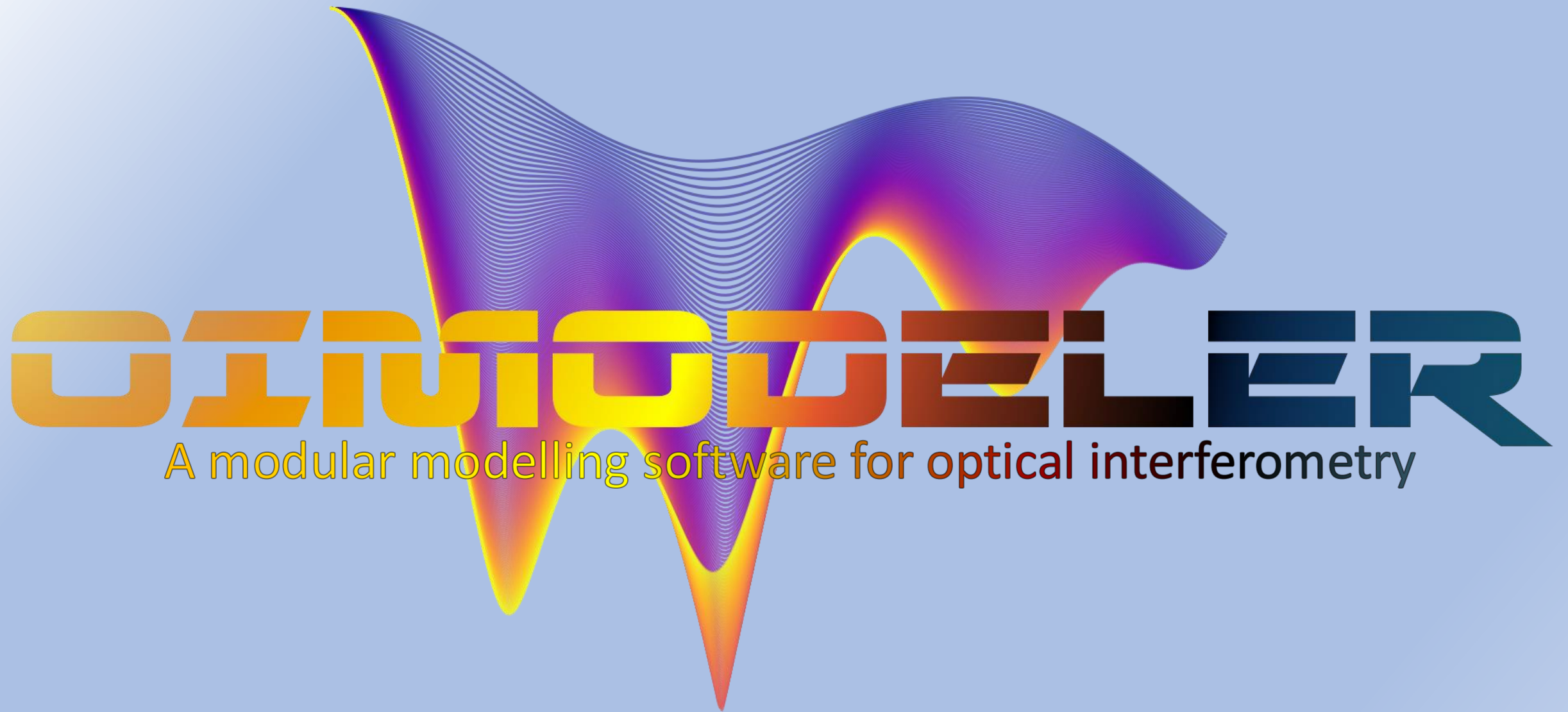




The oimodeler team: Anthony Meilland, Alexis Matter, Marten Scheuck, Luna van Haastere, Margaux Abello, József Varga, James Leftley, Paul Boley, Bruno Lopez, Michiel Hogerheijde, Roy van Boekel, Romain Petrov, Florentin Millour, Julien Drevon, Jacob Isbell, Eric Pantin, Gerd Weigelt, Bill Danchi, Dieter Schertl, Rens Waters, Julia Kobus, Violeta Gámez Rosas, Claire Lykou, Lei Chen, Philippe Priolet, ...

Context of the project (2021)

oimodel.py

Ideas and skeleton for a
general, modular, panchromatic,
and hopefully fast
model-fitting tool
in python

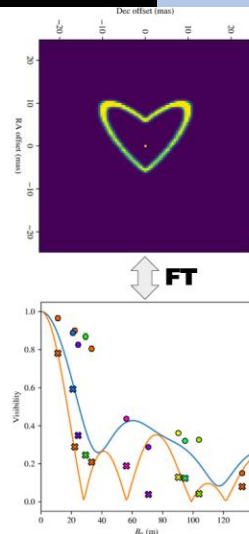
July 2021

The quest for the ultimate modeling tool for optical-IR interferometry

The MATISSE modeling working group:

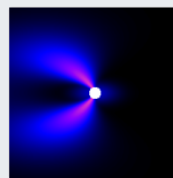
Bill Danchi, Julien Drevon, Violeta Gámez Rosas,
Michiel Hogerheijde, Jacob Isbell, Julia Kobus,
Bruno Lopez, Alexis Matter, **Anthony Meilland**,
Florentin Millour, Eric Pantin, Dieter Schertl, Marten
Scheuck, Roy van Boekel, **József Varga**, Rens
Waters, Gerd Weigelt

MATISSE Science Team meeting, 2021 November 18



MATISSE group

Real time astrophysical models



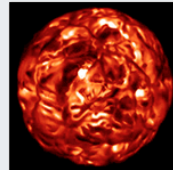
Kinematic Be disk

Model of the geometry (size and shape) and kinematics (rotation and expansion) of circumstellar, flat, rotating disks, relevant to Be stars. It is suited to interpret spectro-interferometric data obtained on emission lines formed in the disk.



Disk and stellar continuum – DISCO

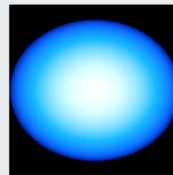
Model
free
given



Evolutionary Stellar simulations famous



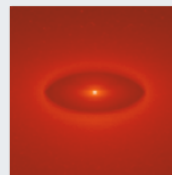
Binary Phenomena mass



Analytical Limb-darkening Elliptical or Spherical – ALDES

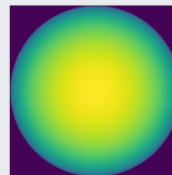
ALDES provides intensity maps (images) or 1d intensity profiles for spherical or elliptical stars showing the limb darkening (LD) effect. Different LD laws are offered: uniform disk, linear, power law, quadratic, square root, logarithmic and four-parameter.

Precalculated grids of astrophysical models



Supergiant B[e] with HDUST

Grid of models for B[e] supergiant stars computed with the 3d Monte Carlo radiative transfer code HDUST. The non-spherical circumstellar envelope (CSE), composed of gas (hydrogen) and dust (silicate), is modelled considering a bimodal outflow description (two-component wind).



Limb-darkening with SATLAS

Grid of models providing intensity maps for spherically symmetric stars, showing the limb darkening effect. The models were computed with the SATLAS model stellar atmospheres for several spectral bands. Data is provided for FGK dwarfs and red giants.

JMMC/AMHRA

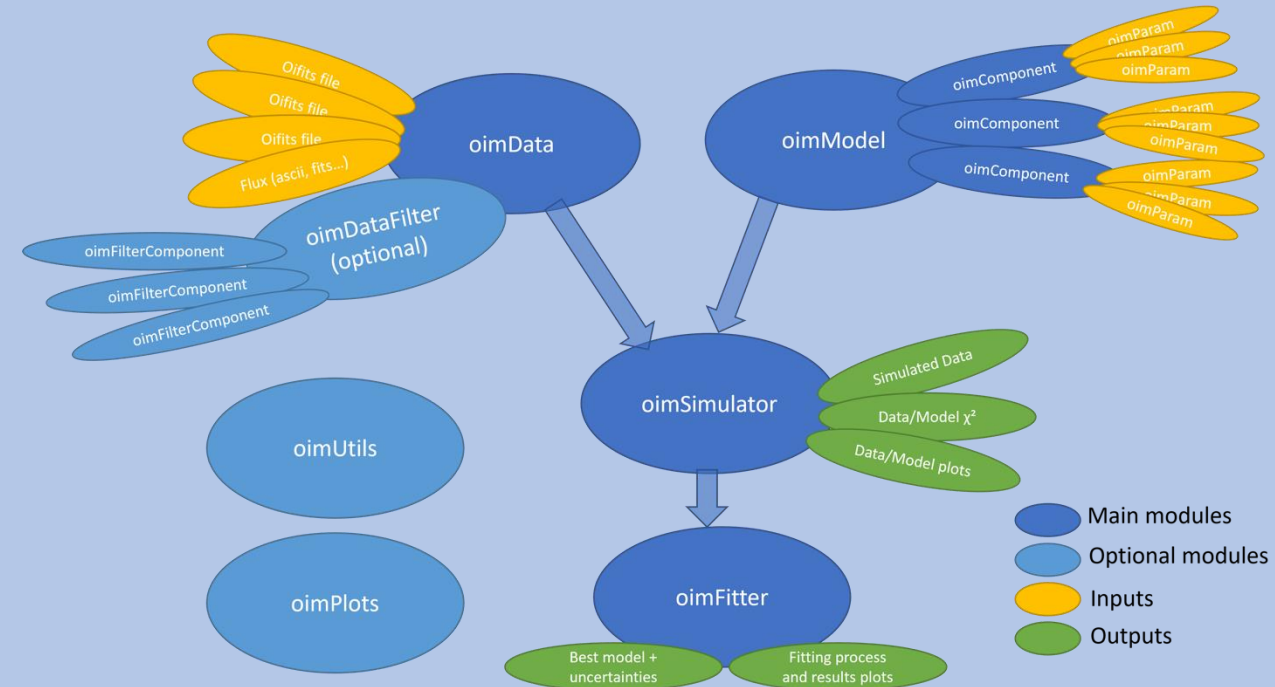
1 - The oimodeler project

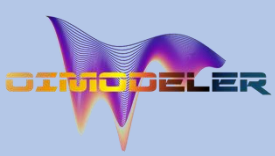
Why?

- Visibility data → Images

Aims: developing an

- open source,
- modular (object-oriented),
- and easily expandable
- python library
- to model optical interferometric data





1 - The oimodeler project

Main Functionalities :

- Build **complex models from components** defined **in Fourier or image plane**
- Add **chromaticity** and **time-dependence** to any model-parameters
- Import **images**, **image-cubes**, and **radial profiles** from **RT-models**
- Simulate **all possible data compatible to the OIFITS2 format**
- Manipulate data in the **OIFITS2 format**: cut, smooth, bin, normalize, recompute errors, calibrate ...
- Add non-OIFITS2 format **flux and spectra measurements** to your fit
- Easy **data/models comparison** and **χ^2 computation**
- Perform **model-fitting** and **error estimation** with various methods
- easily **plot results** and produce high **quality figures for publication**

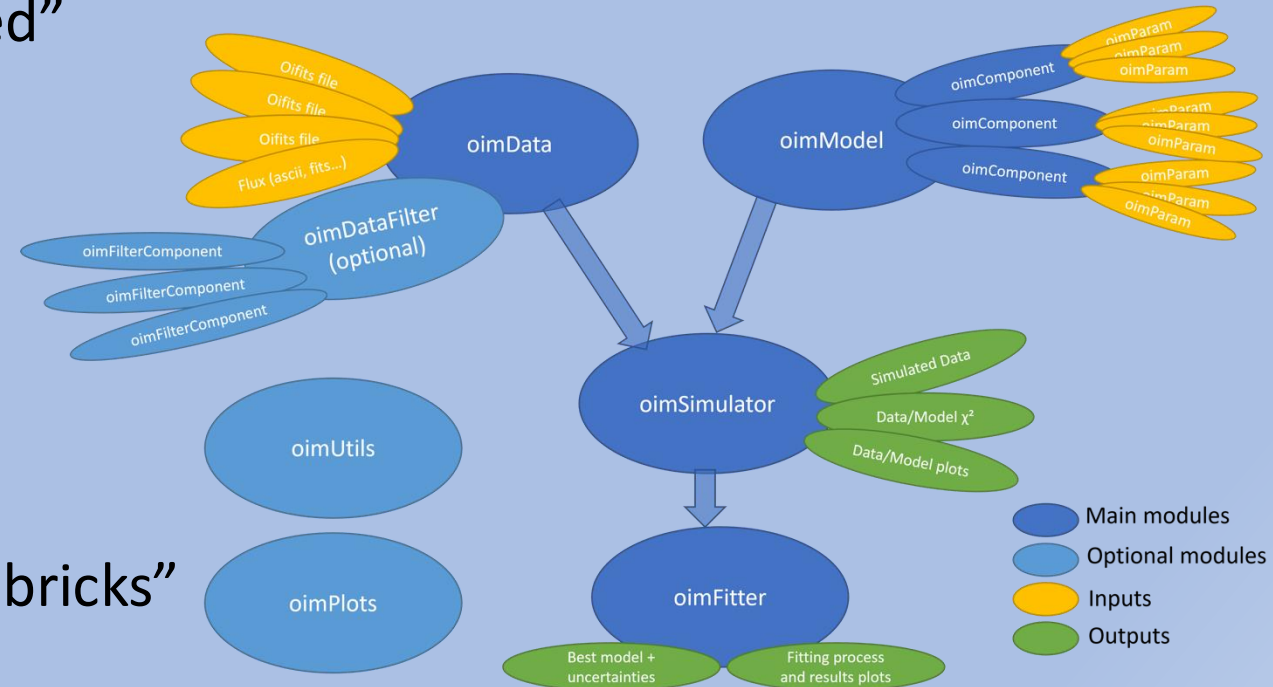
1 - The oimodeler project

- **oimData**: Wrapper for oifits data is astropy.io.fits format

- Contains both the oifits data and “optimized” vectors of data and coordinates (u,v,wl,t)
- Import flux data (from ascii) : **oimFluxData**
 ⇔ conversion to OI_FLUX table
- Possibility of filtering/modifying data (cut, smooth, bin, reflag...): **oimDataFilter**

