Multi-technique observations vs. modelling of stellar systems

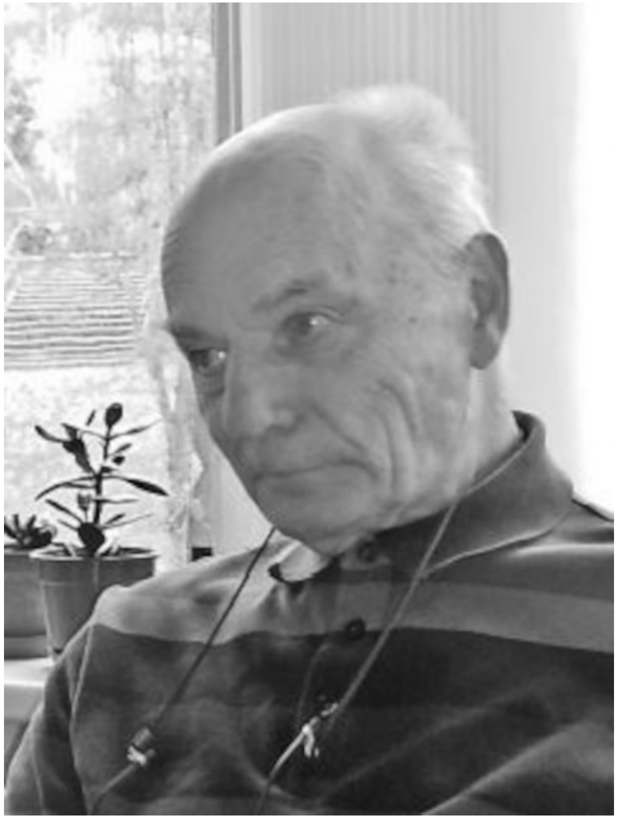
· 1″

Binaries 2024 Conference

M. Brož, P. Harmanec, A. Opluštilová, K. Vitovský, P. Doležal, ...

Thanks goes to...

Prof. Vokrouhlický



DrSc. Mayer

1. Is a model sufficient?

+ SWIFT integrator
+ Kepler
+ N-body perturbations, ...
+ stability!
+ fitting of orbits
+ simplex

1. astrometry (SKY)
2. radial velocity (RV)
3. transit-timing variations (TTV)

Brož et al. (2022, A&A 666, A24)

+ SWIFT integrator
+ Kepler
+ N-body perturbations, ...
+ stability!
+ oblateness
+ multipoles ($l = 10$)
+ parametrized post-Newtonian (PPN)
+ internal tides
+ external tides
− "brute-force"
− variable geometry
+ fitting of orbits
+ fitting of radiative parameters
− fitting of shapes
+ simplex
+ subplex
− annealing

1. astrometry (SKY)
2. differential astrometry (SKY2)
3. angular velocity (SKY3)
4. radial velocity (RV)
5. transit-timing variations (TTV)
6. eclipse duration (ECL)
7. visibility (VIS)
8. closure-phase (CLO)
9. triple product (T3)
10. light curve, u. Wilson-Devinney (LC)
11. light curve, u. polygonal (LC2)
12. synthetic spectra (SYN)
13. spectral-energy distribution (SED)
14. adaptive-optics silhouettes (AO)
15. adaptive-optics imaging (AO2)
16. occultations (OCC)

N-body
Levison & Duncan (1994)

$$\boldsymbol{f} = -\sum_{j \neq i}^{N} \frac{G m_j}{r_{ij}^3} \boldsymbol{r}_{ij} + \boldsymbol{f}_{\mathrm{ppn}} + \boldsymbol{f}_{\mathrm{oblat}} + \boldsymbol{f}_{\mathrm{multipole}} + \boldsymbol{f}_{\mathrm{tides}}.$$

PPN
Standish & Williams (2006)

$$\boldsymbol{f}_{\rm ppn} = \sum_{j\neq i}^{N} [-K_1 (K_2 + K_3 + K_4 + K_5 + K_6 + K_7 + K_8) \boldsymbol{r}_{ij} + K_1 (K_9 + K_{10}) \dot{\boldsymbol{r}}_{ij} + K_{11} \ddot{\boldsymbol{r}}_j],$$

$$K_1 = \frac{1}{c^2} \frac{Gm_j}{r_{ij}^3},$$

$$K_2 = -2(\beta + \gamma) \sum_{k\neq i} \frac{Gm_k}{r_{ik}},$$

$$K_3 = -(2\beta - 1) \sum_{k\neq j} \frac{Gm_k}{r_{jk}},$$

$$K_4 = \gamma v_i^2,$$

$$K_5 = (1 + \gamma) v_j^2,$$

$$K_6 = -2(1 + \gamma)\, \dot{\boldsymbol{r}}_i \cdot \dot{\boldsymbol{r}}_j,$$

$$K_7 = -\frac{3}{2} \frac{(\boldsymbol{r}_{ij} \cdot \dot{\boldsymbol{r}}_j)^2}{r_{ij}^2},$$

$$K_8 = \frac{1}{2} \boldsymbol{r}_{ji} \cdot \ddot{\boldsymbol{r}}_j,$$

$$K_9 = (2 + 2\gamma)\, \boldsymbol{r}_{ij} \cdot \dot{\boldsymbol{r}}_i,$$

$$K_{10} = -(1 + 2\gamma)\, \boldsymbol{r}_{ij} \cdot \dot{\boldsymbol{r}}_j,$$

$$K_{11} = \frac{3 + 4\gamma}{2c^2} \frac{Gm_j}{r_{ij}}.$$

multipoles
Burša et al. (1993)

$$U = -\frac{GM}{r}\sum_{\ell=0}^{N_{\rm pole}}\left(\frac{R}{r}\right)^{\ell}\sum_{m=0}^{\ell}P_{\ell m}(\cos\theta)[C_{\ell m}\cos(m\phi) + S_{\ell m}\sin(m\phi)],$$

$$\frac{{\rm d}U}{{\rm d}r} = -GM\sum_{\ell=0}^{N_{\rm pole}}R^{\ell}(-\ell-1)r^{-\ell-2}\sum_{m=0}^{\ell}P_{\ell m}(\cos\theta)[C_{\ell m}\cos(m\phi) + S_{\ell m}\sin(m\phi)],$$

$$\frac{{\rm d}U}{{\rm d}\theta} = -GM\sum_{\ell=0}^{N_{\rm pole}}R^{\ell}r^{-\ell-1}\sum_{m=0}^{\ell}P'_{\ell m}(\cos\theta)\sin\theta[C_{\ell m}\cos(m\phi) + S_{\ell m}\sin(m\phi)],$$

$$\frac{{\rm d}U}{{\rm d}\phi} = -GM\sum_{\ell=0}^{N_{\rm pole}}R^{\ell}r^{-\ell-1}\sum_{m=0}^{\ell}P_{\ell m}(\cos\theta)[-C_{\ell m}\sin(m\phi)m + S_{\ell m}\cos(m\phi)m],$$

$$\boldsymbol{f}_{\rm mp} = -\left(\frac{{\rm d}U}{{\rm d}r}, \frac{1}{r}\frac{{\rm d}U}{{\rm d}\theta}, \frac{1}{r\sin\theta}\frac{{\rm d}U}{{\rm d}\phi}\right),$$

$$C_{\ell 0} = \frac{1}{MR^{\ell}}\rho\int_V |\boldsymbol{r}|^{\ell}P_{\ell}(\cos\theta)\,{\rm d}V,$$

$$C_{\ell m} = \frac{2}{MR^{\ell}}\frac{(\ell-m)!}{(\ell+m)!}\rho\int_V |\boldsymbol{r}|^{\ell}P_{\ell m}(\cos\theta)\cos(m\phi)\,{\rm d}V,$$

$$S_{\ell m} = \frac{2}{MR^{\ell}}\frac{(\ell-m)!}{(\ell+m)!}\rho\int_V |\boldsymbol{r}|^{\ell}P_{\ell m}(\cos\theta)\sin(m\phi)\,{\rm d}V,$$

$$P_0(x) = 1\,, \quad P_1(x) = x\,, \quad P_2(x) = \frac{1}{2}(3x^2-1), \dots$$

$$P_{11}(x) = (1-x^2)^{\frac{1}{2}}\,, \quad P_{21}(x) = 3x(1-x^2)^{\frac{1}{2}}\,, \dots$$

tides
Mignard (1979)

$$\boldsymbol{f}_{\text{tides}} = K_1 \left[K_2 \boldsymbol{r}' - K_3 \boldsymbol{r} - K_4 (\boldsymbol{r} \times \omega + \boldsymbol{v}) + K_5 (K_6 \boldsymbol{r} - K_7 \boldsymbol{r}') \right],$$

$$K_1 = \frac{3 G m^\star R^5 k_2 \Delta t}{(r' r)^5},$$

$$K_2 = \frac{5}{r'^2} \left[\boldsymbol{r}' \cdot \boldsymbol{r} \, (\boldsymbol{r} \cdot \omega \times \boldsymbol{r}' + \boldsymbol{r}' \cdot \boldsymbol{v}) - \frac{1}{2 r^2} \boldsymbol{r} \cdot \boldsymbol{v} (5 (\boldsymbol{r}' \cdot \boldsymbol{r})^2 - r'^2 r^2) \right],$$

$$K_3 = \boldsymbol{r} \cdot \omega \times \boldsymbol{r}' + \boldsymbol{r}' \cdot \boldsymbol{v},$$

$$K_4 = \boldsymbol{r}' \cdot \boldsymbol{r},$$

$$K_5 = \frac{\boldsymbol{r} \cdot \boldsymbol{v}}{r^2},$$

$$K_6 = 5 \boldsymbol{r}' \cdot \boldsymbol{r},$$

$$K_7 = r^2,$$

$$\boldsymbol{\Gamma} = \boldsymbol{r} \times m' \boldsymbol{f}_{\text{tides}}.$$

Vitovský & Brož (A&A, submit.)

- a continuation of Brož et al. (2021) on β Lyr A
- analytical accretion disk of Shakura & Sunyæv (1973)
- modified for a *general* opacity prescription $\kappa = \kappa_0 \rho^A T^B$
- constrained by the accretion rate $dM/dt = 2 \cdot 10^{-5}\, M_S\, y^{-1}$
- radial profiles $\Sigma(r)$, $T(r)$, $H(r)$ compared to “observations”
- most models excluded due to self-consistency (P_{gas} vs. P_{rad}); κ is Kramers
- Σ must be much higher! 10000 kg m^{-2} at the inner rim, if $\alpha = 0.1$
- T must be much higher! 10^5 K in the mid-plane
- cf. vertical scale height H *is* hydrostatic (low μ)

$$\kappa = \kappa_\star \alpha^{-\frac{7A+2B}{D}} \dot{M}^{\frac{4(A+B)}{D}} M_\star^{\frac{(11A+6B)}{2D}} r^{-\frac{3(11A+6B)}{2D}} \left(1 - \sqrt{\frac{R_\star}{r}}\right)^{\frac{4(A+B)}{D}} , \tag{15}$$

$$\kappa_\star = \left(\kappa_0 \rho_0^A T_0^B\right)^{\frac{10}{D}} , \tag{16}$$

$$H = H_\star \alpha^{-\frac{A+1}{D}} \dot{M}^{\frac{A+2}{D}} M_\star^{-\frac{A-2B+7}{2D}} r^{\frac{3(A-2B+7)}{2D}} \left(1 - \sqrt{\frac{R_\star}{r}}\right)^{\frac{A+2}{D}} , \tag{17}$$

$$H_\star = H_0 \kappa_\star^{\frac{1}{10}} , \tag{18}$$

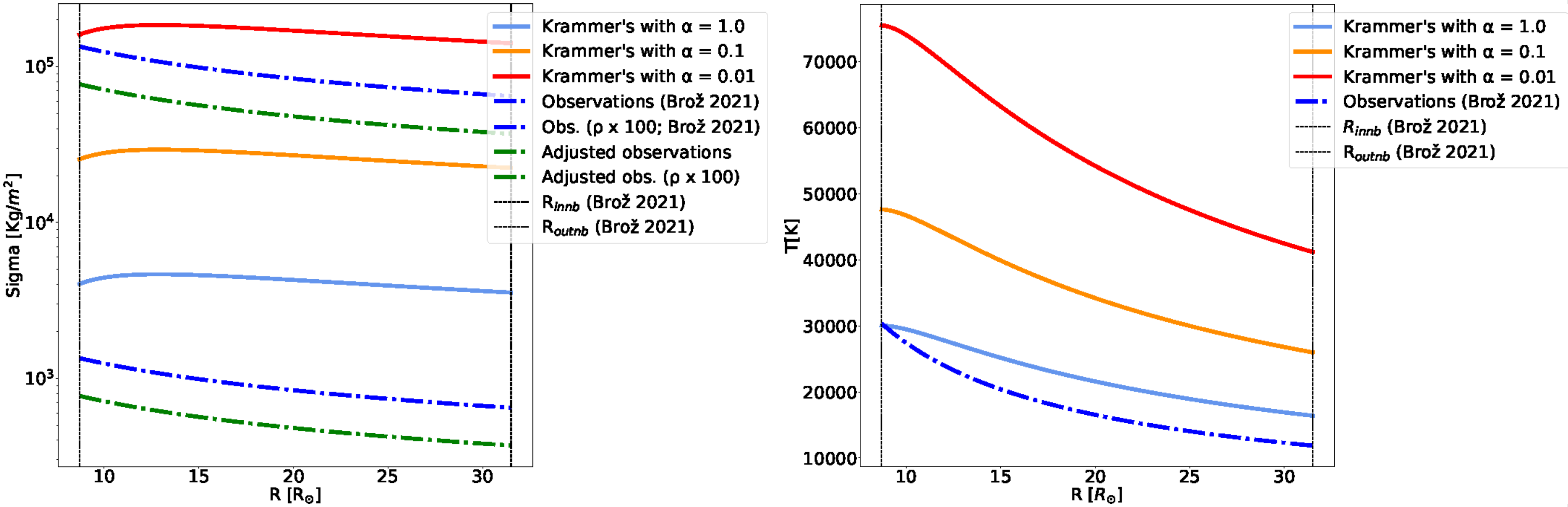
$$\Sigma = \Sigma_\star \alpha^{-\frac{A-2B+8}{D}} \dot{M}^{\frac{A-2B+6}{D}} M_\star^{-\frac{A+2B-4}{2D}} r^{\frac{3(A+2B-4)}{2D}} \left(1 - \sqrt{\frac{R_\star}{r}}\right)^{\frac{A-2B+6}{D}} , \tag{19}$$

$$\Sigma_\star = \Sigma_0 \kappa_\star^{-\frac{1}{5}} , \tag{20}$$

$$T = T_\star \alpha^{-\frac{2(A+1)}{D}} \dot{M}^{\frac{2A+4}{D}} M_\star^{\frac{2A+3}{D}} r^{-\frac{3(2A+3)}{D}} \left(1 - \sqrt{\frac{R_\star}{r}}\right)^{\frac{2A+4}{D}} , \tag{21}$$

$$T_\star = T_0 \kappa_\star^{\frac{1}{5}} , \tag{22}$$

by K. Vitovský



Sigma [Kg/m^2]
10^5
10^4
10^3
10
15
20
25
30
R [$R_\odot$]
Krammer's with α = 1.0
Krammer's with α = 0.1
Krammer's with α = 0.01
Observations (Brož 2021)
Obs. (ρ x 100; Brož 2021)
Adjusted observations
Adjusted obs. (ρ x 100)
R_{innb} (Brož 2021)
R_{outnb} (Brož 2021)
T[K]
70000
60000
50000
40000
30000
20000
10000
10
15
20
25
30
R [$R_\odot$]
Krammer's with α = 1.0
Krammer's with α = 0.1
Krammer's with α = 0.01
Observations (Brož 2021)
R_{innb} (Brož 2021)
R_{outnb} (Brož 2021)

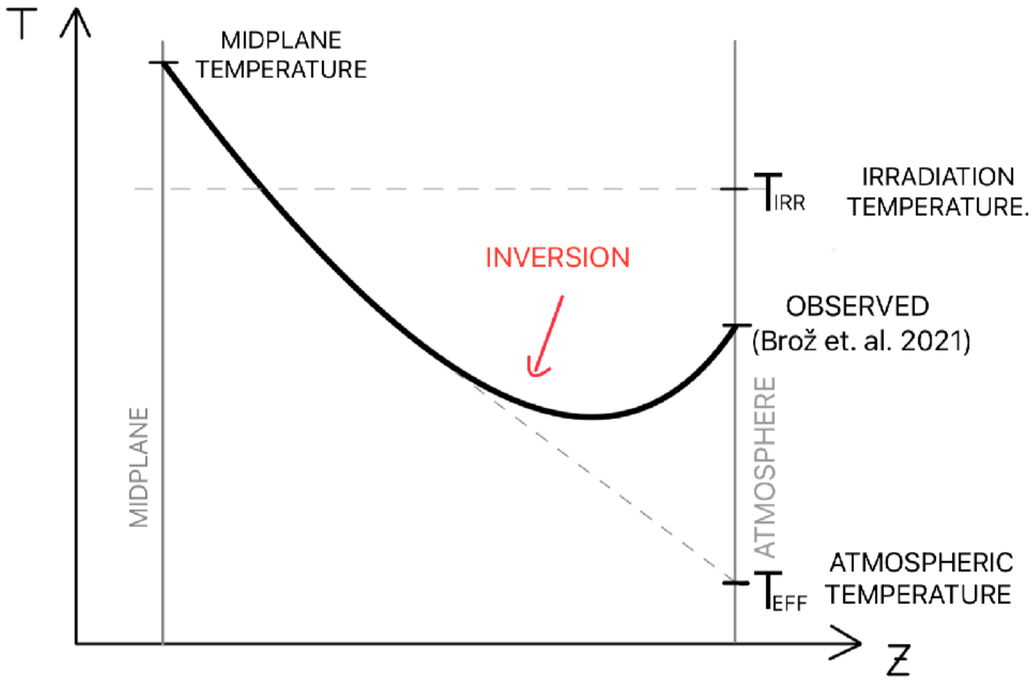


Fig. 12: A sketch of how the presence of a temperature inversion in the vertical profile of the disc could reconcile the computed midplane temperature $T_{\rm mid}$ profile, the observed temperature profile and the calculated atmospheric $T_{\rm eff}$ and irradiation $T_{\rm irr}$ profiles.

Coordinates

- SKY ... photocentric x, y
- SKY2 ... 2-, or 3-centric x, y
- SKY3 ... photocentric v_x, v_y
- RV ... barycentric v_z
- TTV ... barycentric z
- ECL ... heliocentric x, y
- VIS ... barycentric x, y
- LC ... heliocentric x, y, z
- OCC ... Earth, topocentric x, y, z
- cf. Jacobi elements a, e, i, Ω, ω, M
- cf. “times of interest”

```
Coordinate convention:

   y (as DE, N)

   ^    z (v_rad)
   |
   |  /
   | /
   |/
  -|----------> x (-RA, W)
   |

  internal

  (x, y) is plane-of-sky
  x positive towards W
  y positive towards N
  no reflections
  z radial, away from observer
```

2. How to derive observables?

- synthetic spectra: OSTAR, BSTAR, AMBRE, POLLUX, PHŒNIX, POWR, ...

$$L_j(T_{\mathrm{eff}\,j},\,R_j) = 4\pi R_j^2 \int_{\lambda-\Delta\lambda/2}^{\lambda+\Delta\lambda/2} F_{\mathrm{syn}}\,(\lambda,\,T_{\mathrm{eff}\,j},\,\log g_j,\,v_{\mathrm{rot}\,j},\,\mathcal{Z}_j)d\lambda;$$

$$I'_\lambda = \sum_{j=1}^{N_{\mathrm{bod}}} \frac{L_j}{L_{\mathrm{tot}}} I_{\mathrm{syn}}\left[\lambda\left(1 - \frac{v_{z\mathrm{b}j+\gamma}}{c}\right),\,T_{\mathrm{eff}\,j},\,\log g_j,\,v_{\mathrm{rot}\,j},\,\mathcal{Z}_j\right];$$

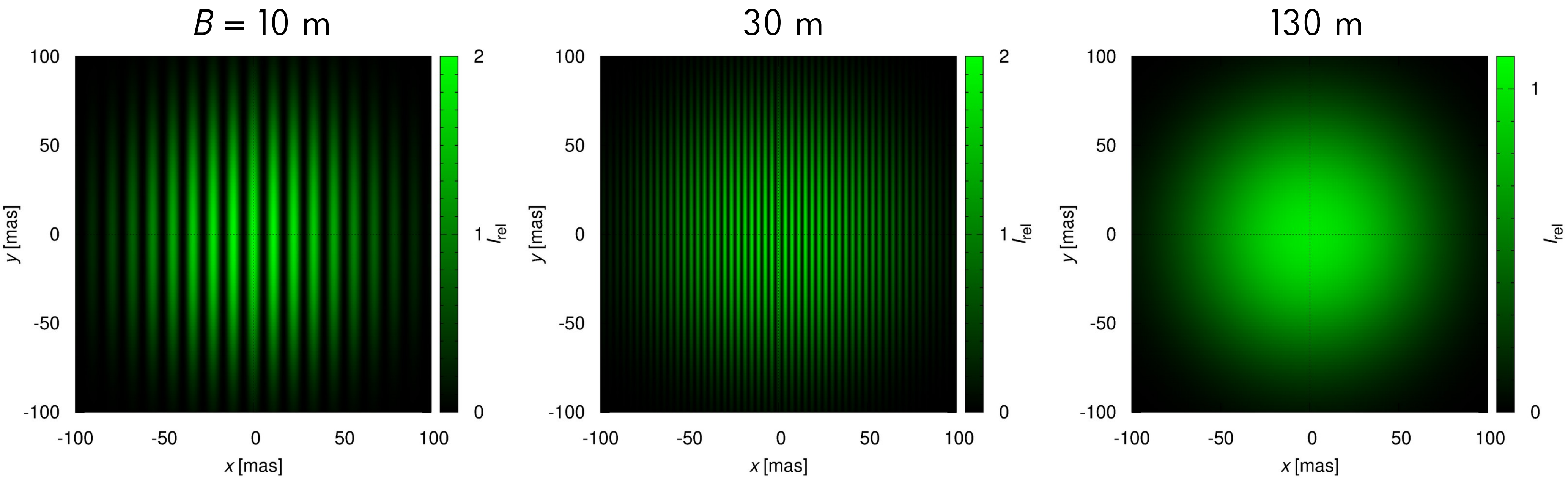
$$F'_V = \sum_{j=1}^{N_{\mathrm{bod}}} \left(\frac{R_j}{d}\right)^2 \int_0^\infty F_{\mathrm{syn}}\,[\lambda,\,T_{\mathrm{eff}\,j},\,\log g_j,\,v_{\mathrm{rot}\,j},\,\mathcal{Z}_j]\,f_V\,(\lambda)d\lambda,$$

- sphere vs. Roche (Lahey & Lahey 2015)
- cf. http://sirrah.troja.mff.cuni.cz/~mira/xitau/

ten Brummelaar etal. (2005)

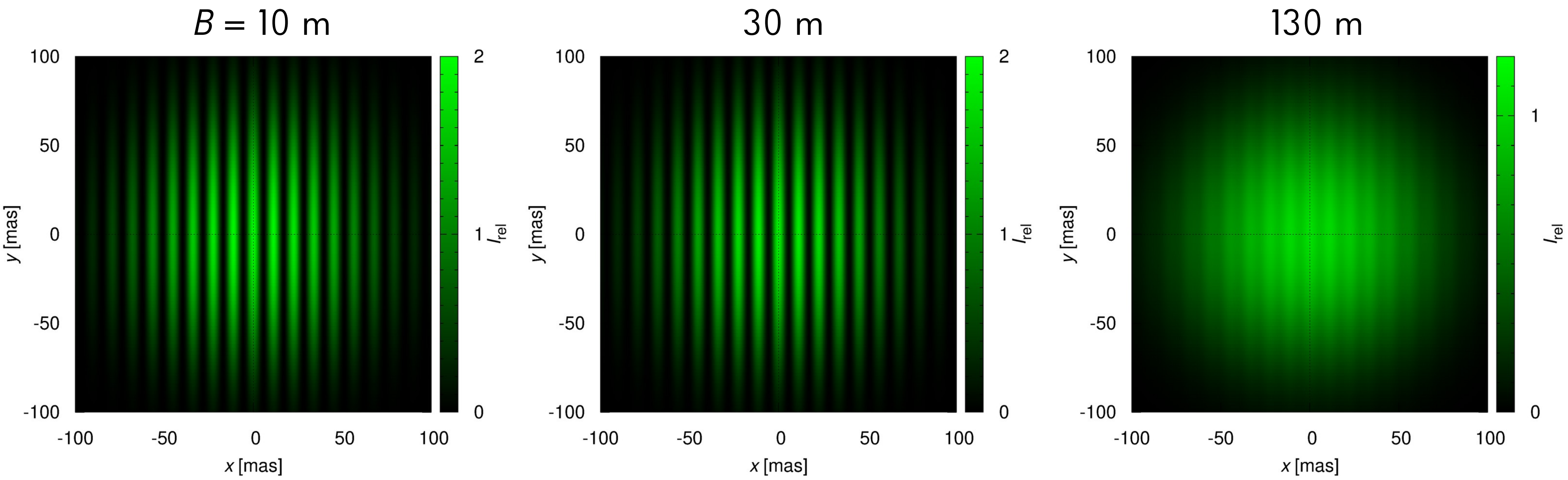
Fringes

- $D = 1$ m, $B = 100$ m, $\lambda = 550$ nm, o. of a disc, $\theta = 1$ mas, no seeing, no $\Delta\lambda_{\mathrm{eff}}$, ...
- a drop in *visibility* (contrast) of fringes, i.e., the goal!

