slitless documentation

Release 0.3.0

Mehdi Outini

20/04/13, 22:03

Table of Contents:

CHAPTER 1

A forward modeling approach of slitless spectroscopy

Version 0.3.0 of 20/04/13, 22:03

Author Mehdi Outini <mehdi.outini@gmail.com>, Yannick Copin <ycopin@ipnl.in2p3.fr>

[a]

Abstract Slitless spectroscopy has long been considered a complicated and confused technique. Nonetheless, with the advent of Hubble Space Telescope (HST) instruments, characterized by a low sky background level and a high spatial resolution (most notably WFC3), slitless spectroscopy has become an adopted survey tool to study galaxy evolution from space. We aim to investigate its application to single-object studies to measure not only redshift and integrated spectral features, but also spatially-resolved quantities such as galaxy kinematics. It is used to improve redshifts and constrain basic rotation curve parameters, meaning the plateau velocity v0 (in km/s) and the central velocity gradient w0 (in km/s/arcsec). This forward approach makes it possible to mitigate the self-contamination effect, a primary drawback of slitless spectroscopy, and therefore has the potential to increase precision on redshifts. In a limited sample of well-resolved spiral galaxies from HST surveys (3D-HST and GLASS) train galaxy rotation curve parameters. This proof-of-concept work is promising for future large slitless spectroscopic surveys, such as Euclid and WFIRST.

This code (slitless[b]) made in Python was written in a purpose to simulate and analyse galaxy properties in slitless spectroscopy. It implements both the construction and modelisation of galaxy slitless spectra and the measure of spatially resolved quantities such as kinematics. It also constrains spectral features such as redshift, emission lines, line width etc. The first measure of galaxy kinematics using slitless spectra from 3D-HST and GLASS survey is detailed in [Outini20].

1.1 References

- 3D-HST dataset and data reduction[c]
- GLASS dataset and data reduction[d]

[a] https://slitless.readthedocs.io/en/latest/?badge=latest
[b] https://github.com/outinim/slitless
[c] https://3dhst.research.yale.edu/Data_v3.0.html
[d] http://glass.astro.ucla.edu/data/

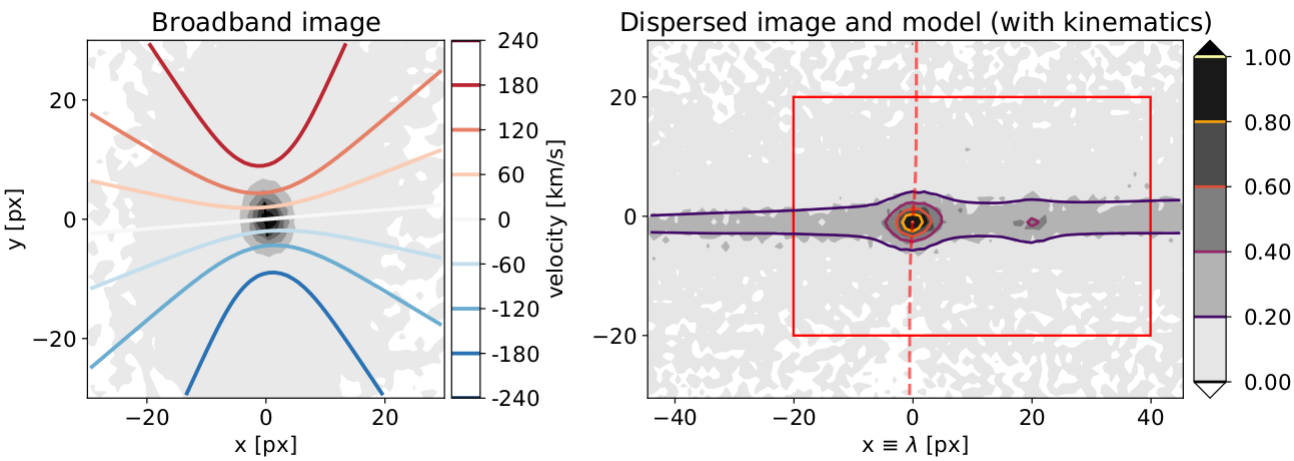


Fig. 1.1: Measure of internal kinematics of the galaxy 1134-MACS-1423 from the GLASS survey. *Left:* adjusted velocity field (contours) over-imposed on broadband image (gray) of galaxy #1134 from the GLASS survey. *Left*: input observed (gray) and modeled (contours) peak-normalized spectrogram centered on the Hα+[NII]+[SII] complex. The red dashed lines represents the adjusted rotation curve at the Hα position.