- **oimModel**: “lego” Model class

- **oimComponents**: model components or “bricks”
 - Fourier-based analytical formulas
 - 2D-image-based: computed using FFT
 - 1D-image-based: using Hankel transform (experimental)
- **oimParam**: components parameters
 - Can be chromatic and/or time-dependent using parameter interpolators
 - Can be linked together by mathematical formula



1 - The oimodeler project

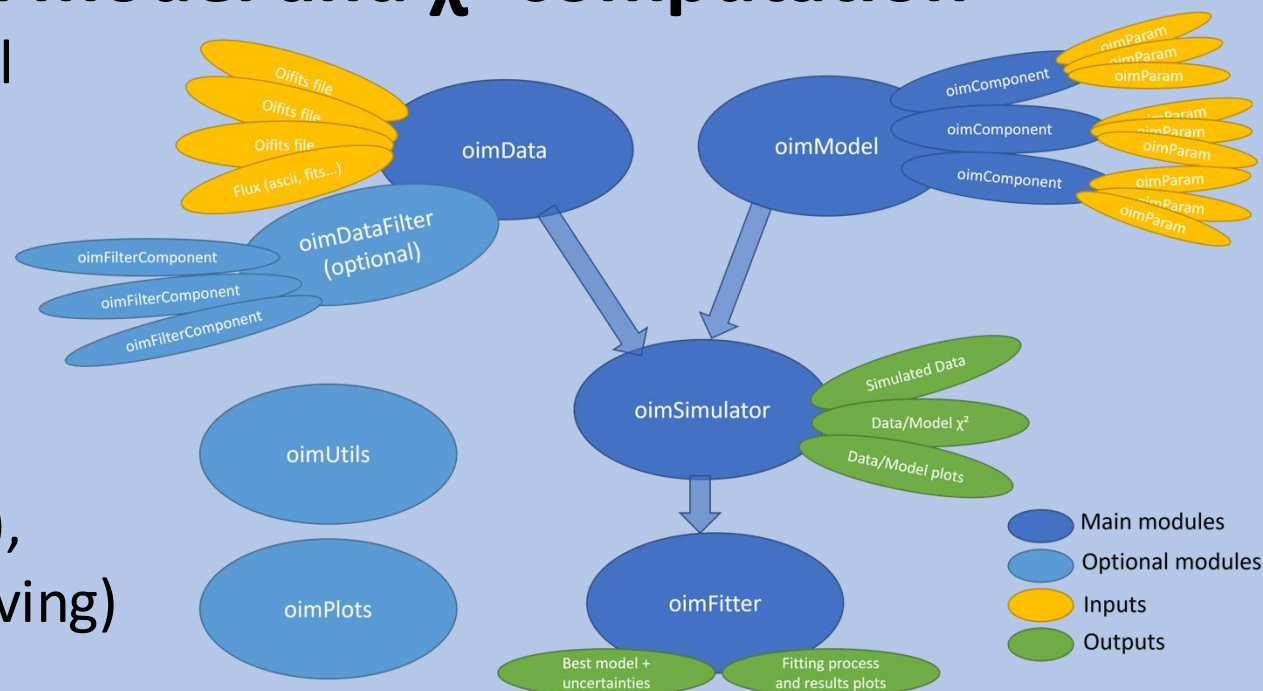
- **oimSimulator**: simulate data from model and χ^2 computation

- Can simulate any kind of oifits2 data for all instruments:
VIS2DATA, VISAMP
(absolute, differential, correlated flux)
VISPHI (absolute and differential),
T3AMP & T3PHI, FLUXDATA
- Two modes: fast (only χ^2 for model-fitting),
slow (simulated data for plotting and saving)

- **oimFitter**: Model fitting class(es)

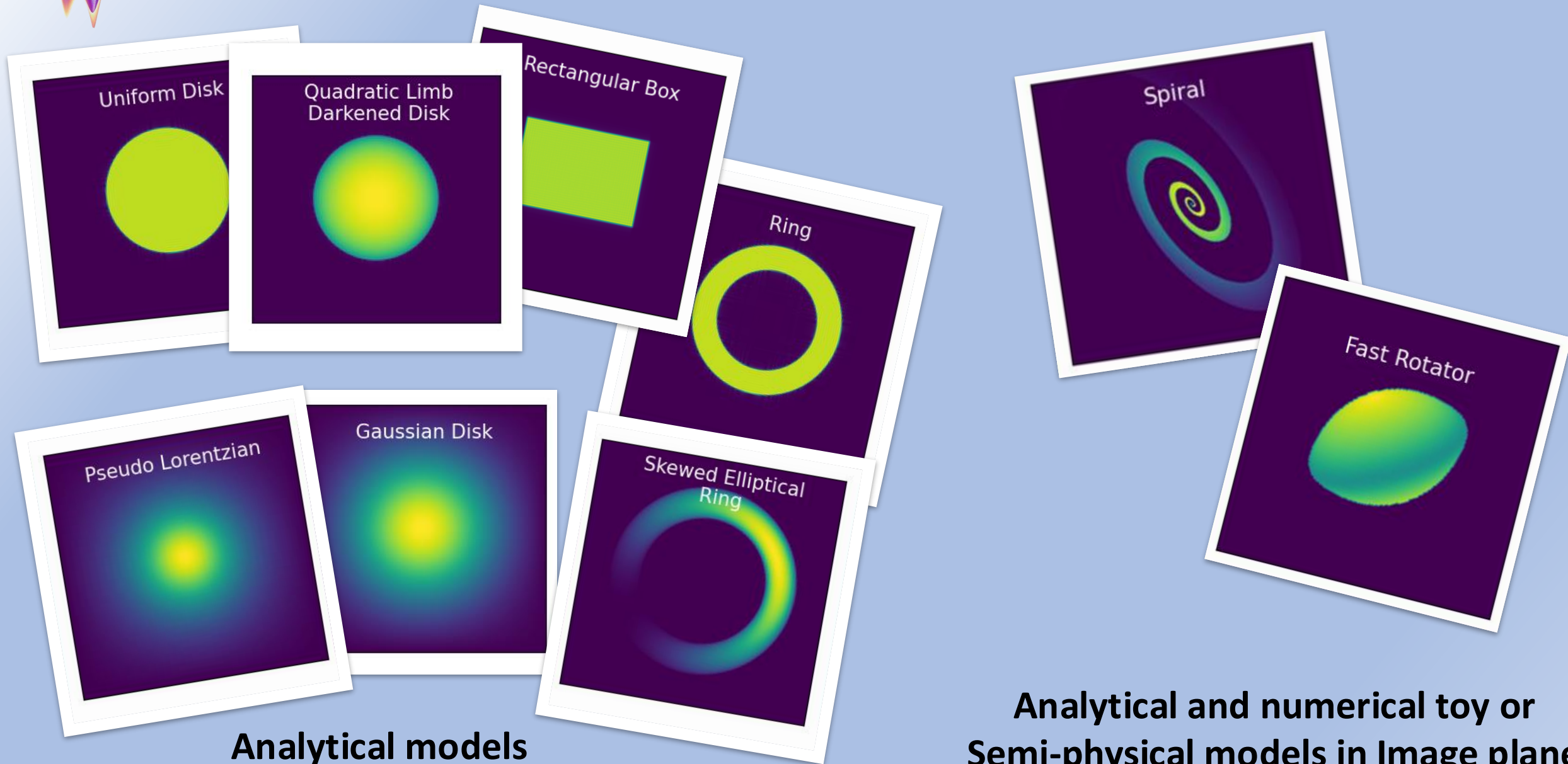
- Currently: only an emcee-based implemented: oimFitterEmcee (MCMC)
- Use the oimSimulator for χ^2 computation
- Minimized χ^2 and give best (or median) parameters and uncertainties (emcee “style”)

- **oimUtils & oimPlots** : Helper classes





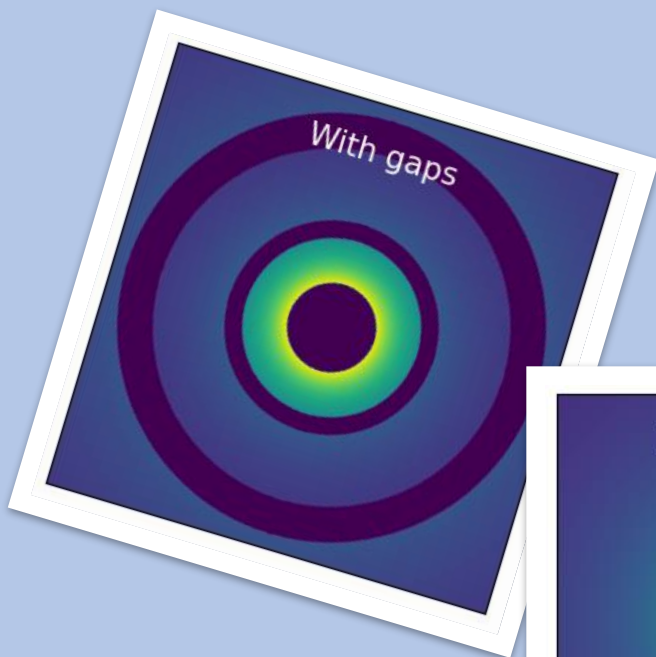
2 - Some of the components available in oimodeler



**Analytical models
directly defined in Fourier plane**

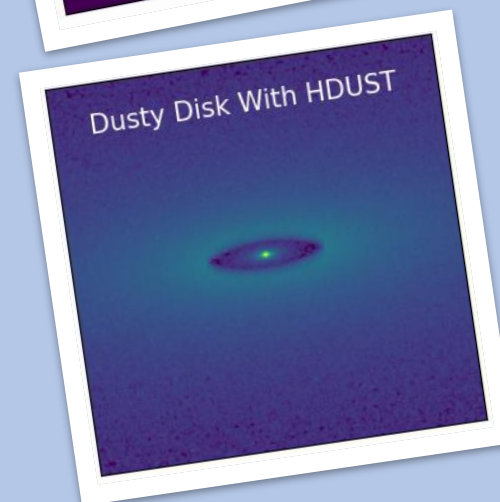
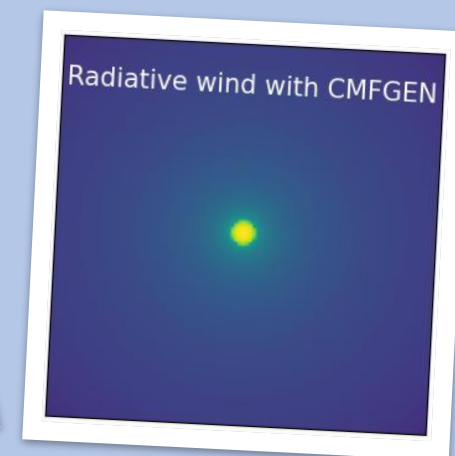
**Analytical and numerical toy or
Semi-physical models in Image plane**
FFT is used to compute visibilities

2 - Some of the components available in oimodeler



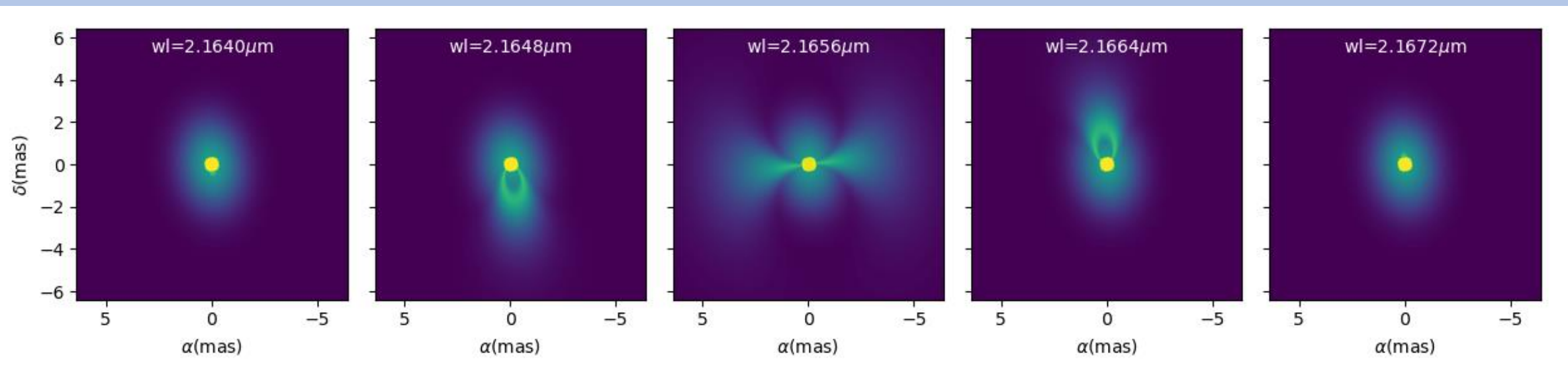
Radial profiles

Toy-models or output from stellar atmosphere models
Hankel-Transform is used to compute visibilities



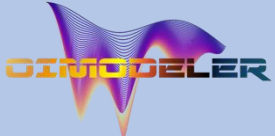
Images or images cubes in fits format
for instance, outputs from radiative transfer codes

