



**HAL**  
open science

## Oimodeler: a modular modelling software for optical interferometry

Anthony Meilland, Marten Scheuck, Jozsef Varga, Alexis Matter, Florentin Millour

► **To cite this version:**

Anthony Meilland, Marten Scheuck, Jozsef Varga, Alexis Matter, Florentin Millour. Oimodeler: a modular modelling software for optical interferometry. *Optical and Infrared Interferometry and Imaging IX*, Jun 2024, Yokohama, Japan. pp.104, 10.1117/12.3019762 . hal-04797236

**HAL Id: hal-04797236**

**<https://cnrs.hal.science/hal-04797236v1>**

Submitted on 22 Nov 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

# OIModeler

## A modular modelling software for optical interferometry

Anthony Meilland<sup>a</sup>, Marten Scheuck<sup>b</sup>, Jozsef Varga<sup>c</sup>, Alexis Matter<sup>a</sup>, and Florentin Millour<sup>a</sup>

<sup>a</sup>Laboratoire Lagrange, Université Côte d’Azur – CNRS, France

<sup>b</sup>Max-Planck-Institute of Astronomy (MPIA), Heidelberg, Germany

<sup>c</sup>HUN-REN Research Centre for Astronomy and Earth Sciences, Konkoly Observatory,  
Hungary

### ABSTRACT

It has been more than 20 years since optical-interferometric facilities first started producing spectro-interferometric data, and yet the number of general user tools able to handle this type of data properly is limited. Moreover, they often either focus on simple generic cases (e.g., simple grey geometrical models defined with analytical functions) or very specific objects (e.g., binaries, kinematics in emissions lines of circumstellar disks, stellar photospheres). To overcome these limitations, we have developed `oimodeler`, a python-based optical-interferometric data modelling software allowing one to build complex models from various type of components ranging from simple geometrical to external outputs of radiative transfer models. The software allows one to add chromaticity (continuum, line and bands) and time-dependence easily. The code is object-oriented to be more flexible and easily expandable. Here we present its concept, implementation and some examples of fitting of real data. The code is available at <https://github.com/oimodeler/oimodeler>.

**Keywords:** optical interferometry, spectro-interferometry, software, model-fitting

### 1. THE OIMODELER PROJECT

The `oimodeler` project aims at developing a modular and easily expandable python-based package for optical interferometry. It allows anyone to manipulate all kind of data in the OIFITS format, build complex models from various components, simulate data from the model at the spatial frequencies of the observations, compute  $\chi_r^2$ , perform model fitting using various methods, infer best parameters and uncertainties, and plot results easily.

Components can be defined in the image or Fourier plane using analytical formulas or precomputed images. Components or model parameters can be chromatic to give room for source spectra separation (like in HR5171<sup>1</sup>) and/or time dependent (to give room for e.g. orbit fitting<sup>2</sup>). The software is modular and object oriented, to make it easier to expand it by implementing new model components or other features from semi-abstract classes.

The project started end of 2021, and the software is now in a fully functional beta version. The development of the `oimodeler` package builds on knowledge acquired in model-fitting of interferometric data over the past two decades and the development of the previous generation of modelling tools such as LITPro<sup>3</sup> and FitOmatic.<sup>4</sup>

### 2. COMPONENTS AVAILABLE IN OIMODELER

In the `oimodeler` framework, components are the building blocks of models. They all derive from the abstract `oimComponent` class. Components can be defined analytically in the Fourier plane or image plane. Some of the components are shown in Fig 1. All components have parameters (deriving from the `oimParam` class) that can be free or fixed when performing model-fitting. The `oimComponent` can return complex-coherent flux (the base quantity for all interferometric observables) giving a vector of uv-coordinates and optionally wavelength and time coordinates for chromatic and time-dependent components. It can also return the component image itself regardless of it being defined in the Fourier or image planes.

