20250508 ppxf mass to light

1. Data

I am still using IC3392

```
Cube dimensions → nz = 3761, ny = 438, nx = 437
```

2. Wavelength cutoff and velocity scale

''Sky subtraction is clearly not perfect, but the best that we can do for the moment. Below 7000 Å it is generally acceptable, at longer wavelengths the situation is worse.''

So I make a cutoff at $\sim 7000Å$. Actually, now it remains $4750 - 7050Å$:

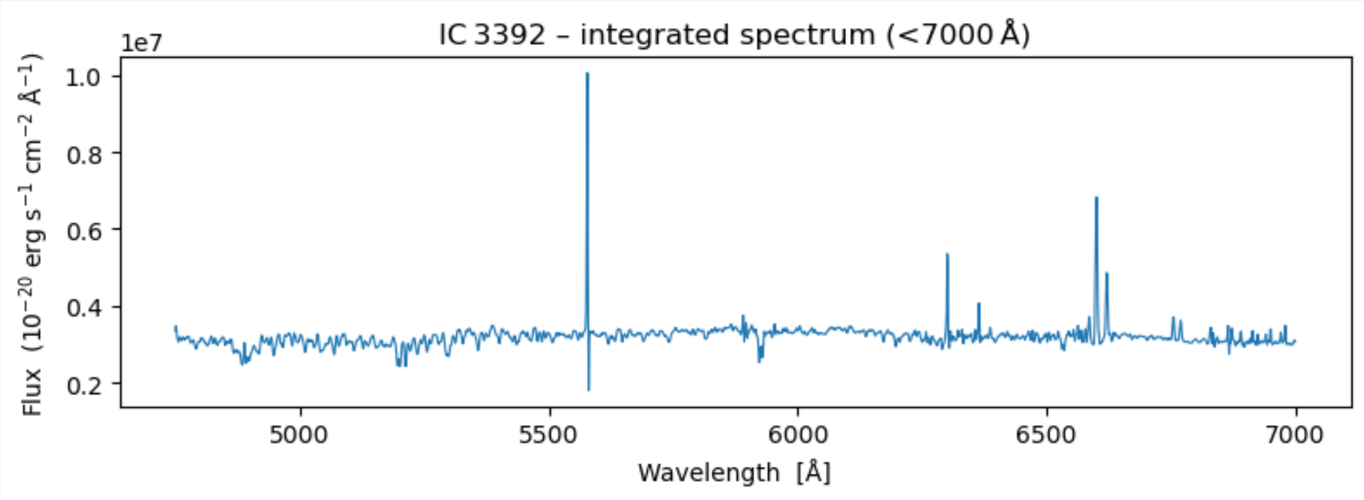
```
lam_min = 4750.0                     # min λ in Å
lam_max = 7050.0                     # max λ in Å
```

Then I compute the velocity scale:

```
c_kms    = c.c.to(u.km/u.s).value        # 299 792.458
dlnλ     = np.diff(np.log(lam_ang))  # dlnλ in Å
velscale = np.min(c_kms * dlnλ) # km/s per pixel
```

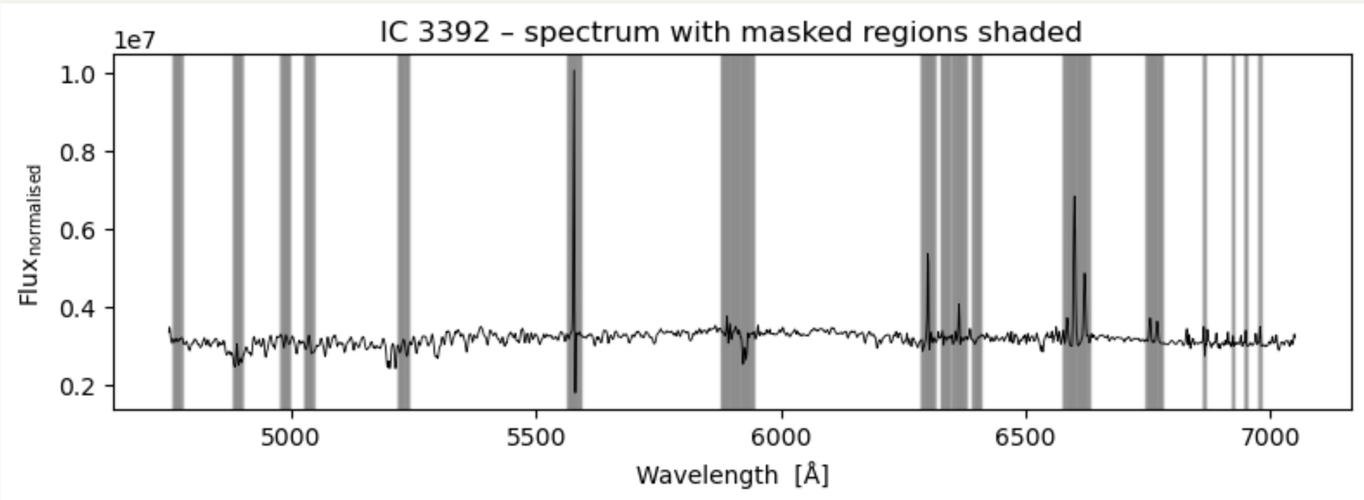
This gives $53.16 km/s$.