- Euclid Mission[e]
- WFIRST Mission[f]

1.1.1 Code documentation

slitless package

Subpackages

slitless.summation package

Module contents

Submodules

slitless.summation.summation module

slitless.utils package

Submodules

slitless.utils.G102_Ha module

slitless.utils.G102_Ha_OIII module

slitless.utils.G102_Ha_OIII_multi module

slitless.utils.G102_Ha_mock module

[e] https://www.euclid-ec.org/
[f] https://wfirst.gsfc.nasa.gov/

slitless.utils.G102_Ha_mock_multi module

slitless.utils.G102_Ha_multi module

slitless.utils.G102_OIII module

slitless.utils.G102_OIII_multi module

slitless.utils.euclid module

slitless.utils.euclid_trueU module

slitless.utils.glass module

slitless.utils.glass_OIII module

slitless.utils.glass_mock module

slitless.utils.mcmc_HST module

slitless.utils.mock_ideal module

slitless.utils.sampler module

slitless.utils.sir_simu module

slitless.utils.syst_pa module

Module contents

Submodules

slitless.analysis module

slitless.constants module

slitless.densities module

slitless.fourier module

slitless.galaxy module

slitless.instrument module

slitless.lines module

slitless.optimize module

slitless.plot module

slitless.rotation module

Module contents

genindex

1.1.2 Useful notebooks

HST slitless spectra simulation

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as pd
from glob import glob

# slitless dependencies
from slitless.optimize import Data, Model, Fitter
from slitless.analysis import names_ha, units_ha
from slitless.rotation import plateau
from slitless.constants import reflines
from slitless.plot import *

# for reproduciblity
np.random.seed(18)
```

Data

[52]:
```
path_data = '../data/'
filename2D = sorted(glob(path_data+'/*_2d.fits'))[1]   # some hst 2d.fits galaxy file

data = Data('Ha', filename2D, instrument='G102', decontaminate=True)
```

Model

[53]:
```
# Create data with kinematics

z = 0.6
v0 = 250   # [km/s]
r0 = 0.6   # [arcsec]
re = 1   # [arcsec]
incl = 60     # [deg]
pa = 15    # [deg]
psnr = 40
profile = 'sersic'
input_params = [v0, v0/r0, (1+z) * reflines['Ha'], 80, 20, 30, 25, 0.5, 0, 1]

model = Model(instrument='G102', method_disp='linear', simuData=True,
              Complex=True, Inputparams=input_params,
              names=names_ha, units=units_ha)
model.init_Data(data)
model.set_lines('Ha')
model.set_instrument(wave='custom')
model.set_FoV(60)    # pixel number in the the photometry image
model.set_galaxy(re, incl, pa)
model.set_rotation(plateau)            # to construct mock
```

(continues on next page)

(continued from previous page)

```
model.set_density(profile=profile)
model.set_sens('apodize')
model.set_dispersion()
model.set_mock(input_params, PSNR=psnr)

print(data)
print(model)
```

```
mock data made with rotation: plateau
------------------------------
--------  Data class  --------
------------------------------
#### Simulation ####