2 - Some of the components available in oimodeler

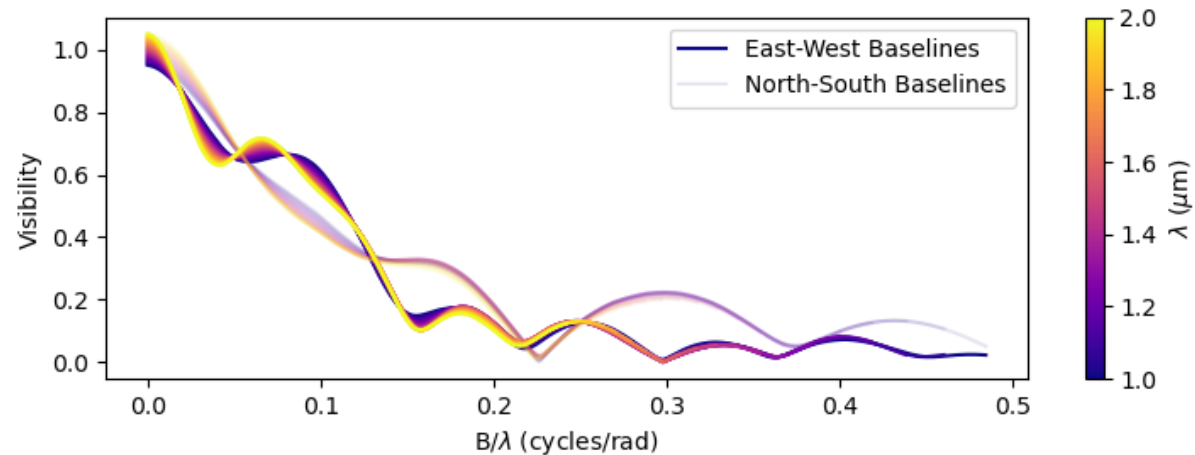
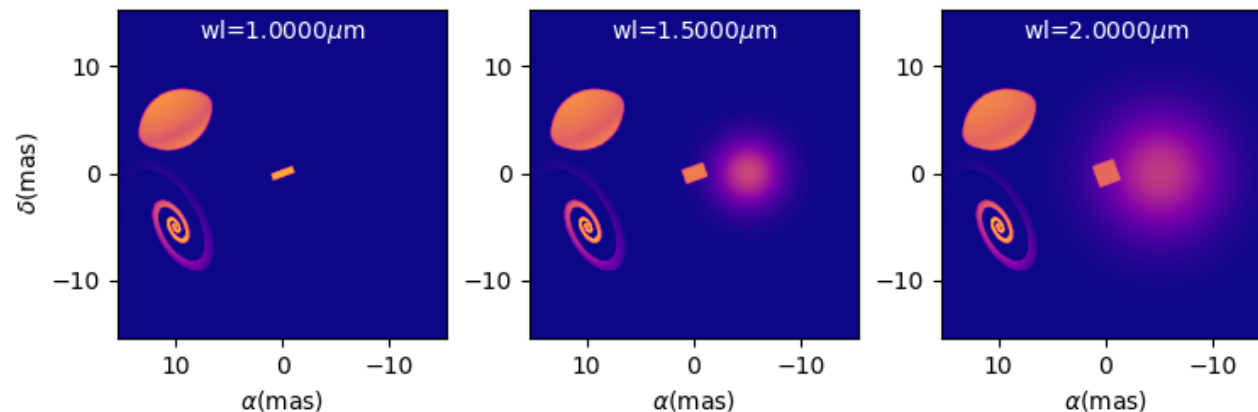
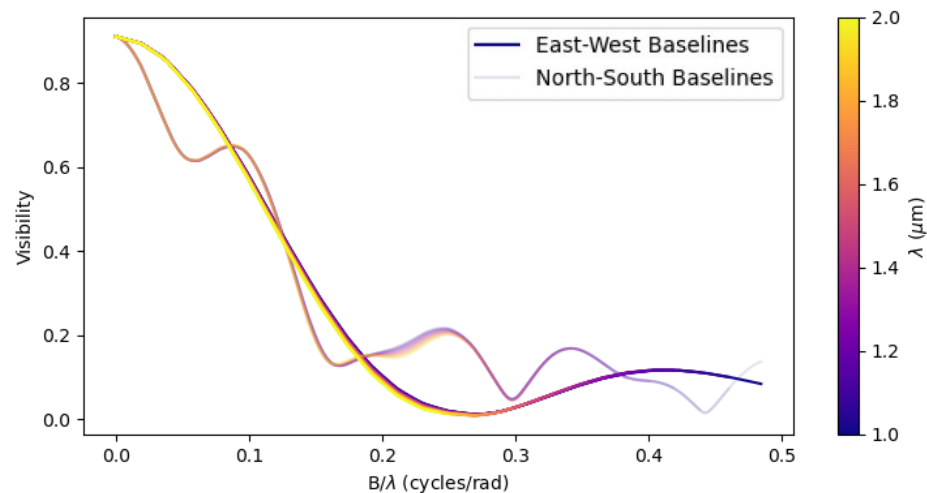
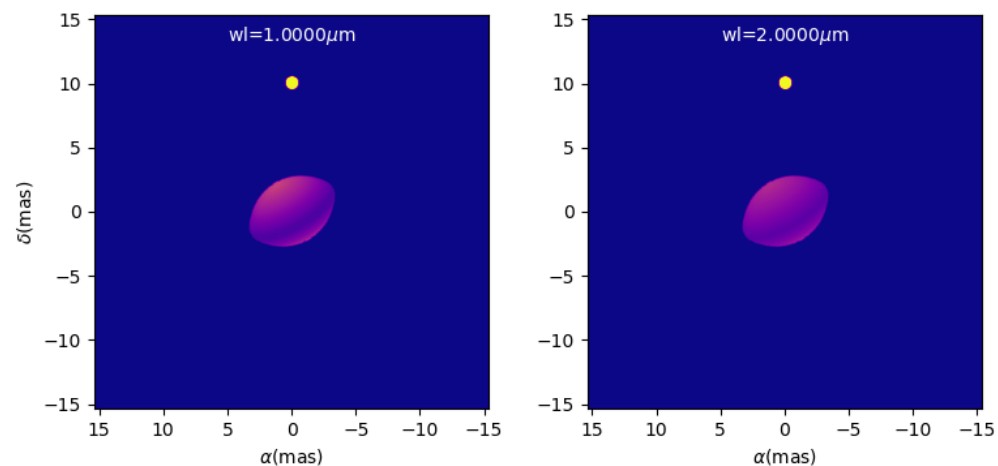


Kinematic models in absorption or emission lines

(such as Rotating/expanding disk from Meilland et al. 2012)

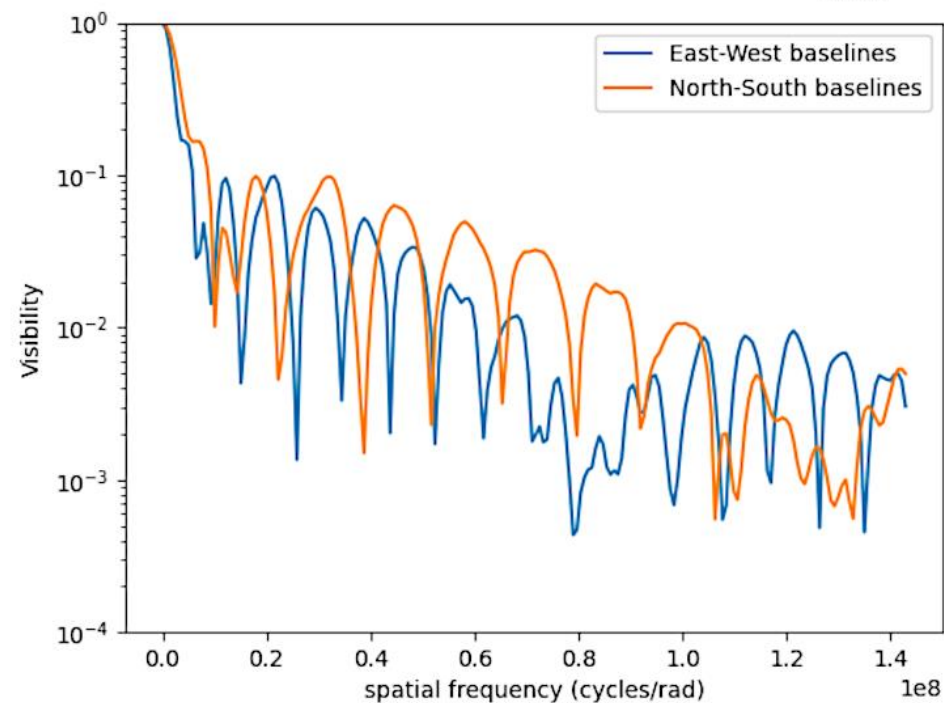
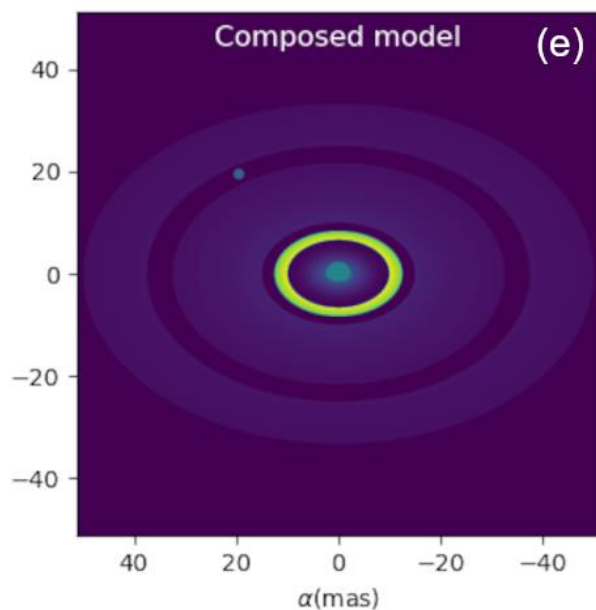
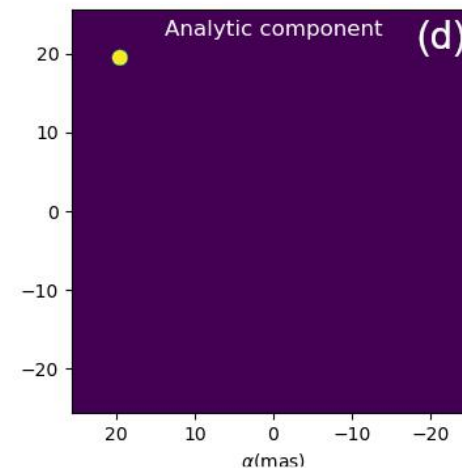
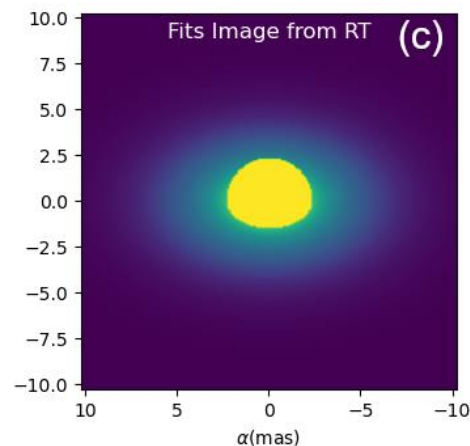
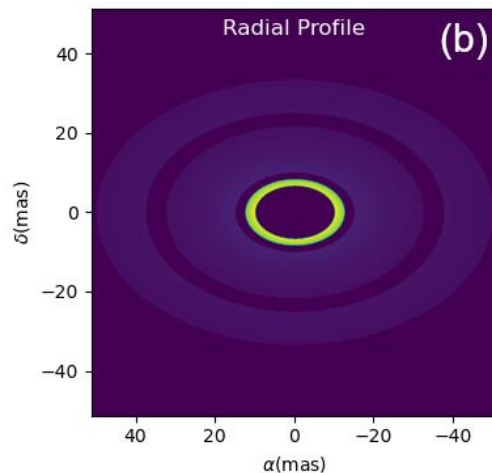
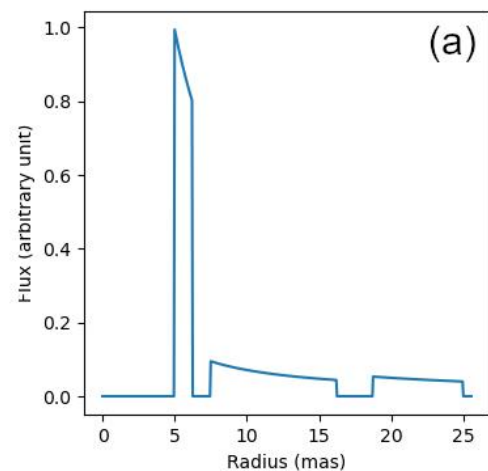