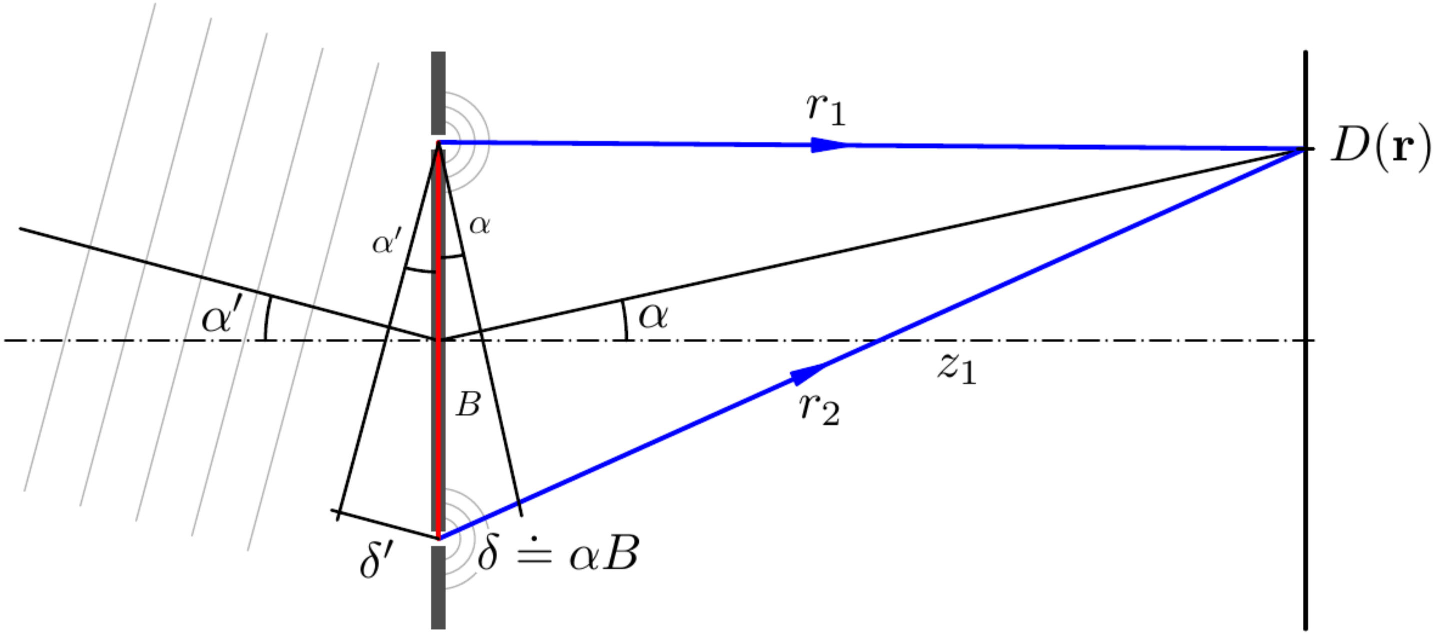


Fringes (cont.)

- delay line, periscopes → rearrangement of pupils (***B*** vs. ***b***) → constant # of f.



Fringes (cont.)



Obrázek 6.51: Uspořádání Youngova experimentu, kde B označuje vzájemnou vzdálenost štěrbin (základnu), z_1 vzdálenost stínítka od překážky, r_1, r_2 vzdálenost studovaného místa na stínítku od štěrbin, α odpovídající odchylka od osy překážky, α' úhel dopadu vlny na překážku, δ, δ' dráhové rozdíly vznikající za a před překážkou.

van Cittert-Zernike theorem

- intensity I [1], angles α, α' [rad], wave number $k = 2\pi/\lambda$ [m^{-1}], baseline B [m]

$$I(\alpha, \alpha') = I_0\{1 + \cos[k(\alpha + \alpha')B]\}\,, \tag{6.156}$$

$$I(\alpha) = \int I(\alpha, \alpha')\mathrm{d}\alpha' = \overbrace{\int I(\alpha')\mathrm{d}\alpha'}^{= I_0} + \overbrace{\int I(\alpha')\cos[k(\alpha + \alpha')B]\mathrm{d}\alpha'}^{= \Re[\mathrm{e}^{\mathrm{i}k\alpha B}\int I(\alpha')\,\mathrm{e}^{\mathrm{i}k\alpha' B}\,\mathrm{d}\alpha']}\,, \tag{6.158}$$

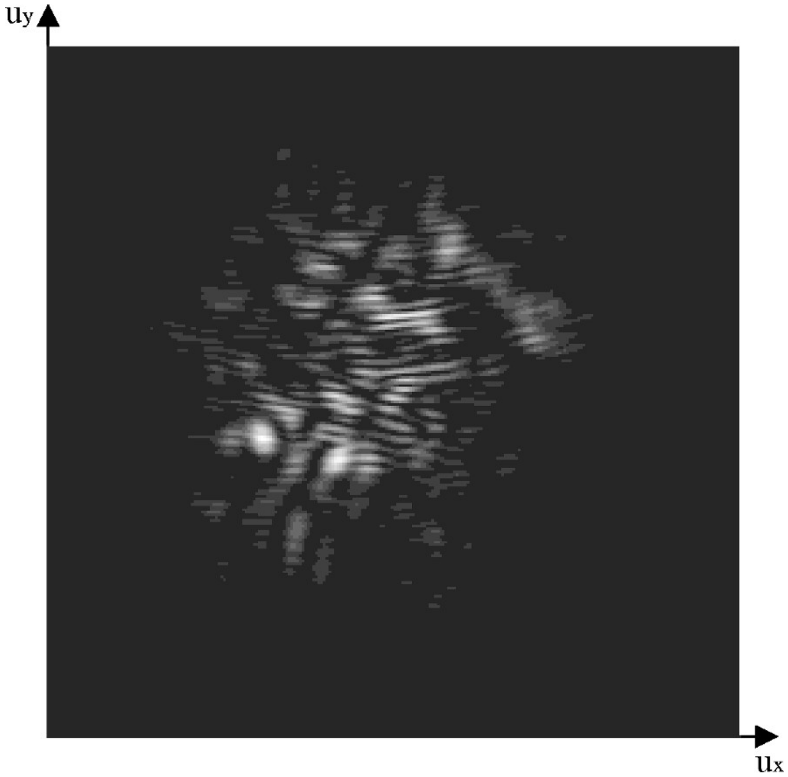
$$I(\vec{\alpha}) = I_0\left\{1 + \Re\left[\mu(\vec{B})\,\mathrm{e}^{-\mathrm{i}k\vec{\alpha}\cdot\vec{B}}\right]\right\}\,, \tag{6.159}$$

$$\mu(\vec{B}) \equiv \frac{\int I(\vec{\alpha}')\,\mathrm{e}^{-\mathrm{i}k\vec{\alpha}'\cdot\vec{B}}\,\mathrm{d}\alpha'}{I_0}\,, \tag{6.160}$$

Interferometric observables

- complex visibility $\mu = F(I)$
- squared visibility $V^2 = \mu\mu^*$
- phase arg μ
- triple product $T_3 = \mu_{12}\ \mu_{23}\ \mu_{31}$
- closure phase arg T_3
- triple product amplitude $|T_3|$
- differential visibility $\Delta V = \mu_{\lambda 1}\ \mu_{\lambda 2}^*$, approx. $V_{\lambda 1} \sim V_{\mathrm{contiuum}}$ (cf. Mourard et al. 2009)
- differential visibility amplitude $|\Delta V|$
- differential phase arg ΔV
- estimator $C_1 = 2E_{\mathrm{fringe}}/E_{\mathrm{speckle}}$, $E \equiv \int W \mathrm{d}f$ (Roddier & Lena 1984, Mourard et al. 1994)
- estimator $C_2 = 2W_{\mathrm{fringe}}(f)/W_{\mathrm{speckle}}(f - B/\lambda)$
- cross-spectrum $W_{12} = \langle \mathrm{F}(I_{\lambda 1})\, \mathrm{F}(I_{\lambda 2})^* \rangle$ (Berio et al. 1999, 2001)
- ...

Interferometric observables



Berio et al. (1999)

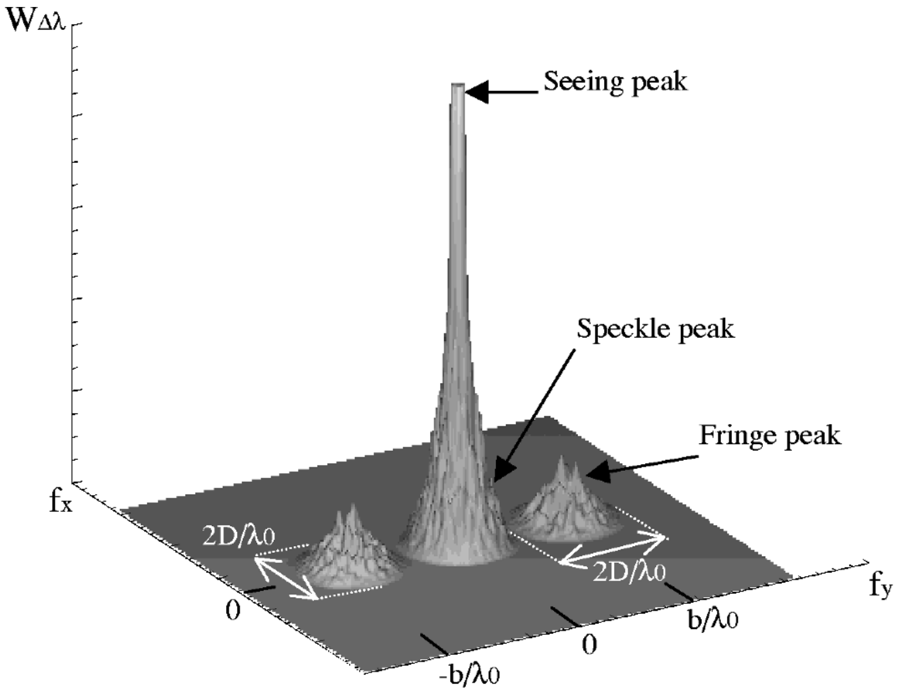
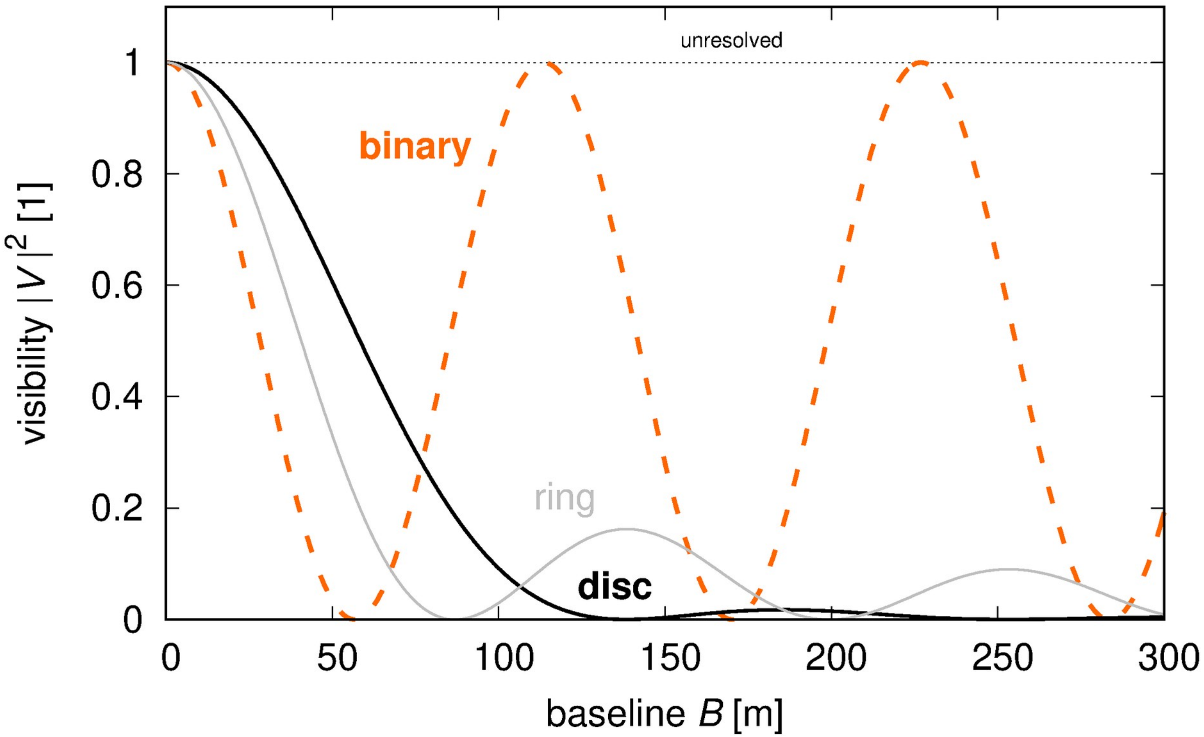


Fig. 1. Top, numerically simulated GI2T interferogram in the multichromatic mode and bottom, the corresponding spectral density. The fringe peaks are centered at $\pm b/\lambda_0$ because of the pupil rearrangement. (Coordinates are in arbitrary units.)

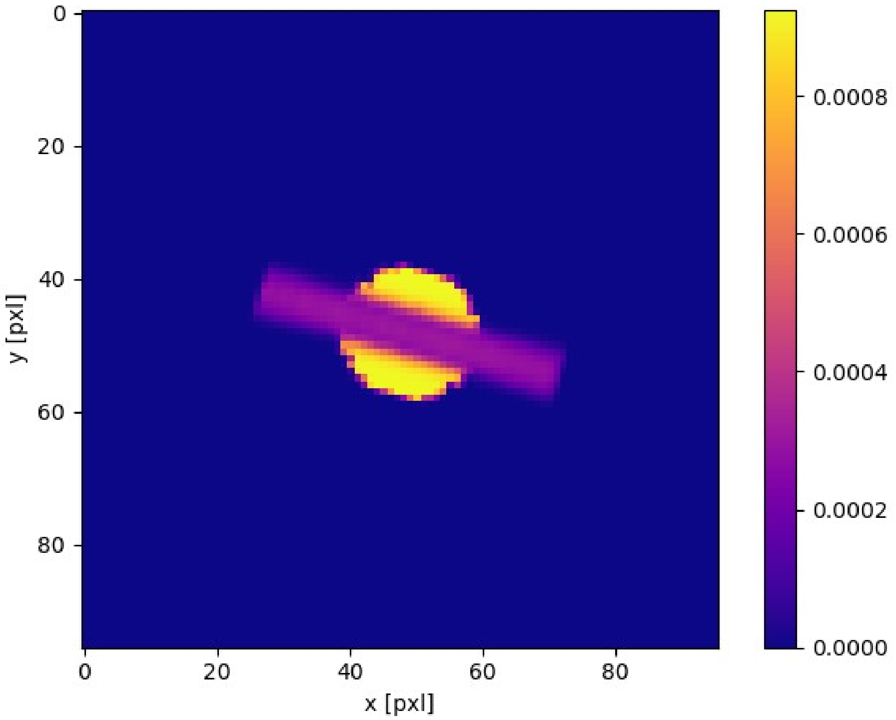
Simple i. models

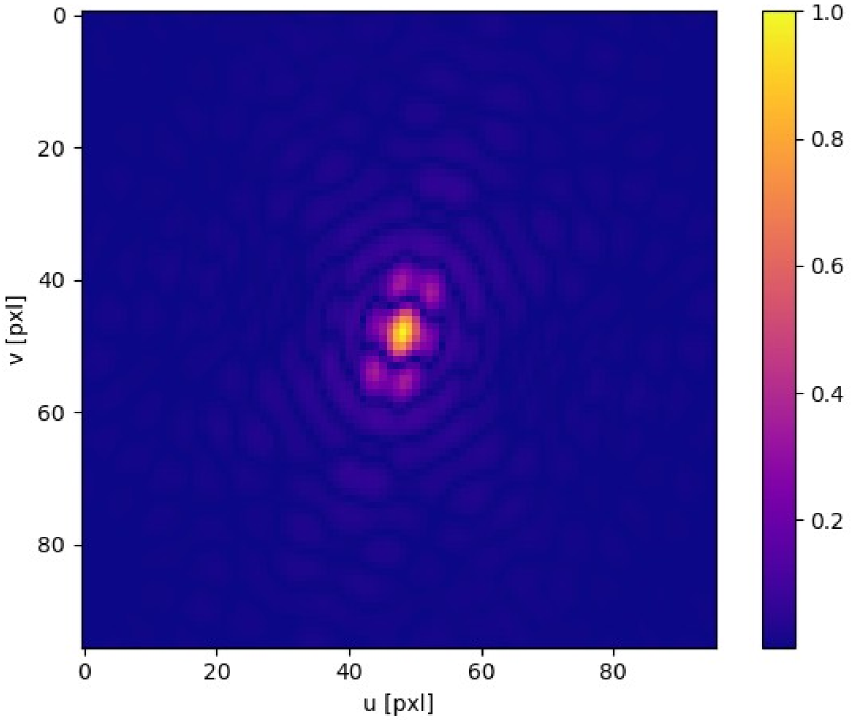
- binaries, multiple *, uniform disk(s), limb-darkened d., ring, ... cf. combinations!



Complex i. models

- synthetic image $I(\alpha_x, \alpha_y)$ → Fourier transform → s. complex visibility $\mu(u, v)$, ...





Phoebe interferometric module

- 3 new datasets: VIS, CLO, T3
- complex i. visibility == Fourier transform
- partial eclipses of triangles!
- limb darkening, gravity brightening, rotation, Roche distortion, eclipses, reflection
- cf. simple models (for comparison)
- observations (O), calculations (C), O-C, fitting, plotting, etc.
- tutorials...

- https://github.com/miroslavbroz/phœbe2/tree/interferometry

'vis' (Interferometric Visibility) Dataset

Background

An interferometer measures *fringes*. When an unresolved source is observed by 2 telescopes, one can see the Airy disk (corresponding to the diameter D) together with fringes originating from interference of light (corresponding to the baseline B). In optical, the light from telescopes is kept coherent with help of a delay line. In radio, the signal from antennas is sent to a correlator and processed off-line.

```
          *        source

     -------------  wavefront

     | |       | |  2 telescopes
     |_|       |_|  diameter D
      |         |
      \---\ /---/   baseline B
          | |       delay line
          \ /

          |||       fringes
```

For a point-like, monochromatic source, the intensity in the focal plane is simply:

`I(alpha, alpha') = I_0 (1 + cos(k(alpha + alpha')B))`

where I_0 denotes the mean intensity, k = 2 pi/lambda, the wave number, alpha, alpha', the angles inside and in front of the instrument, and B, the projected baseline.

Note: For a general orientation, \vec alpha, \vec alpha', and (u, v) = \vec B/lambda are used.