Here is the native spectrum:



3. Mask emission lines

Since we are only interested in the continuum, I mask the emission lines from galaxy (observer frame) and air (rest frame) by `specMask_KIN.txt`. I just mask them without modifying the raw spectrum:



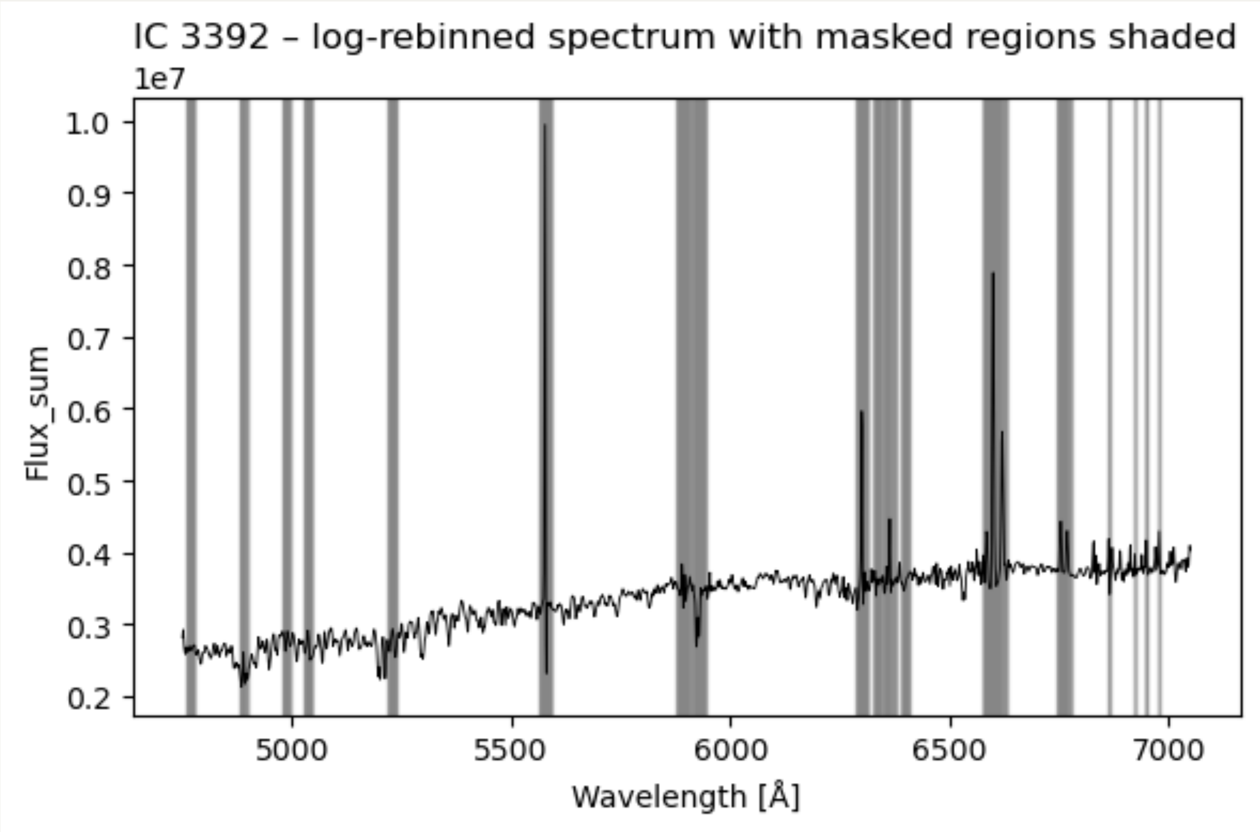
Note: "Because the MUSE spectrographs do not operate in vacuum the wavelength calibration is based on arc line wavelengths in standard air (Weilbacher et al. 2020)." Thus, it is correct that we are taking the lines measure in air.

4. `log_rebin`

Now I need to do `log_rebin`. It seems that this is one of the requirements in `pPXF`. But the question is that, why in natual log rather than $\log_{10}$ (same question for `velscale`)? No need to multiply an extra constant when taking derivative?