3 - Combining components to build complex models





3 - Combining components to build complex models





4 - Adding time-dependence or chromaticity to any parameter

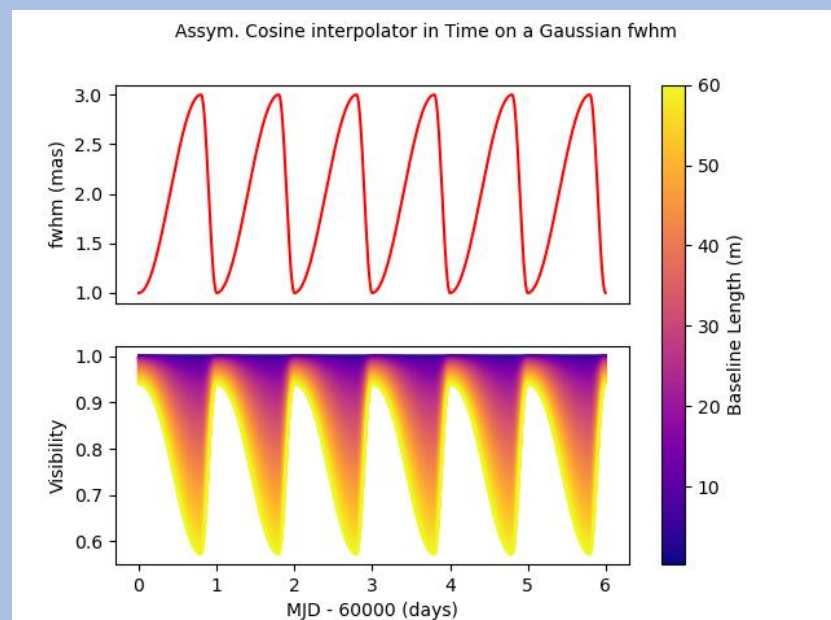
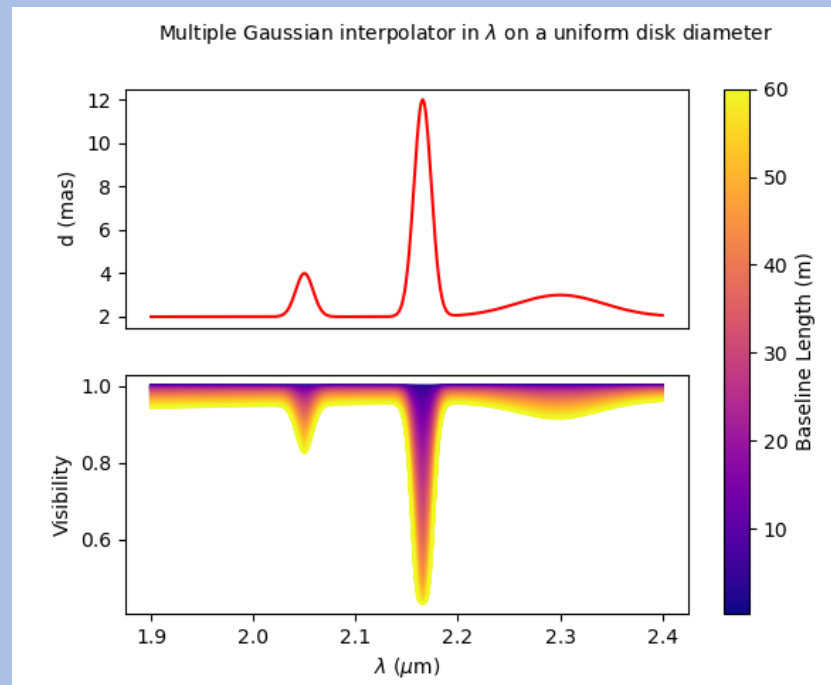
Parameter interpolators can be used to :

- simulate **chromatic changes to the intensity distribution**
- Simulate **time dependence** : pulsation, binarity ...

Many interpolators are available in oimodeler:

Class name	oimInterp macro	Description	Parameters
oimParamInterpolatorWl	"wl"	Interp between key wl	wl, values
oimParamInterpolatorTime	"time"	Interp between key time	mjd, values
oimParamGaussianWl	"GaussWl"	Gaussian in wl	val0, value, x0, fwhm
oimParamGaussianTime	"GaussTime"	Gaussian in time	val0, value, x0, fwhm
oimParamMultipleGaussianWl	"mGaussWl"	Multiple Gauss. in wl	val0 and value, x0, fwhm
oimParamMultipleGaussianTime	"mGaussTime"	Multiple Gauss. in time	val0 and value, x0, fwhm
oimParamCosineTime	"cosTime"	Asym. Cosine in Time	T0, P, values (optional x0)
oimParamPolynomialWl	"polyWl"	Polynomial in wl	coeffs
oimParamPolynomialTime	"polyTime"	Polynomial in time	coeffs

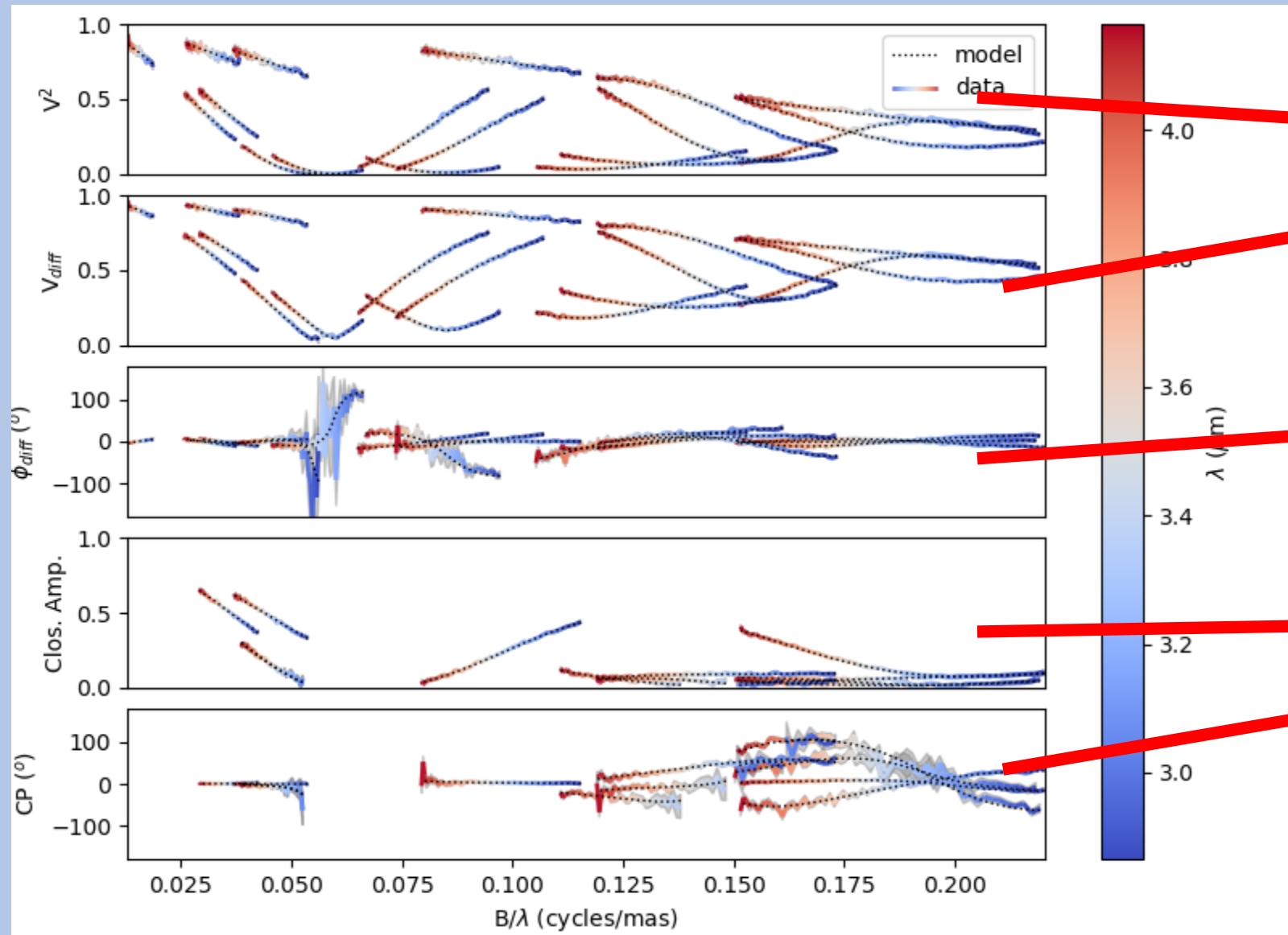
New interpolators can easily be **implemented by users**





5 - Compute synthetic OIFITS2 quantities

Taking data from any instrument and a model oimodeler can simulate any OIFITS2 quantities



VIS2DATA: square visibility

VISAMP:

- absolute visibility,
- differential visibility
- correlated flux

VISPHI:

- absolute phase
- differential phase

T3AMP: amplitude of the triple product

T3PHI: Closure phase

+ FLUXDATA:

- absolute flux
- normalized flux

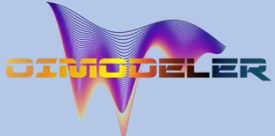


6 - Performing model-fitting using various methods

In the current version oimodeler includes 4 model-fitting algorithms:

- **MCMC sampler** based on the **emcee** package
- **Dynamic Nested (DN) sampler** based on the **dynesty** package
- **Levenberg-Marquardt (LM)** χ^2 minimizer
- **Regular grid** χ^2 explorer

Uncertainties can be estimated using the posterior probability function (MCMC or DN sampler) or the covariant matrix (LM)



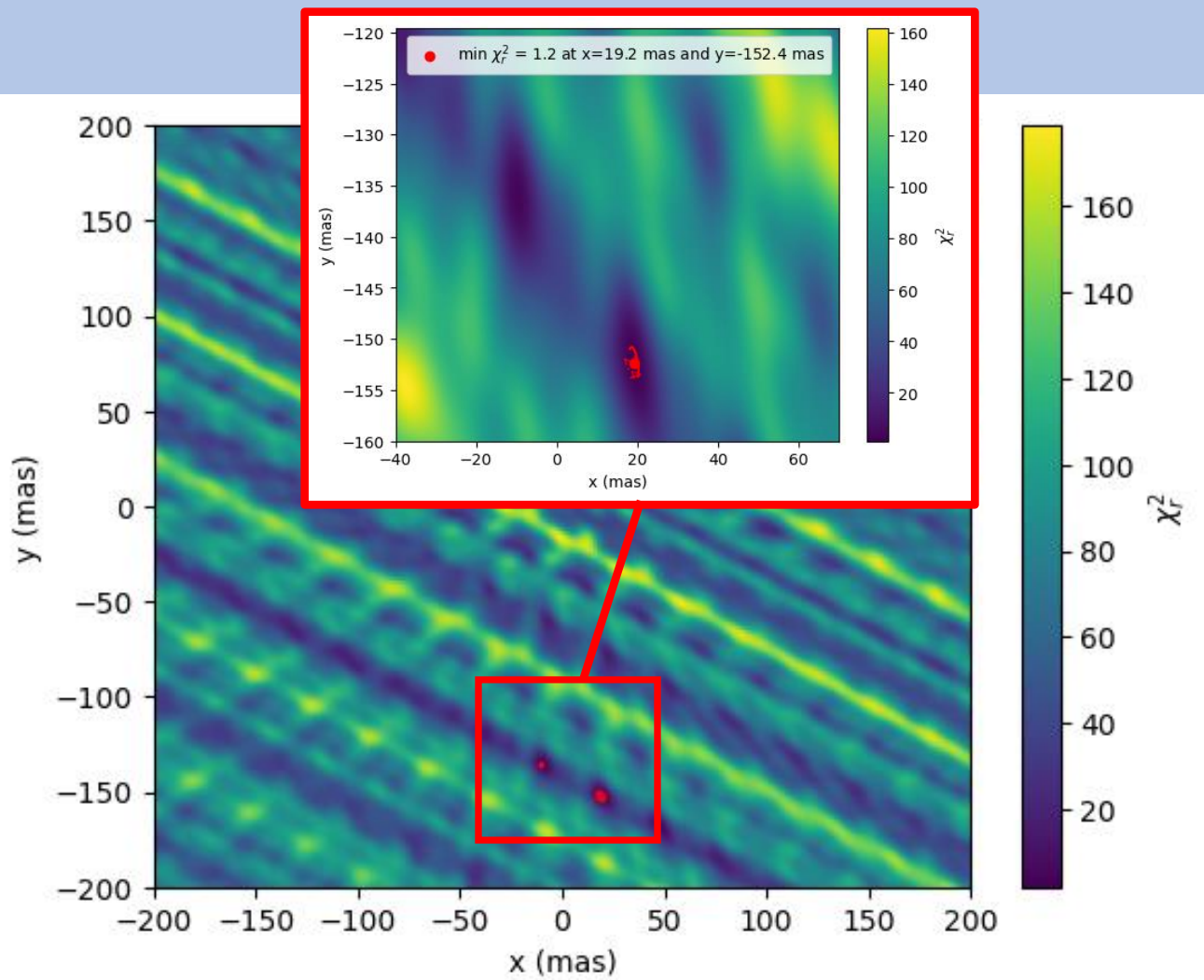
6 - Performing model-fitting using various methods