grism used: G102
broaband: 60 x 60 px
2D-spectrum: 60 x 360 px
2D-data flux: 168.2
average dispersion: 23.66 A/pixel
spatial scale: 0.13 ''/pixel
redshift = 0.6000
magnitude = 21.7
effective radius = 7.7 px = 1.0 arcsec
inclinaison = 60.00 degree
PA = 15.00 degree
line [Ha] at 10503.4 Angstroms
Peak Signal to Noise: 40.0
---------------------------------------
------------  Model class  -------------
---------------------------------------
Modelisation of mock
Instrument : G102
kinematic resolution = 343 [km/s/pixel]
Rotation model : plateau
Density profile: sersic
lines emission: [Ha] at 10503.4 [A]
effective radius = 1.0 [arcsec] = 7.7 [pixel]
Parameters for mock data:
v0_sini = 250.00 [km/s]
w0_sini = 416.67 [km/s/"]
wHa = 10503.38 [A]
iHa = 80.00 [arb.units]
iNII = 20.00 [arb.units]
iSII = 30.00 [arb.units]
sigma = 25.00 [A]
cte = 0.50 [arb.units]
dy = 0.00 [px]
eta = 1.00 [arb.units]
2D-data shape: 60 pixel x  360 pixel
2D-data flux: 168.2
2D-model shape: 60 pixel x  360 pixel
2D-model flux: 168.2
dispersion method: linear
galaxy PA (fit): 15.0
```

Fit

[54]:
```
guessKin = [200, 200]   # v0, w0
l1, l2 = model.instrument.Wavelength_coverage
```

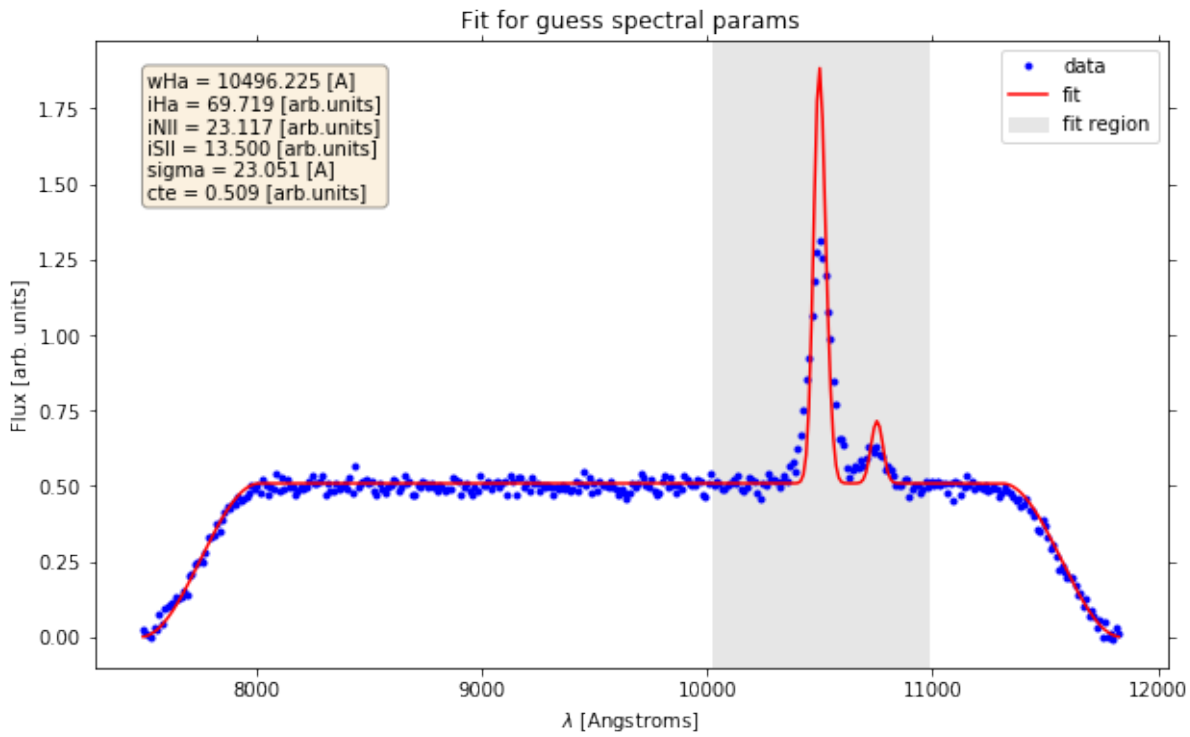
(continues on next page)

(continued from previous page)

```
errors = (30, 30, 25, 10, 5, 5, 10, 0.05, 0.3, 0.1)
lims = (None,None,(7500,17500),None,None,None,None,None,(0,0),(0.5,2))

fit = Fitter(data, model, guessKin=guessKin, guessdy=0, guess_eta=1,
             lims=lims, errors=errors, zoom=True, pixel_zoom=(20,30),
             center_ha_sii=True)
```

[55]:
```
# Prefit the 1D spectrum like on real data