For an extended, monochromatic source, we should integrate over alpha':

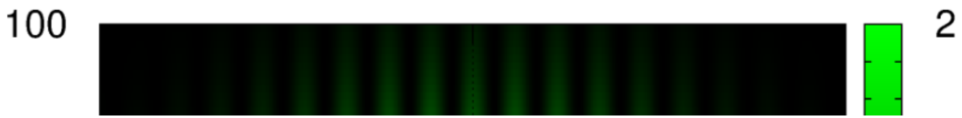
`I(alpha) = \int I(alpha, alpha') dalpha' = \int I_0 dalpha' + Re \int I_0 exp(-i k(alpha + alpha')B) dalpha'`

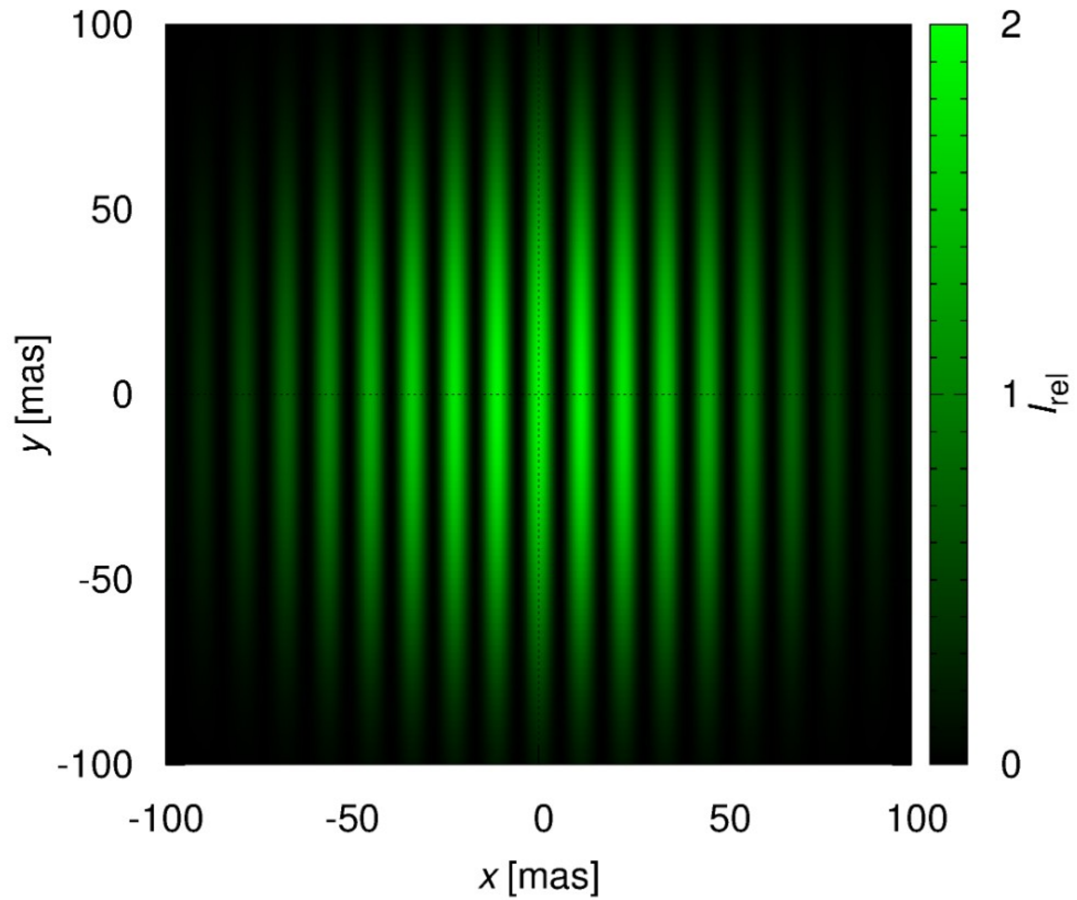
where Re denotes the real part. Grouping together the terms dependent on alpha and alpha':

`I(alpha) = I_0(1 + Re(mu(B) exp(-i k alpha B)))`

`mu(B) = 1/I_0 \int I(alpha') exp(-i k alpha' B) dalpha'`

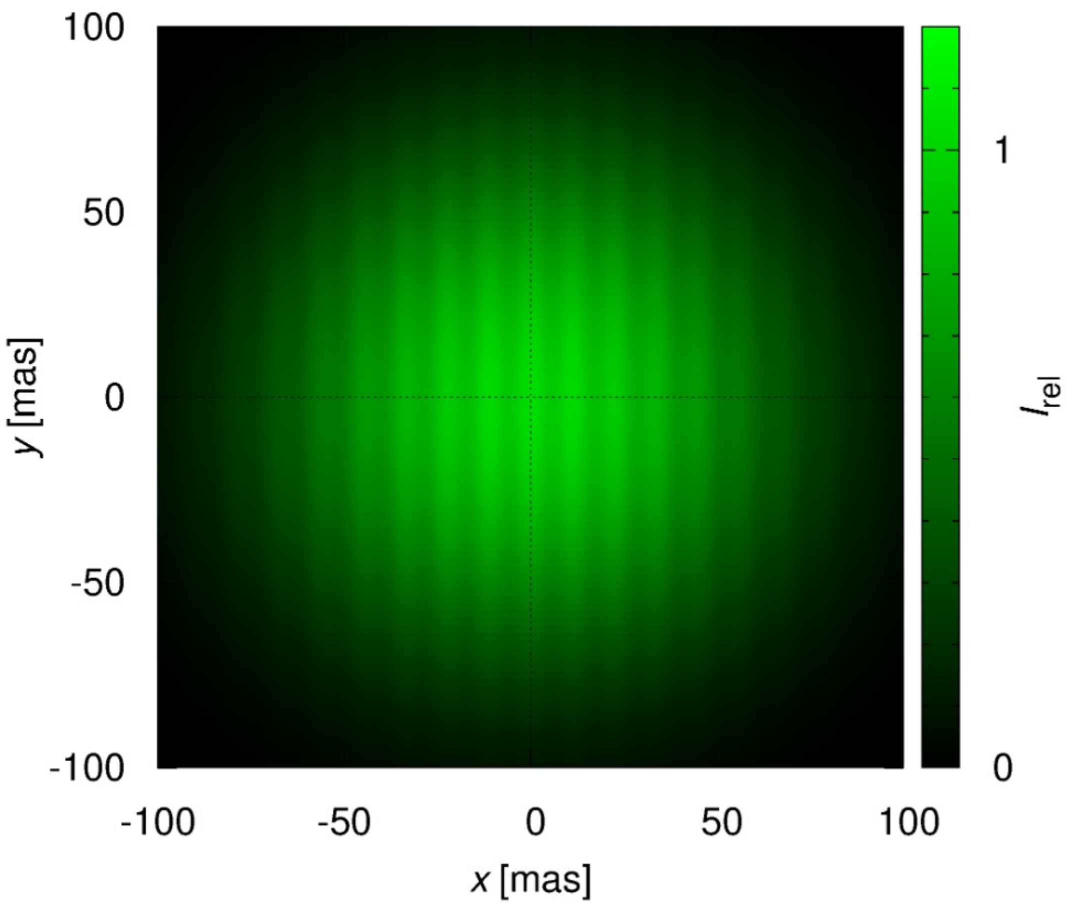
where the complex factor mu(B) determines the *visibility* (contrast) of the fringes. It corresponds to the spatial Fourier transform of the intensity of the source, I(alpha'). This is the statement of the van Cittert and Zernike theorem.





Example of an interferometric observation of a disk, phi = 1 mas, D = 1 m, B = 10 m, lambda = 550 nm, no seeing, no bandwidth.

Example of an interferometric observation of a disk, phi = 1 mas, D = 1 m, B = 10 m, lambda = 550 nm, no seeing, no bandwidth.



Ditto for B = 130 m (w. rearranged pupils, i.e., constant # of fringes). A *drop* in visibility (contrast) of fringes is the goal!

Setup

Setup

As a preparatory task, we create a file `Vis.dat` , containing common interferometric data. Apart from times, we also need baselines (u, v) and wavelengths (and bandwidths). Here, an interferometer composed of 2 telescopes "observed" on baselines up to 330 m (like CHARA). For simplicity, the "observed" squared visibility |V|^2 = mu mu* was set to 0.0. Of course, in real life this value is observed (between 0 and 1), as read from OIFITS files.

In [1]:
```
f = open("Vis.dat", "w")
f.write("# time u v wavelength bandwidth vis sigma\n")

t = 0.25                # d
u1 = 0.0                # m
u2 = 330.0              # m
du = 1.0                # m
v = 0.0                 # m
wavelength = 550.0e-9   # m
bandwidth = 100.0e-9    # m
vis = 0.0               # 1
sigma = 0.1             # 1

u = u1
while u < u2:
    u += du
    f.write("%.8f  %.8e  %.8e  %.8e  %.8e  %.8f  %.8f\n" % (t, u, v, wavelength, bandwidth, vis, sigma))

u = 0.0                 # m
v1 = 0.0                # m
v2 = 330.0              # m
dv = 1.0                # m

v = v1
while v < v2:
    v += dv
    f.write("%.8f  %.8e  %.8e  %.8e  %.8e  %.8f  %.8f\n" % (t, u, v, wavelength, bandwidth, vis, sigma))

f.close()
```

Note: Make sure to have the latest version of PHOEBE 2.5 installed (uncomment this line if running in an online notebook session such as colab).

In [2]:
```
#!pip install -I "phoebe>=2.5,<2.6"
```

As always, let's do imports and add a new Bundle.

In [3]:
```
import phoebe

b = phoebe.default_binary()
```

Parameters

Next read `Vis.dat` back and add the corresponding 'vis' dataset:

Parameters

Next read `Vis.dat` back and add the corresponding 'vis' dataset:

In [4]:
```
import numpy as np

times, u, v, wavelengths, vises, sigmas = np.loadtxt("Vis.dat", usecols=[0, 1, 2, 3, 5, 6], unpack=True)

b.add_dataset('vis', times=times, u=u, v=v, wavelengths=wavelengths, vises=vises, sigmas=sigmas, if_method='integra
```

Out[4]:
```
<ParameterSet: 56 parameters | contexts: figure, dataset, compute>
```

To verify:

In [5]:
```
print(b.get_dataset(kind='vis'))
```

```
ParameterSet: 12 parameters
              times@vis01@dataset: [0.25 0.25 0.25 ... 0.25 0.25 0.25] d
                  u@vis01@dataset: [1. 2. 3. ... 0. 0. 0.] m
                  v@vis01@dataset: [  0.   0.   0. ... 328. 329. 330.] m
        wavelengths@vis01@dataset: [5.5e-07 5.5e-07 5.5e-07 ... 5.5e-07 5.5e-07
 5.5e-07] m
              vises@vis01@dataset: [0. 0. 0. ... 0. 0. 0.]
      compute_times@vis01@dataset: [] d
             sigmas@vis01@dataset: [0.1 0.1 0.1 ... 0.1 0.1 0.1]
          if_method@vis01@dataset: integrate
           passband@vis01@dataset: Johnson:V
   intens_weighting@vis01@dataset: energy
    ld_mode@primary@vis01@dataset: interp
   ld_mode@secondary@vis01@dat...: interp
```

times

To see explanations:

In [6]:
```
print(b.get_parameter(kind='vis', qualifier='times', context='dataset'))
```

```
Parameter: times@vis01@dataset
                       Qualifier: times
                     Description: Observed times
                           Value: [0.25 0.25 0.25 ... 0.25 0.25 0.25] d
                  Constrained by: 
                      Constrains: None
                      Related to: None
```

u

Alternatively, one can use the `twig` syntax.

Note: Here, u coordinate is in metres. Above, (u, v) = B/lambda was in cycles per baseline.

Model

Eventually, a computation is run as:

In [14]:
```
b.run_compute()
```

```
100%|██████████| 1/1 [00:01<00:00,  1.04s/it]
```

Out[14]:
```
<ParameterSet: 7 parameters | qualifiers: vises, u, comments, if_method, v, wavelengths, times>
```

Sometimes, when I don't know anything, I list all twigs:

In [15]:
```
f = open('twigs.txt', 'w')
for twig in b.twigs:
  f.write("%s\n" % (twig))
f.close()
```

Now, I know that I can print, e.g.:

In [16]:
```
print(b.get_model(kind='vis'))
```

```
ParameterSet: 6 parameters
R              times@latest@model: [0.25 0.25 0.25 ... 0.25 0.25 0.25] d
R                  u@latest@model: [1. 2. 3. ... 0. 0. 0.] m
R                  v@latest@model: [  0.   0.   0. ... 328. 329. 330.] m
R        wavelengths@latest@model: [5.5e-07 5.5e-07 5.5e-07 ... 5.5e-07 5.5e-07
 5.5e-07] m
R              vises@latest@model: [0.99995203 0.99980813 0.99956834 ... 0.85391424
 0.85308086 0.85224563]
           if_method@latest@model: integrate
```

To save results:

In [17]:
```
times = b['times@vis01@phoebe01@latest@vis@model'].value
u = b['u@vis01@phoebe01@latest@vis@model'].value
v = b['v@vis01@phoebe01@latest@vis@model'].value
wavelengths = b['wavelengths@vis01@phoebe01@latest@vis@model'].value
vises = b['vises@vis01@phoebe01@latest@vis@model'].value

np.savetxt('model.out', np.c_[times, u, v, wavelengths, vises], header='times u v wavelenghts vises')
```

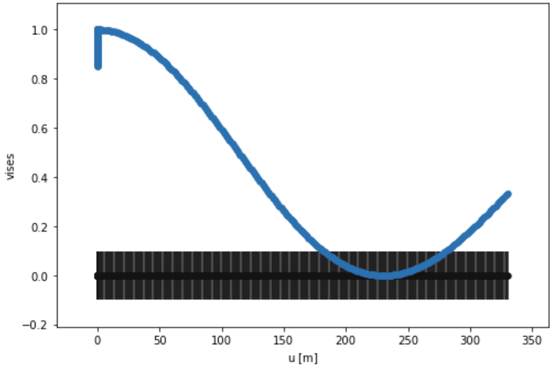
Plotting

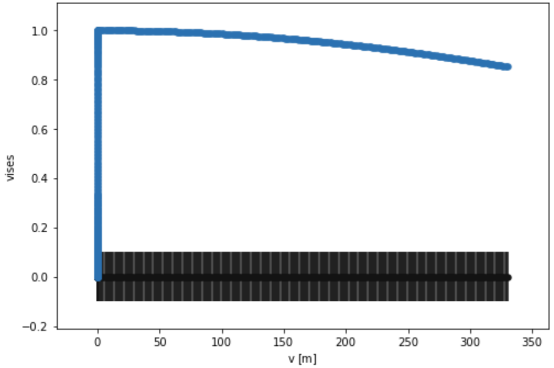
To plot results:

In [18]:
```
b.plot(show=True)
```

```
/home/mira/.local/lib/python3.9/site-packages/phoebe/dependencies/autofig/axes.py:1273: UserWarning: Attempting to
set identical left == right == 0.25 results in singular transformations; automatically expanding.
  ax.set_xlim(xlim)
```

In [19]:
```
b.plot(x='u', marker='.', linestyle='none', show=False)
b.plot(x='v', marker='.', linestyle='none', show=True)
```

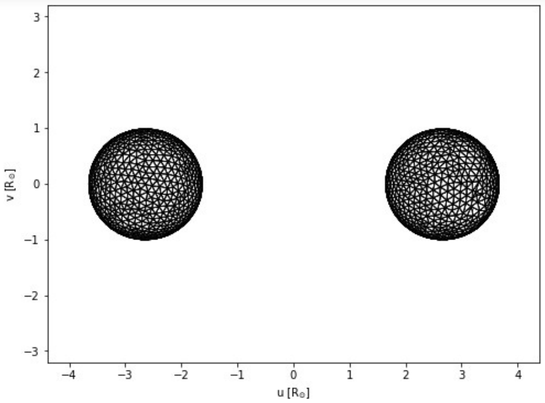




Out[19]:
```
(<autofig.figure.Figure | 2 axes | 4 call(s)>,
 <Figure size 576x864 with 2 Axes>)
```

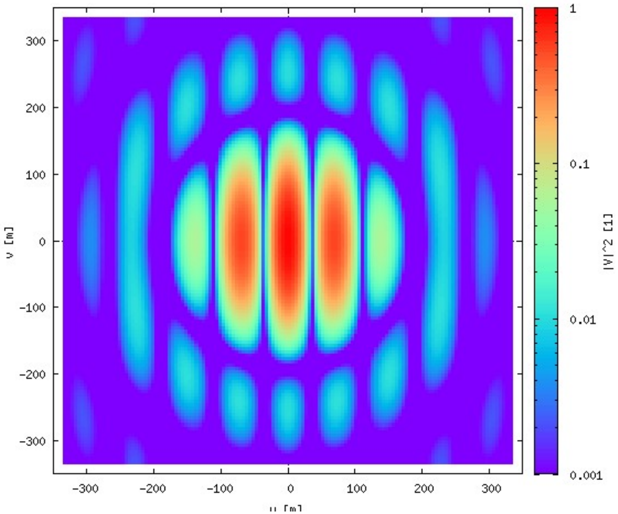
|V|^2 vs. u is sensitive to the mutual separation of `b.default_binary()`, which is 5.3 R_S. On the other hand, |V|^2 vs. v is sensitive to the diameters of components, which is 2.0 R_S, hence the visibility decreases more slowly. The smaller angular size, the slower decrease of |V|^2, and *vice versa*. Cf.

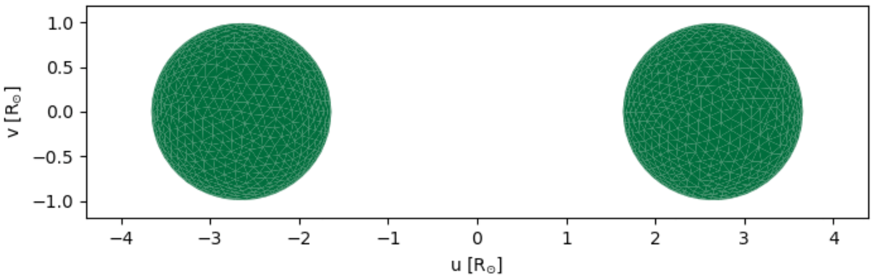
In [23]:
```
b.add_dataset(kind='mesh', compute_times=[0.25])
b.run_compute()
b.plot(kind='mesh', show=True)
```

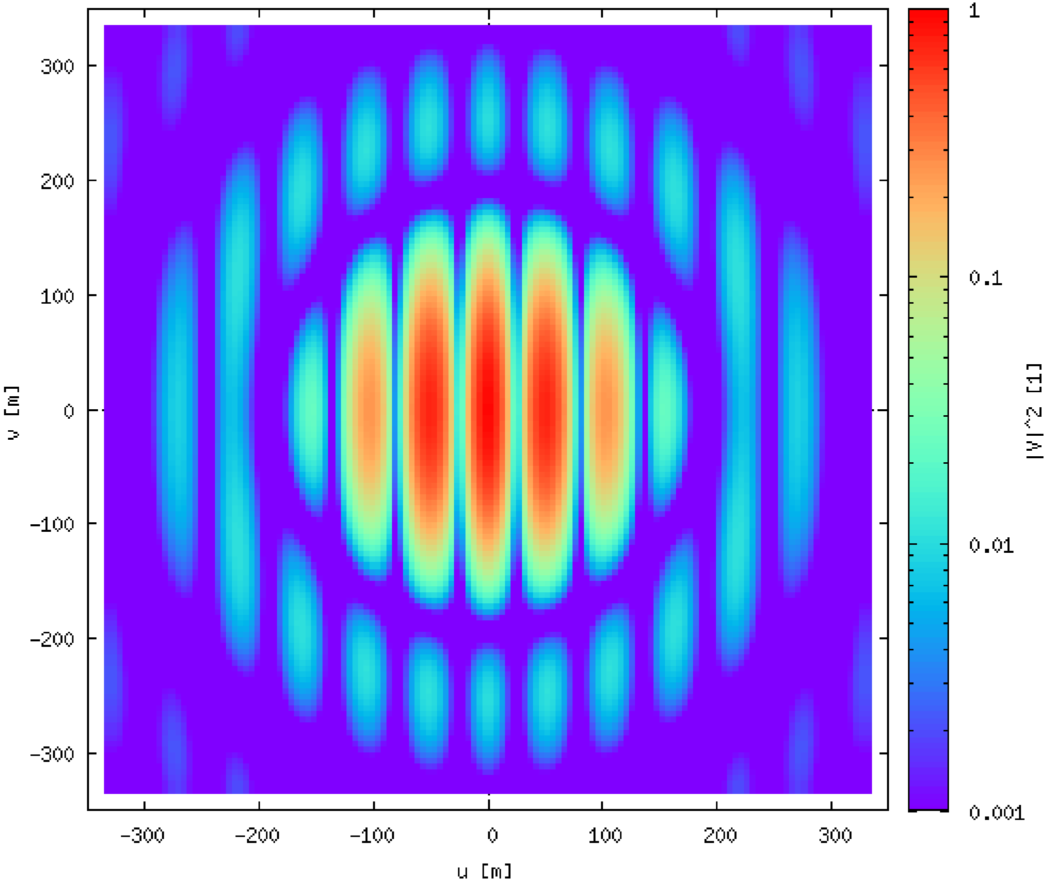


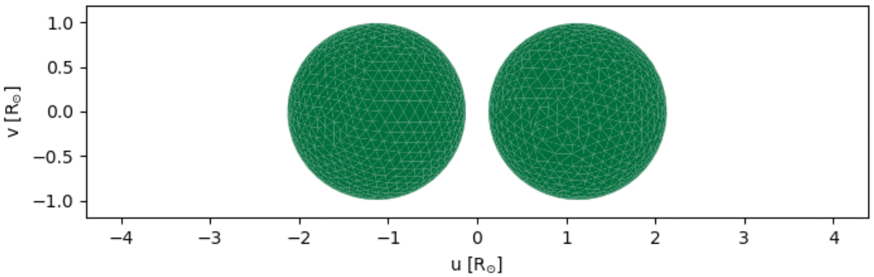

```
Out[23]: (<autofig.figure.Figure | 1 axes | 10 call(s)>,
 <Figure size 576x432 with 1 Axes>)
```

If more baselines were set up, the spatial Fourier transform would be seen more clearly:

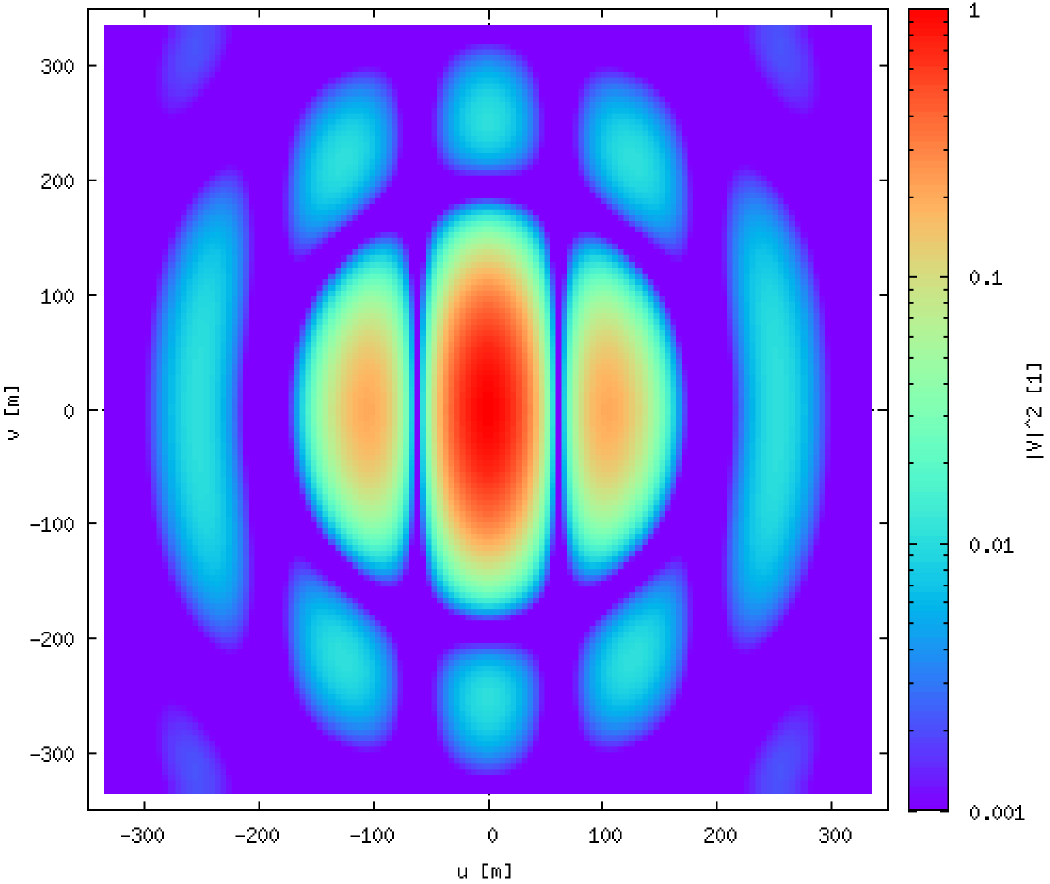



1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]