Some examples from the 2024 VLTI School

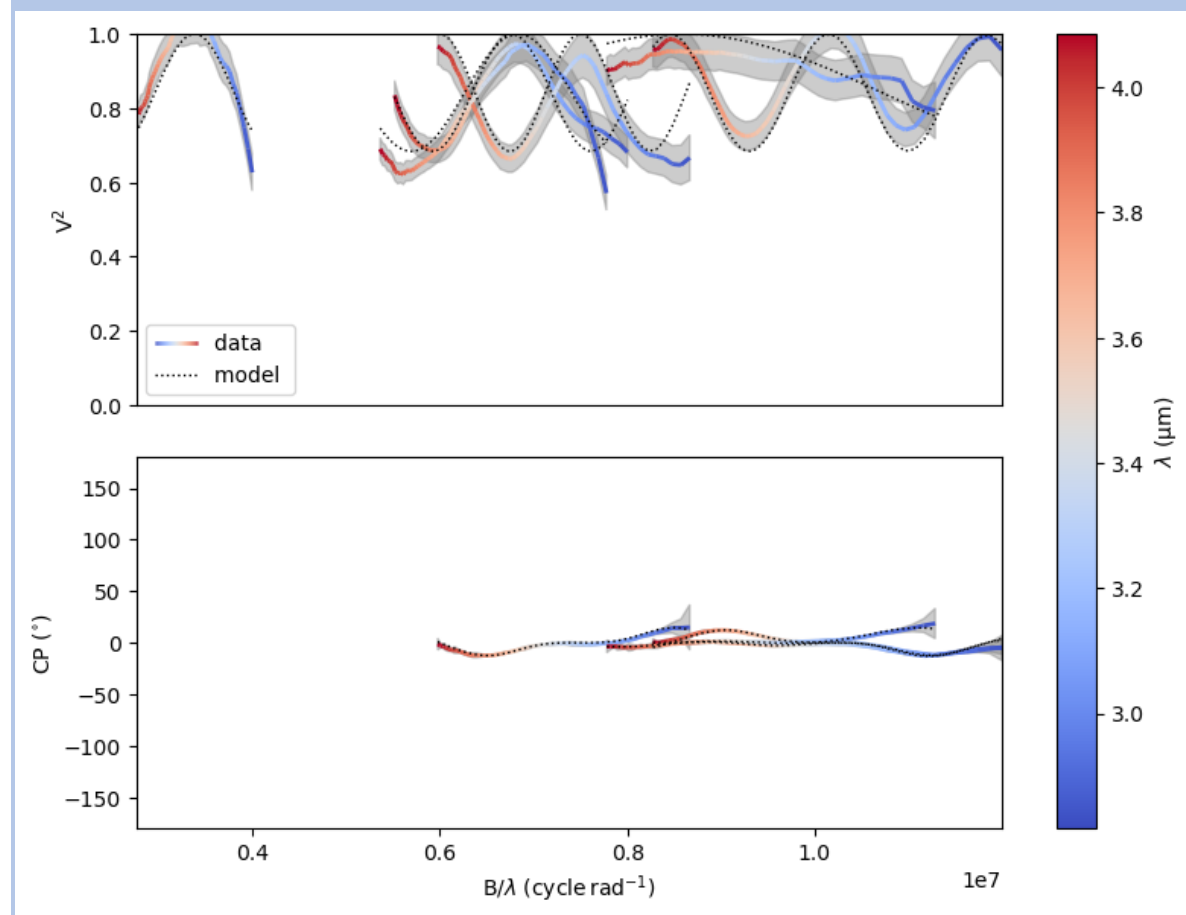
- the stellar surface of Canopus (PIONIER)
- the Binary star 94 Aqr (MATISSE-LOW)
- The T Tauri star HD 179278 (MATISSE-LOW)
- The Herbig star MWC 297 (MATISSE-HIGH)
- The Herbig star HD58647 (GRAVITY-HIGH)
- The B[e] star FS CMa (MATISSE-LOW)

6 - Performing model-fitting using various methods

Binary star 94 Aqr observed with the VLT/MATISSE

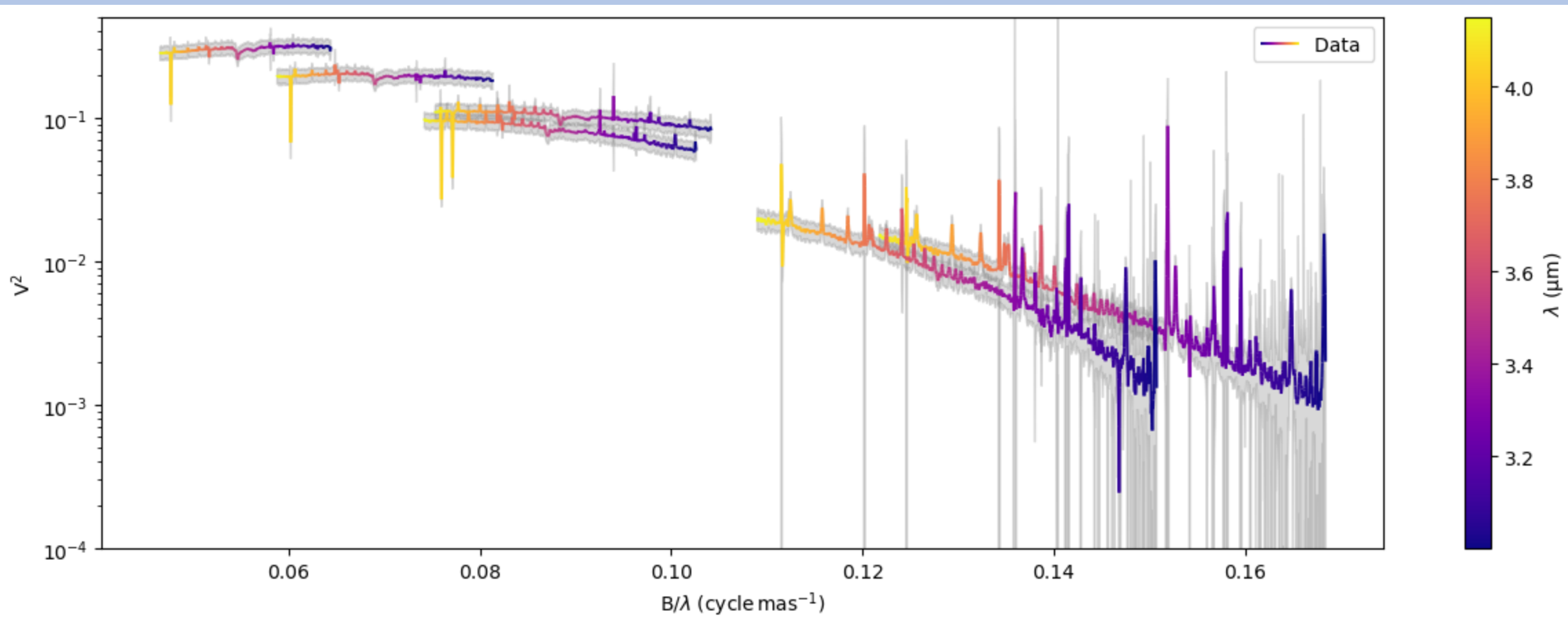


Grid exploration



6 - Performing model-fitting using various methods

Herbig Be star MWC297 observed with MATISSE



Modelling the Continuum + line emission

6 - Performing model-fitting using various methods

Herbig Be star MWC297 observed with MATISSE

```
wl0 = 4.0518e-6 # meter
```

```
star = oim.oimPt()
```

```
continuum = oim.oimEGauss()
```

```
line = oim.oimEGauss(f=oim.oimInterp('GaussW1',val0=0, value=100, x0=wl0, fwhm=5e-9))
```

```
line.params['pa']=oim.oimParamLinker(continuum.params['pa'])
```

```
line.params['elong']=oim.oimParamLinker(continuum.params['elong'])
```

```
model = oim.oimModel(star,continuum,line)
```

Building a chromatic model
with 3 components

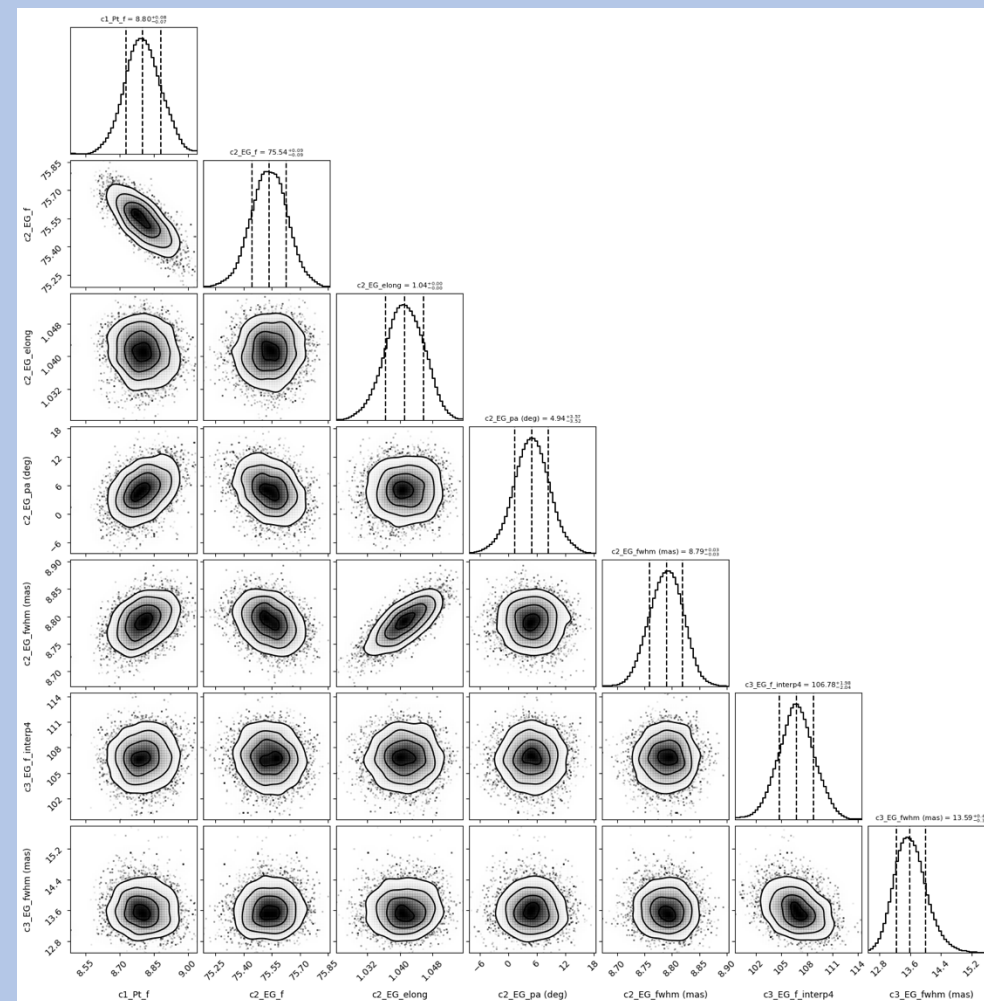
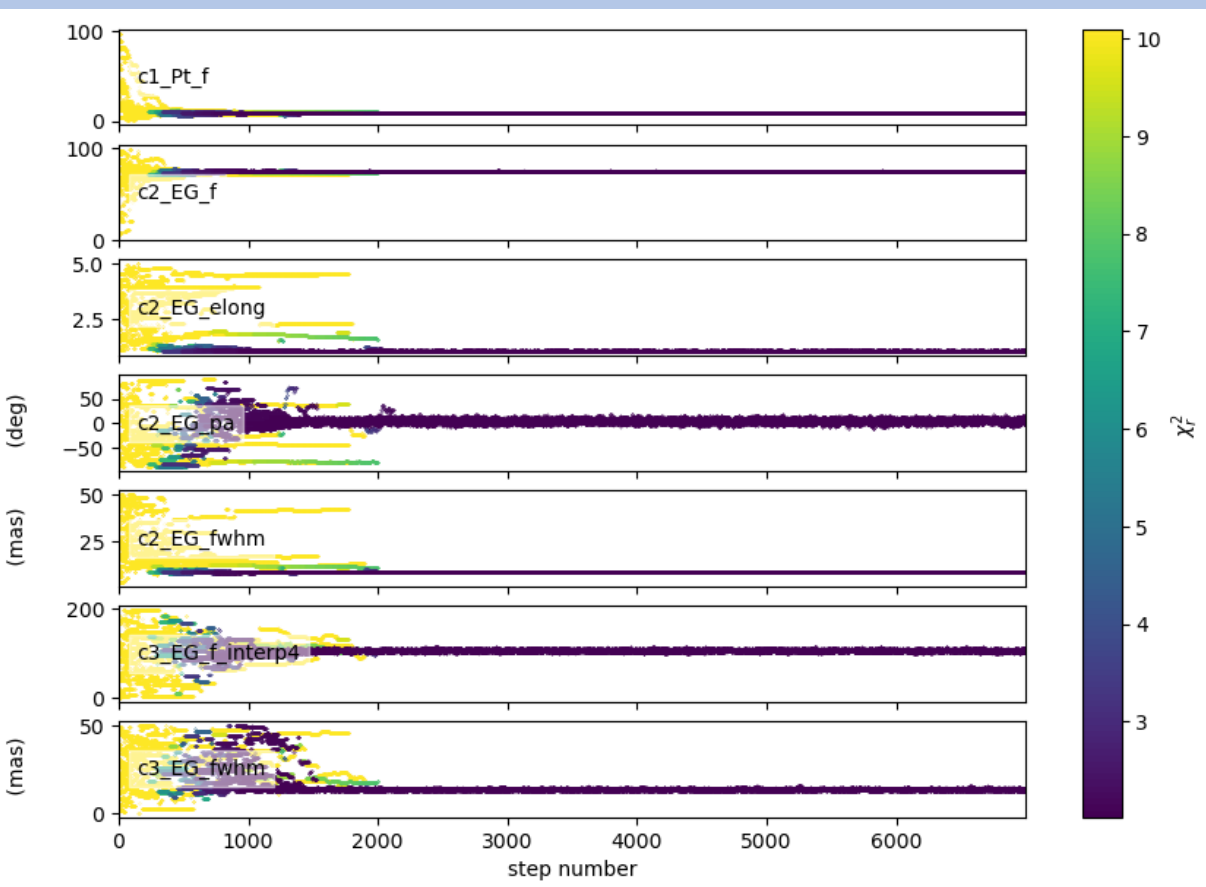
```
model.getFreeParameters()
```

7 Free parameters

```
{'c1_Pt_f': oimParam at 0x1d5e8c56fd0 : f=1 ± 0 range=[0,100] free=True ,
 'c2_EG_f': oimParam at 0x1d5e8c58990 : f=1 ± 0 range=[0,100] free=True ,
 'c2_EG_elong': oimParam at 0x1d5e8c363d0 : elong=1 ± 0 range=[1,5] free=True ,
 'c2_EG_pa': oimParam at 0x1d5e8c5ac50 : pa=0 ± 0 deg range=[-90,90] free=True ,
 'c2_EG_fwhm': oimParam at 0x1d5e8c5a090 : fwhm=0 ± 0 mas range=[0,50] free=True ,
 'c3_EG_f_interp4': oimParam at 0x1d5e782c5d0 : f=115 ± 0 range=[0,200] free=True ,
 'c3_EG_fwhm': oimParam at 0x1d5e8c53290 : fwhm=0 ± 0 mas range=[0,50] free=True }
```