Most simple components are defined analytically in the Fourier plane for model fitting optimisation and are derived from the `oimComponentFourier` semi-abstract class. This class implements some basic features such as translation, rotation and ellipticity. Simple Fourier-based components include point source, background, uniform

and limb darkened disks (linear, quadratic, power, square-roots), Gaussian distribution, ring (infinitesimal and uniform in width), skewed ring, pseudo-Lorentzian (see Fig 1a)

Components defined in the image plane derive from the semi-abstract class `oimComponentImage` which implements basic features for computing interferometric data from images. However, the Fourier transform is performed by external backends that can be specified by the user. Current backends are Fast-Fourier transform (including zero-padding) perform either by `numpy.fft` module or by the `pyFFTW` package, if available. Users can easily create new FT-backends and foreseen extension are, for instance, discrete Fourier transform (DFT).

The `oimComponentImage` can be used to implement simple components for which no analytical formula exist in the Fourier plane. An example is the `oimSpiral` component as shown in Fig 1c (left image) or the `oimKinematicDisk` based on a toy-model for classical Be stars.<sup>5</sup> The `oimComponent` can also be used to wrap any external code as demonstrated by the wrapping of a flattened and gravitationally darkened fast-rotator shown in Fig 1c (right image).

The `oimodeler` package also contains a specific class, `oimComponentFitsImage`, to import precomputed images or hyperspectral image-cubes such as outputs from radiative-transfer models in a fits-image format (see Fig 1b). The loaded image can be shifted, rotated and scaled if necessary and will behave as a component with five parameters: positions  $x$  and  $y$ , flux ( $f$ ), position angle ( $pa$ ) and scale.

Finally, the `oimComponentRadial` can be used to build 1D-radial profile components. Interferometric quantities are then computed using the Hankel transform. The radial-profile can also be used to build flattened components as shown in Fig 1e.

Planned extensions for 2024 include, for instance, grids of models with interpolation, and a binary component with orbital elements.