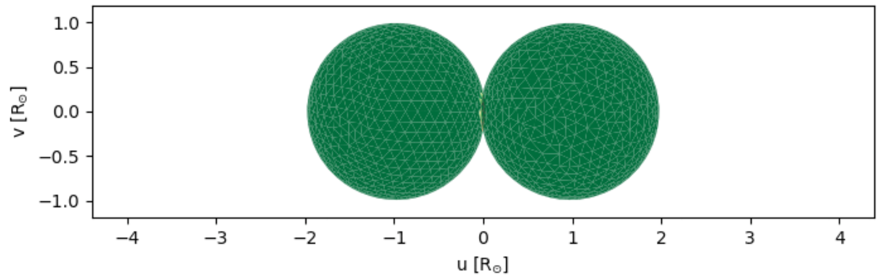

300
200
100
0
-100
-200
-300
v [m]
-300
-200
-100
0
100
200
300
u [m]
1
0.1
0.01
0.001
|V|^2 [1]

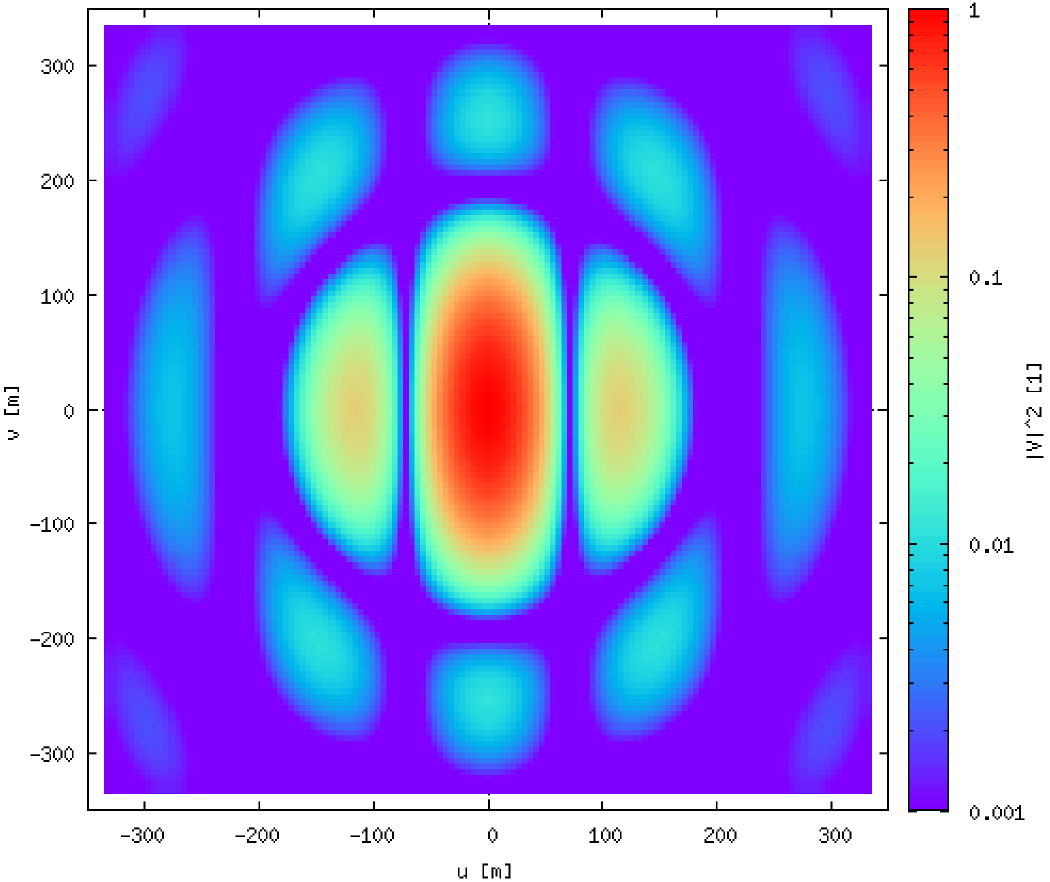


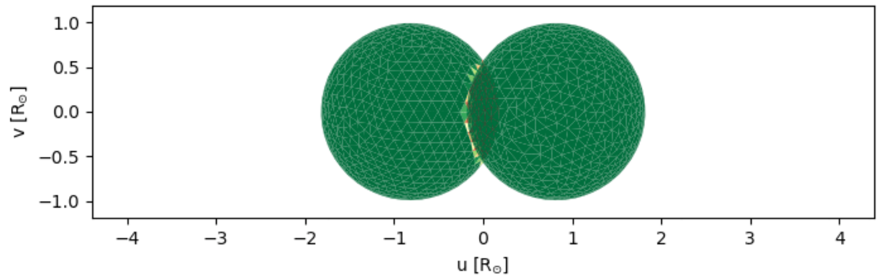
v [R⊙]
u [R⊙]



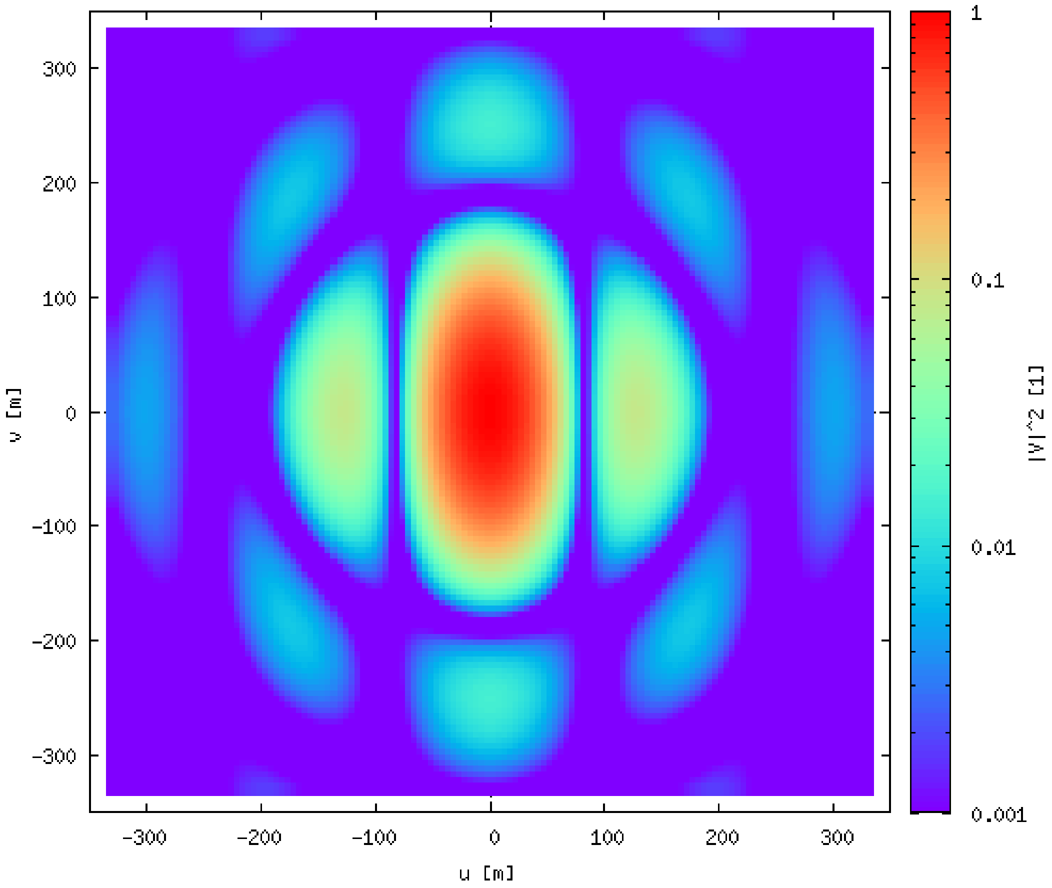
v [m]
u [m]
|V|^2 [1]


1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]

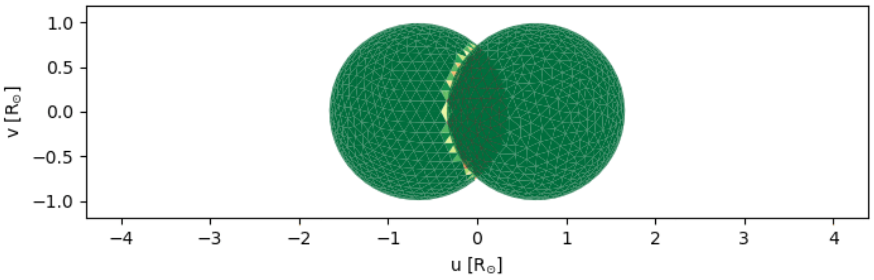

300
200
100
0
−100
−200
−300
v [m]
−300
−200
−100
0
100
200
300
u [m]
1
0.1
0.01
0.001
|V|^2 [1]



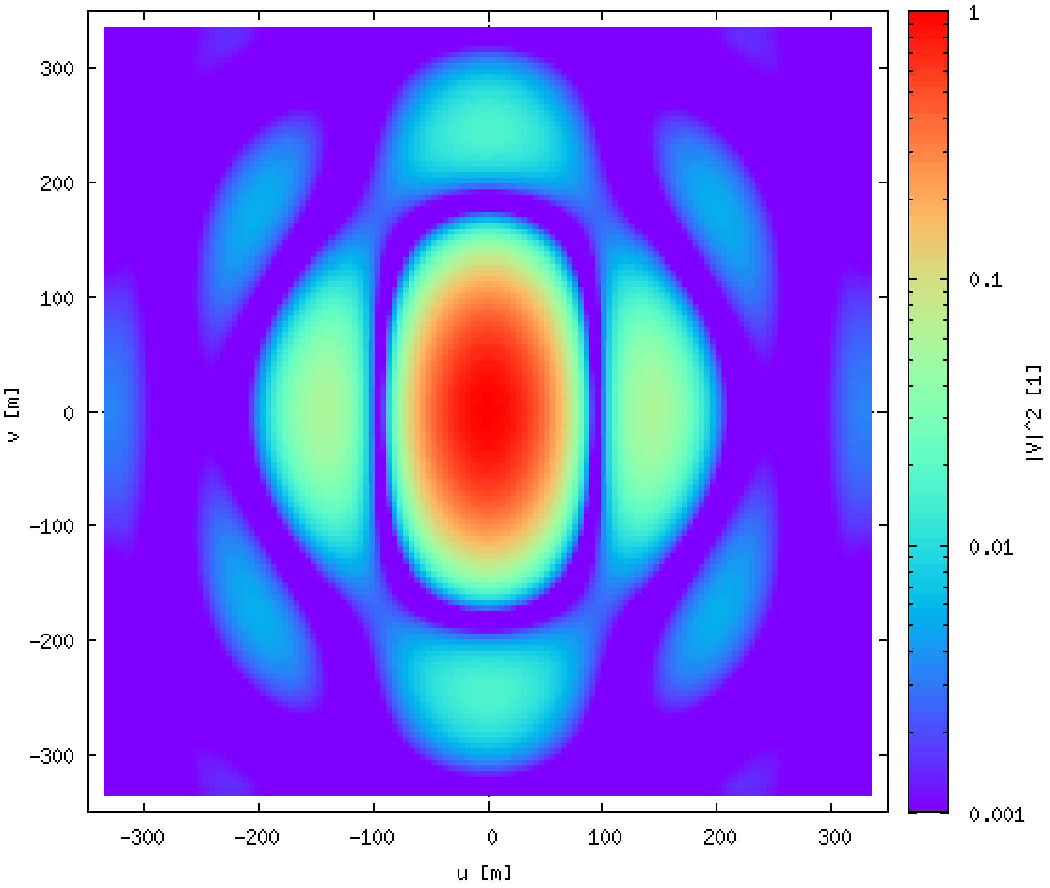
1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]



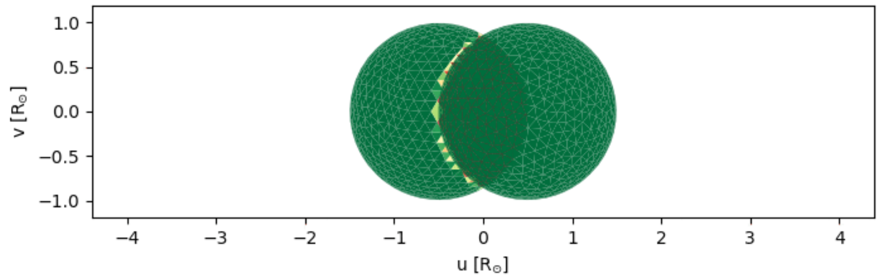
300
200
100
0
−100
−200
−300
v [m]
−300
−200
−100
0
100
200
300
u [m]
1
0.1
0.01
0.001
|V|^2 [1]

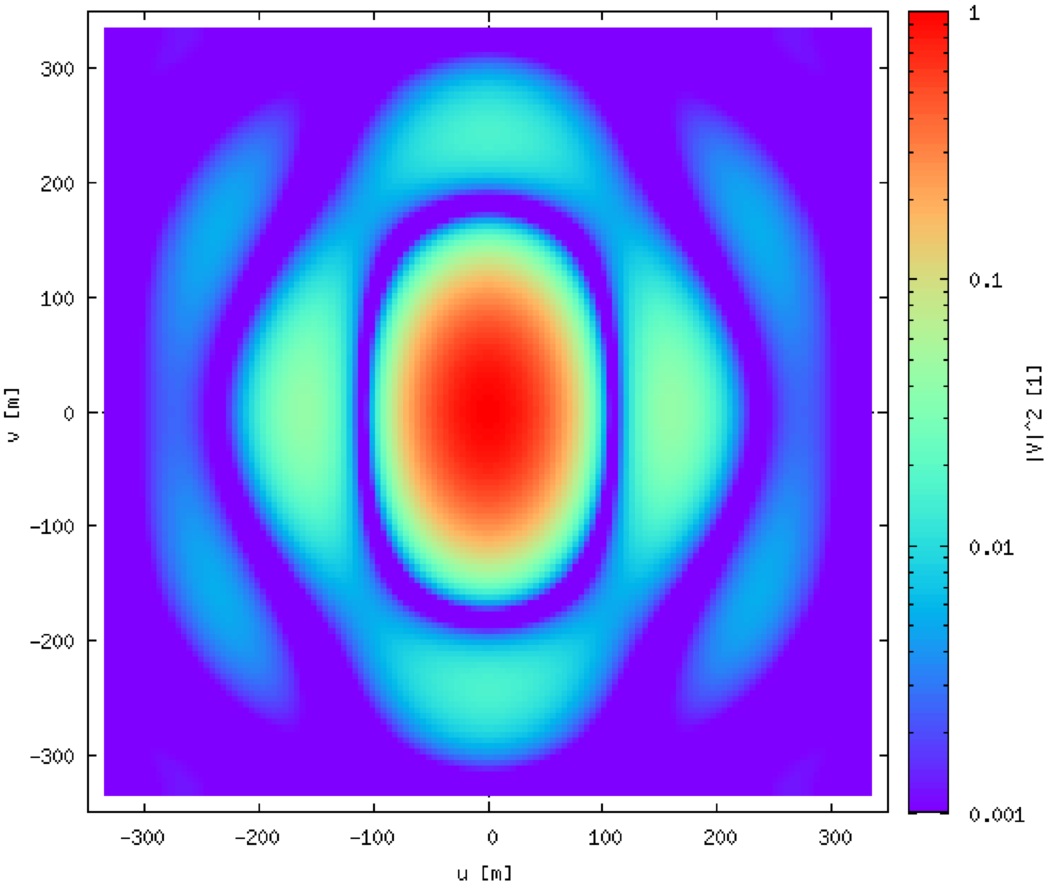


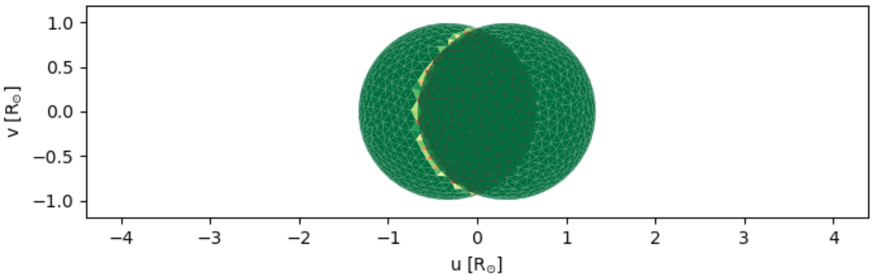
v [R⊙]
u [R⊙]



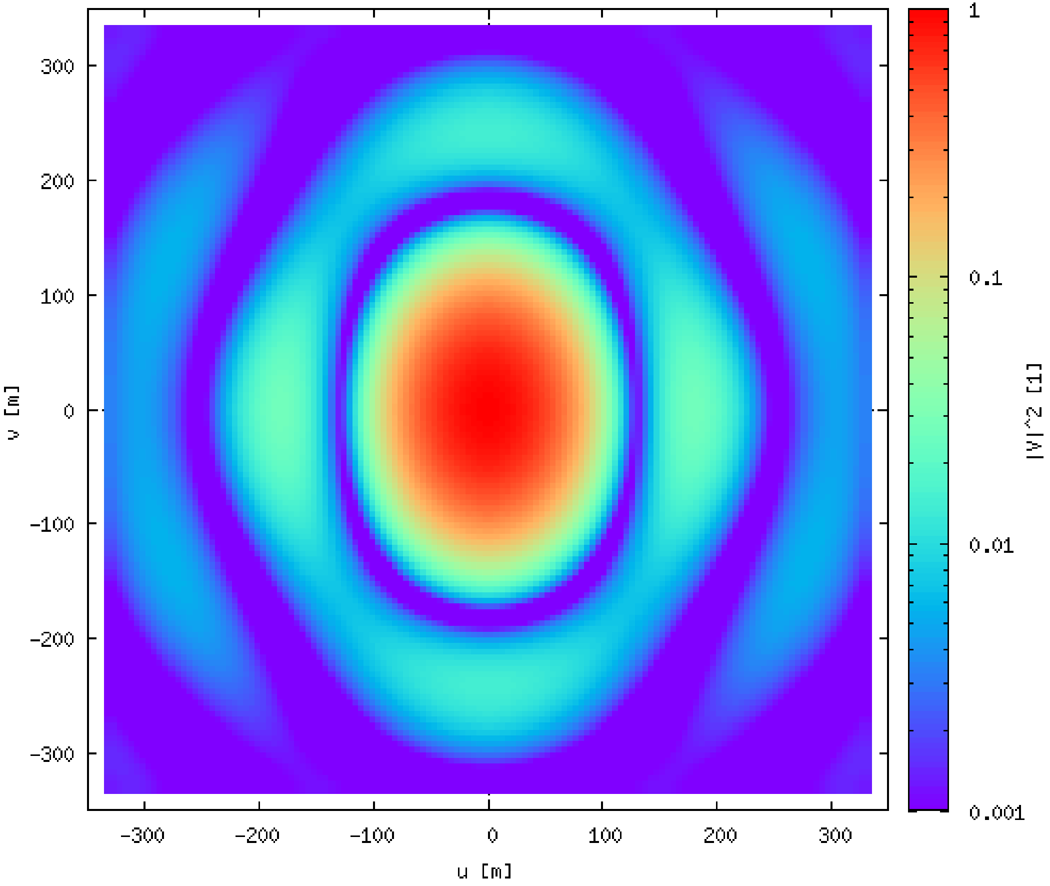
v [m]
u [m]
|V|^2 [1]


1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]

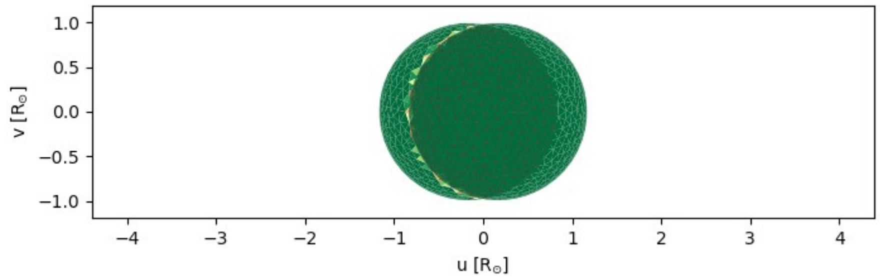

300
200
100
0
−100
−200
−300
v [m]
−300
−200
−100
0
100
200
300
u [m]
1
0.1
0.01
0.001
|V|^2 [1]



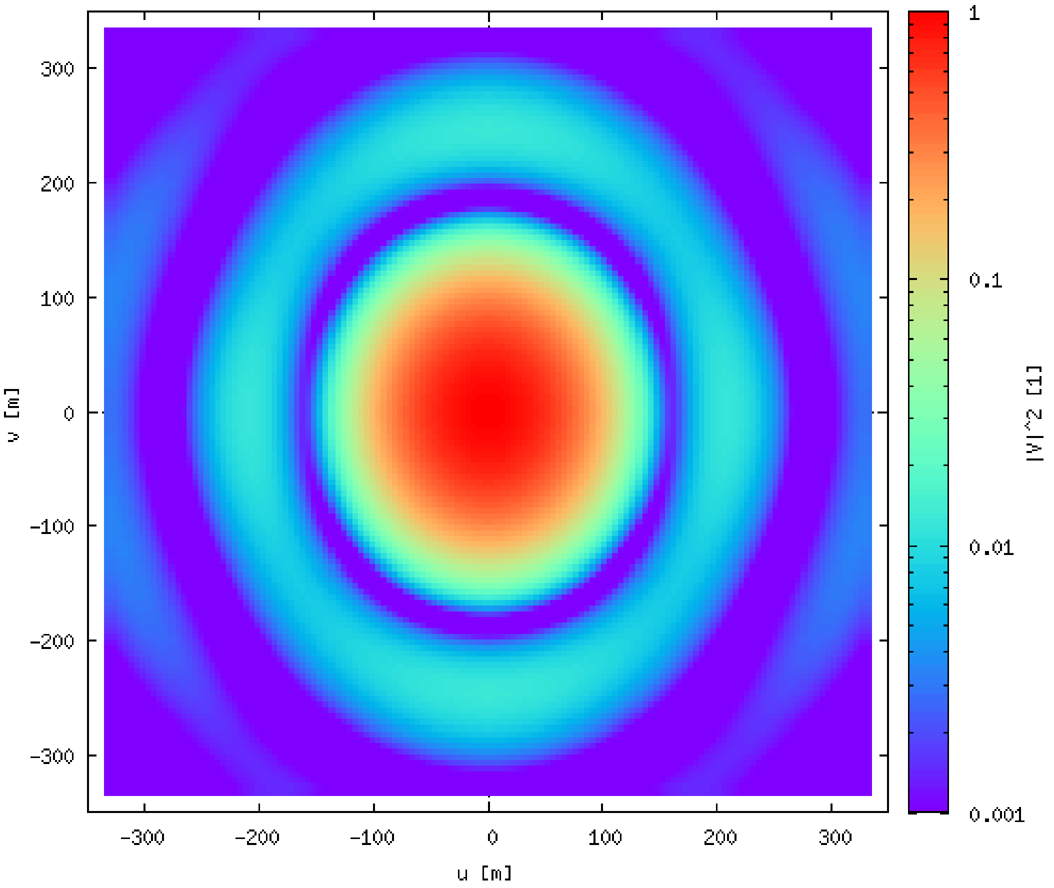
1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]