6 - Performing model-fitting using various methods

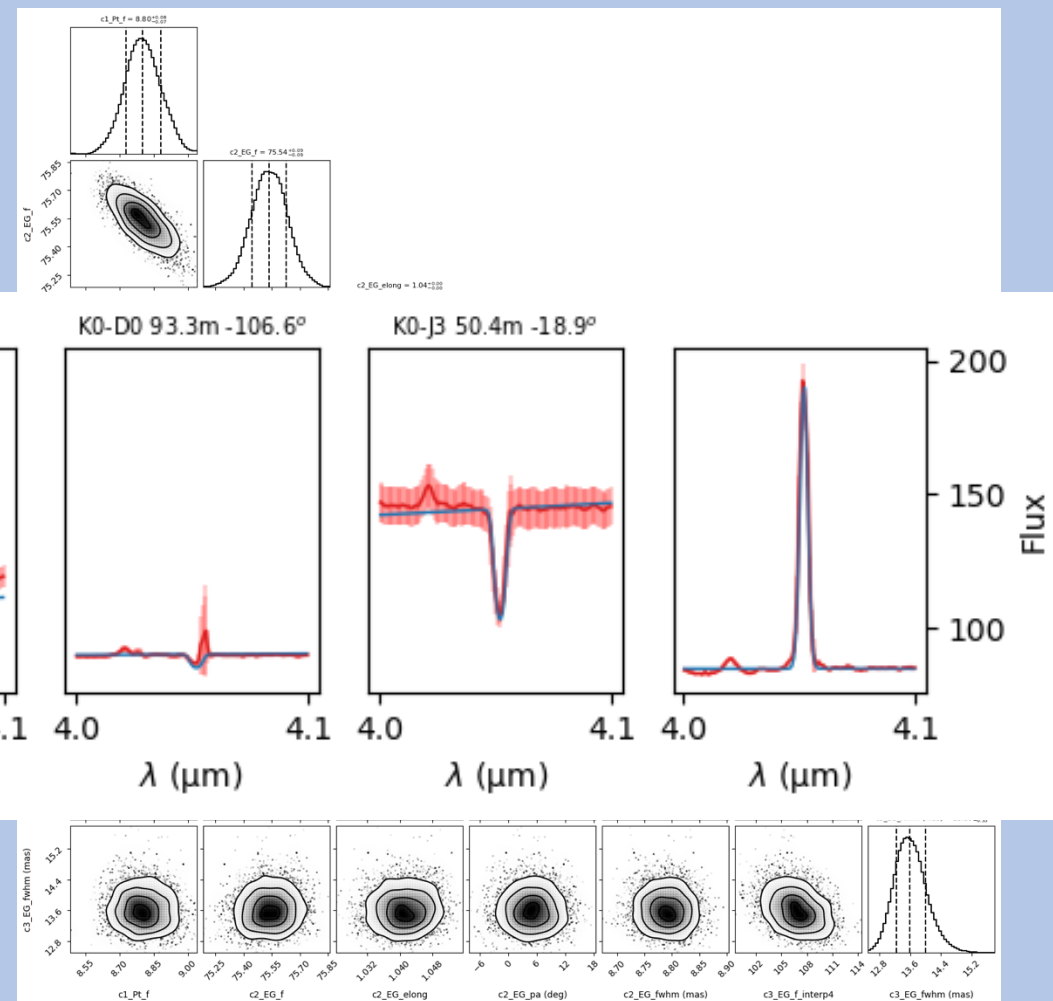
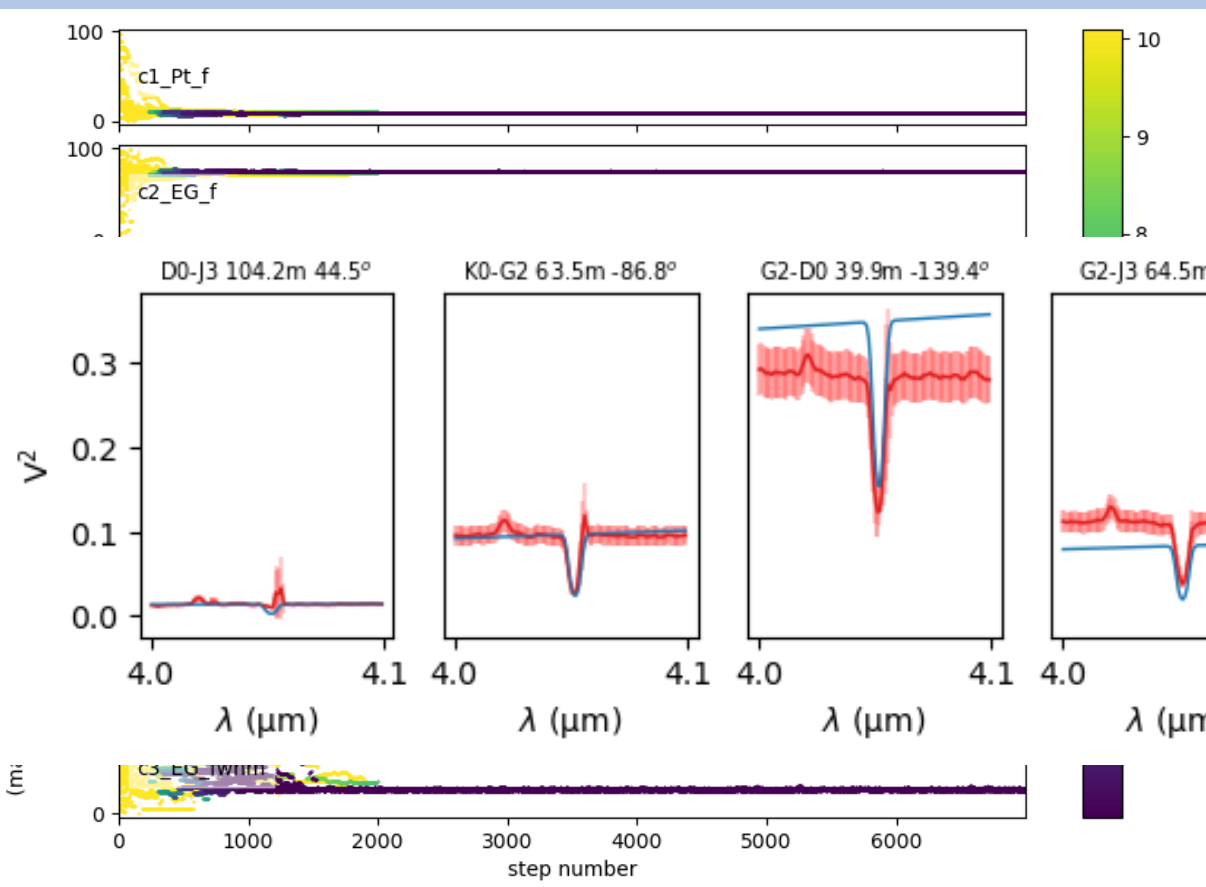
Herbig Be star MWC297 observed with MATISSE



MCMC sampling

6 - Performing model-fitting using various methods

Herbig Be star MWC297 observed with MATISSE

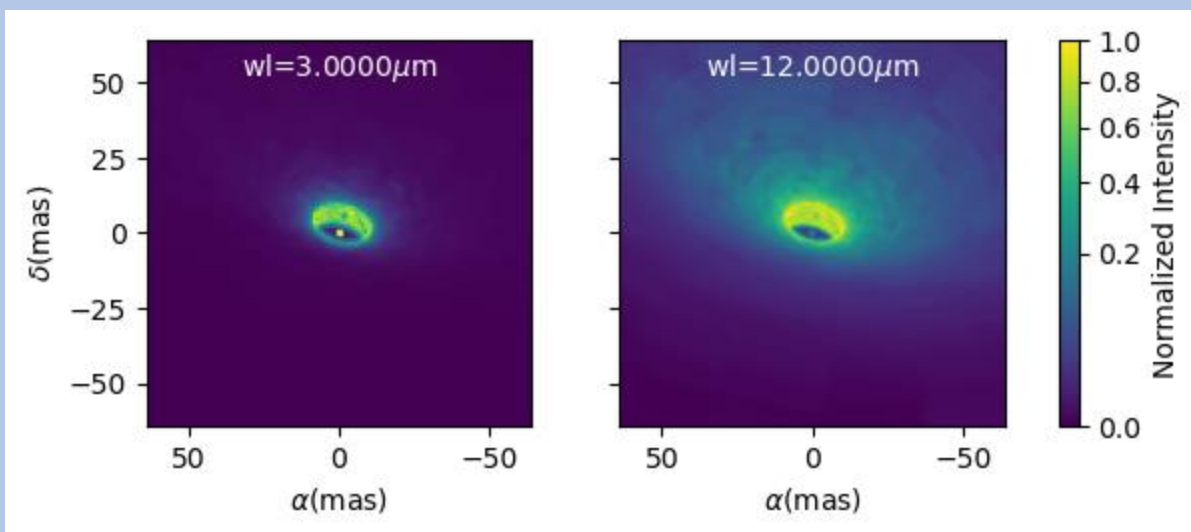
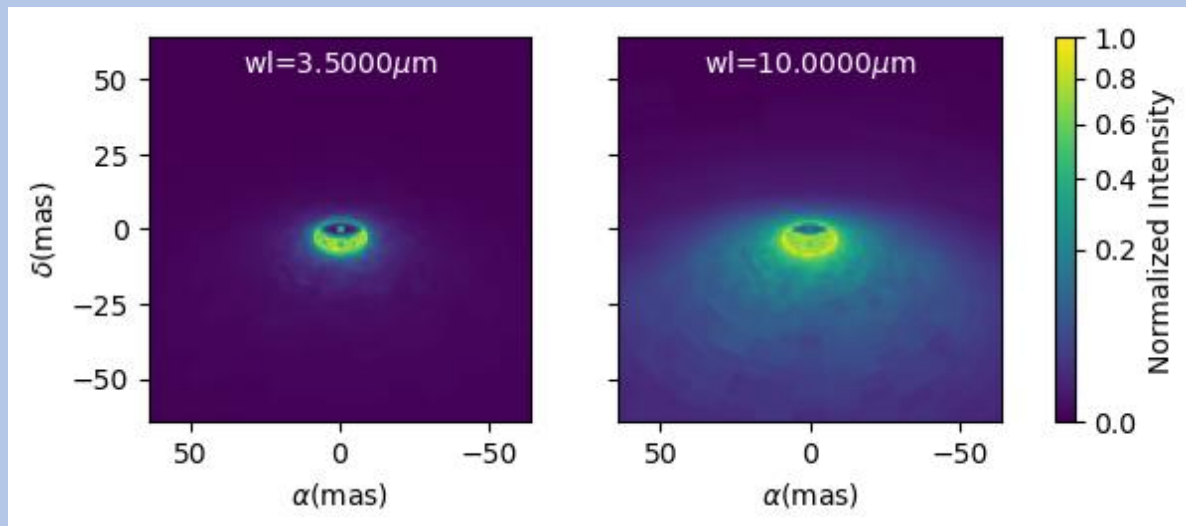


MCMC sampling

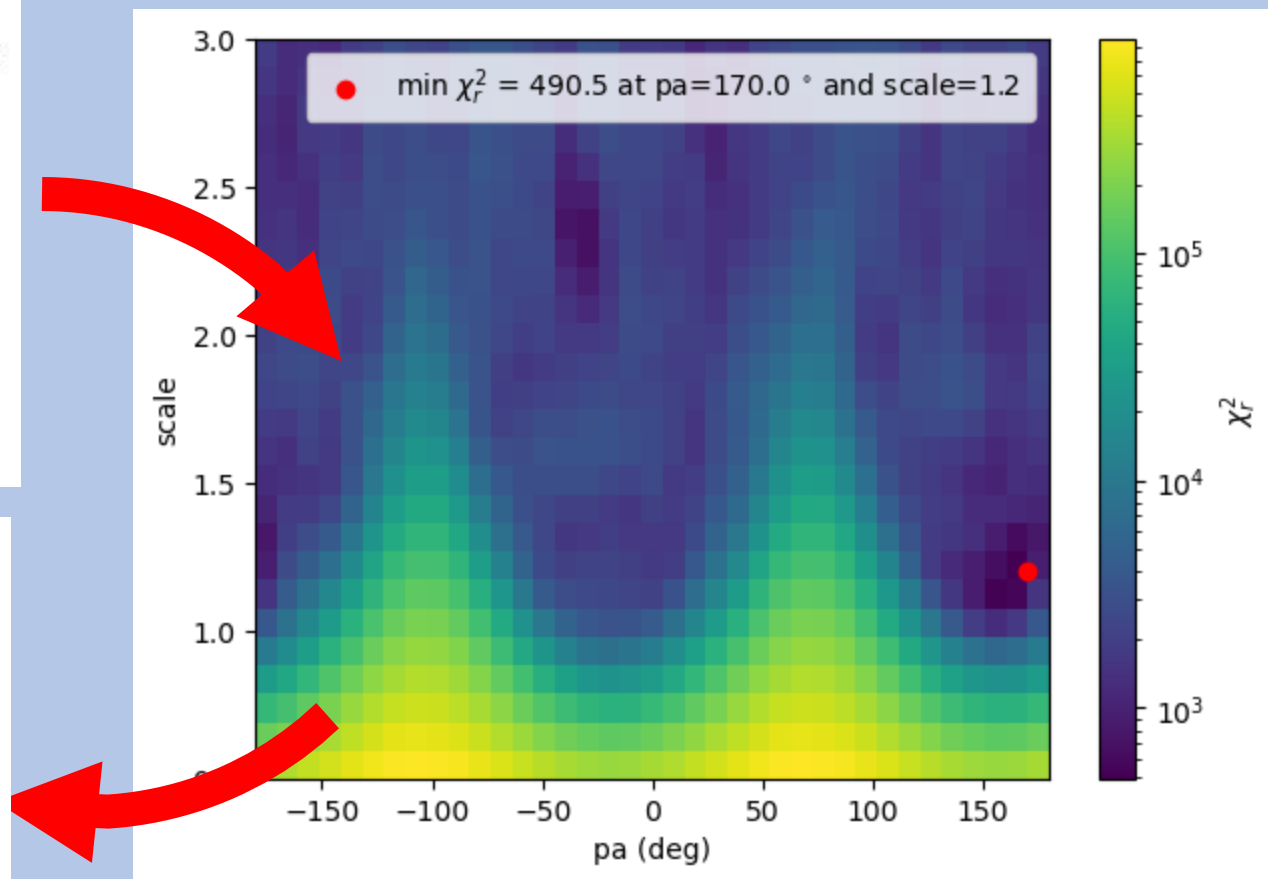
6 - Performing model-fitting using various methods