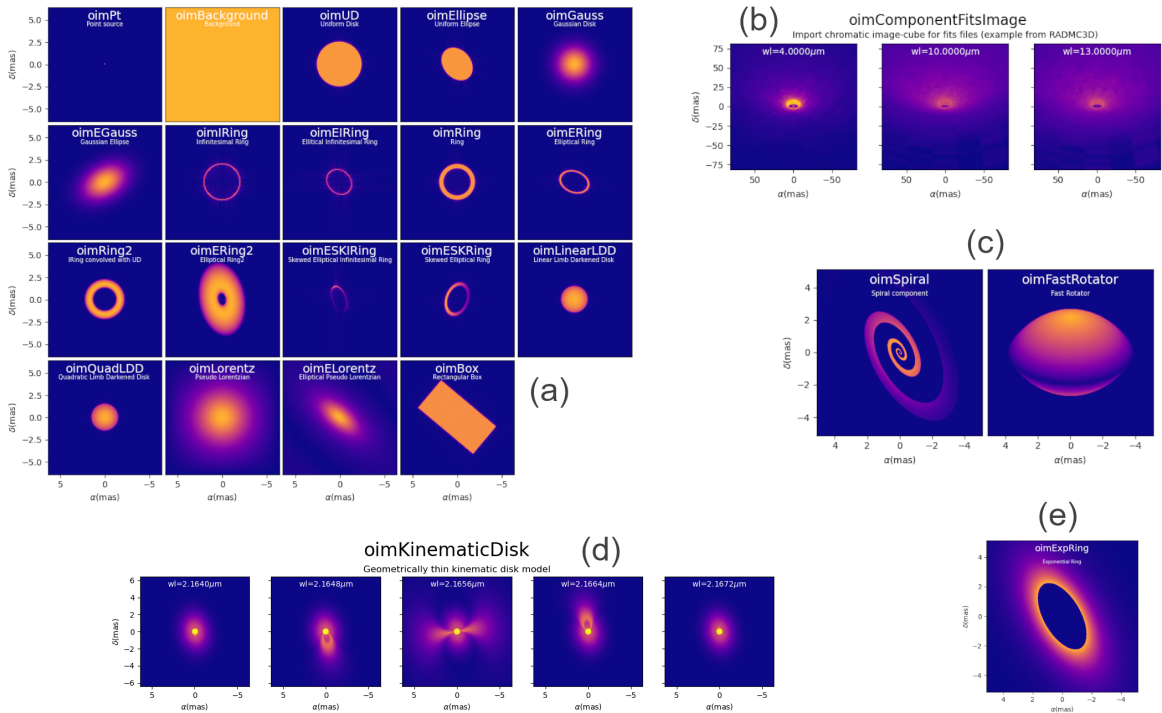


Figure 1. Different types of components available in `oimodeler`: (a) simple geometric ones defined in Fourier space, (b) image or image cubes imported in fits image format from radiative transfer codes, (c) analytical models defined in direct 2D space, (d) kinematic model through lines such as a rotating disk seen in an emission line, (e) radial intensity profiles defined analytically or outputs from RT codes.

### 3. COMBINING COMPONENTS TO BUILD COMPLEX MODELS

The basic components presented in the previous section can be combined to build complex models regardless of the nature of the components. This is done by the `oimModel` class. In the example shown in Fig 2, we combine a fits-image from the semi-physical model DISCO\*, a uniform disk defined in the Fourier plane and an ad-hoc radial profile with an exponential law with gaps, and simulate its visibility function. The interferometric observables for these components are computed using:

- the analytic formula in the Fourier plane for the the uniform disk
- the Fast-Fourier-Transform for the DISCO image
- Hankel transform for the radial profile

and then combined with their proper flux ratio and translation to obtain the final result in a transparent way for the user.

