Fitting galaxy spectra with STECKMAP: a user guide

P. Ocvirk

April 23, 2010

STECKMAP stands for (STEllar Content and Kinematics via Maximum A Posteriori). It is written in Yorick, and altho some information is available on the official webpage and the tutorials, as well as by typing “info, sfit” (for instance) in the Yorick prompt, the information about the code and its use is rather sparsely distributed. This user guide is an attempt at concentrating this information into one single place and give the user a more efficient and useful experience with the code.

Contents

Chapter 1

What is STECKMAP?

STECKMAP is a software aiming at interpreting galaxy spectra in terms of their stellar populations: star formation history, age-metallicity relation, extinction are constrained. From these, a number of integrated quantities can be further computed, such as luminosity-weighted age and metallicity, and mass-weighted age and metallicity (respectively, LWAge, LWZ, MWAge, MWZ).

To do so, the observed spectrum is projected onto a time-sequence of models of single stellar populations, so as to determine a linear combination of these models, that fit the observed spectrum best (via a penalized χ^2). The weights of the various components of this linear combination indicate the stellar content of the population.

Since in practice observed spectra are altered by the kinematics and the instrumental signature, STECKMAP also determines a broadening function (BF), which is the convolution of the line-of-sight velocity distribution (LOSVD) of the stars and the instrumental point-spread function (PSF). Since the latter is usually well known the BF can be further interpreted in terms of the kinematics of the observed population. This is also done in a Maximum A Posteriori formalism.

Chapter 2

Installing the code

One needs to install Yorick and STECKMAP separately.

2.1 Installing yorick

You will thus need to first install yorick (http://www.maumae.net/yorick/doc/). This should in principle be painless, especially using the pre-compiled binary distributions. You should even be able to install it through fink (for Mac) or the linux equivalent. Also yorick is very light (much lighter than the STECKMAP package itself) so you wont have to worry about disk space.

It is important for this version that yorick be installed in the Yorick/ directory such as: $HOME/Yorick/

Successful installation should result in the yorick executable to have a full path similar to:

$HOME/Yorick/yorick-2.1/yorick/yorick (for yorick-2.1 that is). This note provides few guidelines for installing yorick. Refer to yorick's README for more, as it is more complete. On Macs, yorick can be installed through fink. I recommend using yorick through emacs. This allows for 2 very cool things to happen:

- enables emacs code editing features: colors, indentation, etc... i.e. a helpful coding environment.
- keyboard shortcuts: the buffer you are currently editing can be executed with Ctrl-X Ctrl-S.

Follow the yorick.el instructions to set this up. You will need to make a straightforward modification of your /.emacs. and inside yorick.el dont forget to provide the full path to the yorick executable (search for the string ”yorick-executable-name” and replace the adjacent string ”yorick” with the full path to the executable). You can check if yorick is correctly installed by launching it typing

```
bash-3.2$ yorick
 Copyright (c) 2005.  The Regents of the University of California.
 All rights reserved.  Yorick 2.1.05 ready.  For help type 'help'
> include, "demo2.i"
> demo2
```

A cool animation should ensue, greeting you to a successful install of Yorick.

2.2 Installing STECKMAP

You can get the latest package from here: http://www.aip.de/People/POcvirk/
Download the latest version. The versions are tagged as:
STECKMAP_laptop_ddmmyy.tar
or
STECKMAP_laptop_ddmmyyyy.tar
In your Yorick directory ($HOME/Yorick/), untar the STECKMAP archive

```
tar -xvf STECKMAP.tar
```

Some routines use the environment variable $HOME to define absolute paths to some objects, so be sure it is correctly defined by typing in a shell:

```
echo $HOME
```

2.3 Checking the successful installation with an example

Launch yorick by typing "yorick" on the shell command line. Once yorick is launched, load STECKMAP by typing

```
> include, "STECKMAP/Pierre/POP/sfit.i"
```

You will find a couple of example data in Yorick/Pierre/POP/EXAMPLES/. They are provided in order to give the user a taste of what can be done and how to proceed. Follow this small tutorial.

STECKMAP is a tool package aiming at constraining the stellar content and kinematics from galaxy spectra. To do so, you need two things:

- DATA
- Models against which to compare your data

The detailed use of the function will be explained in the further chapters. For the moment we just want to check STECKMAP is installed properly.