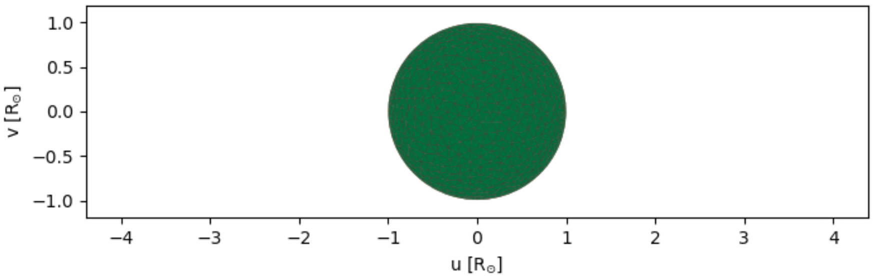
300
200
100
0
−100
−200
−300
v [m]
−300
−200
−100
0
100
200
300
u [m]
1
0.1
0.01
0.001
|V|^2 [1]

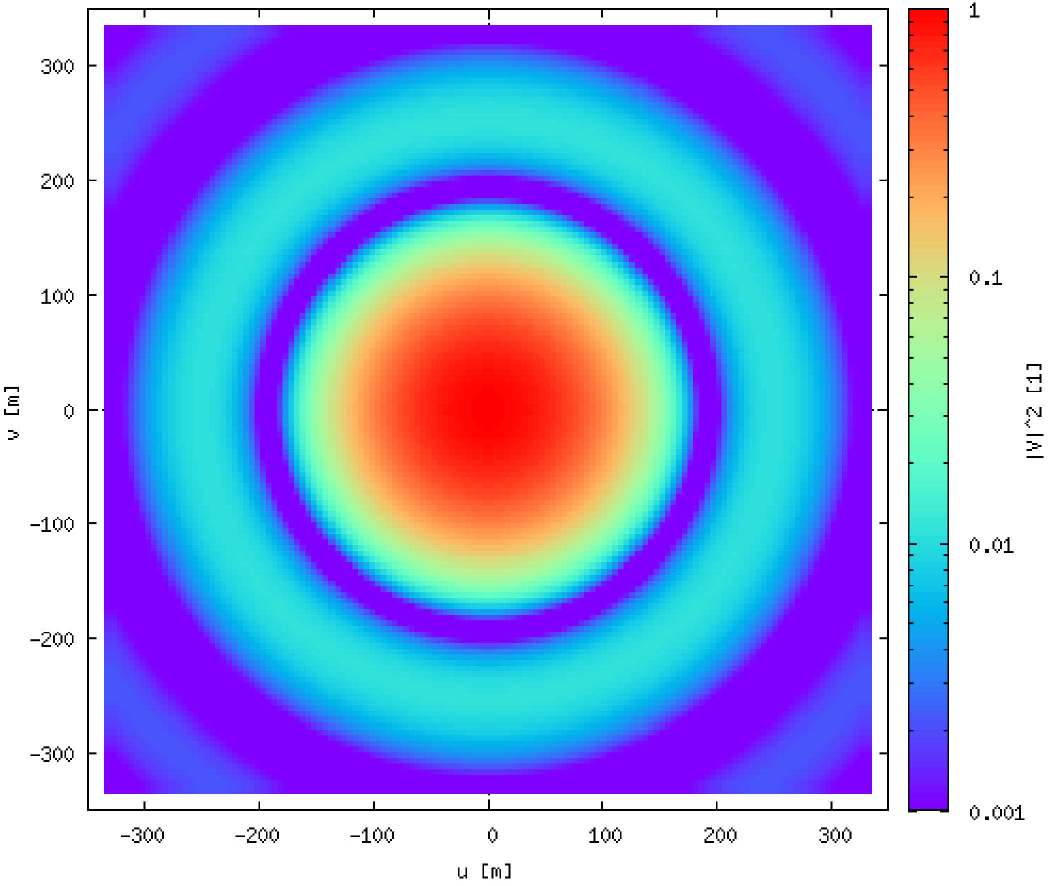


1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]



300
200
100
0
−100
−200
−300
v [m]
−300
−200
−100
0
100
200
300
u [m]
1
0.1
0.01
0.001
|V|^2 [1]


1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]


300
200
100
0
−100
−200
−300
v [m]
−300
−200
−100
0
100
200
300
u [m]
1
0.1
0.01
0.001
|V|^2 [1]

```python
def complexvis_integrate(b, system, ucoord=None, vcoord=None, wavelengths=None, info={}):
    """
    Compute interferometric complex visibility mu.
    A complex model w. integration over meshes.

    b           .. Bundle object
    system      .. System object
    ucoord      .. baselines [m]
    vcoord      .. baselines [m]
    wavelengths .. wavelengths [m]
    info        .. dictionary w. 'original_index' to get (u, v)'s

    Note: Applicable to contact or eclipsing binaries.
      ___   ___          ___ ___
     /   \__/   \        /   /   \
    |     __     |  or  |   |     |
     \___/  \___/        \___\___/

    """

    meshes = system.meshes
    components = info['component']
    dataset = info['dataset']

    visibilities = meshes.get_column_flat('visibilities', components)

    if np.all(visibilities==0):
        return {'complexvis': np.nan}

    # Note: intensity should be per-wavelength!
    abs_intensities = meshes.get_column_flat('abs_intensities:{}'.format(dataset), components)
    mus = meshes.get_column_flat('mus', components)
    areas = meshes.get_column_flat('areas_si', components)
```

```
j = info['original_index']
d = system.distance                    # m
u = ucoord[j]                          # m
v = vcoord[j]                          # m
lambda_ = wavelengths[j]               # m

d *= (units.m/units.solRad).to('1')                       # solRad
centers = meshes.get_column_flat('centers', components)   # solRad
xs = centers[:,0]                                         # solRad
ys = centers[:,1]                                         # solRad
x = xs/d                                                  # rad
y = ys/d                                                  # rad
u /= lambda_                                              # cycles per baseline
v /= lambda_                                              # cycles per baseline

Lum = abs_intensities*areas*mus*visibilities              # J s^-1 m^-1
mu = np.exp(-2.0*np.pi*(0.0+1.0j) * (u*x + v*y))          # 1
mu *= Lum

mutot = np.sum(mu)
Lumtot = np.sum(Lum)

val = mutot/Lumtot                                        # 1

return {'complexvis': val}
```

Phoebe spectroscopic module

- 2 new datasets: SPE, SED
- relative/absolute spectrum == integration over surface(s)
- a 'miniaturised' version of Pyterpolmini (Nemravová et al. 2016)
- *per-triangle* synthetic spectra (T, log g, v sin i, Z)
- a.k.a. Doppler tomography
- alternatively, see Abdul-Masih et al. (2020)
- tutorials...

- https://github.com/miroslavbroz/phœbe2/tree/spectroscopy

'spe' (Spectroscopic) Dataset

Setup

As a preparatory task, we create a file `Spe.dat`, containing common spectroscopic data. Apart from times, we also need wavelengths. For simplicity, the "observed" flux was set to 1.0. Of course, in real life this value is observed (between 0 and 1), as read from FITS files.

In [1]:
```
f = open("Spe.dat", "w")
f.write("# time wavelength flux sigma\n")

t = 0.25            # d
wave1 = 6500.0e-10  # m
wave2 = 6600.0e-10  # m
dwave = 1.0e-10     # m
flux = 1.0          # 1
sigma = 0.01        # 1

wave = wave1
while wave < wave2+0.5*dwave:
    f.write("%.8f  %.8e  %.8f  %.8f\n" % (t, wave, flux, sigma))
    wave += dwave

f.close()
```

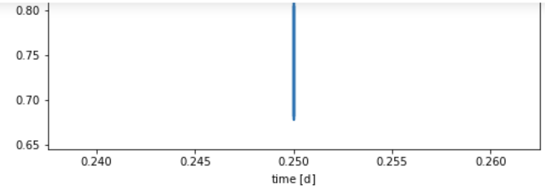
Modelling of SED requires a grid of synthetic spectra (usually, >>10 GB). Here, a tiny 'test' grid will be used, so that this tutorial works without any installation; it is set up as:

In [15]:
```
from phoebe.backend import spectroscopy
from phoebe.backend import pyterpolmini

pyterpolmini.grid_directory = 'grids'
pyterpolmini.grid_dict = dict(
    identification=[
        'test',
        ],
    directories=[
        ['TEST'],
        ],
    families=[
        ['test'],
        ],
    columns=[
        ['filename', 'teff', 'logg', 'z'],
        ],
    )

spectroscopy.sg = pyterpolmini.SyntheticGrid(mode='test', flux_type='relative')
```

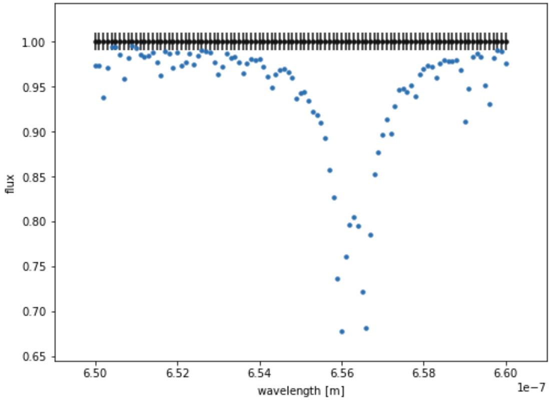
Note: Make sure to have the latest version of PHOEBE 2.5 installed (uncomment this line if running in an online notebook session such as colab).



Out[30]:
```
(<autofig.figure.Figure | 1 axes | 2 call(s)>,
 <Figure size 576x432 with 1 Axes>)
```

Well, this was for a single time, wasn't it? However, it is much more interesting to see the dependence on wavelengths!

In [31]:
```
b.plot(x='wavelengths', marker='.', linestyle='none', show=True)
```

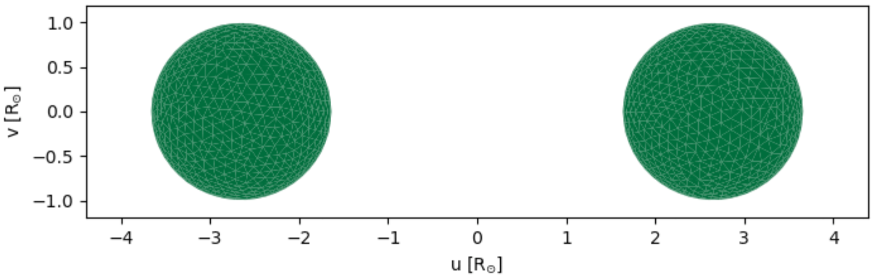


Out[31]:
```
(<autofig.figure.Figure | 1 axes | 2 call(s)>,
 <Figure size 576x432 with 1 Axes>)
```

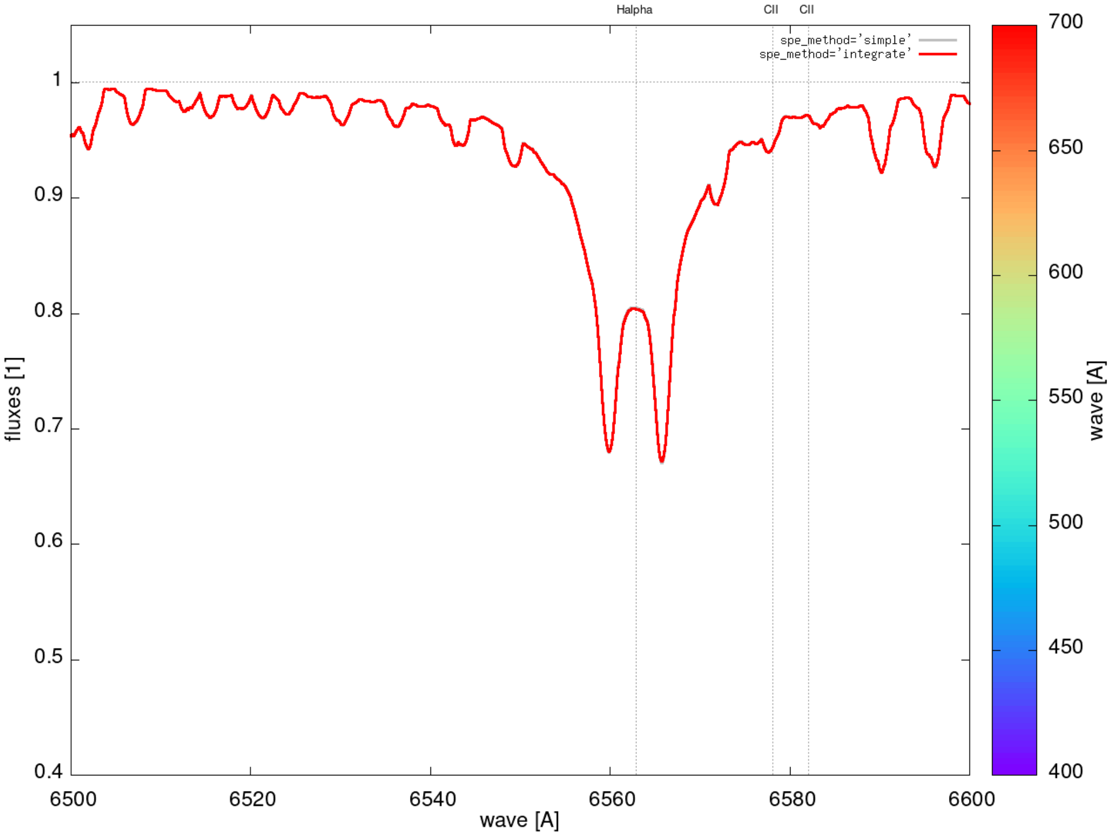
Halpha line is sensitive to teff, logg, rvs of binary components. In the model, an integral over mesh is computed, every triangle has its own synthetic spectrum, so the composite spectrum is influenced by rotation, Roche geometry, von Zeipel, or limb darkening.

Note: A 'miniaturized' version of Pyterpol3 (Nemravová et al. 2016, A&A 594, A55) is used internally to interpolate synthetic spectra. Supported grids include: OSTAR, BSTAR, POLLUX, AMBRE, PHOENIX, or POWR. They must be downloaded (http://sirrah.troja.mff.cuni.cz/~mira/xitau/) and present in the `grids` directory.

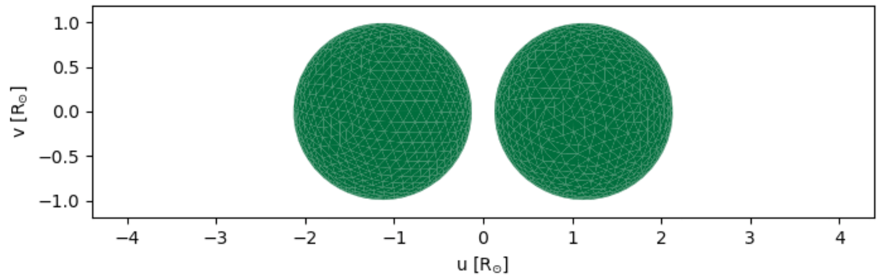
In [ ]:



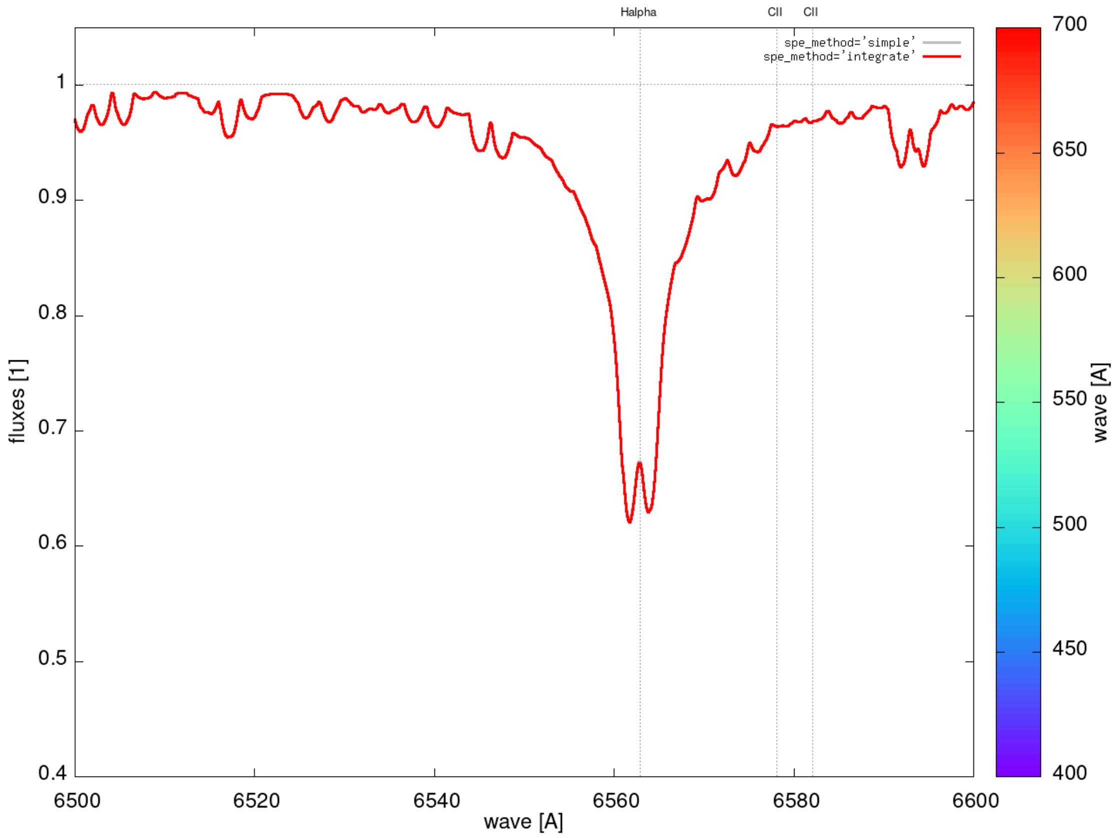
v [R⊙]
u [R⊙]



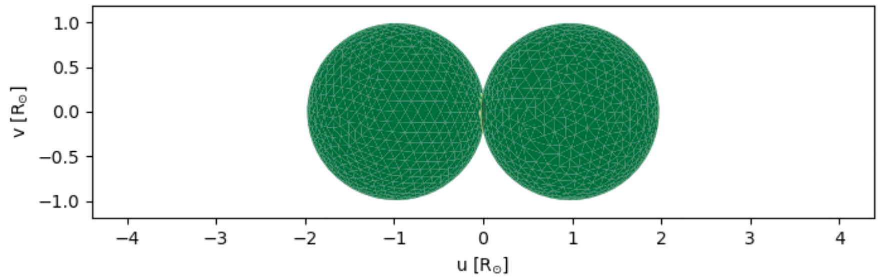
Halpha
CII
CII
spe_method='simple'
spe_method='integrate'
fluxes [1]
wave [A]
wave [A]



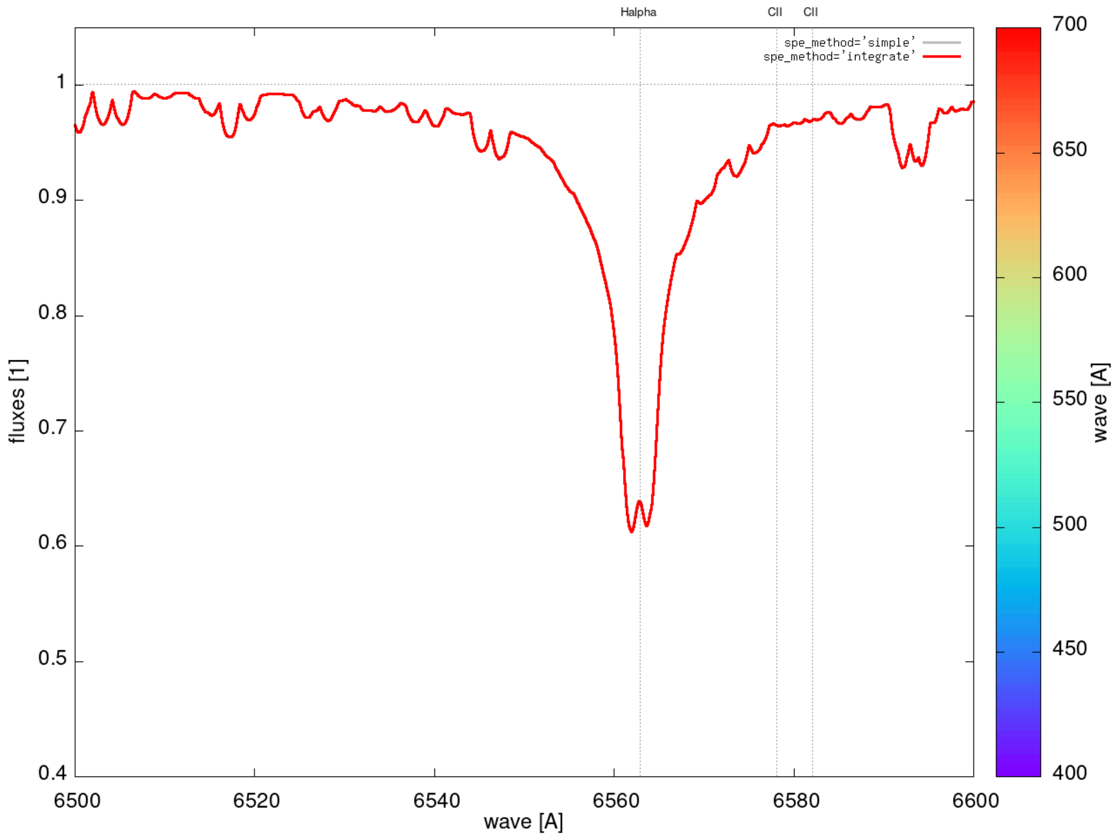
v [R⊙]
u [R⊙]