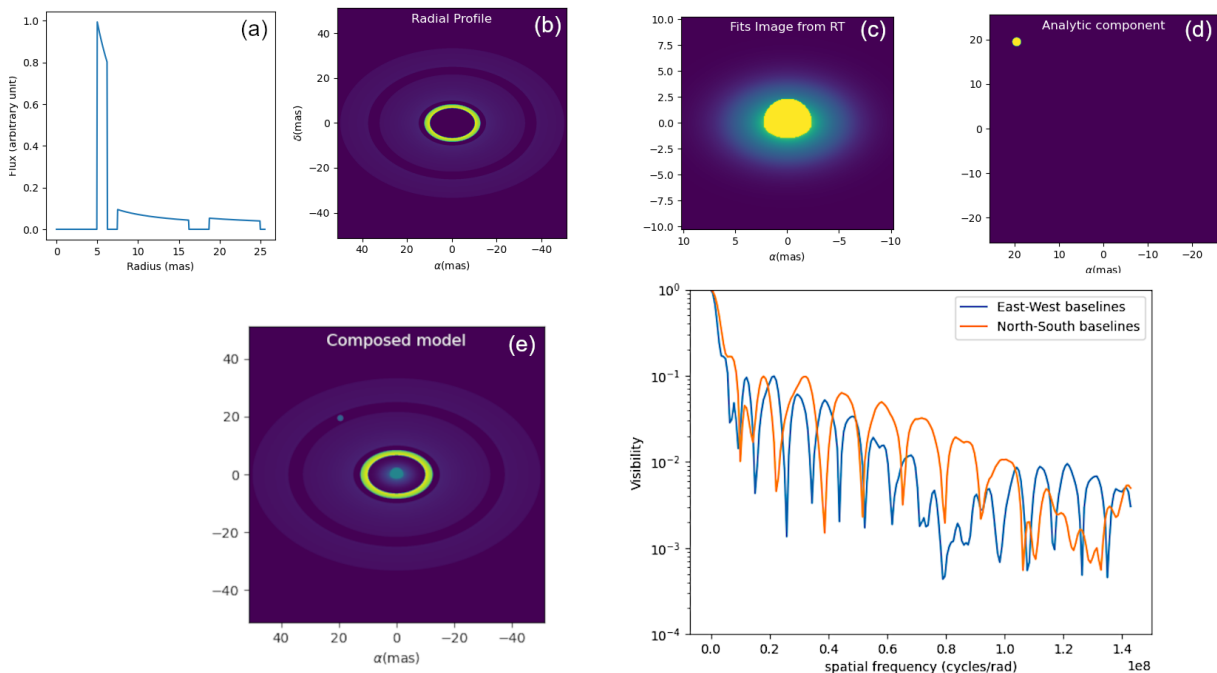


Figure 2. Building complex models from simple components of different nature. Here, we used a radial profile with gaps (a and b), the output in fits-image format from the DISCO semi-physical model (c), and a simple shifted uniform disk defined in the Fourier plane (d) to build a complex model of a star with gaseous and dusty disk with gaps and a companion in one of the gaps (e). `oimodeler` can easily compute interferometric data such as square visibility (f) or closure-phase from this model and perform model fitting on the free-parameters.

### 4. ADDING TIME-DEPENDENCE OR CHROMATICITY TO ANY PARAMETER

Beyond building multi-components models, `oimodeler` provides another way to add complexity to the models using the so called parameter interpolators. They allow to replace any component parameter by a function that depends either on the wavelength, i.e. to add chromaticity to the model, or time, to create time dependent models.

\*based on<sup>6</sup> and available at the [AMHRA website](#)

The parameter interpolator replaces the single model parameter by a series of parameters that define the chromatic and/or time behaviour and can be used as free or fixed parameters for model-fitting: for instance, values at key wavelengths for a linear interpolator in wavelength or, period, epoch, asymmetry, amplitude for an asymmetric cosine time-interpolator.

Many chromatic interpolators are already implemented in `oimodeler`. Some are suited to simulate a slow variation of some parameter through a band such as the linear or quadratic interpolator with key wavelengths, or the polynomial interpolator. Others are more suited to emulate the spectral profiles of atomic or molecular lines or bands such as the Gaussian or multi-Gaussian interpolators as shown in Fig 3 (left).

Time interpolators can be used to model stellar pulsation like the asymmetric cosine time interpolator shown in Fig 3 (right), or expanding envelope (for instance with a linear, quadratic, or polynomial interpolators).

