

# PyNeb\_manual\_5

June 2, 2020

## 1 The extinction class: RedCorr()

```
[1]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import pyneb as pn
```

The class RedCorr manages the extinction (reddening) correction. It can compute the logarithmic extinction at  $H\beta$  by comparing an observed ratio to a theoretical one (usually  $H\alpha/H\beta$ , but any other ratio can be used). The object is also able to compute the correction to be applied to any intensity, given the wavelength of the line.

Various extinction laws are included in the class, and any user-defined function can also be implemented. The available extinction laws can be listed by entering (here no need to instantiate an object):

To explore some properties, you can directly use the class methods:

```
[2]: pn.RedCorr().printLaws()
```

'No correction':

No correction, return 0.0

'CCM89':

Cardelli, Clayton & Mathis 1989, ApJ 345, 245

<http://adsabs.harvard.edu/abs/1989ApJ...345..245C>

Comments: Depends on  $R_V$ , default value being 3.1

Scope: Applicable to both dense and diffuse ISM

Range: UV through IR

'CCM89 Bal07':

Galactic extinction law based on Cardelli et al 1989, modified by  
Blagrove et al 2007

for  $3.3 < x < 8$  ( $1250 < \lambda < 3030$ )

Blagrove et al 2007, ApJ, 655, 299  
<http://adsabs.harvard.edu/abs/2007ApJ...655..299B>  
Cardelli, Clayton & Mathis 1989, ApJ 345, 245  
<http://adsabs.harvard.edu/abs/1989ApJ...345..245C>

Comments:

Same as CCM89 for  $x < 3.3$  and  $x > 8$   
Revised values for  $3.3 < x < 8$   
Based on observation of Orion stars  
Depends on  $R_V$ , default value being 3.1

Range: UV through IR

'CCM89 oD94':

Galactic extinction law based on Cardelli et al 1989, modified by  
O'Donnell 1994  
for  $1.1 < x < 3.3$  ( $9100 < \lambda < 3030$ )

O'Donnell 1994, ApJ, 422, 1580  
<http://adsabs.harvard.edu/abs/1994ApJ...422..1580>  
Cardelli, Clayton & Mathis 1989, ApJ 345, 245  
<http://adsabs.harvard.edu/abs/1989ApJ...345..245C>

Comments:

Same as CCM89 for  $x < 1.1$  and  $x > 3.3$   
Revised values for  $1.1 < x < 3.3$   
Produces lower correction in the near UV at low  $R_V$

Scope: Galactic

Range: UV through IR

'S79 H83 CCM89':

Galactic extinction law (0-33000 Å range):  
- In the UV, from Seaton 1979  
- In the opt/NIR (3600-9100) Howarth 1983  
- In the FIR (9100-33000) Cardelli et al 1989

Seaton 1979, MNRAS, 187, 73) and  
<http://adsabs.harvard.edu/abs/1979MNRAS.187P..73S>  
Howarth 1983, MNRAS, 203, 301) Galactic law  
<http://adsabs.harvard.edu/abs/1983MNRAS.204.1091H>  
Cardelli, Clayton and Mathis 1989, ApJ, 345, 245  
<http://adsabs.harvard.edu/abs/1989ApJ...345..245C>

Scope: Galactic

Range: UV through IR

'K76':

Kaler 1976, ApJS, 31, 517  
<http://adsabs.harvard.edu/abs/1976ApJS...31..517K>

Comments:

This function returns the correction relative to Hbeta ( $f_{\lambda}$ ) and  
not the extinction law ( $X(1/\lambda)$ ).  
It cannot be used for absolute correction.

Range: 3000 to >20000

'SM79 Gal':

Galactic extinction law  
Savage & Mathis 1979, ARA&A, 17, 73  
<http://adsabs.harvard.edu/abs/1979ARA%26A..17...73S>

Comments:

Average of several extinction laws  
 $R_V=3.1$

Scope: Galactic  
Range: UV through IR

'G03 LMC':

Extinction curve for the LMC  
Gordon et al. (2003, ApJ, 594,279)  
<http://adsabs.harvard.edu/abs/2003ApJ...594..279G>

Comments:

Average curve for the LMC  
 $R_V = 3.41$

Scope: LMC  
Range: 1200 through fIR

'MCC99 FM90 LMC':

In the UV, this method returns the extinction curve proposed for the LMC  
by Misselt et al 1999 based on the 1990 variant of the Fitzpatrick &  
Massa law

In the opt/IR, it returns the Fitzpatrick & Massa 1990 law.

Misselt, Clayton & Gordon 1999 , ApJ, 515, 128

<http://adsabs.harvard.edu/abs/1999ApJ...515..128M>  
Fitzpatrick & Massa 1990, ApJS, 72, 163  
<http://adsabs.harvard.edu/abs/1990ApJS...72..163F>

Comments:

The Fitzpatrick & Massa 1990 law in the UV depends on 6 parameters, stored in RedCorr.FitzParams.