fit.guessSpe = fit.guess_spe(save=True)
```

```
---------------------------------- ----------------------------------
----------------------- Pre-fit on spectrum 1D -----------------------
------------------------------------------------------------------
| FCN = 0.02241                 |     Ncalls=181 (181 total)     |
| EDM = 0.00013 (Goal: 1E-05)   |            up = 1.0            |
------------------------------------------------------------------
|  Valid Min.   | Valid Param.  | Above EDM | Reached call limit |
------------------------------------------------------------------
|     True      |     True      |   False   |       False        |
------------------------------------------------------------------
| Hesse failed  |   Has cov.    | Accurate  | Pos. def. | Forced |
------------------------------------------------------------------
|     False     |     True      |   True    |   True    | False  |
------------------------------------------------------------------
wHa = 10496.225 [A]
iHa = 69.719 [arb.units]
iNII = 23.117 [arb.units]
iSII = 13.500 [arb.units]
sigma = 23.051 [A]
cte = 0.509 [arb.units]
```



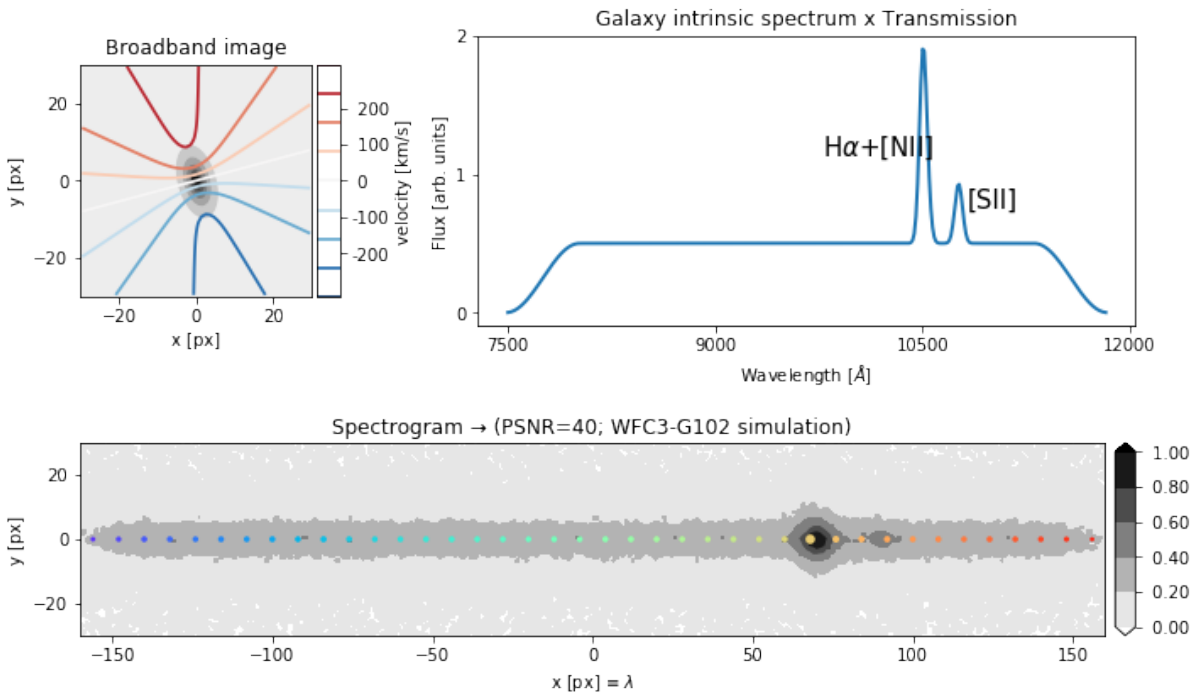
Let's detail the galaxy model that we constructed in order to simulate the **galaxy spectrogram** (slitless spectroscopy spectrum):

- spatial flux distribution $F(x, y)$ (*top left* panel): a sersic profile with r_e = 1 arcsec, i=60°, PA=15°

- velocity field $v(x, y)$ with $v_0 \sin i$ = 250 km/s and $w_0 \sin i$ = 420 km/s/arcsec
- Hα + [NII] + [SII] complex (including instrumental transmission $T(\lambda)$

[56]:
```
_ = input_model_dima(fit, trace=True, yticks=[-20, 0, 20], cmap='gray_r', zoom=160)
```

```
/home/outini/.local/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:3508:␣
↪PearsonRConstantInputWarning: An input array is constant; the correlation coefficent is not␣
↪defined.
  warnings.warn(PearsonRConstantInputWarning())
```



[57]:
```
# Initialize minuit fitter
fit.set_minuit(tol=0.1, callback=None)
print(fit)   # values before fit
```

```
-----------------------------------------------------------------------
---------------------------  Fitter class  ---------------------------
-----------------------------------------------------------------------
Guess parameters:
v0_sini = 200.000 +/- 30.000 [km/s] ; lim: (-inf, inf)
w0_sini = 200.000 +/- 30.000 [km/s/"] ; lim: (-inf, inf)
wHa = 10496.225 +/- 25.000 [A] ; lim: (10023.10, 10969.35)
iHa = 69.719 +/- 13.944 [arb.units] ; lim: (0.00, 278.88)
iNII = 23.117 +/- 6.972 [arb.units] ; lim: (0.00, 174.30)
iSII = 13.500 +/- 6.972 [arb.units] ; lim: (0.00, 174.30)
sigma = 23.051 +/- 4.610 [A] ; lim: (0.00, 34.58)
cte = 0.509 +/- 0.102 [arb.units] ; lim: (0.00, 1.53)
dy = 0.000 +/- 0.300 [px] ; lim: (0.00, 0.00)
eta = 1.000 +/- 0.100 [arb.units] ; lim: (0.50, 2.00)
zoom: True
pixel zoom: 60 px x 40 px
edm_max = 2.0E-04
centered region between NII and SII: True
```

Then, in order to quantify the kinematics detection we're gonna do a fit with a model without kinematics first and then with a model with kinematics.

Fit without kinematics

[58]:
```
# Create data and model class for the fit class without kinematics

model0 = Model(instrument='G102', method_disp='linear', simuData=True,
              Complex=True, Inputparams=input_params,
              names=names_ha, units=units_ha)
model0.init_Data(data)
model0.set_lines('Ha')
model0.set_instrument(wave='custom')
model0.set_FoV(60)    # pixel number in the the photometry image
model0.set_galaxy(re, incl, pa)
model0.set_rotation(none)            # no kinematics
model0.set_density(profile=profile)
model0.set_sens('apodize')
model0.set_dispersion()

fit0 = Fitter(data, model0, guessKin=guessKin, guessdy=0, guess_eta=1,
             lims=lims, errors=errors, zoom=True, pixel_zoom=(20,30),
             center_ha_sii=True)
fit0.guessPars = input_params
fit0.set_noRotation()