B[e] star FS CMa

RADMC3D RT model (from 2021 VLTI School)

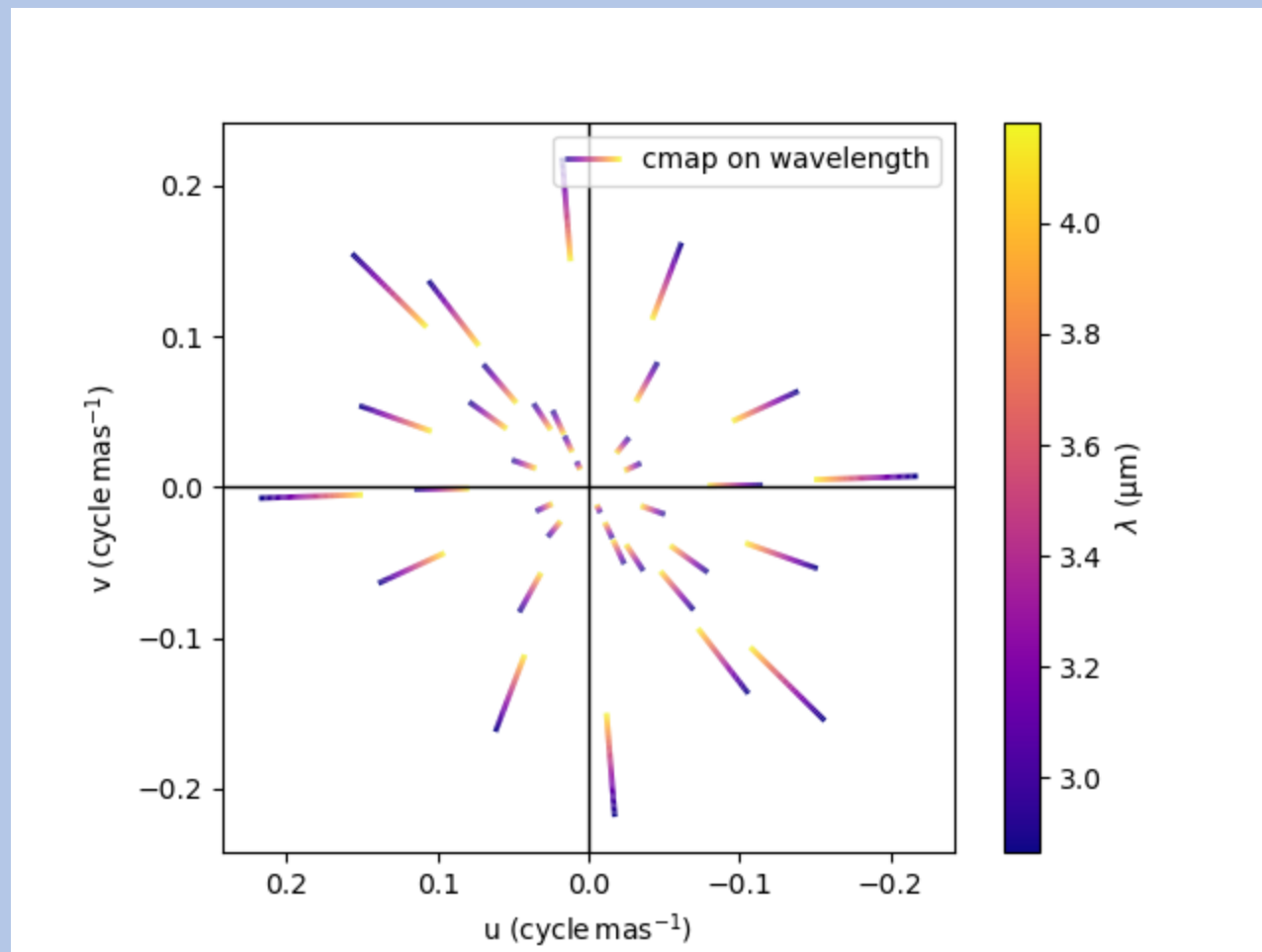
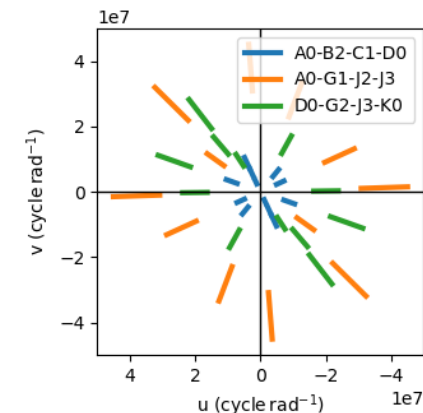
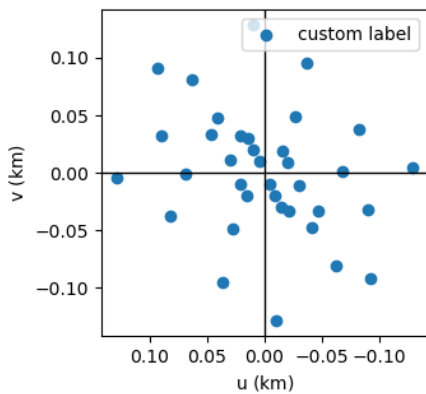
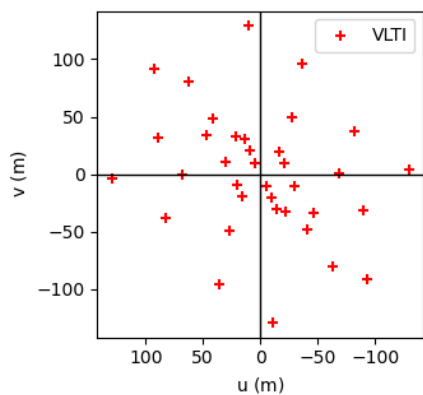
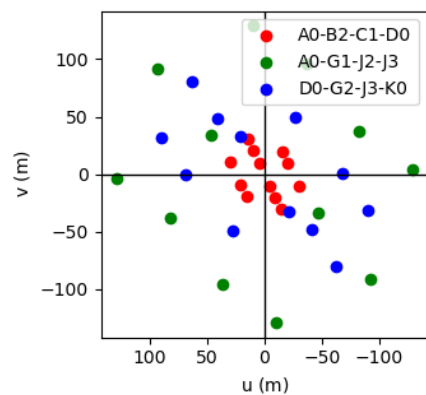
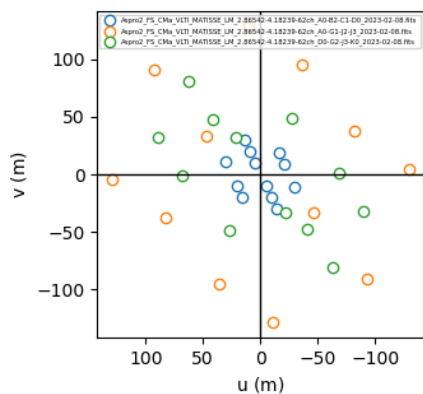
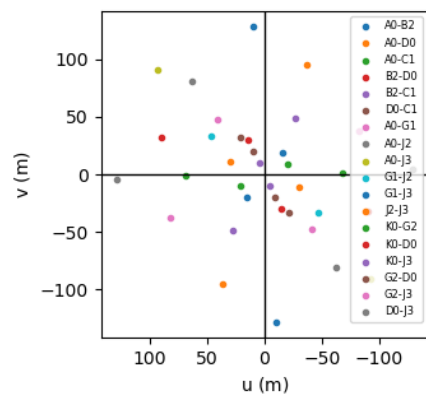