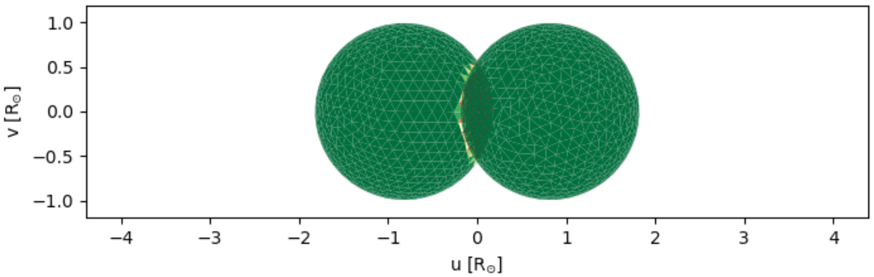
Halpha
CII
CII
spe_method='simple'
spe_method='integrate'
fluxes [1]
wave [A]
wave [A]



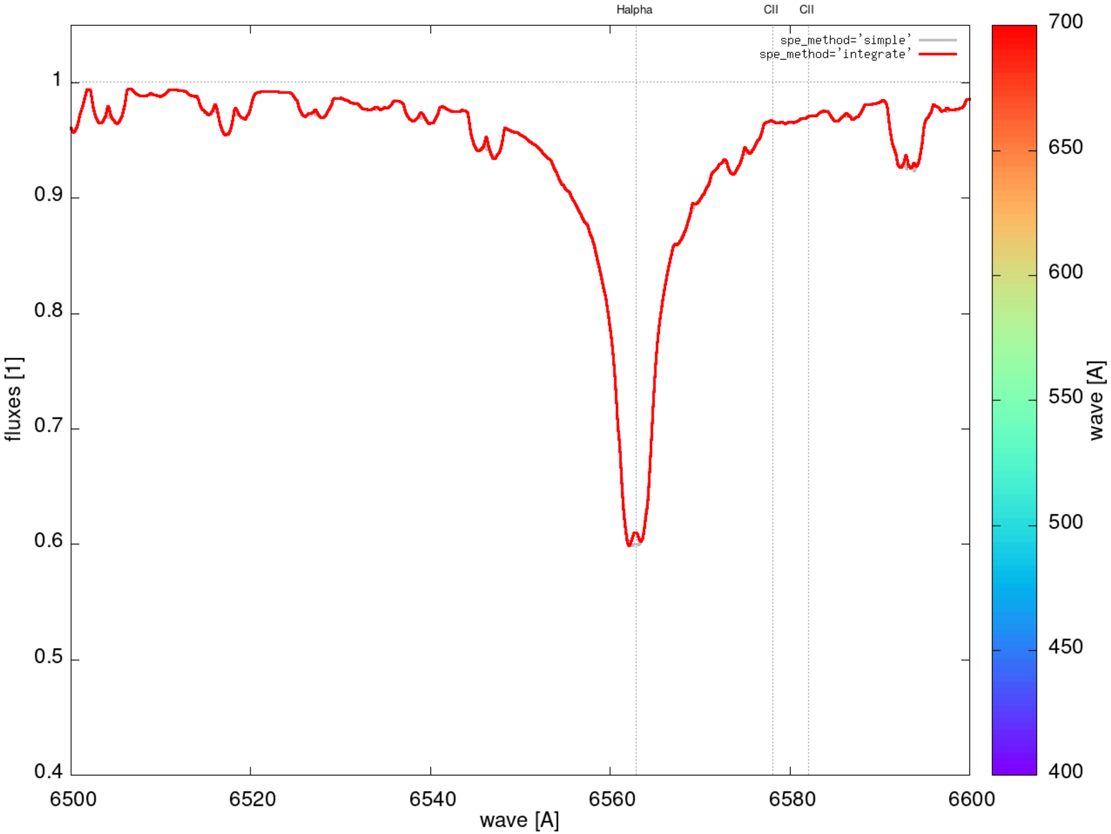
v [R⊙]
u [R⊙]



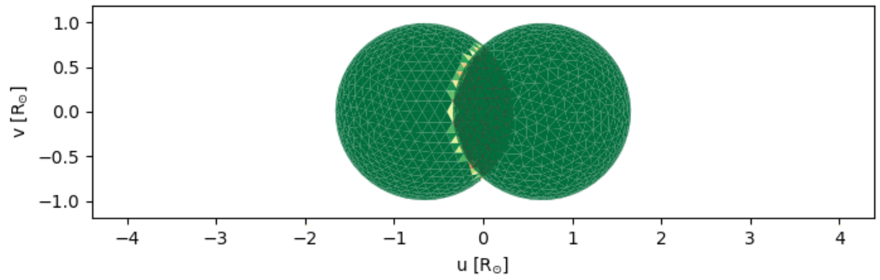
Halpha
CII
CII
spe_method='simple'
spe_method='integrate'
fluxes [1]
wave [A]
wave [A]



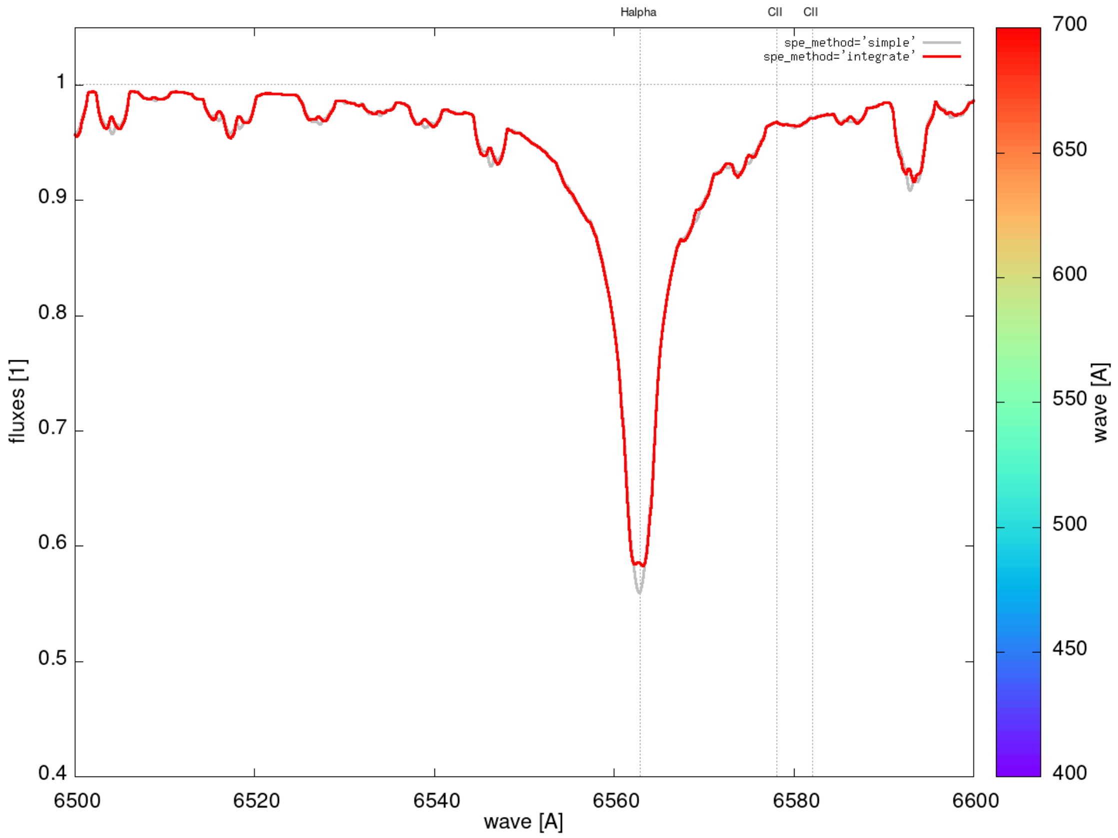
1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]



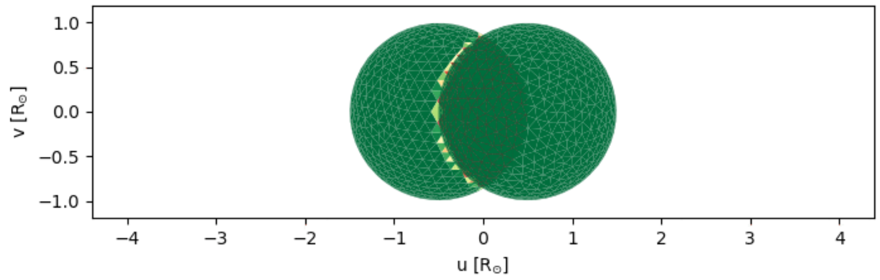
Halpha
CII
CII
spe_method='simple'
spe_method='integrate'
fluxes [1]
1
0.9
0.8
0.7
0.6
0.5
0.4
6500
6520
6540
6560
6580
6600
wave [A]
700
650
600
550
500
450
400
wave [A]



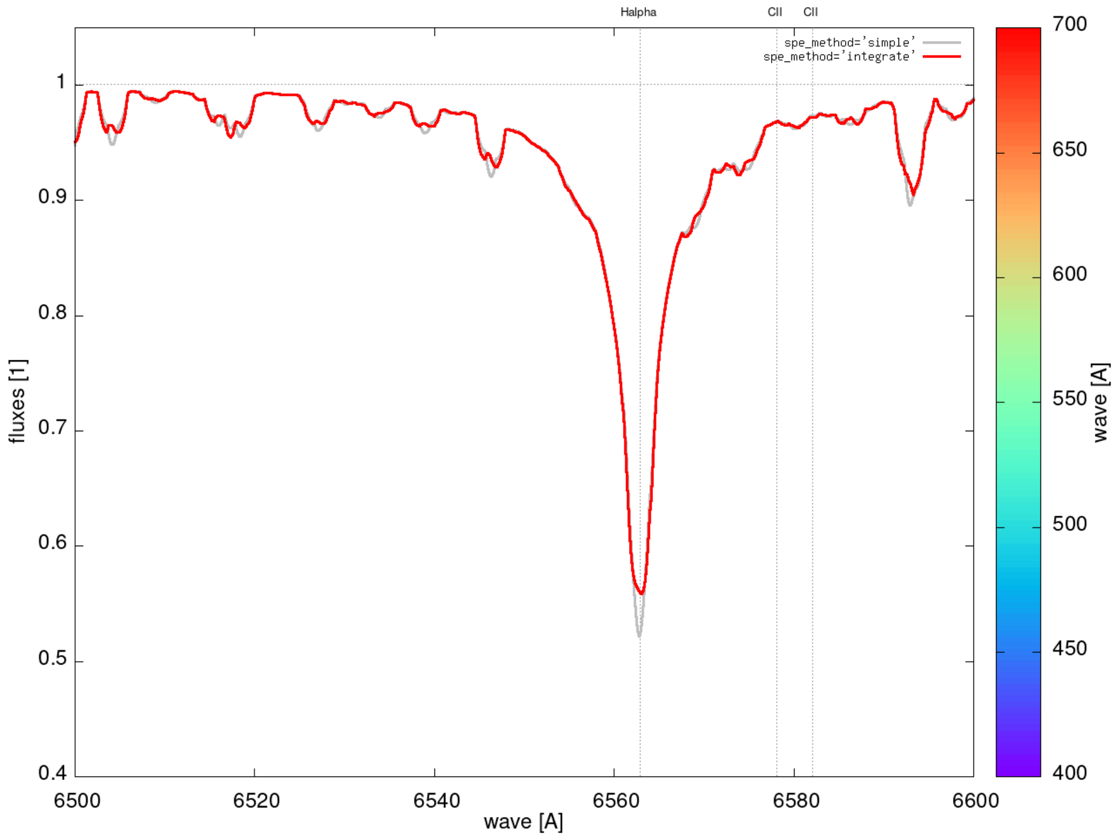
1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]



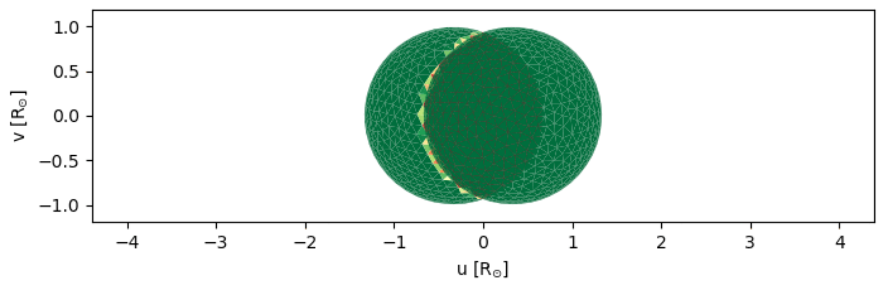
Halpha
CII
CII
spe_method='simple'
spe_method='integrate'
1
0.9
0.8
0.7
0.6
0.5
0.4
fluxes [1]
6500
6520
6540
6560
6580
6600
wave [A]
700
650
600
550
500
450
400
wave [A]

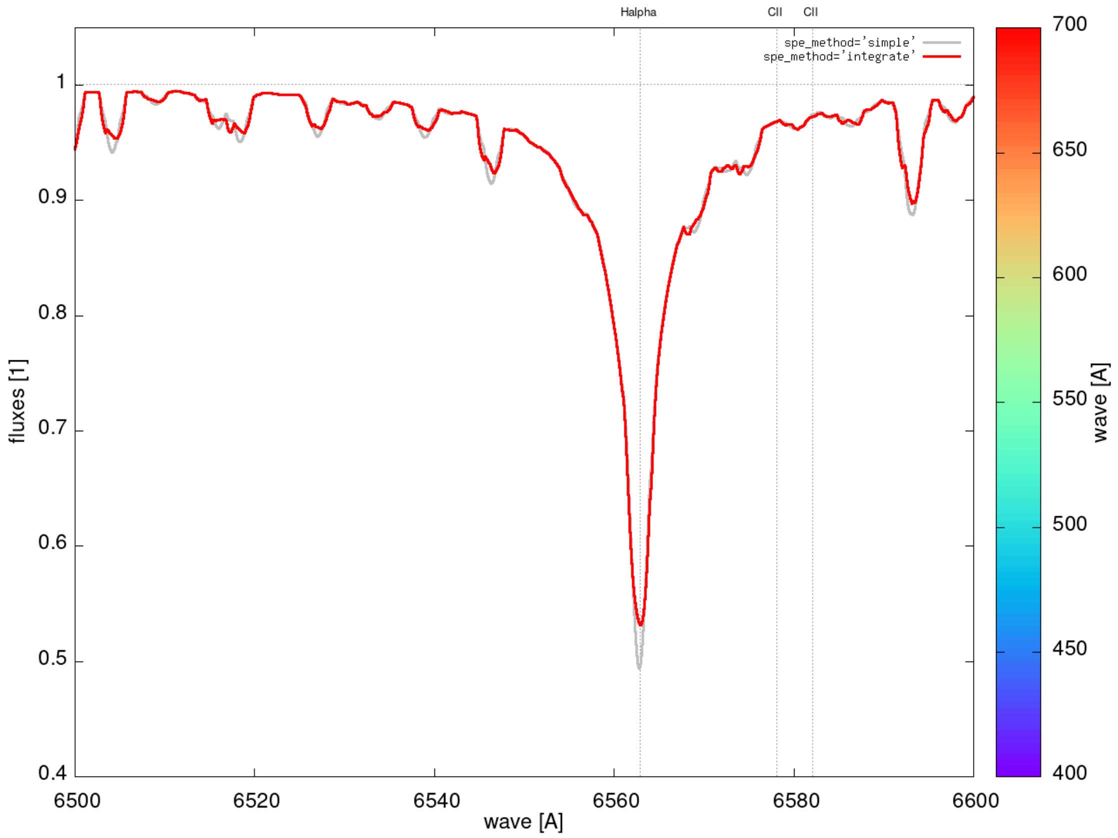


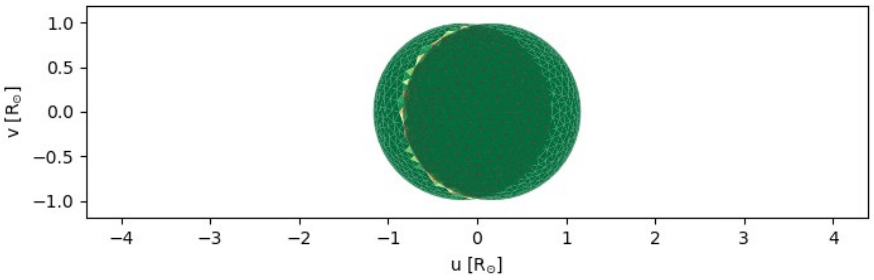
1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]



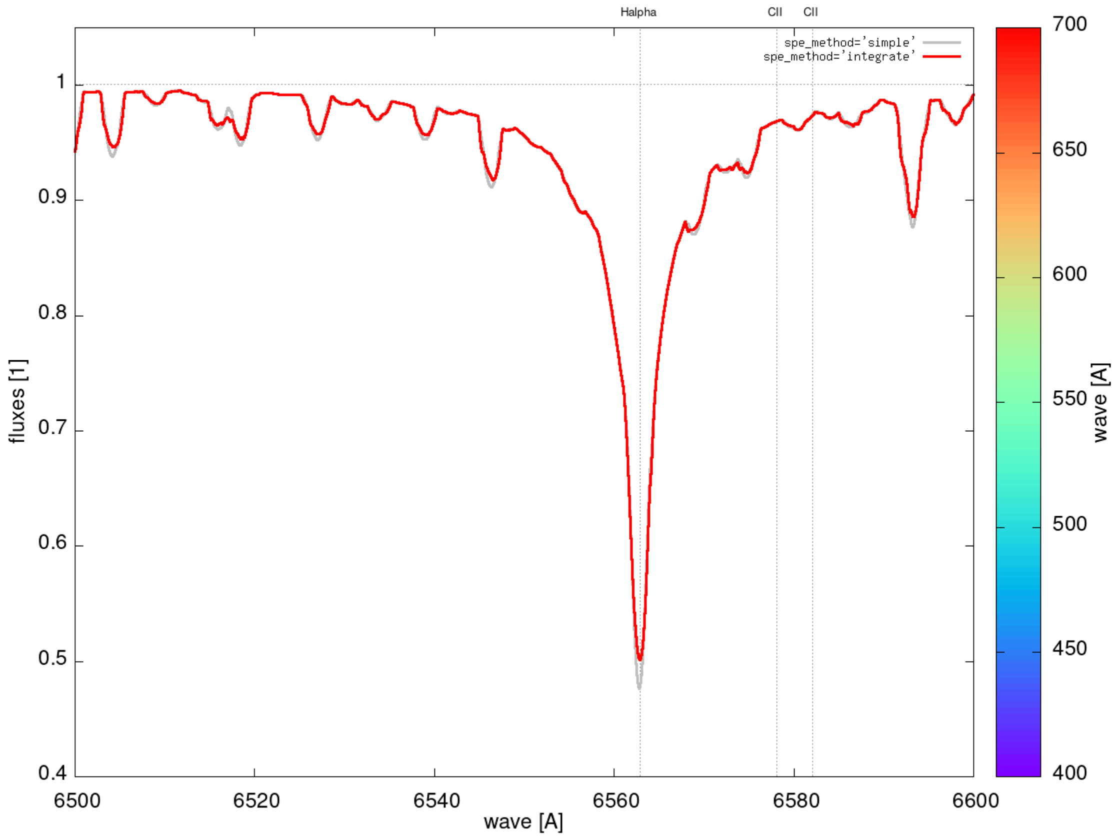
Halpha
CII
CII
spe_method='simple'
spe_method='integrate'
1
0.9
0.8
0.7
0.6
0.5
0.4
fluxes [1]
6500
6520
6540
6560
6580
6600
wave [A]
700
650
600
550
500
450
400
wave [A]


1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]

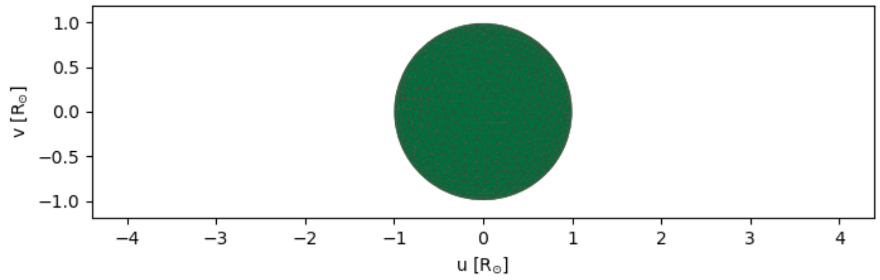

Halpha
CII
CII
spe_method='simple'
spe_method='integrate'
1
0.9
0.8
0.7
0.6
0.5
0.4
fluxes [1]
6500
6520
6540
6560
6580
6600
wave [A]
700
650
600
550
500
450
400
wave [A]



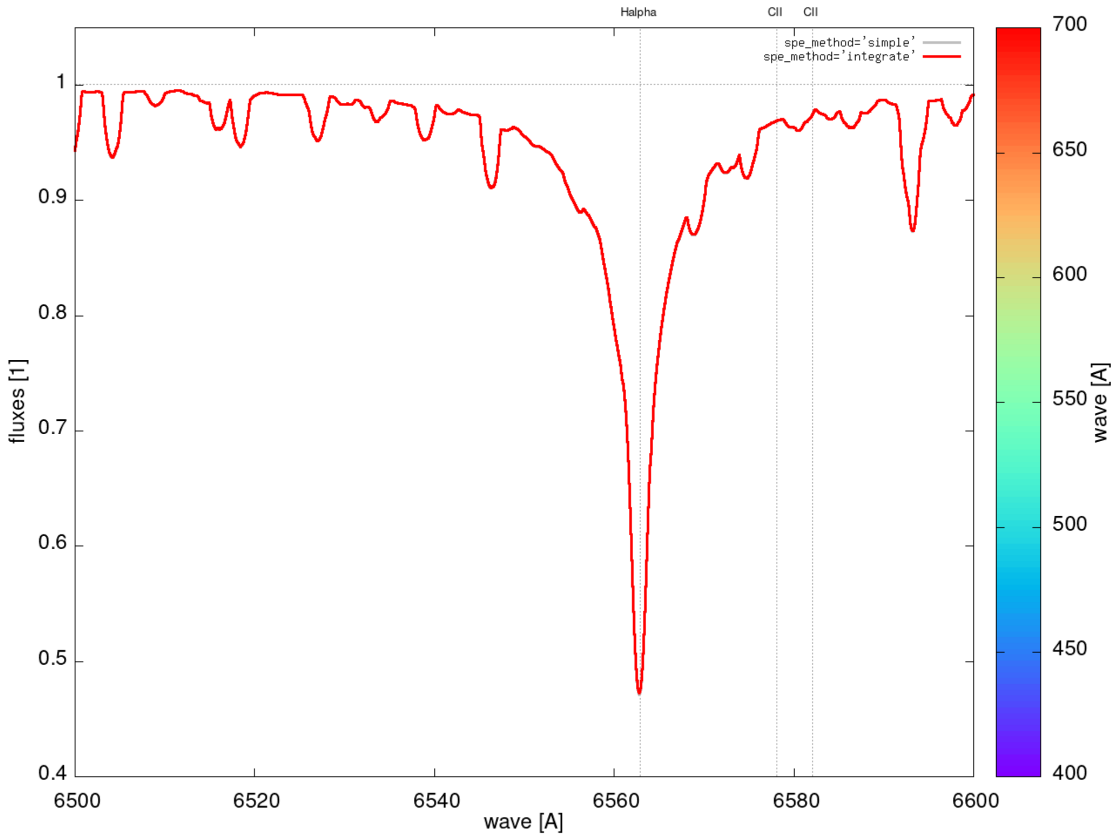
1.0
0.5
0.0
−0.5
−1.0
v [R⊙]
−4
−3
−2
−1
0
1
2
3
4
u [R⊙]