# Initalize
fit0.set_minuit(tol=100, callback=None)
print(fit0)   # values before fit
```

```
----------------------------------------------------------------------
---------------------------  Fitter class  ---------------------------
----------------------------------------------------------------------
Guess parameters:
v0_sini = 0.000 +/- 30.000 [km/s] ; lim: (0.00, 0.00)
w0_sini = 0.000 +/- 30.000 [km/s/"] ; lim: (0.00, 0.00)
wHa = 10503.376 +/- 25.000 [A] ; lim: (7500.00, 17500.00)
iHa = 80.000 +/- 10.000 [arb.units] ; lim: (-inf, inf)
iNII = 20.000 +/- 5.000 [arb.units] ; lim: (-inf, inf)
iSII = 30.000 +/- 5.000 [arb.units] ; lim: (-inf, inf)
sigma = 25.000 +/- 10.000 [A] ; lim: (-inf, inf)
cte = 0.500 +/- 0.050 [arb.units] ; lim: (-inf, inf)
dy = 0.000 +/- 0.300 [px] ; lim: (0.00, 0.00)
eta = 1.000 +/- 0.100 [arb.units] ; lim: (0.50, 2.00)
zoom: True
pixel zoom: 60 px x 40 px
edm_max = 2.0E-01
centered region between NII and SII: True
```

[59]:
```
_, _ = fit0.fit_minuit(verbose=True)
print(fit0.result())
```

```
------------------------------------------------------------------
| FCN = 2667                    |     Ncalls=110 (110 total)     |
| EDM = 0.153 (Goal: 0.01)      |            up = 1.0            |
------------------------------------------------------------------
|  Valid Min.   | Valid Param.  | Above EDM | Reached call limit |
------------------------------------------------------------------
|     True      |     True      |   False   |       False        |
------------------------------------------------------------------
| Hesse failed  |   Has cov.    | Accurate  | Pos. def. | Forced |
------------------------------------------------------------------
|     False     |     True      |   True    |   True    | False  |
------------------------------------------------------------------
------------------------------------------------------------------------------------------
|   | Name    |   Value   | Hesse Err | Minos Err- | Minos Err+ | Limit-  | Limit+  | Fixed |
------------------------------------------------------------------------------------------
```

(continues on next page)

(continued from previous page)

```
| 0 | v0_sini |     0     |    30     |           |           |         |         |  yes  |
| 1 | w0_sini |     0     |    30     |           |           |         |         |  yes  |
| 2 | wHa     |  1.050E4  |  0.000E4  |           |           |  7500   |  17500  |       |
| 3 | iHa     |    80     |     8     |           |           |         |         |       |
| 4 | iNII    |    19     |     8     |           |           |         |         |       |
| 5 | iSII    |   30.7    |    2.1    |           |           |         |         |       |
| 6 | sigma   |   24.7    |    1.7    |           |           |         |         |       |
| 7 | cte     |   0.499   |   0.005   |           |           |         |         |       |
| 8 | dy      |   0.00    |   0.30    |           |           |         |         |  yes  |
| 9 | eta     |   1.000   |   0.005   |           |           |   0.5   |    2    |       |
------------------------------------------------------------------------------------------------
------------------------------------------------------------
|       |    wHa    iHa   iNII   iSII  sigma    cte    eta |
------------------------------------------------------------
|   wHa |  1.000  0.959 -0.966  0.366  0.850 -0.276 -0.034 |
|   iHa |  0.959  1.000 -0.986  0.462  0.914 -0.377 -0.073 |
|  iNII | -0.966 -0.986  1.000 -0.352 -0.864  0.252  0.027 |
|  iSII |  0.366  0.462 -0.352  1.000  0.589 -0.816 -0.121 |
| sigma |  0.850  0.914 -0.864  0.589  1.000 -0.565  0.030 |
|   cte | -0.276 -0.377  0.252 -0.816 -0.565  1.000 -0.121 |
|   eta | -0.034 -0.073  0.027 -0.121  0.030 -0.121  1.000 |
------------------------------------------------------------
----------------------------------------------------------------------
--------------------------- Result migrad ----------------------------
----------------------------------------------------------------------
minimization time : 2 min and 11 sec
Minuit ncalls : 189
edm : 0.152530
is_valid: True
has_valid_parameters: True
hesse_failed: False
has_covariance: True
has_posdef_covar: True
has_accurate_covar: True
has_made_posdef_covar: False
is_above_max_edm: False
has_reached_call_limit: False
FIT HAS CONVERGED y("0)y

