For the processing of the multiple files of a whole session the automation script `rundifxstruc.sh` can be used.) The `<DiFX .input file>` is the DiFX .input file of the scan to be processed. The `<src file>` contains the parametrization of the source structure in terms of Gaussian component. This file will typically be the result of the generation with the script `fitngauss.py`, as described in the section above. Please note that the source structure that is described in the src file should match the expected structure of the scan that is to be processed.

Important!

The efficacy of the source structure correction heavily depends on the accuracy of the model parameters provided in the src file. If that parametrization does not sufficiently match the source structure that is observed in the scan under consideration, then the resulting visibilities might even **contain more structure** than before the processing.

3.3 rundifxstruc.sh

Processing of a whole VLBI session, containing observations of multiple sources, is possible in an automated way with the script `rundifxstruc.sh`.

```
rundifxstruc.sh <DIFXDIR> <DESTDIR> <SRCDIR> [<SOURCE>]
```

The arguments `<DIFXDIR>`, `<DESTDIR>`, and `<SRCDIR>` are mandatory, `<SOURCE>` is optional. `<DIFXDIR>` is the directory that contains the original SWIN files, that are to be

processed. The .input files have to be in the <DIFXDIR>, while the actual SWIN files must be in the .difx sub-directories. The <DESTDIR> is the directory where the processed (i.e., corrected) SWIN files are written to. All other files of the <DIFXDIR> (and in particular the unchanged SWIN files) are copied to the <DESTDIR> as well, such that this directory is ready to be processed just as the original directory. In particular, it is possible to run `difx2mark4` on the <DESTDIR> for the conversion to the mk4 format needed for fringe-fitting with the HOPS fourfit program. <SRCDIR> is the name of a directory that contains the .src files for the correction of the session to be processed. The .src files located in this directory should contain appropriate models of the source structure for the correction of that particular session. In practical terms this means that the image information, that these models are based on, should not be separated in time by more than about two weeks from the observing date of the session to be corrected. With the optional <SOURCE> argument it is possible to only process scans related to a particular source. All other scans will be left unchanged.

References

- J. M. Anderson and M. H. Xu. Source Structure and Measurement Noise Are as Important as All Other Residual Sources in Geodetic VLBI Combined. *Journal of Geophysical Research (Solid Earth)*, 123(11):10,162–10,190, Nov. 2018. doi: 10.1029/2018JB015550.
- R. Blandford, D. Meier, and A. Readhead. Relativistic Jets from Active Galactic Nuclei. *Annual Review of Astronomy and Astrophysics*, 57:467–509, Aug. 2019. doi: 10.1146/annurev-astro-081817-051948.
- P. Charlot. Radio-Source Structure in Astrometric and Geodetic Very Long Baseline Interferometry. *Astronomical Journal*, 99:1309, Apr. 1990. doi: 10.1086/115419.
- F. Jaron, L. Baldreich, J. Johm, P. Charlot, A. Collioud, J. Gruber, H. Krasna, I. Martí-Vidal, A. Nothnagel, and V. Pérez-Díez. Mitigating source structure in geodetic VLBI on the visibility level. In E. Ros, P. Benke, S. A. Dzib, I. Rottmann, and J. A. Zensus, editors, *Proceedings of the 16th EVN Symposium*, pages 145–148, Sept. 2024.
- F. Jaron, V. Pérez-Díez, H. Krasna, J. Böhm, P. Charlot, A. Collioud, J. Gruber, I. Martí-Vidal, and A. Nothnagel. Removing the Effect of Source Structure from Correlated VLBI Data. *Radio Science*, submitted.
- A. Niell, J. Barrett, A. Burns, R. Cappallo, B. Corey, M. Derome, C. Eckert, P. Elosegui, R. McWhirter, M. Poirier, G. Rajagopalan, A. Rogers, C. Ruszczyk, J. SooHoo, M. Titus, A. Whitney, D. Behrend, S. Bolotin, J. Gipson, D. Gordon, E. Himwich, and B. Petrachenko. Demonstration of a Broadband Very Long Baseline Interferometer System: A New Instrument for High-Precision Space Geodesy. *Radio Science*, 53(10): 1269–1291, Oct. 2018. doi: 10.1029/2018RS006617.
- W. D. Pence, L. Chiappetti, C. G. Page, R. A. Shaw, and E. Stobie. Definition of the Flexible Image Transport System (FITS), version 3.0. *Astronomy & Astrophysics*, 524: A42, Dec. 2010. doi: 10.1051/0004-6361/201015362.
- V. Pérez-Díez, I. Martí-Vidal, E. Albentosa-Ruiz, J. González-García, F. Jaron, T. Savolainen, M. H. Xu, and R. Bachiller. Towards an astronomical use of new-generation geodetic observations. I. From the correlator to full-polarization images. *Astronomy & Astrophysics*, 688:A151, Aug. 2024. doi: 10.1051/0004-6361/202348633.

O. J. Sovers, J. L. Fanelow, and C. S. Jacobs. Astrometry and geodesy with radio interferometry: experiments, models, results. *Reviews of Modern Physics*, 70(4):1393–1454, Oct. 1998. doi: 10.1103/RevModPhys.70.1393.