The method sets RedCorr.FitzParams to the values of set in the Fitzpatrick 1999 paper,

which includes an explicit dependence on  $R_V$ .

$R_V$  must be provided, as the law depends on its value.

We refer to FM90 and not to the original FM88 because the value of a constant in  $F(\lambda)$  slightly changed (0.0564  $\rightarrow$  0.05644)

The value of another constant of  $F(\lambda)$  appears to change from FM90 to MCC99, but it is probably a typo (0.5392  $\rightarrow$  0.5329)

Scope: LMC

'F99-like':

In the UV, it returns the Fitzpatrick & Massa 1990 law.

In the opt/IR, it returns the Fitzpatrick & Massa 1990 law.

Fitzpatrick 1999, PASP, 11, 63  
<http://adsabs.harvard.edu/abs/1999PASP...111..63F>  
Fitzpatrick & Massa 1990, ApJS, 72, 163  
<http://adsabs.harvard.edu/abs/1990ApJS...72..163F>

Comments:

The FM90 depends on 6 parameters which must be set by the user and are stored in RedCorr.FitzParams.

For the predefined set of parameters defined in FM99, use instead the F99 law.

$R_V$  must be provided, as the law depends on it. The dependence with  $R_V$  follows Table 4 in the F99 paper

Range: UV through IR

'F99':

This method returns the R-dependent IR-through-UV extinction curve proposed by Fitzpatrick 1999.

Fitzpatrick 1999, PASP, 11, 63  
<http://adsabs.harvard.edu/abs/1999PASP...111..63F>  
based on:  
Fitzpatrick & Massa 1990, ApJS, 72, 163  
<http://adsabs.harvard.edu/abs/1990ApJS...72..163F>

Comments:

The Fitzpatrick & Massa 1990 law in the UV depends on 6 parameters, stored in `RedCorr.FitzParams`.

The method sets `RedCorr.FitzParams` to the values of set in the Fitzpatrick 1999 paper, which includes an explicit dependence on `R_V`. `R_V` must be provided, as the law depends on its value.

Range: UV through IR

'F88 F99 LMC':

This method returns:

- in the UV, the average LMC extinction curve derived by Fitzpatrick & Massa 1988
- in the opt/IR, the R-dependent extinction curve proposed by Fitzpatrick 1999.

Fitzpatrick 1999, PASP, 11, 63

<http://adsabs.harvard.edu/abs/1999PASP..111...63F>

Fitzpatrick & Massa 1988, ApJ, 328, 734

<http://adsabs.harvard.edu/abs/1988ApJ...328..734F>

Comments:

The Fitzpatrick and Massa law in the UV depends on 6 parameters, stored in `RedCorr.FitzParams` and

here set to the LMC values derived in FM88

`R_V` must be provided, as the law depends on it

Scope: LMC

Range: UV through IR

Less detailed output is obtained with:

```
[3]: pn.RedCorr().getLaws()
```

```
[3]: dict_keys(['No correction', 'CCM89', 'CCM89 Bal07', 'CCM89 oD94', 'S79 H83  
CCM89', 'K76', 'SM79 Gal', 'G03 LMC', 'MCC99 FM90 LMC', 'F99-like', 'F99', 'F88  
F99 LMC'])
```

To apply a correction, you need to instantiate the object:

```
[4]: rc = pn.RedCorr(E_BV = 1.2, R_V = 3.2, law = 'F99')
```

The parameters can also be defined and modified after the instantiation:

```
[5]: rc = pn.RedCorr()
      rc.E_BV = 1.34
      rc.law = 'S79 H83 CCM89'
```

$c(H\beta)$  and  $E_{B-V}$  are related through:

$$(1 - f_\lambda).c(H\beta) = 0.4E_{B-V}X_\lambda$$

applied to  $\lambda = 4861$ , with  $f_\beta = 0$ . so that, once one of the two parameters is defined, the other is also automatically defined; to output its value, enter:

```
[6]: rc.cHbeta
```

```
[6]: 1.9457101047071228
```

```
[7]: rc.cHbeta = 2.
      print(rc.E_BV)
```

```
1.3773891565431358
```

The reddening of a given spectrum is determined by using the ratio of two observed line intensities relative to the theoretical value, for example:

```
[8]: rc.setCorr(obs_over_theo=6.5 / 2.86, wave1=6563., wave2=4861.)
```

```
[9]: print(rc.cHbeta)
```

```
1.1134093776707101
```

Once a law and either  $c(H\beta)$  or  $E_{B-V}$  are defined, the correction for any wavelength is obtained by:

```
[10]: wave = 5007.0
      corr = rc.getCorr(wave)
      print(corr)
```

```
11.846898279423378
```

where **wave** can either be a single wavelength or a list or array of wavelengths.