Intrinsic parameters:
v0_sini = 0.0000 +/- 0.0000 [km/s]
w0_sini = 0.0000 +/- 0.0000 [km/s/"]
wHa = 10503.0488 +/- 1.4269 [A]
iHa = 79.6726 +/- 8.2102 [arb.units]
iNII = 19.3898 +/- 7.7915 [arb.units]
iSII = 30.6851 +/- 2.1364 [arb.units]
sigma = 24.6718 +/- 1.7265 [A]
cte = 0.4992 +/- 0.0050 [arb.units]
dy = 0.0000 +/- 0.0000 [px]
eta = 1.0000 +/- 0.0046 [arb.units]

Peak Signal to Noise: 40.0
Chi2/DoF = 2667/2393
Total peak-normalized RMS: 0.016
Data: [Ha] line at 10503.4 [A] : z = 0.60000
Fit: [Ha] line at 10503.0 [A] : z = 0.59995 +/- 0.00022
ecart relatif en redshift : 0.008%
----------------------------------------------------------------------
Correlation matrix:
[ 1.    0.96 -0.97  0.37  0.85 -0.28 -0.03]
[ 0.96  1.   -0.99  0.46  0.91 -0.38 -0.07]
```

(continues on next page)

(continued from previous page)

```
[-0.97 -0.99  1.   -0.35 -0.86  0.25  0.03]
[ 0.37  0.46 -0.35  1.    0.59 -0.82 -0.12]
[ 0.85  0.91 -0.86  0.59  1.   -0.57  0.03]
[-0.28 -0.38  0.25 -0.82 -0.57  1.   -0.12]
[-0.03 -0.07  0.03 -0.12  0.03 -0.12  1.  ]

Extra info on the fit:
```

Fit with kinematics

[65]:
```
fit.guessPars[:2] = v0, v0/r0
fit.set_minuit(tol=100, callback=None)
_, _ = fit.fit_minuit(verbose=True)
print(fit.result())
```