I still do `log_rebin` anyway for both flux and noise, and I force `velscale=velscale`, so I got:

```
Log-grid length : 2228 pixels
velscale        : 53.159 km/s
```



5. SPS templates: E-MILES

Then I load the SPS templates. Here I choose `spectra_emiles_9.0.npz` because it seems to be more suitable for IFS data.

E-MILES SPS model templates: Vazdekis et al. (2016).

6. FWHM and MUSE LSF

`pPXF` requires thestellar templates and the galaxy spectrum to have the same instrumental resolution before it adds any extra broadening for the LOSVD.

Emsellem+2022 use this equation for MUSE LSF:

$$FWHM\left(\lambda\left[\mathring{A}\right]\right) = 5.866\times10^{-8}\lambda^2 - 9.187\times10^{-4}\lambda + 6.040.$$

The idea is that the templates should have the same FWHM as muse, so:

```
lam_temp_arr = np.arange(lam_min, lam_max+1, dtype=int)

fwhm_gal = {
    "lam": lam_temp_arr,
    "fwhm": 5.866e-8 * lam_temp_arr**2
            - 9.187e-4 * lam_temp_arr
            + 6.040
}

sps = lib.sps_lib(
    filename, velscale_out, fwhm_gal,
    lam_range=[lam_min, lam_max+1],
)
```

7. `pPXF` fitting

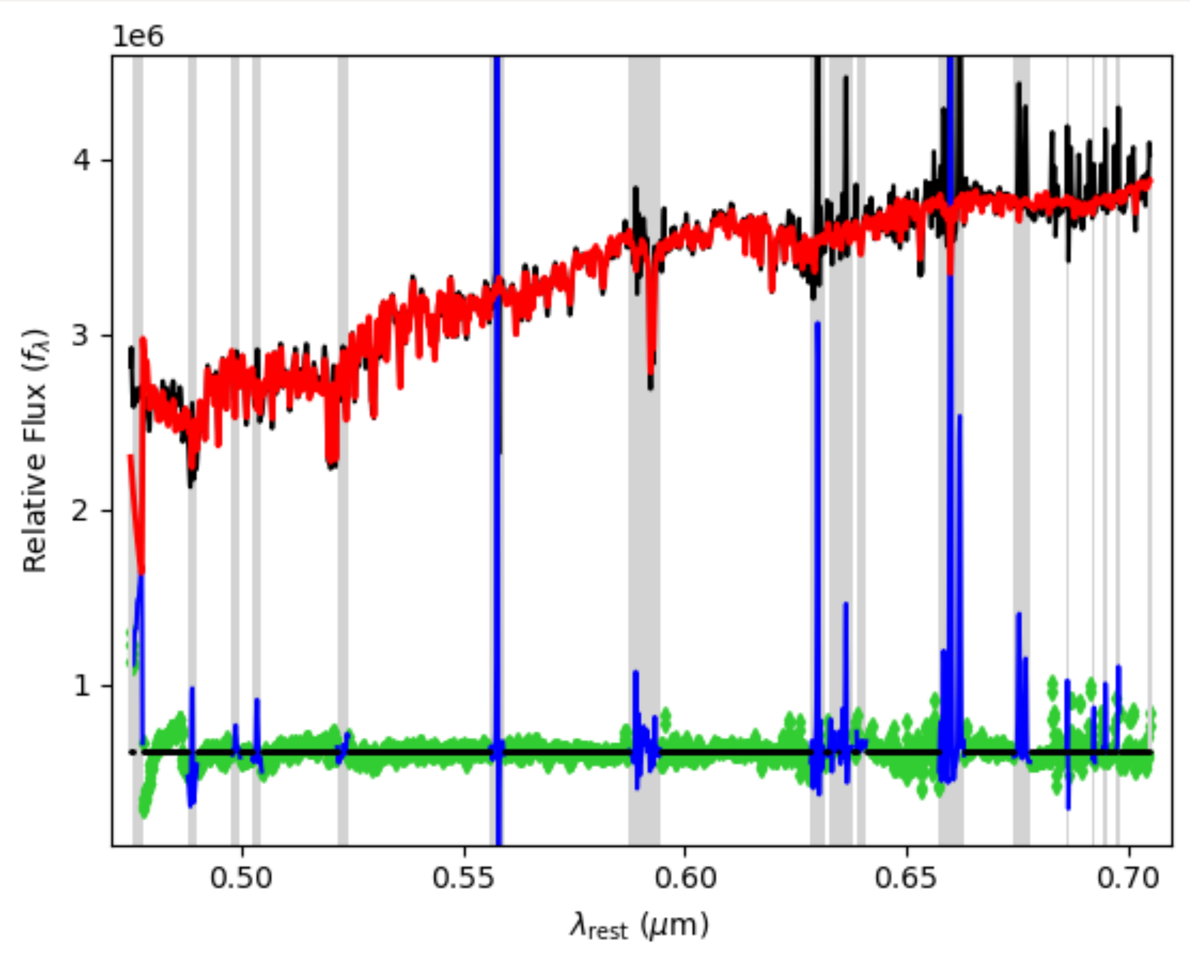
By passing the mask regions as the `goodpixels`, we can run `pPXF`:

```
start_V   = v_guess
start_sig = 5 * velscale_out

pp = ppxf.ppxf(
    templates    = sps.templates,
    # templates    = log_templates_hr,
    galaxy       = log_flux,
    noise        = log_noise,
    velscale     = velscale_out,
    start        = [start_V, start_sig],
    degree       = 12,
    mdegree      = 0,
    moments      = 2,
    # clean        = True,
    goodpixels   = goodpixels,
    lam          = np.e**(log_lam),
```

```
    plot          = True)

print(f"V = {pp.sol[0]:.3f} km/s,  σ = {pp.sol[1]:.3f} km/s")
```