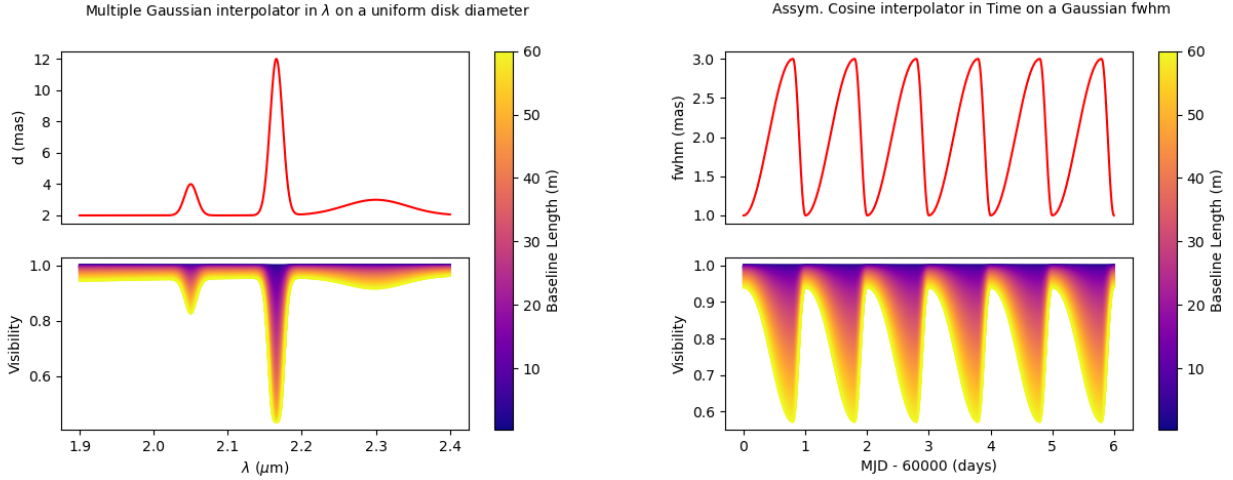


Figure 3. The chromatic multi-Gaussian-lines (left) and the asymmetric-cosine-time (right) interpolators used on a uniform disk diameter (left) and on a Gaussian FWHM (right) Bottom: corresponding visibility variations for 0 to 60m baselines.

## 5. COMPUTING SYNTHETIC OIFITS2 QUANTITIES

The `oimSimulator` class allows to import OIFITS2<sup>7</sup> data from any interferometric instrument as well as external flux measurements and spectra in ascii format (converted into a OIFITS2 FLUXDATA table). Using these data and an `oimModel` instance as described in the previous sections, `oimodeler` can simulate all type of interferometric quantities in the OIFITS2 format:

- VIS2DATA: squared visibility
- VISAMP: absolute visibility, differential visibility or correlated flux
- VISPHI: absolute phase or differential phase
- T3AMP: amplitude of the triple product
- T3PHI: closure phase
- FLUXDATA: absolute flux, or normalized flux

Finally, `oimSimulator` can also compute  $\chi_r^2$  between the data and the model and produce comparison plots (see Fig 4).

Note that, even without the use of OIFITS2 data through the `oimSimulator` class, an `oimModel` class can directly return complex coherent fluxes for a given vector of spatial coordinates (and optionally spectral and time coordinates) from which all the other interferometric observables can be derived.

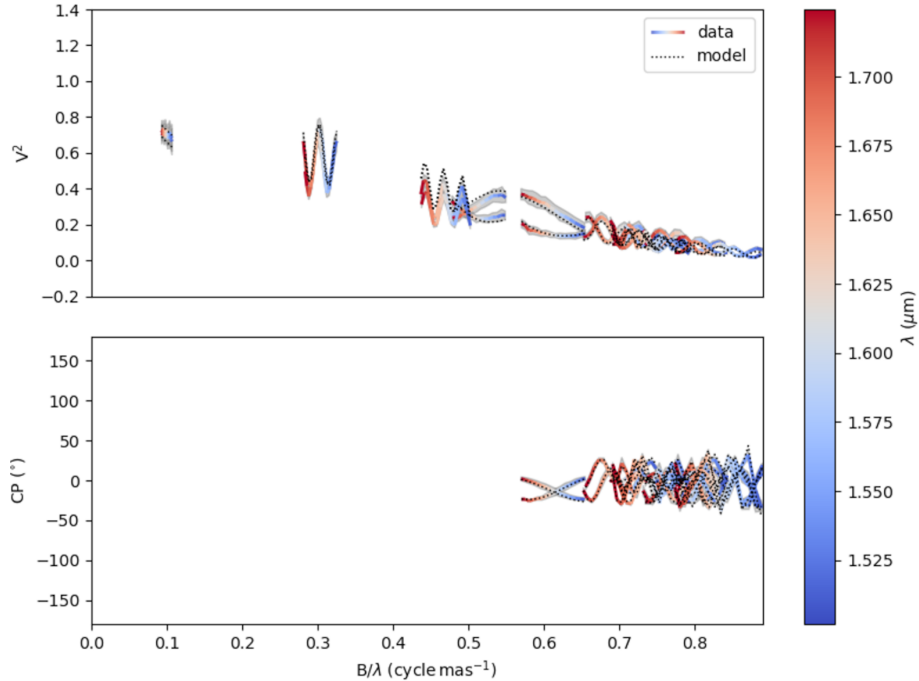


Figure 4. CHARA/MIRCX squared visibility (top) and closure phase (bottom) obtained on the binary star  $\beta$  Ari compared to a simple binary model with two uniform disks. The plot was generated by the `oimSimulator` class using the best parameters determined as described in the next section.