The correction relative to the  $H\beta$  correction is given by:

```
[11]: corr = rc.getCorrHb(wave)
      print(corr)
```

```
0.9124210200172395
```

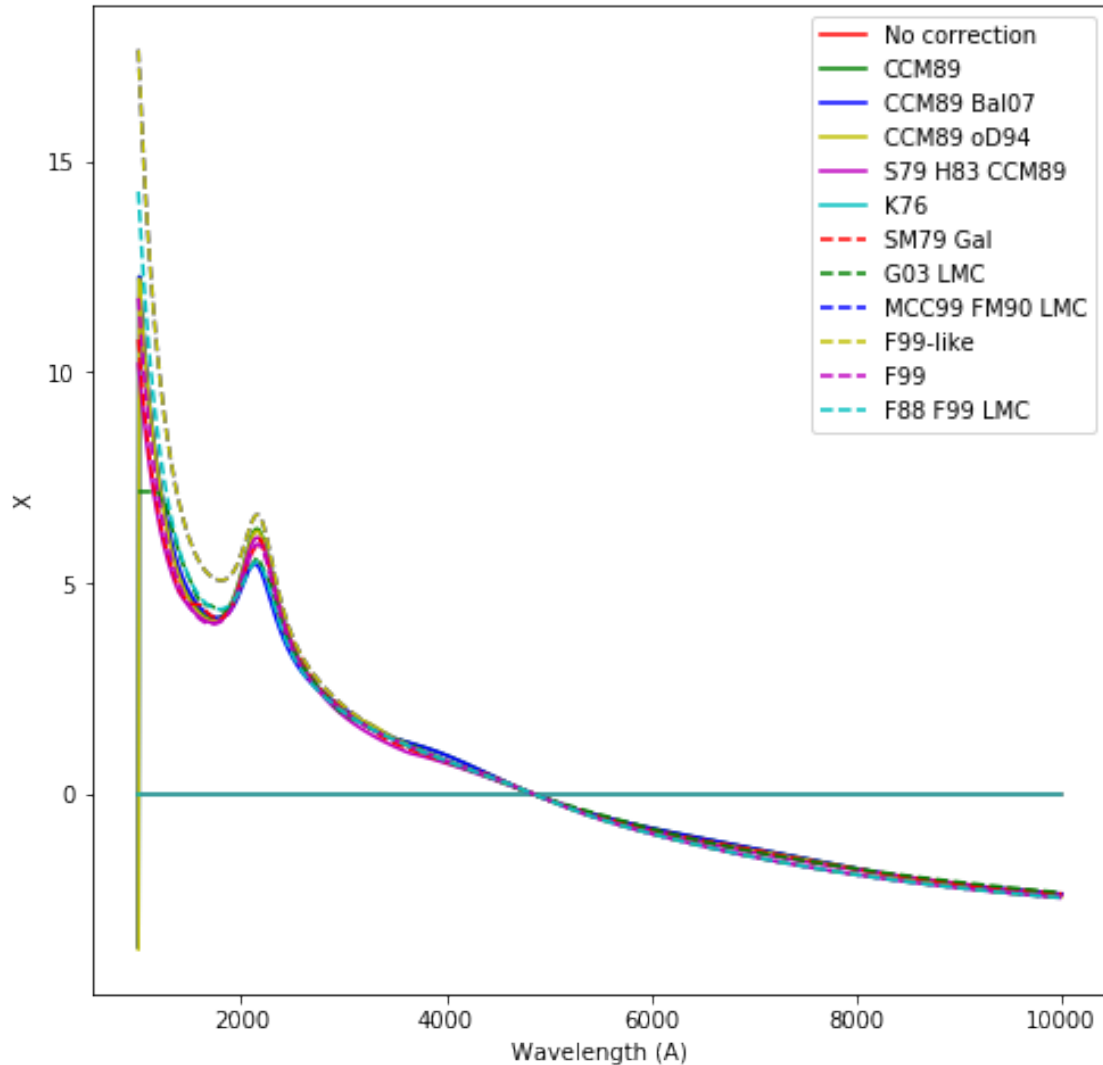
and the correction relative to any other wavelength (p. ej.,  $H\alpha$ ) is given by:

```
[12]: corr = rc.getCorr(5007., 6563.)
      print(corr)
```

2.07325069462699

The class includes a plotting tool to have a quick look at the different extinction laws:

```
[13]: f, ax = plt.subplots(figsize=(8,8))  
      rc.plot(laws = 'all', ax=ax)
```



A user-defined method can also be used. User-defined methods must accept 2 parameters: the first is the wavelength (or wavelength array), in Angstrom, and the second is an optional parameter (which can also be a list). The method must return  $X(\lambda) = A(\lambda)/E_{B-V} = R_V A(\lambda)/A_V$ . The correction is then:  $10^{0.4E_{B-V}X(\lambda)}$

Here is an example of a user-defined function:

```
[14]: def my_X(wave, params = [5000., 1., 2., 3.]):  
        return params[1] * (wave/params[0]) + params[2] * (wave/params[0])**-1 +  
        ↪params[3] * (wave/params[0])**-2  
rc.UserFunction = my_X  
rc.UserParams = [6000., 1., 5., 1.]  
rc.law = 'user'  
print(rc.getCorr(5007))
```

342.0931612017285