```
------------------------------------------------------------------
| FCN = 2454                    |     Ncalls=168 (168 total)     |
| EDM = 0.164 (Goal: 0.01)      |            up = 1.0            |
------------------------------------------------------------------
|  Valid Min.   | Valid Param.  | Above EDM | Reached call limit |
------------------------------------------------------------------
|     True      |     True      |   False   |       False        |
------------------------------------------------------------------
| Hesse failed  |   Has cov.    | Accurate  | Pos. def. | Forced |
------------------------------------------------------------------
|     False     |     True      |   True    |   True    | False  |
------------------------------------------------------------------
------------------------------------------------------------------------------------------
|   | Name    |   Value   | Hesse Err | Minos Err- | Minos Err+ | Limit-  | Limit+  | Fixed |
------------------------------------------------------------------------------------------
| 0 | v0_sini |    320    |    50     |            |            |         |         |       |
| 1 | w0_sini |    440    |    50     |            |            |         |         |       |
| 2 | wHa     |  1.050E4  |  0.000E4  |            |            | 10023.1 | 10969.3 |       |
| 3 | iHa     |    79     |     8     |            |            |    0    | 278.876 |       |
| 4 | iNII    |    19     |     7     |            |            |    0    | 174.297 |       |
| 5 | iSII    |   30.1    |    2.1    |            |            |    0    | 174.297 |       |
| 6 | sigma   |   25.2    |    1.6    |            |            |    0    | 34.576  |       |
| 7 | cte     |   0.500   |   0.005   |            |            |    0    | 1.52577 |       |
| 8 | dy      |   0.00    |   0.30    |            |            |         |         |  yes  |
| 9 | eta     |   1.000   |   0.005   |            |            |   0.5   |    2    |       |
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
|         | v0_sini w0_sini     wHa     iHa    iNII    iSII   sigma     cte     eta |
------------------------------------------------------------------------------------------
| v0_sini |   1.000  -0.479  -0.016  -0.031   0.015  -0.057  -0.020   0.053   0.043 |
| w0_sini |  -0.479   1.000  -0.011  -0.005   0.012   0.008   0.029  -0.009  -0.028 |
|     wHa |  -0.016  -0.011   1.000   0.955  -0.963   0.348   0.839  -0.265  -0.028 |
|     iHa |  -0.031  -0.005   0.955   1.000  -0.984   0.449   0.908  -0.372  -0.069 |
|    iNII |   0.015   0.012  -0.963  -0.984   1.000  -0.333  -0.853   0.241   0.020 |
|    iSII |  -0.057   0.008   0.348   0.449  -0.333   1.000   0.579  -0.812  -0.117 |
|   sigma |  -0.020   0.029   0.839   0.908  -0.853   0.579   1.000  -0.564   0.038 |
|     cte |   0.053  -0.009  -0.265  -0.372   0.241  -0.812  -0.564   1.000  -0.126 |
|     eta |   0.043  -0.028  -0.028  -0.069   0.020  -0.117   0.038  -0.126   1.000 |
------------------------------------------------------------------------------------------
----------------------------------------------------------------------
--------------------------- Result migrad ----------------------------
----------------------------------------------------------------------
minimization time : 3 min and 12 sec
Minuit ncalls : 274
edm : 0.163897
is_valid: True
has_valid_parameters: True
```

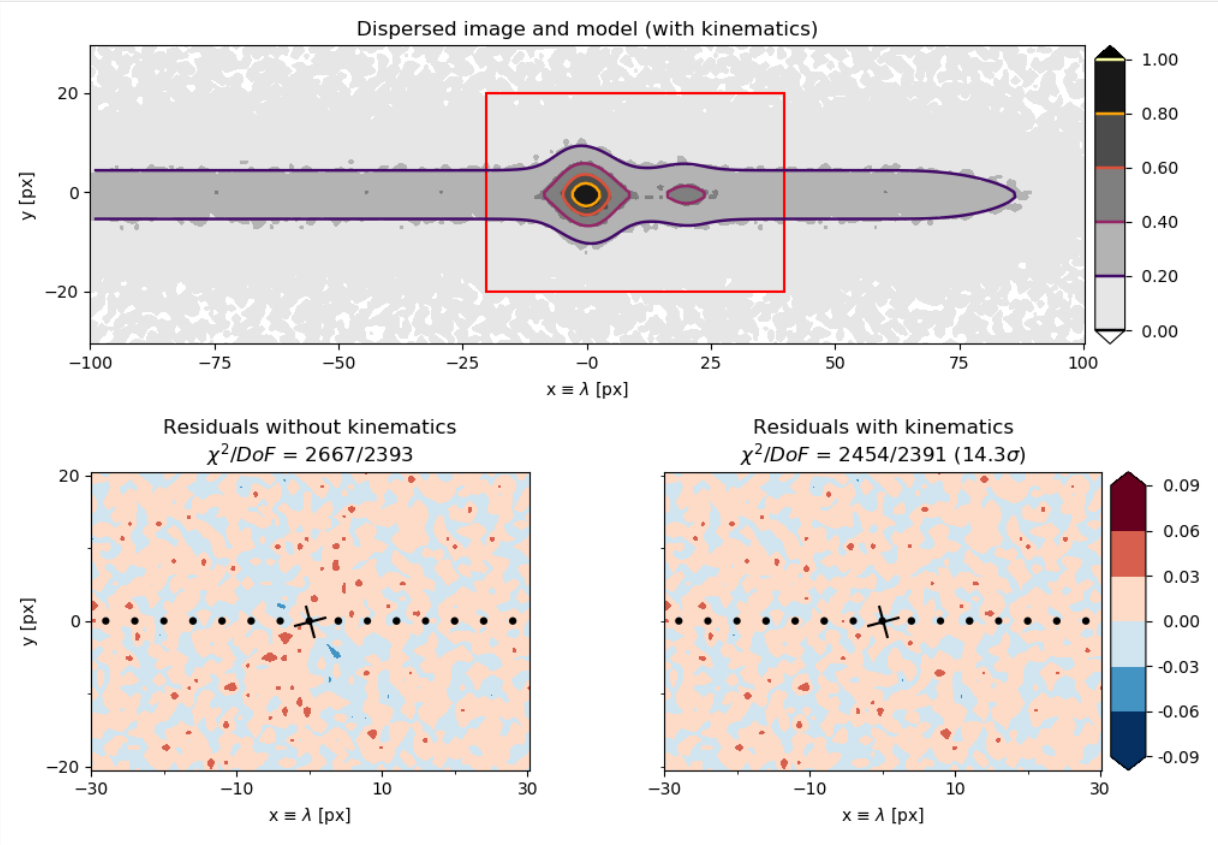
(continues on next page)

(continued from previous page)

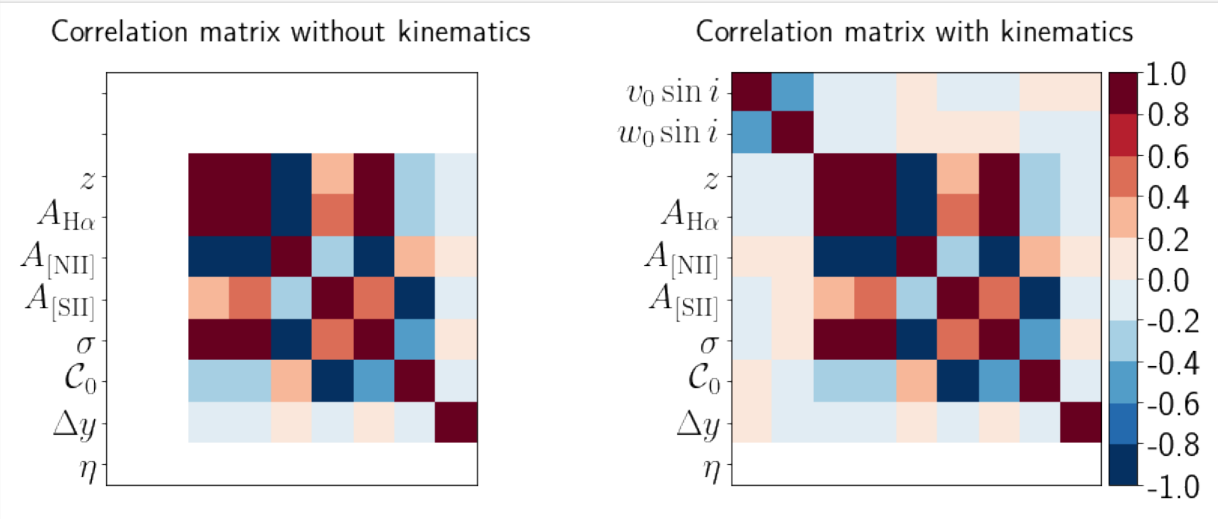
```
hesse_failed: False
has_covariance: True
has_posdef_covar: True
has_accurate_covar: True
has_made_posdef_covar: False
is_above_max_edm: False
has_reached_call_limit: False
FIT HAS CONVERGED y("0)y