## 6. PERFORMING MODEL-FITTING USING VARIOUS METHODS

In the `oimodeler` framework, model-fitting is performed by a class deriving from the abstract class `oimFitter`. This class encapsulates an `oimSimulator` instance, which will determine the  $\chi^2$  value between the data and the model;  $\chi^2$  value that will be used in the minimisation process. In the current version, `oimodeler` includes four model-fitting algorithms:

- a MCMC sampler based on the `emcee`<sup>8</sup> python module
- a dynamic nested sampler based on the `dynesty`<sup>9</sup> python module
- a simple Levenberg-Marquardt  $\chi^2$  minimizer
- a regular grid with  $\chi^2_r$  explorer

These various algorithms allow the user to find the best-fit values of all free parameters of the model (minimum of  $\chi^2$ ) and, depending on their nature, to evaluate their statistic. For instance, uncertainties can be estimated using the posterior probability function in the case of MCMC or dynamic nested samplers, or using the covariant matrix for the Levenberg-Marquardt one.

In the example presented in Fig 5, we performed an MCMC-run on CHARA/MIRCX data using the `oimFitterEmcee` class, which wraps specifically the `emcee` sampler. The fitter class provides methods to set the parameters space up, the initial values of the parameters for the walkers, to estimate the posterior probability function and then the uncertainties on the parameters, to plot easily the results (walkers, corner, data/model comparison) and produce quality figures for publication.

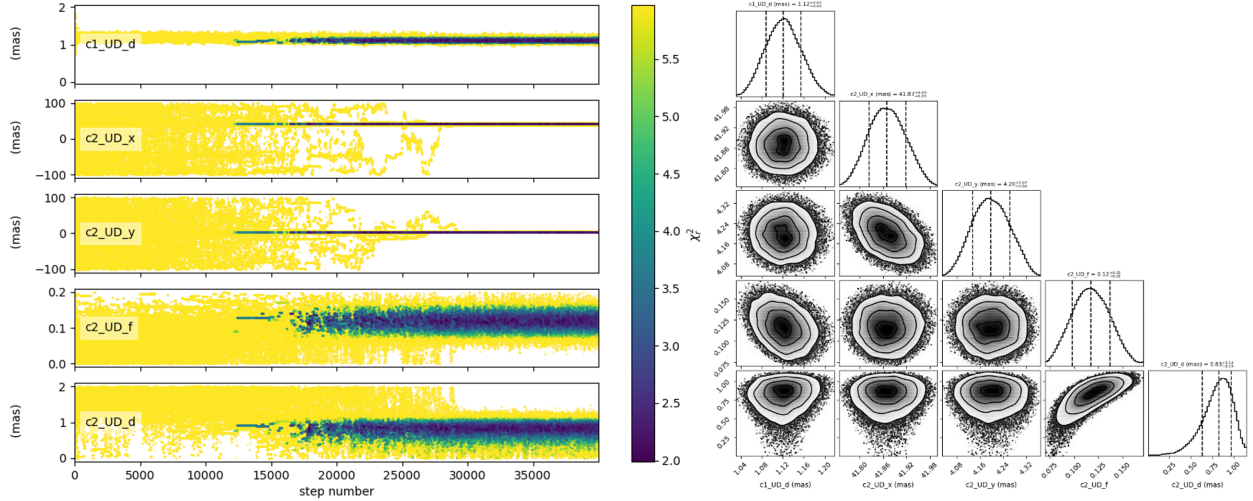


Figure 5. Emcee-based fitter used on CHARA/MIRCX data of the binary star  $\beta$  Ari: Plot of the walkers versus the simulation steps with a  $\chi_r^2$  colour-scale showing convergence of the five free parameters (left), corner plot of the posterior probability estimated on the last 5000 steps (right).