Data

See the path and name of an example file I set up by typing

```
> fV
```

fV is just a variable storing the path and name of the example file. To make things easier, it is loaded by default. Convert the example file by typing:

```
> a=convert_all(fV)
```

this will create the .pdb file, plot the data, and write the redshdift of the example galaxy if it is supplied in the fits keywords. Otherwise it will just assume 0, as in this case. Note that it will plot the 2d plate and the spectrum obtained by summing the whole 2d spectrum in the spatial direction. This is the default behaviour of convert_all.

A model basis is a structure as defined in "Pierre/POP/base_struct.i". This structure contains the sequence of SSPs in time and metallicity. As such, it is a data cube. The structure contains this cube and also the wavelengths, the resolution, the metallicity scale, and the ages of wach element of the basis.

SSP basis

To generate a basis, use the function bRbasis3:

```
> b=bRbasis3()
```

This will call bRbasis3 with all the defaults arguments. To get help and see information about the various arguments and their default values you can type

```
> help, bRbasis3
```

or

```
> info, bRbasis3
```

Here the default is a PEGASE-HR sequence of SSPs with 10 age bins from 10Myr to 20 Gyr with Salpeter IMF and Padova tracks. It is flux-normalized by default (see the STECMAP paper for details). PEGASE-HR has a resolution R=10000 and the wavelength coverage is $4000 - 6900$ Å. All these values can of course be changed when calling bRbasis3.

To have an idea of the looks of the basis you just generated, type

```
> ws
```

this is to clear the display

```
> plb, b.flux(,,1),b.wave
```

This will result in a nice plot of the basis for the first metallicity of the basis. It can be printed by typing:

```
> b.met(1)
```

since the metallicities are stored in b.met. Note that for computational reasons, the metallicities are renormalized. Hence, you can read the original metallicity of the first constant-metallicity slice of the SSP cube b.flux(,,1) by typing:

```
> Zrescalem1(b.met(1))
```

it should be 0.05.

Fitting engine

The fitting engine is called with the sfit command, which takes as arguments, the data, the basis and a bunch of options. Type

```
> help,sfit
```

to see the current documentation for sfit. To fit the data, the structure of which is stored in the variable a, using the SSP basis, the structure of which is stored in the variable b, type:

```
> x=sfit(a,b,kin=1,epar=3,noskip=1,sav=1)
```

The various options will be explained in the next chapter.

Chapter 3

Fitting your data

3.1 Preparing your data: convert_all

3.2 Preparing a sequence of SSP models: bRbasis

3.3 Running the fitting engine: sfit

Chapter 4

Interpreting the results

When the option sav= 1 is given to sfit, the results are saved. The various results files are created in the same directory as the initial data file, upon success of the fitting procedure. The .pdb file is the result of the convert_all procedure. It is a binary file containing the initial data, a wavelength vector, an error spectrum and sometimes a mask if it has been supplied in the proper way (such as when dealing with SDSS data). This chapter describes the other files available.

4.1 Description of the results files

The format of the files is text as follows:

1. variable name
2. length of vector
3. values
4. variable name
5. length of vector
6. ...

4.1.1 Stellar content

These are the files: res-SAD, res-MASS, res-SFR and res-AMR. They look like this (here for the res-MASS file):

```
 Ages (Myr)       10
        10.0000        23.2692        54.1455        125.992        293.173
        682.190        1587.40        3693.75        8595.06        20000.0
 Masses in each time bin       10
        39.4829        17.2968        5.35236        2.45517       0.997133
        5.72450        251.843        2432.88        16218.1        39665.3
```

The variables relative to the stellar content all refer to a time axis, which can be seen as lookback time or the age of the stellar component. It is always given first in these files. The (Myr) indicates the time axis is in Myr. Then comes the field of interest, here the masses. Here is a short description of the content of each file:

- **res-SAD**: The Stellar Age Distribution, i.e. the contribution in flux of each component to the observed spectrum. It is normalized so that the sum of the SAD is 1. It is the basic quantity STECKMAP is working with (rather than masses or SFR). In a typical STECKMAP run the SAD is shown in the top plot of the results panel.
- **res-AMR**: The Age-Metallicity Relation. This gives the metallicity $Z(i)$ of each component i of age $age(i)$, hence defining effectively an age-metallicity relation. In extragalactic astronomy, the metallicity is usually understood as a total abundance of metals, i.e. the fractional mass of everything which is not H or He. With this convention, in the models used here the metallicity of the sun is 0.02. Some authors will use 0.019 instead, but in practice this does not make a huge difference in the interpretation of the results, in particular as long as one does not embark into computing abundances of individual elements. In a typical STECKMAP run the AMR is shown as the middle plot of the results panel.
- **res-MASS**: This file gives the stellar mass as a function of the age. Note that it is understood as an "initial mass", i.e. the mass the given component had at the time of its birth. No mass loss is taken into account here. The masses for each time bin i are simply computed as:

  $$mass(i) = \frac{sad(i)}{M/L(age(i), Z(i))} \tag{4.1}$$

  where $M/L(age, Z)$ is the mass to light ratio of a stellar population of age age and metallicity Z. Again since no mass loss is taken into account this refers to the *initial mass*, i.e. it only accounts for the dimming of the population, not for the decrease/recycling of its stellar mass. The masses are given in principle in solar masses but the normalization is arbitrary: the relative masses (i.e. between the various bins) are correct but the actual normalization results from the fact that *sad* is normalized to unity, i.e. $\sum_i sad(i) = 1$.
- **res-SFR**: The Star Formation Rate (SFR) as a function of lookback time. It is obtained as:

  $$SFR(i) = \frac{mass(i)}{\Delta t(i)}\,, \tag{4.2}$$

  where $\Delta t(i)$ is the duration or extent of the age bin i. Note that this is *not* $t(i+1) - t(i)$ but is rather computed as like $\Delta t(i) = t(i+1/2) - t(i-1/2)$. The SFR is given in $M_\odot/yr$ but is not physically normalized. As for the masses, the relative variations are correct but not their absolute values, and the actual normalization is essentially inherited from that of the masses *mass*.

4.1.2 Kinematics

The broadening function of the data results from the convolution of the LOSVD and the instrument's PSF or LSF. However, I will often abusively call it the LOSVD, and talk about kinematics eventhough I should be talking about kinematics + instrumental characteristics. The broadening function is stored in the **res-LOSVD.txt** file. It begins as follows:

```
 v (km/s)      369
       -1003.75       -998.323       -992.892       -987.460       -982.029
       -976.597       -971.165       -965.733       -960.301       -954.869
       ...
```

This describes the velocity range (in km/s) on which the LOSVD has been computed. Next come the actual values of the broadening function for each of these velocities:

```
 g(v)      369
    1.57828e-06    1.62290e-06    1.65155e-06    1.65992e-06    1.63955e-06
    1.61256e-06    1.54141e-06    1.47136e-06    1.33613e-06    1.13671e-06
...
```

The LOSVD or BF $g(v)$ is normalized to unity, i.e. its sum is equal to 1.

4.1.3 Spectra

A number of spectra are available in the **res-spectra.txt** file. In order of appearance are the wavelength range (in Angstrom), the data, the best fitting model, the weight vector, and, if the option epar=3 has been used, the non-parametric extinction curve (maybe calling it non-parametrically adjusted continuum (i.e. NPAC would make more sense...). Although these are readily plotted in a usual STECKMAP run, this results file should allow the user more flexibility in displaying/examining his fit.

4.2 How to improve your fit ?

4.2.1 Masking bad pixels/emission lines

4.2.2 Dependence on hyperparameters

4.3 Robustness of the results: running Monte Carlo experiments

Chapter 5

Tweaking STECKMAP

In some cases you will want to use STECKMAP in a way that was not initially planned by the author. This chapter gives a few tricks and corresponding yorick codes to do so.

5.1 Adding a constraint on the AMR

5.2 Fitting the spectrum with SSPs + power-law

Appendix A

Common emission lines in the optical

Emission lines of a variety of chemical elements are usually seen in optical spectra of galaxies. Here is a compilation of the most common ones. It is not meant to be exhaustive but will help the user to design masks efficiently. Wavelengths are given in the air (**check that!**).

Emission line	Wavelength (air, Å)	Ref.
Hα	6563	(2)
Hβ	4861	(2)
Hγ	4340	(2)
Hδ	4101	(2)
Hϵ	3970	(2)
Hζ	3889	(2)
HeI	5876	(1)
[O I]	6300	(1)
[O II]	3727	(1)
[O III]	4363	(1)
[O III]	4959	(1)
[O III]	5007	(1)
[N II]	6584	(1)
[N II]	6548	(1)
[Ne III]	3869	(1)
[S II]	6716	(1)
[S II]	6731	(1)

Table A.1: Common emission lines in optical spectra. References: (1): http://www.mpa-garching.mpg.de/SDSS/DR7/raw_data.html, (2): http://en.wikipedia.org/wiki/Balmer_series