Note that "Unless a good initial guess is available, it is recommended to set the starting sigma >= 3*velscale in km/s (i.e. 3 pixels)." Here I choose initial guess to be `start_V` as my estimation when matching the emission line at observer frame, and `start_sig` to be 5 times of velcity scale. Fitting model is set as an additive polynomial of 12 with no multiplicative polynomia and first 2 moments of Gauss–Hermite expansion. This yields:

```
 Best Fit:       Vel     sigma
 comp.   0:       1701        45
chi2/DOF: 1.316e-10; DOF: 1833; degree = 12; mdegree = 0
method = capfit; Jac calls: 4; Func calls: 14; Status: 2
linear_method = lsq_box; Nonzero Templates (>0.1%): 2/150
V = 1701.001 km/s,  σ = 44.569 km/s
```



8. Mass-to-Light ratio

Now with the fitting results, I can extract the weight to get M/L. Since we pick wavelength within $4750 \sim 7000 Å$, I choose the M/L in r band of SDSS.

```
reg_dim = sps.templates.shape[1:]
weights = pp.weights.reshape(reg_dim)/pp.weights.sum()
                              # enforce Σw = 1

# Get fitted redshift
z_fit = pp.sol[0] / c_kms
print(f"Fitted redshift  : {z_fit:.5f}")

ML_r = sps.mass_to_light(weights, band="SDSS/r", redshift=z_fit,)
                    # intrinsic M/L (dimensionless)
print(f"M/L (r band)  : {ML_r:.3f}  M☉/L☉ =
log({np.log10(ML_r):.3f}  M☉/L☉)")
```

This returns:

```
Fitted redshift  : 0.00567
(M*/L)=8.010 (SDSS/r at z=0.0057)
M/L (r band)  : 8.010  M☉/L☉ = log(0.904  M☉/L☉)
```

Now I can also find the r band luminosity and compute the total stellar mass:

```
from speclite import filters
f_r   = filters.load_filter('sdss2010-r')
lam_r = f_r.wavelength
R_r   = f_r.response

# best-fit continuum in erg s⁻¹ cm⁻² Å⁻¹
cont_flux = pp.bestfit * 1e-20 * normalisation

# best-fit continuum in physical units (L_sun Å⁻¹)
cont_L = cont_flux * 4*np.pi*(D.to('cm').value)**2 * erg_to_Lsun
```

```
#    cont_flux [erg s⁻¹ cm⁻² Å⁻¹] at wavelengths lam_obs =
exp(log_lam)
lam_obs       = np.exp(log_lam)
flux_at_lam_r = np.interp(lam_r, lam_obs, cont_flux)

# integrate the best-fit spectrum over the r-band filter
L_r    = np.trapezoid( np.interp(lam_r, np.exp(log_lam), cont_L) *
R_r, lam_r )  # L_sun

# r band apparent AB magnitude

# Compute the AB magnitude
m_r = f_r.get_ab_magnitude(
    flux_at_lam_r * u.erg / (u.cm**2 * u.s * u.AA),
    wavelength=lam_r
)

print(f"r-band apparent AB mag = {m_r:.2f}")

# M/L in r-band
M_star = ML_r * L_r

print(f"Log L  (r-band)    : log({np.log10(L_r):.3f}  L☉)")
print(f"log M_star         : log({np.log10(M_star):.3f}  M☉)")
```

Finally, we have

```
r-band apparent AB mag = 12.26
Log L  (r-band)    : log(8.250  L☉)
log M_star         : log(9.153  M☉)
```