Intrinsic parameters:
v0_sini = 321.2298 +/- 51.6880 [km/s]
w0_sini = 443.6140 +/- 51.7726 [km/s/"]
wHa = 10503.1178 +/- 1.3466 [A]
iHa = 79.4085 +/- 7.6988 [arb.units]
iNII = 19.2493 +/- 7.2746 [arb.units]
iSII = 30.0687 +/- 2.0901 [arb.units]
sigma = 25.1893 +/- 1.6120 [A]
cte = 0.5001 +/- 0.0049 [arb.units]
dy = 0.0000 +/- 0.0000 [px]
eta = 0.9999 +/- 0.0046 [arb.units]

Peak Signal to Noise: 40.0
Chi2/DoF = 2454/2391
Total peak-normalized RMS: 0.015
Data: [Ha] line at 10503.4 [A] : z = 0.60000
Fit: [Ha] line at 10503.1 [A] : z = 0.59996 +/- 0.00021
ecart relatif en redshift : 0.007%
----------------------------------------------------------------------
Correlation matrix:
[ 1.   -0.48 -0.02 -0.03  0.02 -0.06 -0.02  0.05  0.04]
[-0.48  1.   -0.01 -0.01  0.01  0.01  0.03 -0.01 -0.03]
[-0.02 -0.01  1.    0.95 -0.96  0.35  0.84 -0.27 -0.03]
[-0.03 -0.01  0.95  1.   -0.98  0.45  0.91 -0.37 -0.07]
[ 0.02  0.01 -0.96 -0.98  1.   -0.33 -0.85  0.24  0.02]
[-0.06  0.01  0.35  0.45 -0.33  1.    0.58 -0.81 -0.12]
[-0.02  0.03  0.84  0.91 -0.85  0.58  1.   -0.56  0.04]
[ 0.05 -0.01 -0.27 -0.37  0.24 -0.81 -0.56  1.   -0.13]
[ 0.04 -0.03 -0.03 -0.07  0.02 -0.12  0.04 -0.13  1.  ]

Extra info on the fit:
```

We can then compare the two fits by plotting the residuals and compute the **p-value** which tell us how much the addition of kinematics in the model is *relevant* to describe the data.

[96]:
```
_ = paper_simu_residuals_old(fit, fit0, zoom=100, alpha=5, trace=True,
                             npts=80, s_cross=400, norm_flux=40)
```



[74]:
```
# Correlation matrices
corr0, corr = fit0.correlation(), fit.correlation()

_ = correlation_both(corr, corr0, cmap='RdBu_r', r0=None, line='Ha', N=10, useTex=True)
```



Credits

This package was made using part of the Cookiecutter[h] and the audreyr/cookiecutter-pypackage[i] project template.

[h] https://github.com/audreyr/cookiecutter
[i] https://github.com/audreyr/cookiecutter-pypackage

Bibliography

[Outini20] 2020A&26A...633A..43O[g]

[g] https://www.aanda.org/articles/aa/abs/2020/01/aa36318-19/aa36318-19.html