Halpha
CII
CII
spe_method='simple'
spe_method='integrate'
fluxes [1]
1
0.9
0.8
0.7
0.6
0.5
0.4
6500
6520
6540
6560
6580
6600
wave [A]
700
650
600
550
500
450
400
wave [A]



v [R⊙]
u [R⊙]



Halpha
CII
CII
spe_method='simple'
spe_method='integrate'
fluxes [1]
wave [A]
wave [A]

```python
def spe_integrate(b, system, wavelengths=None, info={}, k=None):
    """
    Compute relative monochromatic flux F_nu.
    A complex model w. integration over meshes.

    b           .. Bundle object
    system      .. System object
    wavelengths .. wavelengths [m]
    info        .. dictionary w. 'original_index' to get wavelengths
    k           .. index to run computation

    Note: All wavelengths are computed at once, but returned sequentially 0, 1, 2, ...

    Note: Applicable to contact or eclipsing binaries.

    """
    global sg
    global fluxes

    if sg is None:
        sg = pyterpolmini.SyntheticGrid(flux_type='relative', debug=False)

    j = info['original_index']
    if k > 0:
        return {'flux': fluxes[j]}

    meshes = system.meshes
    components = info['component']
    dataset = info['dataset']

    visibilities = meshes.get_column_flat('visibilities', components)

    if np.all(visibilities==0):
        return {'flux': np.nan}
```

```
# Note: intensity should be per-wavelength!
abs_intensities = meshes.get_column_flat('abs_intensities:{}'.format(dataset), components)
mus = meshes.get_column_flat('mus', components)
areas = meshes.get_column_flat('areas_si', components)
rvs = (meshes.get_column_flat("rvs:{}".format(dataset), components)*u.solRad/u.d).to(u.m/u.s).value
teffs = meshes.get_column_flat('teffs', components)
loggs = meshes.get_column_flat('loggs', components)
zs = 10.0**meshes.get_column_flat('abuns', components)

Lum = abs_intensities*areas*mus*visibilities        # J s^-1 m^-1

step = 0.01                                         # Ang
angstroms = wavelengths*1.0e10                      # Ang
fluxes = np.zeros(len(wavelengths))                 # 1

for i in range(len(Lum)):
    if Lum[i] == 0.0:
        continue

    props = {'teff': teffs[i], 'logg': loggs[i], 'z': zs[i]}

    s = sg.get_synthetic_spectrum(props, angstroms, order=2, step=step, padding=20.0)

    rv = rvs[i]*1.0e-3                                                      # km/s
    wave_ = pyterpolmini.doppler_shift(s.wave, rv)                          # Ang
    intens_ = pyterpolmini.interpolate_spectrum(wave_, s.intens, angstroms) # 1

    fluxes += Lum[i]*intens_

Lumtot = np.sum(Lum)
fluxes /= Lumtot

return {'flux': fluxes[0]}
```

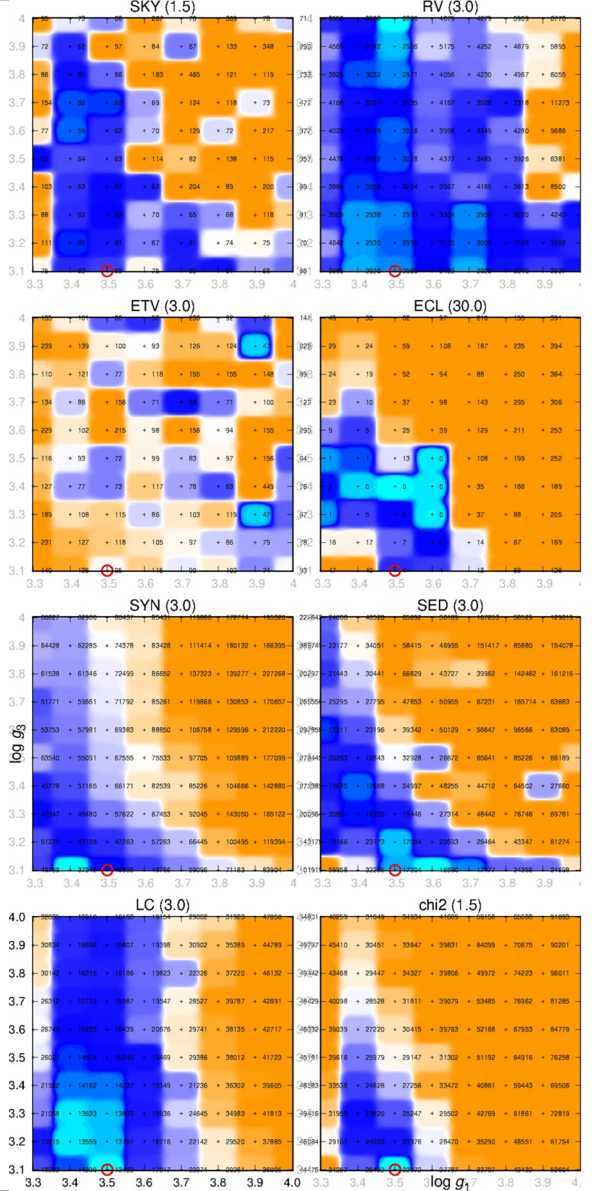
Future work

3. How to combine measurements?

- Oplištilová et al. (2023, A&A 672, A31)
- a joint χ^2 metric: SKY, RV, ETV, ECL, SYN, SED, LC
- **keep track of individual contributions to χ^2!**
- a detection of systematics
- the choice of weights?
- cf. MCMC analysis
- “Back to measurements.”

δ Ori A
best-fit models
convergence
log g_1 vs. log g_3
contributions to χ^2
correlations
orthogonality

SKY
ETV
SYN
LC



RV
ECL
SED
total

best fits
good fits
poor fits

Oplištilová et al. (A&A, in prep.)

- Orion belt == the closest of the most massive *
- δ, ε, ζ, σ Ori
- successful ESO proposal (ID 112.25JX, PI A. Oplištilová, 12 + 1 h)
- **VLTI/GRAVITY (Abuter et al. 2016)**
- VLTI/PIONIER (LeBouquin et al. 2012)
- angular positions and diameters, up to 10-microarcsec precision
- **known distances of faint components from Gaia!**
- a combination w. other types measurements...

Table 3. Information about bright stars and their companions in Orion.

HD	Name	V [mag]	Spectral type	A_V [mag]	V_0 [mag]	M_V	$V_0 - M_V$	*Gaia* DR3 parallax [mas]	Notes
36486	δ Ori	2.22 (*)	O9.5 II	0.13	2.09	–5.81	7.90		OB1b association, multiple
37128	ε Ori	1.68 (*)	B0 Ia	0.14	1.54	–6.25 (**)	7.79		OB1b, single, variable 0.05 mag
37742	ζ Ori	1.75 (*)	O9.5 Ib	0.17	1.58	–6.28 (**)	7.92		OB1b, multiple
37468	σ Ori	3.82 (*)	O9.5 V	0.17	3.65	–4.14	7.79		OB1b, multiple
37043	ι Ori	2.75	O8.5 III	0.09	2.66	–5.13	7.79		OB1d (Trapezium), multiple
36486 Aa1	δ Ori Aa1	2.55	O9.5 II		2.42	–5.7 (**)	8.12		cf. this work
36486 Aa2	δ Ori Aa2	5.5?	B2 V		5.4?	–2.5?			0.00052″ from Aa1, Shenar et al. (2015)
36486 Ab	δ Ori Ab	3.83	B0 IV		3.70	–4.0 (***)	7.70		0.32″
36486 B	δ Ori B	14.0	K?		13.9	+6.6		3.5002±0.0119	33″, UCAC3 180-24383
36485 Ca	δ Ori Ca	6.62	B3 V		6.49	–1.6 (***)	8.09	2.6244±0.0538	52″, helium star, Leone et al. (2010)
36485 Cb	δ Ori Cb	9.8?	A0 V		9.7?	+1.8?			0.0012″ from Ca
37742 Aa	ζ Ori Aa	2.1	O9.5 Ib						Hummel et al. (2000)
37742 Ab	ζ Ori Ab	4.3	B0.5 IV						0.042″
37743	ζ Ori B	4.0	B0 III						2.4″
37742 C	ζ Ori C	9.54	A?					2.5876±0.0387	57″
37468 Aa	σ Ori Aa	4.61	O9.5 V						Simón-Díaz et al. (2015)
37468 Ab	σ Ori Ab	5.20	B0.5 V						0.00042″
37468 B	σ Ori B	5.31	B?						0.25″
37468 C	σ Ori C	8.79	B0.5 V					2.4720±0.0292	11″
37468 D	σ Ori D	6.62	B2 V					2.4744±0.0621	13″
37468 E	σ Ori E	6.66	B2 V					2.3077±0.0646	42″, helium star
37043 Aa1	ι Ori Aa1	2.8?	O8.5 III						Bagnuolo et al. (2001)
37043 Aa2	ι Ori Aa2		B0.8 III						0.0015″, eccentric
37043 Ab	ι Ori Ab		B2 IV						0.15″
37043 B	ι Ori B	7.00	B8 III					2.7869±0.0476	11″
37043 C	ι Ori C	9.76	A0 V					2.6057±0.0241	49″, Parenago (1954), Brun 731

from news.cnrs.fr

(u, v) coverage

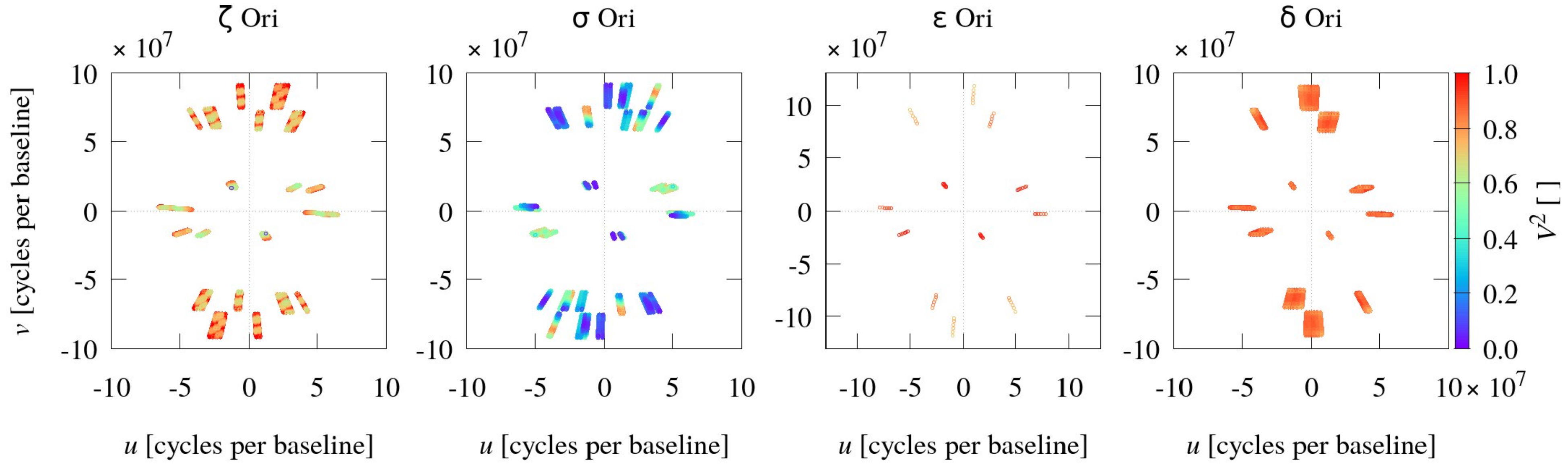
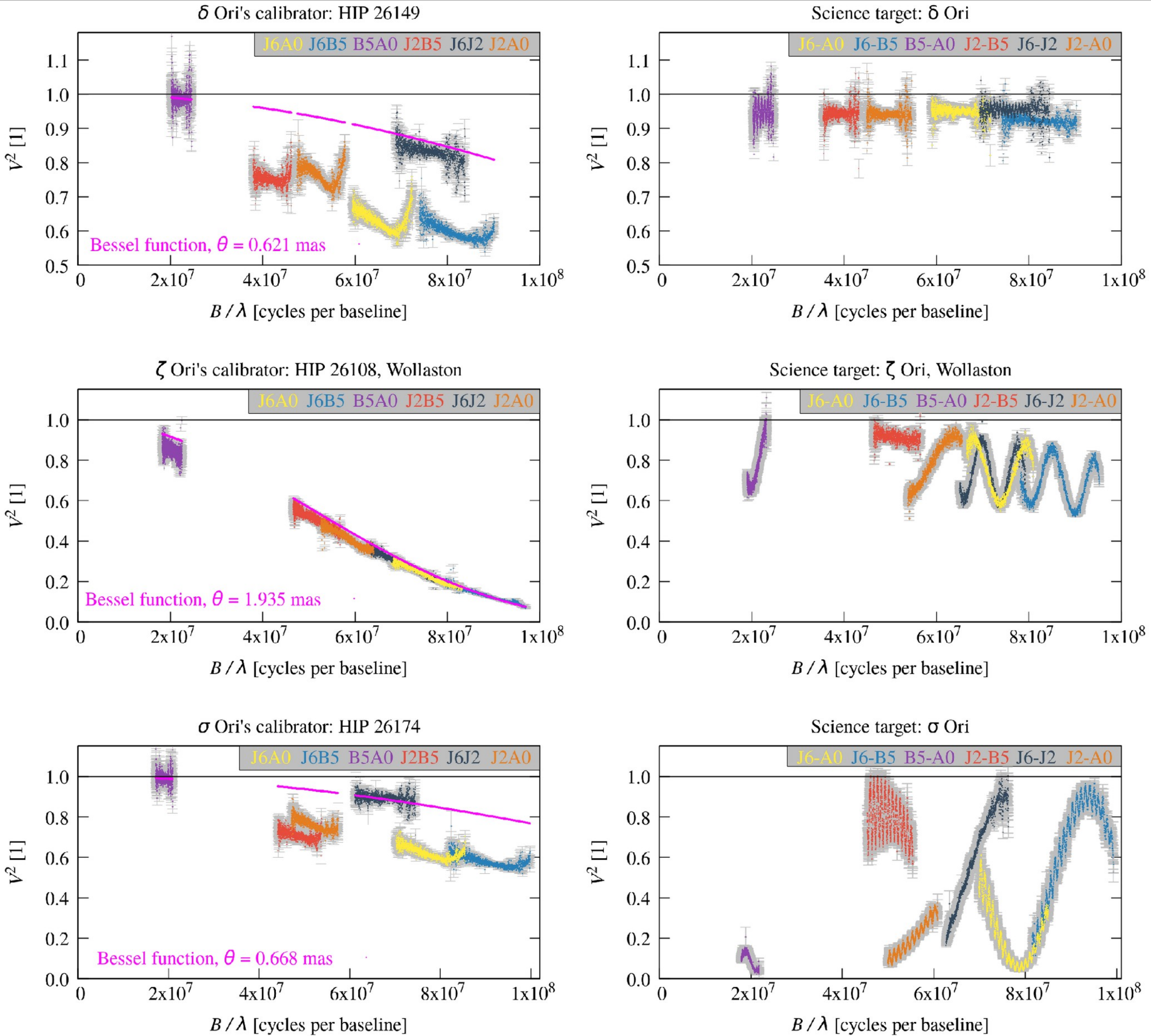


Fig. 4. Coverage in *uv*-planes during all nights.

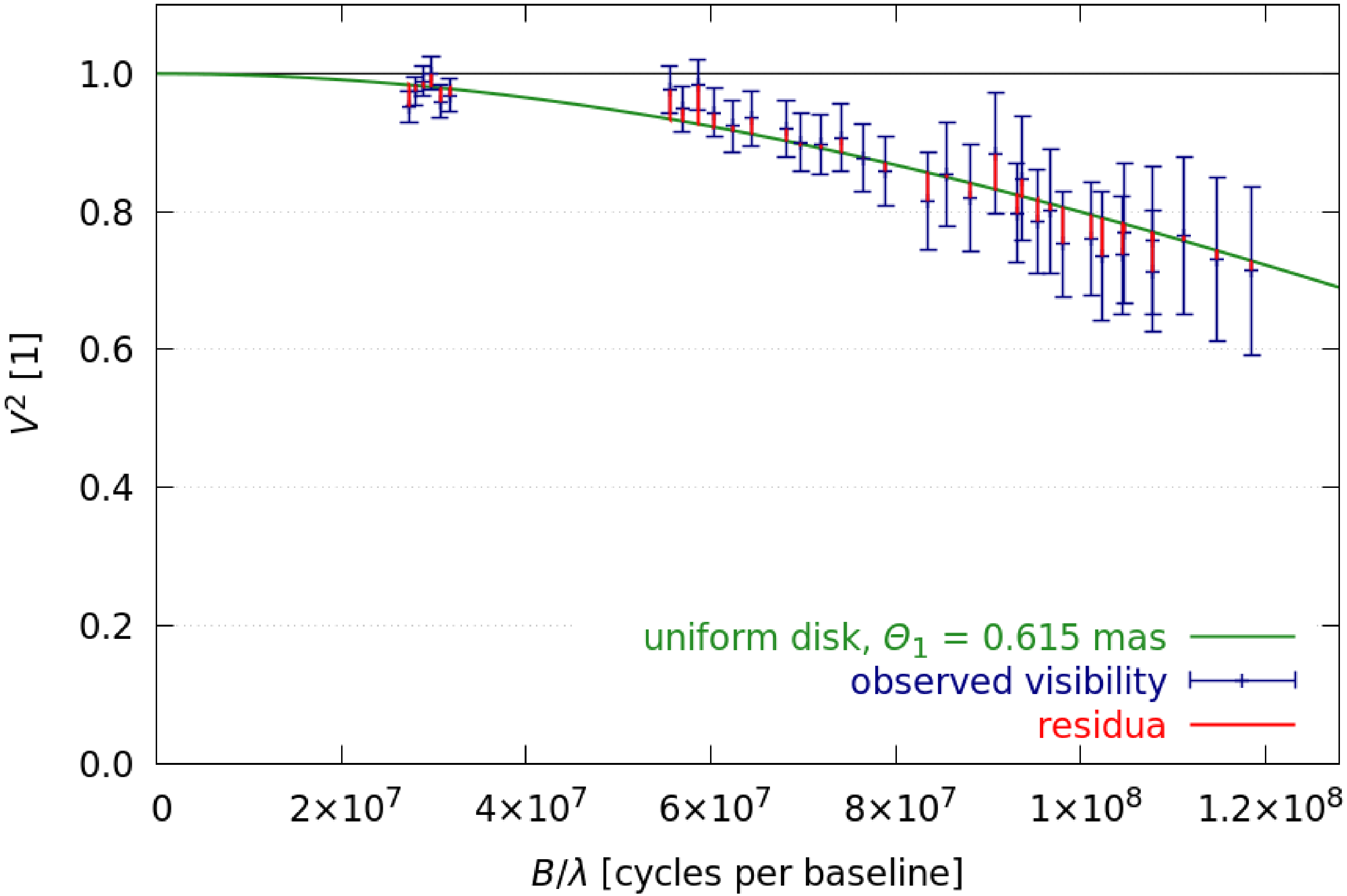
calibration
problems
w. HIP 26108



by A. Opluštilová

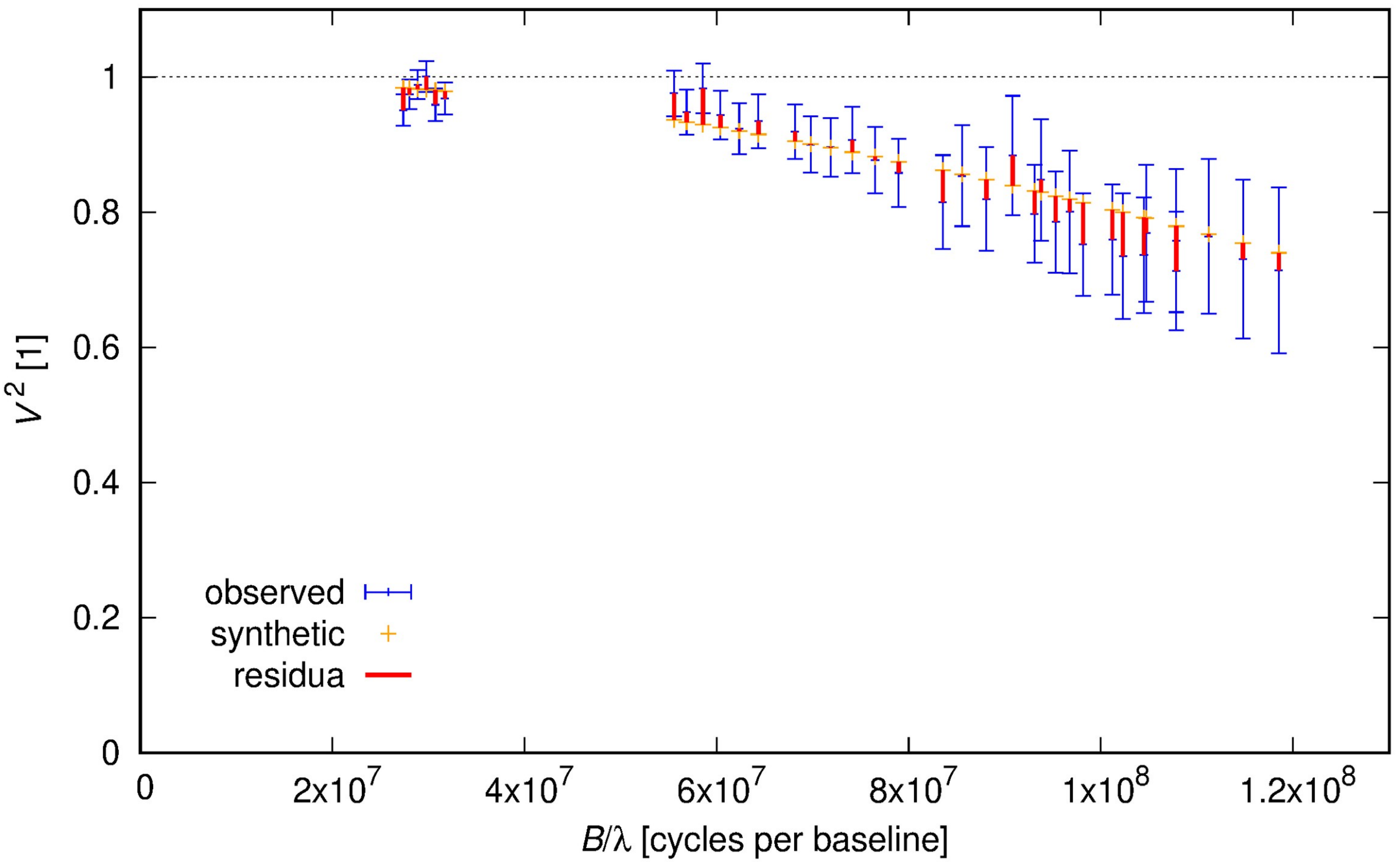
ε Ori

Model (Xitau)



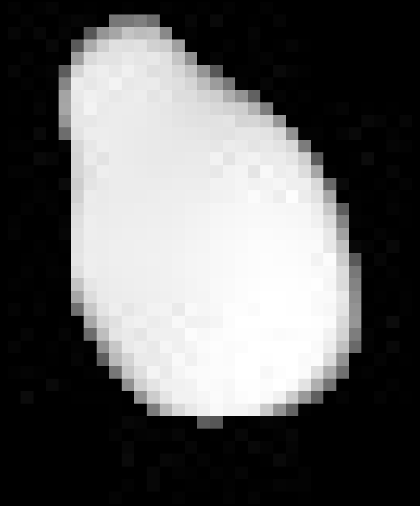
ε Ori

Model (Phoebe)



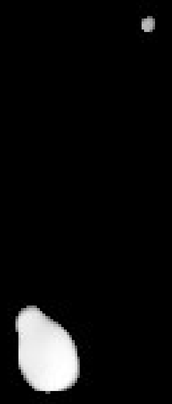
Brož et al. (2023, A&A 676, A60)

- (22) Kalliope + 1 satellite
- data: SKY, AO, LC2, OCC
- occultations, transits, eclipses (20 mmag)
- TRAPPIST, SPECULOOS (3-5 mmag)
- see also Ferrais et al. (2022)



Brož et al. (2023, A&A 676, A60)

- (22) Kalliope + 1 satellite
- data: SKY, AO, LC2, OCC
- occultations, transits, eclipses (20 mmag)
- TRAPPIST, SPECULOOS (3-5 mmag)
- see also Ferrais et al. (2022)



Correlation of P and C_{20}

- high $|C_{20}|$ → precession rate $\mathrm{d}\Omega/\mathrm{d}t$ → low P
- cf. local minima; P must be preset!
- 2 or 3 precession cycles
- a preferred solution $C_{20} = -0.20$, i.e., differentiated oblate interior?
- a homogeneous body excluded!