Best-fit (i.e., scaled and rotated) model

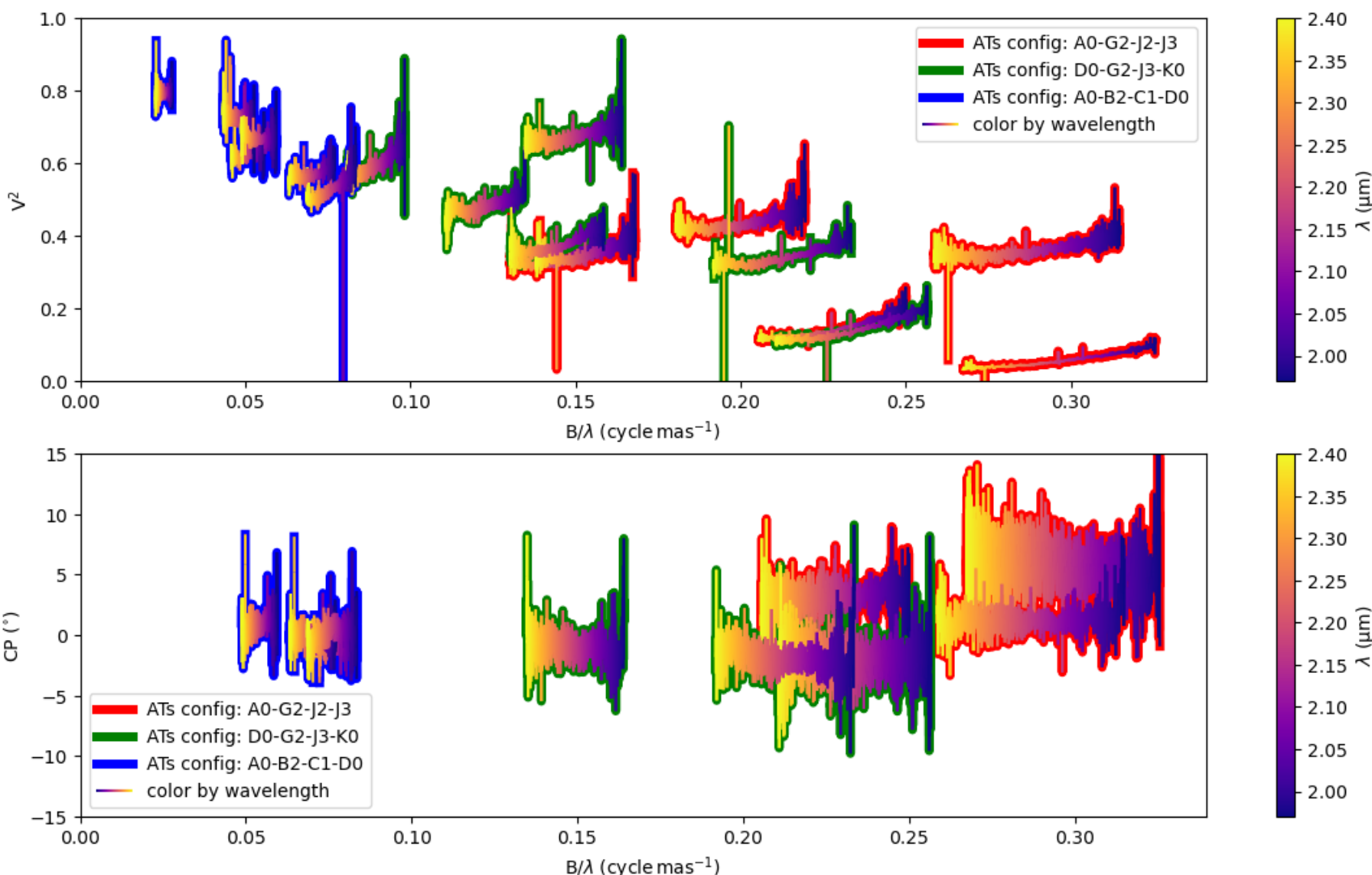


Grid exploration on major-axis PA and scaling factor

7 – Easily plot results and produce high quality figures

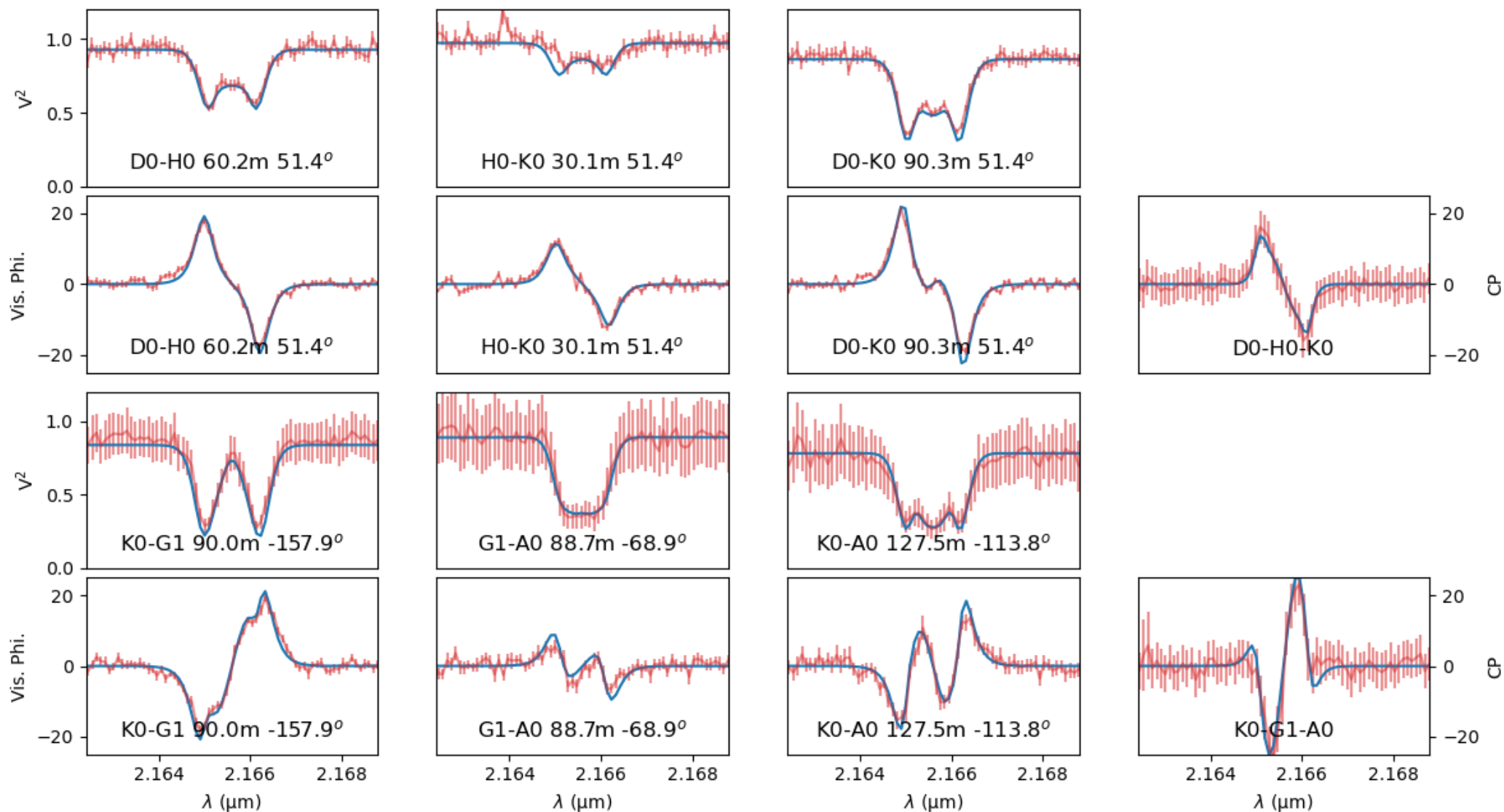


7 – Easily plot results and produce high quality figures



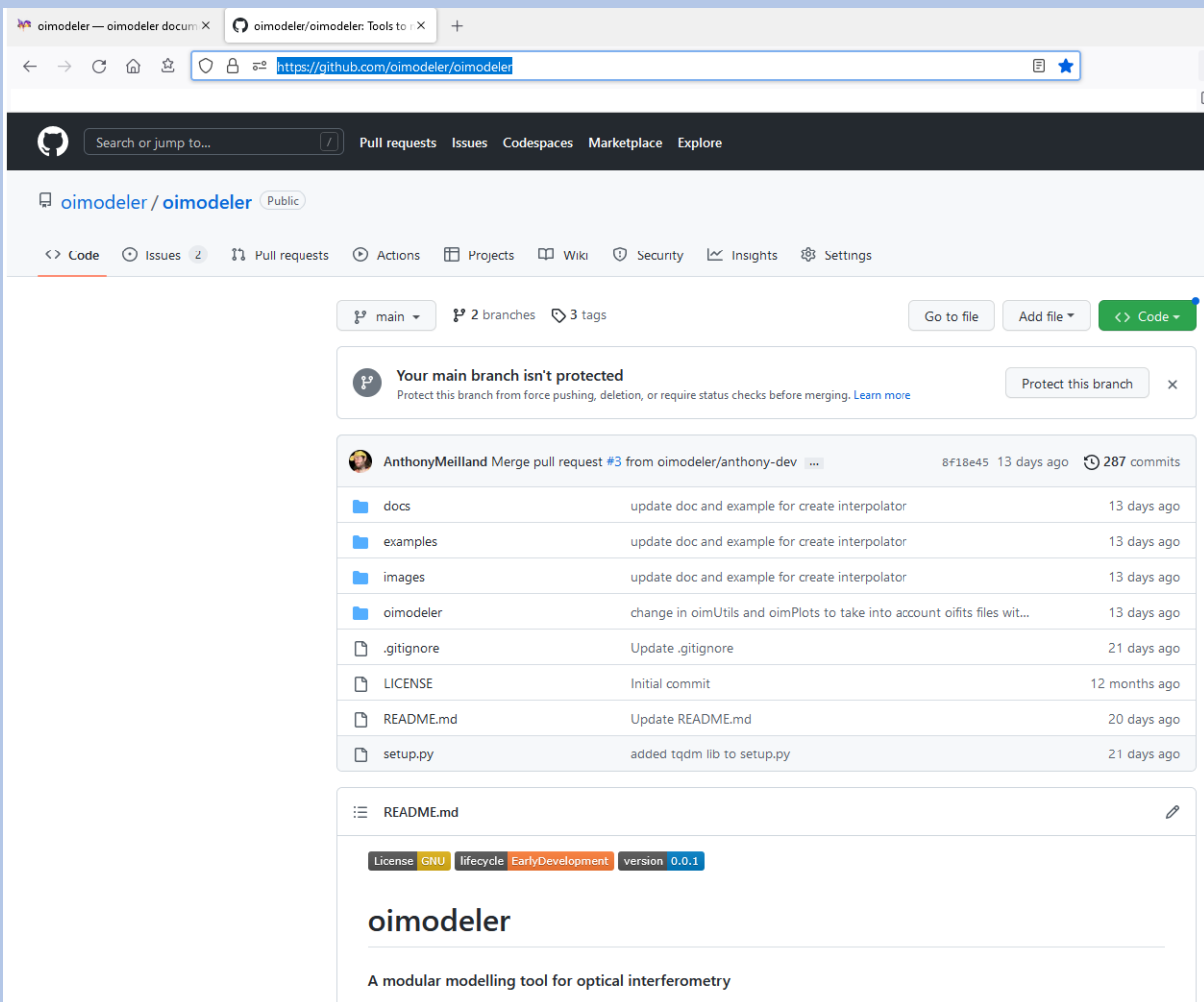
7 – Easily plot results and produce high quality figures

α Col AMBER data + Kinematic disk model: $\chi^2_r = 0.6$





8 - oimodeler on the web



The screenshot shows the GitHub repository for oimodeler. The repository is public and has 2 branches and 3 tags. The main branch is not protected. The repository contains a README.md file, a LICENSE file, a .gitignore file, and a setup.py file. The README.md file is displayed at the bottom of the page, showing the oimodeler logo and the text "A modular modelling tool for optical interferometry".

oimodeler / oimodeler (Public)

main 2 branches 3 tags

Your main branch isn't protected

AnthonyMeilland Merge pull request #3 from oimodeler/anthony-dev 8f18e45 13 days ago 287 commits

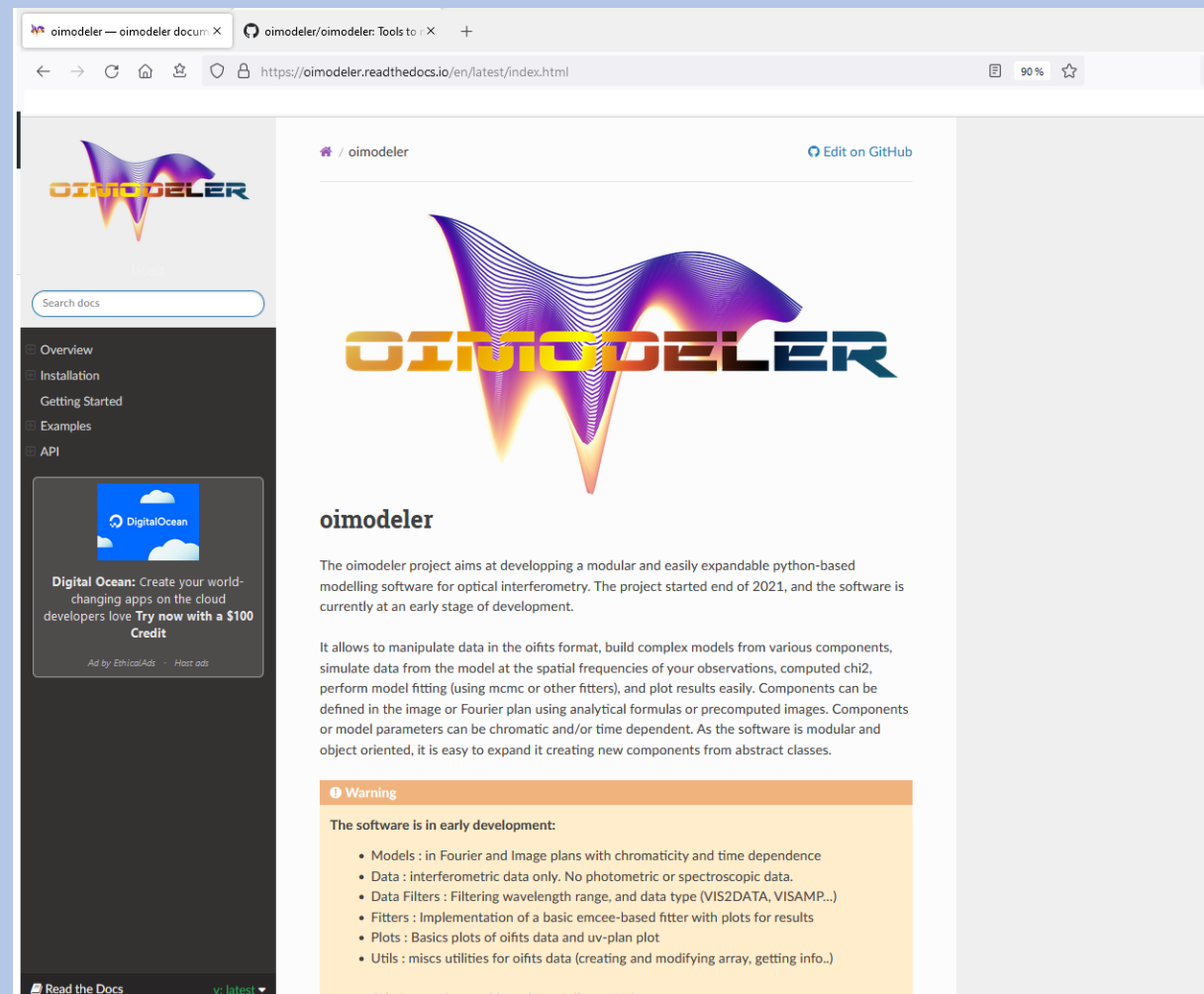
File	Commit	Time
docs	update doc and example for create interpolator	13 days ago
examples	update doc and example for create interpolator	13 days ago
images	update doc and example for create interpolator	13 days ago
oimodeler	change in oimUtils and oimPlots to take into account oifits files wit...	13 days ago
.gitignore	Update .gitignore	21 days ago
LICENSE	Initial commit	12 months ago
README.md	Update README.md	20 days ago
setup.py	added tqdm lib to setup.py	21 days ago

License GNU lifecycle EarlyDevelopment version 0.0.1

oimodeler

A modular modelling tool for optical interferometry

Code available on github
+ automatic installation through pip
<https://github.com/oimodeler>



The screenshot shows the oimodeler documentation page on readthedocs. The page features the oimodeler logo and a search bar. The left sidebar contains a navigation menu with links to Overview, Installation, Getting Started, Examples, and API. The main content area displays the oimodeler logo and a description of the project. A warning box indicates that the software is in early development. A Digital Ocean advertisement is also visible.

oimodeler

Search docs

Overview
Installation
Getting Started
Examples
API

oimodeler

The oimodeler project aims at developing a modular and easily expandable python-based modelling software for optical interferometry. The project started end of 2021, and the software is currently at an early stage of development.

It allows to manipulate data in the oifits format, build complex models from various components, simulate data from the model at the spatial frequencies of your observations, computed chi2, perform model fitting (using mcmc or other fitters), and plot results easily. Components can be defined in the image or Fourier plan using analytical formulas or precomputed images. Components or model parameters can be chromatic and/or time dependent. As the software is modular and object oriented, it is easy to expand it creating new components from abstract classes.

Warning

The software is in early development:

- Models : in Fourier and Image plans with chromaticity and time dependence
- Data : interferometric data only. No photometric or spectroscopic data.
- Data Filters : Filtering wavelength range, and data type (VIS2DATA, VISAMP...)
- Fitters : Implementation of a basic emcee-based fitter with plots for results
- Plots : Basics plots of oifits data and uv-plan plot
- Utils : miscs utilities for oifits data (creating and modifying array, getting info..)

Read the Docs v: latest

Documentation + examples available on readthedocs
<https://oimodeler.readthedocs.io>



Conclusion & Perspective

Code is fully functional, and core development ended in 2023

New features added in 2024:

- Data: Importing photometric/spectroscopic data
- Components: temperature gradient, pseudo-Lorentzian, LDDs
- Fitters: grid explorer & LM minimization
- Filters: expression-based flagging on many criteria: length, PA, SNR...
- Other: new template plots

Presentation of the code

- at SPIE 2024 (Poster + Paper)
- at the VLT School 2024 (Tutorials on MATISSE, PIONIER and GRAVITY data)

Multiple papers using oimodeler in preparation:

- MATISSE: B[e] I Pup (Abello et al.), MATISSE B[e] Survey (Abello et al.)
- Binaries with CHARA SPICA + MIRCX: (Jonak et al)
- Binary Be star Del Cen Disk and orbit: AMBER + MATISSE + PIONIER (Houlbreque et al.)
- Disk of the YSO HD 100453 (van Haastere et al.)
- Disk around the young spectroscopic binary DX Cha (Juhász et al.) ...