## 7. CONCLUSIONS & PERSPECTIVES

Although still under development, `oimodeler` is in a fully functional beta version, and can be used to model any OIFITS2-formatted interferometric data from the current instruments installed at VLTI and CHARA. The versatility of the code allows to build complex models which include chromaticity, time-dependence, and the use of external images or chromatic image-cubes computed from radiative-transfer codes. Consistency checks on interferometric models and fitting results have been performed using JMMC software ASPRO2 and LITPro.

## ACKNOWLEDGMENTS

Development of `oimodeler` is funded by the Lagrange Laboratory, the Observatoire de la Côte d’Azur and the [ANR project MASSIF](#). JV acknowledges support from the Fizeau exchange visitors program. The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement 101004719 (ORP).

## REFERENCES

- [1] Chesneau, O., Meilland, A., Chapellier, E., Millour, F., van Genderen, A. M., Nazé, Y., Smith, N., Spang, A., Smoker, J. V., Dessart, L., Kanaan, S., Bendjoya, P., Feast, M. W., Groh, J. H., Lobel, A., Nardetto, N., Otero, S., Oudmajer, R. D., Tekola, A. G., Whitelock, P. A., Arcos, C., Curé, M., and Vanzi, L., “The yellow hypergiant HR 5171 A: Resolving a massive interacting binary in the common envelope phase,” *A&A* **563**, A71 (Mar. 2014).
- [2] Lamberts, A., Millour, F., Liermann, A., Dessart, L., Driebe, T., Duvert, G., Finsterle, W., Girault, V., Massi, F., Petrov, R. G., Schmutz, W., Weigelt, G., and Chesneau, O., “Numerical simulations and infrared spectro-interferometry reveal the wind collision region in  $\gamma^2$  Velorum,” *MNRAS* **468**, 2655–2671 (July 2017).
- [3] Tallon-Bosc, I., Tallon, M., Thiébaud, E., Béchet, C., Mella, G., Lafrasse, S., Chesneau, O., Domiciano de Souza, A., Duvert, G., Mourard, D., Petrov, R., and Vannier, M., “LITpro: a model fitting software for optical interferometry,” in [*Optical and Infrared Interferometry*], Schöller, M., Danchi, W. C., and Delplancke, F., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **7013**, 70131J (July 2008).
- [4] Millour, F., “fitOmatic: Interferometric data modeling.” Astrophysics Source Code Library, record ascl:2301.005 (Jan. 2023).

- [5] Meilland, A., Millour, F., Kanaan, S., Stee, P., Petrov, R., Hofmann, K. H., Natta, A., and Perraut, K., “First spectro-interferometric survey of be stars. i. observations and constraints on the disk geometry and kinematics,” *A&A* **538**, A110 (Feb. 2012).
- [6] Vieira, R. G., Carciofi, A. C., and Bjorkman, J. E., “The pseudo-photosphere model for the continuum emission of gaseous discs,” *MNRAS* **454**, 2107–2119 (Dec. 2015).
- [7] Duvert, G., Young, J., and Hummel, C. A., “OIFITS 2: the 2nd version of the data exchange standard for optical interferometry,” *A&A* **597**, A8 (Jan. 2017).
- [8] Foreman-Mackey, D., Hogg, D. W., Lang, D., and Goodman, J., “emcee: The MCMC Hammer,” *PASP* **125**, 306 (Mar. 2013).
- [9] Speagle, J. S., “DYNESTY: a dynamic nested sampling package for estimating Bayesian posteriors and evidences,” *MNRAS* **493**, 3132–3158 (Apr. 2020).