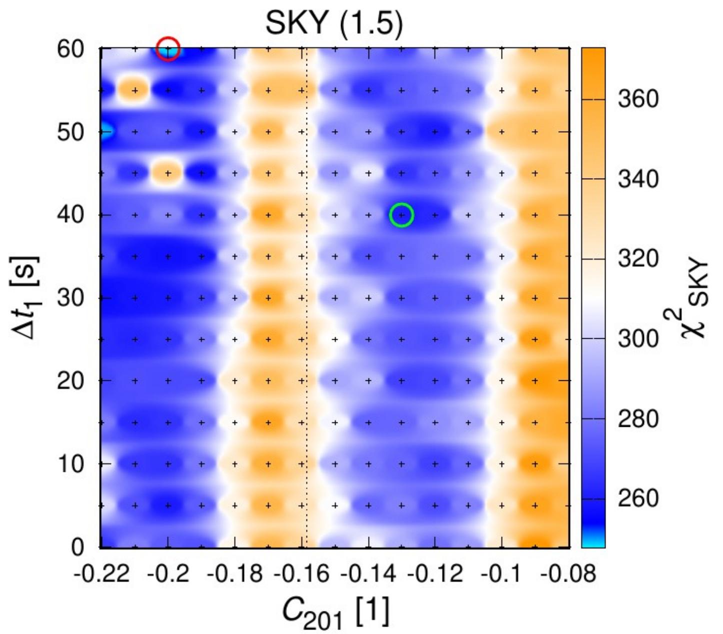


Fig. 12. Quadrupole moment $C_{20,1}$ vs. the tidal time lag Δt_1 of the central body. The corresponding $\chi^2_{\rm sky}$ values are plotted as colours (cyan, blue, white, and orange) and as numbers (gray). SKY and AO datasets were used. Models were converged for 195 combinations of the fixed parameters; all other parameters were free. For each combination, 1000 iterations were computed, that is, 195 000 models in total. Homogeneous body with $C_{20,1} = -0.1586$ is excluded. Preferred solutions are either $\simeq -0.20$, or $\simeq -0.12$, indicated by red and green circles.

Polygonal algorithm

1st clipping:
partial shadowing
back-projection
2nd clipping:
partial visibility
back-projection
'killed' d. errors

Vatti (1992)
Prša et al. (2016)
Clipper2 C++ library

structures[1]
optimisations[2]

[1] set of sets of polygons
[2] bounding-box tests

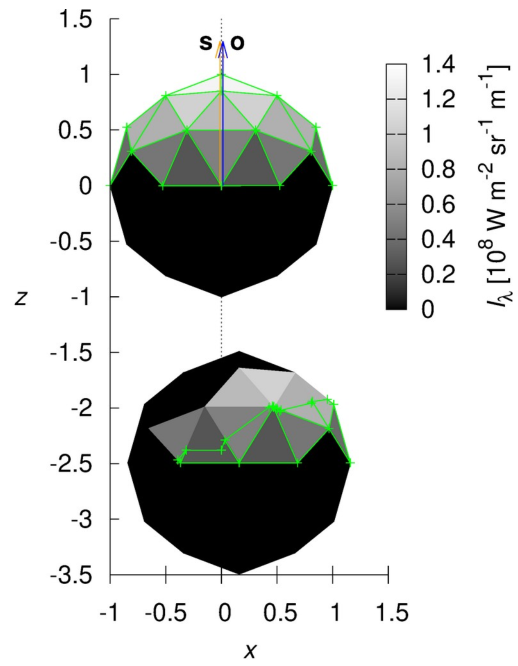


Fig. 1. Two-sphere test of the polygon light curve algorithm. We can even use a very coarse discretisation of 42 nodes for each sphere, because we compute partial eclipses, partial occultations, or partial transits. Shades of gray show the monochromatic intensity I_λ (in W m^{-2} sr^{-1} m^{-1}), green lines show the non-eclipsed and non-occulted polygons used to compute the surface areas. The orange arrow shows the direction towards the Sun and blue towards the observer. The test bodies are metre-sized, 1 au from the Sun and 1 au from the observer. See also Fig. 2.

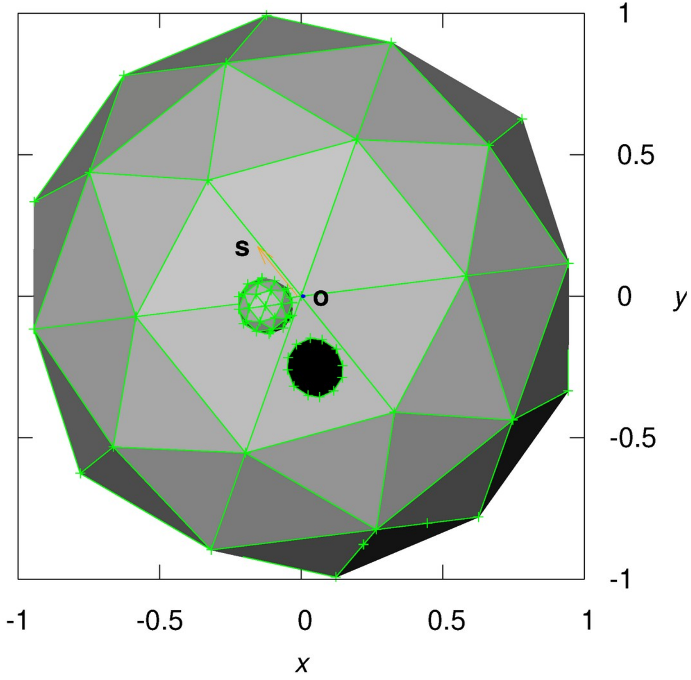


Fig. 3. Similar to Fig. 1. A tiny-triangle test, where one body is large and other body is small. It demonstrates that annular eclipses, as well as partial eclipses, partial occultations, and partial transits, are computed exactly. The polygon corresponding to the shadow (black) has a negative signed area.

```
! Notation:
!
! c .. count
! s .. set of polygons
! p .. polygon
!
! size(polys1)            .. count of sets
! polys1(1)%c             .. count of polygons
! polys1(1)%s(1)%c        .. count of nodes
! polys1(1)%s(1)          .. 1st polygon in a set
! polys1(1)%s(1)%p(1,:) .. 1st node in a polygon
! polys1(1)%s(1)%p(1,1) .. 1st node, x-coordinate

module polytype_module

use iso_c_binding

include 'polytype.inc'

type, bind(c) :: polytype
  integer(c_int) :: c = 0
  real(c_double), dimension(MAXPOLY,3) :: p
end type polytype

type, bind(c) :: polystype
  integer(c_int) :: c = 0
  type(polytype), dimension(MAXPOLYS) :: s
end type polystype

end module polytype_module
```

```fortran
call boundingbox(polys2, boxes)

polys3(:)%c = 0

!$omp parallel do private(i,j,poly_i,poly_j,poly_k) shared(polys2,polys3,boxes)
do i = 1, size(polys2,1)
  if (polys2(i)%c.eq.0) cycle
  poly_i = polys2(i)
  c = 0

  do j = 1, size(polys2,1)
    if (j.eq.i) cycle                                                   ! self-shadowing
    if (poly_i%c.eq.0) exit                                             ! no-polygons-in-set
    if (polys2(j)%c.eq.0) cycle                                         ! no-points-in-polygon
    if ((boxes(j,2).lt.boxes(i,1)).or.(boxes(j,1).gt.boxes(i,2))) cycle ! bounding-box-in-u
    if ((boxes(j,4).lt.boxes(i,3)).or.(boxes(j,3).gt.boxes(i,4))) cycle ! bounding-box-in-v
    if (boxes(j,6).lt.boxes(i,6)+EPS) cycle                             ! is-in-front

    call clip_in_c(poly_i, polys2(j), poly_k)

    c = c+1

    include 'c1.inc'
  enddo

  polys3(i) = poly_i
  clips(i) = clips(i)+c
enddo
!$omp end parallel do

deallocate(boxes)
```

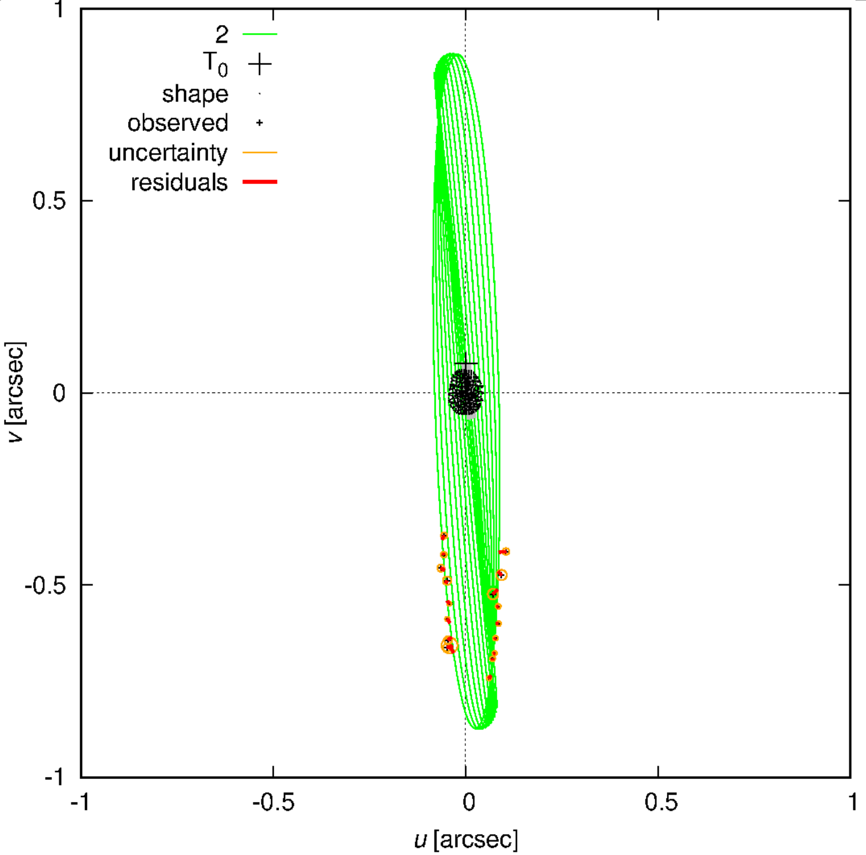


Fig. 9. Orbit of Linus in the (u, v) plane, derived from the short-arc astrometric + photometric model. It fits the PISCO dataset around 2459579, namely, close to the mutual occultation events, when the orbit is seen from the edge. The synthetic orbit of Linus (i.e., body 2) is plotted in green, the observed astrometry in yellow, the residuals in red, the shape of (22) in black. The viewing geometry is changing in the course of time; otherwise the orbit is elliptical. The position at the reference epoch T_0 is marked by the cross. The contribution to χ^2 is $\chi^2_{\rm sky} = 27$ and $n_{\rm sky} = 36$.

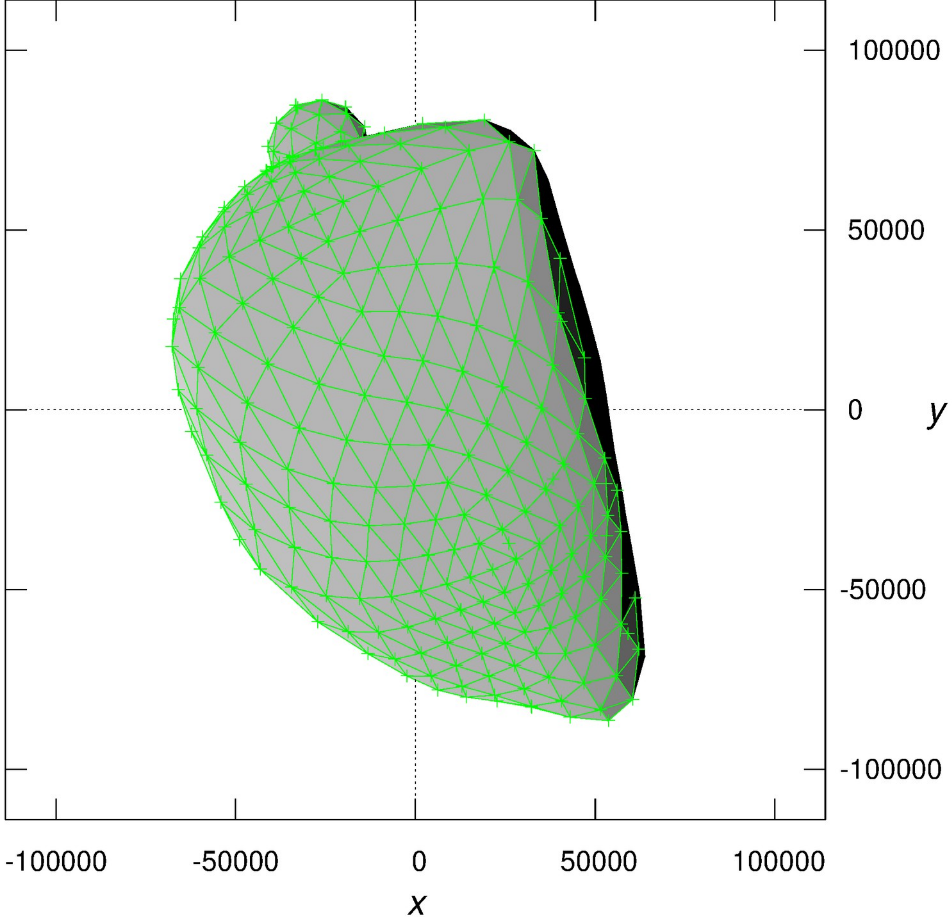


Fig. 11. Example of geometry for the mutual occultation of Linus by (22) Kalliope, namely, the event 2459546. The monochromatic intensity I_λ (in $\mathrm{W\,m^{-2}\,sr^{-1}\,m^{-1}}$) is shown as shades of gray. The ADAM shape model with 800 faces was used for (22), and a sphere with 80 faces for Linus. It is sufficient because partial occultations of faces were computed by the polygonal light curve algorithm.

exact light curve
scattered light
Hapke law

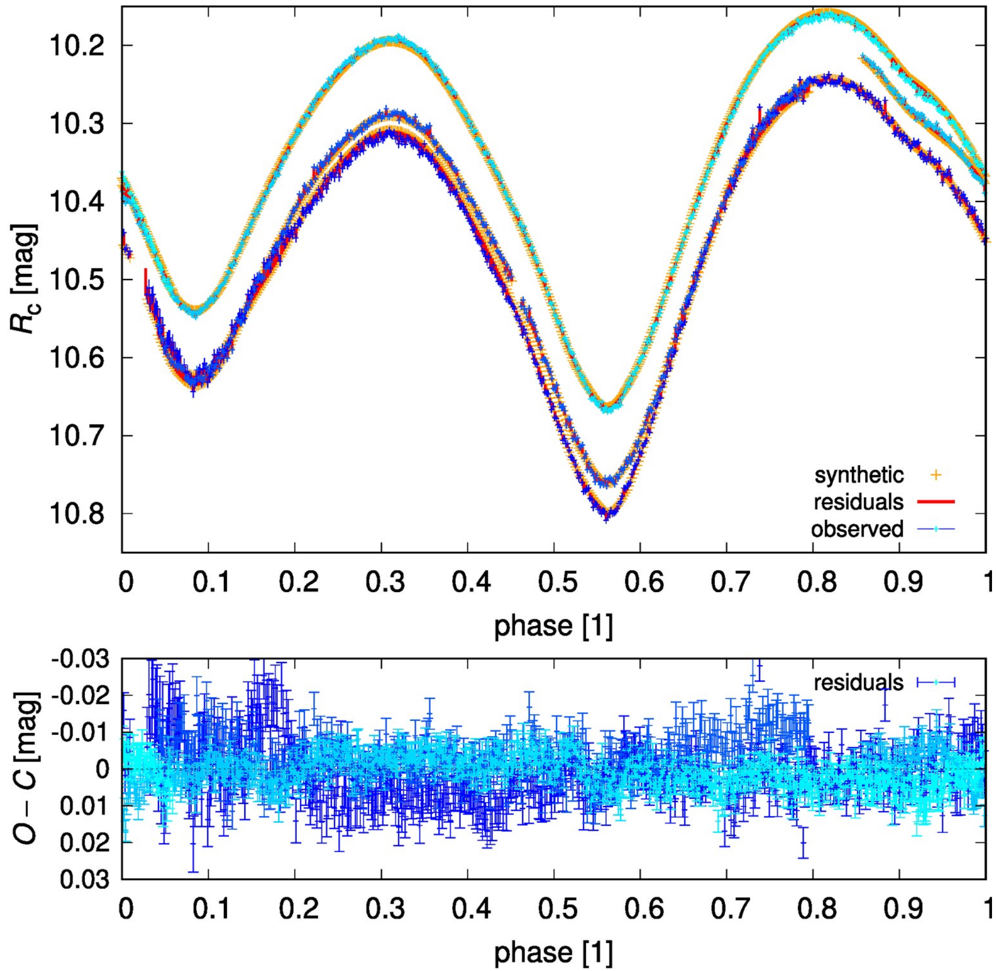
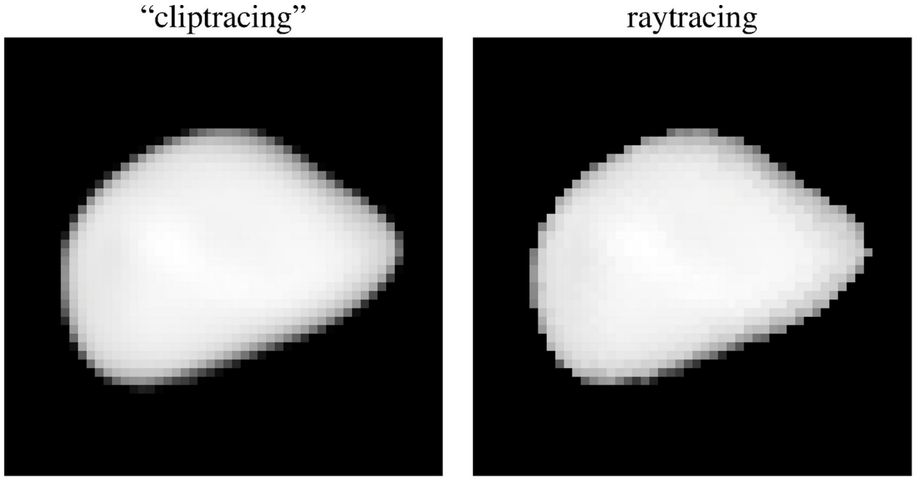


Fig. A.4. Same as Fig. 10, but referring the adjusted shape model of (22) Kalliope. Systematics on the light curves related to the shape were at least partly eliminated. The respective contribution has decreased to $\chi^2_{\rm lc} = 3980$, $n_{\rm lc} = 1829$.

‘Cliptracing’ algorithm

exact synthetic image, pixel = polygon, 3rd clipping: **partial flux-contributions**, no artifacts!



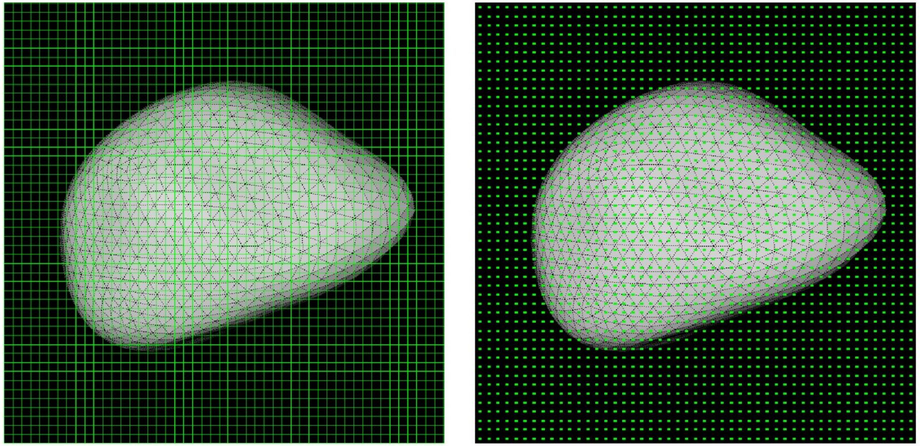


Fig. 4. 1:1 comparison of the ‘cliptracing’ (left) and the raytracing (right) algorithms. In the former, polygons were clipped by individual pixels (analytically) and the synthetic image of (22) is very smooth. In the latter, a simple inside-polygon test was used for each ray, which creates discretisation artefacts and the synthetic image is then ‘noisy’. The Lambert scattering law was used in this test.

Fig. 5. Same as Fig. 4, but showing the corresponding shape composed of polygonal faces (gray) and a grid of either square pixels or points (green).

residuals
not images
3.6 mas/pxl
rotation
non-trivial